Logic and Accuracy Test Reporting Form

Pursuant to Elections Code section 15000, no later than seven days prior to any election, the election official shall conduct a test or series or tests to ensure that every device used to tabulate ballots accurately records each vote. The exact methods employed in this test shall conform to the voting procedures for the specific voting systems, as adopted by the Secretary of State as well as section 20279 of the California Code of Regulations.

A. Election Official:

Jurisdiction

Ventura County

Contact Name and Title

Michelle Ascencion, County Clerk-Recorder & Registrar of Voters

Contact Phone Number

805-654-2664

Contact Email Address

elections@ventura.org

B. Election Information:

Title of the Election

November 5, 2024, General Election

Date of the Election

November 5, 2024

Date of Notice to the Public

September 19, 2024

Date of Logic and Accuracy Testing

September 24, 2024 to October 1, 2024

C. Election Technology:

Certified Voting System used by the Jurisdiction

Dominion Democracy Suite 5.10A

Certified Ballot on Demand System used by the Jurisdiction, if applicable.

Dominion MBP 5.10A

Description of any technical issues and/or mitigation, if applicable, that occurred during the conduct of the logic and accuracy test.

During Pre-LAT, we identified a configuration change for the Hi-Pro ICCs. Specifically, the paper sensor setting was adjusted from 'near' to 'far' to reduce sensitivity to paper placement in the output feed. This change also enabled the counter head to move to a higher position, improving overall scanning efficiency.

Logic and Accuracy Test Reporting Form

Pursuant to Elections Code section 15000, no later than seven days prior to any election, the election official shall conduct a test or series or tests to ensure that every device used to tabulate ballots accurately records each vote. The exact methods employed in this test shall conform to the voting procedures for the specific voting systems, as adopted by the Secretary of State as well as section 20279 of the California Code of Regulations.

Additional space to provide further description of any technical issues and/or mitigation, if applicable, that occurred during the conduct of the logic and accuracy test.

N/A

ATTESTATION AND SIGNATURES - *"I hereby certify that the logic and accuracy test has been performed, that the test results have been compared with predetermined correct totals for each office and ballot measure, and that the cause of any discrepancy was found and corrected. Additionally, I hereby certify that the logic and accuracy test was conducted pursuant to the requirement of Elections Code section 15000 and that the information submitted on this form is true, accurate and complete."*

Michelle Ascencion

Signature

Michelle Ascencion

Name (printed)

10/1/2024

Date

DominionHistory

Peter Bernegger

Worth Every Minute of Your Time: Obama And Dominion Voting Co.'s shocking history. Found the original source, article is publish in my tweet below, and here at this link: https://defyccc.com/shocking-history-of-dominion-voting/… As I've been saying for years it is sovereign fraud, i.e. our own government is stealing our elections. There is a long list, a few on it: DHS, CISA, CIS, Crowdstrike, Falcon Endpoint Services, Cradlepoint, the corrupt ES&S voting systems, the extremely corrupt Dominion Voting systems, Carlyle Group, David Rubenstein, Eric Holder, the Defense Intelligence Services (DIS), the Defense Intelligence Agency (DIA), the CIA, the FBI, Albert Sensors (whom CISA funds and implements) and 50 others.

https://defyccc.com/dominion-voting-systems-corp/

Dominion Voting Systems Corp

November 8, 2020dems, Dominion, elections fraud

Dominion Voting Systems Corp. ("Dominion") is the company behind the voter tabulation software, linked to cases of switching votes from President Donald Trump to Joe Robinette Biden. Dominion was founded in 2002-2003 in Canada, and remained there until at least 2009, when it opened offices in the US. Its current domain DominionVoting[.]com has been saved in archive.org since 2006. Dominion has always been in the business of election manipulation in favor of the progressives. It was a wet dream of progressives – voting software which does not only count votes, but **mobilizes** voters. A familiar jargon, isn't it? From the Dominion's website in July 2009:

*"**Turning constituents into voters [headline]***

There are those who always vote, those who never vote and those who may or may not vote. Maximizing voter turnout in the third group is the holy grail of the electoral official."

It is a holy grail of a leftist agitator. The US law recognizes a citizen's freedom not to vote as equal to the voting for a candidate of his or her choosing. Some people do not vote because they do not care, some because they do not know which candidate to choose, and some because they are angry at the society. Abstractly speaking, reluctant voters are not helpful to democratic process, just like reluctant jurors are not helpful for court proceedings. Practically, voter mobilization (from the word *'mob'*) is a tactic of the Left. People who do not care, or lack knowledge, or are angry, can be easily convinced to vote for the Left.

*"There are two primary barriers: voter apathy and a lack of voter **mobilization**. There is no easy answer to the former, but there is a way to overcome the latter. **Dominion Democracy™** takes voting to the voter."*

Yes, taking voting to the voter! It has been widely implemented by Democrats in the US. California sends ballots to everybody. In some places, a person who did not vote before the end of the election day,

might be visited by Democrat thugs. I can only imagine: “Nice kids you have here. Would you mind filling in your ballot and giving it to us? You can even do that secretly, and to seal the envelope. We will see your vote at the precinct, when we open envelopes.”

The text continues:

“Remote voting takes this idea a step further. Via traditional mail or the Internet, it takes polling right into the home or office, and across the globe.”

Vote over the Internet, across the globe! Vote from Pyongyang. Vote from Beijing. Vote from Moscow. Otherwise, prove you did not vote from Moscow, and that Moscow did not hack these internet votes. The Internet is everywhere; proxy servers, virtual machines, and VPNs allow anybody with minimal technical expertise to appear to be in any location. To be clear, voting over the Internet has not been implemented in the US. Also, the current Dominion software has little in common with one offered in 2009.

On its main page, Dominion described its software as:

“This [software] suite allows you to choose from a number of unique tools to custom build your perfect election. The tools are efficiently integrated into a simple system that you can run easily from one chair.”

Though incredulous sounding, apparently the offer “to custom build your own election” was accepted. This is when the ruling Democrat party took Dominion under its wings before the 2010 midterms elections. Before this, the company was a little shifty outfit, like Crowdstrike in 2016. Then, suddenly, it had a meteoric rise, and the Democrats started winning elections in highly unlikely places.

Dominion develops its software in many different places in the world. In 2016, it went on record stating that it had been developing its software in Serbia for the previous 10 years. The US was at war with Serbia in 1990’s, and there is no reason to expect its citizens to have a favorable attitude toward the US. Why then did Dominion select Serbia for software development?

I would not trust Dominion to replace a lightbulb in my bathroom, much less with voting machines. Unfortunately, it is the biggest supplier of the voting software and hardware in the US, serving about 40% of the voters (according to the claims on its website).

Vote tabulation in all states using Dominion software should be reviewed, including those where Democrats appear to have a large margin. If its software can change 6,000 votes, as it did in one county in Michigan, it can change any number of votes in every place it was used. Dominion Voting Systems software is also used in Pennsylvania, Georgia, Wisconsin, Nevada, Minnesota and in Maricopa County, Arizona. Democrat officials responsible for conduct of elections have selected this untrustworthy and compromised voting systems vendor, probably because it is biased in their favor.

https://defyccc.com/shocking-history-of-dominion-voting/

Shocking History of Dominion Voting

November 10, 2020cybersecurity, dems, Dominion, elections fraud

2020-11-15 update: *since 2009, Dominion Voting Systems operated from* ***215 Spadina Ave., Toronto, ON, M5T 2C7,Canada*** *– an office space of the radical Tides Foundation. This building houses (or housed until a few months ago) a Toronto office of Tides Canada and a Tides' incubation space for leftist groups.*

Dominion Voting Systems Corp. is the Canadian company behind the ballot switching software.

Dominion was founded in 2003, with a mission **to provide electronic voting systems friendly for progressives**. Because of such partisanship, it languished with almost no customers for the next 5-6 years, until the Obama administration came to power. In 2010, the Obama administration confiscated electronic voting systems assets (software, intellectual property, manufacturing tools, customer base, etc.) from two established American companies, and gave them to Dominion. At the same time, Dominion got some employees and assets from a foreign EVS company, tied to Hugo Chavez.

Its software has been used by some 40% of the voters in these elections, mostly by Democrat-controlled states and election commissions. Apparently, no protections were put in place against ballot switching, deletion, or creation. According to Dominion's own website, it software was used in "battleground" states and the largest Democrat states, including MI, GA, AZ, NV, NM, CO, AK, UT, NJ, CA, NY.

Dominion Early History

Dominion Voting System Corp., was founded in Canada in 2002-2003 with an openly progressive mission – to develop electronic voting software which would not just process ballots, but also "mobilize voters" – a popular slogan of the Left.

It is not clear what products or services the company has developed. It found almost no buyers, until Obama was elected in 2008. In 2009, New York ordered a few dozens of systems from it. In 2010, Obama's DOJ (Holder – Mueller) took the EVS unit, purchased from Diebold, away from the market leader ES&S, and gave it to Dominion. This gift included the installed base of about 30% of the US electronic voting systems (EVS) market. Within two weeks, Dominion also acquired Sequoia, which was formally spun from Smartmatic, but ties between these two companies remained. Smartmatic is a UK based EVS vendor, whose software was used by Chavez to "win" the Venezuelan referendum in 2004. Smartmatic's unit Sequoia faced troubles in the US. Those troubles quickly ended when its assets were purchased by Dominion.

Thus, the new Democratic party created a pocket pet corporation, gave it the lion share of the US electronic voting systems market. Dominion is ideologically aligned with the Democratic Party, owes it everything it has, dependent on it for most of its business, and needs it in power to avoid prosecution for corruption. Sounds like a conflict of interest.

Electronic Voting

Using electronic voting machines has always been controversial. The pros for electronic voting – saving working time of the ballot counters – are minuscule. The cons however are infinite. Because software is inherently complex, non-transparent, and volatile, there is always a risk of significant errors. There are also suspicions and doubts about election results. The complexity of software and hardware on which voting machines run has been continually increasing, aggravating these concerns.

At the beginning of 2009, there were four major US EVS suppliers: ES&S, Premier (a unit of Nixdorf-Diebold), Sequoia (linked to Smartmatic), and Hart Intercivic. The market size was a few hundred million dollars a year and growing. EVS vendors competed among themselves and against traditional pen and paper voting. There were no barriers to entry for other competitors, other than government's regulations.

Electronic voting, which sounded like a good idea in the 1980s is so no more. Electronic voting machines and their vendors were under criticism for many years. In 2007-2008, this criticism materialized in the SEC, DOJ, and states lawsuits against the voting machines vendors. Diebold was catching flack for having a prominent Republican party supporter among their top executives. It spun its EVS unit as a separate company Premier, and was looking for a buyer. The existing vendors were burdened with liabilities, including DOJ investigations. This opened up an opportunity for the Obama administration.

Technical Vulnerabilities of EVS Systems

The voting software developers can easily insert code, changing numbers in favor of or against one candidate. No hacking is necessary. The malicious code can be designed to pass tests and to be triggered only at the time of a real election, automatically or manually. Both cases are possible even the the machine is disconnected from the internet and has no ordinary I/O devices. The malicious code can be activated manually in real time by inserting a ballot or another paper with a pre-defined QR or image code. An audit of the source code is necessary, but not sufficient. Dominion software runs on Windows, and the malicious code can be hidden in any part of the operating system. Malicious code can be hidden in the firmware, too.

If a state wants to take risks and to rely on testing and the source code audit, they should be conducted with the participation of technically competent representatives of both parties. If the system passes testing and auditing, the machine image must be securely stored. All supplied machines must have exactly the same hardware and the software as the audited system.

As far as I know, thorough tests and source code audits are conducted very rarely, if at all. Further, the vendors are not required to use only the audited image, and are allowed to update the software almost at will. That means that election commissions are forced to blindly trust the vendors. Blind trust is always wrong and invites abuse. But even "trust but verify" is applicable only to trustworthy vendors. Dominion Voting is the opposite of trustworthy.

The only real solution to the vulnerability of EVS is not to use them at all. Manual ballot counting has no software vulnerabilities, and is much cheaper. Virginia appears to be the only state that decided to use only manual ballots.

How Dominion went from nothing to everything in two weeks

In September 2009, ES&S acquired Premier [8], without any objections from the DOJ. But in March 2010, the Obama's DOJ (Eric Holder – Robert Mueller) forced ES&S to "sell" Premier's assets to Dominion, but to keep its liabilities. In addition, ES&S was forced to license to Dominion some of its software, in perpetuity and free of charge. The pretext for the DOJ action was antitrust.

The "Sale"

This is how the assets transfer was structured, per DOJ [1] (March 8, 2010).

"WASHINGTON — The Department of Justice announced today that it will require Election Systems & Software (ES&S) to divest voting equipment systems assets it purchased in September 2009 from Premier Election Solutions Inc. ***in order to restore competition****. The assets to be divested include the means to produce all versions of Premier's hardware, software and firmware used to record, tabulate, transmit or report votes, including the Assure 1.2 system, and a license to better serve disabled voters. The department said that today's settlement will restore competition in voting equipment systems in the United States… "*

"In order to restore competition" sounds funny, because the same document also required ES&S to not compete against the buyer (with exceptions).

"… the acquisition substantially reduced competition as it combined the two largest providers of systems used to tally votes in federal, state and local elections in the United States. ES&S's acquisition of Premier made ES&S the provider of more than 70 percent of the voting equipment systems in the United States. The department said that because the cash value of the deal between ES&S and Premier was $5 million, far below the mandatory reporting threshold for mergers under the Hart-Scott-Rodino Antitrust Improvements Act of 1976, the department's investigation of the transaction did not begin until the companies had combined their assets and dismantled many of Premier's operating divisions."

Sounds like a poor pretext. The DOJ has been investigating these companies even before the merger, and was aware of it. Further, the DOJ does not allege that the merger has not been reported. Even so, why not simply demand unrolling the merger? The DOJ provides a poor excuse to demand divestiture rather than a normal unrolling.

"Under the terms of the settlement, ES&S must divest all of the intellectual property associated with all versions –past, present and in development –of the Premier voting equipment systems to another company. ES&S also must divest all Premier tooling and fixed assets, as well as inventory of parts and components. In order to allow the divestiture buyer to better serve disabled voters, ES&S must also grant a fully paid-up, irrevocable, perpetual license to use the AutoMARK, ES&S's ballot marking device for which Premier had a limited license prior to the acquisition. The buyer of the divestiture assets will have the right to modify and improve both Premier products and the AutoMARK."

Thus, the Obama's DOJ stripped ES&S not only acquired Premier assets, but also coerced it to license rights to its pre-merger product.

"ES&S must sell the divestiture assets to a buyer approved by the department."

This is not selling. This is confiscation multiplied by corruption.

"The settlement prohibits ES&S from bidding on new voting equipment system contracts using the Premier equipment. [transferred to Dominion]"

Wait, didn't they say that the purpose was to increase the competition?

"The department also required that ES&S grant the divestiture buyer an opportunity to compete to provide services to Premier customers currently under contract with ES&S, giving customers the option to switch to the divestiture buyer or to remain with ES&S … ES&S also must provide access to knowledgeable Premier employees and agree to offer a supply agreement to allow the divestiture buyer time to establish its own manufacturing of voting equipment systems."

The approved divestiture buyer, Dominion Voting, is not mentioned in this press release. But this quote shows that the DOJ has already determined the "approved buyer," and knew that it had no manufacturing base.

After the "Sale"

Dominion announced the acquisition of the Diebold products on May 19, 2010 [2] and the acquisition of Sequoia Voting assets on June 4, 2010 [3]. Dominion also hired much of its personnel, probably retaining ties to extremely sketchy Smartmatic. Sequoia/Smartmatic systems had been used in the Venezuela 2004 referendum, which Hugo Chavez "won". Smartmatic is a British company with Hugo Chavez ties, headed by "Lord" Malloch-Brown (former UN Deputy Secretary-General, UNDP, UNHCR, VP of Soros' Quantum Fund, and Vice Chair of Soros' Open Society Foundation) [7], and linked to electoral scandals all over the world [5].

In August 2009 (corrected), the rough breakdown of the EVS market in the US was (per Brad Friedman):

40% ES&S

30% Diebold/Premier

20% Sequoia/Smartmatic

10% Hart Intercivic

0% Dominion Voting

Less than a year later, after the "antitrust" actions of Obama's DOJ, it became:

50% Dominion

40% ES&S (restricted in competing against Dominion)

10% Hart Intercivic

Thus, the DOJ's actions did the exact opposite of its words.

An elections system vendor should be non-partisan, in a demonstrable way. Dominion is not just partisan, but hyper-partisan in favor of the Democrat party, or even its pocket vendor.

Dominion has many more ties to the Democrat party and its prominent supporters in the US and abroad, which are not covered in this article.

Software Development in Serbia

Dominion develops much of its software in Belgrade, Serbia. Russia is a close friend to Serbia, if not its only one. If anybody sincerely thought that Putin wanted to hack American elections, their first location of interest would be the offices of Dominion Voting in Belgrade, rather than the Trump Tower in New York.

By the way, Serbian and Russian languages use the Cyrillic alphabet. Most letters have the same Unicode encoding in Serbian and Russian (the Basic Multilingual Plane, range 0410-04FF). If any election officials found Cyrillic text on a Dominion voting machine in 2016, it was probably left by its developers in Serbia.

Remarks

This is the Agreement between Michigan & Dominion, including specs of many Dominion products (PDF, 161 pages). Wi-Fi connection and even a dial up modem are offered as an option.

Some of the companies referenced here as foreign based or foreign originating re-registered in the US.

Some References

[1] *Justice Department Requires Key Divestiture in Election Systems & Software/Premier Election Solutions Merger,* justice.gov, March 8, 2010

[2] *Dominion Voting Systems, Inc. Acquires Premier Election Solutions Assets From ES&S*, press release, May 19

[3] *Dominion Voting Systems Corporation Acquires Assets of Sequoia Voting Systems*, press release, June 4, 2010

[3b] *On Heels of Diebold/Premier Purchase, Canadian Firm Also Acquires Sequoia, Lies About Chavez-Ties in Announcement* – contemporary commentary

[4] *Marcos warns of 'another Smartmatic situation'* – Smartmatic was accused of election fraud in the 2016 elections in Philippines

[5] *Smartmatic* in Wikipedia, November 9, 2020 (not verified)

[6] *Sequoia Voting Systems* in Wikipedia, November 9, 2020 (not verified)

[7] *"Lord" Malloch Brown* in Wikipedia, November 8, 2010 (not verified)

[8] Diebold Sells U.S. Elections Systems Business to ES&S, press release, September 3, 2009

[9] Dominion Voting Systems Corp – discussion of Dominion's ideology and highly partisan offers

[10] Dominion Voting Systems' profile in Bloomberg

CIA Operative Gives 27-Year Timeline On Stolen Elections: Connects Smartmatic, Dominion, Software Softer (Bizta) with Venezuela's Hugo Chavez

Written by Michael Hernandez

MOUNTAINTOPTIMES, OCT 23

BOCA RATON, FLORIDA—Former CIA operative (1992-2005) Gary Berntsen has "gone public" (on YouTube) with a 16-minute video released Oct. 19, a 27-year timeline and website (StolenElectionsFacts.com) outlining the connections between Smartmatic (election software), Dominion (voting machines), Software Softer (Bizta) with Venezuela's 52nd President Hugo Chavez (1999-2013) who led the Bolivarian Revolution, a socialist political program inspired by both Simon Bolivar and Fidel Castro.

The intent of these inter-connected companies is to control elections worldwide (both global and in America) and change voting results without anyone in the public knowing.

Venezuela has mastered how to steal elections when candidates are within three to five percent of each other. The voting fraud is detected when candidates win by massive amounts. Working with Venezuela is China, Cuba and Serbia.

Berntsen shares how swing-state voting is monitored by servers in Belgrade, Serbia.

He also states how fraudulent voting was not only done in Venezuela elections but also in the Philippines. In his YouTube video, Berntsen shares how Dominion Voting Machines are manufactured in China and shipped to Taiwan. He shares how the voting election soft code is controlled by the Venezuela regime which is now run by President Nicholas Maduro, who has a U.S. indictment of 15 million dollars on his head. Six other cabinet officials are now also indicted.

Berntsen, is a veteran of U.S. Air Force and served in intelligence operations including Afghanistan where he was Chief of Herzbollah Operations and concluded his service in the CIA in Latin America. He then started an investigation of the largest international criminal cartel in the world—Cartel de los Soles (The Cartel of the Suns) involved in international drug trade headed by high-ranking members of the Armed Forces of Venezuela. They have stolen more than one trillion dollars and embezzled 500 billion dollars to fund their operations. The Cartel of the Suns moves 25-40 metric tons of cocaine every month into the world and controls a dozen countries and leaders.

Smartmatic Corporation, the voting software company incorporated in Delaware on April 11, 2000 but gives its principal physical address as 19591 Dinner Key Drive, Boca Raton, Florida. Likewise, Bizta Corporation, an American company incorporated in Delaware on April 12, 2000 by the same shareholders of Smartmatic also declared its principal office at the same address as Smartmatic Corporation. Bizta on Jan. 24, 2001 filed for incorporation to commence operations in the State of Florida with tax address in the State of Delaware. Meanwhile, Software Softer is incorporated in Caracas, Venezuela by Antonio Jose Mugica Rivero, Antonio Mugica, Jr. and Antonio Julian Mugica Sesma (Antonio Mugica, Sr.).

Basically, the election software companies and the voting machine companies used in

U.S. elections were all connected to deceased Venezuela dictator Hugo Chavez. A Reuters fact-check found that Dominion systems were used in at least 24 states for the 2020 elections and California predominately uses the Dominion Voting Systems. Venezuela owns the source code for Smartmatic which is stored in a vault in the Venezuela central bank.

According to Berntsen, "President Chavez decided to weaponize elections beyond Venezuela and entered Cook County in Illinois and the State of New Jersey in 2006. In 2005, Sequoia Voting Systems had been conducting elections for over 100 years and had a 22 percent voting share of the U.S. market. They were purchased by Smartmatic and the source code was put into these voting machines.

"Dominion Voting Systems—a little company in Toronto, Canada that had only managed one local election was found by the Smartmatic executives and arranged for Dominion to purchase Sequoia and this allowed Dominion to inherit the source code from Smartmatic—the source code owned by the Venezuelan regime.

"Smartmatic and Dominion would ultimately sign an agreement providing Dominion with the U.S. market and Smartmatic with the international global market. Two exceptions are that Smartmatic does elections in California—the Los Angeles County market and Puerto Rico.

"More than 200 software engineers worked side-by-side in Caracas, Venezuela and their efforts for more than a decade was to perfect the techniques of stealing all elections and defeating audits. In 2024, Smartmatic exited Venezuela to allow the regime to maintain control of the worldwide election global market."

"Smartmatic has built a factory just outside of Beijing, China and now ships the hardware to a warehouse in Taiwan. The hardware is marked as manufactured in Taiwan and marketing for use in U.S. elections against the law. Dominion Voting Systems manages almost all the elections in U.S. swing states which determines who wins the presidency. The evidence of the source codes in both Smartmatic and Dominion Voting Machines are owned by the Venezuela narco-regime.

"Every citizen needs to be asking where is the DOJ, the FBI and the CIA? Is our national security apparatus defending national security or enforcing the law?

"Dominion has moved its research and development and servers to Belgrade, Serbia and they alter elections as directed by the Cartel de los Soles. In 2020, when faced with calls to investigate elections, a conference call was done with Smartmatic and Dominion to address the public concerns and assure everyone that there were no irregularities. It is shocking that the criminals were consulted to respond to the American public.

"In August, 2024, three former Smartmatic executives were indicted in Florida for the 2016 elections in the Philippines, including the founder and current president of Smartmatic. He paid $8.5 million dollars in bail. The bribery was paid to alter election results.

"We have the source code. We will surrender it to appropriate authorities. It can easily be matched with other systems to a family of altering elections."

Berntsen stated that he has tried to release his investigative findings to the Federal Bureau of Investigation two years ago in Washington, D.C. and was told "to leave" D.C. so that the "FBI would not destroy our efforts." He later went to a U.S. attorney and two U.S. assistant attorneys from the

Department of Justice and they said they forwarded his investigations to the Office of Public Integrity which never did follow-up with any investigation of their own.

Both Fox News and Newsmax settled with Smartmatic and Dominion Voting Machines when they were sued by the two companies The Fox News settlement with Smartmatic for defamation was for $787.5 million dollars. Originally the lawsuit was for $2.7 billion dollars. The settlement for Newsmax has not be revealed to the public but the original lawsuit was for $1.7 billon and the settlement is expected to be between $370 to $400 million.

To view the 27-year time frame released by StolenElectionsFacts.com go to:

https://stoleneletionsfacts.com/press/:

Key Dates:

• 1997 (Oct. 15): Antonio Mugica form Software Softer (Bizta) in Caracas, Venezuela.

• 2000 (Sept. 7): Florida's "hanging chads" issue complicates Bush vs. Gore balloting.

• 2003 (Jan. 14): Dominion Voting Systems is created in Canada.

• 2003 (June 10): Chavez regime takes 28 percent of Bizta research and development.

• 2004 (April 20): Miami Herald: Smartmatic machines 'could be used to manipulate' voter tally.

• 2004 (Sept. 21): 'Mathematically impossible' Smartmatic results.

• 2005 (March 9): Smartmatic buys Sequoia Voting Systems for $16 million.

• 2005 (May 26): Cook County, Illinois picks Sequoia with Smartmatic system.

• 2006 (April 27): Chicago Tribune: 'Angst about Sequoia' due to its owner's Venezuela roots.

• 2006 (July 26): California commissioned study of the Smartmatic-Sequoia systems source code and its documentation does not comply with the standards.

• 2006 (Oct. 25): California de-certified Sequoia systems, citing institutional flaws in source code.

• 2006 (Nov. 8): Sequoia/Smartmatic split creates 100 percent American-owned and independent company.

• 2008 (Aug. 11): Smartmatic & Dominion partner in Philippines.

• 2010 (June 4): Dominion buys Sequoia assets, including all software (including Smartmatic software).

• 2011 (May 2): Dominion begins operation in Belgrade, Serbia.

• 2012 (Dept. 12): Smartmatic announces lawsuit against Dominion.

• 2013 (May 1): Smartmatic loses lawsuit against Dominion for control of Puerto Rico elections.

• 2013 (June 11): Lawsuit filed in Philippines over Smartmatic 'irregular, graft-ridden contracts.'

• 2016 (Sept. 28): CA Democratic Congressman Ted Lieu sees national election risk in swing states and counties.

• 2018 (Aug. 27): Obama election expert: 11-year-old hacker shows ease of sabotaging U.S. votes.

• 2019 (June 8): Huawei seen as 'threat' to democratic societies.

• 2019 (July 30): Georgia buys 30,000 Dominion machines for 2020 vote.

• 2019 (Sept. 23): California Rep. Susan Lofgren introduces SHIELD Act to require inspection of election source codes.

• 2020 (March 3): Politico: Smartmatic flaws in LA County 'could provide a gateway' for fraud.

• 2020 (Oct. 26): PBS asks if Georgia's new voting machines will make things worse.

• 2020 (Nov. 15): Smartmatic USA chairman joins Biden-Harris transition team.

• 2020 (Dec. 4): Smartmatic Chairman named president of Soros Foundation.

• 2023 (Jan. 12): Texas Attorney General seeks new laws to fight election crimes.

• 2024 (Aug. 8): Feds criminally charge Smartmatic president.

• 2024 (Aug. 9): Indicted: Smartmatic president & COO, brother-in-law of chairman and CEO, other executives.

• 2024 (Aug. 12): Politico: Best hackers find voting machine vulnerabilities but no time to fix them.

Berntsen will be featured on the Ralph Pezzulo book on "Stolen Elections: The Plot To Destroy Democracy" to be released by Amazon on Dec. 10.

Elon Musk Gives Support To Dominion Voting Systems Fraud While In Pennsylvania For Trump Campaign Event

"When you have mail-in ballots and no proof of citizenship, it's almost impossible to prove cheating," Elon Musk said, responding to an audience member's question about election fraud. "Statistically there are some very strange things that happen that are statistically incredibly unlikely. There's always this question of, say, the Dominion voting machines.

"The last thing I would do is trust a computer program," he said in his remarks at the Oct. 17 Philadelphia campaign event for Donald Trump. "This election is going to decide the fate of America and, along with America, the fate of Western civilization," Musk told the crowd as he stood in front of an American flag."

Musk has given nearly $75 million to his pro-Trump America PAC, which he established this year. He joined Trump onstage when the former President returned to Butler, Pennsylvania at the site of the first assassination attempt against him in July.

Gary Bernsten Video:

https://x.com/atensnut/status/1849077654135177272

https://defyccc.com/dominion-voting-systems-is-caught/

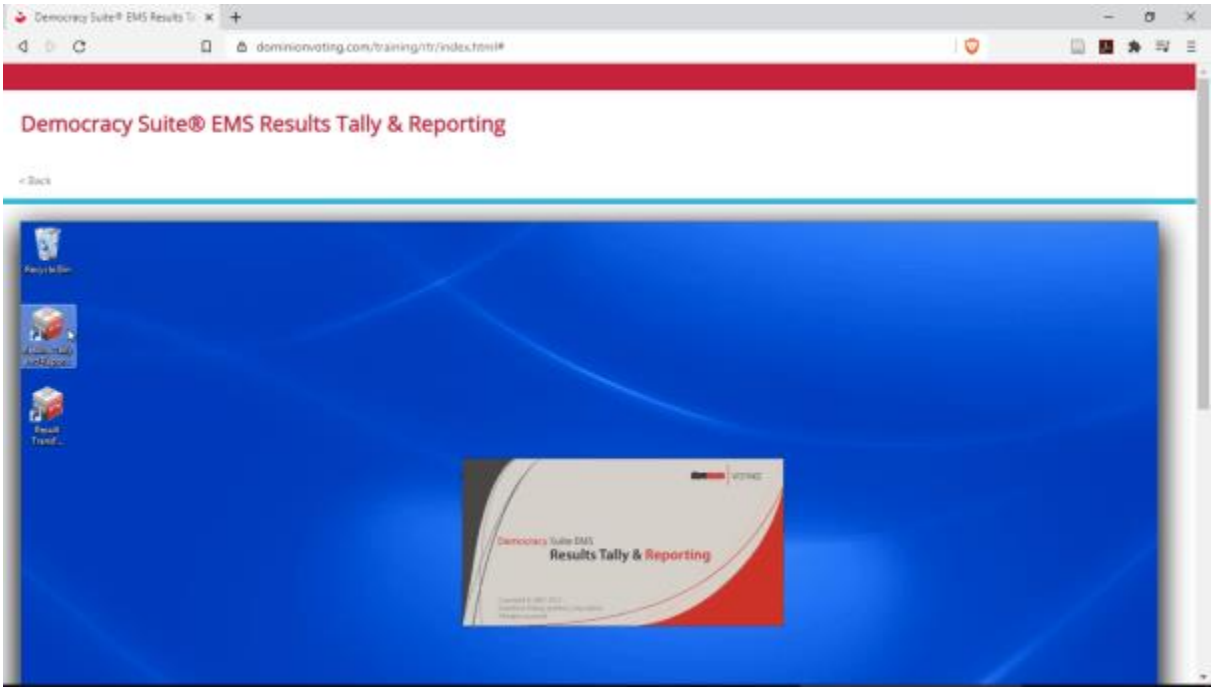
Dominion Voting Systems is Caught

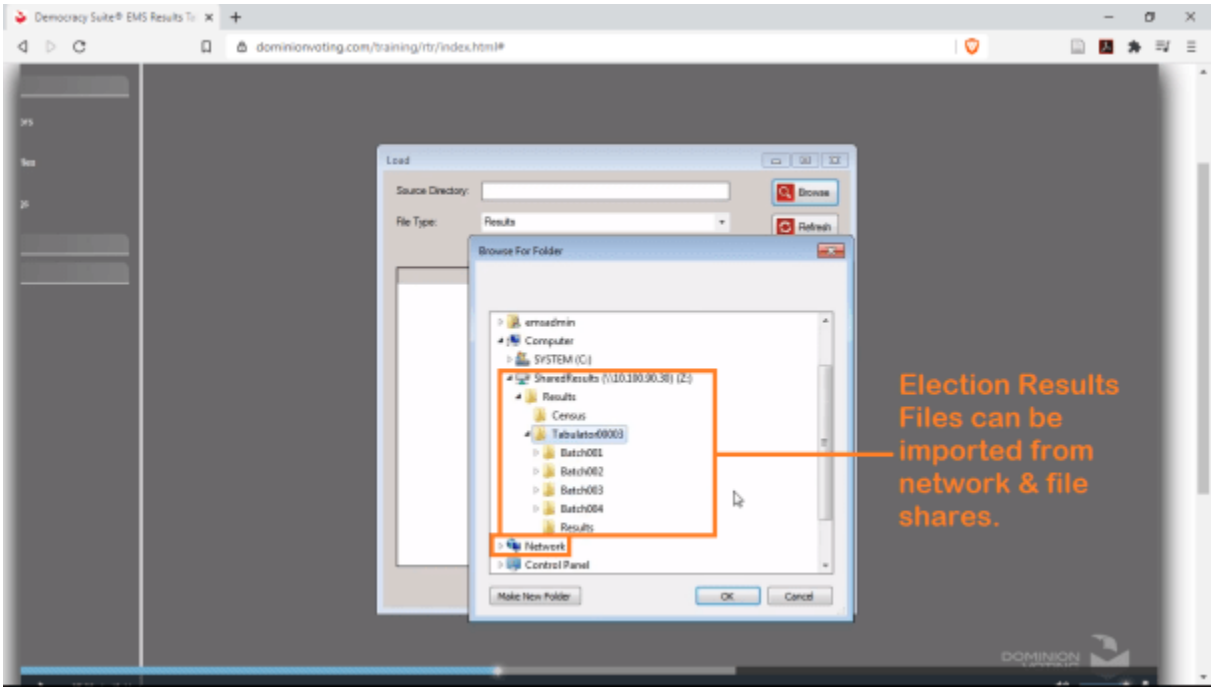
November 13, 2020cybersecurity, dems, Dominion, elections fraud

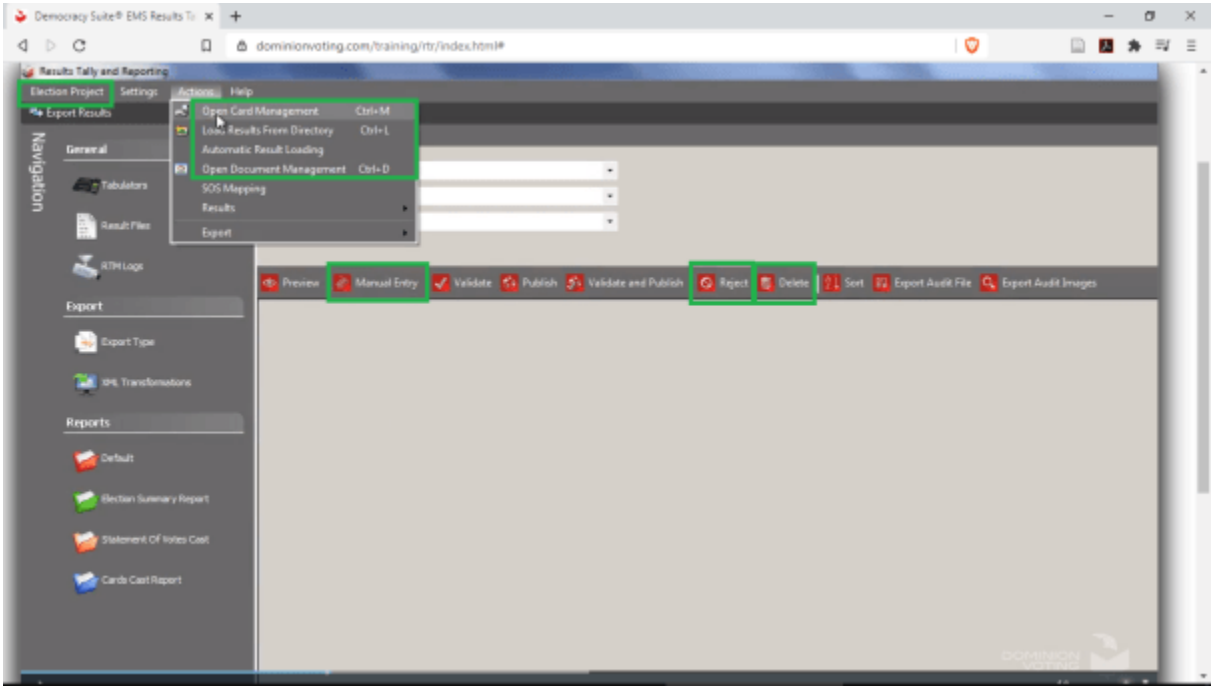
Dominion Voting Systems' Democracy Suite has features that allow for election results manipulation. The back-end software has an elections results editor, called *Results Tally and Reporting* (*RTR*). Its users are election officials. RTR is an equivalent of Microsoft Excel, but for election results. The software allows its users to enter "election results" from removable memory cards, local file system, and network. It allows you to merge multiple election results files. It allows the users to manually edit election result files. It allows users to reject election results files. In other words, it allows arbitrary change of results.

RTR runs not on a voting machine, but on an ordinary Windows laptop, which can be connected to the Internet, and even controlled remotely.

The Dominion's training video (https://www.dominionvoting.com/training/rtr/index.html) has a subsection *Flexible management of results after the election occurred* (starting at 4:20). Look at a few screenshots from it:

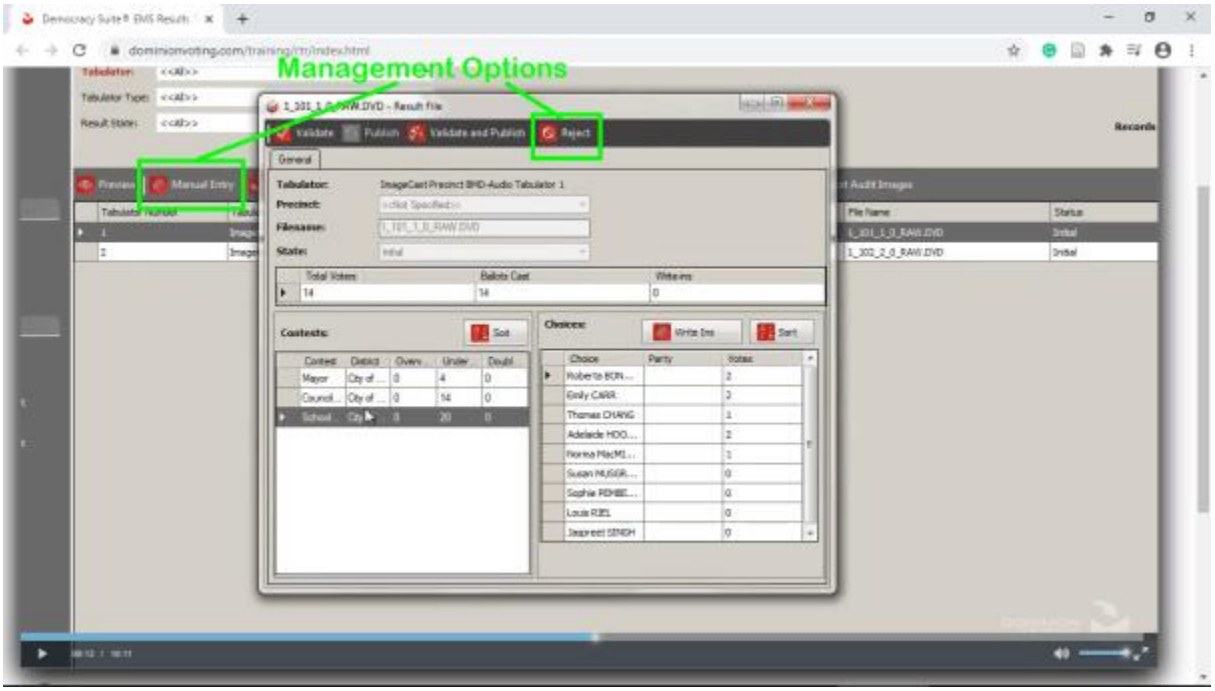






Various suspicious options (Election Project, Manual Entry, Reject, Delete, Open Document Management etc.) are highlighted.

Yes, you read it right: Managing Election Result Files (~8:40)



Such software likely allows running arbitrary scripts changing election result files, even remotely.

Dominion's Manual (**Democracy Suite® EMS Results Tally & Reporting User Guide, Version: 5.11-CO::7, May 28, 2019, available at** [Colorado Secretary of State](#)) openly explains how to use Dominion's software to alter election results:

"The following Result States exist:

- *Initial: this is the first Result State that is assigned once a Result File is loaded or manually entered into the system. In this state Write-in resolution is allowed. If the result file is manually entered, the result data is still editable. Result files in this state can be deleted by the user. …"*

The Manual repeats: a result file can be loaded or manually entered. The result data is editable. Results files can be deleted. In the Dominion's old words, "*choose from a number of unique tools to custom build your perfect election.*"

On the other hand, Dominion's systems also contain a log file for each result file. If the system works as promised, it would show where in the workflow the votes were switched, deleted, or added. Correctly identifying the source of fraud is not less important than recounting votes.

Dominion Voting Systems is a Canada-based company (possibly re-registered in the US), which develops much of its software in Serbia. Between September 6 and September 16, Dominion deleted a lot of information from its website, including the names and bios of the management. These actions do not increase public trust.

Dominion's Reaction

Dominion's response, which is continuously changing, is highly unusual. It is aggressive, desperate, and contains obvious falsehoods.

On November 13, Dominion updated its response, moved to their homepage, and wrote it in huge capital letters: "*DOMINION VOTING SYSTEMS CATEGORICALLY DENIES FALSE ASSERTIONS ABOUT VOTE SWITCHING ISSUES WITH OUR VOTING SYSTEMS*."

The statement continues: *"First and foremost, Dominion Voting Systems categorically denies any claims about any vote switching or* ***alleged software issues*** *with our voting systems."*

No software company can guarantee that its software has no issues, especially voting software that has never been used on such a large scale. An innocent company would express trust in its software and either promise to investigate or assure the public that it had already investigated and found no significant problems. It would also apologize for any minor problems.

Instead, the company is publishing rants and making patently false statements like "*THERE WERE NO DOMINION SOFTWARE GLITCHES*". This statement is simply not true, as evidenced by the multiple witnesses and recounting actions of multiple counties and some states. Another sentence refers to the Georgia Secretary of State as saying that all votes were accurately counted, despite the fact that he just announced a manual recount.

Further, no company issues categorical denials, essentially shifting blame onto its government customers. Also, businesses usually keep their statements very short in situations when "everything you say can and will be used against you."

Here are a couple of snapshots from earlier versions of Dominion's statements (Nov 12 and Nov 11).

https://defyccc.com/dominion-voting-in-california/

Dominion Voting in California

November 16, 2020Dominion-

California Secretary of State, Frequently Requested Information –> Dominion Voting [1]:

California certified **Dominion Democracy Suite** **5.10A** in September 2020, despite (or because of) its openness to abuse (not "vulnerabilities"). From the Secretary of State Staff Test Report (PDF):

"The Democracy Suite 5.10A voting system consists of eight major components."

Each component has its own security holes and/or backdoors, and many of them are sufficient to compromise the whole system. It is not just a voting machine.

"The Vote Center/Polling Place was setup with the Voter Activation Card Laptop, ICX machines, ICE machine, and Mobile Ballot Production for on demand printing. EED and RTR were setup as per the logical map below [the picture]"

RTR stands for *Results Tally and Reporting*, which I call ***Election Results Editor***.

"EMS-Results Tally Reporting (RTR), v. 5.10.50.83 RTR is the main application for post-voting activities. It receives election results from the tabulators, allows for validation of the results, and reports the results. ***RTR can be used for the addition, and deletion of tabulator files.*** *It also allows for manual resolution of qualified write-ins."*

Addition and deletion of tabulator files with all votes in them.

From the Democracy Suite Use Procedure (customized for CA)

*"**3.6 DATA CLEAR***

The Data Clear page allow administrators to erase results, voter data, and elections definitions." (p. 17)

Erasing results, no less!

From the Dominion Democracy Suite ImageCast Remote 5.10 RAVBMS Security and Telecommunications Report:

"The DS 5.10 RAVBMS does not have a built-in hash verification method for the system to verify that the source code [probably meant 'the system] is not running modified code."

There is a whole section ***Potential Vulnerabilities***. There are many: JavaScript code injection, Cross-site scripting (reflected), Open Redirection (DOM-Based). But genuine vulnerabilities are beyond the scope of this post. Why to attempt exploiting the vulnerabilities, when one can replace the original code with modified one, or to replace whole results files! Finally

"As directed by the California Secretary of State, this security testing report does not include any recommendation as to whether or not the system should be approved."

These are just a few quotes from the latest documents related to the Dominion Voting software used in these elections.

[1] California Secretary of State, Frequently Requested Information –> Dominion Voting

DominionLoanArticles

https://nationalfile.com/murdoch-family-newscorp-takes-loan-into-hundreds-of-millions-of-dollars-from-central-bank-of-china/

Murdoch Family, NewsCorp Takes Loan into Hundreds of Millions of Dollars from Central Bank of China

Murdoch skewers Xi Jinping's CCP daily, then takes nine figure sums of cash from the Communist-run bank.

by **NOEL FRITSCH**

September 4, 2024
Last Updated on September 4, 2024

April 1, 2022 — In the middle of a billion dollar litigation fight with Dominion Voting Systems, Rupert Murdoch's News Corp empire, which owns Fox News, Sky News Australia, *The Australian,* Fox News and *The New York Post,* agreed to take $100 million loan from the state-owned Bank of China's New York branch, among other loans, totaling over one billion dollars.

As our own Frankie Stockes reported at the time,

On March 29th, 2022, Rupert Murdoch, under the banner of his News Corp conglomerate, entered into a massive credit agreement that features mega-loans from multiple banks, including the CCP-run Bank of China. Also involved in the deal are a number of American banks that have been longtime sponsors of anti-American activities. In all, the loan is worth $1.25 billion, with $100,000,000 of it coming from the CCP.

The loan from the Central Bank of China is part of a much larger $1.25 billion loan package from the likes of Chase, CitiBank, J.P. Morgan, and Bank of America.

The one-and-a-quarter BILLION dollar loan package was announced on Thursday, March 31st of 2022, nearly exactly one year after Dominion Voting Systems filed a $1.6 billion defamation lawsuit against Fox News and alleged that Fox News "falsely claimed that the voting company had rigged the 2020 election," according to AP reports

Schedule I

Commitments

Name of Initial Lender	Term A Commitment	Revolving Credit Commitment	Letter of Credit Commitment[1]
Bank of America, N.A.	$ 111,700,000.01	$ 85,000,000.00	$ 33,333,333.34
Citibank, N.A.	$ 56,650,000.00	$ 85,000,000.00	$ 33,333,333.33
JPMorgan Chase Bank, N.A.	$ 56,650,000.00	$ 85,000,000.00	$ 33,333,333.33
Bank of China, New York Branch	$ —	$ 100,000,000.00	
Deutsche Bank AG, New York Branch	$ 40,000,000.00	$ 60,000,000.00	
Goldman Sachs Bank USA	$ 40,000,000.00	$ 60,000,000.00	
HSBC Bank USA, National Association	$ 40,000,000.00	$ 60,000,000.00	
Morgan Stanley Bank, N.A.	$ 35,000,000.00	$ 30,000,000.00	
MUFG Bank, Ltd.	$ 35,000,000.00	$ 30,000,000.00	
Australia and New Zealand Banking Group Limited	$ 28,333,333.33	$ 42,500,000.00	
Commonwealth Bank of Australia	$ 28,333,333.33	$ 42,500,000.00	
National Australia Bank Limited	$ 28,333,333.33	$ 42,500,000.00	
Westpac Banking Corporation	$ —	$ 27,500,000.00	
TOTAL:	**$ 500,000,000.00**	**$ 750,000,000.00**	**$ 100,000,000.00**

[1] The Letter of Credit Commitment is part of and not in addition to the Revolving Credit Commitment.

A year after receiving the $1.25 billion dollar loan, Fox News ended up caving to the pressure, and ultimately agreed to pay Dominion Voting Systems nearly $800 million.

Aussie Chi-Comm Lapdogs Rupert and Lachlan Murdoch (Image: EPA/Andrew Gombert)

The curious timing of the massive loan made to Fox News by China's central bank, among others, has been referred to by some observers as the Dominion Squeeze.

Perhaps explains the 'Dominion Squeeze' explains the Murdochs' need for an extra 1.25 BILLION of loose change.

You can read the 180-page News Corp filing here:

Crikey, an Australian website who covers all some things Australia, partially covered the news of the CCP-Murdoch loan back in April of 2022.

In "Limited Hangout" Fashion, Crikey attempted to cover the Murdoch pay-off through the lens of potential mergers, acquisitions, and/or sales of News Corp, which has a market cap of $13.2 billion.

Crikey also referenced the Murdochs' holdings in Disney, which the Aussie rag called "famously prudent and morally conservative."

Evidently Deep State Aussie news outlet Crikey hasn't been paying attention to Disney all that closely.

The Murdochs have even accused Crikey of using the legal threat of the Dominion lawsuit to attract subscribers and notoriety, and ultimately sued Crikey for it.

Crikey's refusal to cover the Dominion angle of the much larger Dominion suit against the Murdochs leaves questions about whether Crikey is somehow in league with Dominion.

Crikey "reports" as follows:

In a classic case of "do as I say, not as I do", the Murdoch family has tapped into a US$100 million loan from a Chinese government bank, at the same time as its global media empire is going to war with the Chinese Communist Party.

The Murdochs have gerrymandered control over two public companies — News Corp and Fox Corp — and ever since COVID hit two years ago their outlets — such as Sky News Australia, The Australian, Fox News and The New York Post — have become virulent critics of the Chinese regime.

However, China has become even more toxic globally in recent weeks given its collaboration with Vladimir Putin, and concerns in Australia have been amplified by its attempts to establish a naval base in the Solomon Islands.

So what should the world make of News Corp's decision to accept a US$100 million loan from the state-owned Bank of China's New York branch as part of a massive US$1.25 billion loan package revealed on Thursday? Is it a case of Beijing seeking to influence the Murdochs by stepping up as a key family financier? And should the Murdochs have refused the money, effectively joining the global ESG boycott of Russia in which more than 400 global companies have pulled out or temporarily closed their Russian operations?

The Bank of China is the second biggest player in the 13-bank News Corp syndicate after Bank of America, which is contributing US$111.7 million. The Bank of China is massive — it has an estimated US$3.7 trillion in assets and is the largest and oldest bank on the Chinese mainland.

This 180-page News Corp filing with the US Securities and Exchange Commission and the ASX in Australia contains details of who will be funding the surprisingly large new loan for News Corp. If only ASX-listed companies were required to be this transparent in disclosing who their financiers are.

All Australia's big four banks are in the News Corp syndicate, but there are no UK banks — with the exception of HSBC. Perhaps the likes of London-based Natwest and Barclays don't want to be associated with the notorious phone-hacking company which is still paying hundreds of millions to victims, as Crikey reported this week.

CBA, NAB and ANZ are contributing US$28.333 million each to the first tranche of US$500 million. All four are contributing to the second tranche, a US$S700 million revolving credit line, although Westpac is making the smallest contribution, US$27.5 million. CBA, NAB and ANZ are stumping up US$42.5 million each, even though all claim to be committed to serious action on climate change and the Murdoch empire is one of the world's worst climate denialist outfits.

It seems ESG lending in Australia is increasingly blackballing coalminers but hasn't yet reached key propagandists such as News Corp.

So why does News Corp need this money. Well, according to the filing it's for unstated "general corporate purposes" — a big takeover in the offing?

As Crikey reported this time last year, News Corp first went into a net debt position 12 months ago when it committed $1.07 billion to three separate bolt-on acquisitions in one week. However, last August it then announced a much bigger US$1.15 billion cash acquisition of OPIS, the Oil Price Information Service that S&P was forced to sell by US anti-trust regulators. This was partially funded

by a US$500 million notes issue in February 2022 — an issue of debt securities to institutional investors rather than borrowing directly from banks.

The December 31 accounts for News Corp show it already had gross borrowings totalling US$2.27 billion, down slightly from US$2.313 billion at June 30. However, this was largely cancelled out by US$2.187 billion of cash holdings.

News Corp's market capitalisation is US$13.2 billion, so it has capacity to borrow a few billion.

The Fox Corp balance sheet is in slightly worse shape with US$8 billion in gross debt, although this is partially offset by US$4.25 billion in cash, leaving net debt at just US$3.75 billion, against a market capitalisation of US$22 billion.

The Murdochs have a clear majority of their estimated A$30 billion family fortune tied up in Disney shares, although it has no influence over the US$250 billion business. There is also no transparency as to which particular Murdoch family members have hung on to their Disney shares after the US$91 billion sale of 21st Century Fox's entertainment assets in March 2019.

Disney is famously prudent and morally conservative. Indeed, the Sky after dark crew last night was lashing the company for being too politically correct with its casting decision and treatment of minorities.

Faced with the ethical dilemma of whether to accept a US$100 million loan from the Bank of China in the current environment, it's hard to imagine the Disney board agreeing to such a proposition.

But for Rupert and Lachlan Murdoch, it was no problem whatsoever — meaning that, as of yesterday, parts of the Murdoch media empire are "partially brought to you by the autocratic dictators in Beijing". But don't expect this to be disclosed by News Corp's anti-China shock jocks any time soon.

https://nationalfile.com/could-100-million-loan-from-chinese-state-bank-to-rupert-murdoch-explain-tuckers-ouster/

Could $100 Million Loan from Chinese State Bank to Rupert Murdoch Explain Tucker's Ouster?

Tucker Carlson's ouster from Fox News has raised questions about the network's ties to Communist China.

by **FRANKIE STOCKES** May 2, 2023

Could the Chinese Communist Party and its global influence network be responsible for Tucker Carlson's ouster from Fox News? Just last year, Rupert Murdoch, the international media mogul who founded Fox News, took a whopping $100 million loan from the CCP-run Bank of China, to fund his continued media ventures. The loan coincided with a much-observed turn in Fox's coverage, towards a model that's been called "CNN-lite."

On March 29th, 2022, Rupert Murdoch, under the banner of his News Corp conglomerate, entered into a massive credit agreement that features mega-loans from multiple banks, including the CCP-run Bank of China. Also involved in the deal are a number of American banks that have been longtime sponsors of anti-American activities. In all, the loan is worth $1.25 billion, with $100,000,000 of it coming from the CCP.

Commitments

Name of Initial Lender	Term A Commitment	Revolving Credit Commitment	Letter of Credit Commitment1
Bank of America, N.A.	$ 111,700,000.01	$ 85,000,000.00	$ 33,333,333.34
Citibank, N.A.	$ 56,650,000.00	$ 85,000,000.00	$ 33,333,333.33
JPMorgan Chase Bank, N.A.	$ 56,650,000.00	$ 85,000,000.00	$ 33,333,333.33
Bank of China, New York Branch	$ —	$ 100,000,000.00	
Deutsche Bank AG, New York Branch	$ 40,000,000.00	$ 60,000,000.00	
Goldman Sachs Bank USA	$ 40,000,000.00	$ 60,000,000.00	
HSBC Bank USA, National Association	$ 40,000,000.00	$ 60,000,000.00	
Morgan Stanley Bank, N.A.	$ 35,000,000.00	$ 30,000,000.00	
MUFG Bank, Ltd.	$ 35,000,000.00	$ 30,000,000.00	
Australia and New Zealand Banking Group Limited	$ 28,333,333.33	$ 42,500,000.00	
Commonwealth Bank of Australia	$ 28,333,333.33	$ 42,500,000.00	
National Australia Bank Limited	$ 28,333,333.33	$ 42,500,000.00	
Westpac Banking Corporation	$ —	$ 27,500,000.00	
TOTAL:	**$ 500,000,000.00**	**$ 750,000,000.00**	**$ 100,000,000.00**

Rupert Murdoch's massive loan includes $100,000 from the CCP-run Bank of China. SOURCE

Though the loan deal was made under News Corp, which is separate from Fox Corp, Murdoch's dealings with the Chinese Communist Party and Fox's gradual move to the left cannot be ignored, especially when it's taken into account that in 2022, Murdoch floated the idea of merging Fox Corp and News Corp into one super-conglomerate, a plan that was later aborted when it was determined that it'd be bad for shareholders.

Before his ouster from Fox News, Tucker Carlson, whose audience dwarfed that of all the other shows on the network, took a hardline stance against Communist China and was one of the few cable news hosts willing to broach the subject of COVID-19 and its ties to the Chinese regime, as well as America's own government, which operates biolabs inside of Communist China.

As a Fox News host, Carlson also covered the Communist Chinese takeover of Brazil, in which the hotly-contested 2022 Brazilian Presidential Election was directly meddled in by the Chinese Communist Party and its vast network of foreign influence operations, a saga that was also closely covered by National File.

Related: Polling Shows Tucker More Liked than Fox News

In a recent video interview, Carlson blasted Communist China, calling the CCP and its global influence a major "problem" for the United States and the world.

"China is a problem that is very hard for the United States to solve," Carlson said, before calling out the intentional ceding of global influence and power to the totalitarian state by Western governments and elites.

"By problem, I mean sort of giving hegemony over the world to a country that doesn't believe anything really that we believe."

"Our former Secretary of State Madeleine Albright…Got rich effectively making China's case to the American business community. Many others in our diplomatic core have done the same. That's a straight-up sell-out. Hollywood sells out. The NBA sells out," said Carlson.

Hear Tucker Carlson's comments on Communist China and the Western elites who've sold out to it in the video below:

https://youtu.be/wqDCyast7_U

REDACTED VERSION

Security Analysis of Georgia's ImageCast X Ballot Marking Devices

Expert Report Submitted on Behalf of Plaintiffs Donna Curling, et al.
Curling v. Raffensperger, Civil Action No. 1:17-CV-2989-AT
U.S. District Court for the Northern District of Georgia, Atlanta Division

Prof. J. Alex Halderman, Ph.D.

With the assistance of Prof. Drew Springall, Ph.D.

July 1, 2021

Contents

1 Overview 4
1.1 Principal Findings 4
1.2 Main Conclusions 6
1.3 Organization of this Report 8
2 Georgia's Voting Equipment 9
2.1 Certification and Testing History 9
2.2 ImageCast X Hardware and Software 9
2.3 ImageCast Precinct Hardware and Software 11
3 Threats to Georgia Elections 12
3.1 Threat Actors 12
3.2 BMD Ballot Manipulation Attacks 13
4 Methodology and Testing Process 17
4.1 Testing Methodology 17
4.2 Materials Examined 18
4.3 Testing Process 18
4.4 Proof-of-Concept Attacks 19
5 Manipulating Ballots via the ICX Printer 20
5.1 Decoding Ballot QR Codes 20
5.2 Defeating QR Code Authentication 21
5.3 Demonstration Hardware-Based Attack 23
6 Attacks Against ICX Smart Cards 26
6.1 Extracting Election Secrets from Poll Worker Cards 28
6.2 Forging Technician Cards to Install Malware on any ICX 29
6.3 Creating "Infinite" Voter Cards 30
7 Constructing ICX Malware 32
7.1 Overview of the Approach 32
7.2 Obtaining the Real APK 33
7.3 Decompiling and Reverse-Engineering 33
7.4 Modifying the ICX App to Change Votes 34
7.5 Defeating Applicable Defenses 35
7.6 Conclusions 38
8 Installing Malware Locally 39
8.1 Attaching USB Devices to the ICX 39
8.2 "Escaping" the ICX App 41
8.3 Accessing a Root Shell via the Built-In Terminal App 43
8.4 Manual Malware Installation Process 43
8.5 Automating Malware Installation 44
8.6 Local Malware Installation using a Forged Technician Card 45
8.7 Local Malware Installation via Android Safe Mode 46
9 Installing Malware Remotely 48
9.1 ICX Election Definitions 48

9.2 Directory Traversal Vulnerability 50
9.3 Arbitrary Code Execution as Root 50
9.4 Installing Malware from the Election Definition File 51
9.5 Defeating Security Precautions More Easily 52
9.6 Conclusions 53
10 Manipulating Logs and Protective Counters 54
10.1 Vulnerable Storage Design 54
10.2 Manual and Automated Modification 55
11 Weaknesses in the ICP Scanner 56
11.1 The ICP Accepts Photocopied Ballots 56
11.2 A Dishonest Poll Worker with Access to the ICP Memory Card can Deanonymize All Voted Ballots 56
11.3 Installed Tamper-Evident Seal could be Bypassed or Defeated . . . 57
Expert Qualifications 60
References 61
Exhibit A: October 2020 Software Update Instructions 67
Exhibit B: Georgia Logic and Accuracy Procedures 78
Exhibit C: Pro V&V Field Audit Report 90

1 Overview

In 2020, Georgia replaced its insecure, decades-old DRE voting machines with new ballot scanners and ballot marking devices (BMDs) manufactured by Dominion Voting Systems. Although the same BMDs are used for accessibility in parts of approximately 15 other states, Georgia is unique in using them statewide as the primary method of in-person voting [89]. This unusual arrangement places potentially malicious computers between Georgia voters and their paper ballots. In contrast, in most of the United States, voters mark paper ballots directly by hand, and BMDs are reserved for those who need or request them [87]. Georgians who vote at a polling place generally have no choice but to use the BMDs.

All voting systems face cybersecurity risks. As the National Academies of Sciences, Engineering, and Medicine recently concluded "[t]here is no realistic mechanism to fully secure vote casting and tabulation computer systems from cyber threats" [58]. However, not all voting systems are equally vulnerable. Curling Plaintiffs contend that Georgia's universal-use BMD voting system is *so insecure* that it violates voters' constitutional rights.

To assist the Court in understanding the risks that the system creates, Curling Plaintiffs asked me to conduct a security analysis of the ImageCast X (ICX) BMD and associated equipment used in Georgia elections. Using an ICX provided by Fulton County, I played the role of an attacker and attempted to discover ways to compromise the system and change votes. I, along with my assistant, spent a total of approximately twelve person-weeks studying the machines, testing for vulnerabilities, and developing proof-of-concept attacks. Many of the attacks I successfully implemented could be effectuated by malicious actors with very limited time and access to the machines, as little as mere minutes. This report documents my findings and conclusions.[1]

1.1 Principal Findings

I show that the ICX suffers from critical vulnerabilities that can be exploited to subvert all of its security mechanisms, including: user authentication, data integrity protection, access control, privilege separation, audit logs, protective counters, hash validation, and external firmware validation. I demonstrate that these vulnerabilities provide multiple routes by which attackers can install malicious software on Georgia's BMDs, either with temporary physical access or remotely from election management systems (EMSs). I explain how such malware can alter voters' votes while subverting all of the procedural protections practiced by the State, including acceptance testing, hash validation, logic and accuracy testing, external firmware validation, and risk-limiting audits (RLAs).

The most serious vulnerabilities I discovered include the following:

1. Attackers can alter the QR codes on printed ballots to modify voters' selections. Critically, voters have no practical way to confirm that the QR codes

[1] I hereby incorporate my previous declarations as if fully stated herein. I have personal knowledge of the facts in this report and, if called to testify as a witness, I would testify under oath to these facts.

match their intent, but they are the only part of the ballot that the scanners count. I demonstrate how the QR codes can be modified by compromising the BMD printer (Section 5) or by installing malware on the BMD (Section 7).

2. The software update that Georgia installed in October 2020 left Georgia's BMDs in a state where anyone can install malware with only brief physical access to the machines. I show that this problem can potentially be exploited in the polling place even by non-technical voters (Section 8).
3. Attackers can forge or manipulate the smart cards that the ICX uses to authenticate technicians, poll workers, and voters. Without needing any secret information, I created a counterfeit technician card that can unlock any ICX in Georgia, allowing anyone with physical access to install malware (Section 6).
4. I demonstrate that attackers can execute arbitrary code with root (supervisory) privileges by altering the election definition file that county workers copy to every BMD before each election. Attackers could exploit this to spread malware to all BMDs across a county or the entire state (Section 9).
5. The ICX contains numerous unnecessary Android applications, including a Terminal Emulator that provides a "root shell" (a supervisory command interface that overrides access controls). An attacker can alter the BMD's audit logs simply by opening them in the on-screen Text Editor application (Section 10).
6. In a given election, all BMDs and scanners in a county share the same set of cryptographic keys, which are used for authentication and to protect election results on scanner memory cards. An attacker with brief access to a single ICX or a single Poll Worker Card and PIN can obtain the county-wide keys.
7. The ImageCast Precinct (ICP) scanner stores ballot scans in the order they were cast. A dishonest election worker (like that emphasized by the Defendants and their expert Michael Shamos) with just brief access to the scanner's memory card could violate ballot secrecy and determine how individual voters voted (Section 11).

Proof-of-Concept Attacks In addition to discovering and validating the vulnerabilities described above, I developed a series of proof-of-concept attacks that illustrate how vulnerabilities in the ICX could be used to change the personal votes of individual Georgia voters. I am prepared to demonstrate:

1. An attack that uses malicious hardware hidden inside the BMD's printer to alter the votes on printed ballots (Section 5).
2. Malware that runs on the BMD and alters votes while avoiding hash validation, firmware validation, and logic and accuracy testing (Section 7).
3. An automated method of installing malware by briefly unplugging the printer cable and attaching a malicious USB device (Section 8).
4. Vote-stealing malware that can be installed remotely from the EMS, by altering the BMD's election definition file (Section 9).

Mitigation Some of the critical vulnerabilities I discovered can be at least partially mitigated through changes to the ICX's software, and I encourage Dominion and the State of Georgia to move as quickly as possible to remedy them.[2] However, merely patching these specific problems is unlikely to make the ICX substantially more secure. I did not have the resources to find *all* possible exploitable security bugs in the ICX software. Once I found one that satisfied a particular adversarial objective, I usually turned to investigating other aspects of the system. It is very likely that there are other, equally critical flaws in the ICX that are yet to be discovered. Fully defending it will require discovering and mitigating them all, but attackers would only have to find one.

1.2 Main Conclusions

On the basis of the technical findings described in this report, I reach the following conclusions:

- The ICX BMDs are not sufficiently secured against technical compromise to withstand vote-altering attacks by bad actors who are likely to attack future elections in Georgia. Adversaries with the necessary sophistication and resources to carry out attacks like those I have shown to be possible include hostile foreign governments such as Russia—which has targeted Georgia's election system in the past [49]—and domestic political actors whose close associates have recently acquired access to the same Dominion equipment that Georgia uses through audits and litigation in other jurisdictions.
- The ICX BMDs can be compromised to the same extent and as or more easily than the AccuVote TS and TS-X DREs they replaced.[3] Both systems have similar weaknesses, including readily bypassed user authentication and software validation, and susceptibility to malware that spreads from a central point to machines throughout a jurisdiction. Yet with the BMD, these vulnerabilities tend to be even easier to exploit than on the DRE system, since the ICX uses more modern and modular technology that is simpler to investigate and modify.
- Despite the addition of a paper trail, ICX malware can still change individual votes and most election outcomes without detection. Election results are determined from ballot QR codes, which malware can modify, yet voters cannot check that the QR codes match their intent, nor does the state compare them to the human-readable ballot text. Although outcome-changing fraud conducted in this manner could be detected by a risk-limiting audit, Georgia requires a risk-limiting audit of only one contest every two years, so the vast majority of elections and contests have no such assurance. And even the

[2] Over the past six months, I have repeatedly offered (through Curling Plaintiffs' counsel) to meet with Dominion and share my findings, so that the company could begin developing software fixes where possible, but they have yet to take me up on this offer.

[3] I conducted similar analyses of the TS in 2006 [31] and the TS-X in 2007 [11].

most robust risk-limiting audit can only assess an election outcome; it cannot evaluate whether individual votes counted as intended.

- The ICX's vulnerabilities also make it possible for an attacker to compromise the auditability of the ballots, by altering both the QR codes and the human readable text. Such cheating could not be detected by an RLA or a hand count, since all records of the voter's intent would be wrong. The only practical way to discover such an attack would be if enough voters reviewed their ballots, noticed the errors, and alerted election officials, and election officials identified the problem as a systemic hack or malfunction; but human-factors studies show that most voters do not review their ballots carefully enough, and election officials likely would consider such reports the product of voter error. This means that in a close contest, ICX malware could manipulate enough ballots to change the election outcome with low probability of detection. In contrast, risk-limiting audits of *hand-marked* paper ballots, when used with appropriate procedural precautions, provide high confidence that individual votes are counted as intended and election outcomes are correct even if the election technology is fully compromised.
- Using vulnerable ICX BMDs for all in-person voters, as Georgia does, greatly magnifies the security risks compared to jurisdictions that use hand-marked paper ballots but provide BMDs to voter upon request. When use of such BMDs is limited to a small fraction of voters, as in most other states, they are a less valuable target and less likely to be attacked at all. Even if they are successfully compromised, attackers can change at most a small fraction of votes—which, again, creates a strong disincentive to undertake the effort and risk to change any such votes.
- The critical vulnerabilities in the ICX—and the wide variety of lesser but still serious security issues—indicate that it was developed without sufficient attention to security during design, software engineering, and testing. The resulting system architecture is brittle; small mistakes can lead to complete exploitation. Likewise, previous security testing efforts as part of federal and state certification processes appear not to have uncovered the critical problems I found. This suggests that either the ICX's vulnerabilities run deep or that earlier testing was superficial. In my professional experience, secure systems tend to result from development and testing processes that integrate careful consideration of security from their inception. In my view, it would be extremely difficult to retrofit security into a system that was not initially produced with such a process.

My technical findings leave Georgia voters with greatly diminished grounds to be confident that the votes they cast on the ICX BMD are secured, that their votes will be counted correctly, or that any future elections conducted using Georgia's universal-BMD system will be reasonably secure from attack and produce the correct results. No grand conspiracies would be necessary to commit large-scale fraud, but rather only moderate technical skills of the kind that attackers who

are likely to target Georgia's elections already possess. Unfortunately, even if such an attack never comes, the fact that Georgia's BMDs are so vulnerable is all but certain to be exploited by partisan actors to suppress voter participation and cast doubt on the legitimacy of election results.

1.3 Organization of this Report

I begin in Section 2 by providing an overview of the Democracy Suite voting equipment used in Georgia. In Section 3, I establish a threat model, including the most likely kinds of attacks and attackers facing the election system, and ways in which these attackers might attempt to manipulate BMD ballots. I then discuss my methodology and testing process in Section 4.

Next, I present my technical findings, which I organize into several parts. Section 5 explains how the barcodes on ICX ballots can be decoded and manipulated, and how such manipulation could be accomplished in the supply chain through alteration of the BMD printer hardware. In Section 6, I analyze the smart cards that the ICX uses to authenticate workers and voters, and I show numerous ways that they can be attacked to create counterfeit cards and to extract cryptographic secrets. Section 7 describes how I created malicious software that can run on the ICX and manipulate ballots while subverting Georgia's procedural defenses. In Section 8, I describe several ways that such malware could be installed on individual BMDs by attackers with temporary physical access, including by exploiting a weakness introduced in the process of installing the October 2020 BMD software update. In Section 9, I describe a remote code execution vulnerability that makes it possible to install malware over a wide area without physical access to individual BMDs. Section 10 explains how even non-technical attackers can easily manipulate the ICX's audit log and protective counters. Finally, Section 11 details security problems that I discovered in the ICP ballot scanner incidentally to my study of the ICX.

2 Georgia's Voting Equipment

As of November 2020, approximately 24 states used one or more components of the Dominion Democracy Suite voting system [88], which encompasses various models and versions of ballot scanners, BMDs, and election management system software. Georgia uses Democracy Suite version 5.5–A, including ImageCast X Prime (ICX) BMDs, ImageCast Precinct (ICP) precinct-count optical scanners, ImageCast Central (ICC) central-count optical scanners, and the Democracy Suite EMS.

My analysis focuses on the ICX BMD. In 2020, the ICX was used in parts of 16 states: Alaska, Arizona, California, Colorado, Georgia, Illinois, Kansas, Louisiana, Michigan, Missouri, Nevada, New Jersey, Ohio, Pennsylvania, Tennessee, and Washington. Although the vast majority of jurisdictions provide the ICX BMD to voters on request to assist with accessibility, Georgia is the only state to mandate ICX BMDs as the primary method of in-person voting state-wide [89].

2.1 Certification and Testing History

Democracy Suite 5.5–A is the successor to version 5.5, which was certified by the U.S. Election Assistance Commission (EAC) in September 2018 under the Voluntary Voting Systems Guidelines (VVSG) 1.0 (2005) standard [85, 86] following testing by Pro V&V, an EAC-accredited Voting System Test Laboratory (VSTL) [67]. Version 5.5–A was certified in January 2019 as a modification to 5.5. As a modification, it required only limited review, which was conducted by another VSTL, SLI Compliance [74].

Georgia entered into an agreement to purchase 5.5–A in July 2019 [34], and the Secretary of State engaged Pro V&V to evaluate it against state requirements. This evaluation was completed in August 2019 [66], and, two days later, the Secretary of State certified that the system was "in compliance with the applicable provisions of the Georgia Election Code and Rules of the Secretary of State" [33].

Over the past four years, Democracy Suite has been the subject of security testing on at least seven occasions as part of state certification processes in other states, as summarized in Table 1. In California and Pennsylvania, tests were conducted by Pro V&V and SLI, and in Texas by statutorily appointed examiners. These tests involved source code review and/or hands-on testing. Some of the tests raised serious concerns, but only Texas declined to certify the Dominion system. Based on the public test reports, it appears that none of these tests uncovered the critical security issues that I document here.

2.2 ImageCast X Hardware and Software

The ICX [25] is an Android-based touch-screen device that can be operated as either a BMD or a DRE. In Georgia, it is exclusively used as a BMD, allowing voters to mark ballots on-screen and print them to an attached laser printer.

The ICX hardware, shown in Figure 1, is a commercial off-the-shelf (COTS) Avalue HID-21V-BTX-B1R "Industrial Panel PC" [8]. On the front of the

Date	Version	State	VSTL	Findings	Result
Oct 2017	5.2	CA	SLI	Issues related to audit logging, passwords, anti-virus, and installation	Accept [13, 77, 78]
				Potential vulnerability related to software execution from attached USB drive	
Oct 2018	5.5	PA	SLI	Concerns regarding system hardening documentation	Reject [64]
Jan 2019	5.5–A	PA	SLI	None	Accept [64]
Jun 2019	5.5	TX	—	"concerns about whether [it] preserves the secrecy of the ballot [and] operates efficiently and accurately"	Reject [27]
Oct 2019	5.10	CA	SLI	Issues related to audit logging, passwords, anti-virus, and installation	Accept [12, 75, 76]
Jan 2020	5.5–A	TX	—	"concerns about whether [it] operates efficiently and accurately; and is safe from fraudulent or unauthorized manipulation"	Reject [28]
Jul 2020	5.10–A	CA	Pro V&V	None (source-code only)	Accept [55, 65]

Table 1: **Prior Security Testing During State Certifications.** Various versions of the ICX and ICP were subjected to forms of security testing during state certification tests in California, Pennsylvania, and Texas. Although some tests flagged concerns, only Texas declined to certified the equipment. None of these tests appear to have uncovered the critical security issues we found.

device, there is a 21.5-inch touch-screen display and a smart-card slot used for authentication. On the back, there are four externally-accessible compartments covered by plastic doors: three containing various ports and the machine's power button, and one with a battery for backup power.

The ICX I tested runs a modified version of the Android 5.1.1 ("Lollipop"). This version of Android was released in December 2015. Even at that time, the next major version of Android ("Marshmallow") had been available for months. Today, the current release is Android 11, which shipped in September 2020 [3].

Most of the ICX's functionality is provided by an Android application developed by Dominion, which I will refer to as the "ICX App". Unlike with consumer phones and tablets, the ICX App is not distributed through an "app store" (and could not be without connecting the ICX to the Internet). Instead, it is installed through a process called "side-loading", in which an Android application package (APK) file containing the software is loaded from a USB device.

The ICX App itself does not contain any election-specific information, such as races or candidates. Rather, these are loaded to the device from a USB drive before each election, in the form of an election definition file created using the Democracy Suite EMS software.

(a) ImageCast X [8] (b) ImageCast Precinct [23]

Figure 1: **The ICX BMD and ICP Scanner Used in Georgia**

2.3 ImageCast Precinct Hardware and Software

While not the focus of this study, I briefly examined the ICP scanner. The ICP [23], shown in Figure 1, is used to count voted ballots. It can process ballots that are produced by the ICX or those that are marked by hand. Inserted ballots are automatically pulled through the paper path, scanned on both sides, and deposited into a ballot box.

In contrast to the ICX, the ICP uses a custom hardware design. A small touch-screen display provides administrative controls and feedback to voters. A built-in thermal printer produces "poll tapes" that record vote tallies. Whereas the ICX uses standard smart cards for user authentication, the ICP uses a device called an iButton [47], which Dominion refers to as a "security key".

There are three externally-accessible compartments on the ICP, all with plastic doors that can be covered with a tamper-evident seal. A compartment on the right side contains a USB Type-A port and an RJ-45 jack. On the front are two compartments for inserting Compact Flash cards used to load the election definition and store results.

The ICP I tested runs a variant of the Linux operating system, μClinux version 20070130. μClinux is a Linux variant intended for use in embedded devices; version 20070130 was released in February 2007 [83] and is more than 14 years older than the most recent Linux version. A custom application named `cf200.sig` runs on top of μClinux and provides most of the scanner's functionality.

3 Threats to Georgia Elections

Georgia elections face a growing risk of attack by a range of capable adversaries, including hostile foreign governments, domestic political actors, and election insiders. Here I describe these threat actors, their capabilities, and what they are likely to seek to accomplish through technical manipulation. I also discuss strategies they could use to manipulate ballots voted using the ICX BMD.

3.1 Threat Actors

Hostile Foreign Governments. Georgia's election system continues to face a high risk of being targeted by hostile foreign governments, such as Russia, which mounted a complex campaign of cyber attacks against U.S. election infrastructure—including Georgia's—during the 2016 election [48, 49]. Hostile governments could attempt to hack Georgia's election system to achieve a variety of goals, including causing fraudulent election outcomes.

Russia and other foreign governments continue to threaten Georgia's elections today. Less than a year ago, the U.S. Intelligence Community assessed that foreign threats to the 2020 election included "ongoing and potential activity" from Russia, China, and Iran, concluding that "[f]oreign efforts to influence or interfere with our elections are a direct threat to the fabric of our democracy". These adversarial governments may "seek to compromise our election infrastructure for a range of possible purposes, such as interfering with the voting process, stealing sensitive data, or calling into question the validity of the election results." [63]

Nation-state actors are among the most well resourced and technically sophisticated adversaries, and some of the most difficult to defend against. They frequently discover vulnerabilities in widely used software with which they can compromise protected systems, and they capable of creating advanced malicious software tailored for individual high-value targets [29, 72].

Nation-state actors likely can obtain access to election equipment with which to develop attacks via physical intrusion, theft, or by purchasing it under false pretenses. They also have developed a variety of techniques for infiltrating non-Internet-connected systems, including by compromising hardware and software supply chains [15, 61, 62] and by spreading malware on removable media that workers use to copy files in and out of protected environments.[4] Such methods could be used to target the EMS systems that are used to prepare and distribute election definitions files for the ICX. The attackers could then exploit vulnerabilities I discovered to spread vote-stealing malware to BMDs throughout Georgia.

Domestic Political Actors. In addition to the threat from foreign governments, Georgia's election system faces increasing risks from domestic political actors. Politically motivated attackers might seek to directly alter individual votes and

[4] A well-known example of this ability, which is known as "jumping an air gap", is the Stuxnet computer virus, which was created to sabotage Iran's nuclear centrifuge program by attacking factory equipment that was not directly connected to the Internet [92].

thereby change the outcome of a future election through hacking. They are also likely to exploit the fact that the election system has vulnerabilities to cast doubt on the legitimacy of results or suppress voter participation.

My work demonstrates that discovering and exploiting vulnerabilities in the ICX requires only a moderate time investment from technical experts. In recent months, numerous technically-skilled outside parties have gained access [17, 68].

For example, contractors have been given unsupervised access to ICX and ICP equipment in Maricopa County, Arizona, in the context of a controversial forensic audit of the November election [14, 43]. The audit is being led by a cybersecurity firm called Cyber Ninjas, whose owner is said to promote baseless conspiracy theories that the 2020 Presidential election was hacked to defeat Donald Trump [26]. The proliferation of access to the equipment by possibly untrustworthy and politically-motivated actors and their associates has greatly increased the risk that information sufficient to attack Georgia's election system will fall into the wrong hands.

Election Insiders. As the Defendants and their expert Michael Shamos have emphasized, dishonest election insiders also pose a high risk to Georgia elections. County technicians, vendor support personnel, and poll workers need to have access to election equipment—sometimes without supervision—in order to carry out their job functions. I detail a wide variety of attacks that could be performed with such access, including infecting BMDs with malware on a wide scale.

Although discovering vulnerabilities and developing malware likely requires a degree of technical skill beyond that of most election workers, malware, once developed, can be implanted by unskilled attackers. Dishonest insiders could be recruited (or planted) by the sophisticated foreign and domestic threat actors described above to attack Georgia's voting system in this manner.

Voters. The least privileged category of attacker I consider is ordinary voters, who have brief access to the ICX within the polling place. Most voters are unlikely to have the technical expertise to develop attacks on their own, but, like election insiders, non-technical voters could be recruited by more sophisticated threat actors. I assume only that a dishonest voter has the ability to follow instructions provided by a more technical criminal. And, of course, there no doubt are many among the millions in Georgia who themselves possess the requisite technical skills to develop and implement one or more of the attacks I detail here, among others not yet identified. Under this model, I show that even typical voters could potentially infect Georgia BMDs with vote-stealing malware.

3.2 BMD Ballot Manipulation Attacks

The ICX, as used in Georgia, produces ballots like the one shown in Figure 2. They are printed on one or more sheets of letter-size paper. The ballot design uses a QR code (a kind of two-dimensional barcode) to represent the voter's selections in machine-readable form. Although the ballot also contains human-readable text

that summarizes the selected choices for each contest, Dominion scanners ignore the ballot text and exclusively count the votes that are encoded in the QR code. Voters have no practical way to read the QR codes, so they cannot verify the representation of their vote that is counted.

In later sections, I will show how attackers can manipulate ICX ballots through attacks on the BMD printer or on the ICX software. By either of these means, attackers could apply two different strategies for altering votes:

Altering only the barcodes. Attackers could cause the BMDs to print QR codes that differ from voters' selections while leaving the human-readable text of the ballot unchanged. Since voters cannot read QR codes unaided, they would be unable to detect the alterations, but, since the QR code is the only part of the ballot the scanners count, the impact would be a change to the tabulation of those individual votes affected and potentially to the election results. The only known safeguard that can rule out such an attack is to compare the human-readable text on every voted ballot to the QR codes, which Georgia has never done in any election and which does not appear to be required or anticipated for future elections.

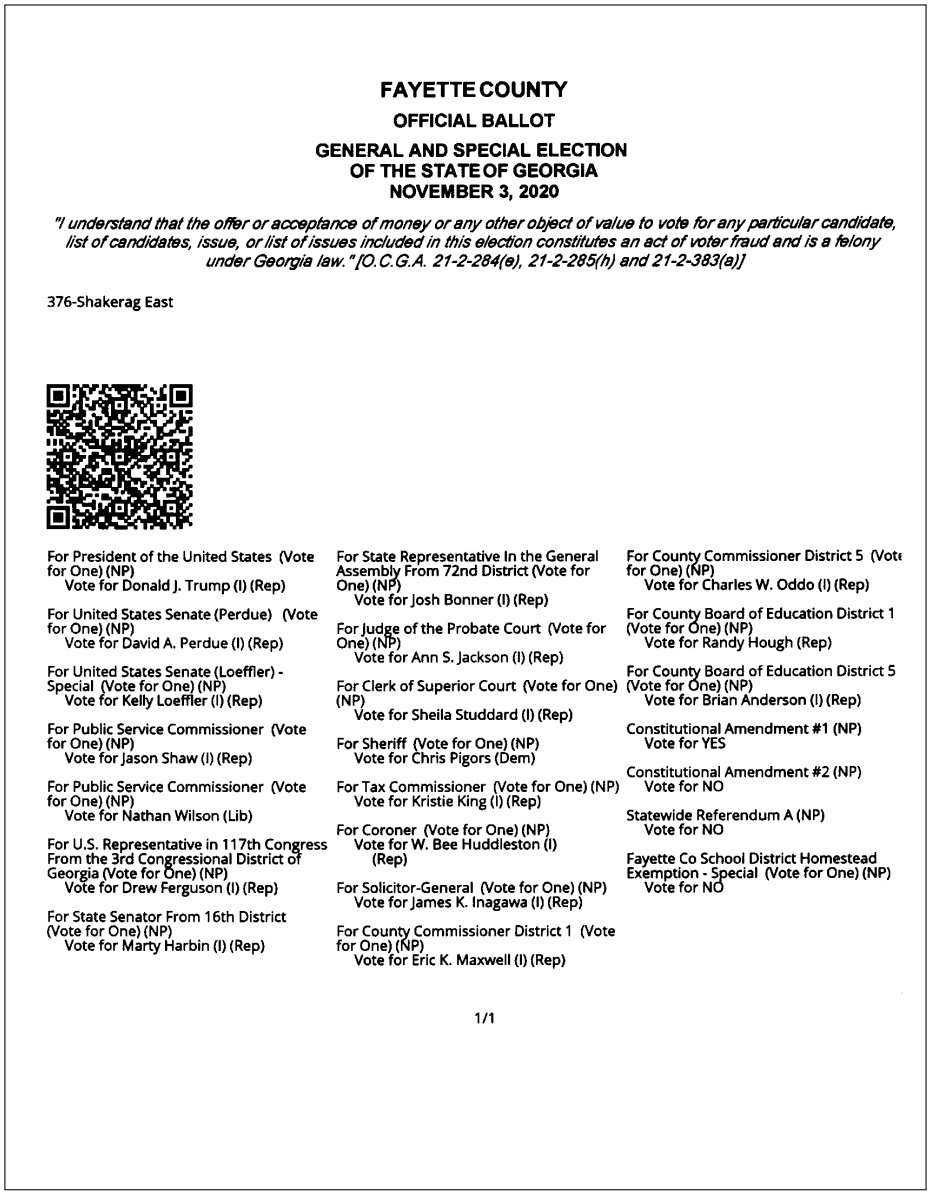
Since attackers might choose to target any race in any election, every race and every election would need to be subjected to a rigorous risk-limiting audit (RLA). Georgia rules currently require an RLA of only a single state-wide *contest* every two years [69]. In the vast majority of races—even high-profile ones, such as the U.S. Senate races in November 2020 and January 2021—the state does not audit the human-readable ballot text at all, and so it is highly likely that barcode-only attacks would go undetected.

Altering *both* the barcodes and the text. Attacks on the BMDs could also change *both* the barcode and the human-readable text on a fraction of the printed ballots, so that both represented the same set of fraudulent selections. Research shows that few voters carefully review BMD ballots [9, 54]. Consequently, when most voters use BMDs, manipulation of enough votes to change the winner of a close race would likely go undetected, and individual voters would be disenfranchised, even if the election outcome were unchanged.[5] No audit or recount could detect this fraud—not even an RLA—because all records of the voter's intent would be wrong. Pre-election or parallel testing also cannot reliably detect such cheating [80]. Even if officials did suspect that the BMDs had been attacked, there would probably be no straightforward way to determine the correct outcome and no way at all to determine each individual voter's intended vote. The only recourse might be to rerun the election.

Both attack strategies could be accomplished using the same technical methods, so attackers can choose between them depending on the contest being targeted. In contests where no audit or recount is likely, attackers can cheat

[5] I review the research concerning voter-verifiability of BMD ballots (which includes my own award-winning peer-reviewed work [9]) in a prior declaration [39, ¶ 23–33]. Data from subsequent research lends further support to my conclusions [54].

REDACTED VERSION



FAYETTE COUNTY

OFFICIAL BALLOT

GENERAL AND SPECIAL ELECTION
OF THE STATE OF GEORGIA
NOVEMBER 3, 2020

"I understand that the offer or acceptance of money or any other object of value to vote for any particular candidate, list of candidates, issue, or list of issues included in this election constitutes an act of voter fraud and is a felony under Georgia law." [O.C.G.A. 21-2-284(e), 21-2-285(h) and 21-2-383(a)]

376-Shakerag East

For President of the United States (Vote for One) (NP)
Vote for Donald J. Trump (I) (Rep)

For United States Senate (Perdue) (Vote for One) (NP)
Vote for David A. Perdue (I) (Rep)

For United States Senate (Loeffler) - Special (Vote for One) (NP)
Vote for Kelly Loeffler (I) (Rep)

For Public Service Commissioner (Vote for One) (NP)
Vote for Jason Shaw (I) (Rep)

For Public Service Commissioner (Vote for One) (NP)
Vote for Nathan Wilson (Lib)

For U.S. Representative in 117th Congress From the 3rd Congressional District of Georgia (Vote for One) (NP)
Vote for Drew Ferguson (I) (Rep)

For State Senator From 16th District (Vote for One) (NP)
Vote for Marty Harbin (I) (Rep)

For State Representative In the General Assembly From 72nd District (Vote for One) (NP)
Vote for Josh Bonner (I) (Rep)

For Judge of the Probate Court (Vote for One) (NP)
Vote for Ann S. Jackson (I) (Rep)

For Clerk of Superior Court (Vote for One) (NP)
Vote for Sheila Studdard (I) (Rep)

For Sheriff (Vote for One) (NP)
Vote for Chris Pigors (Dem)

For Tax Commissioner (Vote for One) (NP)
Vote for Kristie King (I) (Rep)

For Coroner (Vote for One) (NP)
Vote for W. Bee Huddleston (I) (Rep)

For Solicitor-General (Vote for One) (NP)
Vote for James K. Inagawa (I) (Rep)

For County Commissioner District 1 (Vote for One) (NP)
Vote for Eric K. Maxwell (I) (Rep)

For County Commissioner District 5 (Vote for One) (NP)
Vote for Charles W. Oddo (I) (Rep)

For County Board of Education District 1 (Vote for One) (NP)
Vote for Randy Hough (Rep)

For County Board of Education District 5 (Vote for One) (NP)
Vote for Brian Anderson (I) (Rep)

Constitutional Amendment #1 (NP)
Vote for YES

Constitutional Amendment #2 (NP)
Vote for NO

Statewide Referendum A (NP)
Vote for NO

Fayette Co School District Homestead Exemption - Special (Vote for One) (NP)
Vote for NO

1/1

Figure 2: **BMD Ballot, Showing QR Code.** This is a real ballot cast in Fayette County during the November 2020 election, as captured by an ICP scanner. Note the small and densely printed text. Although the selected candidates are printed in human-readable form, the scanners ignore the text and exclusively count the votes encoded in the QR code, which voters have no practical means to verify.

arbitrarily by altering only the ballot barcodes. Otherwise, as long as the margin of victory is likely to be small, attackers can still change the election outcome with low risk of detection by altering both the barcodes and the ballot text on a small fraction of ballots across many BMDs.

Notably, both styles of ballot manipulation are far greater risks when BMDs are used for all in-person voters, as in Georgia, than when only a small fraction of voters use them, as in most other states. When few voters use BMDs, even changing *every* BMD ballot could only affect the outcome of contests with very narrow margins, and successful fraud would usually require cheating on such a large fraction of BMD ballots that it would likely be discovered. This makes the BMDs an unappealing target and reduces the risk that they will be attacked at all. In contrast, Georgia's universal-use BMDs would be a very appealing target, since they expose all in-person voters to potential ballot manipulation.

In sections that follow, I demonstrate ballot manipulation attacks in several contexts: via attacking the BMD's printer, by installing malicious software onto BMDs with physical access, and by spreading malware to all BMDs across wide areas from central locations. In my implementations of these attacks, I alter only ballot barcodes, but altering both the barcodes and the ballot text would require only straightforward changes to the malicious code.

4 Methodology and Testing Process

4.1 Testing Methodology

Security testing is a widely-recognized best practice, especially for critical systems. My tests of the Georgia voting equipment applied a form of the security testing methodology known as Open Ended Vulnerability Testing (OEVT) [59], which was recommended for voting system testing by the U.S. EAC's Technical Guidelines Development Committee. As described by in a report by the National Institute of Standards and Technology (NIST) [59]:

> The goal of OEVT is to discover architecture, design and implementation flaws that have crept into the system which may not be detected using systematic functional, reliability, and security testing and can be exploited to change the outcome of an election, can provide erroneous results for an election, can cause denial of service, can compromise [the] secrecy of [the] vote, or can compromise [the] security audit log.

OEVT pursues this by having testers play the role of an adversary and attempt to compromise the system. They engage in an iterative process in which they: (1) work to understand how the system functions through observation, review of documentation, hands-on experimentation, and reverse-engineering; (2) generate hypotheses about how security might be compromised; and (3) validate those hypotheses through experiments. Forms of OEVT have been applied in comprehensive voting system security reviews commissioned by the Secretaries of State of California [10] and Ohio [57], and in numerous research studies of deployed election equipment [42].

Since I was provided with access to equipment but not to the software source code, I applied a "black-box" testing approach, in which I relied entirely on reverse-engineering and experimentation to discover vulnerabilities. Though less efficient than "white-box" testing (i.e., analysis conducted with access to source code), a black-box approach has the advantage of more closely mimicking the capabilities of the largest number of potential attackers. That is, any vulnerabilities I found could also be discovered by real attackers without access to Dominion's source code.

OEVT methodology has important limitations. It is highly dependent on the skill, resources, and experience of the testers, and also on good luck. I was fortunate that many of the observations that I decided to pursue through detailed testing proved to be productive, but there were many other observations that I decided not to pursue, and I almost certainly overlooked clues to other important weaknesses. Due to time and resource constraints, once I found one way to accomplish an adversarial objective (e.g., installing malware remotely), I usually moved on to another goal, rather than attempting to find all ways of accomplishing it. For these reasons, I stress that while my methodology is effective for discovering and proving the existence of security problems, the vulnerabilities I uncovered are almost certainly not the only such problems affecting the equipment I studied.

4.2 Materials Examined

I received access to Georgia election system components that were provided to Plaintiffs by Fulton County in compliance with this Court's orders. The major components were an ImageCast X Prime (ICX) BMD, serial number 1910250020, running software version 5.5.10.30, and an ImageCast Precinct (ICP) ballot scanner, serial number AAFAJKL0064, running software version 5.5.3-0002.

In addition, I received a Poll Worker Card and a Technician Card (but not a Voter Card) for the BMD, together with the PINs for both cards. I was also provided a Centon USB drive containing an ICX election definition file for a mock election used for testing and training. The scanner came with two compact flash cards prepared for use in the same mock election, as well as an iButton Security Key and passwords for operating administrative functions.

Fulton County did not provide the off-the-shelf laser printer used in conjunction with the BMD. Instead, Plaintiffs acquired a unit of the same model, an HP LaserJet M402dne, from a commercial source. I was not provided access to the Democracy Suite EMS software.

Analysis of the ICX's audit logs indicated that the unit had been previously tested but had not been used in an election. The unit provided to us had two tamper-evident seals on one of its four compartments. I opted not to remove the seals or open the device's chassis during the course of this investigation.

In June 2021, I received access to further election system data. State Defendants provided Plaintiffs copies of data sent to the Secretary of State by Georgia counties following the November 2020 and January 2021 elections. Although this data was significantly incomplete, many counties returned Election Packages—backups created by the Democracy Suite EMS—which I briefly examined to ascertain how counties typically configured their BMDs. I also extracted the ICX election definition file used by Fulton County in the November 2020 election, which I used for further tests.

Five of the county data sets provided by State Defendants contained copies of the installation file for the October 2020 ICX software update, version 5.5.10.32. The presence of this file may indicate that the counties returned data to the Secretary on the same USB drives that they used to receive or distribute the software update, without first wiping the device.

I completed initial testing with the original ICX software version, 5.5.10.30. I later used the update installation file and the official installation instructions (attached as Exhibit A) to upgrade to version 5.5.10.32 and reverify the findings. Except for a vulnerability reported in Section 8 that is a *consequence* of Georgia updating the software, both versions exhibited the same vulnerabilities.

4.3 Testing Process

Throughout the analysis, the voting equipment was maintained in a secure facility in Atlanta. Due to the COVID-19 pandemic, I performed most testing remotely via videoconference, directing on-site work by my assistant, Dr. Springall. I have

personally verified all findings in this report, and all opinions and conclusions are solely my own.

I began working with the Fulton County equipment on September 4, 2020 and conducted 11 work sessions through June 25, 2021. Between sessions, I reviewed documentation, analyzed collected data, performed reverse-engineering, and prepared tests, so as to make efficient use of time with the equipment. The entire process took approximately twelve person-weeks of effort.

4.4 Proof-of-Concept Attacks

For some of the ICX's vulnerabilities, I prepared proof-of-concept attacks that demonstrate how the problems could be exploited by a malicious actor. Such proof-of-concept "exploits" are widely recognized in the security field as a means of proving that a system suffers from a particular technical flaw. However, they are *not* intended to be exemplars of "weaponized" attacks, such as a sophisticated adversary would seek to deploy.

As such, some of these demonstrations have minor imperfections (such as delays or small visual glitches) that a real attacker could remove with moderate investments in engineering and testing. I also built the proof-of-concept vote-stealing attacks to be demonstrated with a particular election definition, rather than to work generally in any election as a real attacker would do. Implementing these refinements would not require the discovery of any further vulnerabilities, so I chose instead to use my limited resources to analyze additional aspects of the equipment.

5 Manipulating Ballots via the ICX Printer

In this section, I show how BMD-printed ballots can be manipulated without any malicious modifications to the ICX hardware or software. I first examine the structure of the ballot QR codes, show that they are unencrypted, and explain how they can be fully decoded. Next, I show that weaknesses in the QR code design make it possible to manipulate ballots in spite of a security mechanism intended to authenticate the QR codes. Finally, I demonstrate that attackers can automatically manipulate ballots cast on the ICX with no access to the BMD itself, by instead attacking the attached off-the-shelf laser printer. I show how such an attack can be implemented by adding concealed malicious hardware to the printer, which could be accomplished as part of a supply-chain attack.

5.1 Decoding Ballot QR Codes

Dominion's documentation claims that the QR codes are encrypted [19, § 2.6.1.1], and, at least as recently as January 2021, Secretary of State Chief Operating Officer Gabriel Sterling has repeated this claim to the media as a security feature of Georgia's voting system [91]. In actuality, as I testified last year, no part of the QR codes is encrypted [40, ¶ 37–40]. While voters have no practical way to read or verify the votes encoded in the QR codes, they can be decoded by attackers and can be replaced or manipulated to steal voters' votes.

Although the QR codes are not encrypted, they use a data format this is incompatible with most off-the-shelf barcode reader software. A QR code can encode data in several data formats: numeric, alphanumeric, or byte mode [30]. Byte mode can encode arbitrary data, but QR code readers typically interpret the byte sequence as UTF-8 or Latin-1 encoded text. If an application needs to represent arbitrary binary data in a QR code, the recommended practice for ensuring compatibility is to encode the data using characters available in alphanumeric mode (e.g., Base45 encoding) [30]. However, the ICX QR codes appear to be designed only for compatibility with Dominion scanners. They encode binary data in byte mode, and the data typically begins with a byte with value zero. As a result, most QR code reader software either fails to read them because the data does not represent valid UTF-8 or Latin-1 characters or incorrectly treats the zero byte as the end of a null-terminated string.

In August 2020, my research group tested reading the ICX QR codes with a variety of publicly available barcode reader apps for Android and iOS devices. At the time, only one app we tested was able to read them correctly (Scandit Barcode Scanner for iOS [73]), and later versions of that same app no longer do. Several publicly available programming language libraries for reading QR codes had similar compatible problems when used with their default settings. However, we found that we could correctly decode the data using recent versions of the open-source ZBar barcode reader library by setting the `ZBAR_CFG_BINARY` option to force the software to emit the data as raw bytes. For example, using ZBar version 0.23.90,[6] ICX QR codes can be decoded with the command:

[6] Available at https://github.com/mchehab/zbar/releases/tag/0.23.90.

```
zbarimg --quiet --raw -Sbinary ballot.png | hd
```

After extracting the raw data from the QR codes, my research group reverse-engineered the binary data format. To do so, we examined Dominion's ImageCast Remote Accessible Vote-By-Mail (RAVBM) software, a web-based app that generates a ballot with a similar QR code for printing and returning through the mail [24]. (Since ImageCast Remote runs in the voter's browser, the JavaScript source code that it uses to generate the QR codes is publicly visible.) We also examined ICX QR codes from publicly available ballot scans from a variety of elections and determined that they used the same data format. Through this process, we created a computer program (`dvsqrtool.py`) that interprets and unpacks all data fields in the ICX QR code.[7]

The decoded data contains the voter's selections, write-in votes, and ballot metadata. No encryption key is necessary to extract this data, which demonstrates that the QR code is not encrypted. The data structure represents voter selections as a series of binary digits (ones and zeros), as shown in Figure 3. Each digit corresponds to one of the available candidates, typically in the same order that the contests and choices are displayed on the BMD's screen or on the equivalent hand-marked ballot. A `1` signifies that the candidate was selected, and a `0` signifies that the candidate was not selected. Therefore, with knowledge of the ballot design, the selected choices can be readily extracted from the QR code.

5.2 Defeating QR Code Authentication

Issue: *ICX QR codes are not protected against "replay" attacks, so copies of valid QR codes will be accepted as genuine.*

As an authentication mechanism, the QR code contains a cryptographic message authentication code (MAC) computed using the HMAC-SHA256 algorithm. A MAC is a value (a number) calculated based on an input and a secret key. Without knowing the key, it is infeasible to calculate the correct MAC for a modified input. In a given election, the ICX and ballot scanner have copies of the same key. Whenever an ICX generates a QR code, it uses this key to calculate the MAC of the ballot data. When a scanner reads the QR code, it extracts the data, repeats the MAC calculation using its copy of the key, and verifies that the MAC value it calculated matches the MAC in the QR code. Under the assumption that an attacker cannot discover the secret key,[8] this arrangement allows the scanners to confirm that the data in the QR code really was generated by an ICX and was not subsequently modified.[9]

[7] This work was completed in connection with my research at the University of Michigan before Plaintiffs received the Dominion equipment and without use of confidential information. Therefore, I consider it to be outside the scope of the Protective Order.

[8] In fact, the MAC key is not well safeguarded. I show in Section 6.1 that the key used throughout a county can be easily extracted from any Poll Worker Card, given brief physical access to the card and its PIN. It can also be extracted from an ICX after L&A testing by escaping kiosk mode using the techniques in Section 8.

[9] A MAC is very different cryptographic algorithm than a digital signature, although Defendants' experts have repeatedly confused the two [40, ¶37–39]. Both are sometimes

REDACTED VERSION

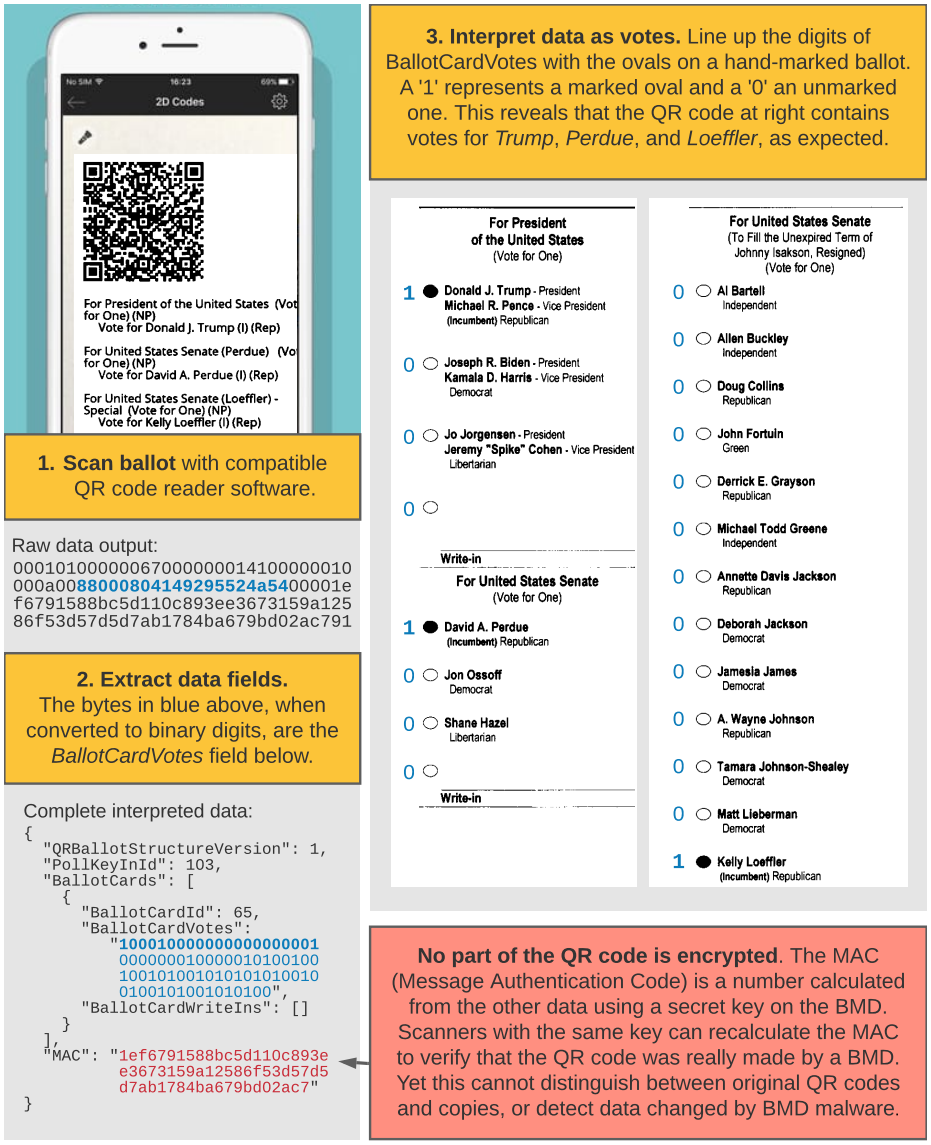


Figure 3: **Decoding the QR Code.** Using the procedure illustrated above, the QR code from Fig. 2 can be fully decoded. No secret information is required, because the QR code is not encrypted. Although the data includes a MAC, the design does not protect against duplicated QR codes or malware running on the BMD.

Despite this use of a MAC, attackers can manipulate ICX QR codes through several means to alter recorded votes or cast fraudulent votes. The ICX QR code design as used in Georgia has a serious weakness: the codes do not contain a serial number or other unique identifier, so, for a given ballot design, all QR codes that contain identical votes are indistinguishable, including having identical MACs. As a consequence, there is no mechanism for detecting *duplicate* QR codes. This enables two important attacks:

Copying Ballots A copy of a genuine ICX ballot will be indistinguishable from a second genuine ICX ballot with the same votes. In tests, the ICP accepted ballots copied using an office photocopier (see Section 11.1). This could allow a variety of ballot-box stuffing attacks.

Replay Attacks Although the MAC prevents attackers who do not know the secret key from generating new valid QR codes, they can still substitute other valid QR codes they have seen before. In a "replay" attack, attackers observe genuine printed ballots and save copies of QR codes with votes they favor. They then alter ballots with votes they disfavor by replacing the QR codes with the ones they have saved. Since the QR codes on the altered ballots contain valid MACs, the scanners accept them as genuine, even though they are duplicates. I demonstrate this style of attack and discuss the implementation details below.

5.3 Demonstration Hardware-Based Attack

An attacker can implement a fully-automatic ballot manipulation attack without tampering with the ICX itself in any way, by instead targeting the laser printer attached to the BMD. Georgia's BMDs use off-the-shelf HP LaserJet M402dne printers connected via a USB cable. Like most modern printers, they contain capable embedded computers that run complex, field-updatable software. By modifying the printer's software or hardware—or even by hiding tiny malicious hardware in a modified USB cable [38, 60]—an attacker can arbitrarily change what the printer prints. This can be employed to alter the ballot QR codes (alone or in conjunction with ballot text) and steal votes.

Since the printer is an off-the-shelf device, it is likely to receive less security scrutiny from officials than the ICX, even though attacks on the printer could be equally consequential. Attackers could potentially compromise the printers at any time during the lifecycle of the voting system, including before they are delivered (in the supply chain), while in storage, or during transport to or from polling places.

I developed a proof-of-concept attack to illustrate these risks. It consists of hardware hidden by the attacker inside the printer's housing that manipulates

used to verify data integrity. For purposes here, the most important difference is that anyone who has the key needed to verify a MAC can also *forge* valid MACs for any data they choose. In contrast, the information needed to verify a digital signature can be widely distributed or even made public without jeopardizing its security.

Figure 4: **Demonstration Malicious Hardware.** I developed a hardware-based attack that modifies data sent from the ICX to the printer, altering ballot QR codes to change recorded votes. The attack device (the two red modules seen in the right photo) is completely hidden inside the printer's plastic housing. Similar malicious hardware could be added in the supply-chain or while in storage.

the data sent from the ICX to the printer. I demonstrated an early version of the attack during the September 2020 hearing, after having access to the ICX for only about one week. To implement the attack, I used a pair of Raspberry Pi Zero W devices. These are small (approximately 1 × 2.5 inches), self-contained computers with WiFi and Bluetooth radios that are capable of simulating a USB device or host system. They are widely available for a cost of about $10.

In my attack implementation, one Pi Zero receives ballot data via the original printer cable that attaches to the BMD; I refer to this device as Pi-Input. The second Pi Zero connects to the printer itself and outputs data to be printed; I refer to it as Pi-Output. Both run the Linux operating system. A real attacker would likely integrate these functions into a single purpose-built hardware device, but I needed to split them because the off-the-shelf Pi Zeroes I used each have only a single functional USB port. Even when using two Pi Zeroes, the entire setup (shown in Figure 4) is small enough to be concealed in the empty space within the laser printer's housing, and it is low-power enough to be operated from the printer's internal power supply.

Redaction

Pi-Input is configured to behave as a USB peripheral [redacted] and runs software I developed (`in/device.py`) that simulates the printer. When connected to the BMD, the Pi Zero sends USB device descriptors and identification strings that match those of the LaserJet M402dne. This makes it indistinguishable from the real printer to the BMD's software. However, when the BMD sends data to print, Pi-Input relays that data (over a local wireless network) to the second Pi Zero, which proceeds to manipulate it.

Pi-Output connects to the real printer and operates as a normal USB host. It runs software I wrote (`out/prproxy.py`) that receives from Pi-Input the data that the BMD attempts to print. My software parses the PCL data to extract the QR code as a bitmap image, passes it to the `zbarimg` barcode reader tool to decode the QR code data, and uses my `dvsqrtool.py` tool (discussed in Section 3.2) to extract the votes.

For this attack, I assume that the adversary does not know the secret key used to compute the MAC in the QR code. Without this key, the attacker cannot modify the data in the QR code, but they can still manipulate votes by performing a replay attack, i.e., selectively copying valid QR codes from previously-seen ballots. To accomplish this, Pi-Output inspects the votes in each QR code to determine whether the attacker's preferred candidate is selected. Then:

If the attacker's candidate is selected, the device passes the ballot to the printer unmodified but saves a copy of the QR code to its internal storage.

Otherwise, the device picks one of the stored QR codes at random and substitutes it for the QR code sent by the BMD. Since the stored QR codes contained valid MACs, and the system design does not detect duplicated QR codes, these copied QR codes will be accepted as valid by the ballot scanner.

As a result, once at least one ballot has been voted for the attacker's preferred candidate, subsequently printed ballots will contain QR codes that encode votes for that candidate.[10]

For demonstration purposes, I hard-coded the target contest and favored candidate, and I programmed the device to cheat as often as possible.[11] In practice, an attacker could remotely (e.g., using WiFi or Bluetooth) select the fraction of votes to shift and which candidate in which contest should receive them. Similarly, the attacker could remotely enable or disable the cheating, thereby defeating any pre-election testing. With wireless control, the attack device could be installed in the printer once and cheat in any subsequent election.

Adding hardware to the printer is only one of several ways that attackers could manipulate ballots cast using Georgia's ICX BMDs. An easier and more powerful mode of attack would be to modify the software in the ICX itself. When I demonstrated the printer attack prototype in September 2020, I testified that software-based attacks on the ICX were very likely achievable with further analysis. This has proven to be the case. In later sections of this report, I will explain how it is possible to construct vote-stealing malware that runs entirely in the ICX, and how attackers can infect ICXs with such malware remotely throughout entire counties or even the entire state.

[10] In a realistic attack scenario, the attacker would likely choose to alter only a fraction of the ballots, so as to avoid drawing suspicion.

[11] My proof-of-concept implementation sometimes introduces a spurious delay of up to about 20 seconds before the ballot is printed. The most likely cause is a bug in the code. Having demonstrated the attack concept, I opted not to spend further resources debugging and removing the delay, and instead focused on attacking the ICX software.

6 Attacks Against ICX Smart Cards

Smart cards, such as many modern debit and credit cards, have an embedded integrated circuit chip that exchanges data with the card reader. Some smart cards are capable of storing secret data securely and performing cryptographic operations. Such cards are often used to authenticate identity in high-security applications, such as the U.S. Department of Defense (DoD) Common Access Card (CAC) that provides access to defense computer networks and systems [16].

The ICX uses smart cards to authenticate voters, poll workers, and service technicians. There are kinds of cards:

Technician Cards Service technicians are assigned a Technician Card and PIN. By inserting the card and entering the correct PIN on the screen, they can access the Technical Administrative menu shown in Figure 5a. This menu is used before each election to load new election definitions from a USB drive. It also allows more sensitive actions, such as exiting the ICX application and accessing the underlying Android operating system, from which the ICX's software can be updated or modified.

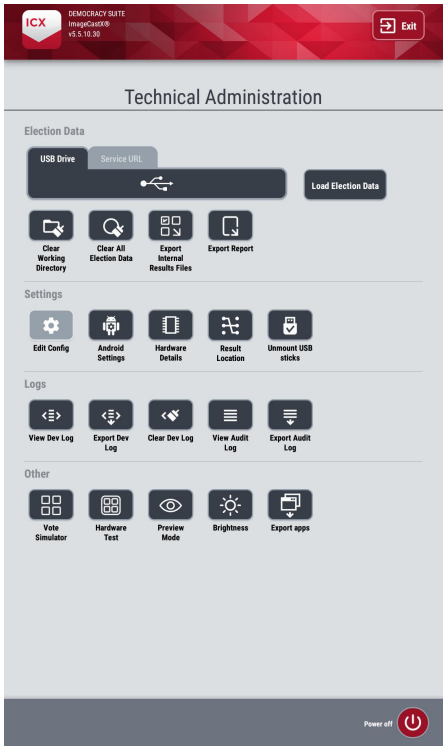
Poll Worker Cards Poll workers are assigned a Poll Worker Card and PIN, with which they can access the Poll Administration menu shown in Figure 5b. This menu allows the poll worker to open the polls using a previously-loaded election definition, manually activate a voting session (without use of a Voter Card), reset the machine's public counter, close the polls, or shut down the BMD. Poll Worker Cards are specific to each election and contain the cryptographic keys necessary to operate the ICX in the election.

Voter Cards When a voter checks in at a polling place, a poll worker uses an electronic poll book to issue them a Voter Card. The voter inserts the Voter Card into the ICX to unlock the BMD for a single voting session using the voter's assign ballot style. Upon printing the ballot, the ICX deactivates the Voter Card, preventing it from being used again, and the voter returns it to a poll worker.

I examined the communications between the ICX and the smart cards to determine the authentication protocol and evaluate its security. While I expected the BMD to use a modern cryptographic challenge-response protocol, which would render the cards resistant to cloning and forgery, the machine instead uses a simplistic and highly insecure protocol. The actual protocol is conceptually very similar to the protocol used by the Diebold AccuVote DREs, which also used smart cards. The Diebold smart card protocol was shown to be insecure by researchers as early as 2003 [53], and its vulnerabilities were documented in detail by the California Secretary of State's Top-to-Bottom Review in 2007 [11]. The ICX smart cards used today suffer from essentially all of the same vulnerabilities and some serious additional ones.

All three kinds of ICX smart cards use the same protocol, which is implemented as a series of ISO 7816-4 commands:

REDACTED VERSION



(a) Technical Administration Menu

(b) Poll Administration Menu

Figure 5: **ICX Administration Menus.** Unlocking the ICX with a Technician Card or a Poll Worker Card provides access to a variety of privileged operations.

1. The ICX attempts to open a particular file stored on the card, and the card responds with whether the file exists. Technician and Poll Worker Cards use file ID 0x[redacted], while Voter Cards use file ID 0x[redacted]. This allows the ICX to determine whether a voter card or an administrative card was inserted. **Redaction**
2. The ICX sends a password to the card to unlock the file. For Technician and Poll Worker Cards, the password is a PIN that is entered on-screen by the user. For Voter Cards, the ICX automatically sends a preconfigured PIN.
3. The card checks whether the PIN matches a value stored on the card before allowing access to the file. I assume (but did not verify) that the cards lock themselves and prevent further use if too many incorrect PINs are attempted.
4. Once the file is unlocked, the card allows the ICX to read or write to it.

The file formats for all three types of cards are simple and readily determined by inspecting the data. Each file consist of 36 records, each up to 15 bytes long, for a maximum length of 540 bytes. Their more relevant features are:

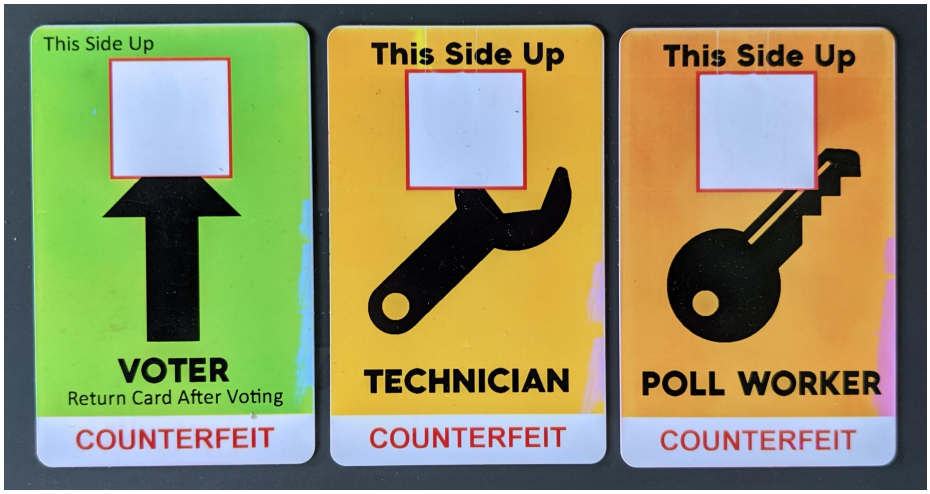


Figure 6: **Forged ICX Smart Cards.** Weaknesses in the ICX authentication protocol allow an attacker to read and forge Voter, Technician, and Poll Worker cards. I added explicit markings and allowed minor discolorations to minimize any risk of misuse, but a real attacker could create nearly indistinguishable counterfeits.

- *Technician Cards:* The first record is the value 0. Other records contain the user's name, the date and time that the card was created, and the date and time that it expires.
- *Poll Worker Cards:* The first record is the value 1. In addition to records found on the Technician Card, the file contains all of the election-specific cryptographic keys and other secrets that the ICX and scanner use for security: the admin PIN, the encryption key and IV, the MAC key, and the Election Signature (a secret value that uniquely identifies the election).
- *Voter Cards:* Records contain the ballot style, language, and accessibility mode, whether the card has been used, the date and time it was activated, and the Election Signature value (to prevent the card from being used in a different election).

I determined that an attacker can extract the data from all three kinds of cards, as well as create counterfeit cards (shown in Figure 6). In the sections that follow, I explain how these capabilities could be used for a variety of attacks.

6.1 Extracting Election Secrets from Poll Worker Cards

Issue: *Anyone with access to a single Poll Worker Card and the corresponding PIN can easily extract secret keys and other values used for securing election data throughout the county.*

The ICX smart card protocol does not authenticate the device reading the card. As a result, anyone with the correct PIN can read the data on the card in a

few seconds by simply following the protocol. I created a simple Python program (`cardutil.py`) that uses a commodity USB smart card reader and mimics the ICX's behavior, allowing us to extract the contents of the cards provided by Fulton County.

This weakness causes a serious information exposure vulnerability due to the cryptographic secrets stored on Poll Worker Cards. With access to the encryption and MAC keys from the Poll Worker Card, an attacker could decrypt or alter the ballot definitions used by the scanners and BMDs, forge ballot QR codes, or decrypt or modify election results on scanner memory cards before the results are returned to the EMS for reporting.

Poll Worker Cards and PINs are distributed to every polling place and entrusted to thousands of volunteer poll workers across the state during every major election. It would be practically impossible to ensure that none of these cards could be temporarily accessed by a malicious party.

County election databases from the November 2020 and January 2021 elections shows that Georgia counties use the same cryptographic keys county-wide for each election. This means that if a single Poll Worker Card and PIN anywhere in a county is temporarily accessed by an attacker, the attacker can easily obtain the keys necessary to compromise election data throughout the county.

To make matters worse, if a county suspected that its keys had been compromised, the only way to change them would be to load new election definitions into every ICX and ICP in the county. Doing so would likely take days or longer and might necessitate repeating logic and accuracy testing on every device.

There problems are the result of an extremely dangerous approach to cryptographic design. Best practice calls for avoiding sharing keys widely over multiple devices or authentication tokens, so as to prevent the compromise of any one device or authentication token from compromising them all.

6.2 Forging Technician Cards to Install Malware on any ICX

Issue: *Anyone can create forged Technician Cards without using any secret information. Such cards could be used to access any ICX's Android operating system and the ability to install malware.*

Although Technician Cards allow the user to access highly sensitive functions of the BMD, the ICX protocol does not authenticate them using any secret values. This makes it possible to create a forged Technician Card without knowledge of any passwords, PINs, or secret keys.

To create forged technician cards, I used the Java Card platform. A Java Card is a smart card that can execute small software applications written in the Java programming language, allowing it to emulate the behavior of other smart cards. I used [redacted] Java Cards, which are commercially available for less than $10 each [redacted]

Redaction
Redaction

I programmed a Java Card as follows. No matter what file ID the machine requests, the card always reports that it is present. (The first request is usually for an administrative card, so the attacker does not need to know what the real

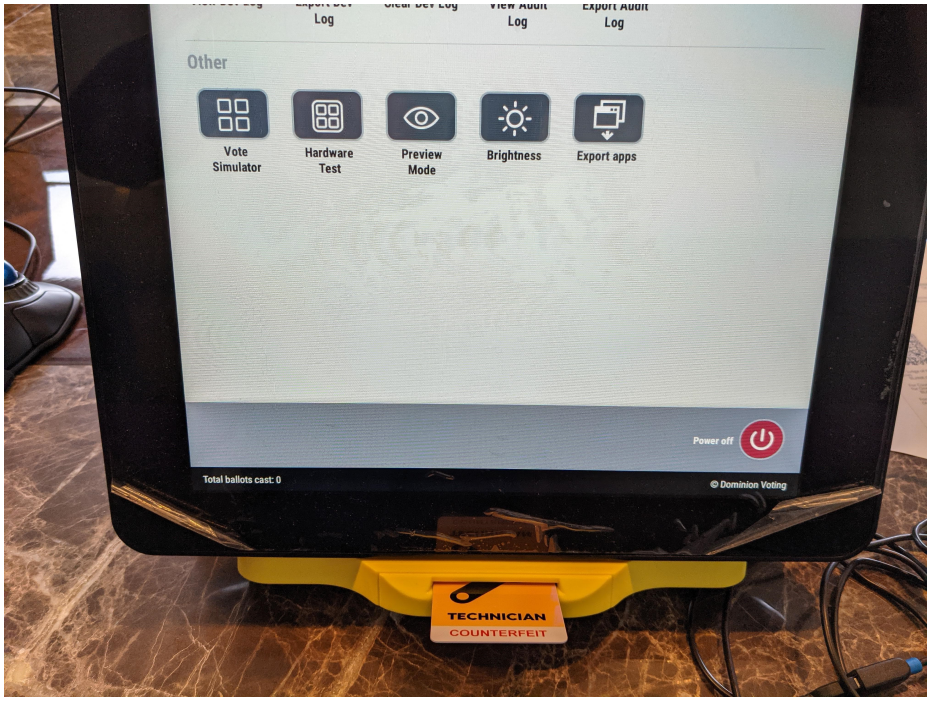


Figure 7: **Forged Technician Card.** Technician Cards can be forged without using any secret information. The self-created card can be used to unlock any ICX in Georgia (and likely those in other jurisdictions) and install malicious software.

file ID is.) To unlock the file, it accepts *any* password, so the user can enter any PIN. The card then returns a file that is *completely empty*, with every record consisting of zeroes. Remarkably, the ICX accepts the card as if it were a genuine Technician Card.

ICX Technician Cards are not restricted to a particular election or a particular jurisdiction. Consequently, the forged Technician Cards I created will work in any ICX across the State of Georgia, and likely in any other jurisdiction that uses a compatible version of the machine.

After forging a Technician Card, an attacker with physical access to a BMD can exit the ICX application and access the underlying Android operating system. With this access, the attacker can arbitrarily change the BMD's configuration, alter audit logs, or install malicious software.

6.3 Creating "Infinite" Voter Cards

Issue: *Voters can clone Voter Cards or create "infinite" Voter Cards that allow printing an unlimited number of ballots of any available ballot style.*

Forging Voter Cards requires additional steps compared to forging a Technician Card, because the BMD will only accept Voter Cards that contain the correct Election Signature, a secret value specific to the current election and county. An attacker could obtain the Election Signature by several means, such as the attack in Section 6.1, but I developed an attack method that requires only the level of access of an ordinary voter:

1. The voter enters the polling place and is issued a real Voter Card,[12] which contains the Election Signature. However, the voter cannot read the card without the election-specific Voter Card PIN.
2. To obtain the PIN, the voter inserts a specially programmed Java Card into any BMD in the polling place. I programmed a Java Card to mimic the initial steps of the ICX protocol. It reports that file ID 0x[redacted] is present, which causes the ICX to send the card the Voter Card PIN to unlock the file. The Java Card records the PIN in its internal memory and reports that it was invalid, causing the BMD to reject the card. **Redaction**
3. The voter uses a smart card reader to read the Voter Card PIN from the Java Card. They then insert the real Voter Card into the reader and use the PIN to extract its contents (as in Section 6.1), including the Election Signature.
4. Finally, the voter creates a forged Voter Card by loading the data into a second Java Card, using a process similar to the attack in Section 6.2.

These steps can be completely automated, so that the attacker need only insert the three cards into the reader for a few seconds each. An attacker in the polling place could use a battery-powered Raspberry Pi and card reader concealed on their person. Alternatively, they could smuggle the real Voter Card out of the polling place and prepare the forged card elsewhere.

When creating the forged card, the attacker can modify its data or behavior. For instance, the attacker can change the ballot style identifier to cause the BMD to print a ballot for a jurisdiction in which they are not eligible to vote.

An attacker can also create an "infinite" Voter Card that does not deactivate after the ballot is printed, allowing it to be used arbitrarily many times. Normally, the BMD deactivates voter cards by writing a value to a particular record on the card. To circumvent this, I programmed a Java Card to ignore these write operations and leave the card in an activated state, allowing it to be used indefinitely. An attacker and their accomplices could use an "infinite" Voter Card to each vote multiple ballots, although there is a risk that poll workers would notice someone printing or scanning more than one ballot in a voting session.

I previously demonstrated a similar infinite voter card attack against Georgia's old AccuVote DREs. The attack was more complicated to carry out against the DREs than against the BMDs, since the DREs verified that the card had been deactivated before returning it to the voter. The ICX does not even perform this basic check.

[12] Fulton County did not provide a Voter Card as part of the equipment. However, I was able to construct a working counterfeit Voter Card through reverse engineering.

7 Constructing ICX Malware

From an attacker's perspective, the most powerful position from which to manipulate votes is by using malicious software (malware) installed on polling place equipment. Georgia's ICX BMDs are highly vulnerable to malware-based attacks. This report describes numerous ways that an attackers can install malware on the machines, either with physical access (Section 8) or remotely (Section 9). Once malware is installed by any of these routes, other weaknesses in the ICX give the attacker complete control over the behavior of the machine and the ability to conceal the malware's presence very effectively.

In this section, I explain how an attacker can create malware for the ICX that, once installed by any of the methods described elsewhere in this report, manipulates printed ballots to steal votes. I explain how such malware can defeat the technical and procedural safeguards applied in Georgia, including logic and accuracy testing (LAT), firmware validation, and hash verification practices. I also present working proof-of-concept malware that demonstrates these attacks.

As I explain in Sections 8.3 and 9.3, the ICX is subject to multiple means of privilege escalation, which allow attackers to obtain "root access", i.e., full control of the device. Such access provides a variety of means by which attackers can modify the system's behavior to introduce malicious functionality, including modifying the operating system, intercepting system calls, patching the application in memory, and modifying stored data, among others. Malware could potentially utilize any of these means to manipulate ballots cast on the ICX. However, I will describe and demonstrate a different technique that is simple and effective: directly modifying the ICX Android application.

7.1 Overview of the Approach

Issue: *The ICX does not require that applications be signed by a trusted source, allowing the installation of arbitrary APKs.*

The ICX's election functionality is implemented as an Android application (the "ICP App") that is automatically launched when the device powers on. The ICX App is technically very similar to a smartphone app that a consumer would download from an app store, except that it is either pre-installed at the factory or manually installed as a software update in the form of an Android application package (APK). The actual APK filename can vary, but for simplicity I will refer to it as `ICX.apk`.

Widely-available software tools allow an attacker who can obtain a copy of the APK to quickly reverse-engineer its functionality and add arbitrarily-complex malicious logic. By this method, an attacker can alter the original APK to generate a new APK that appears identical to users but contains malicious behavior. This malicious app can simply be installed in place of the real software. Once installed, the modified APK has access to all of the same data, cryptographic secrets, and device capabilities as the original app, making possible a very wide range of attack payloads.

The Android platform requires every APK to be digitally signed by the software developer. Dominion could have used digital signatures to limit installation of apps to those signed by the company, which would complicate attempts to install a maliciously modified app. However, my tests show that the ICX does not verify the identity of the signing party. It allows the installation of APKs created, compiled, and signed by anyone. Consequently, APK code signing present no obstacle to installing malware on the ICX.

7.2 Obtaining the Real APK

Issue: *The ICX App's APK can be easily extracted given only brief, one-time access to a single BMD.*

In order to modify the ICX App, the attacker must first obtain a copy of the original software. This is easy to accomplish, given temporary physical access to an ICX. For this investigation, I copied the APK to a USB stick by using a Technician Card to access the Technical Administration menu and pressing the "Export apps" button, shown in Figure 5a.

For an attacker to do this, they would only need access a single ICX once for a few minutes. (As explained in Section 6.2, anyone can forge a Technician Card that will work in all ICXs, so access to a genuine Technician Card and PIN is *not* necessary.) Such access could potentially be gained with the help of an insider accomplice, or by breaching physical security at any point in the equipment's lifecycle: before it is delivered to the state, while it is in storage, while it is being set up and tested before an election, during transport to or from a polling place, or potentially while in use at a polling place.

Although Georgia uses physical security measures to make such access more difficult, these measures are imperfect (see Section 11.3), and it is implausible that they could prevent a determined attacker from ever accessing even a single device. Brief, one-time access to any one of the tens-of-thousands of BMDs used across Georgia would be sufficient—or to a machine from any of the many other states and local jurisdictions that use the same Dominion BMDs and software version for accessible voting. The physical security procedures for election equipment vary from jurisdiction to jurisdiction, and Georgia cannot ensure that ICXes used elsewhere are well protected.

Alternatively, an attacker could obtain a copy of the `ICX.apk` file used to update the ICX software, such as the update that Georgia installed in October 2020. The software update process involves distributing the file to all counties, copying it to hundreds or thousands of USB sticks (which are necessarily unencrypted), and having workers insert them into every BMD. An attacker who obtained a copy of any one of these USB sticks would have all the necessary information to create working ICX malware.

7.3 Decompiling and Reverse-Engineering

Having obtained the `ICX.apk` file, the next step is to reverse-engineering it to understand the functionality and how to modify it. This can be accomplished

by using the publicly available `apktool` software [46] to disassemble the original APK and translate the code into "smali" [52], an annotated, human-readable representation of the Dalvik bytecode used by Android. Although reverse-engineering the APK file is more labor-intensive than working with the original software source code, the APK file is likely to be more readily available to a wider range of potential attackers.

Based on my experiences developing similar malware for both Georgia's DREs and its new BMD system, I can compare the difficulty of attacking both types of equipment. Qualitatively, reverse-engineering the ICX app was much easier than reverse-engineering the software used in the AccuVote DREs [31]. The DREs ran Windows CE applications that were compiled into native code for SuperH and ARM processors. Unlike this native machine code, the Android Java bytecode as used in the ICX includes package, type, variable-name, and other information that makes it much easier for an analyst to interpret what the code is doing. The manual effort required to reverse-engineer it was significantly less than I expected, making it possible to alter the ICX App's functionality with relative ease. Quantitatively, reverse-engineering the app and developing basic proof-of-concept malware required approximately 25 hours of effort. This is far less effort than was required when I reverse-engineered the AccuVote DRE and developed similar malware in 2007. For these reasons, I conclude that malware is easier to create for the ICX system than it was for Georgia's old DRE system.

7.4 Modifying the ICX App to Change Votes

Due to the structure of Android applications, it is relatively straightforward to make arbitrary changes to the ICX App's behavior. We used Java, a high-level programming language, to implement demonstration malicious functionality as a Java package. Using a high-level programming language is much less labor intensive than writing the malicious logic in low-level bytecode. We compiled the Java package into low-level smali instructions using the publicly available `java2smali` software tool [51] and inserted the smali files into the disassembled APK's file structure. This arrangement allows the new code to be invoked with only small, targeted changes to the original app's code.

For example, in my demonstration malware, one place where such malicious logic is injected is in the code that generates the QR code for printing. Through reverse engineering, we located the existing code that constructs the vote data that will be encoded in the QR code. Changing just two bytecode instructions in this function[13] causes it to pass the data to a function in the new Java package, giving the malicious logic an opportunity to change the data before the QR code is produced.

As a simple demonstration, I implemented malicious logic that modifies the QR code so that the vote recorded for a specific "Yes or No" contest is always "No". The logic clears the "No" bit and sets the "Yes" bit for a specific byte

Redaction

[13] Specifically, the function [redacted]

within the data representation (see Figure 3) and returns the modified data to the original logic to be packaged into the QR code and printed. The result is that the data in the QR code—and the vote counted by the scanner—reflects a fraudulent choice I control, rather than the voter's intended selection.

An attacker could, of course, implement different or more complex logic to determine when and how to cheat. Malware on the ICX has access to the complete ballot design, and could be programmed to cheat in favor of candidates from a specific party, in contests for a particular office, or in particular kinds of elections. For example, it could always favor one party's candidate in U.S. House races during general elections. An attacker also could choose to change only the QR code or both the QR code and the human-readable text. Malware with such variations could be constructed in the same manner as the proof-of-concept malware described here.

7.5 Defeating Applicable Defenses

Malware running on the ICX can defeat the various technical and procedural defenses that the Dominion system and the State of Georgia currently employ.

Defeating Logic and Accuracy Testing In logic and accuracy testing (LAT), workers cast a small number of votes with known selections, then check whether the voting system's output reflects the correct totals. This form of testing is designed to detect errors in the ballot design or counting logic. It can be easily defeated by ICX malware.

Georgia's LAT procedures (Exhibit B) involve only minimal testing of the ICXs. Only a single test ballot per ICX is required to be printed. To avoid detection, the demonstration malware simply tracks how many ballots have been printed since the machine was turned on and skips cheating on the first n ballots (for an attacker-configuration number). If Georgia were to improve its LAT process by testing with a greater number of ballots, attackers could simply increase the number of ballots the malware skipped accordingly.

Even if the state adopted a much more complex LAT procedure, so long as the testing process was publicly documented, attackers could design malware to maximize cheating while minimizing the probability of getting caught. Much as Volkswagen's emission systems were famously designed to detect that they were being tested by the EPA and to only cheat while not under test [84], ICX malware can be programmed to detect and circumvent LAT. For example, malware could be programmed to only cheat on the day of the election, or only during specific hours on that day. It could also be programmed to monitor how the machine was used and to only start cheating if the rate of voting, pattern of votes, number of corrected mistakes, and other characteristics matched the expected behavior of real voters. No practical method of pre-election or parallel testing can rule out malware-based fraud [80].

Defeating the QR Code MAC Although the QR code contains a cryptographic message authentication code (MAC) that scanners use to verify its integrity (as explained in Section 5.2), this poses no obstacle to ICX malware.

Figure 8: **Defeating Hash Validation.** The ICX App displays the SHA-256 hash of its APK on the screen, as shown here. However, this behavior is controlled by the app itself, so a maliciously modified app can simply show the expected hash value instead of its real one, thereby avoiding detection.

The demonstration malware changes vote data before the app computes the MAC. This allows such malware to add, remove, change, or spoil votes in the QR code while ensuring that the MAC remains valid. Alternatively, since the secret key used to generate the MAC is necessarily accessible to the ICX App, malicious logic in a modified app could use the key to generate valid MACs itself.

Defeating APK Hash Validation As shown in Figure 8, the ICX can display the SHA-256 hash of the installed APK on its screen, supposedly allowing both election officials and voters to confirm what software is running. However, much like the QR code MAC, this hash value is computed by the ICX App itself and can therefore be trivially defeated by malicious logic added to the app.

In the demonstration malware, we identified the code that computes and displays the hash,[14] and modified it to simply replace the computed hash value with the hash of the unmodified APK. This ensures that the ICX always displays the official APK's hash even though it is running a maliciously modified APK.

Redaction

[14] Within the function [redacted]

Defeating External APK Validation As described in Section 7.2, the ICX App contains functionality to export the currently installed APK to a USB stick for verification. Once the APK file has been exported, its hash can be securely computed using a trusted, external device. Exhibit C shows that this was the method used by Pro V&V used in November 2020 to validate the software on a small number of ICXs in six Georgia counties.

A malicious ICX App can easily defeat this safeguard, too, because the export process is performed by the app itself. Just as the modified app can display the hash of the original APK, it can also export the original APK file instead of its own. To accomplish this, we store a copy of the original APK and modify part of the export code[15] to change the location from which the exported APK file is copied to be the location of the original APK. Since the exported APK is identical to the original APK, any hash validation or forensic analysis of it will fail to detect the malware, including the kind of analysis Pro V&V performed.

Defeating Voter Verification and Auditing As discussed in Section 3.2, voters have no practical way to verify the contents of the QR codes. Since the scanners read only the QR codes, and the voter can only review the printed text, there is no way for voters to verify the portion of their ballots that is actually tallied. Therefore, attacks that change the QR code and leave the human-readable portion of the ballot unmodified would almost certainly not be detected by voters.

In principle, *election officials* could verify the QR codes by decoding them and comparing the output to the text on the ballots. To our knowledge, no jurisdiction has ever done so, and Georgia has announced no plans to do so.

A rigorous risk-limiting audit (RLA) would also be likely to detect an attack that changed only the QR codes, if the attack changed sufficiently many votes to alter the outcome of the contest targeted by the audit. However, Georgia regulations call for an RLA only in the November election of even-numbered years, and only targeting a single, state-wide contest chosen by the Secretary of State [69]. Therefore, such cheating likely would not be detected in the vast majority of elections and contests.

As discussed in Section 5, attackers could also choose to cheat by changing both the QR codes and the human-readable text on a small fraction of ballots, such that both reflected the same fraudulent choices. This would be completely undetectable by an RLA or a hand count. Although, in principle, voters might notice that the printed ballots were wrong, human-subjects research indicates that only a small fraction of voters verify their ballots closely enough to notice such errors [9, 54]. As a result, when vulnerable BMDs are used for all in-person voting, as in Georgia, malware could alter enough votes to change the outcome of a close race while likely triggering too few voter complaints to alert election officials that there was a systemic problem [9].

[15]The function [redacted]. **Redaction**

7.6 Conclusions

I have demonstrated how it is possible to create a malicious version of the ICX App that selectively alters ballot QR codes to steal votes and favor an attacker's preferred candidate.

I have also demonstrated that such malware can take steps to effectively defeat Georgia's procedural defenses. Once installed on an ICX, the proof-of-concept malware I created would not be detected by the state's logic and accuracy testing, hash checking, and APK validation procedures. Even a post-election forensic audit, if conducted using the methodology that Pro V&V applied following the November election, would not detect well designed malware.

Although cheating by malware that changed only ballot QR codes could be detected by a rigorous risk-limiting audit if the malware altered enough votes to change the outcome of the contest targeted by the audit, the vast majority of elections and contests in Georgia (even high-profile ones) are not audited at all. Even in contests that are subject to an RLA, malware that changed both the QR codes and the ballot text could likely avoid detection while changing individual votes and the outcome of a close race.

While I have created a concrete example of BMD malware as a proof-of-concept, numerous variations are possible, both in terms of the technical means by which the malware affected the ICX's operation and the specific effects. Many of these variations could accomplish the same result: stealthily changing Georgia citizens' votes.

8 Installing Malware Locally

An attacker who has access to an ICX BMD has multiple ways to install malicious software, such as the vote-stealing malware described in Section 7. In this section, I describe three separate techniques for accomplishing this that I have successfully tested with the ICX from Fulton County.

These techniques do not require any secret passwords, PINs, or keys, nor does the attacker have to open the device's chassis or break any tamper-evident seals. They only need physical access to the BMD for a few minutes. Attackers could gain such access before machines are delivered from the manufacturer, while they are in storage, while they are being prepared for use in an election, or at the polling place. As I will show, malware could potentially even be installed by regular voters, without any special level of access or technical skill.

8.1 Attaching USB Devices to the ICX

Issue: *The ICX fails to adequately restrict the kinds of devices that can be attached to its USB ports, including the externally exposed USB cable that connects to the printer.*

The malware installation techniques described here involve attaching USB devices to the ICX. The machine has several external USB ports behind plastic doors on the rear of its enclosure. One of the USB ports is used to attach a cable that connects to the printer. They are also used to attach USB drives from which election definition files and occasional software updates are loaded.

Dominion could have designed the ICX to limit the kinds of devices that are allowed to attach to each USB port—i.e., by allowing only specific models of printers to communicate with the port used for the printer, and only specific models of USB drives to connect to the port used for loading data. Instead, all of the exposed USB ports can be used interchangeably, and there do not appear to be any technical restrictions on the devices that may be connected.

I understand that Georgia requires the USB port doors to be closed and secured with tamper-evident seals while the machine is in use at a polling place. The kinds of tamper-evident seals typically used in election systems are known to be easily bypassed using commonly available tools [7]. However, this is unnecessary for the attacks described here, because the seals present no practical obstacle to connecting new USB devices.

Figure 9 shows how the ICX is deployed in polling places in Fulton County and other Georgia localities. The USB cables that attach two printers to a pair of BMDs are visible. Observe that the ends of the cables that attach to the back of the printers are not sealed to the printers. It would be possible for voters to reach behind the printers and disconnect the cables without leaving physical evidence.

Using an inexpensive adapter, a USB drive or other device can be attached to the end of the cable, and it will function as if it was plugged in directly to the BMD's USB port. An example of this arrangement is shown in Figure 10.

REDACTED VERSION

Figure 9: **ICX USB Interfaces are Exposed to Voters and Unsealed.** A USB cable connects the BMD to an off-the-shelf laser printer. At polling places, the end of the cable attached to the printer is physically accessible to voters, and it is not protected by a tamper-evident seal. Voters could install malware on the ICX by attaching a device to the end of this cable. *Photograph taken by Harri Hursti during polling place observation in Fulton County, November 3, 2020.*

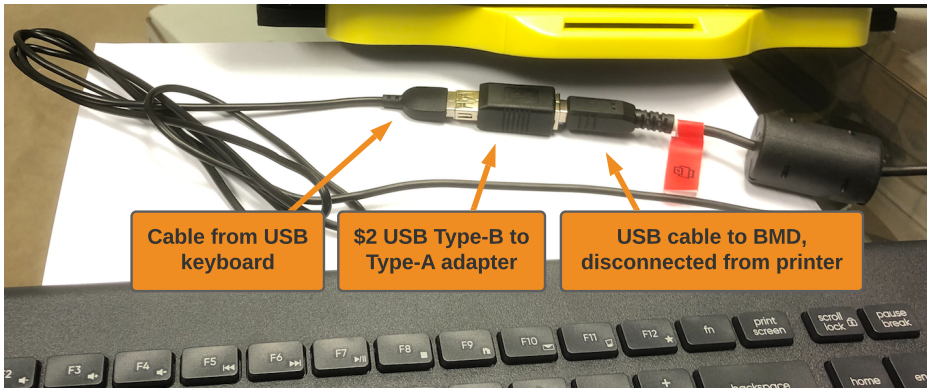


Figure 10: **Attaching a USB Device to the ICX via the Printer Cable.** The BMD's USB cable is not sealed to the printer, and voters can simply reach behind the printer and disconnect it. Using an inexpensive and widely available adapter, any standard USB device (such as the keyboard shown here) can attach to the end of the cable and operate as if it were plugged in directly to the ICX.

8.2 "Escaping" the ICX App

Issue: *As a result of Georgia's installation of a software update in October 2020, the ICX's Android operating system settings can be accessed by attaching a USB keyboard, allowing the installation of malware.*

In October 2020, shortly before the start of early voting in the November election, Georgia installed a purportedly *de minimis* software update on its BMDs to correct a user-interface glitch. In support of Plaintiffs' opposition to this change, I testified that "in complex computerized systems like Georgia's election equipment, last-minute changes, even seemingly small ones, can introduce serious and difficult-to-foresee consequences" [41, ¶ 5]. I drew an analogy to the Boeing 737 MAX aircraft, where a small, last-minute change to correct a single problem inadvertently created a much more dangerous failure mode that reportedly led to two fatal crashes [56].

My testing shows that installing the ICX software update did indeed create a dangerous security problem. It left the BMDs in a state where anyone with physical access, including non-technical voters, could install malicious software.

The problem relates to the method by which the Android operating system on the ICX is "locked down". When Android is used on consumer devices like phones and tablets, users can open any installed app, switch between apps, and access the Android Settings app, which allows the installation of new software and changes to security-critical settings. If Android is used for building special-purpose devices that serve the public (often referred to as "kiosks"), the manufacturer needs to take steps to restrict access to these functions, usually by preventing unauthorized users from leaving a particular app that provides the device's user interface.

Recent versions of Android provide a "dedicated devices" programming interfaces that device makers can use to securely lock down the operating system [4]. However, instead of using such an API, the ICX takes an *ad hoc* approach. It sets the ICX App as the system's "launcher", i.e., the app that provides the user interface for the device's "home screen" or "desktop" [18]. This ensures that the ICX App is automatically started when the device powers on, and it prevents users from directly launching other apps via the normal launcher interface.

This approach has dangerous limitations. Making the ICX App the launcher does not block users from *switching* to other apps. One way in which users can still switch to other apps is by attaching a USB keyboard and pressing the Alt+Tab key combination, which cycles through apps in the Android Overview screen[16] [1]. This keyboard shortcut does not allow the user to switch to any app *installed* on the device, but rather only to an app that has previously been started. In the version of Android installed on the ICX, apps are added to the Overview screen whenever they are used, and they remain accessible via Alt+Tab even after the device is rebooted, unless they are explicitly removed through the Overview interface [5, 90].

There would not be a problem if other apps had not previously been used, or if they had been properly removed. However, crucially, Dominion's 40-step process for installing the ICX software update (Exhibit A) used two sensitive apps, File Manager and Settings, and neglected to remove them from the Overview screen. This means these apps are accessible through the use of a keyboard on any BMD where the software was updated according to Dominion's instructions. It is a reasonable inference that the instructions were not subjected to rigorous security testing before use, since the update was installed on BMDs across Georgia only days after being created.

To prevent these apps from being accessible, it would have been necessary to perform a process like this after installing the update:

1. Use the "Toggle" button at the bottom of the screen to enable the Android navigation controls, confirm the change, and click OK to reboot the ICX.
2. Launch Android Settings from the Technical Administration menu.
3. Press the "Toggle" button again and press OK to confirm, but do *not* immediately reboot.
4. Press the square App Overview button at the bottom of the screen.
5. Swipe right on the pictures of every previously opened app to remove them from the Overview screen. Once all are closed, the ICX App will reappear.
6. Power off the device.

This would prevent the Alt+Tab vulnerability, but Dominion's instructions included no such steps. Instead, testing shows that after completing the update, the Settings and File Manager apps remain perpetually available through use of a keyboard, even after the device is powered off and on again multiple times. As I describe below, attackers can exploit ICXs in this vulnerable state to install malicious software.

[16] This key combination switches between previously started apps, just like Alt+Tab on Windows and Command+Tab on macOS switch between open windows.

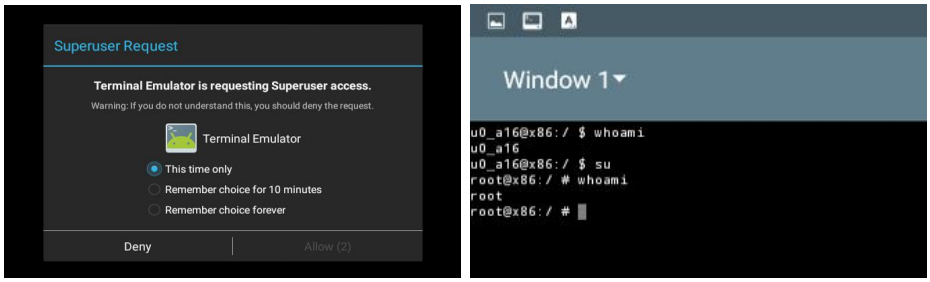


Figure 11: The ICX has a pre-installed **Terminal Emulator app** that provides access to a Linux command-line interface. Simply by confirming an on-screen prompt (*left*), the user can obtain "root" access (*right*), allowing subsequent commands to bypass Android's access control and privilege separation defenses.

8.3 Accessing a Root Shell via the Built-In Terminal App

Issue: *The ICX has a built-in Terminal Emulator app that is configured so that the user can easily obtain a command-line shell with supervisory privileges.*

After escaping kiosk mode, an attacker can easily launch any app installed on the ICX. The machine contains 20 pre-installed apps, most of which appear unnecessary for its use as a BMD. Most notably, there is a Terminal Emulator that provides access to a Linux shell, a powerful text-based user interface.

Moreover, the ICX is configured such that the Terminal Emulator user can easily obtain supervisory ("root") access privileges by simply selecting "Allow" at an on-screen prompt, shown in Figure 11. With root privileges, terminal commands can completely bypass the Android operating system's access control restrictions and make arbitrary changes to the device's data and software.

The Terminal Emulator made analysis of the device much more efficient, since I was able to easily access, control, and modify any part of the data or software. It also makes it easy for an attacker to install programs or run automated commands for malicious purposes.

8.4 Manual Malware Installation Process

I will now walk through a process that exploits the vulnerable state of the BMD to install malware. This involves the attacker attaching a USB keyboard and then a USB thumb drive to the machine, as described in Section 8.1. These manual steps are relatively cumbersome and time-consuming, and they would be impractical to carry out in the polling place, but I describe them here for expository purposes. I will later show how the entire process can be automated, so that the attacker need only briefly attach a single USB device.

Preparing for Installation The attacker attaches a standard USB keyboard to the BMD, by attaching it to the end of the exposed printer cable through a USB adapter, as shown in Figure 10. By pressing Alt+Tab, the attacker switches

from the ICX App to the Android Settings app, from which they can access the Terminal Emulator. From there, the attacker escalates to root privileges by typing the `su` command and confirming the on-screen prompt shown in Figure 11. They then use Linux shell commands to copy the APK of the installed ICX App to a temporary location, so that the malware can export it for verification, and to copy any election definition files stored by the ICX App to a location where the malware can access them.

Installing Malicious App Next, the attacker returns to Android Setting, then disconnects the USB keyboard and connects a USB drive containing the installation file for a malicious modified ICX App like the one created in Section 7. Using Android Settings, the attacker uninstalls the original ICX App, enables a configuration setting to "[a]llow installation of apps from unknown sources", installs the malicious ICX App from the USB drive, then disables the configuration setting, all in a manner similar to Dominion's official software update instructions.

Post-Installation Clean Up Finally, the attacker reattaches the USB keyboard and uses the terminal to clean up traces from the installation process and to restore the previously-saved election definition files to their original location in the now-malicious ICX App's storage hierarchy.[17] The attacker then disconnects the USB keyboard and launches the malicious app. The BMD appears to function normally, but the maliciously modified software can tamper with printed ballots to steal votes.

8.5 Automating Malware Installation

The process described above can be completely automated, so that an attacker can install malware by attaching a single USB device to the exposed printer cable for less than two minutes. The automated process is simple and fast enough that it could potentially be carried out by a voter in the polling place.

To automate the attack, I used a device called a "Bash Bunny", which is commercially available for less than $100 [37]. A widely-used tool for penetration testing, the Bash Bunny (shown in my hand in Figure 12) looks similar to a typical USB thumb drive, but it acts simultaneously as a USB storage device and a simulated keyboard. Once attached to a target machine, it sends a pre-programmed sequence of keystrokes to execute the attacker's objectives. I prepared the Bash Bunny by copying the malicious APK to its USB storage and programming it to send keystrokes that carry out the installation process, following a sequence of operations similar to those in Section 8.4.

Once the Bash Bunny is programmed, launching the attack requires no technical skills. A voter could do so by following simple directions like these:

1. Take a pre-programmed Bash Bunny and a USB adapter to a polling place. Check in normally, then select an out-of-the-way BMD with a screen that is difficult for poll workers to observe.

[17] In fact, the malicious app has the ability to execute commands with root privileges itself, as described in Section 9.3, so these steps could also be executed automatically by the malware once it was installed.

Figure 12: **Installing Malware in the Polling Place.** An attacker can install vote-stealing malware on the ICX by attaching a small USB device for under two minutes. This sequence shows me reaching behind the printer, unplugging the cable that leads to the BMD, and connecting the device to the end of the cable. This can be accomplished in seconds and does not require breaking any tamper-evident seals. It could potentially be carried out surreptitiously by a voter.

2. With one hand, reach behind the printer and unplug the USB cable. Attach the Bash Bunny to the end of the cable (as shown in Figure 12), and leave it out-of-sight behind the printer.
3. Stand in front of the BMD and pretend to vote, carefully blocking the screen. Wait until the process completes (less than two minutes).
4. Discreetly unplug the Bash Bunny and reconnect the cable to the printer.
5. On the BMD screen, tap the icon that says "ImageCast X".
6. Quickly proceed to print your ballot, then scan it like any other voter.

Of course, an attack at a polling place might be more likely to be detected (depending on the circumstances) than an attack conducted in a non-public setting. Similar steps, but with less need for subterfuge, could be used by election workers or outsiders who had brief private access to BMDs.

8.6 Local Malware Installation using a Forged Technician Card

While the attack method demonstrated above exploits the vulnerability created when the October 2020 software update was installed, there are also other means

of installing malware. One is to use a forged Technician Card created using the technique described in Section 6.2, which requires no secret passwords, keys, or PINs, but only a widely available $10 Java Card with some simple programming.

By inserting a forged Technician Card like the one I created, the attacker can access the Technical Administration menu, exit the ICX App, and then proceed to install malware using essentially the same on-screen process that is used to install official software updates. As before, a Bash Bunny could be programmed to automate the necessary steps, so that malware installation could be performed quickly by anyone with brief physical access to an ICX.

8.7 Local Malware Installation via Android Safe Mode

Issue: *A local user can reboot the ICX into "Safe Mode", allowing full control of the Android operating system.*

A third method for installing malware is to exploit a publicly known security flaw in the ICX. According to a Dominion customer advisory dated January 2020, "[i]f the mechanical power button (behind the ICX door) is pressed a power down option is presented. At this point, if the power down screen button is pressed and held, the 'safe mode' option is presented" [22].

I tested this behavior on the ICX. As shown in Figure 13, holding the power button and selecting "Reboot to safe mode" will cause the BMD to restart with the standard Android Launcher available, providing unrestricted control of the device, including access to the Android Settings, File Manager, and Terminal Emulator apps and the ability to install or remove software.

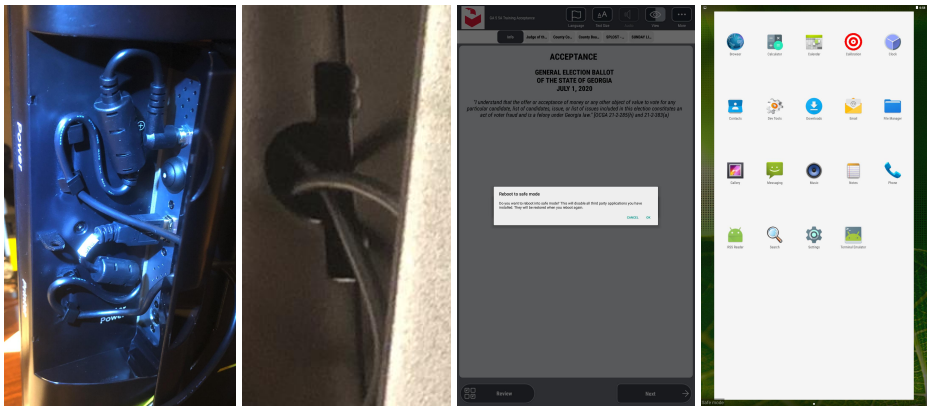


Figure 13: **Rebooting the ICX into Safe Mode.** *Left to right:* (1) The ICX power button is located behind a plastic door on the side of the machine; (2) Even with the door closed, an opening for cables allows the button to be pressed by inserting a metal tool; (3) Holding the button causes the machine to show a "Reboot to safe mode" prompt; (4) When the machine reboots, the user has unrestricted access to Android, including the ability to install malware.

"Safe mode" is an Android feature that is intended for use in recovering from software issues [36]. If the ICX used a more recent version of Android, Dominion could have easily disabled it with the `DISALLOW_SAFE_BOOT` setting introduced in Android 6.0 in 2015 [6]. Instead, Dominion advises that "[i]t is imperative that safety seals be used on the doors on the back side of the ICX to prevent unauthorized access to the mechanical power button" [22].

Unfortunately, the design of the door that covers the power button makes it difficult to secure. The door contains an opening to allow cables to pass through. By inserting a metal tool through this opening, an attacker can press the power button even with the door fully closed.

Even if both the door and the opening were securely sealed while the BMD was in use *by voters*, election workers need to access the power button so they can turn on the machine, both during pre-election testing and at the polling place. Dishonest election workers (like those emphasized by Defendants and their expert Michael Shamos) or intruders who gained access to the machine during these times could exploit the vulnerability to install malware.

9 Installing Malware Remotely

I have described several methods by which attackers can install malware with only brief physical access to an ICX. Although these are severe vulnerabilities, the ICX is also vulnerable to an even more dangerous method of malware installation. By modifying the election definition files that election workers copy to the BMDs before each election, attackers can spread malware to them remotely, with no physical access to the individual machines. By levering this vulnerability, an attacker who infiltrates a county Election Management System (EMS) can spread malware to every ICX in the county, and infiltrating other systems could allow vote-stealing malware to be spread to all ICXs state-wide.

This attack is somewhat more complex than the ones I have described so far, so I will explain it in stages. I first describe how BMD election definitions are produced and distributed. Then I will describe a critical vulnerability that allows a modified election definition file to run arbitrary code on the ICX. Finally, I will explain how this vulnerability can be exploited to remotely install malware.

9.1 ICX Election Definitions

Prior to each election, the ICX must be configured with the available ballot styles, contests, and choices. This data is created using the Democracy Suite EMS software and packaged into an election definition file that is distributed to the BMDs on USB sticks [19].

Structure and Encryption

My testing shows that ICX election definition files are Zip archives that are encrypted using the AES (a.k.a. Rijndael) algorithm. The filename can vary, but I will refer to it as "`ICX.dat`". The Zip archive contains a SQLite database (`electiondata.db3`) that defines the ballot designs and election-specific settings. It also contains assorted graphic files, audio files, and language translation files that are used for presenting the ballots to voters.

I analyzed county election data from the November 2020 and January 2021 elections produced by State Defendants. The data shows that, under current Georgia practice, all BMDs within a county are loaded with the same `ICX.dat` file, which provides every local ballot design used in the county. Moreover, all scanners and BMDs within each county use the same encryption key and initialization vector (IV) during a particular election. Given access to the county EMS or Election Package, the key and IV can be retrieved from the election project database using the SQL command:

```
SELECT RijndaelKey, RijndaelVector from ElectionEvent;
```

I show in Section 6.1 that the same key and IV can also be extracted from any Poll Worker Card in the county, given brief access to the card and PIN.

After obtaining the key and IV, the ICX election definition file can be decrypted using the following shell command:

```
openssl enc -d -aes-128-cbc -K $(xxd -pu <<< 'RijndaelKey')
    -iv $(xxd -pu <<< 'RijndaelVector') -in ICX.dat -out ICX.zip.
```

Issue: *ICX election definition files are not digitally signed, and they can be modified by anyone with access to a symmetric encryption key that is shared by all scanners and BMDs within each county.*

Dominion could have used digital signatures to secure the ICX election definition files against malicious modification. Instead, there does not appear to be any cryptographic integrity protection, beyond verifying that the decrypted file is a properly formed Zip archive. As a result, anyone with access to the encryption key and IV discussed above can decrypt the `ICX.dat` file, modify it, and re-encrypt it using a command similar to the one shown above. My testing shows that the ICX will accept the modified file as if it were genuine.

Distribution and Points of Attack

In Georgia, each county operates a separate election management system (EMS)—a collection of servers and computers that operate the Dominion Democracy Suite EMS application software [20]. Before each election, Dominion centrally prepares an initial Election Project for each county (the data that defines the contests and candidates on the ballots).[18] The company sends each county its Election Project in the form of an Election Package, a Zip archive that contains the election project database used by the EMS software, ballot PDF files for printing, and individual election definition files to be copied to the ballot scanners and BMDs. At the county, workers import the Election Package into the county EMS. As described in Exhibit B, workers then copy the election definition file from the EMS to one or more USB drives, which they insert into each ICX to load the election definition into the machine's internal storage.

This election definition distribution process introduces two kinds of opportunities for remote malware attacks:

At the county level. An attacker who infiltrates a county's EMS can modify the county's `ICX.dat` file before it is copied to USB drives, and thereby spread malware to all BMDs in the county.[19]

At Dominion. An attacker who infiltrates the facility where Dominion prepares Election Projects could modify the election definitions distributed to all Georgia counties, and thereby spread malware to every ICX used in Georgia.

Such attacks could be automated through the use of further malicious software installed on infiltrated EMS systems. That software would be programmed to detect when a new Election Package was loaded. It would then locate the `ICX.dat` file and modify it using the key and IV from the EMS database. Any BMDs on which the modified election definition was subsequently loaded would become infected.

[18] In March 2020, Eric Coomer testified that a single Dominion employee was preparing the Election Projects for all 159 Georgia counties [82, 65:17-69:22].

[19] Alternatively, an attacker with only access to the USB drives could modify the file before it was loaded into the BMDs, given access to a Poll Worker card and PIN from which to obtain the encryption key.

9.2 Directory Traversal Vulnerability

Issue: *The ICX software contains a critical directory traversal vulnerability that allows a maliciously modified election definition file to overwrite arbitrary files.*

The ICX contains a critical vulnerability in the code that loads election definition files. The problems is a so-called "Zip Slip" vulnerability, a common but severe flaw in software that processes Zip files, which has been observed to be "especially prevalent" in Java-based software such as the ICX App [81].

Zip files, such as the ICX election definitions, can contain a hierarchy of folders and files. The Zip format represents this by storing each file's name together with its directory path. For example, a file `logo.png` in a folder `resources` within would be represented in a Zip file using the name `resources/logo.png`.

Normally, when software extracts a Zip file's contents, it recreates the contents inside a specified target folder. The Zip Slip vulnerability allows a maliciously-crafted Zip file to create or overwrite files in any writable location on the system.

To do so, the attacker changes the path names in the Zip file to begin with "`../`". Secure Zip extraction code will detect and ignore these characters, but software that suffers from the Zip Slip vulnerability will treat them as part of the file's location. Operating systems interpret these special characters not as a literal name but as a reference to the target location's *parent* folder. By repeating these characters multiple times, the attacker can traverse to the root folder and direct the file to be stored in any writable location on the system. For example, a Zip file crafted to contain a file named `../../etc/passwd` that was extracted by a vulnerable application inside the target folder `/root/tmp/` would result in an existing file named `/etc/passwd` being overwritten with the new file's contents (so long as the running process had permission to write to `/etc/passwd`).

The ICX suffers from exactly this problem. When an election definition file is loaded, the system decrypts it to a Zip file and extracts the contents in a specific storage location. However, the ICX fails to check whether the file names contain parent folder references. As a result, an attacker can create a modified election definition file that will create or overwrite files in any location on the device that is writable by the ICX App. As I explain below, an attacker can leverage this capability to execute arbitrary code and install malware.

9.3 Arbitrary Code Execution as Root

Issue: *The BMD runs code with root privileges from a file that is writable by the ICX App. When combined with the directory-traversal vulnerability, this allows a malicious election definition file to execute arbitrary code as root.*

The Android OS employs access control and privilege separation to limit what files an app can modify. These defenses normally prevent an app from accessing another app's data, changing its own APK, or installing a new app. However, I find that weaknesses in the ICX software allow attackers to circumvent these defenses. A malicious election definition file can cause attacker-supplied code to

be executed with "root" privileges—complete control of the device's software, including the ability to override all file access restrictions and install malware.

The ICX App contains a native-code executable file named [redacted] Redaction that is delivered as part of the APK. The app does not run this file directly, but rather it makes a copy in the folder [redacted] Redaction, for which the app has write permission. Each time the app starts, it checks whether the file is already at that location, and, if not, it extracts it from the APK and places it there.

The ICX Android distribution includes a vendor-specific system service called [redacted] Redaction that it uses to control various hardware functions. This service uses a dangerous and insecure design that allows the ICX App to execute arbitrary commands with root privileges. Every time the ICX App starts, it uses [redacted] Redaction to run [redacted] as root.

An attacker can exploit these behaviors in combination with the directory traversal vulnerability to create an election definition file containing malicious code that will be executed with root privileges. The attacker merely has to modify the `ICX.dat` file so that, when the Zip archive is extracted, it overwrites [redacted] Redaction with the malicious code. The next time the BMD is powered on, the ICX App will use [redacted] Redaction to run the file with root privileges, giving the attacker's code full control of the device.

9.4 Installing Malware from the Election Definition File

Given the ability to execute arbitrary code as root, the last step to remotely installing malware is replacing the ICX App's code with a maliciously modified version, which can be constructed as described in Section 7. An attacker could replace the app's code by several means; I demonstrate one particularly efficient method that is a variation of a technique presented at the Black Hat Asia conference in 2015 by Paul Sabanal [70].

The ICX App, like most Android apps, is written in the Java programming language. Prior to distribution, the Java source code is compiled into a Dalvik Executable (DEX) file that is combined with other resources to create a self-contained APK file. When the APK is installed on the machine, Android performs a process called ahead-of-time compilation to generate code that is optimized for execution efficiently on the device's hardware. This involves translating the Java bytecode in the DEX file into native code for the machine's processor, which gets stored as what is called an OAT file [2]. When the ICX App runs, it is the translated code in the OAT file that actually gets executed, *not* the original code from the APK.

Sabanal's technique is to replace the OAT file with one containing malicious code, rather than the more obvious approach of replacing the DEX file. In addition to other technical advantages that I discuss below, this avoids introducing a potentially noticeable delay caused by the ahead-of-time compilation process. Though I did not attend Sabanal's original presentation, I found that his publicly available slides were effectively a walk-through of how to perform the

technique [71]. To streamline the process, rather than directly using the `dex2oat` tool to create the malicious OAT file, I simply installed my malicious APK and copied out the OAT file that was generated by the Android installation process. I then modified the OAT file to reflect the DEX path and checksum expected by the operating system, as described in Sabanal's presentation.

Putting the Pieces Together I created proof-of-concept malware that installs automatically when a surreptitiously altered election definition is loaded into the BMD. The key steps are described below:

I started by decrypting the original election definition file and then modified the internal Zip archive to add two new files:

- A maliciously modified version of the ICX App, in the form of an OAT file.
- A shell script (a simple program) that, when run on the BMD with root privileges, overwrites the OAT file for the installed ICX App with the malicious OAT file extracted from the Zip archive, then restarts the ICX App.

I added the shell script to the Zip archive in a way that exploits the directory traversal vulnerability, so that, when the BMD extracts the election definition file, the existing [redacted] program is replaced with the shell script. Finally, I encrypted the modified Zip file with the original encryption key. (These steps are performed automatically by a shell script I created.)

Redaction

The result is a malicious election definition file that appears to behave identically to the original election definition when loaded onto a BMD by an unwitting election worker. However, the next time the BMD is powered on, the shell script runs and invisibly replaces the ICX App's logic with malicious code.

9.5 Defeating Security Precautions More Easily

Like locally installed malware, remotely installed malware could use the mechanisms described in Section 7 to defeat Georgia's logic and accuracy testing (LAT), hash verification, and external APK validation. However, an advantage to the infection technique described here is that can intrinsically bypass these protections with no additional effort on the attacker's part.

Defeating Logic and Accuracy Testing Georgia's pre-election testing procedures (Exhibit B) specify that an election worker should:

1. Insert a USB stick containing the election definition file.
2. Use a Technician Card to copy the file to the BMD.
3. Use a Poll Worker Card to open polls using the election definition.
4. Vote and print at least one test ballot from the BMD.
5. Use a Poll Worker Card to close polls and power off the BMD.
6. Seal the BMD for delivery to the polling place.

When the election definition is loaded from the USB stick in step 2, the file is merely copied to the BMD's storage. Its contents are extracted during step 3,

when the encryption key from the Poll Worker card is provided. This sets the stage for the malware to be installed when the BMD is next powered on, at the polling place. Note, however, that LAT is performed immediately, without restarting the BMD first. This means that testing for the current election will be finished before the malware is activated, so no LAT-circumvention logic is required.[20]

Defeating Hash Validation and External APK Validation The hash value that the ICX App displays is computed by the app itself by hashing its installed APK file, which is stored within the Android filesystem. However, the malware installation technique described here overwrites the dynamically generated OAT file and leaves the original APK intact. As a result, the hash reported by the app does not change, even though the running logic has been maliciously altered. Similarly, when the ICX App exports its APK for external verification, it copies the same locally-stored APK to a USB drive. Since the remotely installed malware does not change the APK, the exported file will contain no evidence of infection.

9.6 Conclusions

I have identified critical vulnerabilities in the ICX software that enable an attacker to remotely execute arbitrary code on the device. These vulnerabilities can be exploited by maliciously altering the election definition files that workers copy to all ICXs before every election.

Security experts consider arbitrary code execution to be one of the most dangerous classes of vulnerabilities, particularly when it can be exploited to run code with root privileges, as it can on the ICX. In 2006, Harri Hursti discovered a similar arbitrary code execution vulnerability that affected Georgia's old AccuVote TS-X DREs [45]. At the time, Defendants' expert Michael Shamos called it "the most serious security breach that's ever been discovered in a voting system" [44]. The vulnerabilities in the ICX are as or more severe.

Using these vulnerabilities, I developed functional proof-of-concept malware that automatically and invisibly installs itself on any ICX on which an infected election definition file is loaded, then manipulates voters' printed ballots to steal votes. By compromising election definition files in this way, an attacker with access to a county's EMS could spread malware to all ICXs in the county, and an attacker who infiltrated the systems that Dominion uses to prepare initial election projects for all Georgia counties could spread vote-stealing malware to every ICX used in Georgia. As I discussed in Section 3.2, even the ICX's use of a paper trail poses no obstacle to vote-stealing attacks in the vast majority of elections and contests, and malware can also evade Georgia's other technical and procedural defenses.

[20] Of course, an attacker might aim to create malware that would cheat in *future elections* too. In that case, the methods in Section 7 could still be used to defeat future rounds of LAT.

10 Manipulating Logs and Protective Counters

Two additional protections that the ICX maintains are an "audit log" of events before, during, and after the election, and a "lifetime counter" of the number of ballots printed. Both can be easily defeated.

Public and Lifetime Counters The ICX App uses two counters to track the number of ballots it prints: the "public counter" and the "lifetime counter". These values are used by election workers to confirm that all ballots are accounted for and that the counts match between the BMDs and scanners.

The public counter is displayed on the Poll Administration Menu and at all times in the lower-left corner of the screen, where it is readily visible to voters. By design, the public counter can be reset by election workers when the poll is closed by using the "Reset" button in the Poll Administration Menu.

The lifetime counter is designed to be a tally of all ballots printed by the machine since its manufacture. It is only displayed on the Poll Administration Menu, and the software does not provide a way to reset it.

Audit Logs The ICX audit logs record timestamped entries related to important events, such as opening or closing the poll, Poll Worker or Technician log-ins, attaching or detaching USB storage devices, software errors, etc. Although the time at which a ballot was displayed to a voter is recorded, the audit log does not contain information about the voters' selections. From the Technical Administration menu, the audit log can be viewed on-screen or exported to a USB drive for review by pressing the "Export Audit Log" button.

10.1 Vulnerable Storage Design

Issue: *ICX audit logs and protective counters are stored in regular files with no protection beyond filesystem permissions, which can be easily bypassed.*

Issue: *The ICX does not provide any mechanism to verify the integrity of exported audit logs.*

Internally, the audit logs and counters are simply files stored in the device's Android filesystem. Reverse-engineering of the ICX App shows that they are stored at these locations:

Redaction – Audit logs: [redacted]
Redaction – Public counter: [redacted]
Redaction – Lifetime counter: [redacted]

Access to these files is controlled using filesystem permissions, but the data they contain does not appear to be protected by any kind of encryption or cryptographic integrity mechanism, such as a MAC or a digital signature. Nor are the audit logs cryptographically protected when exported to a USB device.

To advance the counters, the ICX App simply reads, increments, and overwrites the values in the files. Similarly, to make a log entry, it simply appends

to the current day's log file. When exporting logs, the app merely packages the existing audit log files into a Zip file and writes it to the USB drive.

This leaves the counters and logs highly vulnerable to modification. As I explained in previous sections, weaknesses in the ICX allow attackers to easily gain root privileges, which lets them bypass all filesystem access controls. Consequently, attackers can arbitrarily edit or erase the audit logs, and they can change the protective counters to any value they choose.

10.2 Manual and Automated Modification

An attacker with physical access to the BMD can manipulate the logs and counters via several routes. First, they need to escape from the ICX App, using any of the methods described in Section 8. After accessing the underlying Android operating system, it is a simple matter to locate the applicable file and change its contents to suit the attacker's purposes. Attackers can do this either by installing malicious software that modifies the files automatically or by manually editing them using Android apps that are pre-installed on the ICX.

To give a concrete example, suppose the attacker wants to manipulate the ICX access log. They can hold the power button on the side of the machine to reboot it in "safe mode" (see Section 8.7), then open the File Manager app and navigate to the log file location shown above. By tapping on the log file icon, they can open it in the Android Text Editor app and simply use the touch-screen to select and delete arbitrary log entries.

Modifying the protective counters is slightly more involved due to the need to bypass filesystem permission checking. To do so, the attacker can open the pre-installed Terminal Emulator app and (using the on-screen keyboard or a physical keyboard), execute the `su` command to gain root privileges, as described in Section 8.3. They can then write new values to the counter files using any standard command-line method. I confirmed that this technique can successfully "roll back" the lifetime counter to a previous value, allowing the attacker to conceal having printed arbitrarily many ballots.

While I describe manual modification techniques here, malware can also obtain root privileges (see Section 9.3) and can be programmed to modify the logs and counters in an automated fashion. For example, malware could easily be programmed so that, on first run, it removed any log entries associated with its installation. Since modifying the log files would demonstrate no additional security insights beyond those required to install malware in the first place, I did not include such clean-up behavior in the proof-of-concept malware, but it would be a simple matter for a real attacker to do so.

11 Weaknesses in the ICP Scanner

The Dominion ImageCast Precinct (ICP) ballot scanner was not the focus of my investigation, and time constraints precluded conducting a complete security analysis of the device. Nevertheless, I did uncover some security problems related to the ICP, which I report in this section.

11.1 The ICP Accepts Photocopied Ballots

Issue: *The ICP as tested did not require ballots to be printed on security paper, and it accepted ICX ballots photocopied on normal office paper.*

Georgia uses special "security" paper stock for official ballots, including those printed by BMDs [32, 35]. However, when I tested the Fulton County ICP using ballots printed on normal copier paper, it accepted and counted them normally. I also tested scanning photocopies of BMD-printed ballots, and the ICP again accepted and counted them normally.

As Section 3.2 explains, the message authentication codes in the QR codes do not allow the scanners to distinguish between original and duplicate ballots, so, absent a check on the physical paper stock, the scanners cannot detect photocopied ballots.

Use of security paper *is* potentially valuable during a risk-limiting audit or a hand recount. Assuming access to such paper is carefully controlled, ballots printed on non-official paper could be detected during the auditing process. However, I note once again that Georgia requires risk-limiting audits of only once race in November elections of even numbered years, leaving other contests and elections potentially unprotected.

11.2 A Dishonest Poll Worker with Access to the ICP Memory Card can Deanonymize All Voted Ballots

Issue: *The ICP tested does not encrypt ballot images stored on its memory card.*

Issue: *ICP memory cards store ballot images in the order they were cast.*

The ICP stores a complete digital image of every scanned ballot on its removable memory card, and these images are returned to the EMS for possible later review or adjudication. On the Fulton County scanner I tested, the ballot images were not encrypted, and I could easily extract them. Moreover, my testing shows that the unencrypted ballot images are stored in the order in which they were cast, potentially deanonymizing the secret ballots.

Encrypting ballot images appears to be a configuration option that jurisdictions can enable. That option was not enabled in the ICP I tested, which was purportedly configured in the same way as the scanners used during Georgia elections. In any event, even if jurisdictions were to enable this encryption option, the county-wide encryption keys can be be extracted from any ICX Poll Worker Card, given brief access to the card and PIN (see Section 6.1).

I determined the ballot image storage format by examining what data on the memory card changed when I scanned an additional ballot. The ballot images are not stored as regular files in the card's filesystem. Rather, they are stored in a proprietary data structure in a secondary partition. [redacted] Redaction

Following this volume header, the ballot images are stored sequentially. [redacted] Redaction I created a Python program (`cfextract.py`) to extract the ballots images from the memory card, in the order they were voted, and output them as TIFF files.

Storing the ballots in voted-order raises serious risks to ballot secrecy. A dishonest poll worker could observe voters as they used the scanner and secretly note their names, in order. If, after voting was finished, the poll worker had brief access to the scanner memory card, they could read its contents with an inexpensive and widely available Compact Flash card reader, then use a program like mine to view all the ballots and associate each with the voter's identity.

11.3 Installed Tamper-Evident Seal could be Bypassed or Defeated

Issue: *The ICP modem port door is incompletely closed when sealed, allowing access to connectors inside.*

Issue: *The tamper-evident seal on the ICP tested was improperly installed, leaving it easily defeated.*

The Fulton County ICP was delivered to Plaintiffs with only one tamper-evident seal installed. On the right side of the ICP, a plunger-style security seal was affixed to a small plastic door that the ICP User Guide refers to as the "Modem Port" [21, p. 11], which covers an RJ45 Ethernet port and a USB Type-A port. The seal, Intab part number 03-1366 [50], consists of a braided wire that passes through a metal loop in the machine's case, preventing the door from being fully opened. The sealed door, as we received it, is shown in Figure 14a.

One problem with this sealing arrangement is that, by applying tension to the door, it can be opened several millimeters without removing the seal. As shown in the figure, this is sufficient access to see both ports, and an attacker could almost certainly attach electronic equipment to either port by inserting conductive probes through the gap in the door. The problem could have been avoided by using a different kind of seal. Dominion's manual states that "[a] lock, tamper evident label, or tamper evident tie wrap should be placed on the door lock loop" [21, p. 49], but the seal that was installed is a wire seal, which is thinner and more flexible than a typical tie wrap, allowing more play.

Furthermore, the seal was improperly installed and could easily be removed without breaking it. According to the manufacturer's instructions, when installing the seal, the metal plunger needs to be fully depressed into the seal housing. In the condition we received it, the plunger was incompletely inserted, as shown in

REDACTED VERSION

(a) ICP sealed door, as received (b) Defeating the seal

Figure 14: **Defeating the ICP Tamper-Evident Seal.** The ICP scanner from Fulton County used a plunger-style wire seal to guard access to the "Modem Port". Even with the seal installed, an attacker could open the door enough to access the telephone and USB ports inside it (*left*). While the seal's documentation [50] shows that a properly installed seal will have the metal plunger fully inserted (*top*), the seal as installed only had the plunger partially depressed (*middle*). This allowed easy removal of the seal in such a way that it could be reattached without leaving any visible evidence of tampering (*bottom*).

Figure 14b. I watched as my assistant used his bare fingers to grasp the plunger and simply pull it out of the seal's plastic housing. With the plunger removed, he was able to free the wire from the seal housing using a gentle tugging motion, thus removing the seal and allowing the door to fully open.

After inspecting the internals of the seal's housing, I determined that the wire running through the metal hasp had been only slightly bent due to the incomplete insertion of the plunger. This allowed the seal to be removed without damaging any of its components. It would be possible to reaffix it without leaving any obvious signs that it had been breached.

That Fulton County election workers selected an inappropriate seal *and* failed to properly install it—on a scanner they knew would be subjected to security testing—suggests that Georgia security seal practices are insufficient to reliably protect the state's election equipment from undetected physical access.

Expert Qualifications

My name is J. Alex Halderman. I am Professor of Computer Science and Engineering, Director of the Center for Computer Security and Society, and Director of the Software Systems Laboratory at the University of Michigan in Ann Arbor. I hold a Ph.D. (2009), a master's degree (2005), and a bachelor's degree (2003), *summa cum laude*, in computer science, all from Princeton University. My background, qualifications, and professional affiliations are set forth in my *curriculum vitae*, which is available online at https://alexhalderman.com/home/halderman-cv.pdf.

My research focuses on computer security and privacy, with an emphasis on problems that broadly impact society and public policy. Among my areas of research are software security, network security, computer forensics, and election cybersecurity. I have authored more than 90 articles and books, and my work has been cited in more than 12,000 scholarly publications. I have served as a peer-reviewer for more than 35 research conferences and workshops.

I have published numerous peer-reviewed research papers analyzing security problems in electronic voting systems used in U.S. states and in other countries. I have also investigated methods for improving election security, such as efficient techniques for auditing whether computerized election results match paper ballots. I regularly teach courses in computer security, network security, and election cybersecurity at the graduate and undergraduate levels. I am the creator of Securing Digital Democracy, a massive, open, online course about computer security and elections that has attracted more than 20,000 students.

I serve as co-chair of the State of Michigan's Election Security Advisory Commission, by appointment of the Michigan Secretary of State. I have also performed security testing of electronic voting systems for the Secretary of State of California. I have testified before the U.S. Senate Select Committee on Intelligence and before the U.S. House Appropriations Subcommittee on Financial Service and General Government on the subject of cybersecurity and U.S. elections.

I received the John Gideon Award for Election Integrity from the Election Verification Network, the Andrew Carnegie Fellowship, the Alfred P. Sloan Foundation Research Fellowship, the IRTF Applied Networking Research Prize, the Eric Aupperle Innovation Award, the University of Michigan College of Engineering 1938E Award for teaching and scholarship, and the University of Michigan President's Award for National and State Leadership.

Affirmation

I declare under penalty of the perjury laws of the State of Georgia and the United States that the foregoing is true and correct, and that this report was executed this 1st day of July, 2021.

J. Alex Halderman

References

[1] Android Developers. *Android Lollipop*. 2014. URL: https://developer.android.com/about/versions/lollipop.

[2] Android Developers. *Android Runtime (ART) and Dalvik*. URL: https://source.android.com/devices/tech/dalvik.

[3] Android Developers. *Codenames, Tags, and Build Numbers*. URL: https://source.android.com/setup/start/build-numbers#usecase-codename-references-and-resources.

[4] Android Developers. *Dedicated devices overview*. https://developer.android.com/work/dpc/dedicated-devices. July 2020.

[5] Android Developers. *Recents Screen*. URL: https://developer.android.com/guide/components/activities/recents.

[6] Android Developers. *UserManager*. Apr. 2021. URL: https://developer.android.com/reference/android/os/UserManager.

[7] A. W. Appel. "Security seals on voting machines: A case study." In: *ACM Transactions on Information and System Security* 14.2 (Sept. 2011). URL: https://www.cs.princeton.edu/~appel/voting/SealsOnVotingMachines.pdf.

[8] Avalue. *HID-21V-BTX*. URL: https://www.avalue-tech.com/products/Panel-PC/Industrial-Panel-PC/Light-Industrial-Panel-PC/HID-21V-BTX_2758.

[9] Matthew Bernhard, Allison McDonald, Henry Meng, Jensen Hwa, Nakul Bajaj, Kevin Chang, and J. Alex Halderman. "Can Voters Detect Malicious Manipulation of Ballot Marking Devices?" In: *41st IEEE Symposium on Security and Privacy*. May 2020. https://doi.org/10.1109/SP40000.2020.00118.

[10] D. Bowen et al. *Top-to-Bottom Review of Voting Machines Certified for Use in California*. Tech. rep. https://www.sos.ca.gov/elections/ovsta/frequently-requested-information/top-bottom-review/. California Secretary of State, 2007.

[11] Joseph A. Calandrino, Ariel J. Feldman, J. Alex Halderman, David Wagner, Harlan Yu, and William Zeller. *Source Code Review of the Diebold Voting System*. Part of the California Secretary of State's "Top-to-Bottom" Voting Systems Review. July 2007. URL: https://votingsystems.cdn.sos.ca.gov/oversight/ttbr/diebold-source-public-jul29.pdf.

[12] California Secretary of State. *Conditional Approval of Dominion Voting Systems, Inc. Democracy Suite Version 5.10*. Oct. 2019. URL: https://votingsystems.cdn.sos.ca.gov/vendors/dominion/ds510-cert.pdf.

[13] California Secretary of State. *Conditional Approval of Dominion Voting Systems, Inc. Democracy Suite Version 5.2*. Oct. 2017. URL: https://votingsystems.cdn.sos.ca.gov/vendors/dominion/ds52-cert.pdf.

[14] Cyber Ninjas. *Arizona Audit Statement of Work*. Mar. 31, 2021. URL: https://www.clerkofcourt.maricopa.gov/home/showpublisheddocument/2557/637551085573500000.

[15] Cybersecurity & Infrastructure Security Agency. *Supply Chain Compromise*. URL: https://www.cisa.gov/supply-chain-compromise.

[16] Defense Human Resources Activity. *Common Access Card (CAC)*. https://www.cac.mil/common-access-card/.

[17] Matthew S. DePerno et al. *Plaintiff's Collective Response to Defendants' and Non-Party Counties' Motions to Quash and For Protective Orders. Bailey v. Antrim County*, Michigan Circuit Court for the County of Antrim, Case No. 20-9238-CZ. Apr. 9, 2021. URL: https://www.depernolaw.com/uploads/2/7/0/2/27029178/ex_5-10.pdf.

[18] Android Developers. *Dedicated devices cookbook*. https://developer.android.com/work/dpc/dedicated-devices/cookbook. July 2020.

[19] Dominion Voting Systems. *2.02—Democracy Suite System Overview*. STATE-DEFENDANTS-00047612 *et seq.*

[20] Dominion Voting Systems. *Democracy Suite Election Management System*. URL: https://www.dominionvoting.com/democracy-suite-ems/.

[21] Dominion Voting Systems. *Democracy Suite ImageCast Precinct User Guide (Version 5.5-A.GA::59)*. STATE-DEFENDANTS-00047951 *et seq.*

[22] Dominion Voting Systems. *ICX behavior when powering on in "safe mode"*. Customer advisory notice. Jan. 2020. URL: https://www.eac.gov/sites/default/files/2020-09/ICX%20Safe%20ModevFINAL.pdf.

[23] Dominion Voting Systems. *ImageCast Precinct*. URL: https://www.dominionvoting.com/imagecast-precinct/.

[24] Dominion Voting Systems. *ImageCast Remote Brochure*. URL: https://www.votescount.us/Portals/16/New%20voting%20system/ImageCast%20Remote%20Brochure%20FINAL.pdf.

[25] Dominion Voting Systems. *ImageCast X*. URL: https://www.dominionvoting.com/imagecast-x/.

[26] Jeremy Duda and Jim Small. *Arizona Senate hires a "Stop the Steal" advocate to lead 2020 election audit*. Arizona Mirror. Mar. 31, 2021. URL: https://www.azmirror.com/2021/03/31/arizona-senate-hires-a-stop-the-steal-advocate-to-lead-2020-election-audit/.

[27] Jose Esparza. *Report of Review of Dominion Voting Systems Democracy Suite 5.5*. June 2019. URL: https://www.sos.texas.gov/elections/forms/sysexam/dominion-democracy-suite-5.5.pdf.

[28] Jose Esparza. *Report of Review of Dominion Voting Systems Democracy Suite 5.5–A*. Jan. 2020. URL: https://www.sos.texas.gov/elections/forms/sysexam/dominion-d-suite-5.5-a.pdf.

[29] Nicolas Falliere, Liam O Murchu, and Eric Chien. *W32.Stuxnet Dossier*. Feb. 2011. URL: https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf.

[30] P. Faltstrom, F. Ljunggren, and D. van Gulik. *The Base45 Data Encoding*. IETF Internet-Draft. June 14, 2021. URL: https://datatracker.ietf.org/doc/draft-faltstrom-base45/.

[31] Ariel J. Feldman, J. Alex Halderman, and Edward W. Felten. "Security analysis of the Diebold AccuVote-TS voting machine." In: *USENIX/ACCU-*

RATE Electronic Voting Technology Workshop. Aug. 2007. https://www.usenix.org/legacy/events/evt07/tech/full_papers/feldman/feldman.pdf.

[32] Georgia Secretary of State. *Amendment 1 to the Master Solution Purchase and Service Agreement between Dominion Voting Systems, Inc and the Secretary of State of the State of Georgia*. July 2019. URL: https://sos.ga.gov/admin/uploads/Dominion_Contract-_Amendment_1-_Executed.pdf.

[33] Georgia Secretary of State. *Democracy Suite 5.5–A Certification*. Aug. 2019. https://sos.ga.gov/admin/uploads/Dominion_Certification.pdf.

[34] Georgia Secretary of State. *Election Security is our Top Priority*. URL: https://sos.ga.gov/securevoting/.

[35] Georgia Secretary of State. *Master Solution Purchase and Service Agreement by and between Dominion Voting Systems, Inc as Contractor, and Secretary of State of the State of Georgia as State*. July 2019. URL: https://sos.ga.gov/admin/uploads/Contract.zip.

[36] Google. *Find problem apps by rebooting to safe mode on Android*. Android Help. URL: https://support.google.com/android/answer/7665064.

[37] Hak5. *Bash Bunny*. https://shop.hak5.org/products/bash-bunny.

[38] Hak5. *O.MG Cable – * to USB-A*. URL: https://shop.hak5.org/products/o-mg-cable-usb-a.

[39] J. Alex Halderman. *Declaration*. Dkt. 682. Dec. 16, 2019.

[40] J. Alex Halderman. *Declaration*. Dkt. 855. Sept. 1, 2020.

[41] J. Alex Halderman. *Declaration*. Dkt. 923-1. Sept. 29, 2020.

[42] J. Alex Halderman. "Practical Attacks on Real-world E-voting." In: *Real-World Electronic Voting: Design, Analysis and Deployment*. Ed. by Feng Hao and Peter Y. A. Ryan. 2016, pp. 145–171.

[43] Rosalind Helderman. *Arizona's Maricopa County will replace voting equipment, fearful that GOP-backed election review has compromised security*. June 2021. URL: https://www.washingtonpost.com/politics/arizona-maricopa-2020-audit-review/2021/06/28/98da5e64-d863-11eb-9bbb-37c30dcf9363_story.html.

[44] Ian Hoffman. *Scientists Call Diebold Security Flaw 'Worst Ever'*. Inside Bay Area. 2006. URL: https://www.eastbaytimes.com/2006/05/11/scientists-call-diebold-security-flaw-worst-ever-2/.

[45] Harri Hursti. *Critical Security Issues with Diebold TSx*. Black Box Voting. 2006. URL: https://web.archive.org/web/20120623161935/http://www.bbvdocs.org/reports/BBVreportIIunredacted.pdf.

[46] iBotPeaches. *Apktool*. Github. URL: https://github.com/iBotPeaches/Apktool.

[47] *iButton – iButton Devices – One Wire*. URL: https://www.maximintegrated.com/en/products/ibutton-one-wire/ibutton.html.

[48] Robert S. Mueller III. *Report on the Investigation into Russian Interference in the 2016 Presidential Election (Volume I of II)*. United States Department of Justice. Mar. 2019. URL: https://www.justice.gov/storage/report.pdf.

[49] *Indictment, United States v. Netyksho*. No. 1:18-cr-00215-ABJ, D.D.C. July 3, 2018.

[50] Intab. *Combo Seals*. https://www.intab.net/Combo-Seals/productinfo/03-1366/.

[51] izgzhen. *java2smali*. Github. URL: https://github.com/izgzhen/java2smali.

[52] JesusFreke. *smali*. Github. URL: https://github.com/JesusFreke/smali.

[53] Tadayoshi Kohno, Adam Stubblefield, Aviel D. Rubin, and Dan S. Wallach. "Analysis of an Electronic Voting System." In: *IEEE Symposium on Security and Privacy*. 2004, p. 27. URL: https://doi.org/10.1109/SECPRI.2004.1301313.

[54] Philip Kortum, Michael D. Byrne, and Julie Whitmore. "Voter Verification of BMD Ballots Is a Two-Part Question: Can They? Mostly, They Can. Do They? Mostly, They Don't." In: *Election Law Journal* (Dec. 17, 2020). URL: https://www.liebertpub.com/doi/full/10.1089/elj.2020.0632.

[55] Susan Lapsley. *Letter to Waldeep Singh*. July 2020. URL: https://votingsystems.cdn.sos.ca.gov/vendors/dominion/ds510a/ds510a-approval.pdf.

[56] Majority Staff of the Committee on Transportation and Infrastructure. *The Design, Development & Certification of the Boeing 737 MAX*. Sept. 2020. URL: https://transportation.house.gov/download/20200915-final-737-max-report-for-public-release&download=1.

[57] P. McDaniel, M. Blaze, and G. Vigna. *EVEREST: Evaluation and Validation of Election-Related Equipment, Standards and Testing*. Tech. rep. https://www.eac.gov/assets/1/28/EVEREST.pdf. Ohio Secretary of State, 2007.

[58] National Academies of Sciences, Engineering, and Medicine. *Securing the Vote: Protecting American Democracy*. Washington, DC: The National Academies Press, 2018. ISBN: 978-0-309-47647-8. URL: https://www.nap.edu/catalog/25120/securing-the-vote-protecting-american-democracy.

[59] National Institute of Standards and Technology. *Open Ended Vulnerability Testing for Software Independent Voting Systems*. 2007. URL: https://www.nist.gov/system/files/documents/itl/vote/OEVT.pdf.

[60] National Security Agency. *NSA ANT Catalog*. Media leak. URL: https://www.eff.org/files/2014/01/06/20131230-appelbaum-nsa_ant_catalog.pdf.

[61] National Security Agency. *NSA Cybersecurity Advisory: Malicious Actors Abuse Authentication Mechanisms to Access Cloud Resources*. Dec. 2020. URL: https://www.nsa.gov/News-Features/Feature-Stories/Article-View/Article/2451159/nsa-cybersecurity-advisory-malicious-actors-abuse-authentication-mechanisms-to/.

[62] National Security Agency. *Stealthy Techniques Can Crack Some of SIGINT's Hardest Targets*. Media Leak. Original: http://www.spiegel.de/media/media-35669.pdf. Alternative: https://www.aclu.org/foia-document/stealthy-techniques-can-crack-some-sigints-hardest-targets.

[63] Office of the Director of National Intelligence. *Statement by NCSC Director William Evanina: Election Threat Update for the American Public*. Aug. 7, 2020. URL: https://www.dni.gov/index.php/newsroom/press-releases/item/2139-statement-by-ncsc-director-william-evanina-election-threat-update-for-the-american-public.

[64] Pennsylvania Department of State. *Report Concerning the Examination Results of Dominion Voting Systems Democracy Suite 5.5A*. Jan. 2019. URL: https://www.dos.pa.gov/VotingElections/Documents/Voting%20Systems/Dominion%20Democracy%20Suite%205.5-A/Dominion%20Democracy%20Suite%20Final%20Report%20scanned%20with%20signature%20020119.pdf.

[65] Pro V&V. *Dominion Democracy Suite 5.10-A Software Test Report for the State of California*. June 2020. URL: https://votingsystems.cdn.sos.ca.gov/vendors/dominion/ds510a/provv-source.pdf.

[66] Pro V&V. *Dominion Voting Systems D–Suite 5.5–A Voting System: Georgia State Certification Testing*. Aug. 2019. https://sos.ga.gov/admin/uploads/Dominion_Test_Cert_Report.pdf.

[67] Pro V&V. *Test Report for EAC 2005 VVSG Certification Testing: Dominion Voting Systems Democracy Suite (D-Suite) Version 5.5 Voting System*. Aug. 2018. https://www.eac.gov/sites/default/files/voting_system/files/Dominion_Voting_Systems_D-Suite_5.5_Test_Report_Rev_A.pdf.

[68] Russell Ramsland. *Antrim Michigan Forensics Report (v2)*. Dec. 2020. URL: www.13thcircuitcourt.org/DocumentCenter/View/15743/Antrim-Michigan-Forensics-Report-12-13-2020--v2-REDACTED.

[69] Rules and Regulations of the State of Georgia. *Rule 183-1-15-.04. Audit*. URL: https://rules.sos.ga.gov/GAC/183-1-15-.04?urlRedirected=yes&data=admin&lookingfor=183-1-15-.04.

[70] Paul Sabanal. *Hiding Behind Android Runtime (ART)*. Black Hat Asia 2015. URL: https://www.blackhat.com/asia-15/briefings.html#hiding-behind-android-runtime-art.

[71] Paul Sabanal. *Hiding Behind Android Runtime (ART)*. Black Hat Asia 2015. Presentation slides. URL: https://www.blackhat.com/docs/asia-15/materials/asia-15-Sabanal-Hiding-Behind-ART.pdf.

[72] David E. Sanger. *Obama Order Sped Up Wave of Cyberattacks Against Iran*. The New York Times. June 2012. URL: http://www.nytimes.com/2012/06/01/world/middleeast/obama-ordered-wave-of-cyberattacks-against-iran.html.

[73] Scandit AG. *Scandit Barcode Scanner*. Apple App Store. URL: https://apps.apple.com/us/app/scandit-barcode-scanner/id453880584.

[74] SLI Compliance. *Certification Test Report—Modification: Democracy Suite 5.5–A*. Dec. 2018. https://www.eac.gov/sites/default/files/voting_system/files/Dominion_Voting_Systems_D-Suite_5.5-A_Test_Report_v1.1.pdf.

[75] SLI Compliance. *Dominion Democracy Suite 5.10 Security and Telecommunications Test Report*. Aug. 2019. URL: https://votingsystems.cdn.sos.ca.gov/vendors/dominion/dvs510security-report.pdf.

[76] SLI Compliance. *Dominion Democracy Suite 5.10 Voting System Software Test Report for California Secretary of State*. Aug. 2019. URL: https://votingsystems.cdn.sos.ca.gov/vendors/dominion/dvs510software-report.pdf.

[77] SLI Compliance. *Dominion Democracy Suite 5.2 Security and Telecommunications Test Report*. 2017. URL: https://votingsystems.cdn.sos.ca.gov/vendors/dominion/ds52-sectel.pdf.

[78] SLI Compliance. *Dominion Democracy Suite 5.2 Source Code Test Report for California*. 2017. URL: https://votingsystems.cdn.sos.ca.gov/vendors/dominion/ds52-sc.pdf.

[redacted]

[80] Philip B. Stark and Ran Xie. *Testing Cannot Tell Whether Ballot-Marking Devices Alter Election Outcomes*. July 2020. URL: https://arxiv.org/pdf/1908.08144.pdf.

[81] Snyk Security Team. *Zip Slip*. https://res.cloudinary.com/snyk/image/upload/v1528192501/zip-slip-vulnerability/technical-whitepaper.pdf. June 2018.

[82] *Transcript of Status Conference Proceeding*. Dkt. 722. Mar. 6, 2020.

[83] Greg Ungerer. *[uClinux-dev] [ANNOUNCE] uClinux-dist-20070130 release*. Google Web Cache. Feb. 2007. URL: http://mailman.uclinux.org/pipermail/uclinux-dev/2007-February/041846.html.

[84] United States Environmental Protection Agency. *Volkswagen Violations*. URL: https://www.epa.gov/vw/learn-about-volkswagen-violations.

[85] U.S. Election Assistance Commission. *Democracy Suite 5.5*. Sept. 2018. https://www.eac.gov/voting-equipment/democracy-suite-55.

[86] U.S. Election Assistance Commission. *Voluntary Voting System Guidelines 1.0*. 2005. URL: https://www.eac.gov/sites/default/files/eac_assets/1/28/VVSG.1.0_Volume_1.PDF.

[87] Verified Voting. *The Verifier*. URL: https://verifiedvoting.org/verifier/.

[88] Verified Voting. *The Verifier: Dominion Voting Systems (November 2020)*. URL: https://verifiedvoting.org/verifier/#mode/search/year/2020/make/Dominion%20Voting%20Systems.

[89] Verified Voting. *The Verifier: ImageCast X (November 2020)*. URL: https://verifiedvoting.org/verifier/#mode/search/year/2020/model/ImageCast%20X.

[90] Ryan Whitwam. *Lollipop Feature Spotlight: The Recent Apps List Now Persists Through Reboot*. Android Police. Oct. 17, 2014. URL: https://www.androidpolice.com/2014/10/17/lollipop-feature-spotlight-the-recent-apps-list-now-persists-through-reboot/.

[91] Paul Woolverton. *Fact check: QR codes on Georgia ballots record votes as cast*. USA Today. Jan. 2021. URL: https://www.usatoday.com/story/news/factcheck/2021/01/02/fact-check-qr-code-georgia-us-senate-votes-warnock-loeffler-perdue-ossoff/4115918001/.

[92] Kim Zetter. *An Unprecedented Look at Stuxnet, the World's First Digital Weapon*. Wired. Nov. 3, 2014. URL: https://www.wired.com/2014/11/countdown-to-zero-day-stuxnet/.

REDACTED VERSION

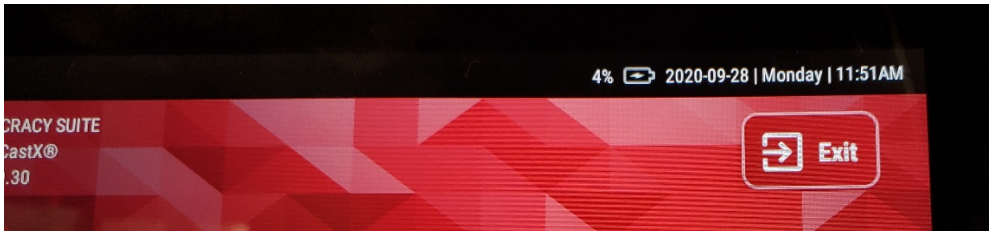
Exhibit A: October 2020 Software Update Instructions

1201 18TH Street, Suite 210
DENVER, CO, 80202
1.866.654.8683
www.dominionvoting.com

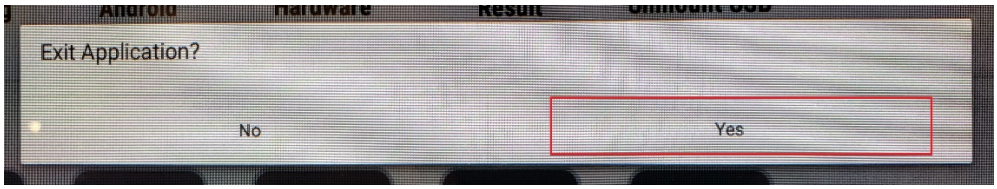
Software Installation for ICX:

1. Press the power button to turn on the ICX.

2. Insert tech card, input password, then click login.
3. Click exit in the top right corner of the screen.

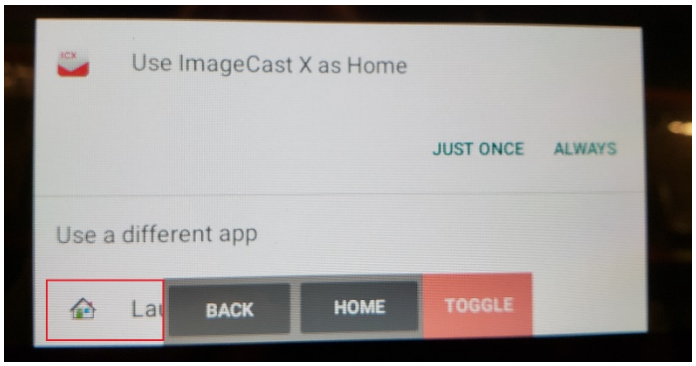


4. Click Yes when asked if you would like to exit the application.

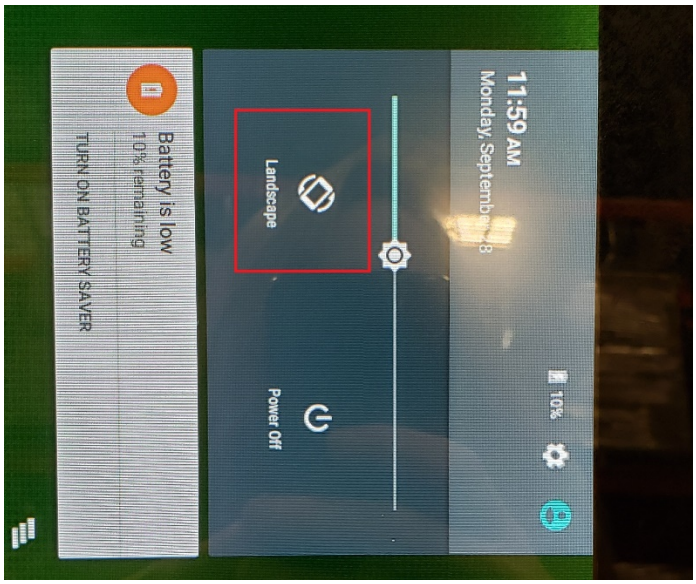


1201 18TH Street, Suite 210
DENVER, CO, 80202
1.866.654.8683
www.dominionvoting.com

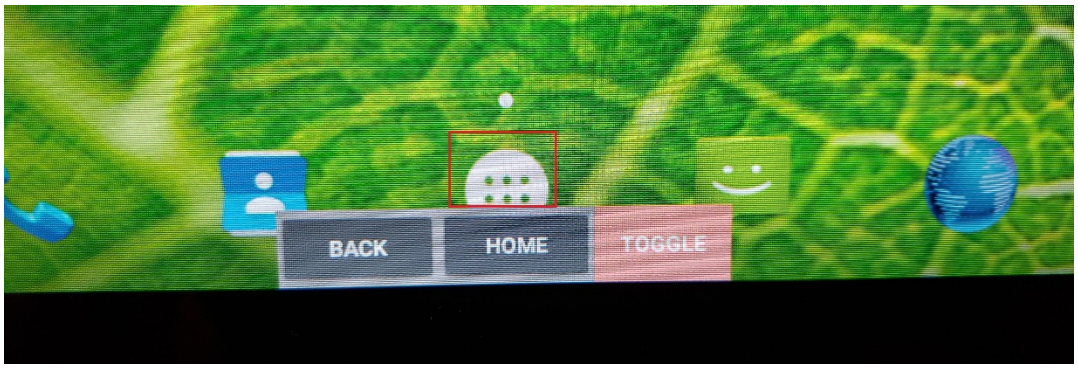
5. Click Launch *note: the launch button may be partially hidden by the back button.



6. Swipe left from the right side of the screen and click the rotate button 3 times. Swipe up when finished.

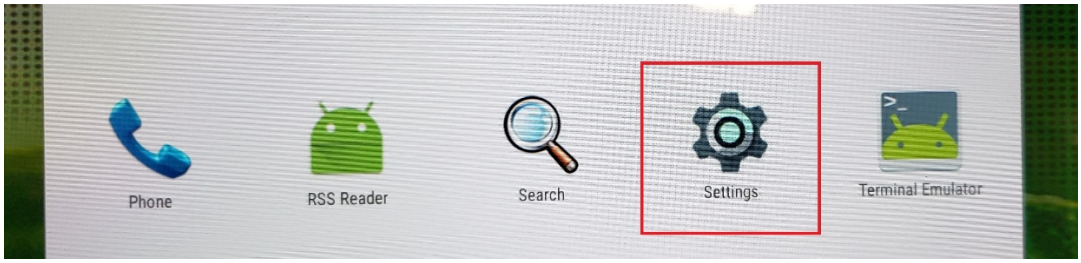


7. Click App Button

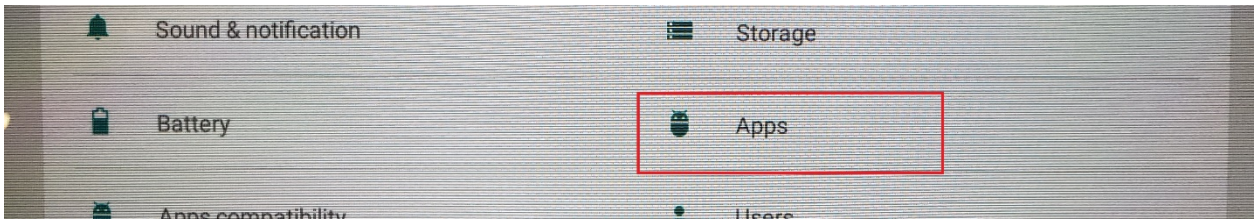


1201 18TH Street, Suite 210
DENVER, CO, 80202
1.866.654.8683
www.dominionvoting.com

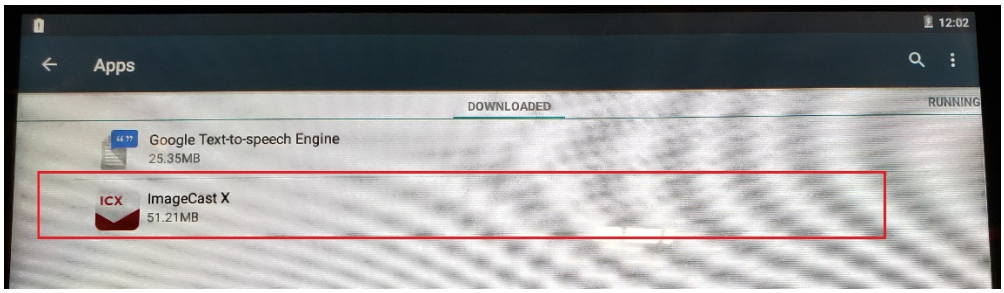
8. Click Settings



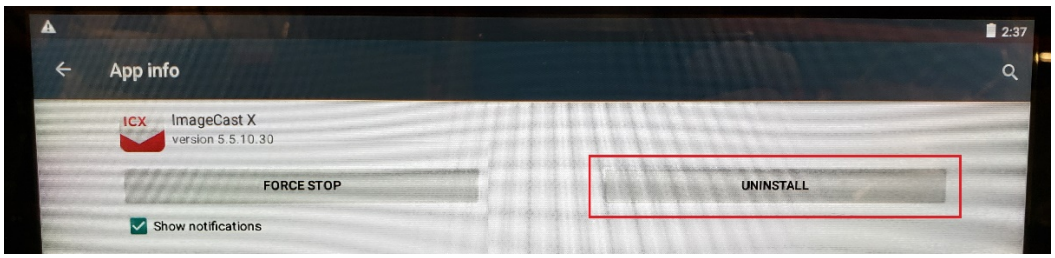
9. Click Apps



10. Click ImageCast X



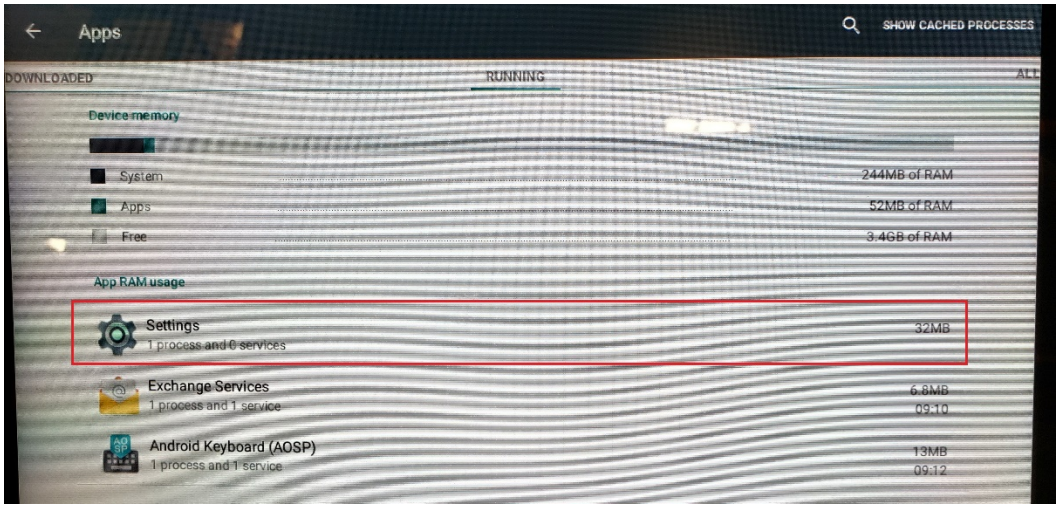
11. Click Uninstall



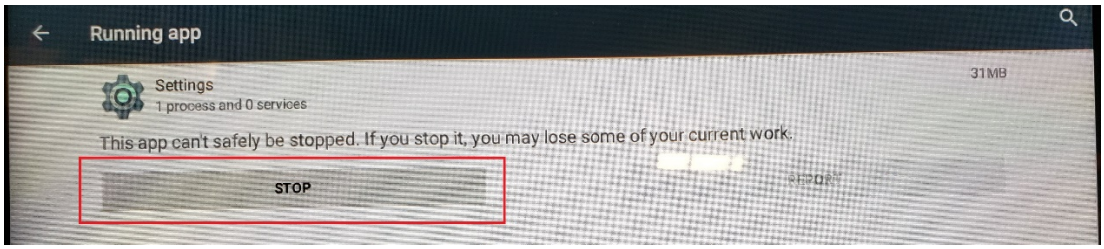
1201 18TH Street, Suite 210
DENVER, CO, 80202
1.866.654.8683
www.dominionvoting.com

12. Click Ok

13. Swipe left to 'Running'. Click Settings

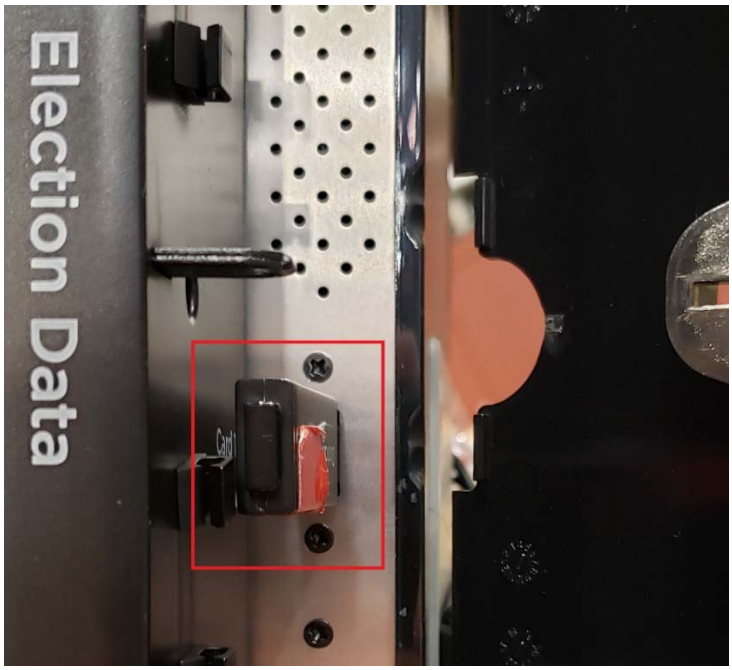


14. Click stop

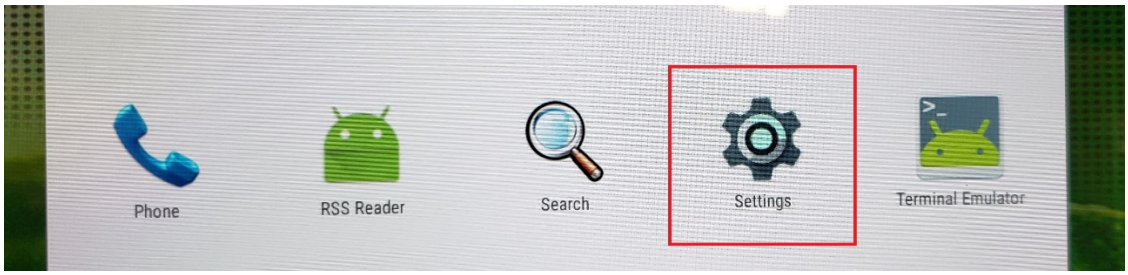


1201 18TH Street, Suite 210
DENVER, CO, 80202
1.866.654.8683
www.dominionvoting.com

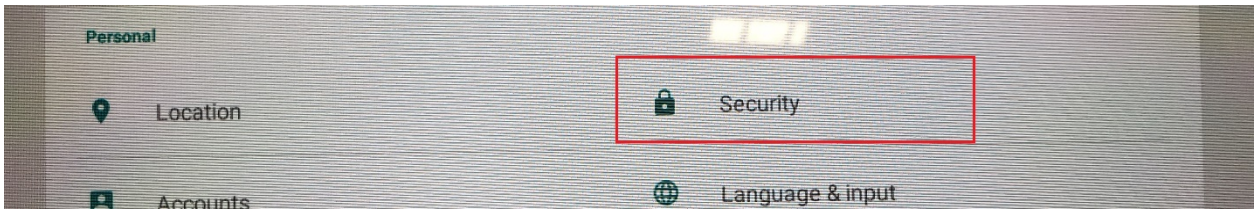
15. Insert USB



16. Click Settings

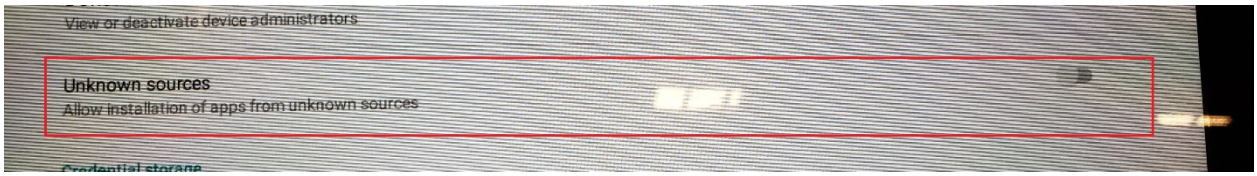


17. Click Security

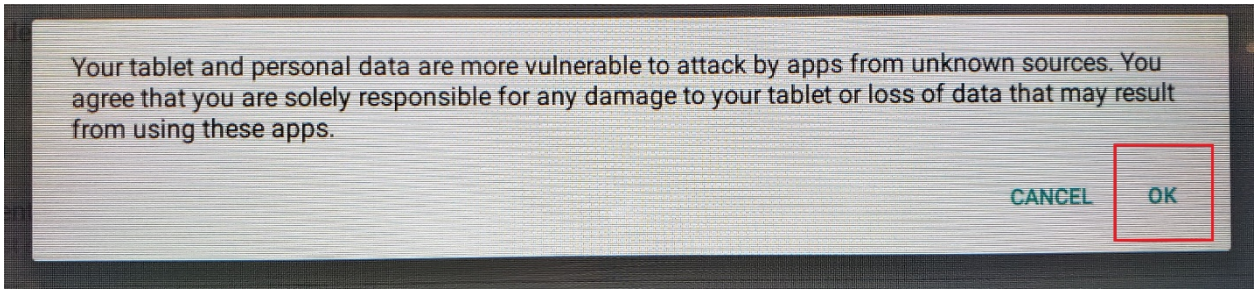


1201 18TH Street, Suite 210
DENVER, CO, 80202
1.866.654.8683
www.dominionvoting.com

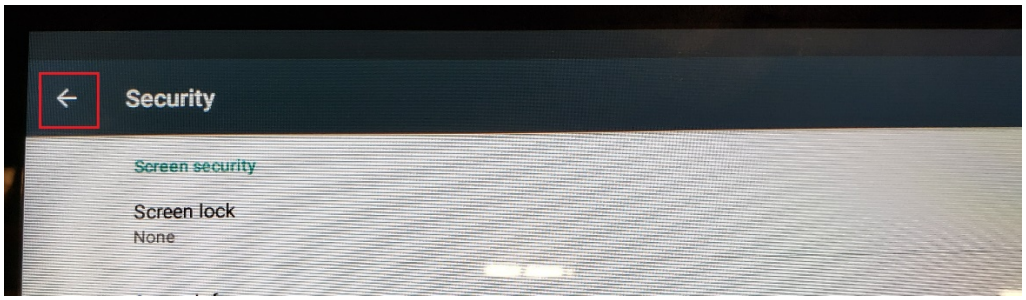
18. Toggle on Unknown Sources *Note: The toggle should be blue after toggling



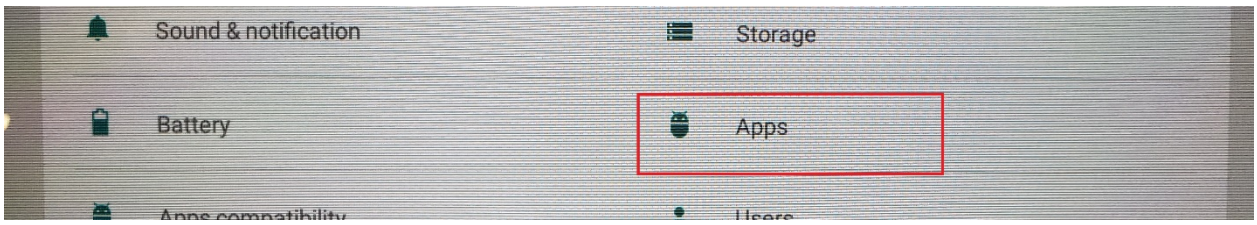
19. Click Ok



20. Click the back arrow in the top left corner.

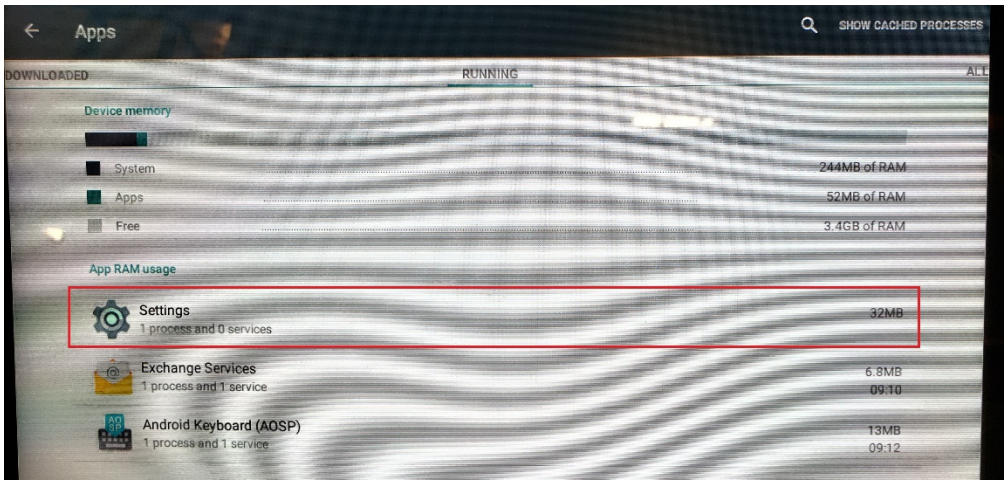


21. Click Apps

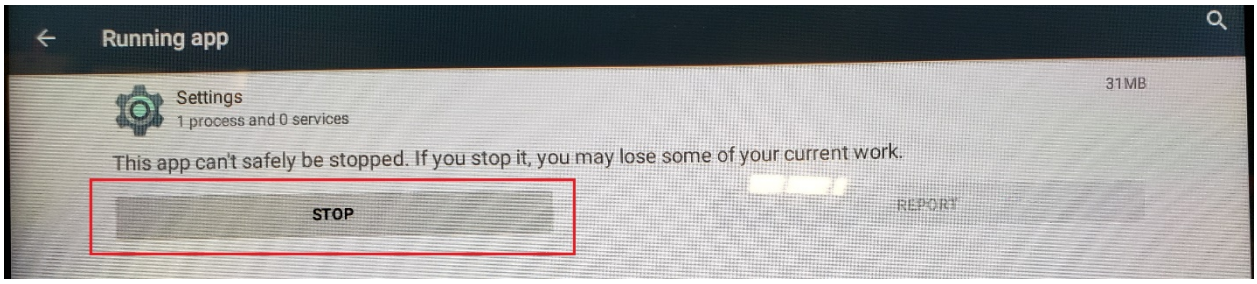


1201 18TH Street, Suite 210
DENVER, CO, 80202
1.866.654.8683
www.dominionvoting.com

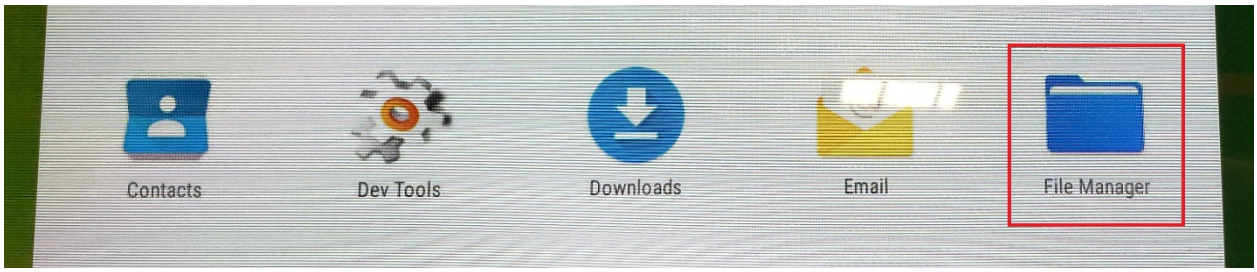
22. Swipe left to 'Running'. Click Settings



23. Click Stop

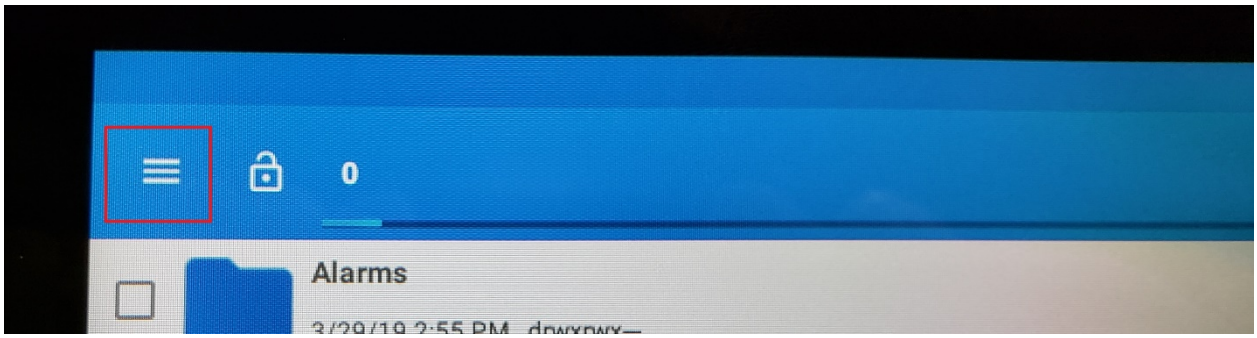


24. Click File Manager

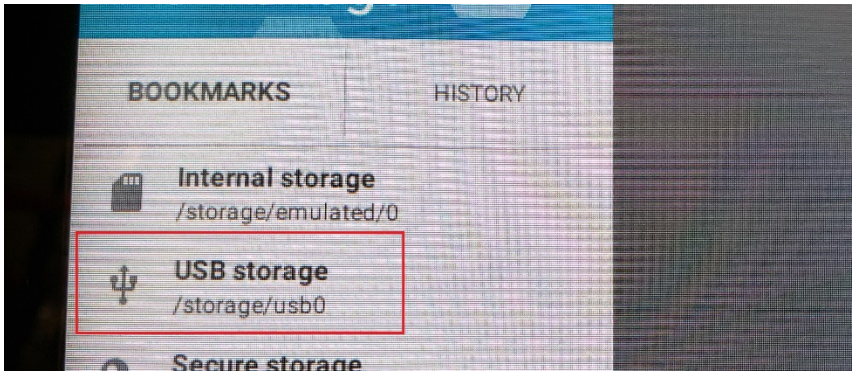


1201 18TH Street, Suite 210
DENVER, CO, 80202
1.866.654.8683
www.dominionvoting.com

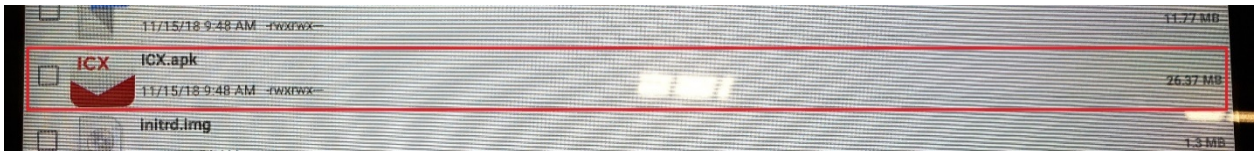
25. Click menu button in the top left corner.



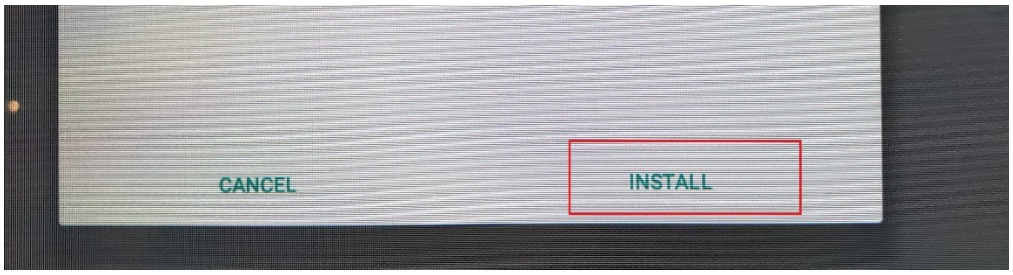
26. Select USB Storage



27. Click ICX.apk

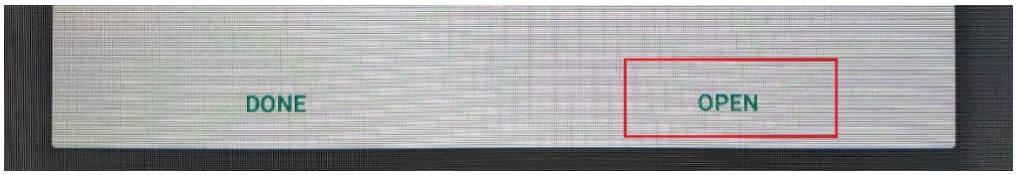


28. Click Install

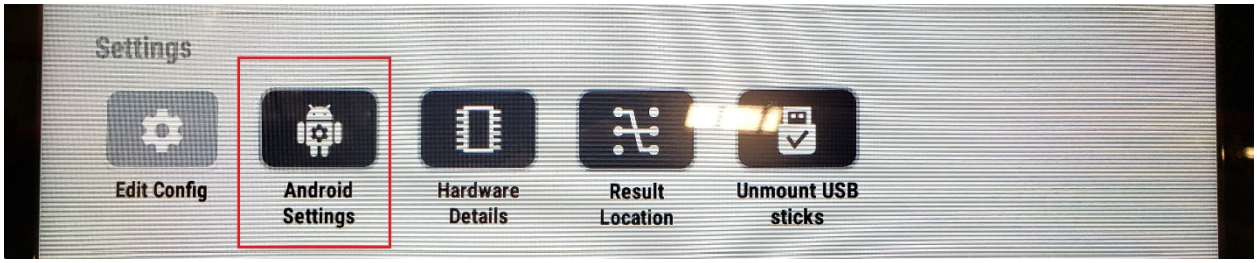


1201 18TH Street, Suite 210
DENVER, CO, 80202
1.866.654.8683
www.dominionvoting.com

29. Click Open

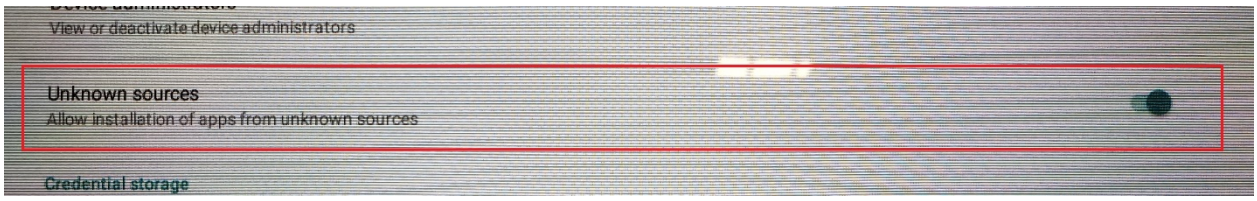


30. Input password and click login
31. Click Android Settings

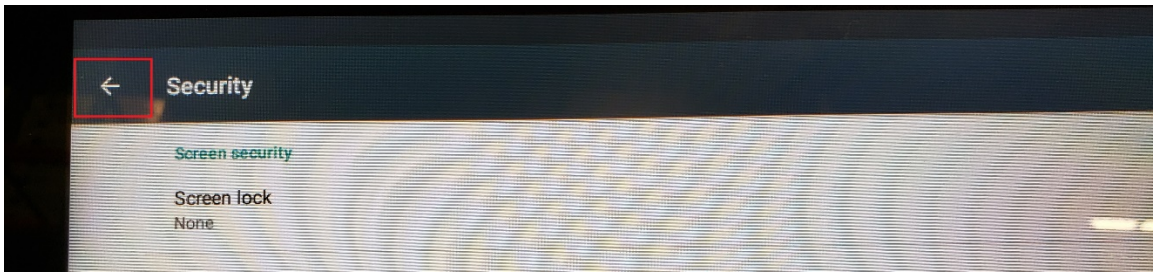


32. Click Security

33. Click to toggle off Unknown Sources *Note: The toggle should be gray when toggled off

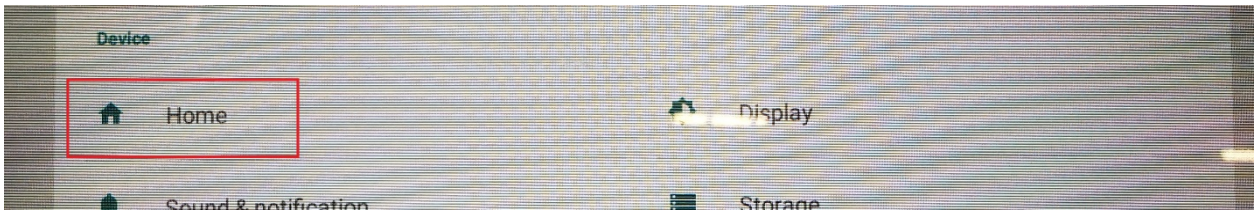


34. Click the back arrow in the top left corner

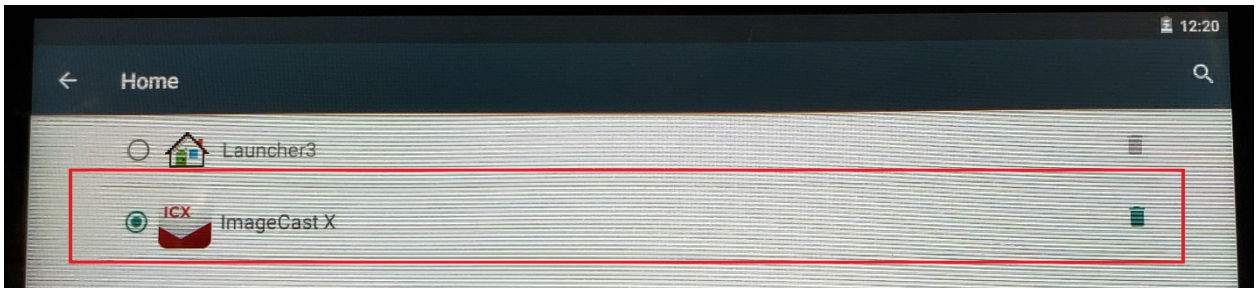


1201 18TH Street, Suite 210
DENVER, CO, 80202
1.866.654.8683
www.dominionvoting.com

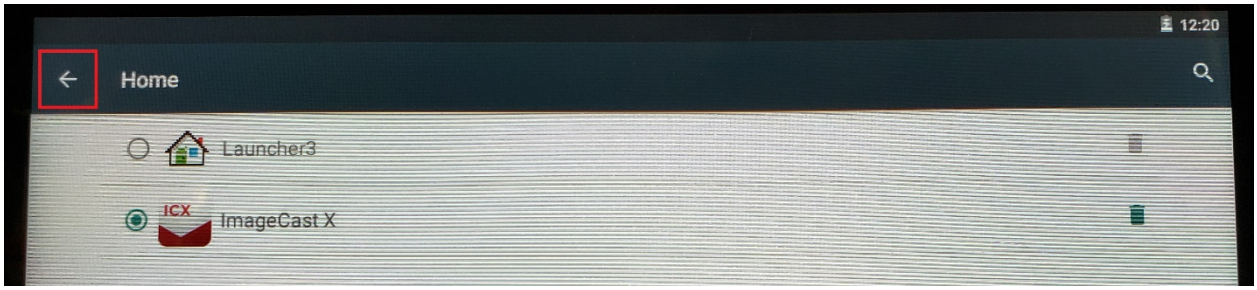
35. Click Home



36. Click Imagecast X



37. Click back arrow



38. Click the Home button at the bottom of the screen.

39. Remove the tech card and USB drive.
40. Software version should now read 5.5.10.32

REDACTED VERSION

Exhibit B: Georgia Logic and Accuracy Procedures

Logic and Accuracy Procedures

Version 1.0
Georgia Secretary of State – Brad Raffensperger

Logic and Accuracy Procedures

Items needed when testing:

From EMS workstation computer create the following items from the Election Project associated to the election for which testing is being conducted:

- Use Election Event Designer Application (EED) for the following:
  - Programmed Technician Card
  - *Programmed Poll Worker Card*
  - USB Drive containing information from GA ICX BMD programming group
  - *Print out of Ballot Activation Codes*
  - *Programmed Compact Flash Cards for Polling Place Scanner*
  - *Programmed Security Key Tab for Polling Place Scanner*
    - **Recommendation**: Create the * items above for each polling location and then use these to L&A test the equipment designated for the same polling place; at completion of L&A test on designated equipment package these items with the tested equipment for delivery to the designated polling place
- Provided by SOS after Election Project Obtained
  - Election Project User names and Passcodes
  - Technician Card Passcode
  - Poll Worker Card Passcode
  - Security Key Tab Passcode
  - Polling Place Scanner Re-zero Passcode
  - Poll Pad User name and Passcode
  - Poll Pad Menu Code

Testing Steps:

A. Preparing the BMDs

- Connect BMD to Printer
- Connect BMD and Printer to power supply
- First, Power Printer On
- Power the BMD On
- Verify installed version in top left corner of screen; v5.5.10.30
- Confirm presence of State Acceptance Test Sticker and seal on top left of BMD
- Insert Technician Card and enter passcode for specific election
- Verify date and time are properly set
  - If time or date needs to be adjusted, touch Modify and set the time and date
  - If time and date are correct, touch Confirm

- Touch Clear All Election Data
- Touch Yes
- Enter passcode
- Touch OK
- Insert USB Drive into an available USB slot in the Election Data compartment of the BMD
- Touch Load Election Data
- Select the data file to be loaded from the USB Drive
    - Touch Select
    - Touch Copy
    - Touch Ok
- Remove the USB Drive from the Election Data compartment of the BMD
- Close the Election Data compartment and attach seal, notate the number of the attached seal on paperwork
- Remove Technician Card
- Insert Poll Worker Card and passcode for specific election
- Touch Select Tabulator
- Select the BMD for the Polling location to which this BMD is being assigned
- Touch OK
- Touch Manual Selection Activation and confirm a checkmark appears in the box
- Touch AVS Controller and confirm a checkmark appears in the box
- Touch Open Polls
- Touch Yes
    - If Warning displayed regarding printer, confirm the Printer is connected and On
    - Touch OK
    - Touch Open Polls
    - Touch Yes
- Name of Polling Place BMD is assigned will display in Black on the top left of BMD Screen; confirm correct Polling Place shown
- Name of Election will display in Gray on the top left of BMD Screen; confirm correct Election shown
- Remove the Poll Worker Card
- Confirm Total Ballots Printed in bottom left corner shows zero (0)
- BMD is now ready

B. <u>Preparing the Polling Place Scanner</u>

- Insert the Primary Compact Flash Card into the Poll Worker Slot
- Insert the Backup Compact Flash Card into the Administrator Slot
- Confirm the Polling Place Scanner connected properly to the Ballot Box
- Confirm presence of State Acceptance Test Sticker on right side of scanner
- Power the Polling Place Scanner ON by plugging the Ballot Box into an AC power supply

- When the Polling Place Scanner begins to beep, beep, beep; align and carefully press down the Security Key Tab to the Security Key Slot
- When prompted on the screen, key in the passcode for the specific election, then press Enter
- Confirm Date and Time, modify the date and time if necessary
- Touch Utilities
- Touch Diagnostics
    - Touch Simple
    - Touch Yes after Thermal Printer test
    - Touch Print
    - Touch No
    - Review printed tape, confirm software version 5.5.3-0002
    - If any item in the Diagnostic test fails, do not proceed
- Touch Open Poll
    - Enter passcode for specific election, if prompted
- Touch Zero
    - Confirm tape shows zero results for all candidates in all races
    - If results are not zero, do not proceed
- Touch No for additional copies
- Confirm Polling Place Scanner shows zero (0) ballots cast
- Polling Place Scanner is now ready to accept ballots

C. <u>Preparing Poll Pads for BMD LA Testing</u>

- Notification of the LA/Advance Voting data set for Poll Pad along with a QR Code image for scanning by Poll Pad will be forwarded to those locations with a scheduled election
- Reference Poll Pad training documents and materials for assistance if the following steps need further explanation
- Power on Poll Pads to be used for LA/Advance Voting; this will not be ALL Poll Pads but only specific Poll Pads
- Connect designated Poll Pads to the appropriate connection
- Launch the Poll Pad application and scan the QR code image; follow prompts displayed on Poll Pad to obtain the Poll Pad LA/Acceptance Data set for the scheduled election
- Once download of the data file is complete, close the Poll Pad application
- Disconnect the Poll Pad from the appropriate connection
- Launch the Poll Pad application again
- Touch Get Started
- Enter User name and Passcode for specific election
- Touch Manual Entry
- Key in the Precinct Name or Precinct ID into the Last Name field
- Touch Search
- Touch the Precinct and Combo record desired and follow the prompts to create a voter card

- Create a voter card from Poll Pad for each unique ballot style within the designated Polling Location
    - Recommend labels be placed on card identifying what ballot style will be displayed by BMD once card is inserted
    - BMD removes the activation code from the Voter Card once used, therefore create the card again from Poll Pad after each use by a BMD

D. Testing the BMD and Printer

Use a combination of Poll Worker Card with Ballot Activation Codes for the polling location, and Voter Cards created from a Poll Pad loaded with the LA/Advance Voting dataset to bring up ballots on the BMD

- Produce at least one printed ballot from each BMD assigned to the polling location
- Produce a test deck from the BMDs assigned to the polling location for each unique ballot style within the polling location. The test deck must contain at least one vote for each candidate listed in each race within the unique ballot style
    - **Example**: Ballot from BMD 1 contains a vote for only the first candidate in each race listed on Ballot Style 1, Ballot from BMD 2 contains a vote only for the second candidate in each race on Ballot Style 1, and continue through the line of devices until all candidates in all races within the unique ballot style have received a single vote
    - **If Number of BMDs outnumber the number of vote positions on the unique ballot style**, start the vote pattern over until all BMDs have produced one printed ballot
    - **If Number of unique ballot styles in the polling place is greater than 1,** once the vote pattern is complete for a unique ballot style, proceed to the next BMD in line to start the review of the next unique Ballot Style
    - **All unique ballot styles do not have to be tested on each BMD**
- Review BMD-generated Test Deck and confirm the vote content before placing in the designated Polling Place Scanner

E. Testing the Polling Place Scanner

- Scan the BMD-generated Test Deck into the Polling Place Scanner
- Scan one blank optical scan ballot style(s) associated to the Polling Place to verify the Polling Place Scanner will recognize the ballot style in case of emergency
- Verify Scanner(s) shows a number of Ballot Cast equal to the number of ballots in the BMD-generated test deck plus the scanned blank Optical Scan ballot styles
- Firmly place the Security Key Tab in the Security Key Slot
- Touch Close Polls
- Enter the passcode
- Touch Enter
- Touch Yes
- Touch No for additional tapes (Scanner will automatically produce 3 copies of the closing tape)

- Review the results tape and confirm result printed matches the known vote content of the BMD-generated and Optical Scan ballot test deck
    - If results do not match, the scanner has failed, do not proceed
- Touch Power Down
- Touch Yes
- Unplug the Ballot Box from the AC power supply
- When the unit is OFF, open the Poll Worker card slot door and remove the Poll Worker Compact Flash Card
- The Poll Worker Compact Flash card for each Polling Place Scanner **MUST** be uploaded to the RTR application to confirm the Compact Flash card can be recognized and results transferred to the EMS for tabulation; then validate and publish the uploaded result file
- After the Compact Flash Card is uploaded to RTR, return the Compact Flash card to its designated Polling Place Scanner and re-insert it into the Poll Worker card slot

F. Preparing the BMD for Election

- Insert the Poll Worker Card and enter passcode
- Touch Admin Menu
- Touch Close Polls
- Touch Yes to confirm
- Touch Reset
- Touch Yes to confirm
- Enter Passcode for specific election
- Touch Ok to confirm
- Confirm Public Counter is at Zero (0); center of BMD screen
- Touch Power off, bottom right corner of screen
- Touch Yes to confirm
- Remove Poll Worker Card
- Turn Printer Off
- Disconnect the Power and Printer from BMD
- Close and seal the Power and Election Data compartments on the right side of the BMD
    - Make note of the seals attached
- Make the BMDs and Printers ready for delivery to the Polling Place

G. Preparing the Polling Place Scanner for Election

- Confirm that both the Poll Worker and Administrator Compact Flash cards are inserted into their assigned slots
- Power the Polling Place Scanner ON by plugging the Ballot Box into an AC power supply
- When the Polling Place Scanner begins to beep, beep, beep; align and carefully press down the Security Key Tab to the Security Key Slot

- When prompted on the screen, key in the needed passcode, then press Enter
- Confirm Date and Time, modify the date and time if necessary
- Touch Utilities
- Touch Re-Zero
- Enter Re-Zero passcode
- Confirm Ballot Cast shows as Zero (0)
- Touch Utilities
- Touch Report
- Touch Election Report
- Touch Zero
- Enter Number of Reports to print = 1
- Touch Enter
- Touch No for additional copies
- Confirm tape shows zero results for all candidates in all races
- Remove the tape and place with L&A Paperwork
- Touch Power Down
- Touch Yes
- Unplug Ballot Box from AC power supply
- Open Ballot Box and remove all test ballots from the Main bin, from the Write-In bin, and from the emergency bin
- Confirm that all bins are empty and properly secured
- Close and seal the ballot box, make note of the seals applied
- Place seals on the Poll Worker and Administrator Compact Flash Card doors, make note of the seals applied
- Make Ballot Box with attached Polling Place Scanner ready for delivery to the Polling Place

H. Testing ICC Workstation and Central Scanner

- Load ICC ABS tabulator Data Set to ICC workstation computer DVS folder
- Launch ICC Application
- Import Tabulator
- Attach Security Key Tab; Enter Passcode
- Enter Name of Project equal to name of ICC ABS tabulator
- Click Load
- Click Configuration
- Set secondary results path
- Verify Scanner is On and recognized by the ICC application
- Click Scan Options
- Set Ballot configuration to Dynamic
- Set scanner to Stop on Overvotes for ALL races

- Set Scanner Continuity to Continuous Scan
- Click OK
- Click Scanning
- Click YES
- Insert Test Deck with known result
- Click Scan
- Verify Scanner recognizes and scans all ballots within the test deck
- Verify Scanner recognizes any error ballots that may be included within the test deck
- Verify Scanner recognizes any overvotes
- Accept the Batch
- Close the ICC application
- Remove ICC ABS tabulator Data Set from ICC workstation computer DVS folder
- Open RTR Application on EMS Workstation
- Open election project
- Load Results from ICC via Secondary path established on ICC workstation
- Click Load
- Click Election Summary Report and generate Election Summary Report prior to validating and publish result file from ICC
- Click Result Files
- Click Search
- Select ICC ABS result file
- Click Validate and Publish
- Click Election Summary Report and generate Election Summary Report
- Verify Results shown for ABS match the known result of the Test Deck scanned by the ICC

I. Upload LA results to ENR

- After results have been loaded to RTR from the Polling Place and ICC scanners
- Create folder on the Desktop of EMS computer labeled State Export
- Open RTR, click Export
- Click Export Type
- Click Search
- Verify that ONLY the GA Export File type is active (contains a checkmark)
- In tool bar, click Settings>Transfer Points
- Click Add
- Click Browse; Select the folder on the Desktop labeled State Export
- In Connection Name type State Export
- Click OK
- Click Save
- Below Tool Bar, Click Start Results Export

- Minimize RTR
- Open State Export folder on Desktop
- Confirm Export file present in State Export folder
- Extract export file and upload to State ENR
- Open EED
- Create folder to Desktop labeled "*Name of Election*-Backups"
- Create a Backup copy of the Election Project
- Copy the saved Backup zip file and accompanying SHA file and place in Backups folder; copy the folder containing the backups to removable media
- Clear results from RTR
- Print new Election Summary Report from RTR confirming all LA results have been cleared
- Close RTR and EED on EMS workstation

J. <u>Loading Election Day Dataset to Poll Pad</u>

- Approximately one week prior to the scheduled Election Day, notification of Election Day data files for Poll Pad along with a QR Code image for scanning by Poll Pad will be forwarded to those locations with a scheduled election
- Power on Poll Pad
- Connect Poll Pads scheduled for use on Election Day to the appropriate connection
- Launch the Poll Pad application and scan the QR code image; follow prompts displayed on Poll Pad to obtain the Poll Pad Election Day Data set for the scheduled election
- Once download of the data file is complete, close the Poll Pad application
- Disconnect the Poll Pad from the appropriate connection
- Launch the Poll Pad application again
- Touch Get Started
- Enter the User name and Passcode for the specific election
- Confirm the proper Election and Polling Location are shown at the top of the screen
- Confirm the number of Precinct Records (voters assigned to location) is accurate
- Confirm Check-Ins are Zero (0)
- Connect Voter Card Encoder to Poll Pad and confirm encoder is recognized by Poll Pad (green indicator at top right of screen)
- Connect power cord to Voter Card Encoder and verify power flows through Voter Card Encoder and charges the Poll Pad
- Touch Scan Barcode
- Confirm camera is operational
- Touch Cancel
- Touch Manual Entry
- Key in last name of known voter in polling place
- Touch Search

- Touch selected voter record, confirm voter information shown is accurate
- Touch Accept
- Confirm Voter Certificate is displayed with signature line
- Put in example signature
- Touch Done Signing
- Confirm Poll Officer Initial box is operational
- Touch Submit
- Touch Touchscreen
- Insert Voter Card into Voter Card Encoder
- Verify Ballot Style and Ballot Activation Code display at bottom of screen
- Confirm Create Card button at top of screen becomes active
- Touch Create Card to verify Voter Card can be created
- Touch Manual Entry
- Find previous Voter
- Touch Wheel and Enter password; confirm password for specific election recognized
- Cancel Voter Check-in
- Spoil Ballot
- Verify mark has been removed
- Press iPad Home button to Close Poll Pad Application

K. <u>Loading Update File to Poll Pad</u>

- On the Saturday prior to the scheduled Election Day, notification of Election Day update data files for Poll Pad will be forwarded to those locations with a scheduled election
- Power on Poll Pad
- Connect Poll Pads scheduled for use on Election Day to the appropriate connection
- Launch the Poll Pad application and follow prompts displayed on Poll Pad to obtain the Poll Pad Election Day Data set for the scheduled election
- Once download of the data file is complete, close the Poll Pad application
- Disconnect the Poll Pad from the appropriate connection
- Launch the Poll Pad application again
- Touch Get Started
- Enter User name and Passcode for the specific election
- Confirm the proper Election and Polling Location are shown at the top of the screen
- Confirm the number of Precinct Records (voters assigned to location) is accurate
- Confirm Check-Ins are Zero (0)
- Connect Voter Card Encoder to Poll Pad and confirm encoder is recognized by Poll Pad (green indicator at top right of screen)
- Connect power cord to Voter Card Encoder and verify power flows through Voter Card Encoder and charges the Poll Pad

- Touch Menu
- Touch Summary Report
- Touch Absentees; confirm expected number of Absentee Voters for polling location
- Touch Home
- Touch Get Started
- Touch Scan Barcode
- Confirm camera is operational
- Touch Cancel
- Touch Manual Entry
- Key in last name of known voter in polling place
- Touch Search
- Touch selected voter record, confirm voter information shown is accurate
- Touch Accept
- Confirm Voter Certificate is displayed with signature line
- Put in example signature
- Touch Done Signing
- Confirm Poll Officer Initial box is operational
- Touch Submit
- Touch Touchscreen
- Insert Voter Card into Voter Card Encoder
- Verify Ballot Style and Ballot Activation Code display at bottom of screen
- Confirm Create Card button at top of screen becomes active
- Touch Create Card to verify Voter Card can be created
- Touch Manual Entry
- Find previous Voter
- Touch Wheel and Enter password; confirm password for specific election recognized
- Cancel Voter Check-in
- Spoil Ballot
- Verify mark has been removed
- Press iPad Home button to Close Poll Pad Application
- Power Poll Pad off
- Place Poll Pad along with Voter Card Encoder, stand, charging cord and AC plug into case
- Close Case and Seal; notate seal on paperwork

REDACTED VERSION

Exhibit C: Pro V&V Field Audit Report

Field Audit Report

Dominion Voting Systems
Democracy Suite (D-Suite) System
Version 5.5-A

Approved by: Jack Cobb

Jack Cobb, Laboratory Director

Approved by: Wendy Owens

Wendy Owens, VSTL Program Manager

December 2, 2020

1.0 INTRODUCTION

The purpose of this Report is to document the procedures that Pro V&V, Inc. followed to perform a Field Audit of the Dominion Democracy Suite (D-Suite) 5.5-AVoting System as fielded in selected counties in the State of Georgia.

1.1 References

The documents listed below were utilized in the development of this Report:

- Election Assistance Commission (EAC) 2005 Voluntary Voting System Guidelines (VVSG) Version 1.0, Volume I, "Voting System Performance Guidelines", and Volume II, "National Certification Testing Guidelines"
- Election Assistance Commission Testing and Certification Program Manual, Version 2.0
- Election Assistance Commission Voting System Test Laboratory Program Manual, Version 2.0
- National Voluntary Laboratory Accreditation Program NIST Handbook 150-2016, "NVLAP Procedures and General Requirements (NIST Handbook 150)", dated July 2016
- National Voluntary Laboratory Accreditation Program NIST Handbook 150-22, 2008 Edition, "Voting System Testing (NIST Handbook 150-22)", dated May 2008
- United States 107th Congress Help America Vote Act (HAVA) of 2002 (Public Law 107-252), dated October 2002
- Pro V&V, Inc. Quality Assurance Manual, Version 7.0
- EAC Requests for Interpretation (RFI) (listed on www.eac.gov)
- EAC Notices of Clarification (NOC) (listed on www.eac.gov)

1.2 Terms and Abbreviations

The terms and abbreviations applicable to the development of this Test Report are listed below:

"COTS" – Commercial Off-The-Shelf

"DRE" – Direct Record Electronic

"EAC" – United States Election Assistance Commission

"EMS" – Election Management System

"FCA" – Functional Configuration Audit

"HAVA" – Help America Vote Act

"ICC" – ImageCast Central

"ICX" – ImageCast X

"ICP" – ImageCast Precinct

"ISO" – International Organization for Standardization

"NOC" – Notice of Clarification

"QA" – Quality Assurance

"RFI" – Request for Interpretation

"VSTL" – Voting System Test Laboratory

"VVSG" – Voluntary Voting System Guidelines

1.3 Background

On Thursday, November 12, 2020, Pro V&V received a request from the Office of the Georgia Secretary of State to perform a Field Audit of the Dominion Democray Suite (D-Suite) 5.5-A Voting System in multiple counties, as selected by the Office, throughout the State. The purpose of this Field Audit was to verify the software/firmware and hardware used during the 2020 General Election was the same as the software/firmware and hardware that were Certified for Use by Georgia's Secretary of State Office.

1.4 System Description

The Democracy Suite 5.5-A Voting System is a paper-based optical voting system consisting of the following major components: the ImageCast Central (ICC) optical ballot scanner, the ImageCast Precinct (ICP) precinct count tabulator, and ImageCast X (ICX) BMD ballot marking device.

ImageCast Central (ICC) Count Scanner

The ICC is a high-speed, central ballot scan tabulator based on Commercial off the Shelf (COTS) hardware, coupled with the custom-made ballot processing application software. It is used for high speed scanning and counting of paper ballots.

ImageCast X (ICX) Ballot Marking Device (BMD)

The ICX consists exclusively of COTS available hardware and operating system, while the applications installed on top customize its behavior to turn it into a Ballot Marking Device (BMD). The ICX is designed to perform the following functions: ballot review and second chance voting, accessible voting, and ballot marking.

ImageCast Precinct (ICP)

The ICP device is a precinct optical scan paper ballot counter designed to provide six major functionalities: ballot scanning, second chance voting, ballot review, tabulation, and poll worker functions.

For ballot scanning functionality the ICP scans marked paper ballots, interprets voter marks on the paper ballots and stores the ballots for tabulation when the polls are closed.

Second Chance voting refers to scenarios in which an error has been detected on the voter's paper ballot (e.g., blank ballot, undervoted ballot, overvoted ballot, misread ballot, cross-over voted ballot), and the ICP notifies the voter by displaying a message or providing an audio visual cue, that one of these situations has been detected, and offers the voter an opportunity to reject and fix their ballot, or to cast the ballot as-is.

The Ballot Review feature allows a voter to review their vote selections using a visual representation, which displays to the voter a complete listing of all contests contained on the ballot and an indication of the results which will be recorded for each contest once the voter's ballot is cast.

The Tabulation of paper ballots cast by voters is performed when the polls are closed on the ICP unit and the unit tabulates the results, generates results files for aggregation into RTR, and prints a results report containing the results of the ballots cast.

For poll worker functions the ICP contains a small touch-screen LCD to allow the poll worker to initiate polling place activities, diagnostics and reports..

1.5 Scope

Pro V&V randomly selected components of the D-Suite system (an ICP, an ICX, and an ICC) from the system in each county that had been utilized in the November 2020 General Election. It was at the discretion of the Pro V&V on-site team which units were subject to verification. The Georgia Secretary of State Office contacted the selected counties and arranged for the Pro V&V team to be granted access to the systems. The selected counties were given less than six hours notice before the Pro V&V team arrived.

2.0 AUDIT OVERVIEW

The evaluation of the D-Suite 5.5-A Voting System consisted of removing a copy of the software/firmware from each component and evaluating the software/firmware against a known SHA-256 hash value outside of the system.

3.0 AUDIT PROCESS AND RESULTS

The following sections outline the audit process that was followed to evaluate the D-Suite 5.5-A Voting System under the scope defined in Section 1.5.

3.1 General Information

The Field Audit was conducted under the guidance of Pro V&V by personnel verified by Pro V&V to be qualified to perform the audit.

3.2 Audit Configuration

An ICX was selected at random from the warehouse in each county. The team member then photographed the seals and the device. All seals that needed to be removed were then removed. After all photographs were taken, the team member inserted a clean USB drive from Pro V&V into left hand access compartment. Next the team member then plugged in the unit and powered it on. At the prompt the team member inserted a Tech Key smart card and selected the option to "Extract Application". The team member then verified the SHA-256 generated by the unit and photographed the popup screen. The team then took the USB drive containing the exported application to a Pro V&V laptop to compare the SHA-256 hash values to the known value from previous testing.

An ICP was selected at random from the warehouse in each county. The team member then photographed the seals and the device. All seals that needed to be removed were then removed. After all photographs were taken, the team member removed any compact flash cards under county supervision and inserted two compact flash cards (one blank and the other containing techextract.enc that was created by Pro V&V during certification testing). The unit was plugged in and powered on. A password was entered and a tech iButton was then read by the ICP and the option to "Extract Firmware" was selected. The compact flash cards, if present, were returned to the same ICP. The team member then took the compact flash card containing the exported firmware to a Pro V&V laptop to compare the SHA-256 hash values to the known value from previous testing.

An ICC was selected at random in each county central office if there were multiple units. The team member then photographed the device. The county provided the creditals to login to the workstation. The Pro V&V team member navigated to the ICC folder on the root of the workstation and copied all application files onto a Pro V&V USB drive. The team member then took the USB drive containing the exported firmware to a Pro V&V laptop to compare the SHA-256 hash values to the known value from previous testing.

3.3 Summary Findings

During the Field Audit, a total of eighteen (18) components located among six (6) counties were evaluated to verify the version of software/firmware running on each device. It was discovered that allversions on all components matched the known certified SHA-256 hash value.

4.0 CONCLUSIONS

Based on the results obtained during the Field Audit, Pro V&V determines the D-Suite 5.5-A Voting System, on all evaluated components, is the voting system certified by the Georgia Secretary of State Office.

Mesa County Colorado Voting Systems

Report #1 with Forensic Examination and Analysis

September 2021

Mesa County, Colorado, Voting Systems

Report #1 with
Forensic Examination and Analysis

15 September 2021

Table of Contents

Executive Summary 1

Introduction 3

Legal References 5

Forensic Examination and Analysis Report 7

Forensic Analysis 8

System Identification 8

Authenticity and Chain of Custody 10

FINDINGS 11

- Overview of System Data Sources 11
- Server Disk Partition Structure Overwritten 12
- Website Server Log Files Missing 15
- Server Microsoft SQL Server Installation Log Files Missing 17
- Server Microsoft SQL Server Log Files Missing 19
- EMS Server Dell Server Updates Missing 20
- Server 'Administrator' WebCache Log Files Overwritten 22
- Server 'emsadmin' WebCache Log Files Overwritten 23
- Server SQL Server Management Studio (SSMS) Log Files Overwritten 25
- Server CBS Log Files Overwritten 26
- Server Election Databases Missing 27
- Server Event Logs Missing/Overwritten 29
- Server System Users are Missing 31
- Server Virtual Directories Log Files Missing 32
- Server Windows Defender Log Files Missing/Overwritten 33
- Server List of .log files in Before Image that were Deleted. 34
- Significant Number of Logfiles Missing 35
- List of .evtx Event Log Files deleted 36

Analysis Summary 41

Conclusion 41

Appendix A. Deleted ".log" files after Dominion Trusted Build update 42

Appendix B. Supporting Documentation: File details and hash sets for screenshots 61

Appendix C. Microsoft EVENT log files 62

Appendix D. List of Figures 76

Appendix E. 2002 Voting Systems Standards (VSS) 77

EXECUTIVE SUMMARY

This report documents initial findings in an ongoing forensic examination of the voting systems of Mesa County, Colorado, used in the November, 2020 General Election. These voting systems represent a portion of overall election systems infrastructure, and this report is limited to the findings of an ongoing investigation. The findings in this report were prepared by the cyber forensic expert retained to advise the County Clerk pursuant to her duties as the county's Chief Election Official as part of the impacted parties' legal team.

Federal law requires the preservation of election records – which includes records in electronic or digital form – for twenty-two months after an election. Colorado law requires the preservation of election records for an additional three months beyond the Federal requirement. The obligation to ensure the integrity of elections and that all election records are preserved pursuant to federal and state law falls to the elected Clerk & Recorder. This report, the first of several, is based on examination of the data obtained from forensic images of the Dominion Voting System EMS server last used in Mesa County for the November, 2020, election, images taken in furtherance of the preservation requirements of federal and state law. Based upon information received by the Clerk's office from various sources in early 2021, the Clerk became concerned that the voting system modifications might jeopardize these preservation and other legal requirements under the responsibility of the County Clerk. For this reason the Clerk ensured a full backup of election records from the County voting systems, both before and after the software modification performed by the vendor and the Secretary of State on May 25-26, 2021, just six months after the November, 2020, election.

Forensic examination[1] found that election records, including data described in the Federal Election Commission's 2002 Voting System Standards (VSS) mandated by Colorado law as certification requirements for Colorado voting systems, have been destroyed on Mesa County's voting system, by the system vendor and the Colorado Secretary of State's office. Because similar system modifications were reportedly performed upon county election servers across the state, it is possible, if not likely, that such data destruction in violation of state and federal law has occurred in numerous other counties.

The extent and manner of destruction of the data comprising these election records is consequential, precluding the possibility of any comprehensive forensic audit of the conduct of any involved election. This documented destruction also undermines the conclusion that these Colorado voting systems and accompanying vendor and Colorado Secretary of State-issued procedures could meet the requirements of Colorado and Federal law, and consequently vitiates the premise of the Colorado Secretary of State certification of these systems for use in Colorado.

Two backup images, using forensic imaging methods, were obtained from the Dominion Voting Systems (DVS) Democracy Suite (D-Suite) Election Management System (EMS) Standard Server in Mesa County, Colorado. The first image was made of that EMS Standard Server in the D-Suite 5.11-CO version configuration, as used in the November, 2020 election. The second image was of the configuration of the EMS Standard Server in the D-Suite 5.13 version configuration, following the modification of the EMS Standard Server by a combined team of DVS vendor personnel and Colorado Secretary of State staff. The forensic information provided in this report is presented using screenshots from forensic analysts' systems running industry-standard forensics software tools. The report includes "before" and "after" screenshots from the forensic tool that shows the differences between the two backup images.

The forensic examination found that numerous logfiles had been deleted or overwritten. These logfiles are required to reconstruct the function of and events taking place on the the voting systems, and based upon information

[1] Many individuals and organizations, some public officials, have made recent claims that no audit performed nor examination conducted on elections or computer-based election systems can be legitimate or credible unless the examiners are "election experts" or accredited election auditors. There is no such thing as an "accredited election auditor," nor are there Federal standards or procedures to credential election auditors.

provided by legal counsel, must, by law, be preserved. By comparing filenames in the two images (before and after the Dominion update on May 25-26, 2021), examination and analysis identified a total of 28,989 files that were deleted. During a software update, some replacement of program files and their related content is normally expected. However the examination found that 695 log and event log files necessary for the determination of election integrity were deleted.

Based upon information provided by legal counsel, Colorado law (Colorado Revised Statute (CRS) § 1-5-601.5) requires that, prior to use in Colorado elections, electronic and computer-based voting systems be certified by the Colorado Secretary of State. This certification is based on the systems' compliance with the requirements of the Federal Electon Commission's 2002 Voting System Standards (VSS), verified by their testing by a Federally-accredited (by vote of the U.S. Election Assistance Commission (EAC)) Voting System Testing Lab (VSTL). While several iterations of newer Voluntary Voting System Guidelines (VVSG) have been issued by the EAC, Colorado's statutory requirement is for compliance with 2002 VSS, which states:

> "Election audit trails provide the supporting documentation for verifying the accuracy of reported election results. They present a concrete, indestructible archival record of all system activity related to the vote tally, and are essential for public confidence in the accuracy of the tally, for recounts, and for evidence in the event of criminal or civil litigation."

The relevant sections of the VSS are cited in Appendix E.

These statutory requirements establish that voting systems are required to generate and preserve, as critical to the ability to determine and reproduce the conditions and details of election conduct using these systems, logfiles of all system functions, including normal activity, connectivity, file and data access, operator- and automated-processes, and errors. Logfiles are critical to the ability to detect improper operation, including the ability to detect malicious intrusions as well as other improper activities and conditions, and configuration changes that could enable alteration of the actual vote count.

Assuming this information to be correct, this forensic examination found that a substantially large number of these requirements have not been met. This examination also found that destruction of critical logfiles has occurred. This destruction is not incidental or minor but is extensive.

The purpose of this initial report is to document these findings and present preliminary evidence demonstrating unacceptable conduct and system defects revealed by the examined images, as necessary for the Chief Election Official to discharge her statutory obligations. The facts and resultant findings support the conclusions that:

1) Election-related data explicitly required to be preserved, as stated in the 2002 VSS criteria referenced in this section, have been destroyed in violation of Federal and State law, and

2) Due to non-compliance with the 2002 VSS requirements, these voting systems and accompanying vendor-provided, Colorado Secretary of state-approved procedures cannot meet the certification requirements of the State of Colorado, and should not have been certified for use in the state.

Comprehensive investigation is required to determine whether these critical failures are the result of malicious intent or negligence, and to what extent the systems may have been compromised or subjected to unauthorized access or operation prior to, during, and after election use. That comprehensive investigation *is beyond the scope of this report*. Subsequent reports will address these issues in detail.

Evidence supporting all of these findings is documented in this report.

Introduction

Election officials, including Secretaries of State, are obligated by law to ensure the integrity of all elections, including the transparency required for citizens to verify that integrity themselves. Modern electronic voting systems are marketed as an efficient solution to streamline the voting process and allow for automated collection, tabulation, and reporting of election results, but the efficiency they promise comes at a cost.

The necessary measures and safeguards to ensure the integrity of the systems and their operation against a severe, mounting and ever-evolving threat from sophisticated nation-state and non-nation-state actors are so complex and dynamic as to outpace the limited capabilities and resources of our government, at all levels. While minimal security safeguards may be within government capacity, modern computer-based voting systems are extremely complex and difficult to secure, even for cybersecurity experts, and since voting systems are not under the direct control of the Federal government's top security experts, any government assurances about the sufficiency of those safeguards can serve only to mislead citizens and policy-makers. Even critical defense systems, relentlessly monitored and defended by highly-trained teams using costly, sophisticated tools, are at risk and are frequently compromised, sometimes before procurement. Earlier generations of voting systems relied on simple, human-scale safeguards, for example "air gaps"– that is – to have no wired network connection to the system. But miniaturized wireless communication technologies and networks have proliferated, with billions of wireless devices installed or in use, and malicious actors have developed sophisticated attacks to bypass air gaps, compromise every kind of hardware, firmware, and software, often before they even come into customer or user possession, and to move laterally through networked systems, often undetected. Supply-chains for these systems, from the initiation of the design of integrated circuits and electronic components, most manufactured overseas with little U.S. insight or oversight, through the fabrication, testing, assembly, integration, and operation of these complex composite systems, are vulnerable and untrustworthy for critical functions of government and lucrative economic and national security targets. For all these reasons logfiles, such as those that have been deleted by the Dominion "Trusted Build" update must be preserved to document the complete operation of the computer system and voting applications, and to be able to verify the authenticity, integrity and accuracy of the vote.

The feature size of individual circuits in the chipsets and components of our voting system computers is at the nanoscale, smaller than the smallest known virus particle, and less than 3/10,000ths of the width of a human hair. So we have lost the ability, if we ever had it, to visually verify what is really happening, even at the physical level, in our computer-based voting system. Regardless of how the systems appear to be configured to authorized users and poll-watchers, the functionality and connectivity in these computers can be enabled and modified remotely and wirelessly, or by the introduction of embedded codes on scanned paper, or triggered by specific unforeseeable and indiscernible predetermined software and hardware conditions, or by specific timing events, or by geographic location, or by the proximity of other devices or combinations of any of these means.

For example, some Colorado voting systems ordered as specified by the voting system vendors, from foreign manufacturing and assembly facilities, have included "Integrated Dell Remote Access Controllers (IDRAC)," which are designed to allow "out-of-band" remote management of those systems, meaning that the computers are explicitly equipped to be controlled by remote automated programs or by individuals other than those logged in locally. Through the IDRAC, voting systems might have any aspect of their Basic Input/Output System (BIOS), operating system, or applications controlled or modified, including the addition and deletion of user accounts, the enabling of communications components like wireless networking cards, and the modification, installation, removal or configuration of software and settings. Like the inclusion of multi-band wireless networking cards, similarly specified and ordered for Colorado voting systems by the vendor, there is no excuse or rational justification for the inclusion of components like these, and the fact that the entirety of U.S. voting system regulatory processes and institutions can apparently neither detect, note, nor address these gross vulnerabilities eviscerates the notion that our computer-based voting systems have been secured.

Faced with incredible miniaturization, the importance of logfiles which are records of operation of a computer system, are more important than ever in managing this technology. When the computer is part of a national critical infrastructure, these operational records become essential, not only for troubleshooting or security alone, but for the integrity of the system itself as a component of the National Critical Infrastructure.

For the purposes of this document and ensuing discussion, two terms are defined to differentiate and clarify the evidentiary findings. *Election Data* is all information regarding Ballot Design, Ballot Marking, Electronic scanning of completed ballots, interpretation of the intention of each voter's choice, including human, machine generated or programatic adjudication in the event that the election system is unable to determine conclusively the correct vote input from any specific ballot, tabulation of the actual vote including the databases used to actually contain the raw vote totals, scanned ballot images and Voter Registration and Voter identification information associated with any specific election, as well as the actual vote totals. This includes a complete record of any realtime changes in databases resident in the cloud such as voter registration data. *Election-Related Data* includes all of the computer log and configuration data that document the complete configuration state and operation of the entire computer system and infrastructure upon which Election Software is executed, as well as the operating system of devices that store log and election data such as Network Attached Storage (NAS). Also included in Election-Related data are logs and configuration of network Routers, Firewalls, Intrusion Detection Systems, Intrusion Prevention Systems, and other network security devices, including VPNs and more[2].

Both Election Data AND Election-Related Data must be preserved as "Election Records" under the law, and this is broadly addressed in both the 2002 VSS and the EAC's successor versions of VVSG.

Securing computer systems is a non-trivial task. It involves a litany of processes, including, but not limited to:

- Engineering systems with a focus on security
- Building systems to meet published high-security standards and applicable regulations
- Patching systems to ensure that vulnerabilities are removed
- Securing networks to ensure highly controlled access
- Logging of all communications, processes, access, system modifications
- Auditing of systems and logs regularly to ensure ongoing compliance
- Adequate training and certification for engineers, administrators, and system users
- Adherence to Industry Best Practices, for example, emphasis on password strength and configured security and group policies

These, among other measures, will help to ensure what is known as the CIA triad. The CIA triad represents the three pillars of information security: confidentiality, integrity, and availability, as follows:

[2] Log and configuration examination of not only the computer system(s) but also all network systems are critical to forensic examination. Compromise of any unrelated information (e.g. plain-text cofiguration data containing normally-encrypted passwords) can be easily prevented, so long as simple, quick forensic examiner and cyber professional industry standards are used to obfuscate private and sensitive data from the network device files.

Confidentiality – preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information

Integrity — guarding against improper information modification or destruction and ensuring information non-repudiation and authenticity

Availability – ensuring timely and reliable access to and use of information

Failure in any of the three pillars can and generally will result in a compromise of the system. Failure in the integrity component can have dire consequences for public perception, election results, the future of our government and our country.

Industry-standard forensics analysis tools were applied to the forensic examination.

Information was forensically evaluated using backup images taken from a Mesa County Election server configured for DVS D-Suite 5.11-CO on Sunday, May 23, 2021, before its modification by Dominion Voting Systems and the Colorado Secretary of State to DVS D-Suite 5.13, and again on Wednesday, May 26, 2021, after the update had been applied. This server was the primary system that was used to process election data in Mesa County for the 2020 general election.The EMS server configuration and administrative standards were prepared by Dominion Voting Systems (DVS), running a combination of COTS and proprietary DVS software, and certified for use by the Colorado Secretary of State. Our conclusions include determining that this system not only failed to meet any reasonable standard or statutory requirement for cybersecurity but was also subject to removal of critical information (data destruction).

Our findings include serious irregularities that resulted in the loss of data integrity on the server, including election data and election-related data.

LEGAL REFERENCES

Several Federal and Colorado state legal standards apply to the preservation and definition of election records, applicable to the data generated by and resident on voting systems. Beginning with 52 USC §20701, retention and preservation of records and papers by officers of elections; deposit with custodian; penalty for violation, which states:

> Every officer of election shall retain and preserve, for a period of twenty-two months from the date of any general, special, or primary election of which candidates for the office of President, Vice President, presidential elector, Member of the Senate, Member of the House of Representatives, or Resident Commissioner from the Commonwealth of Puerto Rico are voted for, all records and papers which come into his possession relating to any application, registration, payment of poll tax, or other act requisite to voting in such election, except that, when required by law, such records and papers may be delivered to another officer of election and except that, if a State or the Commonwealth of Puerto Rico designates a custodian to retain and preserve these records and papers at a specified place, then such records and papers may be deposited with such custodian, and the duty to retain and preserve any record or paper so deposited shall devolve upon such custodian. Any officer of election or custodian who willfully fails to comply with this section shall be fined not more than $1,000 or imprisoned not more than one year, or both.

In addition to 52 USC §20701, multiple sections of Colorado Revised Statutes (CRS) appear applicable, including:

> CRS 1-5-601.5. Compliance with federal requirements (Effective until July 1, 2022)
>
> All voting systems and voting equipment offered for sale on or after May 28, 2004, shall meet the voting systems standards that were promulgated in 2002 by the federal election commission. At his or her discretion, the secretary of state may require by rule that voting systems and voting equipment satisfy voting systems standards promulgated after January 1, 2008, by the federal election assistance commission as long

as such standards meet or exceed those promulgated in 2002 by the federal election commission. Subject to section 1-5-608.2, nothing in this section shall be construed to require any political subdivision to replace a voting system that is in use prior to May 28, 2004.

CRS 1-7-802. Preservation of election records

The designated election official shall be responsible for the preservation of any election records for a period of at least twenty-five months after the election or until time has expired for which the record would be needed in any contest proceedings, whichever is later. Unused ballots may be destroyed after the time for a challenge to the election has passed. If a federal candidate was on the ballot, the voted ballots and any other required election materials shall be kept for at least twenty-five months after the election.

1-13-716. Destroying, removing, or delaying delivery of election records

(1) No person shall willfully destroy, deface, or alter any ballot or any election records or willfully delay the delivery of any such ballots or election records, or take, carry away, conceal, or remove any ballot, ballot box, or election records from the polling location or drop-off location or from the possession of a person authorized by law to have the custody thereof, or aid, counsel, procure, advise, or assist any person to do any of the aforesaid acts.

(2) No election official who has undertaken to deliver the official ballots and election records to the county clerk and recorder shall neglect or refuse to do so within the time prescribed by law or shall fail to account fully for all official ballots and other records in his charge. Informality in the delivery of the ballots and election records shall not invalidate the vote of any precinct if such records are delivered prior to the canvassing of the votes by the county board of canvassers.

(3) Any person who violates any provision of this section is guilty of a misdemeanor and, upon conviction thereof, shall be punished as provided in section 1-13-111.

And several sections of the Code of Colorado Regulations appear applicable, including:

8 CCR 1505-1, Rule 21, 21.4.2: All voting systems must meet the requirements of the 2002 Voting Systems Standards, parts 5 – 7 of article 5 of title 1, CRS, as amended, and this Rule 21.

FORENSIC EXAMINATION AND ANALYSIS REPORT

SYSTEM IDENTIFICATION

The server that was analyzed is capable of operating on a small local area network (LAN). The network consists of several systems, including servers and workstations running in a non-virtualized environment. The server that we evaluated was named EMSSERVER. It is running the Microsoft Windows Server 2016 Standard operating system.

The forensic evaluation and reviews were based upon a forensic image archive collected from the Mesa County Dominion EMS Server. The Before and After forensic images were collected from the same server and same hard drive, as documented below, from the actual acquisition. The serial number of the hard drive shown in each collection data set verifies the data origin to be the same physical device.

Figure 1 – EMS Server (5.11-CO) Image Attributes Before

```
Created By AccessData® FTK® Imager 4.2.0.13

Case Information:
Acquired using: ADI4.2.0.13
Case Number: 052321
Evidence Number: 00003
Unique description: EMSSERVER

---------------------------------------------------------------

Information for F:\EMSSERVER\EMSSERVER:

Physical Evidentiary Item (Source) Information:
[Device Info]
 Source Type: Physical
[Drive Geometry]
 Cylinders: 121,534
 Tracks per Cylinder: 255
 Sectors per Track: 63
 Bytes per Sector: 512
 Sector Count: 1,952,448,512
[Physical Drive Information]
 Drive Model: DELL PERC H730 Adp SCSI Disk Device
 Drive Serial Number: 00222e64128c016e1d004fc54220844a
 Drive Interface Type: SCSI
 Removable drive: False
 Source data size: 953344 MB
 Sector count:    1952448512
[Computed Hashes]
 MD5 checksum:    3d7cf05ca6e42db765bf5c15220c097d
 SHA1 checksum:   eab06a7ea23586de2746b9142461717e075f5c9f

Image Information:
 Acquisition finished:  Sun May 23 2021
 Segment list:
  F:\EMSSERVER\EMSSERVER.E01
```

Figure 2 - EMS Server (5.13) Image Attributes After

```
Created By AccessData® FTK® Imager 4.2.0.13

Case Information:
Acquired using: ADI4.2.0.13
Case Number: 052621
Evidence Number: 00002
Unique description: EMSSERVER_v2

---------------------------------------------------------------

Information for E:\Mesa\EMSSERVER_v2:

Physical Evidentiary Item (Source) Information:
[Device Info]
 Source Type: Physical
[Drive Geometry]
 Cylinders: 121,534
 Tracks per Cylinder: 255
 Sectors per Track: 63
 Bytes per Sector: 512
 Sector Count: 1,952,448,512
[Physical Drive Information]
 Drive Model: DELL PERC H730 Adp SCSI Disk Device
 Drive Serial Number: 00222e64128c016e1d004fc54220844a
 Drive Interface Type: SCSI
 Removable drive: False
 Source data size: 953344 MB
 Sector count:    1952448512
[Computed Hashes]
 MD5 checksum:    52861d5a7750ab535a9d5f7277469c10
 SHA1 checksum:   1bf8f22edb37f72bb29428a591046a1f64279a3f

Image Information:
Acquisition finished:  Wed May 26 2021
 Segment list:
  E:\Mesa\EMSSERVER_v2.E01
```

Two backup images were obtained, using forensic imaging methods, from the Dominion Voting Systems (DVS) Democracy Suite (D-Suite) Election Management System (EMS) Standard Server in Mesa County, Colorado. The first image was made of that EMS Standard Server in the D-Suite 5.11-CO version configuration, as used in the November, 2020 election on May 23, 2021. The second image was of the configuration of the EMS Standard Server in the D-Suite 5.13 version configuration, following the modification of the EMS Standard Server by a combined team of Dominion Voting System vendor personnel and Colorado Secretary of State (SecState) staff, on May 26, 2021. A forensic image (forensic copy) is a bit-by-bit, sector-by-sector duplicate of a physical storage device using specialized hardware and software; it is a much more comprehensive representation of the state and configuration of the imaged system than could be obtained using simple file backup methods. The images include all files, folders, and unallocated, free, and slack space. These forensic images include not only all the files visible to the server operating system but also deleted files and fragments of files left in the slack and free space but every digital bit of data present on the storage medium, in this case, a SCSI hard disk. When forensic images are acquired, a hash function, also known as a Message Digest, is computed. This hash can be used at any time to validate the integrity of the image to ensure that it has not been edited, modified, or changed in any way. The hash function result from the acquisition of data appears in the text above but also appears inside each respective archive and authenticates the data by demonstrating that it has not changed since it was acquired.

These two images were evaluated to gather technical information, including the integrity of the data stored on the system. No effort was made to reverse-design, de-compile or reverse-engineer the Dominion software. Configuration, which is relevant to the operation of the system, was examined to determine whether improper settings could allow undesirable results and were found to contain such errors. Results relevant to this investigation are documented below. Additional supporting documentation can be found in the appendixes. They include directory listings for many of the directories seen in the screenshots and contain complete filenames, full path names where the files are located, and file hashes.

We have included screenshots that can be used to review and verify these findings. These screenshots were obtained from the forensic images of the Dominion server.

AUTHENTICITY AND CHAIN OF CUSTODY

Digital chain of custody is the record of preservation of digital evidence from collection to presentation in the court of law. This is an essential part of the digital investigation process. The chain of custody is probative that the digital evidence presented to the court remains as originally collected, without tampering. The two images analyzed in this report were obtained through AccessData FTK Imager 4.2.0.13. The serial number on the EMS Server drive on both images match, thus establishing that both images were taken from the same physical drive. I have reviewed the documented chain of custody for both images and have determined that the chain of custody is complete from the forensic operator utilizing FTK Imager through the source from which I directly received these images. (Because of the pending civil litigation and criminal investigation, the written documentation remains in the custody of counsel for later introduction in court proceedings and thus cannot be released as part of this report.) Further confirmation that these are genuine images from the Mesa County EMS Server has been provided by the Colorado Secretary of State's office. See:

https://www.sos.state.co.us/pubs/newsRoom/pressReleases/2021/PR20210817MesaCounty.html

Overview of System Data Sources

Figure 3 – EMS Server (5.11-CO) System Data Sources Before

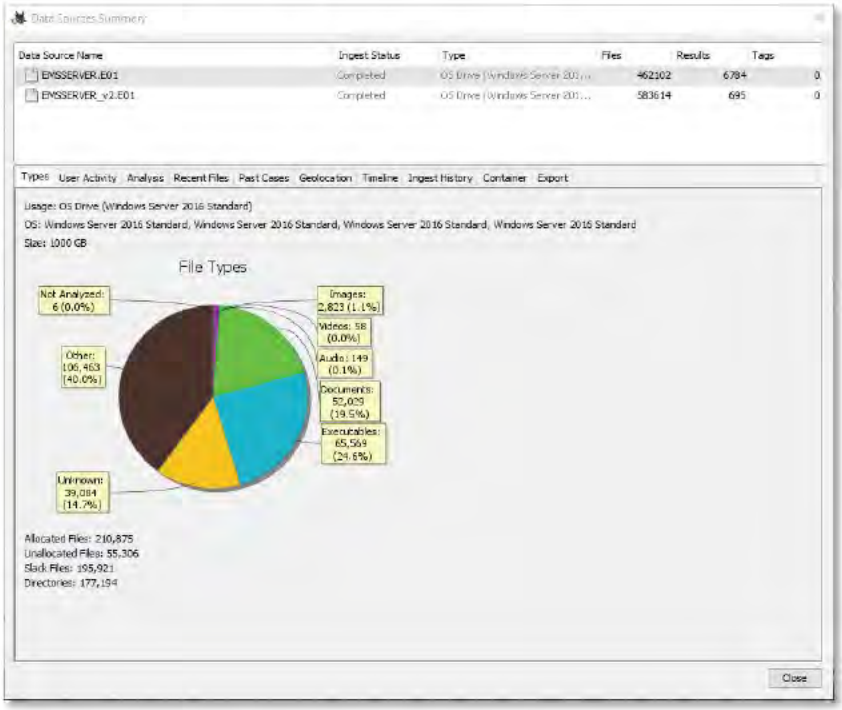
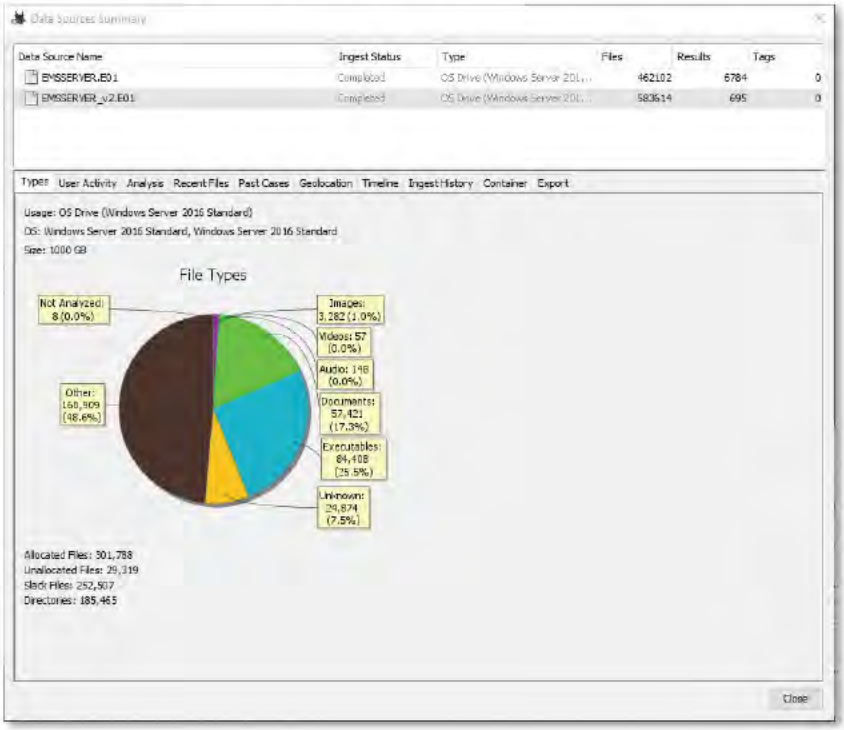


Figure 4 - EMS Server (5.13) System Data Sources After



Purpose: *The disk partition structure is the structure of how the hard drive is divided up.*

Figure 5 - EMSSERVER (5.11-CO) Disk Partition Structure Before

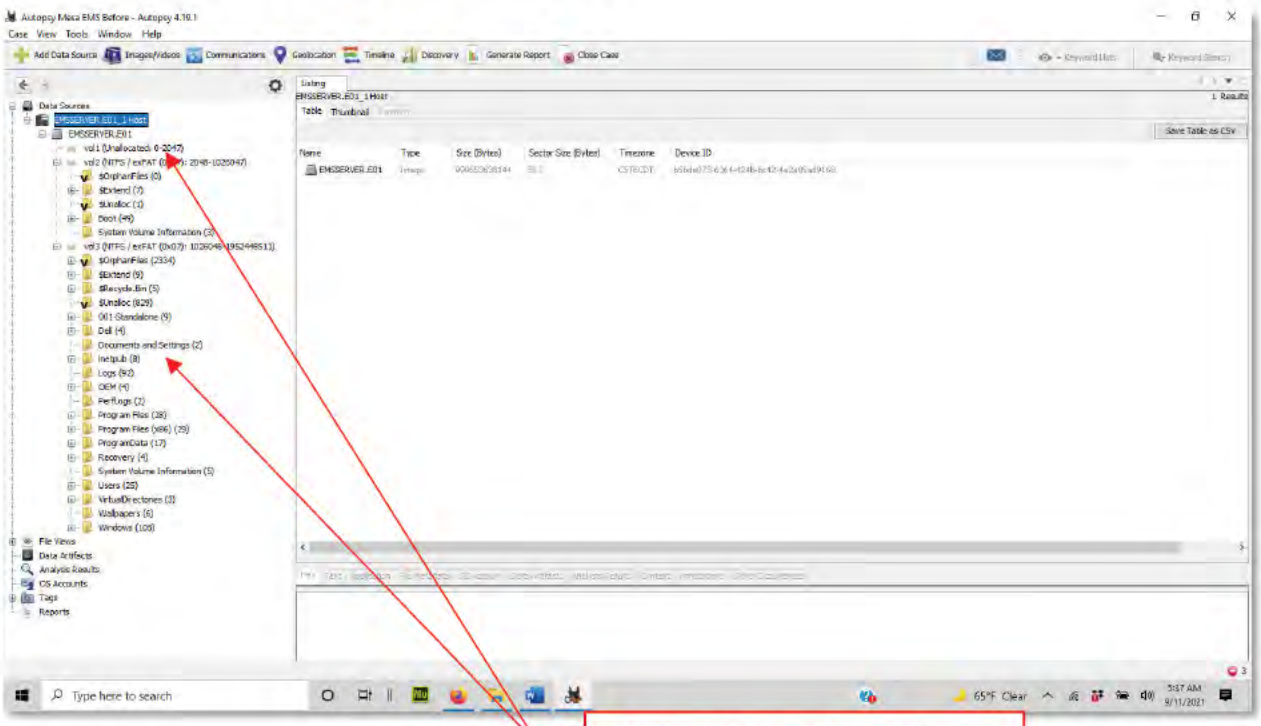


Figure 6- EMSSERVER (5.13) Disk Partition Structure After

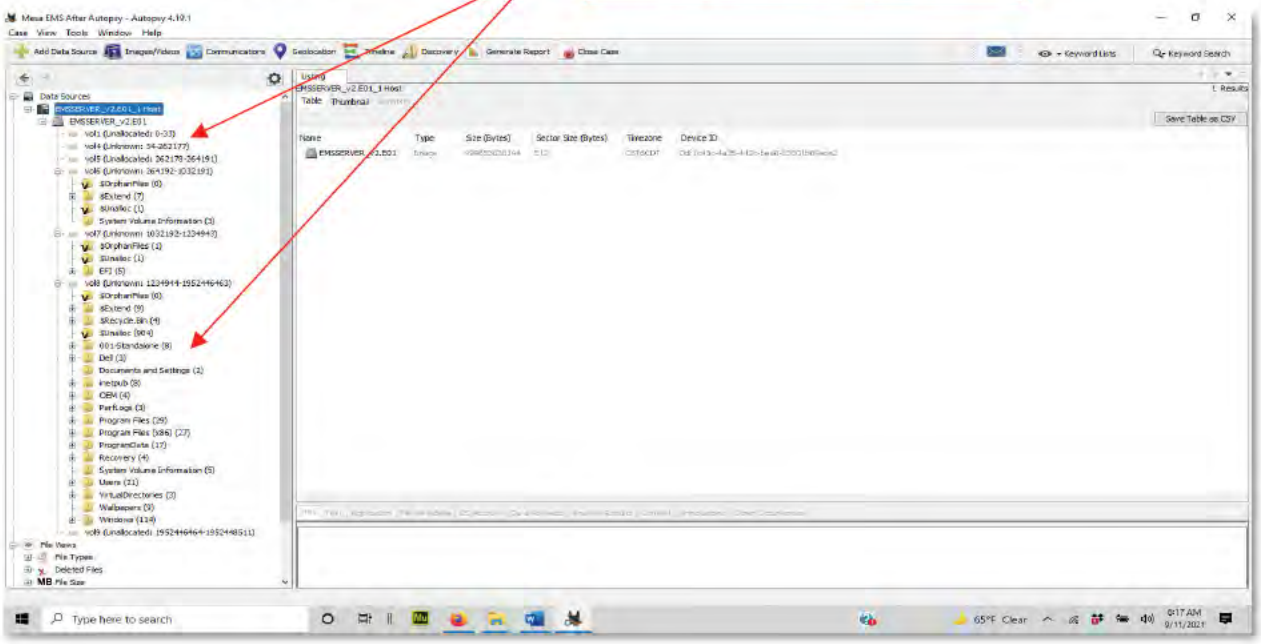
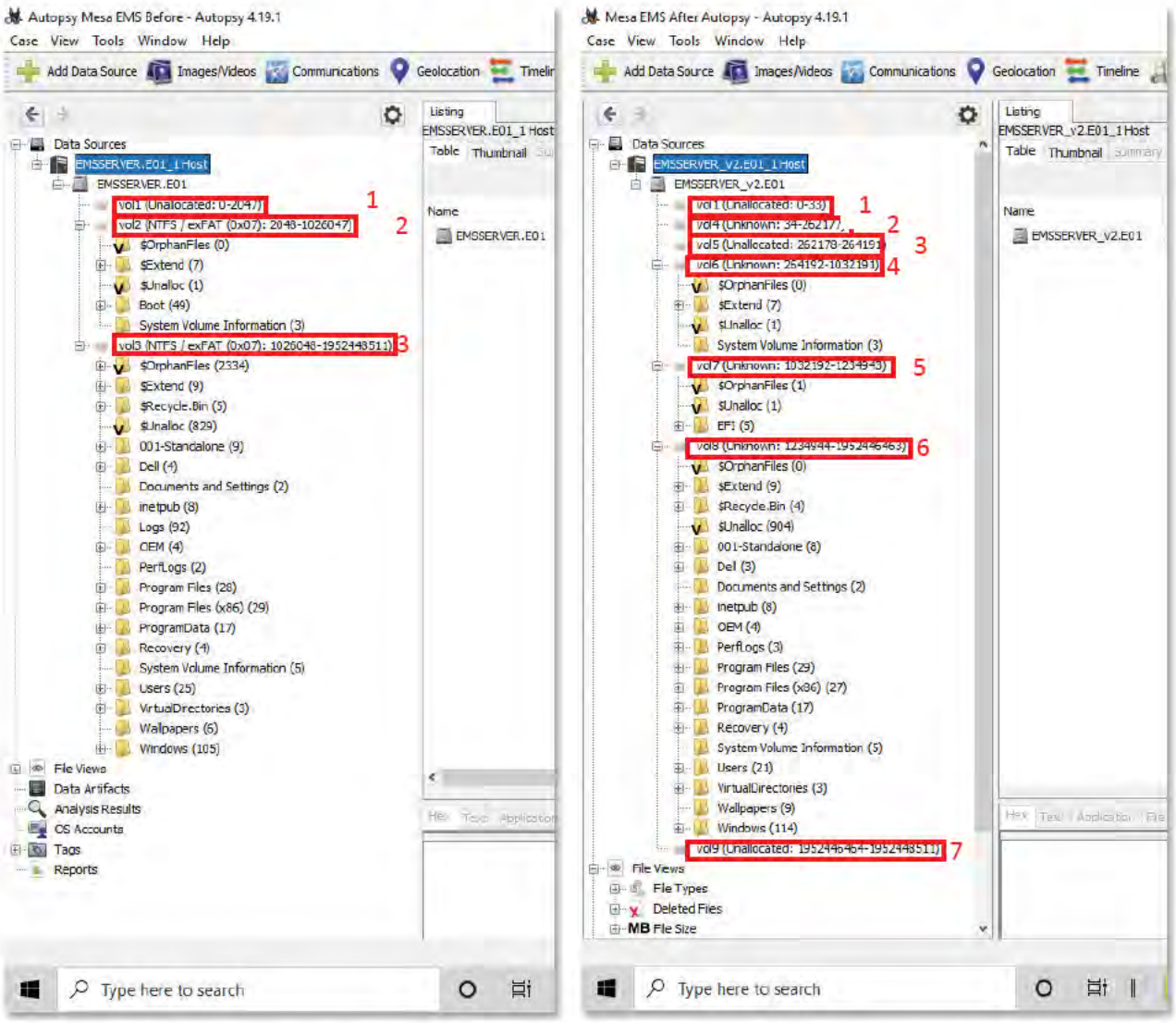


Figure 7 - Server Disk Partition and Directory Changes



Before Dominion Update

After Dominion Update

Computer hard disk drives are data storage devices that must be prepared before use – specifically, they must be partitioned into logical disk volumes and then formatted. Partitioning a hard disk drive is the equivalent of scoring horizontal and vertical rule lines onto blank paper, and then numbering each line, preparing that paper for the orderly recording and look-up of information. A disk is partitioned to organize information into sets of related data. A partition creates a logical drive, C:, D:, E:, etc., that the Master Boot Record (MBR) or Globally Unique Identifier (GUID) Partition Table, which are like maps of the partitioned and formatted memory storage locations on the hard drive, can then use to write and read stored data.

Creation of such a partition, if previous partitions are not preserved, destroys the "map" of underlying data and data locations when the partition is formatted. The previous partition data is then only recoverable by forensic techniques, and is vulnerable to complete destruction if overwritten by data stored according to the new partition "map." Note that in the before image above, each disk partition (Labeled "volX," e.g. "vol1," for "volume") is identified together with the addresses of the beginning block and ending block for each volume.

By comparing the images, it is evident that the disk was re-partitioned, reformatted, and the previous data map completely destroyed by overwriting it with new data, rendering the prior data (mostly) unrecoverable.

Forensic examination of the system can reveal remnants of deleted data. When a computer deletes a file, it does not erase the data; it merely changes the first character of the filename to a non-printable character recognized by software that accesses the disk. This first character tells the operating system to no longer display the file as it is marked as a deleted file, and the space occupied by the disk is marked as reusable.

Each block on the disk is the smallest unit of disk space that can be used. The size of all blocks on the disk are determined when the disk is formatted. The smallest disk block size in common use is 512 bytes. Even if a file only occupies 50 bytes of disk space, the entire 512 byte block is marked as "in use".

If a file of 500 bytes is written to the disk, it occupies one block of disk space, with the last 12 bytes (on a newly formatted disk) each containing the numeric value zero (0). If this file is then deleted, and a file of 50 bytes is written to the same disk block, the first 50 bytes of the block contain the new file, and the next remaining 450 bytes of the disk block contain the data from the deleted file that previously occupied the disk block (followed by the 12 null (0) bytes of data). This data remnant is referred to as "File Slack Space" and is defined as any previous remnant data that remains on the disk and is not accessible via the operating system nor allocated as an accessible file.

Special forensic software is required to access file slack space, and the data it contains are partial remnants of previous system data. This data may be of use in forensic investigation, and forensic tools often identify it. File Slack is identified here for clarity and better understanding of these data.

Figure 8 - EMS Server (5.11-CO) Web Server Log Files Before

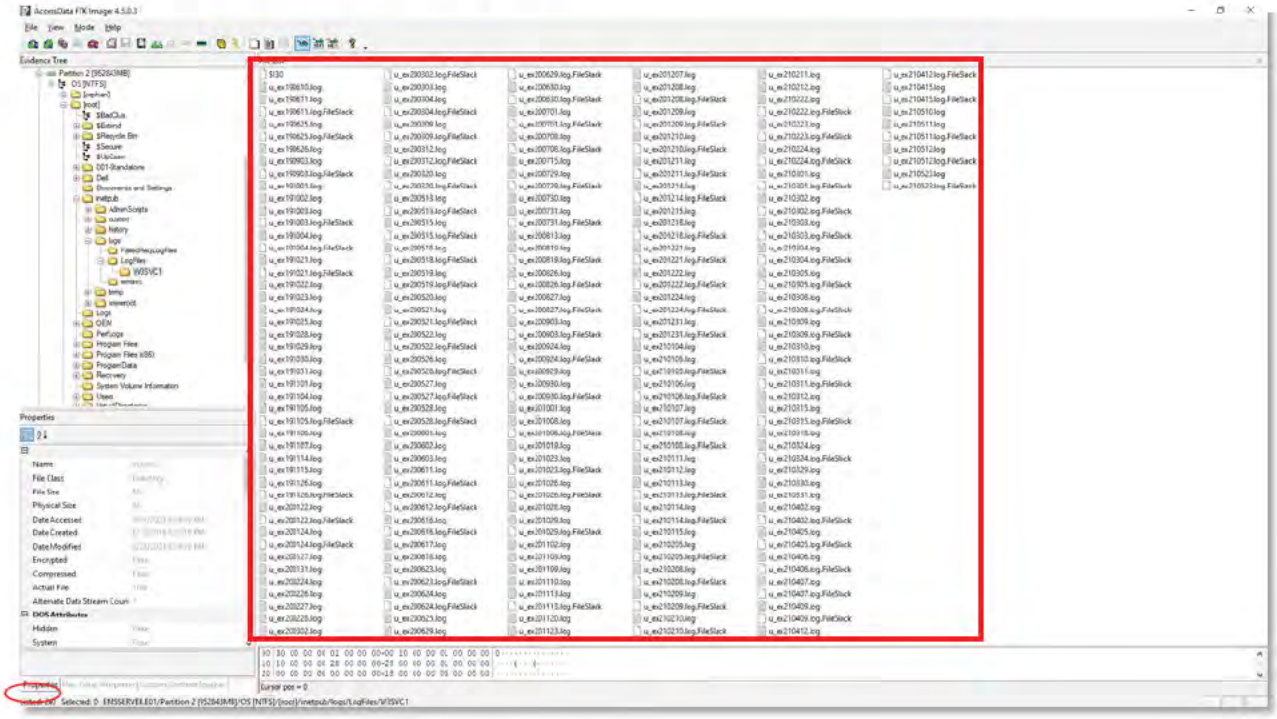
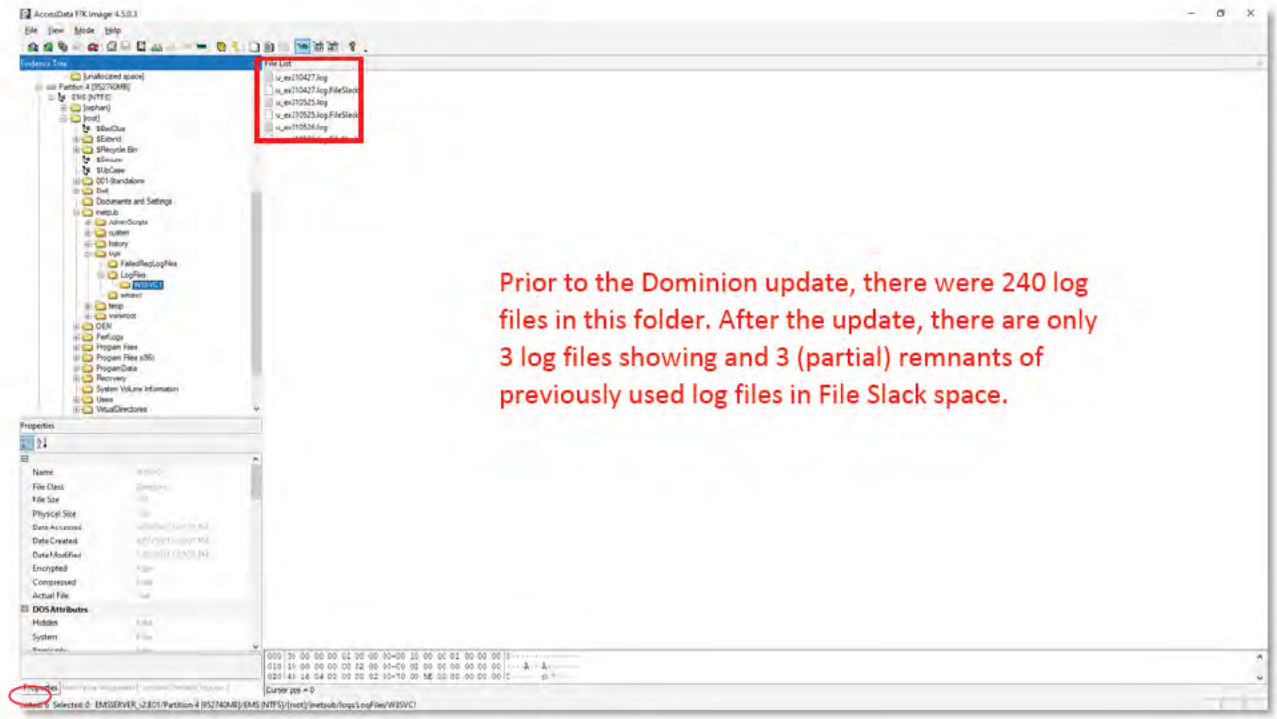


Figure 9 - EMS Server (5.13) Web Server Log Files After



A web server provides information to external web clients (via "web browser" software) using the HyperText Transfer Protocol (HTTP). This information can include both read and write access to databases and static presentation of information.

Some software system designs utilize an Ethernet network interface that is essentially an internal connection to itself, known as a *loopback* interface. Thus the presence of a Web Server, by itself, does not indicate a connection to an external ethernet interface. However, such an external connection may be indicated by the data within web server logs, which are stored by default in Microsoft operating systems with Microsoft Internet Information Services (IIS) installed, in a “logs” subfolder to the “inetpub” folder. That log data would include information regarding what web pages and data were accessed and whether it was accessed from within the server (loopback) or via an external network connection.

In these before and after views of the same web server directories, it is clear that the web server logs have been destroyed by or during the Dominion/CO Secretary of State DVS D-Suite 5.13 modification.

This log data is required to verify that the election system was not accessed by an external, unauthorized device, but due to the specific and unusual installation method for a critical computing system, chosen by Dominion Voting Systems and endorsed by the CO Secretary of State, these critical data files with election-related data have clearly been destroyed on the Mesa County EMS Standard Server.

Purpose: *The Database Management System that is used to hold actual ELECTION DATA – votes from each ballot. These log files contain information detailing the installation events of SQL Server.*

Figure 10 - EMS Server (5.11-CO) MS SQL Server Installation Log Files Before

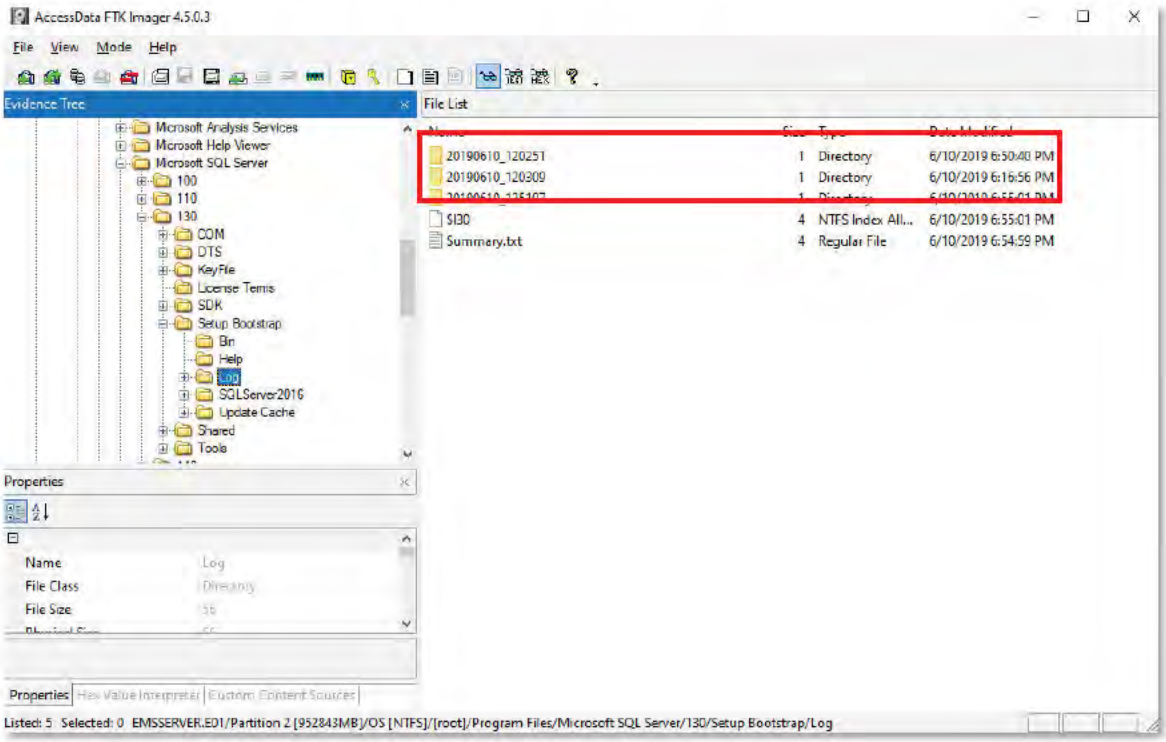
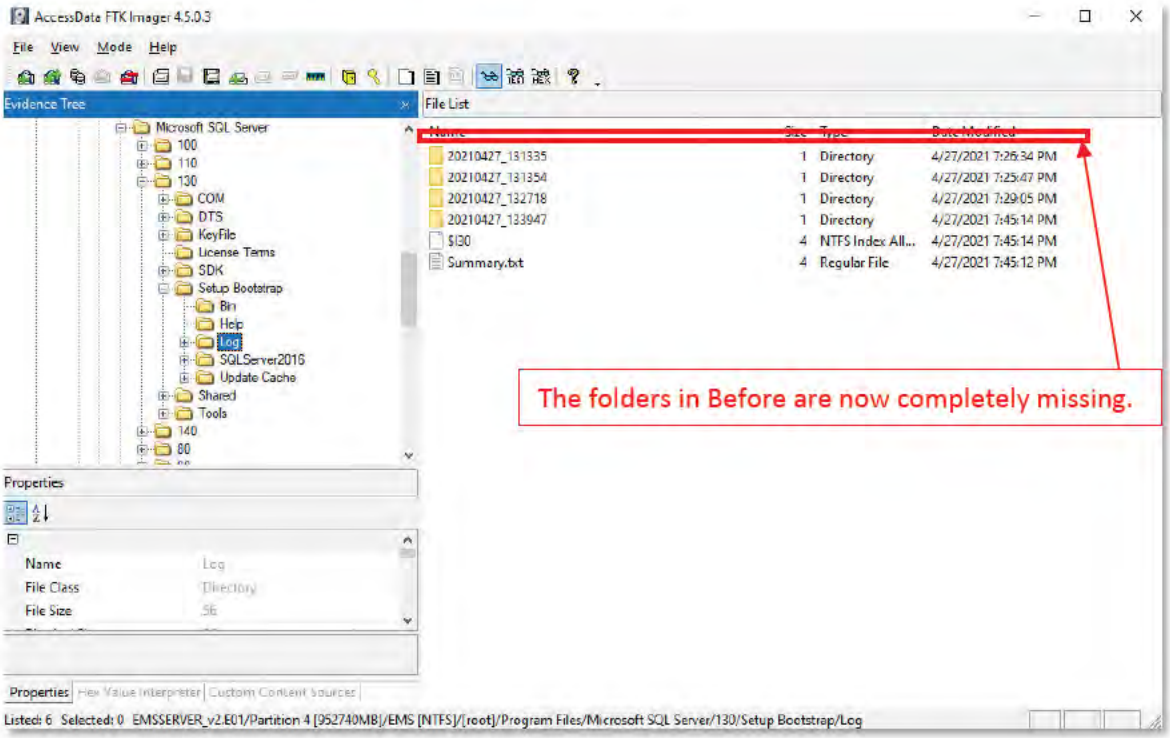


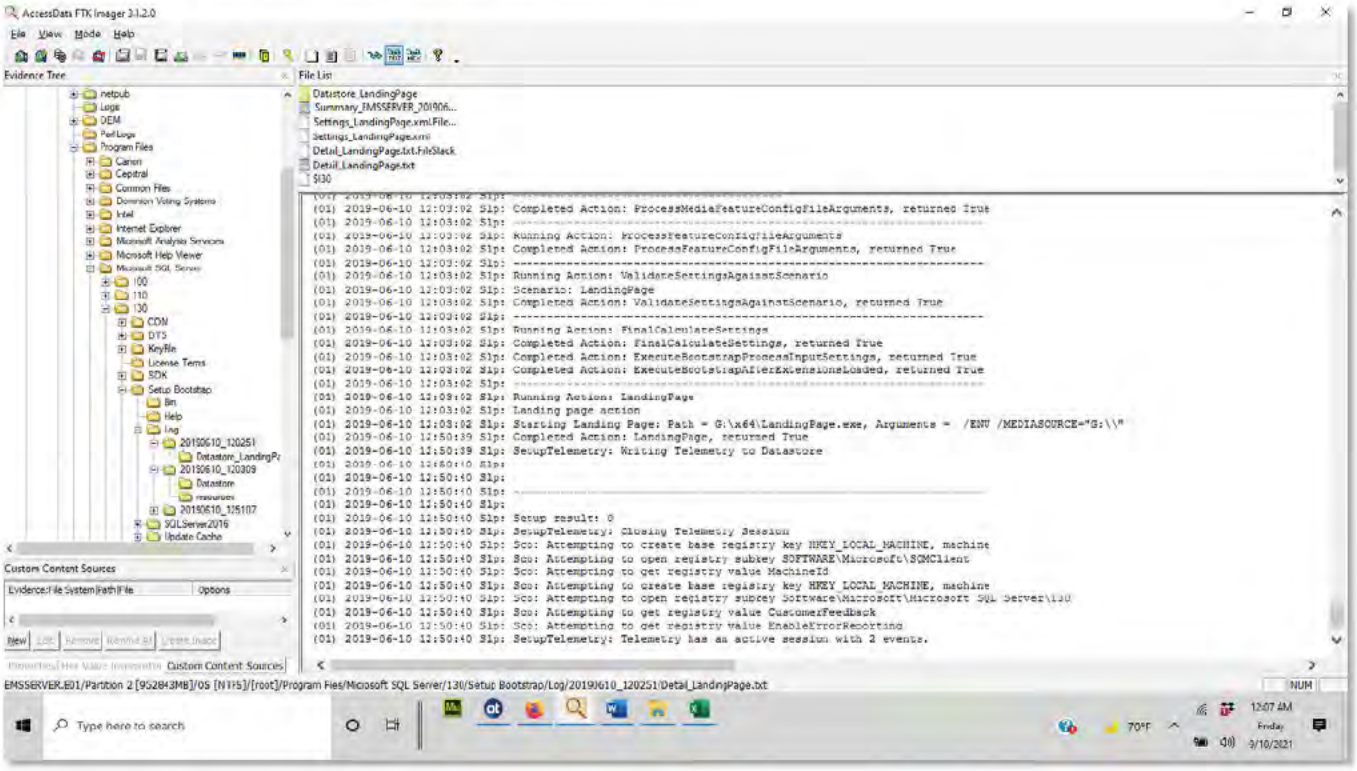
Figure 11 - EMS Server (5.13) MS SQL Server Installation Log Files After



These log files were created by installing the SQL Server Database Management System software and contain data regarding the Initial installation of the software. In a full forensic investigation, these data are part of the information that investigators require to determine a baseline from which can be determined what changes were made, by whom, when the changes were made, and much more on a system with properly configured log recording. Therefore, these data are Election Related as they document not only the configuration but its changes and are relevant to the Integrity of the election.

Figure 12 is an example of log content from the initial software setup. It tells us what (Microsoft) software executes, where data is stored (the G: drive), and it shows us what Registry values have been set during the installation. These are valuable should an investigation of an illegal computer intrusion occur, as they provide a record of the initial configuration during such an investigation.

Figure 12 - Example of Log File Content from EMS Server (5.11-CO) Before



Purpose: *These log files keep track of events that occur within the SQL Server that manages the election databases.*

Figure 13 - EMS Server (5.11-CO) SQL Server Log Files Before

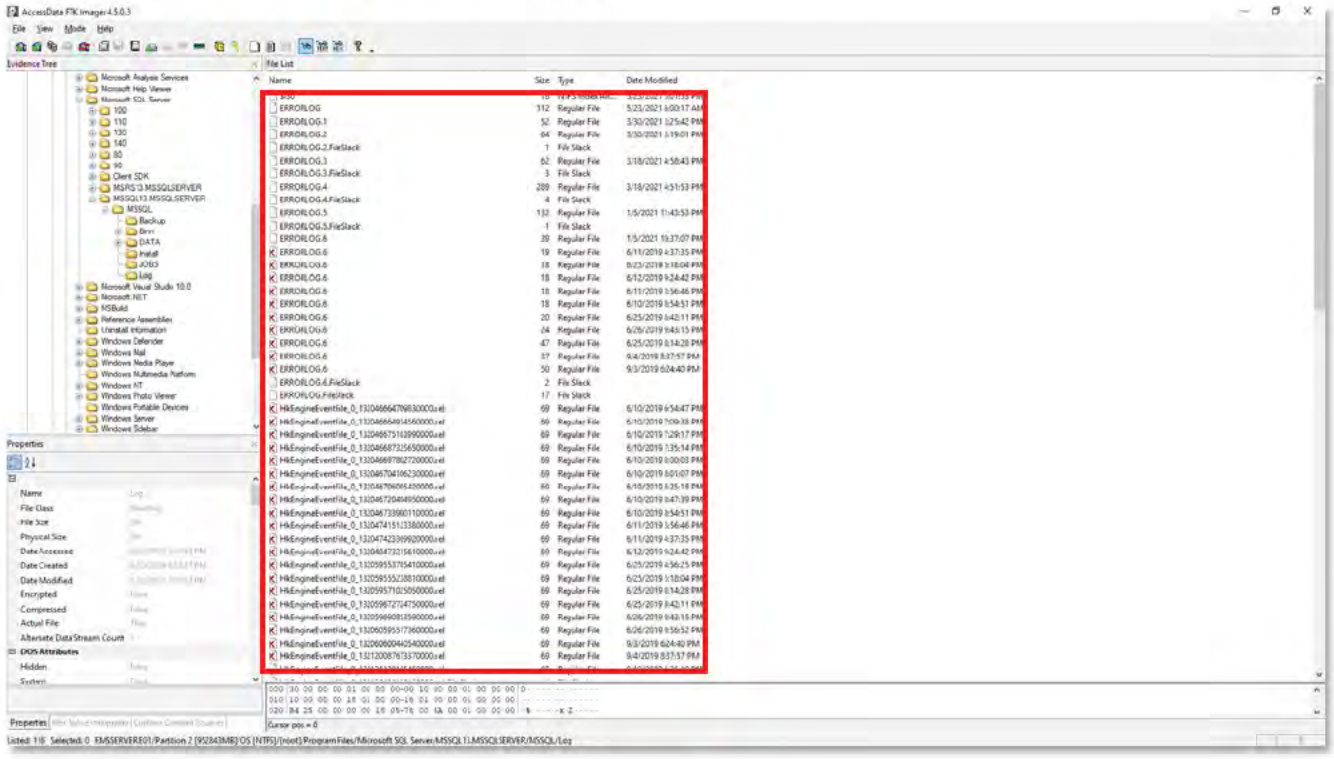
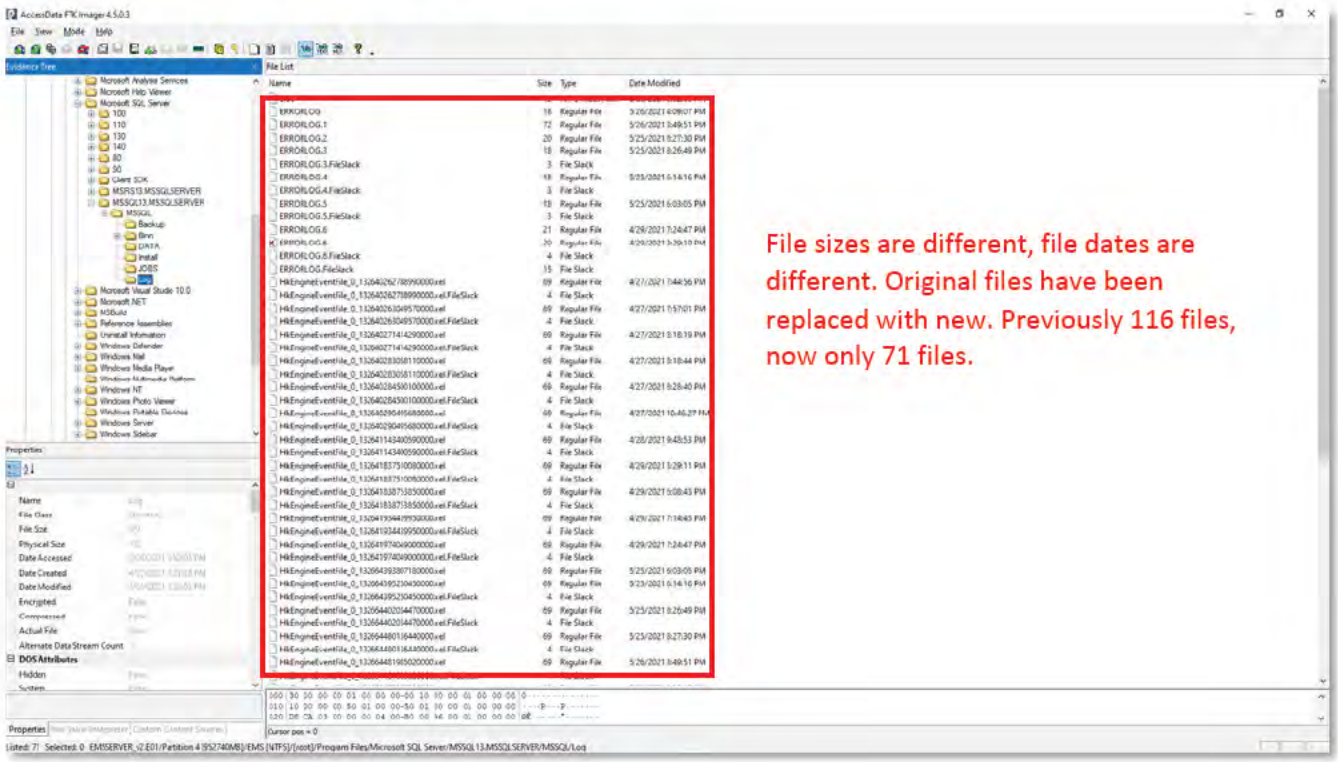


Figure 14 - EMS Server (5.13) SQL Server Log Files After



Purpose: *These log files track installation of updates made to the various components of the servers, including updates to software for a remote-access card.*

Figure 15 - EMS Server (5.11-CO) Dell Server Update Files Before

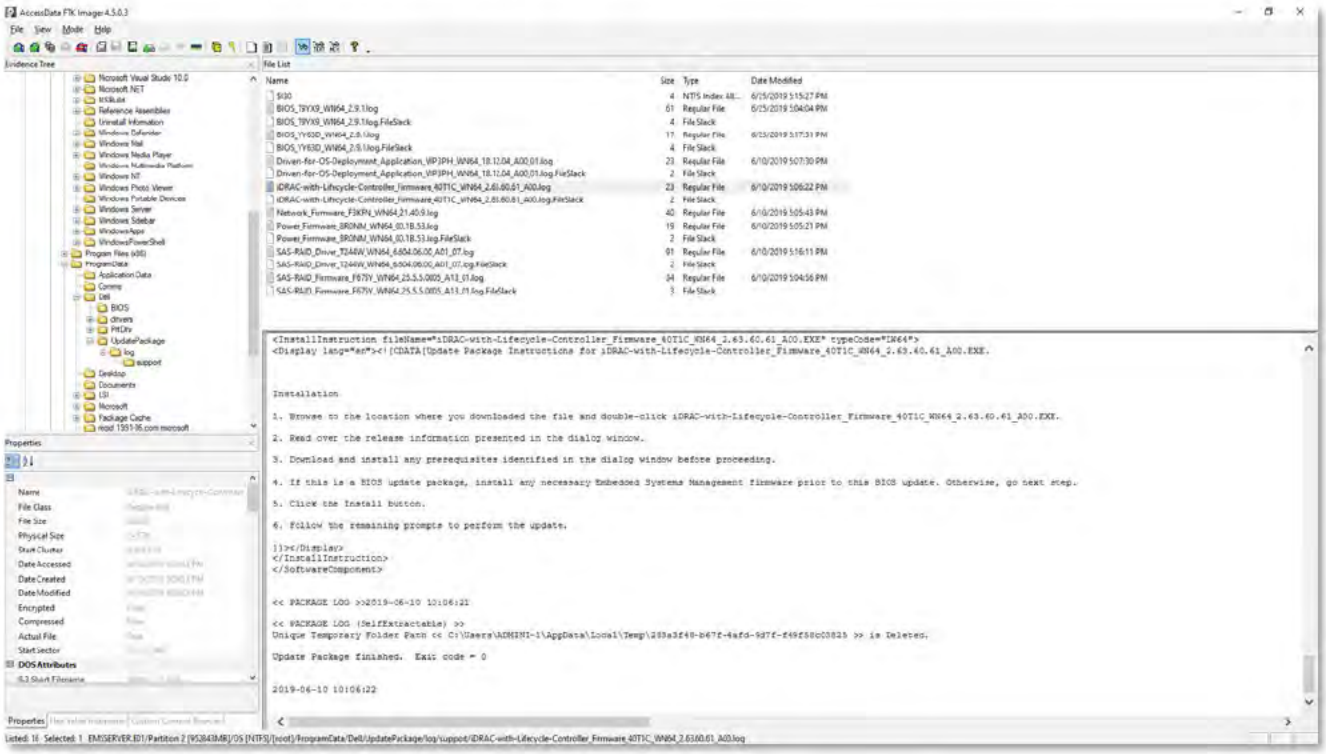
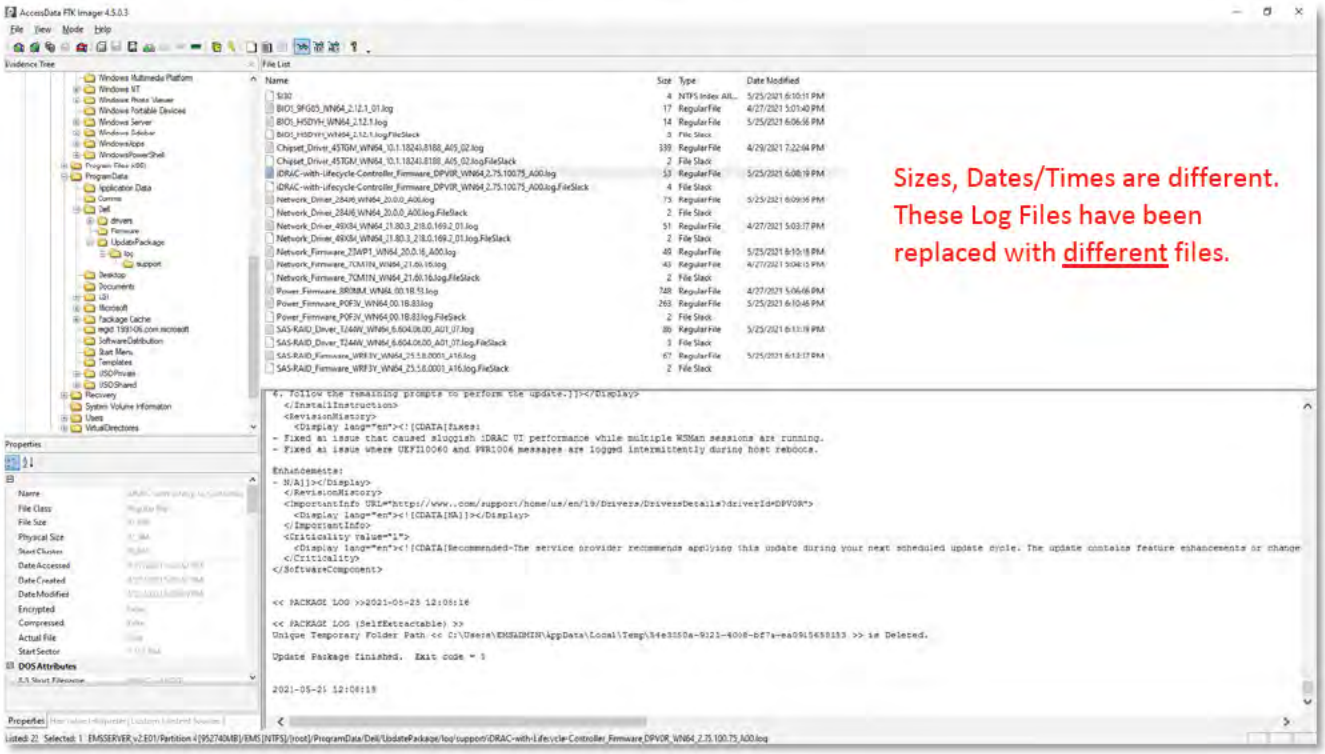


Figure 16 - EMS Server (5.13) Dell Server Update Files After



Several log files of great importance to an investigation are shown in Figure 16. The SAS RAID firmware and drivers logs tell us about the functionality of hard disk controllers (RAID is an acronym for Redundant Array of Independent Disks) and about this storage redundancy's physical capability. Network Firmware logs tell us which hardware devices were updated with new firmware, and the version allows us to trace back to its network (and possibly Internet) functionality. The application of iDRAC controller firmware may indicate the presence of a special hardware controller intended to permit complete remote control of the computer system. This iDRAC controller is often used when a data center must be located an inconvenient distance away from its owner and/or operators, or for example, when such a computer might be physically located at an Internet Service Provider's secure data center. The iDRAC controller permits a remote user to remotely turn on the power to the server, reboot it, access administrative control functions, and make changes to the server, *OUTSIDE THE CONTROL, or even the awareness, of the local computer operator and its operating system.* Among the changes possible via an IDRAC are changes to the BIOS (Basic I/O System) including those firmware settings that include the computer Clock, boot device order, which disks or other data storage devices are used to boot the computer, and some other computer capabilities.

Take note of what files remain following the update.

Not only are the files in an entirely different directory, but the file modification dates have changed, and more importantly, these logs are for DIFFERENT versions of the software, and the previous logs have been overwritten.

Physical examination of the EMS computer system is required to verify the presence or absence of an IDRAC controller, however it is highly irregular for update software to install updates to software for a hardware device that has not been installed.

Purpose: *These log files store information about websites visited, files opened, etc.*

Figure 17 - EMS Server (5.11-CO) Administrator WebCache Log Files Before

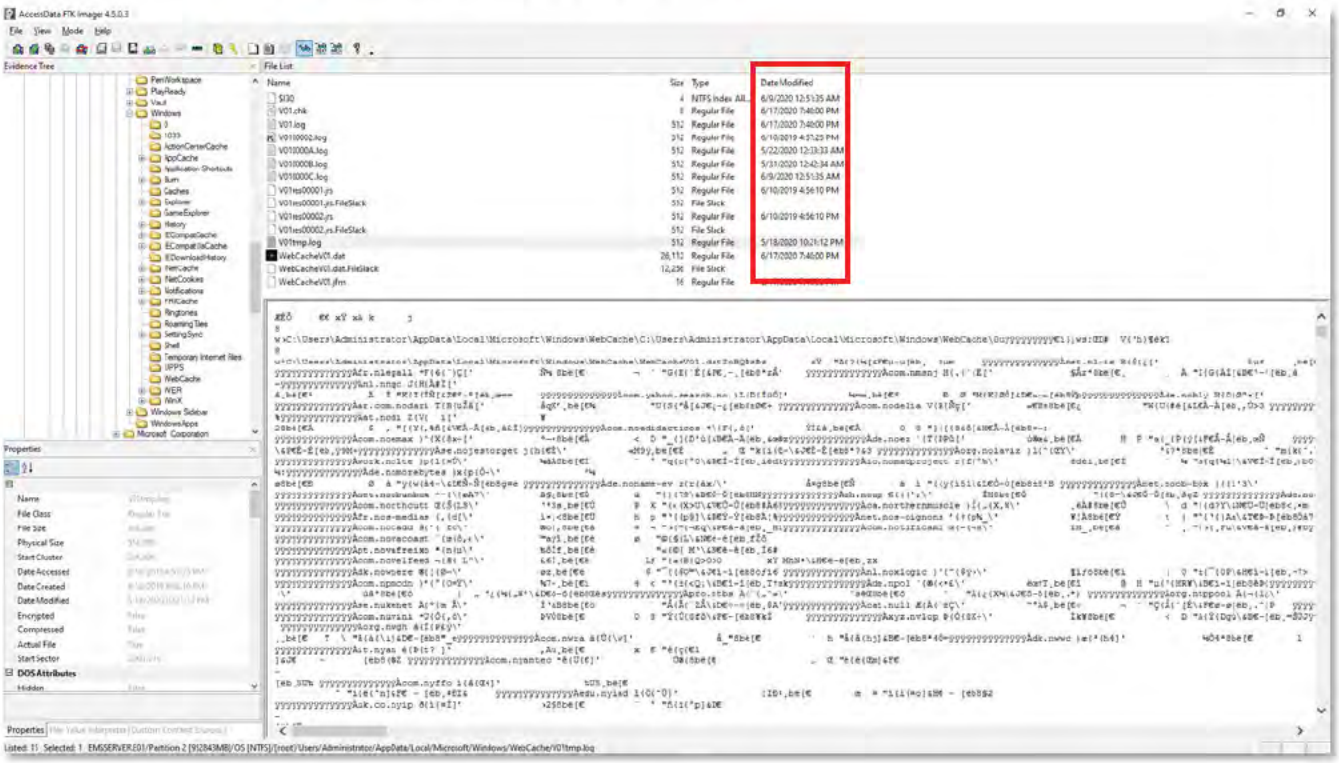
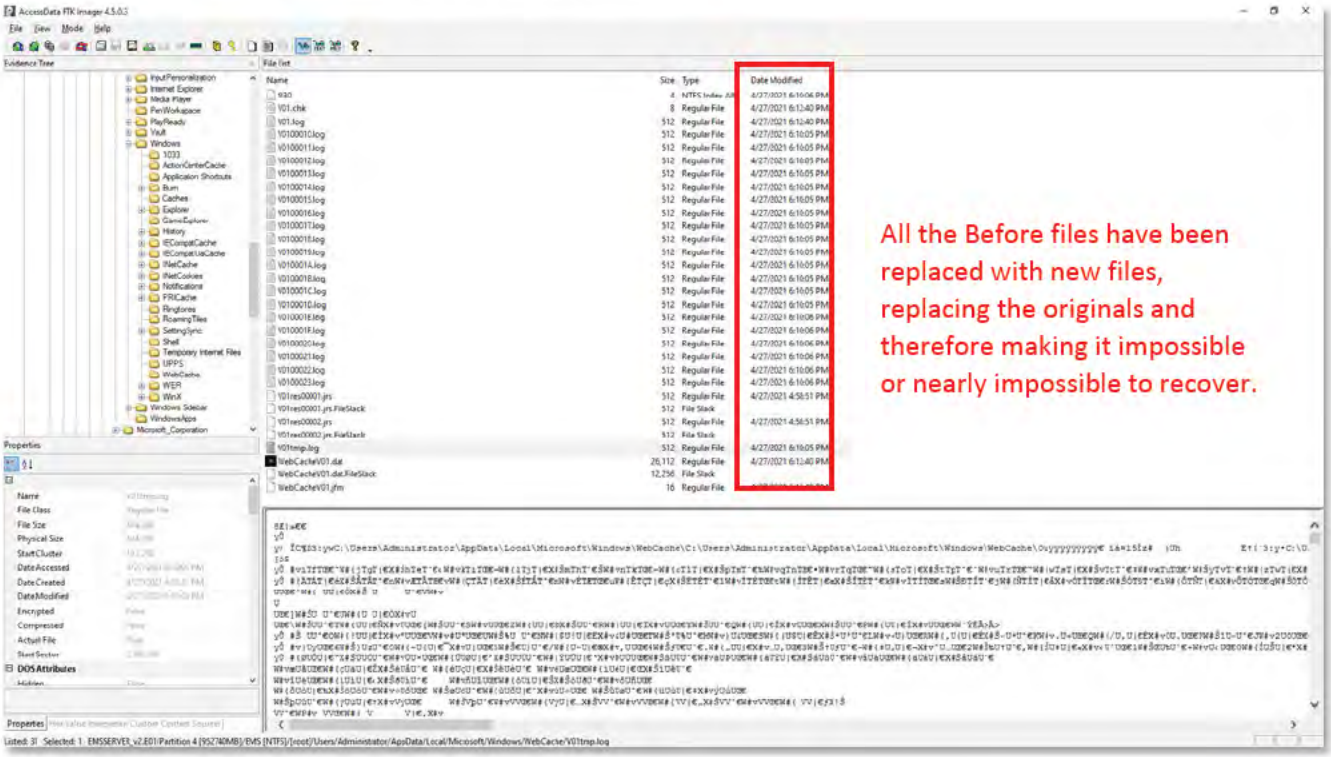


Figure 18 - EMS Server (5.13) Administrator WebCache Log Files After



Purpose: *These log files store information about websites visited, files opened, etc.*

Figure 19 - EMS Server (5.11-CO) "emsadmin" WebCache Log Files Before

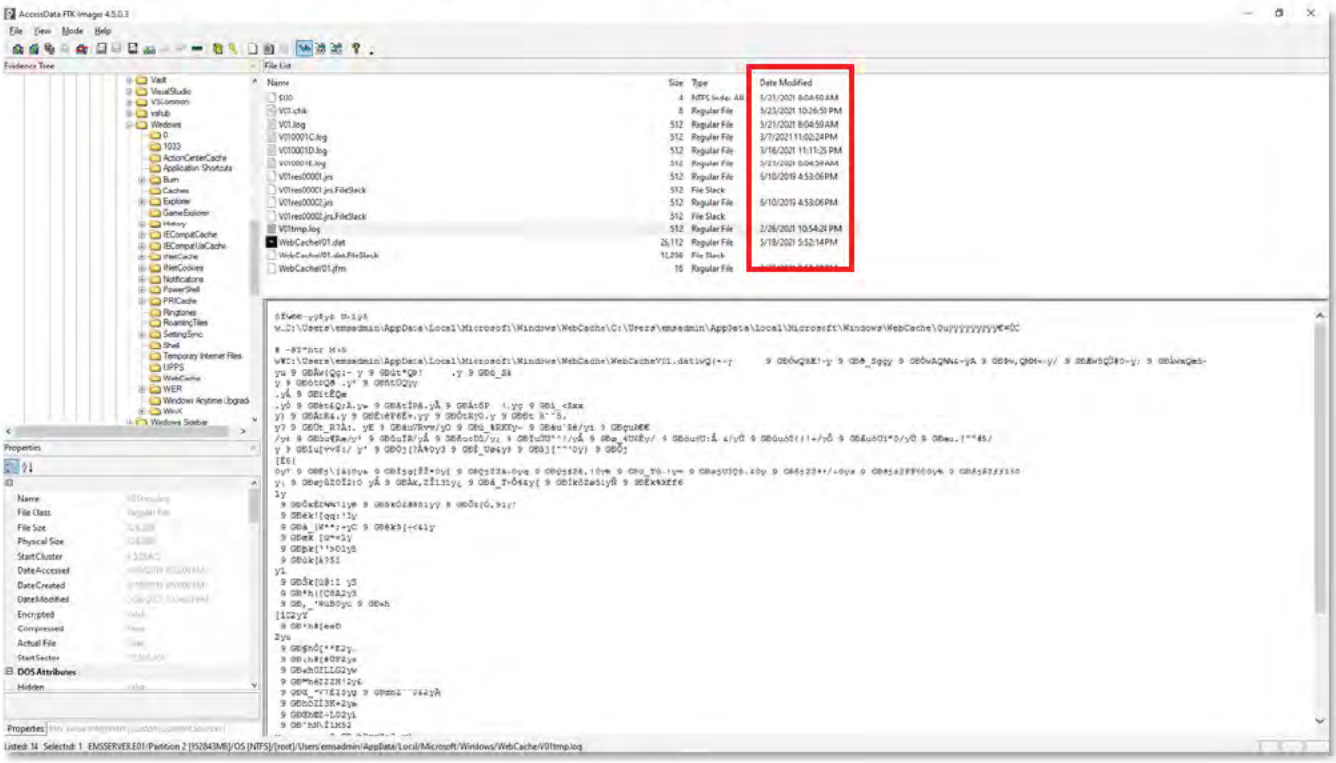
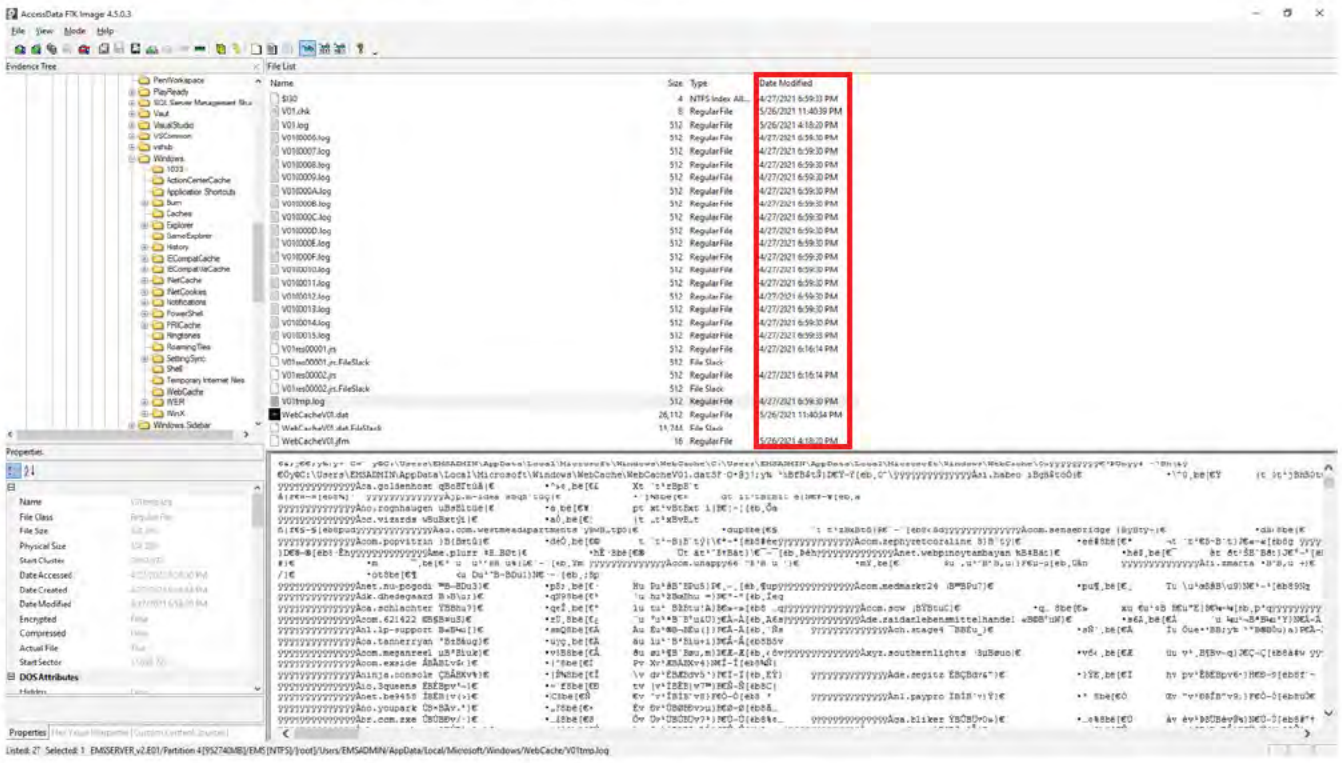


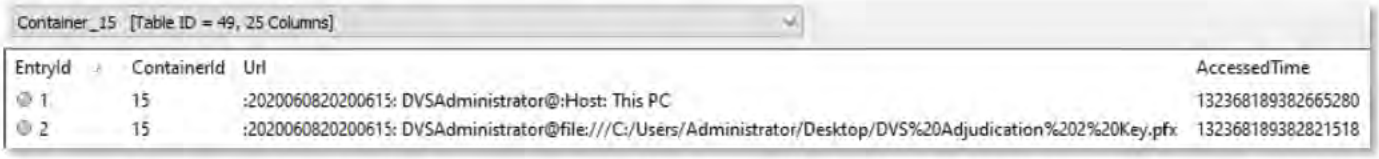
Figure 20 -EMS Server (5.13) "emsadmin" WebCache Log Files After



The WebCache log files have been overwritten. IF the computer has been used on the Internet or with ANY webserver (even one on the local network, including this computer's OWN webserver), these WebCache files indicate the connections that were sought, as well as files that were opened. These may provide *critical* evidence that the system has been connected to a network, including networks that have access to the Internet. THESE ARE NOT the same files in the before and after images. They have been deleted and replaced.

Here is a small subset of some of the information that was found on the Before image in these WebCache log files:

Figure 21 - EMS Server (5.11-CO) Webcache Log File Content Before

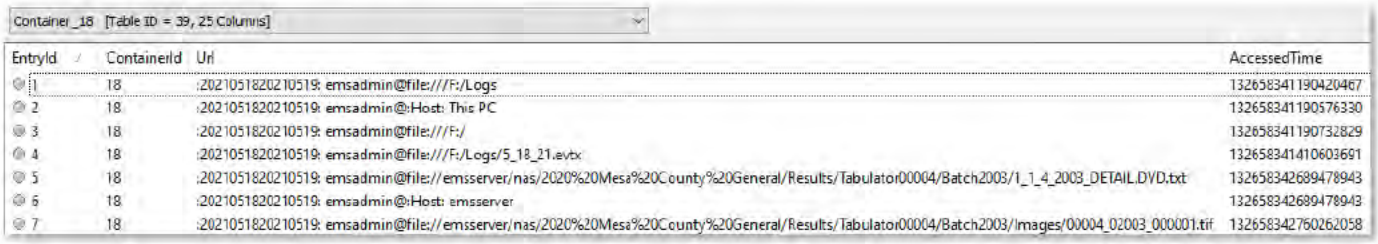


Container_15 [Table ID = 49, 25 Columns]

EntryId	ContainerId	Url	AccessedTime
1	15	:2020060820200615: DVSAdministrator@:Host: This PC	132368189382665280
2	15	:2020060820200615: DVSAdministrator@file:///C:/Users/Administrator/Desktop/DVS%20Adjudication%202%20Key.pfx	132368189382821518

For instance, the above log file entry seems to show a DVS Adjudication Encryption Key was accessed, where it was stored and accessed from, and when it was accessed.

Figure 22 - EMS SErver (5.11-CO) Webcache Log File Content Before - II



Container_18 [Table ID = 39, 25 Columns]

EntryId	ContainerId	Url	AccessedTime
1	18	:2021051820210519: emsadmin@file:///F:/Logs	132658341190420467
2	18	:2021051820210519: emsadmin@:Host: This PC	132658341190576330
3	18	:2021051820210519: emsadmin@file:///F:/	132658341190732829
4	18	:2021051820210519: emsadmin@file:///F:/Logs/5_18_21.evtx	132658341410603691
5	18	:2021051820210519: emsadmin@file://emsserver/nas/2020%20Mesa%20County%20General/Results/Tabulator00004/Batch2003/1_1_4_2003_DETAIL.DVD.txt	132658342689478943
6	18	:2021051820210519: emsadmin@:Host: emsserver	132658342689478943
7	18	:2021051820210519: emsadmin@file://emsserver/nas/2020%20Mesa%20County%20General/Results/Tabulator00004/Batch2003/Images/00004_02003_000001.tif	132658342760262058

In addition, the above log file entry seems to show several interesting files (a windows 'evtx' log file being opened from an external attached USB flash drive, and a ballot detail file and even a ballot image from Batch 2003 being opened from a Network Attached Storage device)

Without a forensic Before image prior to a Dominion 'Update', this type of potentially critically-important forensic information could be, and likely would be, lost forever.

Purpose: *These log files track the installation of the SQL Server Management Studio, which is used to get into the back-end of the election databases.*

Figure 23 - EMS Server (5.11-CO) SSMS Log Files Before

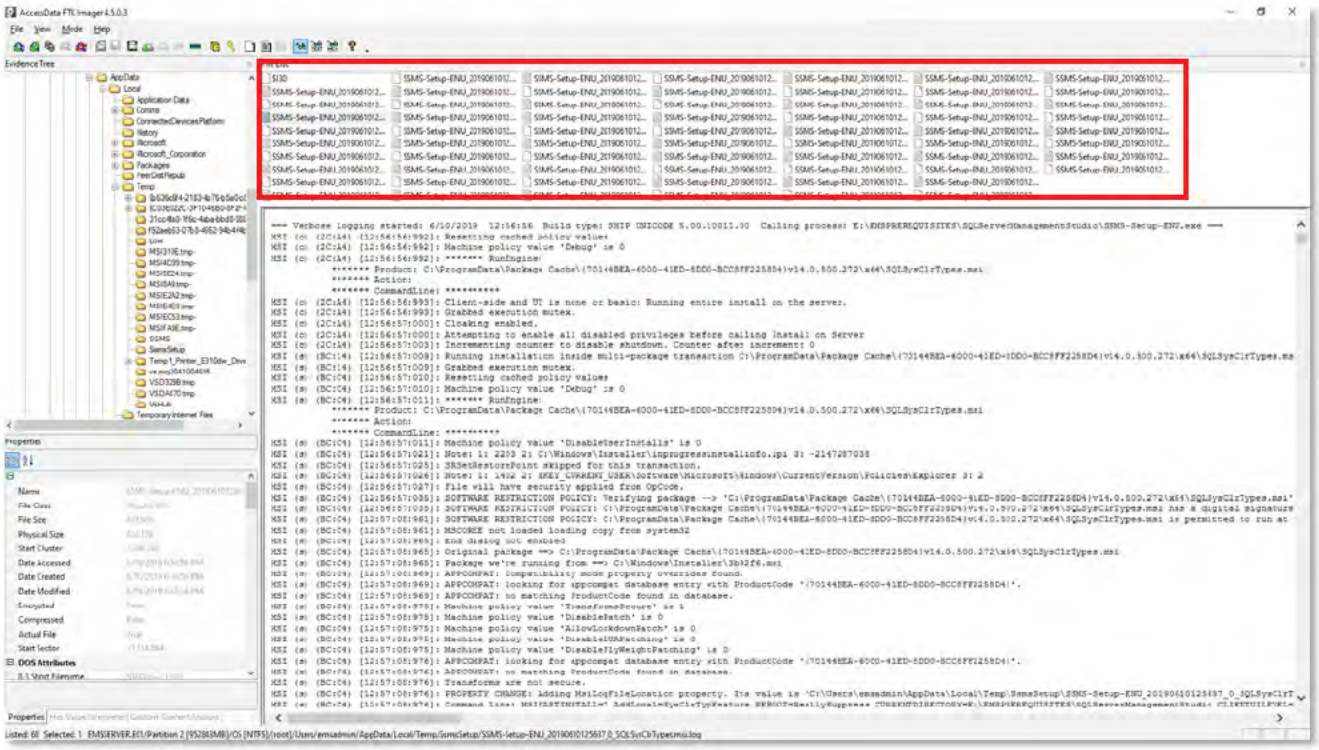
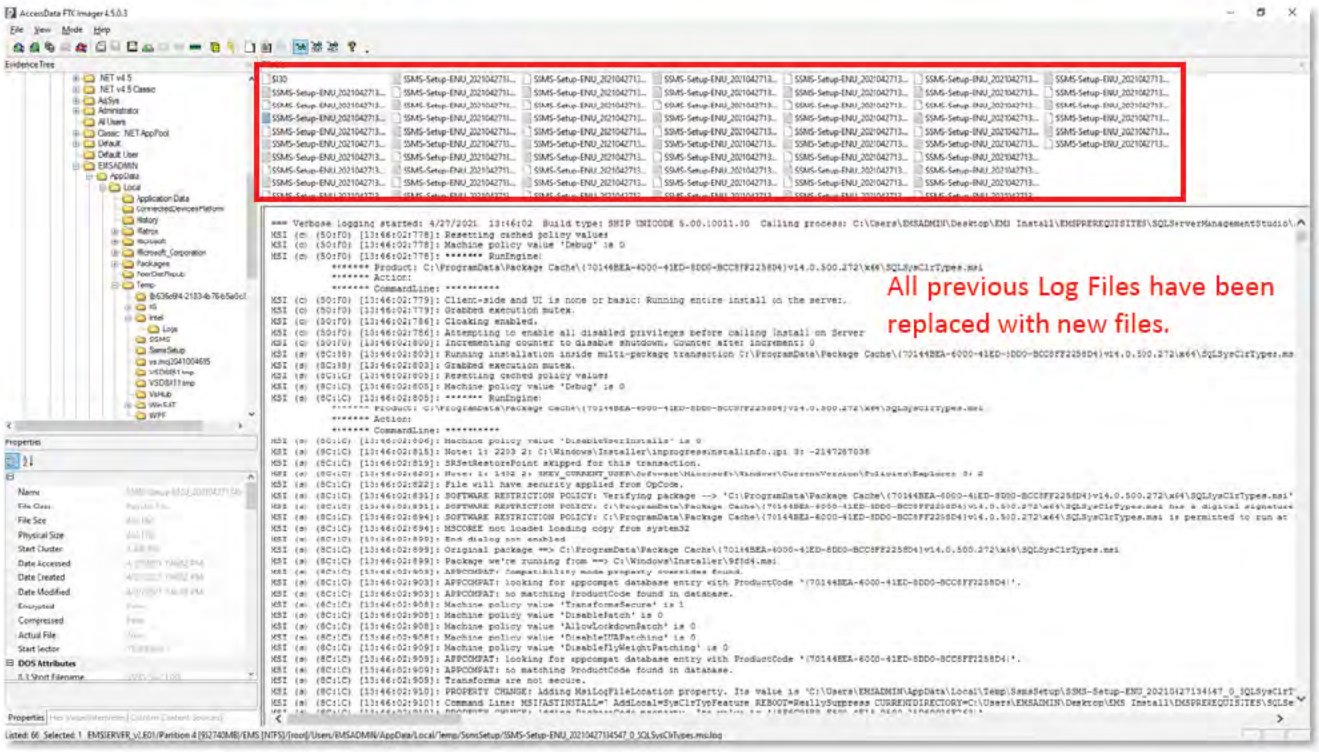


Figure 24 - EMS Server (5.13) SSMS Log Files After



Purpose: ***These Log Files contain detailed information about installed updates. They could contain evidence of changes to the server that would cause decertification of the system.***

Figure 25 - EMS Server (5.11-CO) CBS Log Files Before

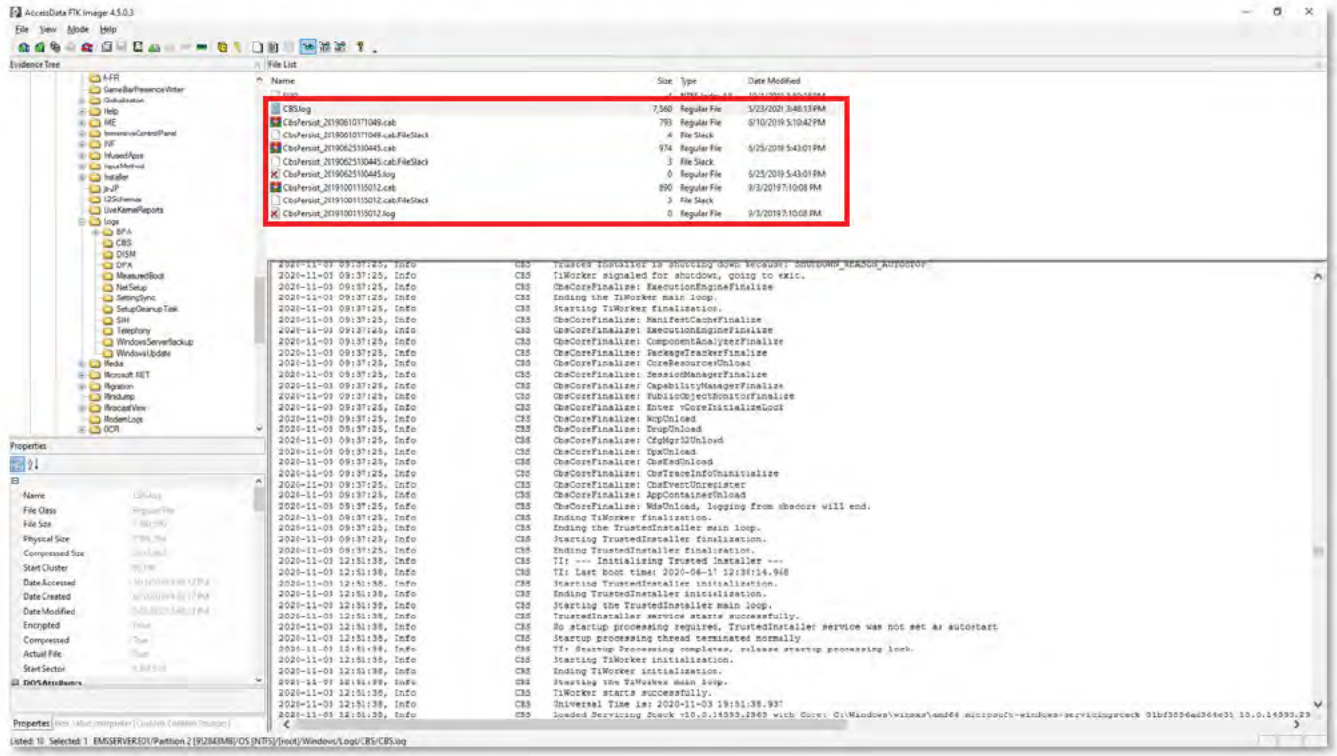
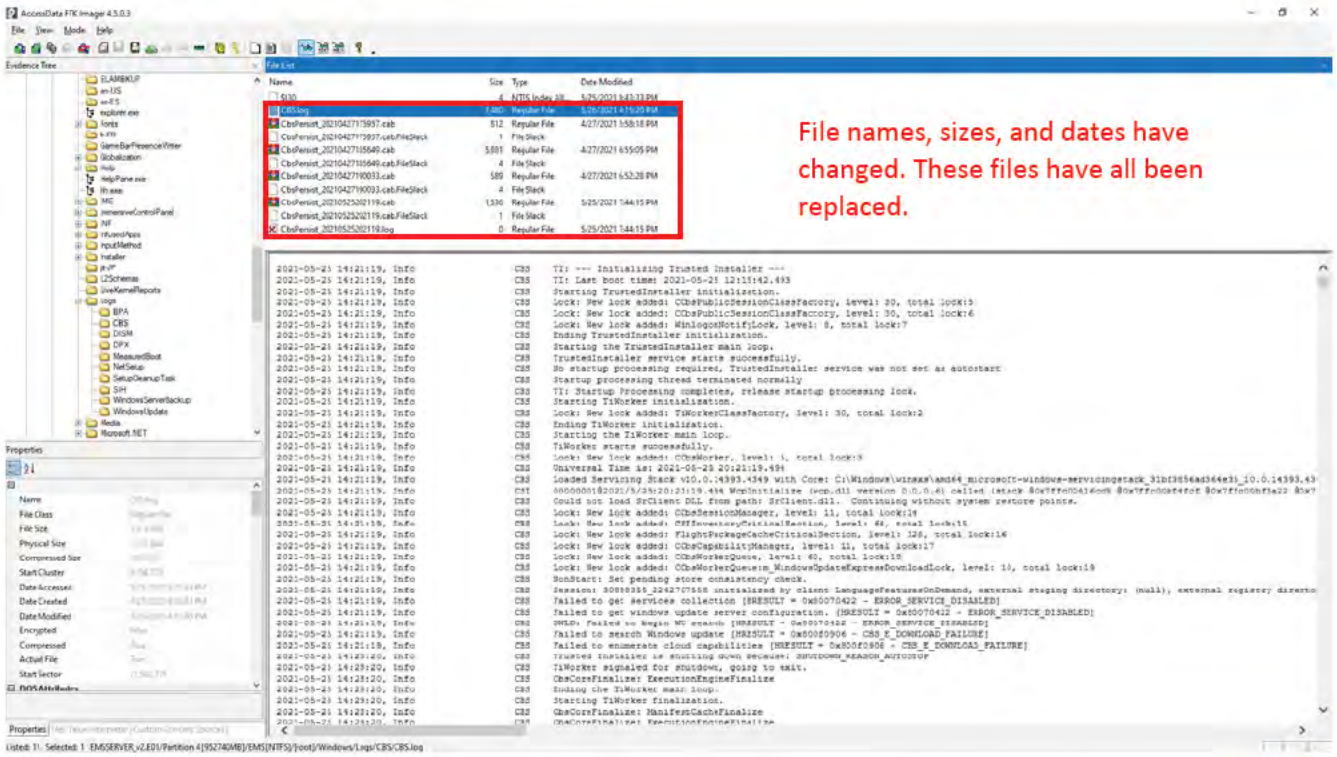


Figure 26 - EMS Server (5.13) CBS Log Files After



Purpose: *This folder holds all the databases (votes, information regarding batches, when they were processed, how many were processed, who they were processed by, and much more). There are also multiple extra databases that contain information regarding ballot adjudication.*

Figure 27 - EMS Server (5.11-CO) Election Databases Before

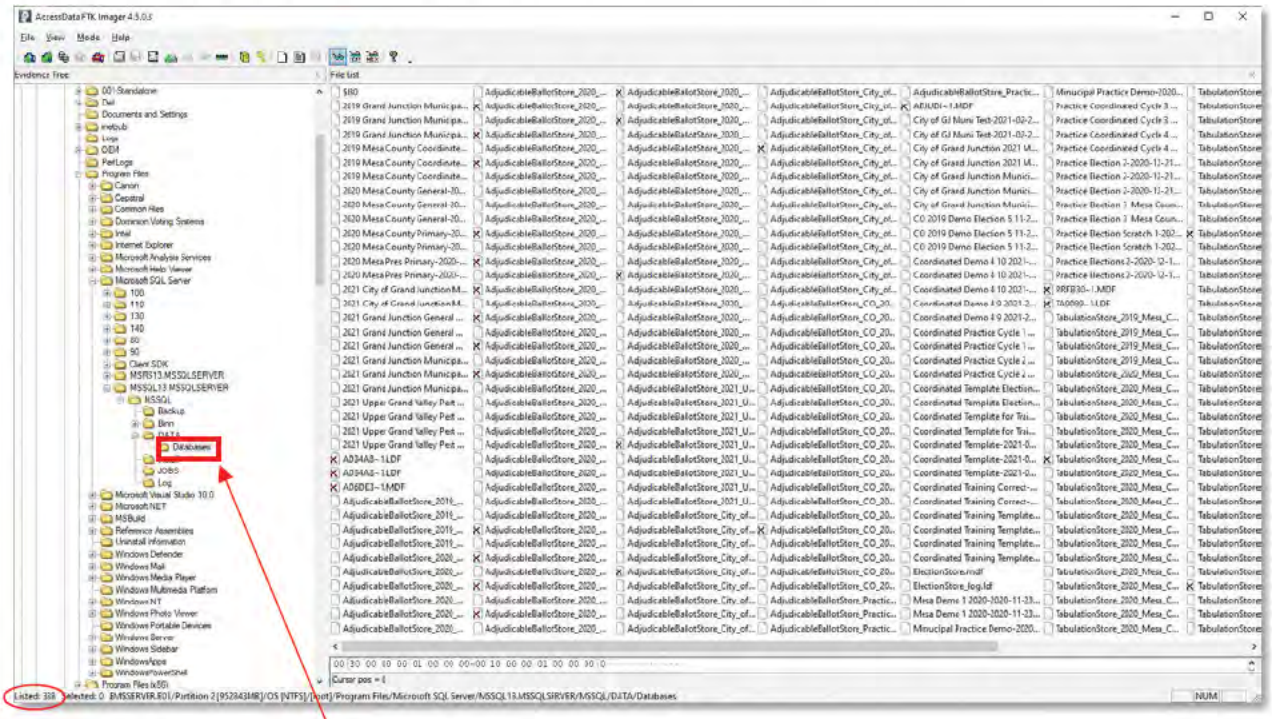
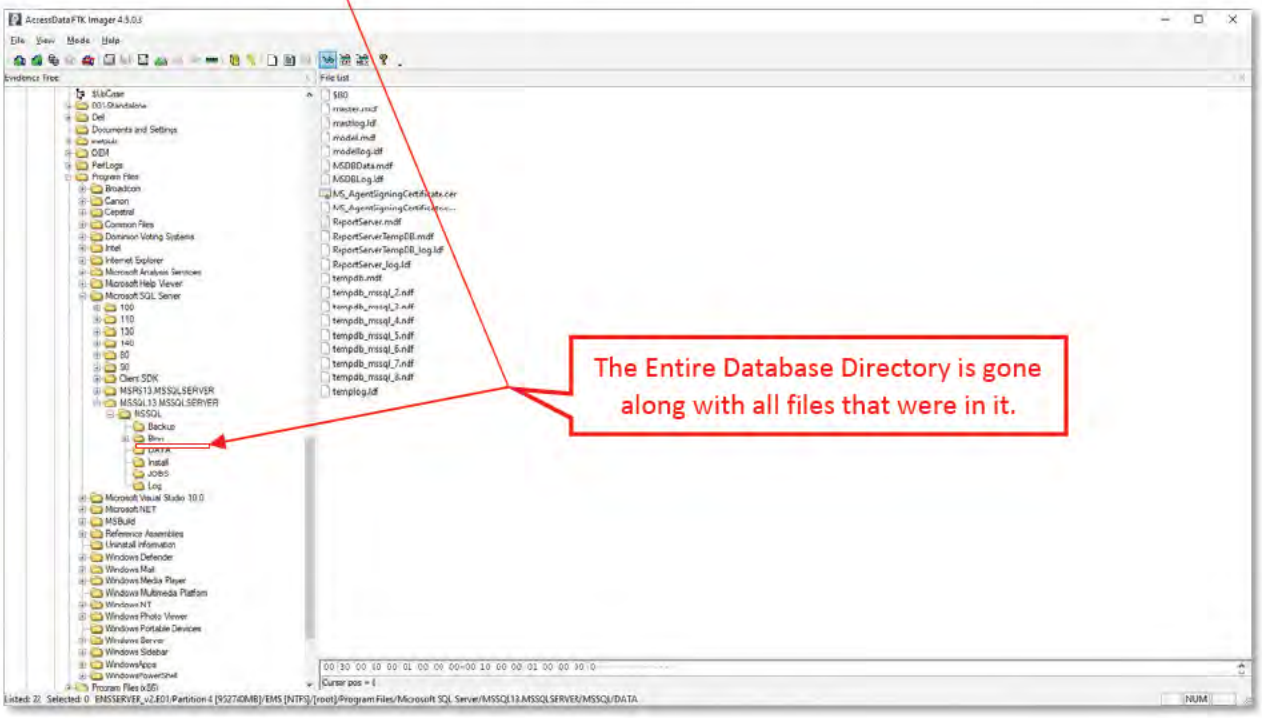


Figure 28 - EMS Server (5.13) Election Databases After



Purpose: *DHCP Log Files can show evidence regarding computers or other devices being connected to the network.*

Figure 29 - EMS Server (5.11-CO) DHCP Log Files Before

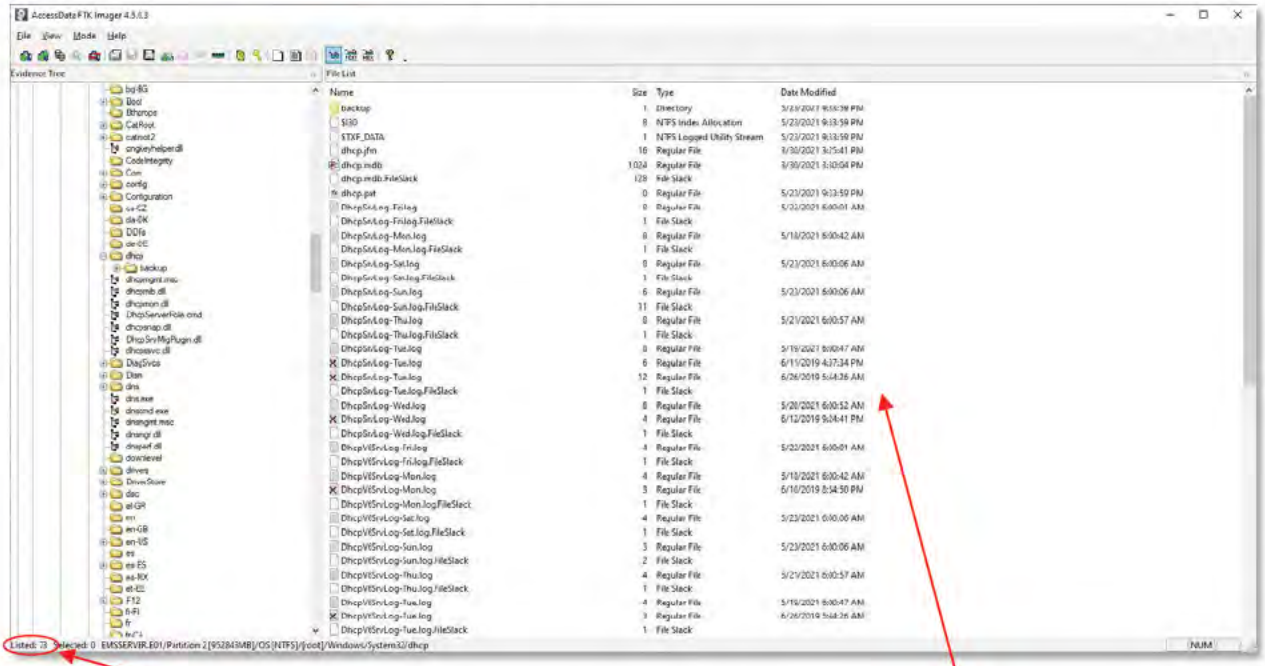
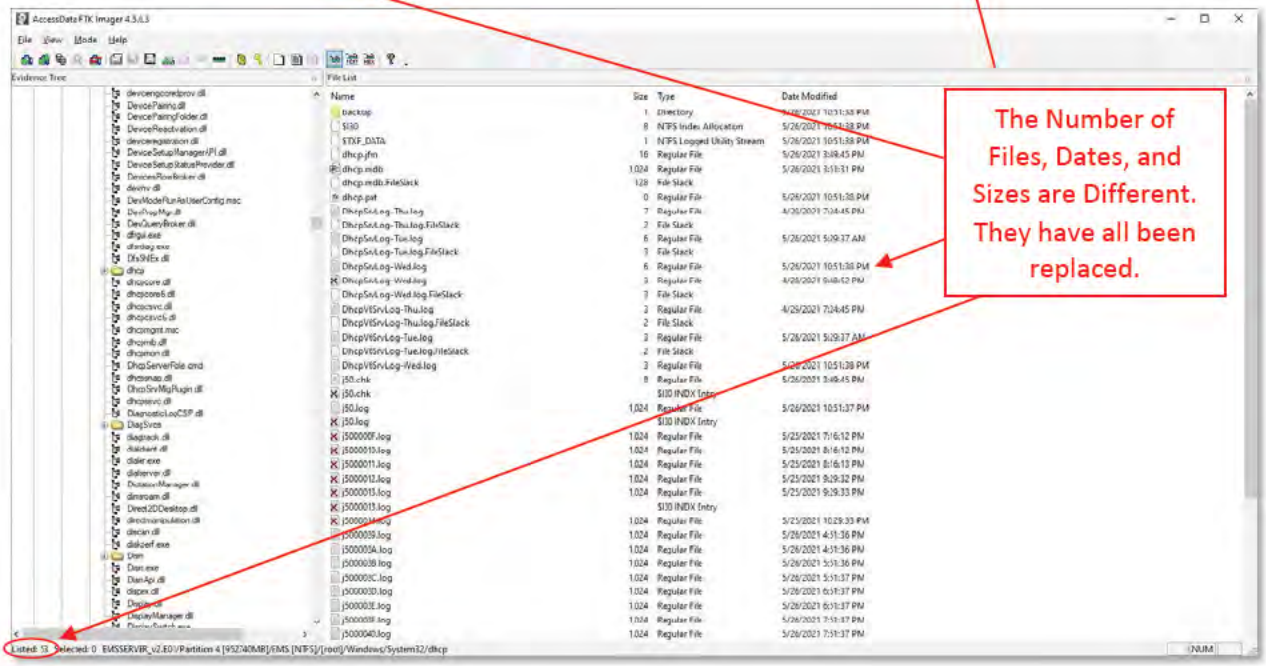


Figure 30 - EMS Server (5.13) DHCP Log Files After



Purpose: *These Dominion Log Files keep track of election/project-related activity. The Windows Server event logs outside the red box keep track of much of the activity on the server.*

Figure 31 - EMS Server (5.11-CO) Event Logs Before

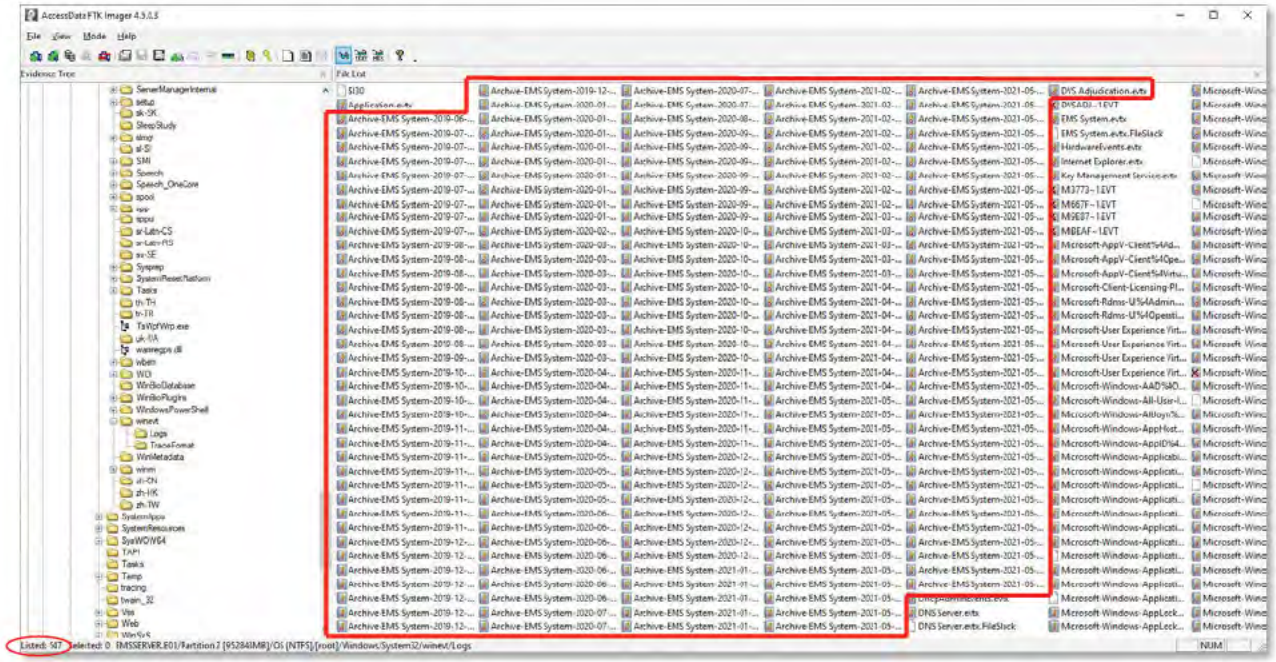
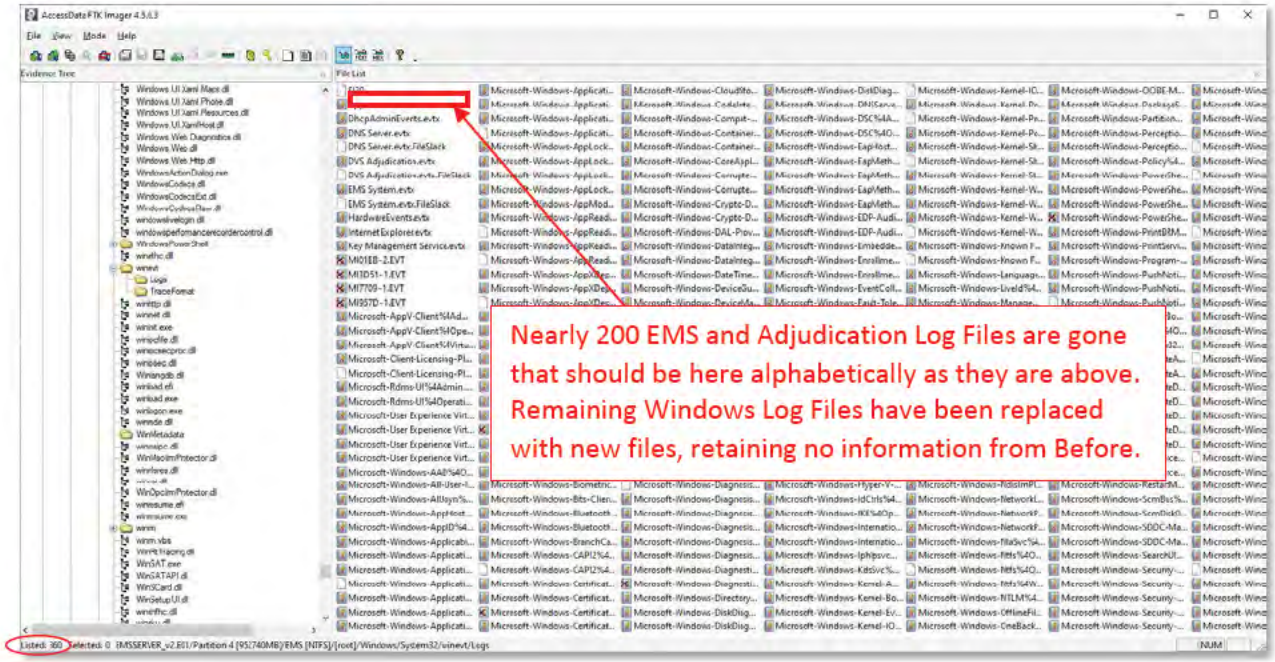
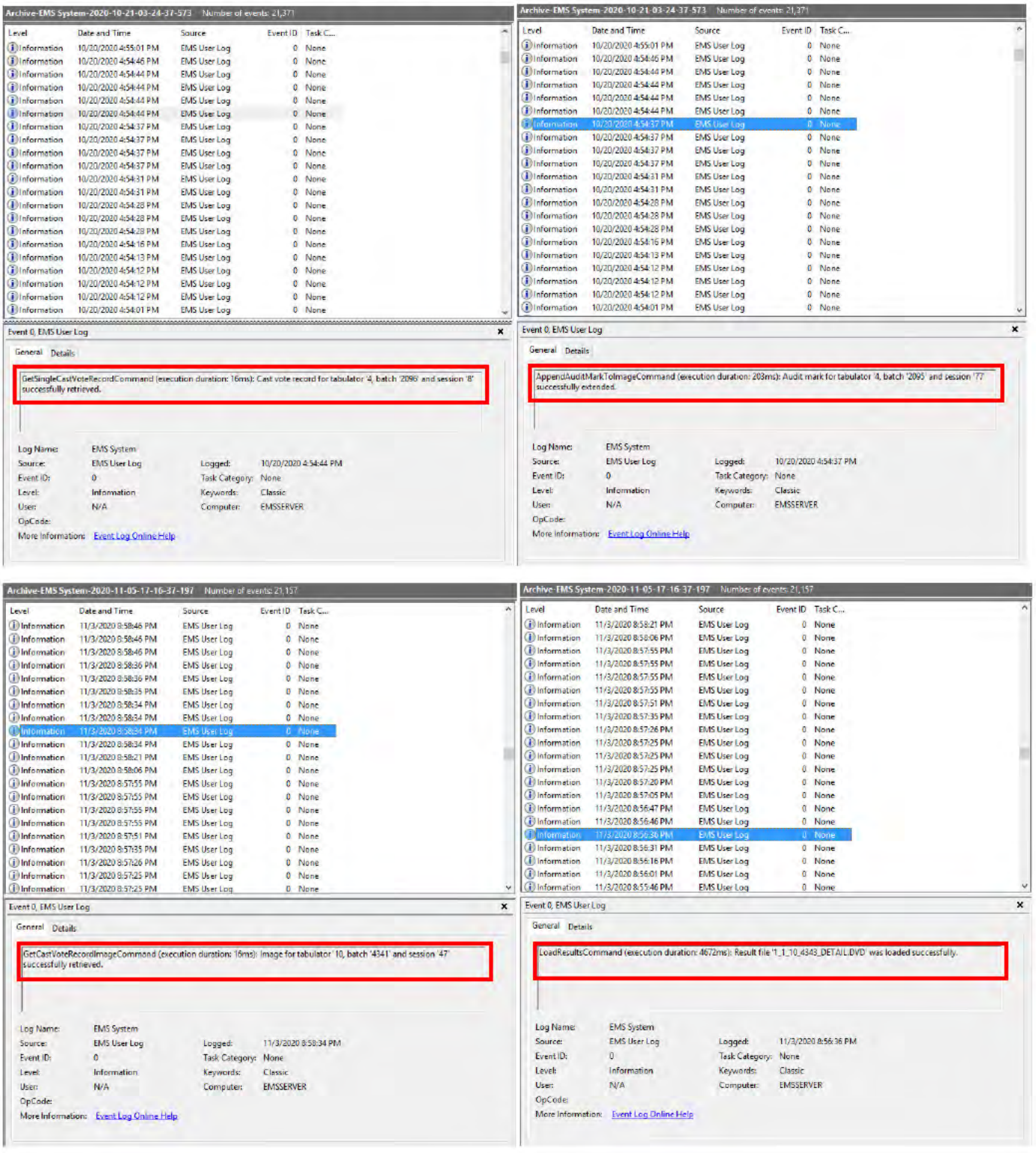


Figure 32 - EMS Server (5.13) Event Logs After



Below are some screen shots of the kind of Election-Related information (such as cast vote records, audit marks, image retrievals, result file loads, etc.) in the EMS Archive Logs that are missing After the Dominion Update:

Figure 33 - Examples of Election Data Missing After Update



Purpose: *These folders store the information for each user account on the server.*

Figure 34 - EMS Server (5.11-CO) System Users Before

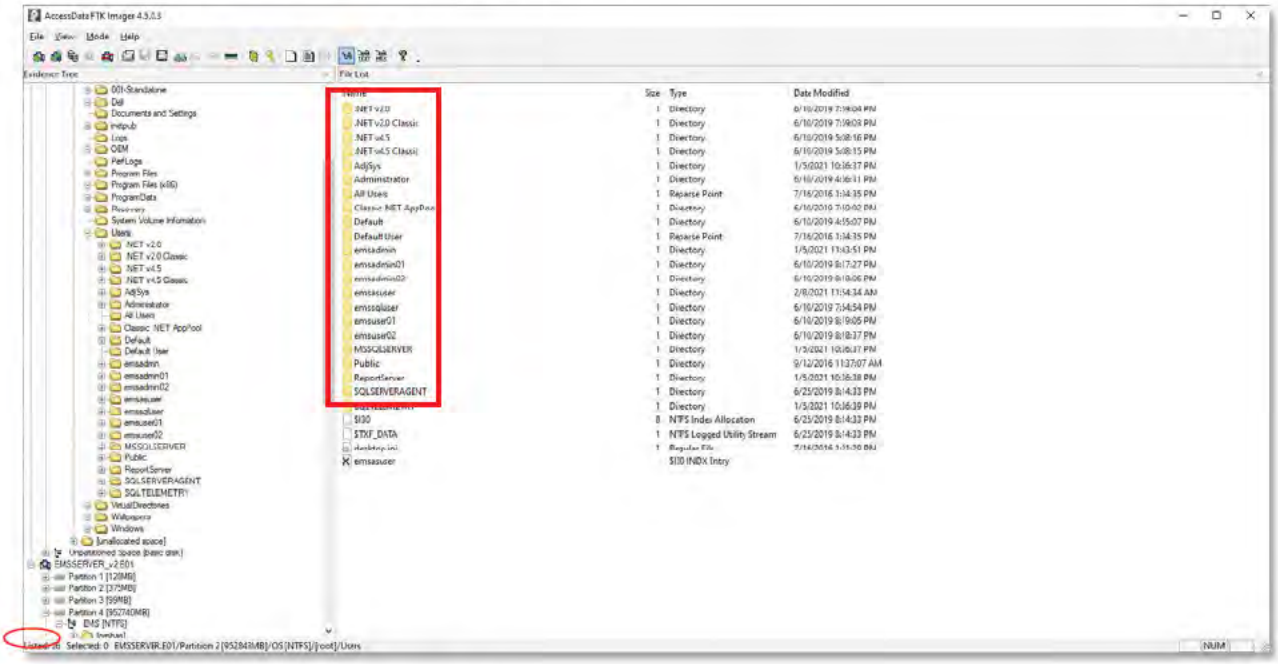
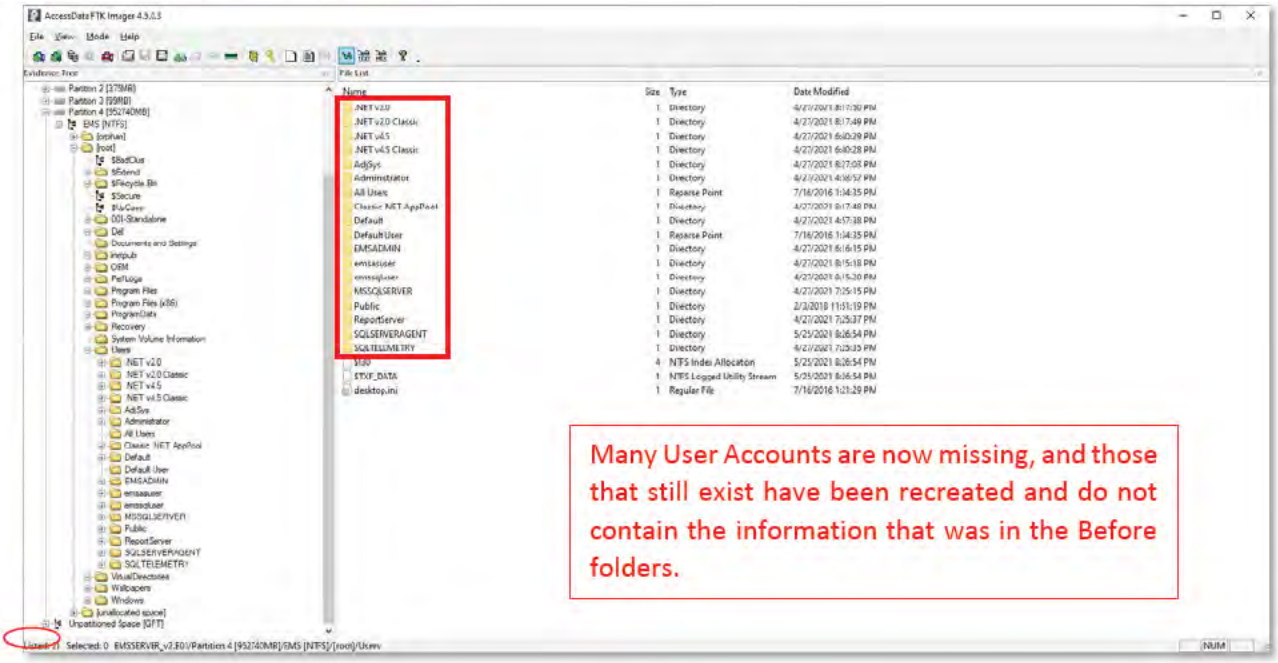


Figure 35 - EMS Server (5.13) System Users After



Purpose: *These are the Log Files that contain information, warnings, and errors relating to the Website Server as the server processes election projects that have been set up.*

Figure 36 - EMS Server (5.11-CO) Virtual Directory Log Files Before

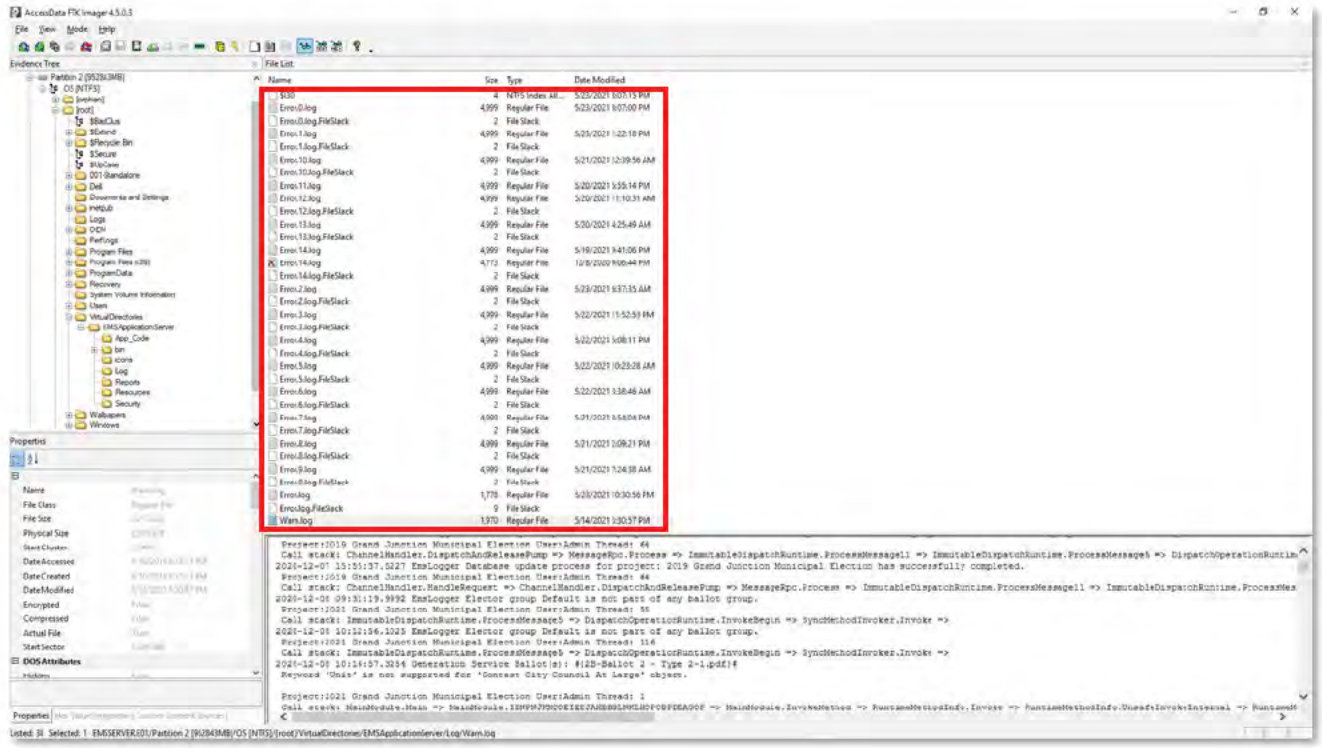
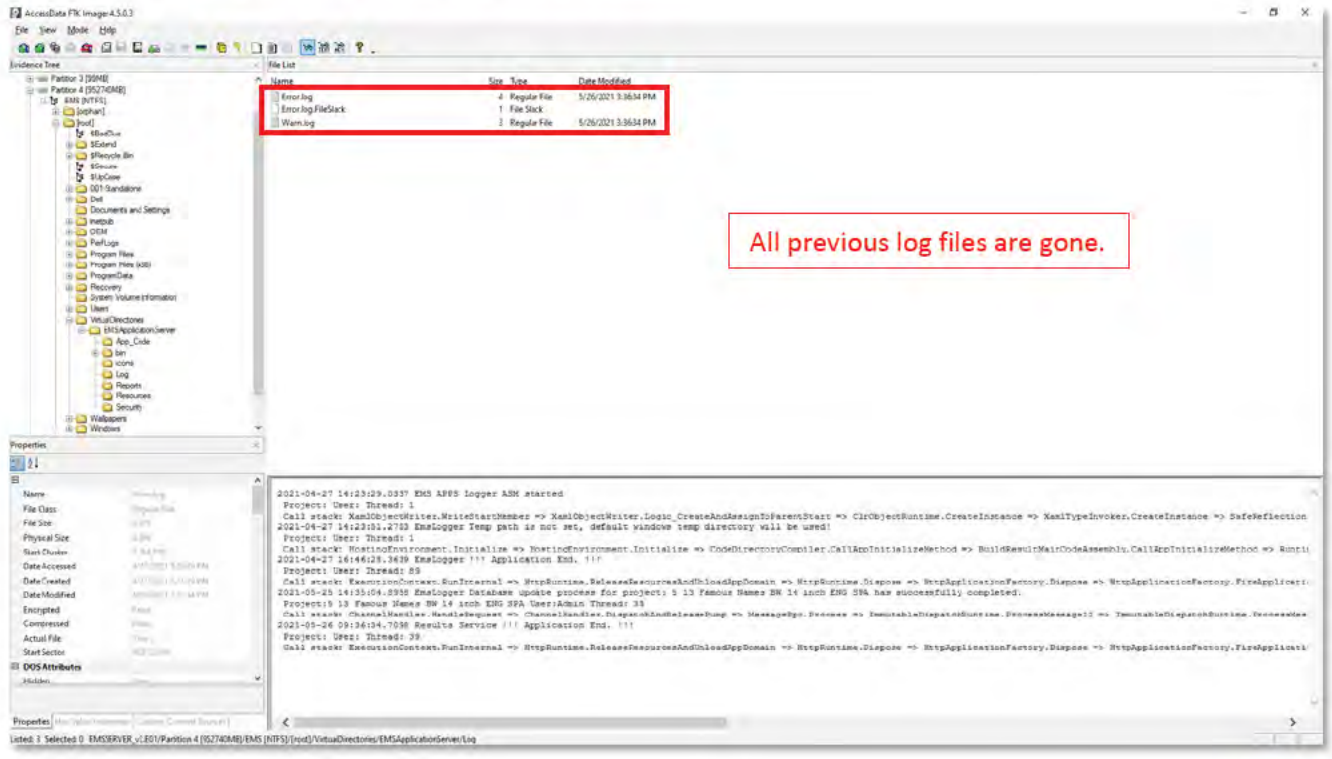


Figure 37 - EMS Server (5.13) Virtual Directory Log Files After



Purpose: *These log files keep track of the activity of the built-in Anti-Virus software.*

Figure 38 - EMS Server (5.11-CO) Windows Defender Log Files Before Dominion Update:

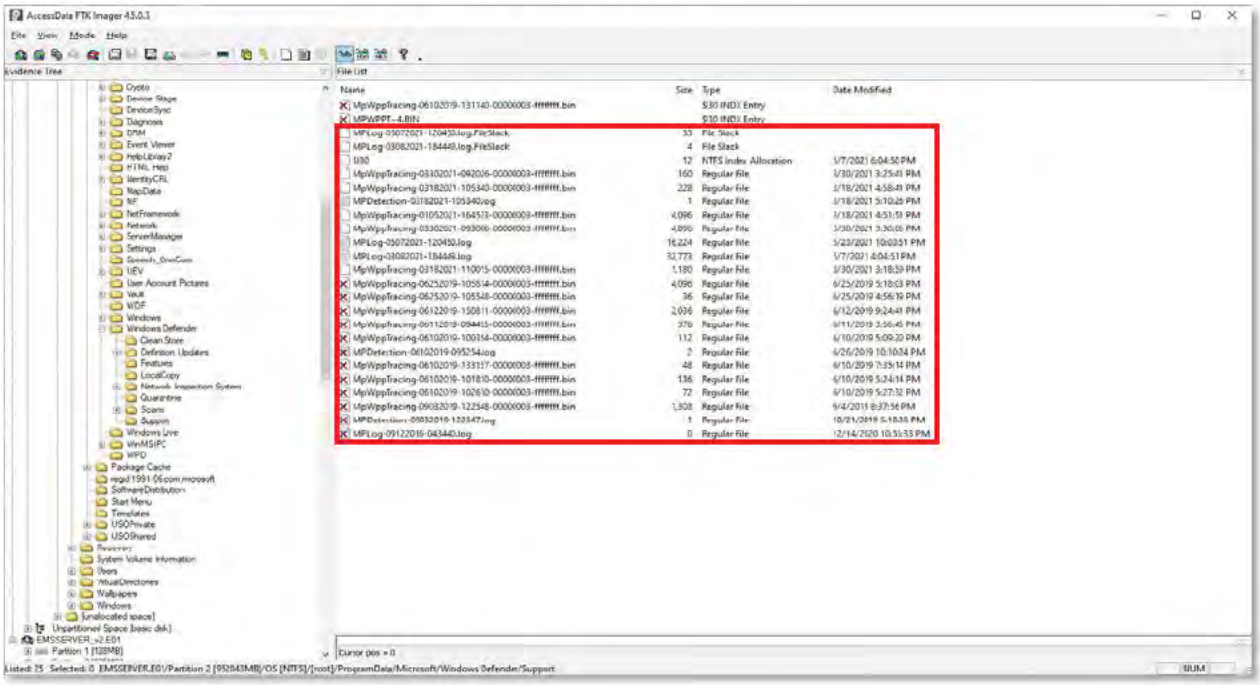
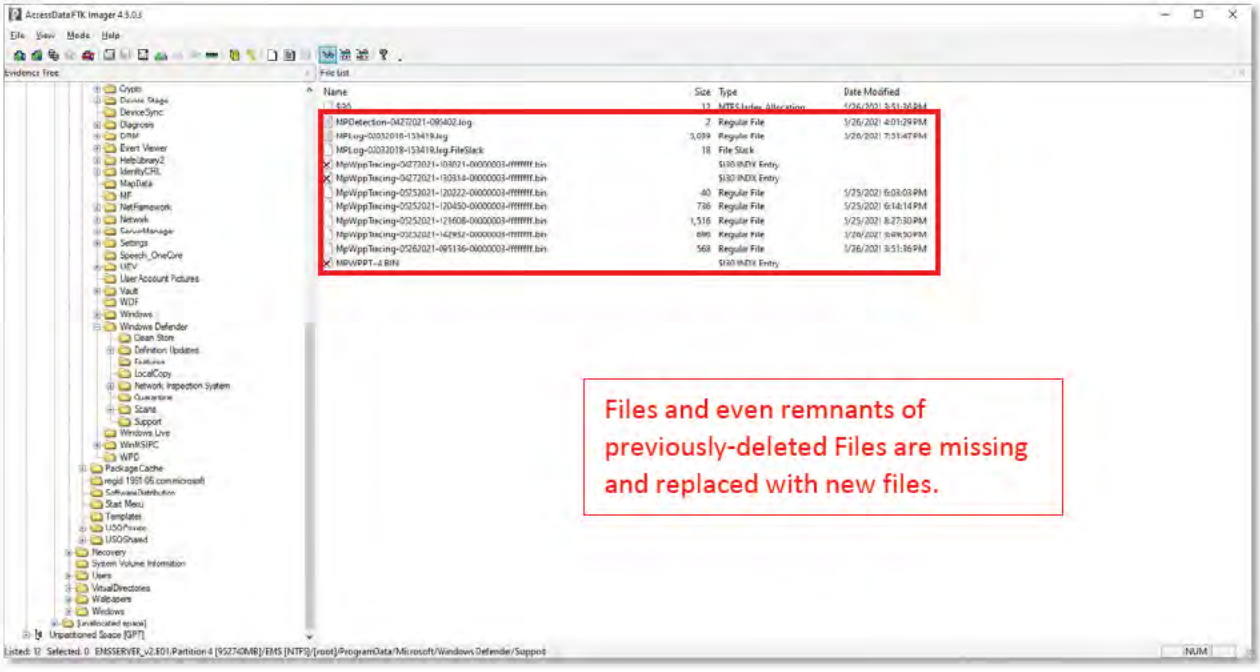


Figure 39 - EMS Server (5.13) Windows Defender Log Files After



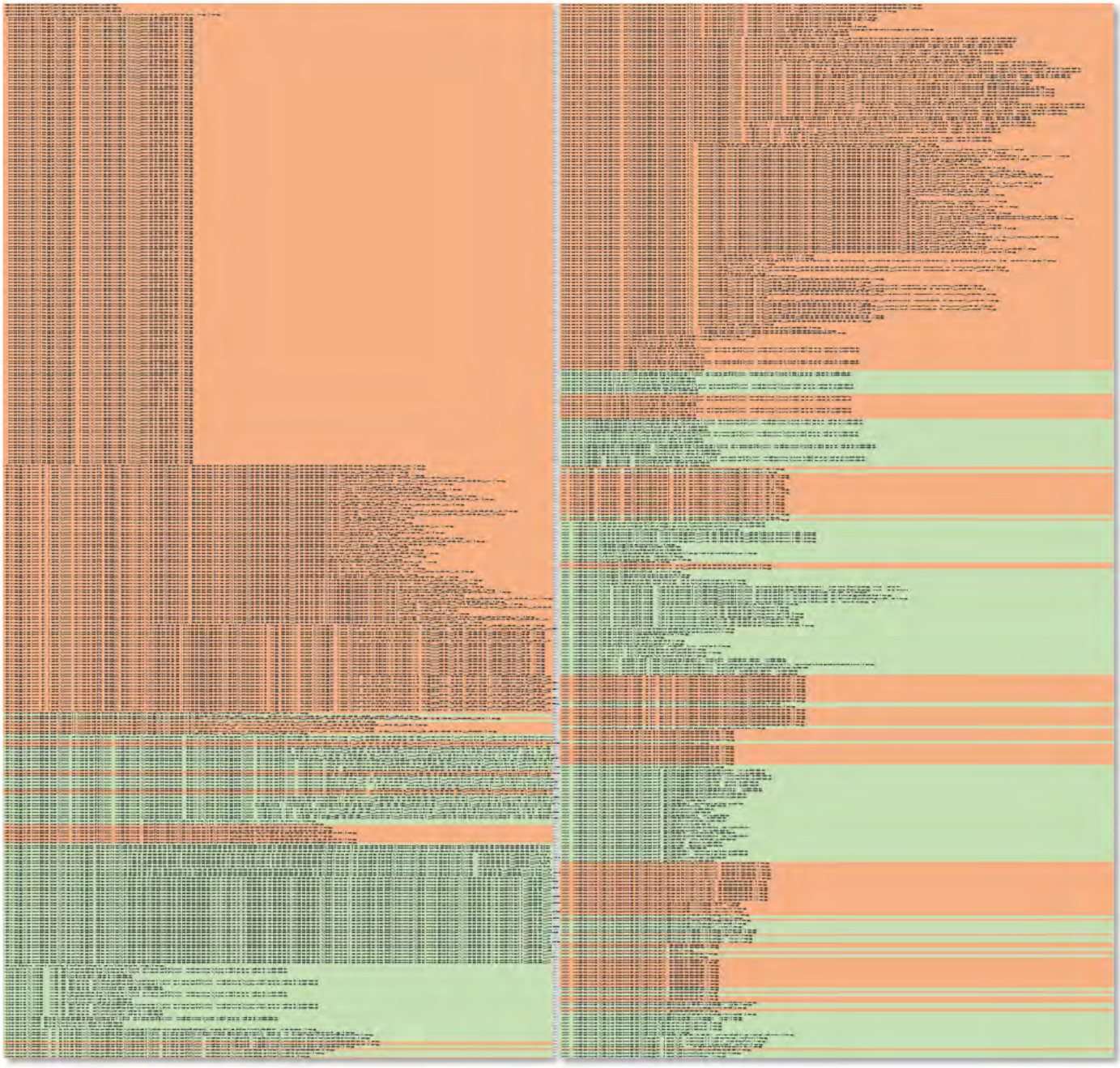
Server List of .log files in Before Image that were Deleted.

This list that follows this paragraph is a comparison of the Before and After forensic images of the Mesa County EMS Server. It is a comparative list of files that were deleted – either deliberately erased or overwritten with disregard for the preservation of these files, some of which contain ELECTION RELATED data.

Each line in the image below is the full path listing to each one of the 807 files that end with the word ".log" found on the EMS Server before the Dominion update was applied.

The Color code shows what happened to them After Dominion's update. Of all the files on the server Before the update – files highlighted in Green are still present on the server after the update, while files highlighted in light red have been overwritten and are deleted and not present in the image taken After the update.

Figure 40 - EMS Server Before/After .log File Comparison List



Significant Number of Logfiles Missing

The dataset from which this spreadsheet was created was extracted from the EnCase images of the original evidence on the hard drives of the EMS Server and had a traceable chain of custody. While the images above are too small to be readable, the entire content of this list is reproduced in Appendix A.

Of the original 807 ".log" files on the EMS Server before Dominion's update, only 302 remain, and 505 ".log" files have been deleted or overwritten.

Of the files *that remain*, the forensic examination has not yet verified whether the content of these files (which have the same filename and Path – e.g., in the same directories) is unchanged. The files that have been deleted DO include files that constitute Election Records and are subject to Federal and State data retention laws.

This list is only 807 files, and the text size is so small that the content is barely readable. The list of files has been broken down into small subsets because the number of files on the entire server totals 363,321 files, many of which are provided by Microsoft as part of the Windows Server 2016 operating system and its associated application programs and are not Election Related and do not contain actual Election Data.

Figure 41 - EMS Server (5.11-CO) List of .evtx Event Log Files Before

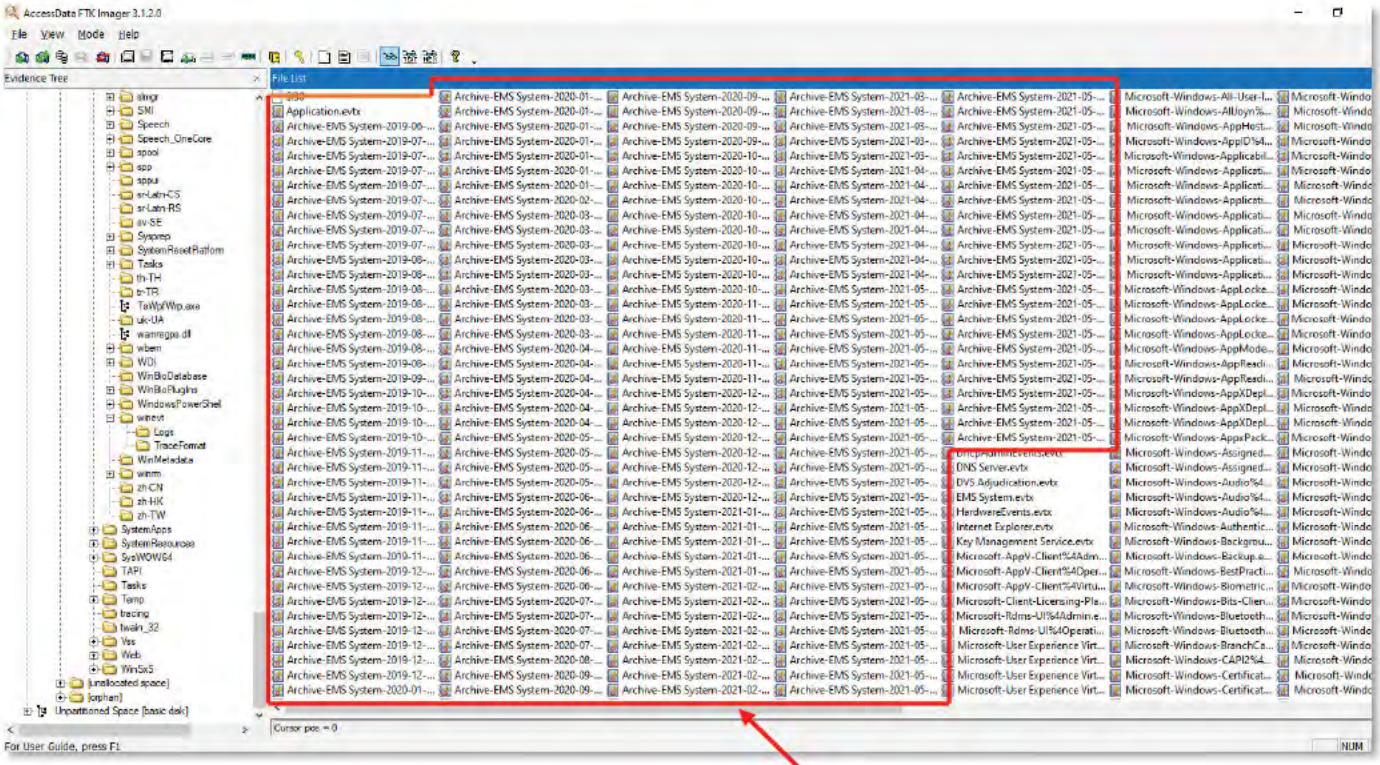
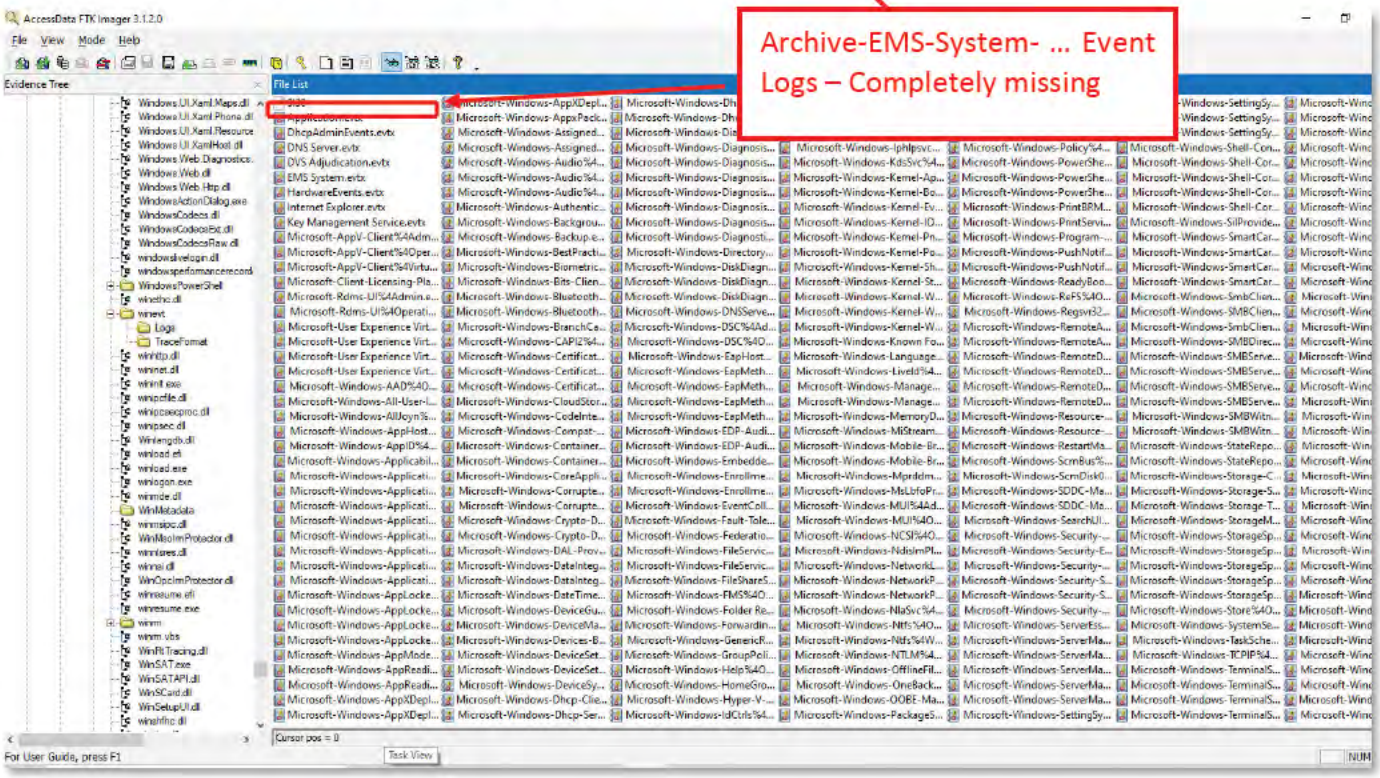


Figure 42 - EMS Server (5.13) List of .evtx Event Log Files After



This list that follows this paragraph is a comparison of the Before and After forensic images of the Mesa County EMS Server. It is a comparative list of files that were deleted – either deliberately erased or overwritten with disregard for the preservation of these files, some of which contain ELECTION RELATED data. A readable list is in Appendix C.

Each line in the image below is the full path listing (e.g., comparision of file names, not content) to each one of the 580 files that end with the word ".evtx" found on the EMS Server before the Dominion update was applied. 190 Event Log Files were deleted.

The Color code shows their status After Dominion's update. Of all the files on the server Before the update – files highlighted in Green are still present (although possibly changed) on the server after the update, while files highlighted in light red have been overwritten and are deleted and not present in the image After the update.

Logs\Microsoft-Windows-Kernel-WHEA%4Errors.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TaskScheduler%4
Logs\Key Management Service.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-VDRVROOT%4Op
Logs\Application evtx	Windows\System32\winevt\Logs\Microsoft-Windows-VHDMP-Operatio
Logs\HardwareEvents.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-AppXDeployment
Logs\Internet Explorer.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-AppXDeployment
Logs\Microsoft-Client-Licensing-Platform%4Admin evtx	Windows\System32\winevt\Logs\Microsoft-Windows-International%4O
Logs\Microsoft-Windows-Application-Experience%4Program-Compatibility-Assistant.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-MUI%4Operation
Logs\Microsoft-Windows-AppModel-Runtime%4Admin.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-MUI%4Admin.evt
Logs\Microsoft-Windows-AppReadiness%4Admin.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Windows Firewal
Logs\Microsoft-Windows-AppXDeployment%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Iphlpsvc%4Opera
Logs\Microsoft-Windows-AppXDeploymentServer%4Restricted evtx	Windows\System32\winevt\Logs\Microsoft-Windows-WMI-Activity%4C
Logs\Microsoft-Windows-CodeIntegrity%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-AppReadiness%4/
Logs\Microsoft-Windows-Containers-Wcifs%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-AppReadiness%4(
Logs\Microsoft-Windows-Containers-Wcnfs%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-ApplicationResou
Logs\Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider%4Admin.evtx	Windows\System32\winevt\Logs\Microsoft-Client-Licensing-Platform%4
Logs\Microsoft-Windows-Crypto-DPAPI%4BackUpKeySvc.evtx	Windows\System32\winevt\Logs\System evtx
Logs\Microsoft-Windows-Crypto-DPAPI%4Operational evtx	Windows\System32\winevt\Logs\Application.evtx
Logs\Microsoft-Windows-DeviceSetupManager%4Admin.evtx	Windows\System32\winevt\Logs\Security.evtx
Logs\Microsoft-Windows-DeviceSetupManager%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Dhcpv6-Client%4.
Logs\Microsoft-Windows-ApplicationResourceManagementSystem%4Operational evtx	Windows\System32\winevt\Logs\Windows PowerShell.evtx
Logs\Microsoft-Windows-Dhcp-Client%4Admin.evtx	Windows\System32\winevt\Logs\Key Management Service.evtx
Logs\Microsoft-Windows-Dhcpv6-Client%4Admin.evtx	Windows\System32\winevt\Logs\Internet Explorer evtx
Logs\Microsoft-Windows-Hyper-V-Guest-Drivers%4Admin.evtx	Windows\System32\winevt\Logs\HardwareEvents.evtx
Logs\Microsoft-Windows-International%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Wcmsvc%4Opera
Logs\Microsoft-Windows-AppReadiness%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Application-Exper
Logs\Microsoft-Windows-Kernel-Boot%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Program-Compat
Logs\Microsoft-Windows-Kernel-IO%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Dhcp-Client%4Ad
Logs\Microsoft-Windows-Kernel-EventTracing%4Admin.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-
Logs\Microsoft-Windows-Kernel-Power%4Thermal-Operational evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-
Logs\Microsoft-Windows-Kernel-ShimEngine%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-WinRM%4Operat
Logs\Microsoft-Windows-Kernel-StoreMgr%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Windows Defend
Logs\Microsoft-Windows-AppXDeploymentServer%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Windows Defend
Logs\Microsoft-Windows-Kernel-WHEA%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-UserPnp%4Devic
Logs\Microsoft-Windows-Known Folders API Service.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-UserPnp%4Actior
Logs\Microsoft-Windows-LiveId%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-DeviceManageme
Logs\Microsoft-Windows-MUI%4Admin.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-Power%4T
Logs\Microsoft-Windows-GroupPolicy%4Operational evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-Boot%4Op
Logs\Microsoft-Windows-MUI%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-StateRepository%
Logs\Microsoft-Windows-NCSI%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-StateRepository%
Logs\Microsoft-Windows-NetworkProfile%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-PnP%4Cor
Logs\Microsoft-Windows-Ntfs%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-ShimEngin
Logs\Microsoft-Windows-Ntfs%4WHC evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Ntfs%4Operation
Logs\Microsoft-Windows-Program-Compatibility-Assistant%4CompatAfterUpgrade.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Ntfs%4WHC.evtx
Logs\Microsoft-Windows-RemoteDesktopServices-RdpCoreTS%4Admin.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-VolumeSnapshot-
Logs\Microsoft-Windows-SettingSync%4Debug.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-IO%4Oper
Logs\Microsoft-Windows-RemoteDesktopServices-RdpCoreTS%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-CodeIntegrity%4(
Logs\Microsoft-Windows-Kernel-PnP%4Configuration evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-WHEA%4E
Logs\Microsoft-Windows-SettingSync%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-WHEA%4(
Logs\Microsoft-Windows-Shell-Core%4ActionCenter.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Windows Firewal
Logs\Microsoft-Windows-Shell-Core%4AppDefaults.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-NCSI%4Operation
Logs\Microsoft-Windows-Shell-Core%4LogonTasksChannel.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-WinINet-Config%
Logs\Microsoft-Windows-Shell-Core%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Crypto-DPAPI%4(
Logs\Microsoft-Windows-SmbClient%4Connectivity.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Crypto-DPAPI%4E
Logs\Microsoft-Windows-SMBClient%4Operational evtx	Windows\System32\winevt\Logs\Microsoft-Windows-DeviceSetupMana
Logs\Microsoft-Windows-SmbClient%4Security.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-DeviceSetupMana
Logs\Microsoft-Windows-SMBServer%4Audit.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Containers-Wcifs%
Logs\Microsoft-Windows-SMBServer%4Connectivity.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Containers-Wcnfs
Logs\Microsoft-Windows-SMBServer%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-GroupPolicy%4Op

Logs\Microsoft-Windows-SMBServer%4Security.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-User Profile Servi
Logs\Microsoft-Windows-SMBWitnessClient%4Admin.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-LiveId%4Operatic
Logs\Microsoft-Windows-SMBWitnessClient%4Informational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-SMBClient%4Ope
Logs\Microsoft-Windows-StateRepository%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-SmbClient%4Con
Logs\Microsoft-Windows-StateRepository%4Restricted.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-SmbClient%4Secu
Logs\Microsoft-Windows-TaskScheduler%4Maintenance evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Winlogon%4Ope
Logs\Microsoft-Windows-TerminalServices-LocalSessionManager%4Admin evtx	Windows\System32\winevt\Logs\Microsoft-Windows-SettingSync%4Op
Logs\Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-SettingSync%4De
Logs\Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Admin evtx	Windows\System32\winevt\Logs\Microsoft-Windows-DateTimeControl
Logs\Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Known Folders A
Logs\Microsoft-Windows-Store%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Shell-Core%4Acti
Logs\Microsoft-Windows-UniversalTelemetryClient%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Shell-Core%4Ope
Logs\Microsoft-Windows-User Profile Service%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Shell-Core%4App
Logs\Microsoft-Windows-UserPnp%4ActionCenter.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Shell-Core%4Log
Logs\Microsoft-Windows-UserPnp%4DeviceInstall evtx	Windows\System32\winevt\Logs\Microsoft-Windows-AppXDeployment
Logs\Microsoft-Windows-VolumeSnapshot-Driver%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-AppModel-Runti
Logs\Microsoft-Windows-Wcmsvc%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-ServerManager-D
Logs\Microsoft-Windows-Windows Defender%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Store%4Operatio
Logs\Microsoft-Windows-Windows Defender%4WHC.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-PushNotification-
Logs\Microsoft-Windows-Windows Firewall With Advanced Security%4ConnectionSecurity evtx	Windows\System32\winevt\Logs\Microsoft-Windows-PushNotification-
Logs\Microsoft-Windows-WinINet-Config%4ProxyConfigChanged.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TWinUI%4Operat
Logs\Microsoft-Windows-Winlogon%4Operational evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Application-Expe
Logs\Microsoft-Windows-WinRM%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Application-Expe
Logs\Setup evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Application-Expe
Logs\Windows PowerShell.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Application-Expe
Logs\Microsoft-Windows-Windows Firewall With Advanced Security%4Firewall evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Security-SPP-UX-
Logs\Microsoft-Windows-WMI-Activity%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-BackgroundTaskI
Logs\System evtx	Windows\System32\winevt\Logs\Microsoft-Windows-ServerManager-M
Logs\Security.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-ServerManager-M
Windows\System32\winevt\Logs\Setup.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Forwarding%4Op
Windows\System32\winevt\Logs\Archive-EMS System-2020-02-27-12-21-20-622.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-ServerManager-M
Windows\System32\winevt\Logs\Archive-EMS System-2019-07-11-20-46-32-189.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Resource-Exhaus
Windows\System32\winevt\Logs\Archive-EMS System-2019-06-30-13-45-03-347.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-DataIntegrityScar
Windows\System32\winevt\Logs\Archive-EMS System-2019-07-08-02-26-04-899.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-DataIntegrityScar
Windows\System32\winevt\Logs\Archive-EMS System-2019-10-30-19-26-37-188.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Diagnosis-Schedu
Windows\System32\winevt\Logs\Archive-EMS System-2019-07-04-08-05-33-867.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-HomeGroup Cont
Windows\System32\winevt\Logs\Archive-EMS System-2019-07-30-16-29-10-602.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Shell-ConnectedA
Windows\System32\winevt\Logs\Archive-EMS System-2019-07-15-15-07-04-381.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-StorageSpaces-M
Windows\System32\winevt\Logs\Archive-EMS System-2020-03-02-02-52-11-569.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Diagnosis-DPS%4
Windows\System32\winevt\Logs\Archive-EMS System-2020-04-19-00-10-16-214.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Storage-ClassPnP
Windows\System32\winevt\Logs\Archive-EMS System-2019-07-26-22-08-42-827.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-RestartManager%
Windows\System32\winevt\Logs\Archive-EMS System-2020-12-11-22-05-26-089.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Diagnosis-PLA%4
Windows\System32\winevt\Logs\Archive-EMS System-2021-02-23-04-20-21-835.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-CAPI2%4Operatic
Windows\System32\winevt\Logs\Archive-EMS System-2020-05-26-12-11-25-223.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-WindowsUpdate
Windows\System32\winevt\Logs\Archive-EMS System-2020-04-15-07-09-24-325.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-NlaSvc%4Operati
Windows\System32\winevt\Logs\Archive-EMS System-2020-06-04-04-35-23-707.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-PowerShell%4Op
Windows\System32\winevt\Logs\Archive-EMS System-2020-04-07-21-05-41-859.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-PowerShell%4Ad
Windows\System32\winevt\Logs\Archive-EMS System-2020-06-18-19-18-33-633.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Diagnosis-PCW%4
Windows\System32\winevt\Logs\Archive-EMS System-2021-02-09-23-20-20-681.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Storage-Storport%
Windows\System32\winevt\Logs\Archive-EMS System-2020-06-07-22-53-09-400.evtx	Windows\System32\winevt\Logs\EMS System.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-19-09-05-03-069.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Application Serve
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-19-03-17-30-918.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Application Serve
Windows\System32\winevt\Logs\Archive-EMS System-2019-10-02-11-09-22-083.evtx	Windows\System32\winevt\Logs\DVS Adjudication.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-02-09-15-51-43-896.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-CloudStorageWiz
Windows\System32\winevt\Logs\Archive-EMS System-2021-02-09-19-15-04-259.evtx	Windows\System32\winevt\Logs\Microsoft-AppV-Client%4Admin.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-04-28-04-14-58-545.evtx	Windows\System32\winevt\Logs\Microsoft-AppV-Client%4Operational
Windows\System32\winevt\Logs\Archive-EMS System-2021-04-06-04-10-43-774.evtx	Windows\System32\winevt\Logs\Microsoft-AppV-Client%4Virtual Appli
Windows\System32\winevt\Logs\Archive-EMS System-2020-03-24-00-59-56-063.evtx	Windows\System32\winevt\Logs\Microsoft-Rdms-UI%4Admin.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-01-08-17-03-22-249.evtx	Windows\System32\winevt\Logs\Microsoft-Rdms-UI%4Operational.evt
Windows\System32\winevt\Logs\Archive-EMS System-2019-11-10-09-23-03-203.evtx	Windows\System32\winevt\Logs\Microsoft-User Experience Virtualizati
Windows\System32\winevt\Logs\Archive-EMS System-2019-11-06-16-37-56-482.evtx	Windows\System32\winevt\Logs\Microsoft-User Experience Virtualizati
Windows\System32\winevt\Logs\Archive-EMS System-2019-12-02-15-06-36-405.evtx	Windows\System32\winevt\Logs\Microsoft-User Experience Virtualizati
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-07-05-51-17-641.evtx	Windows\System32\winevt\Logs\Microsoft-User Experience Virtualizati
Windows\System32\winevt\Logs\Archive-EMS System-2021-03-06-06-07-29-506.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-AAD%4Operation
Windows\System32\winevt\Logs\Archive-EMS System-2019-10-27-06-12-01-889.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-All-User-Install-A
Windows\System32\winevt\Logs\Archive-EMS System-2019-11-14-02-20-35-061.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-AllJoyn%4Operat
Windows\System32\winevt\Logs\Archive-EMS System-2021-03-16-20-09-20-723.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-AppHost%4Admi
Windows\System32\winevt\Logs\Archive-EMS System-2020-09-26-09-07-58-407.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-AppID%4Operati
Windows\System32\winevt\Logs\Archive-EMS System-2020-07-08-10-16-08-709.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-ApplicabilityEngi
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-26-06-16-16-735.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-AppLocker%4EXE
Windows\System32\winevt\Logs\Archive-EMS System-2020-06-21-16-36-38-559.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-AppLocker%4MS
Windows\System32\winevt\Logs\Archive-EMS System-2019-11-03-08-53-17-828.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-AppLocker%4Pac
Windows\System32\winevt\Logs\Archive-EMS System-2019-10-23-15-37-23-347.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-AppLocker%4Pac
Windows\System32\winevt\Logs\Archive-EMS System-2020-11-13-00-00-07-540.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-AppxPackaging%4
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-21-03-24-37-573.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-AssignedAccess%
Windows\System32\winevt\Logs\Archive-EMS System-2021-01-15-03-21-17-842.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-AssignedAccessB
Windows\System32\winevt\Logs\Archive-EMS System-2019-12-10-01-06-03-529.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Audio%4Capture

Windows\System32\winevt\Logs\Archive-EMS System-2019-11-25-05-09-12-916.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Audio%4Operati
Windows\System32\winevt\Logs\Archive-EMS System-2021-01-22-13-12-21-043.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Audio%4Playbac
Windows\System32\winevt\Logs\Archive-EMS System-2019-12-06-08-06-21-993.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Authentication U
Windows\System32\winevt\Logs\Archive-EMS System-2020-11-05-17-16-37-197.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Backup.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-16-23-08-11-827.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-BestPractices%4(
Windows\System32\winevt\Logs\Archive-EMS System-2021-04-17-01-19-18-484.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Biometrics%4Op
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-21-01-39-07-647.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Bits-Client%4Ope
Windows\System32\winevt\Logs\Archive-EMS System-2021-03-04-12-10-17-214.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Bluetooth-BthLEP
Windows\System32\winevt\Logs\Archive-EMS System-2020-07-12-02-56-47-489.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Bluetooth-MTPE
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-21-07-26-54-297.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-BranchCacheSME
Windows\System32\winevt\Logs\Archive-EMS System-2020-04-04-04-03-42-737.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-CertificateService
Windows\System32\winevt\Logs\Archive-EMS System-2020-11-16-16-31-58-059.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-CertificateService
Windows\System32\winevt\Logs\Archive-EMS System-2019-11-21-12-09-41-781.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-CertificateService
Windows\System32\winevt\Logs\Archive-EMS System-2019-12-28-14-04-05-771.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Compat-Appraise
Windows\System32\winevt\Logs\Archive-EMS System-2020-03-12-22-02-14-487.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-CoreApplication%
Windows\System32\winevt\Logs\Archive-EMS System-2019-12-13-18-05-49-770.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-CorruptedFileRec
Windows\System32\winevt\Logs\Archive-EMS System-2019-11-17-19-10-01-322.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-CorruptedFileRec
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-05-14-13-33-324.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-DAL-Provider%4(
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-14-13-10-56-695.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-DeviceGuard%4(
Windows\System32\winevt\Logs\Archive-EMS System-2019-11-28-22-08-50-835.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Devices-Backgrou
Windows\System32\winevt\Logs\Archive-EMS System-2019-12-17-11-04-36-787.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-DeviceSync%4Op
Windows\System32\winevt\Logs\Archive-EMS System-2020-01-05-00-03-39-390.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Diagnosis-Scripte
Windows\System32\winevt\Logs\Archive-EMS System-2020-01-12-10-03-10-333.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Diagnosis-Scripte
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-19-20-40-41-039.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Diagnosis-Scripte
Windows\System32\winevt\Logs\Archive-EMS System-2021-04-24-11-15-42-872.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Diagnostics-Netw
Windows\System32\winevt\Logs\Archive-EMS System-2020-09-18-17-48-53-388.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-DirectoryServices
Windows\System32\winevt\Logs\Archive-EMS System-2020-11-09-10-14-15-715.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-DiskDiagnostic%4
Windows\System32\winevt\Logs\Archive-EMS System-2020-03-31-11-01-47-908.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-DiskDiagnosticDa
Windows\System32\winevt\Logs\Archive-EMS System-2020-01-27-08-29-08-399.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-DiskDiagnosticRe
Windows\System32\winevt\Logs\Archive-EMS System-2020-09-14-23-29-04-158.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-DSC%4Admin.evt
Windows\System32\winevt\Logs\Archive-EMS System-2019-12-24-21-04-12-832.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-DSC%4Operation
Windows\System32\winevt\Logs\Archive-EMS System-2019-12-21-04-04-25-803.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-EapHost%4Opera
Windows\System32\winevt\Logs\Archive-EMS System-2020-01-01-07-03-50-651.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-EapMethods-Ras
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-14-18-29-08-151.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-EapMethods-Ras
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-11-21-02-29-740.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-EapMethods-Sim
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-03-11-33-19-283.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-EapMethods-Ttls
Windows\System32\winevt\Logs\Archive-EMS System-2020-01-19-20-02-44-037.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-EDP-Audit-Regul
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-28-20-58-32-283.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-EDP-Audit-TCB%4
Windows\System32\winevt\Logs\Archive-EMS System-2020-01-16-03-02-57-774.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-EmbeddedAppLa
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-01-21-14-15-340.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-EnrollmentPolicy
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-13-08-12-31-149.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-EnrollmentWebS
Windows\System32\winevt\Logs\Archive-EMS System-2020-04-22-17-12-11-701.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-EventCollector%4
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-16-05-45-04-636.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Fault-Tolerant-H
Windows\System32\winevt\Logs\Archive-EMS System-2020-03-05-12-32-06-091.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-FederationServic
Windows\System32\winevt\Logs\Archive-EMS System-2021-03-26-13-42-04-162.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-FileServices-Serv
Windows\System32\winevt\Logs\Archive-EMS System-2020-03-16-14-59-42-045.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-FileServices-Serv
Windows\System32\winevt\Logs\Archive-EMS System-2020-01-23-13-23-46-471.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-FileShareShadow
Windows\System32\winevt\Logs\Archive-EMS System-2020-05-18-23-53-08-392.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-FMS%4Operation
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-21-13-14-26-512.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Folder Redirectio
Windows\System32\winevt\Logs\Archive-EMS System-2020-03-27-17-59-53-690.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-NdisImPlatform%
Windows\System32\winevt\Logs\Archive-EMS System-2020-11-20-09-29-41-350.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-GenericRoaming%
Windows\System32\winevt\Logs\Archive-EMS System-2020-03-20-07-59-19-878.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Help%4Operatio
Windows\System32\winevt\Logs\Archive-EMS System-2020-01-31-02-47-11-038.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Hyper-V-Guest-D
Windows\System32\winevt\Logs\Archive-EMS System-2021-01-29-23-11-26-924.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-IdCtrls%4Operati
Windows\System32\winevt\Logs\Archive-EMS System-2020-03-09-05-29-49-411.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-IKE%4Operationa
Windows\System32\winevt\Logs\Archive-EMS System-2021-01-26-06-11-56-096.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-International-Reg
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-23-00-00-39-858.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-KdsSvc%4Operat
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-14-18-58-50-004.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-ApphelpC
Windows\System32\winevt\Logs\Archive-EMS System-2020-06-28-08-06-44-934.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-EventTrac
Windows\System32\winevt\Logs\Archive-EMS System-2020-04-11-14-07-34-718.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-WDI%4Op
Windows\System32\winevt\Logs\Archive-EMS System-2020-05-15-07-27-29-016.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-LanguagePackSet
Windows\System32\winevt\Logs\Archive-EMS System-2020-07-04-17-16-43-223.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-ManagementToo
Windows\System32\winevt\Logs\Archive-EMS System-2020-06-24-22-45-04-435.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-ManagementToo
Windows\System32\winevt\Logs\Archive-EMS System-2021-04-20-18-16-33-823.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-MemoryDiagnost
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-18-12-49-02-987.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-MiStreamProvide
Windows\System32\winevt\Logs\Archive-EMS System-2021-04-13-08-24-43-079.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Mobile-Broadbar
Windows\System32\winevt\Logs\Archive-EMS System-2020-09-22-12-08-49-154.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Mobile-Broadbar
Windows\System32\winevt\Logs\Archive-EMS System-2021-03-30-02-15-24-619.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Mprddm%4Oper
Windows\System32\winevt\Logs\Archive-EMS System-2020-05-22-17-53-50-782.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-MsLbfoProvider%
Windows\System32\winevt\Logs\Archive-EMS System-2020-06-11-17-12-21-460.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-NetworkLocation
Windows\System32\winevt\Logs\Archive-EMS System-2021-01-18-20-12-44-811.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-NetworkProvider
Windows\System32\winevt\Logs\Archive-EMS System-2021-04-02-14-34-04-967.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-NTLM%4Operati
Windows\System32\winevt\Logs\Archive-EMS System-2020-09-07-10-51-04-386.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-OfflineFiles%4Op
Windows\System32\winevt\Logs\Archive-EMS System-2020-09-11-05-09-09-286.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-OneBackup%4De
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-31-12-27-41-741.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-OOBE-Machine-D
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-22-20-55-04-936.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-PackageStateRoa

Windows\System32\winevt\Logs\Archive-EMS System-2020-07-01-03-27-07-105 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Partition%4Diagr
Windows\System32\winevt\Logs\Archive-EMS System-2021-04-09-17-04-07-509 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-PerceptionRuntin
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-09-07-12-48-391 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-PerceptionSenso
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-11-00-11-12-292 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-User Control Pan
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-15-12-21-57-907 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Policy%4Operatic
	Windows\System32\winevt\Logs\Microsoft-Windows-PowerShell-Desir
Windows\System32\winevt\Logs\Archive-EMS System-2020-11-03-08-03-17-087 evtx	FileDownloadManager%4Operational evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-22-18-13-07-673 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-PrintBRM%4Adm
Windows\System32\winevt\Logs\Archive-EMS System-2020-08-27-17-53-57-312 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-PrintService%4Ac
Windows\System32\winevt\Logs\Archive-EMS System-2020-09-30-01-16-19-620 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-ReadyBoost%4Op
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-20-19-51-20-073 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-ReFS%4Operatio
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-13-02-24-44-262 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Regsvr32%4Oper
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-15-23-57-17-682 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-RemoteApp and
Windows\System32\winevt\Logs\Archive-EMS System-2020-12-08-20-31-15-022 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-RemoteApp and
Windows\System32\winevt\Logs\Archive-EMS System-2020-12-09-21-34-01-652 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-RemoteDesktopS
Windows\System32\winevt\Logs\Archive-EMS System-2021-02-02-16-11-00-907 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-RemoteDesktopS
Windows\System32\winevt\Logs\Archive-EMS System-2020-12-10-17-14-18-337 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-RemoteDesktopS
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-17-16-31-18-634 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-RemoteDesktopS
Windows\System32\winevt\Logs\Archive-EMS System-2020-12-09-17-35-55-190 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Resource-Exhaus
Windows\System32\winevt\Logs\Archive-EMS System-2020-12-10-17-26-50-697 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-ScmBus%4Certifi
Windows\System32\winevt\Logs\Archive-EMS System-2020-12-08-21-07-21-169 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-ScmDisk0101%4O
Windows\System32\winevt\Logs\Archive-EMS System-2020-12-09-22-01-49-473 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-SearchUI%4Oper
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-18-15-41-56-864 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Security-Audit-Co
Windows\System32\winevt\Logs\Archive-EMS System-2021-02-09-17-14-16-397 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Security-Enterpri
Windows\System32\winevt\Logs\Archive-EMS System-2021-02-09-17-25-20-742 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Security-Netlogo
Windows\System32\winevt\Logs\Archive-EMS System-2021-02-09-17-33-50-215 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Security-SPP-UX-
Windows\System32\winevt\Logs\Archive-EMS System-2021-02-09-17-46-57-607 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Security-UserCon
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-16-17-20-24-507 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-ServerEssentials-
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-14-07-23-24-216 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-ServerManager-C
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-12-14-51-41-139 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-SettingSync-Azur
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-18-04-06-37-355 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-SettingSync-Azur
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-18-09-54-24-839 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-SilProvider%4Ope
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-19-14-52-50-172 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-SmartCard-Audit%
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-22-06-37-33-489 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-SmartCard-Devic
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-12-03-16-22-000 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-SmartCard-TPM-V
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-21-19-02-14-166 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-SmartCard-TPM-V
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-22-00-50-01-529 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-SMBDirect%4Adr
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-14-01-35-37-340 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-SMBWitnessClien
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-13-14-00-17-879 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-SMBWitnessClien
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-23-17-23-46-597 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Storage-Tiering%
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-12-20-37-42-553 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-StorageManagem
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-12-09-03-54-186 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-StorageSpaces-D
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-17-10-43-31-360 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-StorageSpaces-D
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-17-04-55-44-339 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-StorageSpaces-Sp
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-17-22-18-50-801 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-StorageSpaces-Sp
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-13-19-47-50-347 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-SystemSettingsTh
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-15-06-34-10-708 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TCPIP%4Operatic
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-20-02-28-13-043 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-18-21-29-43-807 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-15-00-46-23-325 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-15-18-09-30-390 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-20-14-03-47-942 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-23-11-36-14-332 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-16-11-32-36-872 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-20-08-16-00-636 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-23-05-48-27-189 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-22-12-25-20-731 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices
Windows\System32\winevt\Logs\Microsoft-Windows-UniversalTelemetryClient%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices
Windows\System32\winevt\Logs\Microsoft-Windows-NetworkProfile%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices
Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-StoreMgr%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices
Windows\System32\winevt\Logs\Microsoft-Windows-SMBServer%4Operational.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TZSync%4Operat
Windows\System32\winevt\Logs\Archive-EMS System-2019-07-19-09-27-36-681 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-TZUtil%4Operatic
Windows\System32\winevt\Logs\Microsoft-Windows-SMBServer%4Security.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-UAC%4Operation
Windows\System32\winevt\Logs\Archive-EMS System-2019-07-23-03-48-10-012 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-UAC-FileVirtualiz
Windows\System32\winevt\Logs\Microsoft-Windows-SMBServer%4Connectivity.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-User Device Regis
Windows\System32\winevt\Logs\Archive-EMS System-2019-08-14-17-50-17-089 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-User-Loader%4O
Windows\System32\winevt\Logs\Archive-EMS System-2019-08-22-06-31-22-632 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-VerifyHardwareS
Windows\System32\winevt\Logs\Archive-EMS System-2019-08-03-10-49-41-423 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Volume%4Diagno
Windows\System32\winevt\Logs\Microsoft-Windows-Dhcp-Server%4Operational evtx	Windows\System32\winevt\Logs\Microsoft-Windows-VPN-Client%4Ope
Windows\System32\winevt\Logs\Microsoft-Windows-Dhcp-Server%4FilterNotifications.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-VPN%4Operation
Windows\System32\winevt\Logs\DhcpAdminEvents.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-WFP%4Operation
Windows\System32\winevt\Logs\Archive-EMS System-2019-08-26-00-51-55-728 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Win32k%4Operat
Windows\System32\winevt\Logs\Archive-EMS System-2019-08-07-05-09-14-598 evtx	Windows\System32\winevt\Logs\Microsoft-Windows-WindowsSystemA
Windows\System32\winevt\Logs\DNS Server.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Winsock-WS2HEL

Windows\System32\winevt\Logs\Microsoft-Windows-DNSServer%4Audit.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Wired-AutoConfig%4
Windows\System32\winevt\Logs\Archive-EMS System-2019-08-10-23-29-48-856.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-Workplace Join%4Ad
Windows\System32\winevt\Logs\Archive-EMS System-2019-08-18-12-10-49-482.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-WPD-ClassInstaller%4
Windows\System32\winevt\Logs\Archive-EMS System-2019-09-02-13-32-58-546.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-WPD-CompositeClass
Windows\System32\winevt\Logs\Archive-EMS System-2019-08-29-19-12-25-021.evtx	Windows\System32\winevt\Logs\Microsoft-Windows-WPD-MTPClassDriver
Windows\System32\winevt\Logs\Microsoft-Windows-SMBServer%4Audit evtx	Windows\System32\winevt\Logs\SMSApi.evtx

Analysis Summary

Analysis of the Mesa County Dominion Voting Systems EMS server identified that extensive deletion of both election data and election-related data, comprising election records which must and should have been preserved under Federal and Colorado law, has occurred either as a result of or coincident with the vendor's and CO Secretary of State's modification of the system from version 5.11-CO to 5.13. This deleted data is critical to any effort to reconstruct events taking place on the voting systems, and to determine if unauthorized access or operation of the voting systems took place.

Furthermore, the EMS server application logging functions are configured to "Overwrite events as needed" if arbitrarily-selected file storage sizes are exceeded, which could predictably and likely has resulted in the systematic, automated deletion of logfile content comprising election-related data.

This systemic deletion of logfile data requires additional investigation.

CONCLUSION

This forensic examination found that significant election record preservation requirements under the 2002 VSS and Federal and state law HAVE NOT BEEN MET and further that destruction of Election-Related Data, specifically critical logfiles, has occurred. This destruction is not incidental or minor but is *highly significant*.

These findings have been demonstrated in this report and evidence has been presented demonstrating *conclusively to both computer systems experts as well as legal professionals and the general public at large* that the facts in these findings support the conclusions that:

1) Election-related data and election data explicitly required to be preserved, as described in the 2002 VSS criteria referenced in this section, HAS BEEN DESTROYED IN VIOLATION OF THE LAW, and

2) The specific configuration settings of the server examined lead to the understanding that Certification Requirements for Voting Systems have likely not been met despite this system having been certified and thereby approved for use in Colorado by the Colorado Secretary of State.

Further investigation is required to determine the full scope of non-compliance with legal mandates for voting systems and election records, and whether the non-compliance is deliberate or simply negligent.

APPENDIX A. DELETED ".LOG" FILES AFTER DOMINION TRUSTED BUILD UPDATE

Deleted files are highlighted in light red. Files highlighted in green are still present in the server image.

inetpub\logs\LogFiles\W3SVC1\u_ex210406.log

inetpub\logs\LogFiles\W3SVC1\u_ex200903.log

inetpub\logs\LogFiles\W3SVC1\u_ex191021.log

inetpub\logs\LogFiles\W3SVC1\u_ex191101.log

inetpub\logs\LogFiles\W3SVC1\u_ex201028.log

inetpub\logs\LogFiles\W3SVC1\u_ex191025.log

inetpub\logs\LogFiles\W3SVC1\u_ex191023.log

inetpub\logs\LogFiles\W3SVC1\u_ex200522.log

inetpub\logs\LogFiles\W3SVC1\u_ex191126.log

inetpub\logs\LogFiles\W3SVC1\u_ex200819.log

inetpub\logs\LogFiles\W3SVC1\u_ex191028.log

inetpub\logs\LogFiles\W3SVC1\u_ex210104.log

inetpub\logs\LogFiles\W3SVC1\u_ex191022.log

inetpub\logs\LogFiles\W3SVC1\u_ex200625.log

inetpub\logs\LogFiles\W3SVC1\u_ex210211.log

inetpub\logs\LogFiles\W3SVC1\u_ex201008.log

inetpub\logs\LogFiles\W3SVC1\u_ex191114.log

inetpub\logs\LogFiles\W3SVC1\u_ex200826.log

inetpub\logs\LogFiles\W3SVC1\u_ex210223.log

inetpub\logs\LogFiles\W3SVC1\u_ex210224.log

inetpub\logs\LogFiles\W3SVC1\u_ex210205.log

inetpub\logs\LogFiles\W3SVC1\u_ex210318.log

inetpub\logs\LogFiles\W3SVC1\u_ex200520.log

inetpub\logs\LogFiles\W3SVC1\u_ex201208.log

inetpub\logs\LogFiles\W3SVC1\u_ex210407.log

inetpub\logs\LogFiles\W3SVC1\u_ex191030.log

inetpub\logs\LogFiles\W3SVC1\u_ex191031.log

inetpub\logs\LogFiles\W3SVC1\u_ex191106.log

inetpub\logs\LogFiles\W3SVC1\u_ex191105.log

inetpub\logs\LogFiles\W3SVC1\u_ex191029.log

inetpub\logs\LogFiles\W3SVC1\u_ex191104.log

inetpub\logs\LogFiles\W3SVC1\u_ex200730.log

inetpub\logs\LogFiles\W3SVC1\u_ex210512.log

inetpub\logs\LogFiles\W3SVC1\u_ex201103.log

inetpub\logs\LogFiles\W3SVC1\u_ex191107.log

inetpub\logs\LogFiles\W3SVC1\u_ex191115.log

inetpub\logs\LogFiles\W3SVC1\u_ex200929.log

inetpub\logs\LogFiles\W3SVC1\u_ex200930.log

inetpub\logs\LogFiles\W3SVC1\u_ex200813.log

inetpub\logs\LogFiles\W3SVC1\u_ex210523.log

inetpub\logs\LogFiles\W3SVC1\u_ex200127.log

inetpub\logs\LogFiles\W3SVC1\u_ex200224.log

inetpub\logs\LogFiles\W3SVC1\u_ex201023.log

inetpub\logs\LogFiles\W3SVC1\u_ex200618.log

inetpub\logs\LogFiles\W3SVC1\u_ex210212.log

inetpub\logs\LogFiles\W3SVC1\u_ex200302.log

inetpub\logs\LogFiles\W3SVC1\u_ex200124.log

inetpub\logs\LogFiles\W3SVC1\u_ex200303.log

inetpub\logs\LogFiles\W3SVC1\u_ex210303.log

inetpub\logs\LogFiles\W3SVC1\u_ex200227.log

inetpub\logs\LogFiles\W3SVC1\u_ex201113.log

inetpub\logs\LogFiles\W3SVC1\u_ex201214.log

inetpub\logs\LogFiles\W3SVC1\u_ex201218.log

inetpub\logs\LogFiles\W3SVC1\u_ex200528.log

inetpub\logs\LogFiles\W3SVC1\u_ex201222.log

inetpub\logs\LogFiles\W3SVC1\u_ex200131.log

inetpub\logs\LogFiles\W3SVC1\u_ex210105.log

inetpub\logs\LogFiles\W3SVC1\u_ex201221.log

inetpub\logs\LogFiles\W3SVC1\u_ex200228.log

inetpub\logs\LogFiles\W3SVC1\u_ex200304.log

inetpub\logs\LogFiles\W3SVC1\u_ex200518.log

inetpub\logs\LogFiles\W3SVC1\u_ex210302.log

inetpub\logs\LogFiles\W3SVC1\u_ex200715.log

inetpub\logs\LogFiles\W3SVC1\u_ex200624.log

inetpub\logs\LogFiles\W3SVC1\u_ex210113.log

inetpub\logs\LogFiles\W3SVC1\u_ex200519.log

inetpub\logs\LogFiles\W3SVC1\u_ex200320.log

inetpub\logs\LogFiles\W3SVC1\u_ex210106.log

inetpub\logs\LogFiles\W3SVC1\u_ex210222.log

inetpub\logs\LogFiles\W3SVC1\u_ex210412.log

inetpub\logs\LogFiles\W3SVC1\u_ex200827.log

inetpub\logs\LogFiles\W3SVC1\u_ex200623.log

inetpub\logs\LogFiles\W3SVC1\u_ex210210.log

inetpub\logs\LogFiles\W3SVC1\u_ex200617.log

inetpub\logs\LogFiles\W3SVC1\u_ex200515.log

inetpub\logs\LogFiles\W3SVC1\u_ex200731.log

inetpub\logs\LogFiles\W3SVC1\u_ex201001.log

inetpub\logs\LogFiles\W3SVC1\u_ex201215.log

inetpub\logs\LogFiles\W3SVC1\u_ex200521.log

inetpub\logs\LogFiles\W3SVC1\u_ex210111.log

inetpub\logs\LogFiles\W3SVC1\u_ex200526.log

inetpub\logs\LogFiles\W3SVC1\u_ex200601.log

inetpub\logs\LogFiles\W3SVC1\u_ex200612.log

inetpub\logs\LogFiles\W3SVC1\u_ex210115.log

inetpub\logs\LogFiles\W3SVC1\u_ex210112.log

inetpub\logs\LogFiles\W3SVC1\u_ex210107.log

inetpub\logs\LogFiles\W3SVC1\u_ex200616.log

inetpub\logs\LogFiles\W3SVC1\u_ex210209.log

inetpub\logs\LogFiles\W3SVC1\u_ex210409.log

inetpub\logs\LogFiles\W3SVC1\u_ex200630.log

inetpub\logs\LogFiles\W3SVC1\u_ex200708.log

inetpub\logs\LogFiles\W3SVC1\u_ex200701.log

inetpub\logs\LogFiles\W3SVC1\u_ex210511.log

inetpub\logs\LogFiles\W3SVC1\u_ex200924.log

inetpub\logs\LogFiles\W3SVC1\u_ex201019.log

inetpub\logs\LogFiles\W3SVC1\u_ex201029.log

inetpub\logs\LogFiles\W3SVC1\u_ex201109.log

inetpub\logs\LogFiles\W3SVC1\u_ex201123.log

inetpub\logs\LogFiles\W3SVC1\u_ex201026.log

inetpub\logs\LogFiles\W3SVC1\u_ex201120.log

inetpub\logs\LogFiles\W3SVC1\u_ex201209.log

inetpub\logs\LogFiles\W3SVC1\u_ex201102.log

inetpub\logs\LogFiles\W3SVC1\u_ex210301.log

inetpub\logs\LogFiles\W3SVC1\u_ex210330.log

inetpub\logs\LogFiles\W3SVC1\u_ex210329.log

inetpub\logs\LogFiles\W3SVC1\u_ex210402.log

inetpub\logs\LogFiles\W3SVC1\u_ex210114.log

inetpub\logs\LogFiles\W3SVC1\u_ex210108.log

inetpub\logs\LogFiles\W3SVC1\u_ex210405.log

inetpub\logs\LogFiles\W3SVC1\u_ex210310.log

inetpub\logs\LogFiles\W3SVC1\u_ex210311.log

inetpub\logs\LogFiles\W3SVC1\u_ex210304.log

inetpub\logs\LogFiles\W3SVC1\u_ex210315.log

inetpub\logs\LogFiles\W3SVC1\u_ex210309.log

inetpub\logs\LogFiles\W3SVC1\u_ex210331.log

inetpub\logs\LogFiles\W3SVC1\u_ex210415.log

inetpub\logs\LogFiles\W3SVC1\u_ex210510.log

inetpub\logs\LogFiles\W3SVC1\u_ex190903.log

inetpub\logs\LogFiles\W3SVC1\u_ex190625.log

inetpub\logs\LogFiles\W3SVC1\u_ex190610.log

inetpub\logs\LogFiles\W3SVC1\u_ex190611.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\VC10Redist_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\VC10Redist_Cpu32_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\VSHelp_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\SqlSupport_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\sql_tools_extensions_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\sql_tools_extensions_loc_Cpu64_1033_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\sql_common_core_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\conn_info_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\conn_info_loc_Cpu64_1033_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\sql_batchparser_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\sql_engine_core_shared_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\sql_common_core_loc_Cpu64_1033_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\RsFx_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\sql_engine_core_inst_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\SqlDom_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\sql_engine_core_shared_loc_Cpu64_1033_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\sql_engine_core_inst_loc_Cpu64_1033_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\sql_diag_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\sql_rs_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\sql_dmf_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\sql_dmf_loc_Cpu64_1033_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\sql_xevent_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\smo_extensions_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\smo_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\sql_xevent_loc_Cpu64_1033_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\smo_extensions_loc_Cpu64_1033_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\sql_rs_loc_Cpu64_1033_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\sqlncli_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\msodbcsql_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\smo_loc_Cpu64_1033_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\tsqllangsvc_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\dacfx_Cpu32_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\SqlSqmShared_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\SqlWriter_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\SqlBrowser_Cpu32_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\SqlSupport_KatmaiRTM_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\sql_tools_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_120309\sql_tools_loc_Cpu64_1033_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_125107\MSSQLSERVER\msodbcsql_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_125107\MSSQLSERVER\SqlWriter_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_125107\MSSQLSERVER\SqlBrowser_Cpu32_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_125107\MSSQLSERVER\SqlSqmShared_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_125107\MSSQLSERVER\sql_common_core_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_125107\MSSQLSERVER\sql_rs_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_125107\MSSQLSERVER\sql_common_core_loc_Cpu64_1033_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_125107\MSSQLSERVER\sql_rs_loc_Cpu64_1033_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_125107\MSSQLSERVER\sql_engine_core_inst_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_125107\MSSQLSERVER\sql_engine_core_shared_loc_Cpu64_1033_1.l

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_125107\MSSQLSERVER\RsFx_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_125107\MSSQLSERVER\SqlDom_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_125107\MSSQLSERVER\sql_engine_core_shared_Cpu64_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_125107\MSSQLSERVER\sql_engine_core_inst_loc_Cpu64_1033_1.log

Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20190610_125107\MSSQLSERVER\SqlSupport_Cpu64_1.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\ReportServerService__05_09_2021_00_02_18.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\Microsoft.ReportingServices.Portal.WebHost_03_18_20

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\ReportServerService__05_20_2021_00_02_42.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\ReportServerService__05_19_2021_00_02_39.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\ReportServerService__05_17_2021_00_02_35.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\Microsoft.ReportingServices.Portal.WebHost_03_18_20

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\ReportServerService__05_23_2021_00_02_48.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\ReportServerService__05_22_2021_00_02_47.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\ReportServerService__05_16_2021_00_02_33.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\ReportServerService__05_13_2021_00_02_26.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\ReportServerService__05_12_2021_00_02_24.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\ReportServerService__05_15_2021_00_02_30.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\ReportServerService__05_10_2021_00_02_20.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\ReportServerService__05_11_2021_00_02_22.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\ReportServerService__05_21_2021_00_02_45.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\ReportServerService__05_18_2021_00_02_37.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\ReportServerService__05_14_2021_00_02_28.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\Microsoft.ReportingServices.Portal.WebHost_01_05_20

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\Microsoft.ReportingServices.Portal.WebHost_03_30_20

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\Microsoft.ReportingServices.Portal.WebHost_03_30_20

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\ReportServerService__09_03_2019_12_25_51.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\Microsoft.ReportingServices.Portal.WebHost_06_26_20

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\Microsoft.ReportingServices.Portal.WebHost_06_26_20

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\ReportServerService__06_25_2019_10_55_56.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\Microsoft.ReportingServices.Portal.WebHost_06_25_20

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\ReportServerService__06_25_2019_10_58_23.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\Microsoft.ReportingServices.Portal.WebHost_06_25_20

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\ReportServerService__06_12_2019_15_08_21.log

Program Files\Microsoft SQL Server\MSRS13.MSSQLSERVER\Reporting Services\LogFiles\Microsoft.ReportingServices.Portal.WebHost_06_12_20

ProgramData\Dell\UpdatePackage\log\support\BIOS_YY63D_WN64_2.9.1.log

ProgramData\Dell\UpdatePackage\log\support\SAS-RAID_Driver_T244W_WN64_6 604.06.00_A01_07.log

ProgramData\Dell\UpdatePackage\log\support\Drivers-for-OS-Deployment_Application_WP3PH_WN64_18.12.04_A00_01.log

ProgramData\Dell\UpdatePackage\log\support\Power_Firmware_8R0NM_WN64_00.1B.53.log

ProgramData\Dell\UpdatePackage\log\support\SAS-RAID_Firmware_F675Y_WN64_25.5.5.0005_A13_01.log

ProgramData\Dell\UpdatePackage\log\support\Network_Firmware_F3KFN_WN64_21.40.9.log

ProgramData\Dell\UpdatePackage\log\support\iDRAC-with-Lifecycle-Controller_Firmware_40T1C_WN64_2.63.60.61_A00.log

ProgramData\Dell\UpdatePackage\log\support\BIOS_T9YX9_WN64_2 9.1.log

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.AAD.BrokerPlugin_1000.14393.0.0_neutral_neutral_cw5n1h2txyewy\Act

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.AAD.BrokerPlugin_1000.14393.0.0_neutral_neutral_cw5n1h2txyewy\Act

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.AccountsControl_10.0.14393.0_neutral__cw5n1h2txyewy\ActivationStor

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.AccountsControl_10.0.14393.0_neutral__cw5n1h2txyewy\ActivationStor

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.BioEnrollment_10.0.14393.0_neutral__cw5n1h2txyewy\ActivationStore.

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.BioEnrollment_10.0.14393.0_neutral__cw5n1h2txyewy\ActivationStore.

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.LockApp_10.0.14393.0_neutral__cw5n1h2txyewy\ActivationStore.dat.LO

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.LockApp_10.0.14393.0_neutral__cw5n1h2txyewy\ActivationStore.dat.LO

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.Windows.Apprep.ChxApp_1000.14393.0.0_neutral_neutral_cw5n1h2txye

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.Windows.Apprep.ChxApp_1000.14393.0.0_neutral_neutral_cw5n1h2txye

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.Windows.AssignedAccessLockApp_1000.14393.0.0_neutral_neutral_cw5

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.Windows.AssignedAccessLockApp_1000.14393.0.0_neutral_neutral_cw5

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.Windows.CloudExperienceHost_10.0.14393.0_neutral_neutral_cw5n1h2t

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.Windows.CloudExperienceHost_10.0.14393.0_neutral_neutral_cw5n1h2t

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.Windows.Cortana_1.7.0.14393_neutral_neutral_cw5n1h2txyewy\Activat

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.Windows.Cortana_1.7.0.14393_neutral_neutral_cw5n1h2txyewy\Activat

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.Windows.SecondaryTileExperience_10.0.0.0_neutral__cw5n1h2txyewy\A

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.Windows.SecondaryTileExperience_10.0.0.0_neutral__cw5n1h2txyewy\A

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.Windows.ShellExperienceHost_10.0.14393.0_neutral_neutral_cw5n1h2t

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.Windows.ShellExperienceHost_10.0.14393.0_neutral_neutral_cw5n1h2t

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.XboxGameCallableUI_1000.14393.0.0_neutral_neutral_cw5n1h2txyewy\

ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.XboxGameCallableUI_1000.14393.0.0_neutral_neutral_cw5n1h2txyewy\

ProgramData\Microsoft\Windows\AppRepository\Packages\windows.immersivecontrolpanel_6.2.0.0_neutral_neutral_cw5n1h2txyewy\Activati

ProgramData\Microsoft\Windows\AppRepository\Packages\windows.immersivecontrolpanel_6.2.0.0_neutral_neutral_cw5n1h2txyewy\Activati

ProgramData\Microsoft\Windows\AppRepository\Packages\Windows.MiracastView_6.3.0.0_neutral_neutral_cw5n1h2txyewy\ActivationStore.

ProgramData\Microsoft\Windows\AppRepository\Packages\Windows.MiracastView_6.3.0.0_neutral_neutral_cw5n1h2txyewy\ActivationStore.

ProgramData\Microsoft\Windows\AppRepository\Packages\Windows.PrintDialog_6.2.0.0_neutral_neutral_cw5n1h2txyewy\ActivationStore.dat

ProgramData\Microsoft\Windows\AppRepository\Packages\Windows.PrintDialog_6.2.0.0_neutral_neutral_cw5n1h2txyewy\ActivationStore.dat

ProgramData\Microsoft\Windows Defender\Scans\History\Service\History.Log

ProgramData\Microsoft\Windows Defender\Scans\History\Service\Unknown.Log

ProgramData\Microsoft\Windows Defender\Support\MPLog-03082021-184449.log

ProgramData\Microsoft\Windows Defender\Support\MPLog-05072021-120450.log

ProgramData\Microsoft\Windows Defender\Support\MPDetection-03182021-105340.log

ProgramData\Microsoft\Windows Defender\Support\MPLog-09122016-043440.log

ProgramData\Microsoft\Windows Defender\Support\MPDetection-09032019-122547.log

ProgramData\Microsoft\Windows Defender\Support\MPDetection-06102019-095254.log

ProgramData\Package Cache\02A26E554FBB4232ACD36E70D09F2C7893D399CD\%localappdata%\temp\Ssm
DB65F37EB5E9}_001_kb3095681.log

ProgramData\Package Cache\02A26E554FBB4232ACD36E70D09F2C7893D399CD\%localappdata%\temp\SsmsSetup\VS2015KB3095681Update_

ProgramData\Package Cache\02A26E554FBB4232ACD36E70D09F2C7893D399CD\%localappdata%\temp\Ssms
444C320629FA}_001_kb3095681.log

ProgramData\Package Cache\4F812BBB2BE7E30CED293F8A229A5410D70DE6DB\%localappdata%\temp\SsmsSetup\VSTALS2015_003_RoslynLa

ProgramData\Package Cache\4F812BBB2BE7E30CED293F8A229A5410D70DE6DB\%localappdata%\temp\SsmsSetup\VSTALS2015_004_vsta_lan

ProgramData\Package Cache\4F812BBB2BE7E30CED293F8A229A5410D70DE6DB\%localappdata%\temp\SsmsSetup\VSTALS2015_001_vsta_lslp

ProgramData\Package Cache\4F812BBB2BE7E30CED293F8A229A5410D70DE6DB\%localappdata%\temp\SsmsSetup\VSTALS2015_002_RoslynLa

ProgramData\Package Cache\4F812BBB2BE7E30CED293F8A229A5410D70DE6DB\%localappdata%\temp\SsmsSetup\VSTALS2015_000_vsta_lscc

ProgramData\Package Cache\5E6157D16EC044A823B2FD2C030ED6DECD2E997E\%localappdata%\temp\SsmsSetup\VSTA2015_001_vsta_hostir

ProgramData\Package Cache\5E6157D16EC044A823B2FD2C030ED6DECD2E997E\%localappdata%\temp\SsmsSetup\VSTA2015_002_vsta_finaliz

ProgramData\Package Cache\5E6157D16EC044A823B2FD2C030ED6DECD2E997E\%localappdata%\temp\SsmsSetup\VSTA2015_000_vsta_hostir

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_018_Msi_B

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_020_Msi_B

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_032_vs_isc

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_019_Msi_B

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_021_sdk_t

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_004_vcRun

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_005_vcRun

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_006_vcRun

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_003_vcRun

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_007_vs_vsl

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_008_vsbsln

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_009_vsbsln

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_010_netfx_

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_011_netfx_

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_012_netfx_

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_013_netfx_

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_014_netfxc

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_015_netfx_

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_017_Msi_B

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_022_sdk_t

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_023_sqlsys

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_024_share

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_025_help3_

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_026_Bliss_

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_027_Bliss_

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_030_vs_mi

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_031_vs_mi

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_033_vs_isc

ProgramData\Package Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\%localappdata%\temp\SsmsSetup\VS2015IsoShell_029_vs_mi

System Volume Information\tracking.log

Users\.NET v2.0\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1

Users\.NET v2.0\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2

Users\.NET v2.0\ntuser.dat.LOG1

Users\.NET v2.0\ntuser.dat.LOG2

Users\.NET v2.0 Classic\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1

Users\.NET v2.0 Classic\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2

Users\.NET v2.0 Classic\ntuser.dat.LOG1

Users\.NET v2.0 Classic\ntuser.dat.LOG2

Users\.NET v4.5\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1

Users\.NET v4.5\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2

Users\.NET v4.5\ntuser.dat.LOG1

Users\.NET v4.5\ntuser.dat.LOG2

Users\.NET v4.5 Classic\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1

Users\.NET v4.5 Classic\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2

Users\.NET v4.5 Classic\ntuser.dat.LOG1

Users\.NET v4.5 Classic\ntuser.dat.LOG2

Users\AdjSys\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1

Users\AdjSys\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2

Users\AdjSys\ntuser.dat.LOG1

Users\AdjSys\ntuser.dat.LOG2

Users\Administrator\AppData\Local\ConnectedDevicesPlatform\CDPTraces.log

Users\Administrator\AppData\Local\Microsoft\Internet Explorer\ie4uinit-UserConfig.log

Users\Administrator\AppData\Local\Microsoft\Internet Explorer\ie4uinit-ClearIconCache.log

Users\Administrator\AppData\Local\Microsoft\Windows\SettingSync\metastore\edbtmp.log

Users\Administrator\AppData\Local\Microsoft\Windows\SettingSync\metastore\edb00001.log

Users\Administrator\AppData\Local\Microsoft\Windows\SettingSync\metastore\edb00002.log

Users\Administrator\AppData\Local\Microsoft\Windows\SettingSync\metastore\edb.log

Users\Administrator\AppData\Local\Microsoft\Windows\WebCache\V010000B.log

Users\Administrator\AppData\Local\Microsoft\Windows\WebCache\V01tmp.log

Users\Administrator\AppData\Local\Microsoft\Windows\WebCache\V01.log

Users\Administrator\AppData\Local\Microsoft\Windows\WebCache\V0100002.log

Users\Administrator\AppData\Local\Microsoft\Windows\WebCache\V010000A.log

Users\Administrator\AppData\Local\Microsoft\Windows\WebCache\V010000C.log

Users\Administrator\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1

Users\Administrator\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2

Users\Administrator\AppData\Local\Packages\Microsoft.AAD.BrokerPlugin_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\Administrator\AppData\Local\Packages\Microsoft.AAD.BrokerPlugin_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\Administrator\AppData\Local\Packages\Microsoft.AccountsControl_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\Administrator\AppData\Local\Packages\Microsoft.AccountsControl_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\Administrator\AppData\Local\Packages\Microsoft.BioEnrollment_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\Administrator\AppData\Local\Packages\Microsoft.BioEnrollment_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\Administrator\AppData\Local\Packages\Microsoft.LockApp_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\Administrator\AppData\Local\Packages\Microsoft.LockApp_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\Administrator\AppData\Local\Packages\Microsoft.Windows.Apprep.ChxApp_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\Administrator\AppData\Local\Packages\Microsoft.Windows.Apprep.ChxApp_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\Administrator\AppData\Local\Packages\Microsoft.Windows.AssignedAccessLockApp_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\Administrator\AppData\Local\Packages\Microsoft.Windows.AssignedAccessLockApp_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\Administrator\AppData\Local\Packages\Microsoft.Windows.CloudExperienceHost_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\Administrator\AppData\Local\Packages\Microsoft.Windows.CloudExperienceHost_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\Administrator\AppData\Local\Packages\Microsoft.Windows.Cortana_cw5n1h2txyewy\AC\Temp\StructuredQuery.log

Users\Administrator\AppData\Local\Packages\Microsoft.Windows.Cortana_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\Administrator\AppData\Local\Packages\Microsoft.Windows.Cortana_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\Administrator\AppData\Local\Packages\Microsoft.Windows.SecondaryTileExperience_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\Administrator\AppData\Local\Packages\Microsoft.Windows.SecondaryTileExperience_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\Administrator\AppData\Local\Packages\Microsoft.Windows.ShellExperienceHost_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\Administrator\AppData\Local\Packages\Microsoft.Windows.ShellExperienceHost_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\Administrator\AppData\Local\Packages\Microsoft.XboxGameCallableUI_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\Administrator\AppData\Local\Packages\Microsoft.XboxGameCallableUI_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\Administrator\AppData\Local\Packages\windows.immersivecontrolpanel_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\Administrator\AppData\Local\Packages\windows.immersivecontrolpanel_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\Administrator\AppData\Local\Packages\Windows.MiracastView_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\Administrator\AppData\Local\Packages\Windows.MiracastView_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\Administrator\AppData\Local\Packages\Windows.PrintDialog_cw5n1h2txyewy

Users\Administrator\AppData\Local\Packages\Windows.PrintDialog_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\Administrator\AppData\Local\Packages\Windows.PrintDialog_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\Administrator\AppData\Local\Temp\MpSigStub.log

Users\Administrator\AppData\Local\Temp\wmsetup.log

Users\Administrator\AppData\Local\TileDataLayer\Database\EDBtmp.log

Users\Administrator\AppData\Local\TileDataLayer\Database\EDB00002.log

Users\Administrator\AppData\Local\TileDataLayer\Database\EDB.log

Users\Administrator\ntuser.dat.LOG1

Users\Administrator\ntuser.dat.LOG2

Users\Classic .NET AppPool\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1

Users\Classic .NET AppPool\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2

Users\Classic .NET AppPool\ntuser.dat.LOG1

Users\Classic .NET AppPool\ntuser.dat.LOG2

Users\Default\NTUSER.DAT.LOG2

Users\Default\NTUSER.DAT.LOG1

Users\emsadmin\AppData\Local\ConnectedDevicesPlatform\CDPTraces.log

Users\emsadmin\AppData\Local\Microsoft\Internet Explorer\ie4uinit-UserConfig.log

Users\emsadmin\AppData\Local\Microsoft\Internet Explorer\ie4uinit-ClearIconCache.log

Users\emsadmin\AppData\Local\Microsoft\SQL Server Management Studio\14.0\ComponentModelCache\Microsoft.VisualStudio.Default.catalo

Users\emsadmin\AppData\Local\Microsoft\Windows\SettingSync\metastore\edbtmp.log

Users\emsadmin\AppData\Local\Microsoft\Windows\SettingSync\metastore\edb00001.log

Users\emsadmin\AppData\Local\Microsoft\Windows\SettingSync\metastore\edb00002.log

Users\emsadmin\AppData\Local\Microsoft\Windows\SettingSync\metastore\edb.log

Users\emsadmin\AppData\Local\Microsoft\Windows\WebCache\V010001C.log

Users\emsadmin\AppData\Local\Microsoft\Windows\WebCache\V010001D.log

Users\emsadmin\AppData\Local\Microsoft\Windows\WebCache\V010001E.log

Users\emsadmin\AppData\Local\Microsoft\Windows\WebCache\V01.log

Users\emsadmin\AppData\Local\Microsoft\Windows\WebCache\V01tmp.log

Users\emsadmin\AppData\Local\Microsoft\Windows\Windows Anytime Upgrade\Upgrade.log

Users\emsadmin\AppData\Local\Microsoft\Windows\UsrClass dat.LOG1

Users\emsadmin\AppData\Local\Microsoft\Windows\UsrClass dat.LOG2

Users\emsadmin\AppData\Local\Packages\Microsoft.AAD.BrokerPlugin_cw5n1h2txyewy\Settings\settings dat.LOG1

Users\emsadmin\AppData\Local\Packages\Microsoft.AAD.BrokerPlugin_cw5n1h2txyewy\Settings\settings dat.LOG2

Users\emsadmin\AppData\Local\Packages\Microsoft.AccountsControl_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\emsadmin\AppData\Local\Packages\Microsoft.AccountsControl_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\emsadmin\AppData\Local\Packages\Microsoft.BioEnrollment_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\emsadmin\AppData\Local\Packages\Microsoft.BioEnrollment_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\emsadmin\AppData\Local\Packages\Microsoft.LockApp_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\emsadmin\AppData\Local\Packages\Microsoft.LockApp_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\emsadmin\AppData\Local\Packages\Microsoft.Windows.Apprep.ChxApp_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\emsadmin\AppData\Local\Packages\Microsoft.Windows.Apprep.ChxApp_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\emsadmin\AppData\Local\Packages\Microsoft.Windows.AssignedAccessLockApp_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\emsadmin\AppData\Local\Packages\Microsoft.Windows.AssignedAccessLockApp_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\emsadmin\AppData\Local\Packages\Microsoft.Windows.CloudExperienceHost_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\emsadmin\AppData\Local\Packages\Microsoft.Windows.CloudExperienceHost_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\emsadmin\AppData\Local\Packages\Microsoft.Windows.Cortana_cw5n1h2txyewy\AC\Temp\StructuredQuery.log

Users\emsadmin\AppData\Local\Packages\Microsoft.Windows.Cortana_cw5n1h2txyewy\AppData\Indexed DB\edb.log

Users\emsadmin\AppData\Local\Packages\Microsoft.Windows.Cortana_cw5n1h2txyewy\AppData\Indexed DB\edbtmp.log

Users\emsadmin\AppData\Local\Packages\Microsoft.Windows.Cortana_cw5n1h2txyewy\AppData\Indexed DB\edb00006.log

Users\emsadmin\AppData\Local\Packages\Microsoft.Windows.Cortana_cw5n1h2txyewy\AppData\Indexed DB\edb00007.log

Users\emsadmin\AppData\Local\Packages\Microsoft.Windows.Cortana_cw5n1h2txyewy\AppData\Indexed DB\edb00008.log

Users\emsadmin\AppData\Local\Packages\Microsoft.Windows.Cortana_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\emsadmin\AppData\Local\Packages\Microsoft.Windows.Cortana_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\emsadmin\AppData\Local\Packages\Microsoft.Windows.SecondaryTileExperience_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\emsadmin\AppData\Local\Packages\Microsoft.Windows.SecondaryTileExperience_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\emsadmin\AppData\Local\Packages\Microsoft.Windows.ShellExperienceHost_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\emsadmin\AppData\Local\Packages\Microsoft.Windows.ShellExperienceHost_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\emsadmin\AppData\Local\Packages\Microsoft.XboxGameCallableUI_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\emsadmin\AppData\Local\Packages\Microsoft.XboxGameCallableUI_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\emsadmin\AppData\Local\Packages\windows.immersivecontrolpanel_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\emsadmin\AppData\Local\Packages\windows.immersivecontrolpanel_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\emsadmin\AppData\Local\Packages\Windows.MiracastView_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\emsadmin\AppData\Local\Packages\Windows.MiracastView_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\emsadmin\AppData\Local\Packages\Windows.PrintDialog_cw5n1h2txyewy

Users\emsadmin\AppData\Local\Packages\Windows.PrintDialog_cw5n1h2txyewy\Settings\settings.dat.LOG1

Users\emsadmin\AppData\Local\Packages\Windows.PrintDialog_cw5n1h2txyewy\Settings\settings.dat.LOG2

Users\emsadmin\AppData\Local\Temp\f92aeb63-07b8-4662-94b4-f4bcccba37ec\baseutils.log

Users\emsadmin\AppData\Local\Temp\SSMS\Ssms_20190610_131541566_PID2136_logFile.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_32_sql_ssms_loc_x64_Loc.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_3_DACFramework.msi.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_4_SQLServerBestPracticesPolicies.msi.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_5_TSqlLanguageService_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_6_sql_diag_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_7_adalsql_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_22_sql_as_oledb_x86.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_23_sql_common_core_x86.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_24_sql_common_core_loc_x86.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_30_sql_ssms_extensions_loc_x86.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_31_sql_ssms_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_33_sql_tools_connectivity_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_18_smo_extensions_loc_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_2_msodbcsql.msi.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_8_conn_info_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_9_conn_info_loc_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_0_SQLSysClrTypes.msi.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_1_sqlncli.msi.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_10_sql_batchparser_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_11_sql_xevent_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_12_sql_xevent_loc_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_13_sql_is_scale_management_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_14_sql_is_scale_management_loc_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_15_sql_dmf_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_16_sql_dmf_loc_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_17_smo_extensions_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_19_smo_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_20_smo_loc_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_34_sql_tools_connectivity_loc_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_21_sql_as_oledb_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_35_ssms_rs_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_36_ssms_as_x64.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_29_sql_ssms_extensions_x86.log

Users\emsadmin\AppData\Local\Temp\SsmsSetup\SSMS-Setup-ENU_20190610125637_37_SsmsPostInstall_x64.log

Users\emsadmin\AppData\Local\Temp\VSD329B.tmp\install.log

Users\emsadmin\AppData\Local\Temp\VSDA070.tmp\install.log

Users\emsadmin\AppData\Local\Temp\VsHub\Microsoft.VisualStudio.ExtensionManager.HubServiceModule-kitajpem.rgb.log

Users\emsadmin\AppData\Local\Temp\SqlSetup_1.log

Users\emsadmin\AppData\Local\Temp\dd_vcredist_amd64_20190610120224_000_vcRuntimeMinimum_x64.log

Users\emsadmin\AppData\Local\Temp\dd_vcredist_amd64_20190610120224_001_vcRuntimeAdditional_x64.log

Users\emsadmin\AppData\Local\Temp\SqlSetup.log

Users\emsadmin\AppData\Local\Temp\StructuredQuery.log

Users\emsadmin\AppData\Local\Temp\dd_vcredist_amd64_20190610120224.log

Users\emsadmin\AppData\Local\Temp\dd_vcredist_x86_20190610120041.log

Users\emsadmin\AppData\Local\Temp\dd_vcredist_x86_20190610120041_0_vcRuntimeMinimum_x86.log

Users\emsadmin\AppData\Local\Temp\dd_vcredist_x86_20190610120041_1_vcRuntimeAdditional_x86.log

Users\emsadmin\AppData\Local\Temp\dd_vcredist_amd64_20190610120118.log

Users\emsadmin\AppData\Local\Temp\dd_vcredist_amd64_20190610120118_0_vcRuntimeMinimum_x64.log

Users\emsadmin\AppData\Local\Temp\wmsetup.log

Users\emsadmin\AppData\Local\Temp\dd_vcredist_amd64_20190610120118_1_vcRuntimeAdditional_x64.log

Users\emsadmin\AppData\Local\Temp\dd_vcredist_x86_20190610120157.log

Users\emsadmin\AppData\Local\Temp\dd_vcredist_x86_20190610120157_000_vcRuntimeMinimum_x86.log

Users\emsadmin\AppData\Local\Temp\dd_vcredist_x86_20190610120157_001_vcRuntimeAdditional_x86.log

Users\emsadmin\AppData\Local\Temp\dd_vcredist_x86_20190610125911.log

Users\emsadmin\AppData\Local\Temp\dd_vcredist_amd64_20190610125911.log

Users\emsadmin\AppData\Local\Temp\dd_vcredist_amd64_20190610125912.log

Users\emsadmin\AppData\Local\Temp\dd_vcredist_x86_20190610130115.log

Users\emsadmin\AppData\Local\Temp\MpSigStub.log

Users\emsadmin\AppData\Local\TileDataLayer\Database\EDB.log

Users\emsadmin\AppData\Local\TileDataLayer\Database\EDBtmp.log

Users\emsadmin\AppData\Local\TileDataLayer\Database\EDB00003.log

Users\emsadmin\Documents\EMS\Log\Debug.log

Users\emsadmin\Documents\EMS\Log\Info.log

Users\emsadmin\ntuser.dat.LOG1

Users\emsadmin\ntuser.dat.LOG2

Users\emsadmin01\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1

Users\emsadmin01\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2

Users\emsadmin01\ntuser.dat.LOG1

Users\emsadmin01\ntuser.dat.LOG2

Users\emsadmin02\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1

Users\emsadmin02\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2

Users\emsadmin02\ntuser.dat.LOG1

Users\emsadmin02\ntuser.dat.LOG2

Users\emsasuser\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1

Users\emsasuser\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2

Users\emsasuser\ntuser.dat.LOG1

Users\emsasuser\ntuser.dat.LOG2

Users\emssqluser\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1

Users\emssqluser\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2

Users\emssqluser\ntuser.dat.LOG1

Users\emssqluser\ntuser.dat.LOG2

Users\emsuser01\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1

Users\emsuser01\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2

Users\emsuser01\ntuser.dat.LOG1

Users\emsuser01\ntuser.dat.LOG2

Users\emsuser02\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1

Users\emsuser02\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2

Users\emsuser02\ntuser.dat.LOG1

Users\emsuser02\ntuser.dat.LOG2

Users\MSSQLSERVER\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1

Users\MSSQLSERVER\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2

Users\MSSQLSERVER\ntuser.dat.LOG1

Users\MSSQLSERVER\ntuser.dat.LOG2

Users\ReportServer\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1

Users\ReportServer\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2

Users\ReportServer\ntuser.dat.LOG1

Users\ReportServer\ntuser.dat.LOG2

Users\SQLSERVERAGENT\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1

Users\SQLSERVERAGENT\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2

Users\SQLSERVERAGENT\ntuser.dat.LOG1

Users\SQLSERVERAGENT\ntuser.dat.LOG2

Users\SQLTELEMETRY\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1

Users\SQLTELEMETRY\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2

Users\SQLTELEMETRY\ntuser.dat.LOG1

Users\SQLTELEMETRY\ntuser.dat.LOG2

VirtualDirectories\EMSApplicationServer\Log\Error.4.log

VirtualDirectories\EMSApplicationServer\Log\Error.log

VirtualDirectories\EMSApplicationServer\Log\Error.12.log

VirtualDirectories\EMSApplicationServer\Log\Error.14.log

VirtualDirectories\EMSApplicationServer\Log\Error.5.log

VirtualDirectories\EMSApplicationServer\Log\Error.8.log

VirtualDirectories\EMSApplicationServer\Log\Error.0.log

VirtualDirectories\EMSApplicationServer\Log\Error.11.log

VirtualDirectories\EMSApplicationServer\Log\Error.10.log

VirtualDirectories\EMSApplicationServer\Log\Error.9.log

VirtualDirectories\EMSApplicationServer\Log\Error.3.log

VirtualDirectories\EMSApplicationServer\Log\Error.6.log

VirtualDirectories\EMSApplicationServer\Log\Error.7.log

VirtualDirectories\EMSApplicationServer\Log\Error.2.log

VirtualDirectories\EMSApplicationServer\Log\Error.1.log

VirtualDirectories\EMSApplicationServer\Log\Error.13.log

VirtualDirectories\EMSApplicationServer\Log\Warn.log

VirtualDirectories\EMSApplicationServer\Log\Error.14.log

Windows\appcompat\Programs\Amcache.hve.LOG1

Windows\appcompat\Programs\Amcache.hve.LOG2

Windows\assembly\GAC_MSIL\System.IO.Log

Windows\assembly\NativeImages_v2.0.50727_32\System.IO.Log

Windows\assembly\NativeImages_v2.0.50727_64\System.IO.Log

Windows\assembly\NativeImages_v4.0.30319_32\System.IO.Log

Windows\assembly\NativeImages_v4.0.30319_64\System.IO.Log

Windows\debug\sammui.log

Windows\debug\PASSWD.LOG

Windows\debug\NetSetup.LOG

Windows\Dell\UpdatePackage\log\BrcmSetup.log

Windows\INF\setupapi.dev.log

Windows\INF\setupapi.setup.log

Windows\Logs\CBS\CBS.log

Windows\Logs\CBS\CbsPersist_20191001155012.log

Windows\Logs\CBS\CbsPersist_20190625180445.log

Windows\Logs\DISM\dism.log

Windows\Logs\DPX\setupact.log

Windows\Logs\DPX\setuperr.log

Windows\Logs\SetupCleanupTask\setuperr.log

Windows\Logs\SetupCleanupTask\setupact.log

Windows\Microsoft.NET\assembly\GAC_MSIL\Microsoft.SqlServer.TransferLoginsTask

Windows\Microsoft.NET\assembly\GAC_MSIL\Microsoft.SqlServer.TransferLoginsTaskUI

Windows\Microsoft.NET\assembly\GAC_MSIL\Microsoft.VisualStudio.Dialogs

Windows\Microsoft.NET\assembly\GAC_MSIL\Microsoft.VisualStudio.ImageCatalog

Windows\Microsoft.NET\assembly\GAC_MSIL\Microsoft.VisualStudio.ProductKeyDialog

Windows\Microsoft.NET\assembly\GAC_MSIL\Microsoft.VisualStudio.Text.Logic

Windows\Microsoft.NET\assembly\GAC_MSIL\System.IO.Log

Windows\Microsoft.NET\Framework\v2.0.50727\ngen.log

Windows\Microsoft.NET\Framework\v4.0.30319\ngen.log

Windows\Microsoft.NET\Framework\v4.0.30319\ngen.old.log

Windows\Microsoft.NET\Framework\v4.0.30319\ngen.old.log

Windows\Microsoft.NET\Framework64\v2.0.50727\ngen.log

Windows\Microsoft.NET\Framework64\v4.0.30319\ngen.log

Windows\Microsoft.NET\Framework64\v4.0.30319\ngen.old.log

Windows\Panther\UnattendGC\setupact.log

Windows\Panther\UnattendGC\setuperr.log

Windows\Panther\DDACLSys.log

Windows\Panther\cbs.log

Windows\Panther\setupact.log

Windows\Panther\setuperr.log

Windows\Performance\WinSAT\winsat.log

Windows\security\database\edb.log

Windows\security\database\edbtmp.log

Windows\security\logs\scesetup.log

Windows\ServiceProfiles\LocalService\NTUSER.DAT.LOG1

Windows\ServiceProfiles\LocalService\NTUSER.DAT.LOG2

Windows\ServiceProfiles\NetworkService\AppData\Local\Temp\MpCmdRun.log

Windows\ServiceProfiles\NetworkService\debug\NetSetup.LOG

Windows\ServiceProfiles\NetworkService\NTUSER.DAT.LOG2

Windows\ServiceProfiles\NetworkService\NTUSER.DAT.LOG1

Windows\SoftwareDistribution\DataStore\Logs\edb00222.log

Windows\SoftwareDistribution\DataStore\Logs\edb00227.log

Windows\SoftwareDistribution\DataStore\Logs\edb00223.log

Windows\SoftwareDistribution\DataStore\Logs\edb00224.log

Windows\SoftwareDistribution\DataStore\Logs\edb00225.log

Windows\SoftwareDistribution\DataStore\Logs\edb0022A.log

Windows\SoftwareDistribution\DataStore\Logs\edb0022C.log

Windows\SoftwareDistribution\DataStore\Logs\edb0022D.log

Windows\SoftwareDistribution\DataStore\Logs\edb00230.log

Windows\SoftwareDistribution\DataStore\Logs\edb.log

Windows\SoftwareDistribution\DataStore\Logs\edbtmp.log

Windows\SoftwareDistribution\DataStore\Logs\edb00226.log

Windows\SoftwareDistribution\DataStore\Logs\edb0022B.log

Windows\SoftwareDistribution\DataStore\Logs\edb00228.log

Windows\SoftwareDistribution\DataStore\Logs\edb00229.log

Windows\SoftwareDistribution\DataStore\Logs\edb0022E.log

Windows\SoftwareDistribution\DataStore\Logs\edb0022F.log

Windows\SoftwareDistribution\ReportingEvents.log

Windows\System32\catroot2\edb00018.log

Windows\System32\catroot2\edb0001B.log

Windows\System32\catroot2\edb0001C.log

Windows\System32\catroot2\edb0001D.log

Windows\System32\catroot2\edb.log

Windows\System32\catroot2\edb00013.log

Windows\System32\catroot2\edb00014.log

Windows\System32\catroot2\edb00015.log

Windows\System32\catroot2\edb00016.log

Windows\System32\catroot2\edb00017.log

Windows\System32\catroot2\edb00019.log

Windows\System32\catroot2\edb0001A.log

Windows\System32\catroot2\edbtmp.log

Windows\System32\config\RegBack\SECURITY.LOG1

Windows\System32\config\RegBack\SECURITY.LOG2

Windows\System32\config\RegBack\SOFTWARE.LOG1

Windows\System32\config\RegBack\SOFTWARE.LOG2

Windows\System32\config\RegBack\SYSTEM.LOG1

Windows\System32\config\RegBack\SYSTEM.LOG2

Windows\System32\config\RegBack\DEFAULT.LOG1

Windows\System32\config\RegBack\DEFAULT.LOG2

Windows\System32\config\RegBack\SAM.LOG1

Windows\System32\config\RegBack\SAM.LOG2

Windows\System32\config\BBI.LOG1

Windows\System32\config\BCD-Template.LOG

Windows\System32\config\DEFAULT.LOG2

Windows\System32\config\ELAM.LOG1

Windows\System32\config\SAM.LOG2

Windows\System32\config\SECURITY.LOG2

Windows\System32\config\DEFAULT.LOG1

Windows\System32\config\DRIVERS.LOG1

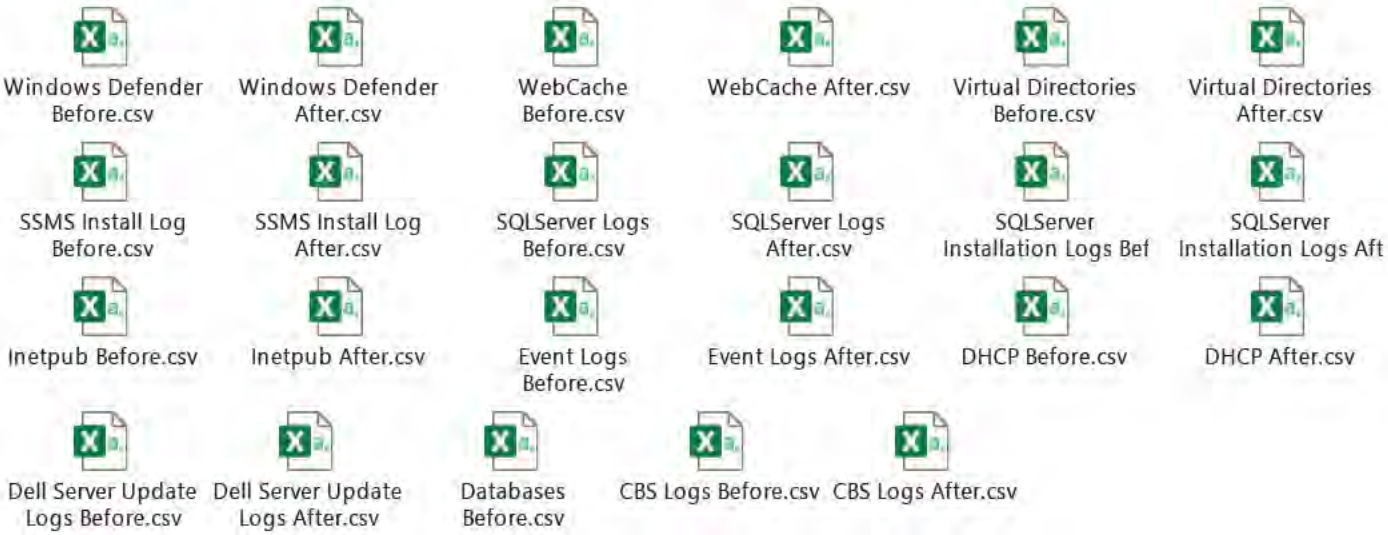
Windows\System32\config\SAM.LOG1

Windows\System32\config\COMPONENTS.LOG2

Windows\System32\config\SECURITY.LOG1

APPENDIX B. SUPPORTING DOCUMENTATION: FILE DETAILS AND HASH SETS FOR SCREENSHOTS

The files below provide integrity data for the graphic screenshots in this document. Each comma-separated-variable (csv) file listed here contain the file name with its full path (e.g., directory structure) and message digest hash values (MD5 and SHA1 algorithms). Due to the length of the data contained in these files, they are provided in a compact disc (CD) addendum to this report.



APPENDIX C. MICROSOFT EVENT LOG FILES

Files in this list were ALL present in the EMS Server Before image. Files listed in RED were deleted or overwritten. Significantly, from their filenames alone, they are OBVIOUSLY Election-Related Records, "Archive-EMS-System-..."

Logs\Microsoft-Windows-Kernel-WHEA%4Errors.evtx
Logs\Key Management Service.evtx
Logs\Application.evtx
Logs\HardwareEvents.evtx
Logs\Internet Explorer.evtx
Logs\Microsoft-Client-Licensing-Platform%4Admin.evtx
Logs\Microsoft-Windows-Application-Experience%4Program-Compatibility-Assistant.evtx
Logs\Microsoft-Windows-AppModel-Runtime%4Admin.evtx
Logs\Microsoft-Windows-AppReadiness%4Admin.evtx
Logs\Microsoft-Windows-AppXDeployment%4Operational.evtx
Logs\Microsoft-Windows-AppXDeploymentServer%4Restricted.evtx
Logs\Microsoft-Windows-CodeIntegrity%4Operational.evtx
Logs\Microsoft-Windows-Containers-Wcifs%4Operational.evtx
Logs\Microsoft-Windows-Containers-Wcnfs%4Operational.evtx
Logs\Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider%4Admin.evtx
Logs\Microsoft-Windows-Crypto-DPAPI%4BackUpKeySvc.evtx
Logs\Microsoft-Windows-Crypto-DPAPI%4Operational.evtx
Logs\Microsoft-Windows-DeviceSetupManager%4Admin.evtx
Logs\Microsoft-Windows-DeviceSetupManager%4Operational.evtx
Logs\Microsoft-Windows-ApplicationResourceManagementSystem%4Operational.evtx
Logs\Microsoft-Windows-Dhcp-Client%4Admin.evtx
Logs\Microsoft-Windows-Dhcpv6-Client%4Admin.evtx
Logs\Microsoft-Windows-Hyper-V-Guest-Drivers%4Admin.evtx
Logs\Microsoft-Windows-International%4Operational.evtx
Logs\Microsoft-Windows-AppReadiness%4Operational.evtx
Logs\Microsoft-Windows-Kernel-Boot%4Operational.evtx
Logs\Microsoft-Windows-Kernel-IO%4Operational.evtx
Logs\Microsoft-Windows-Kernel-EventTracing%4Admin.evtx
Logs\Microsoft-Windows-Kernel-Power%4Thermal-Operational.evtx
Logs\Microsoft-Windows-Kernel-ShimEngine%4Operational.evtx
Logs\Microsoft-Windows-Kernel-StoreMgr%4Operational.evtx
Logs\Microsoft-Windows-AppXDeploymentServer%4Operational.evtx
Logs\Microsoft-Windows-Kernel-WHEA%4Operational.evtx
Logs\Microsoft-Windows-Known Folders API Service.evtx
Logs\Microsoft-Windows-LiveId%4Operational.evtx
Logs\Microsoft-Windows-MUI%4Admin.evtx
Logs\Microsoft-Windows-GroupPolicy%4Operational.evtx
Logs\Microsoft-Windows-MUI%4Operational.evtx
Logs\Microsoft-Windows-NCSI%4Operational.evtx
Logs\Microsoft-Windows-NetworkProfile%4Operational.evtx

Logs\Microsoft-Windows-Ntfs%4Operational.evtx
Logs\Microsoft-Windows-Ntfs%4WHC.evtx
Logs\Microsoft-Windows-Program-Compatibility-Assistant%4CompatAfterUpgrade.evtx
Logs\Microsoft-Windows-RemoteDesktopServices-RdpCoreTS%4Admin.evtx
Logs\Microsoft-Windows-SettingSync%4Debug.evtx
Logs\Microsoft-Windows-RemoteDesktopServices-RdpCoreTS%4Operational.evtx
Logs\Microsoft-Windows-Kernel-PnP%4Configuration.evtx
Logs\Microsoft-Windows-SettingSync%4Operational.evtx
Logs\Microsoft-Windows-Shell-Core%4ActionCenter.evtx
Logs\Microsoft-Windows-Shell-Core%4AppDefaults.evtx
Logs\Microsoft-Windows-Shell-Core%4LogonTasksChannel.evtx
Logs\Microsoft-Windows-Shell-Core%4Operational.evtx
Logs\Microsoft-Windows-SmbClient%4Connectivity.evtx
Logs\Microsoft-Windows-SMBClient%4Operational.evtx
Logs\Microsoft-Windows-SmbClient%4Security.evtx
Logs\Microsoft-Windows-SMBServer%4Audit.evtx
Logs\Microsoft-Windows-SMBServer%4Connectivity.evtx
Logs\Microsoft-Windows-SMBServer%4Operational.evtx
Logs\Microsoft-Windows-SMBServer%4Security.evtx
Logs\Microsoft-Windows-SMBWitnessClient%4Admin.evtx
Logs\Microsoft-Windows-SMBWitnessClient%4Informational.evtx
Logs\Microsoft-Windows-StateRepository%4Operational.evtx
Logs\Microsoft-Windows-StateRepository%4Restricted.evtx
Logs\Microsoft-Windows-TaskScheduler%4Maintenance.evtx
Logs\Microsoft-Windows-TerminalServices-LocalSessionManager%4Admin.evtx
Logs\Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational.evtx
Logs\Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Admin.evtx
Logs\Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational.evtx
Logs\Microsoft-Windows-Store%4Operational.evtx
Logs\Microsoft-Windows-UniversalTelemetryClient%4Operational.evtx
Logs\Microsoft-Windows-User Profile Service%4Operational.evtx
Logs\Microsoft-Windows-UserPnp%4ActionCenter.evtx
Logs\Microsoft-Windows-UserPnp%4DeviceInstall.evtx
Logs\Microsoft-Windows-VolumeSnapshot-Driver%4Operational.evtx
Logs\Microsoft-Windows-Wcmsvc%4Operational.evtx
Logs\Microsoft-Windows-Windows Defender%4Operational.evtx
Logs\Microsoft-Windows-Windows Defender%4WHC.evtx
Logs\Microsoft-Windows-Windows Firewall With Advanced Security%4ConnectionSecurity.evtx
Logs\Microsoft-Windows-WinINet-Config%4ProxyConfigChanged.evtx
Logs\Microsoft-Windows-Winlogon%4Operational.evtx
Logs\Microsoft-Windows-WinRM%4Operational.evtx
Logs\Setup.evtx
Logs\Windows PowerShell.evtx

Logs\Microsoft-Windows-Windows Firewall With Advanced Security%4Firewall.evtx
Logs\Microsoft-Windows-WMI-Activity%4Operational.evtx
Logs\System.evtx
Logs\Security.evtx
Windows\System32\winevt\Logs\Setup.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-02-27-12-21-20-622.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-07-11-20-46-32-189.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-06-30-13-45-03-347.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-07-08-02-26-04-899.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-10-30-19-26-37-188.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-07-04-08-05-33-867.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-07-30-16-29-10-602.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-07-15-15-07-04-381.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-03-02-02-52-11-569.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-04-19-00-10-16-214.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-07-26-22-08-42-827.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-12-11-22-05-26-089.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-02-23-04-20-21-835.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-05-26-12-11-25-223.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-04-15-07-09-24-325.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-06-04-04-35-23-707.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-04-07-21-05-41-859.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-06-18-19-18-33-633.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-02-09-23-20-20-681.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-06-07-22-53-09-400.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-19-09-05-03-069.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-19-03-17-30-918.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-10-02-11-09-22-083.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-02-09-15-51-43-896.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-02-09-19-15-04-259.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-04-28-04-14-58-545.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-04-06-04-10-43-774.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-03-24-00-59-56-063.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-01-08-17-03-22-249.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-11-10-09-23-03-203.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-11-06-16-37-56-482.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-12-02-15-06-36-405.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-07-05-51-17-641.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-03-06-06-07-29-506.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-10-27-06-12-01-889.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-11-14-02-20-35-061.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-03-16-20-09-20-723.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-09-26-09-07-58-407.evtx

Windows\System32\winevt\Logs\Archive-EMS System-2020-07-08-10-16-08-709.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-26-06-16-16-735.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-06-21-16-36-38-559.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-11-03-08-53-17-828.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-10-23-15-37-23-347.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-11-13-00-00-07-540.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-21-03-24-37-573.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-01-15-03-21-17-842.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-12-10-01-06-03-529.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-11-25-05-09-12-916.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-01-22-13-12-21-043.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-12-06-08-06-21-993.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-11-05-17-16-37-197.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-16-23-08-11-827.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-04-17-01-19-18-484.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-21-01-39-07-647.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-03-04-12-10-17-214.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-07-12-02-56-47-489.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-21-07-26-54-297.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-04-04-04-03-42-737.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-11-16-16-31-58-059.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-11-21-12-09-41-781.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-12-28-14-04-05-771.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-03-12-22-02-14-487.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-12-13-18-05-49-770.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-11-17-19-10-01-322.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-05-14-13-33-324.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-14-13-10-56-695.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-11-28-22-08-50-835.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-12-17-11-04-36-787.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-01-05-00-03-39-390.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-01-12-10-03-10-333.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-19-20-40-41-039.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-04-24-11-15-42-872.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-09-18-17-48-53-388.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-11-09-10-14-15-715.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-03-31-11-01-47-908.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-01-27-08-29-08-399.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-09-14-23-29-04-158.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-12-24-21-04-12-832.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-12-21-04-04-25-803.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-01-01-07-03-50-651.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-14-18-29-08-151.evtx

Windows\System32\winevt\Logs\Archive-EMS System-2021-05-11-21-02-29-740.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-03-11-33-19-283.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-01-19-20-02-44-037.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-28-20-58-32-283.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-01-16-03-02-57-774.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-01-21-14-15-340.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-13-08-12-31-149.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-04-22-17-12-11-701.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-16-05-45-04-636.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-03-05-12-32-06-091.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-03-26-13-42-04-162.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-03-16-14-59-42-045.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-01-23-13-23-46-471.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-05-18-23-53-08-392.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-21-13-14-26-512.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-03-27-17-59-53-690.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-11-20-09-29-41-350.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-03-20-07-59-19-878.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-01-31-02-47-11-038.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-01-29-23-11-26-924.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-03-09-05-29-49-411.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-01-26-06-11-56-096.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-23-00-00-39-858.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-14-18-58-50-004.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-06-28-08-06-44-934.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-04-11-14-07-34-718.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-05-15-07-27-29-016.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-07-04-17-16-43-223.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-06-24-22-45-04-435.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-04-20-18-16-33-823.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-18-12-49-02-987.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-04-13-08-24-43-079.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-09-22-12-08-49-154.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-03-30-02-15-24-619.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-05-22-17-53-50-782.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-06-11-17-12-21-460.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-01-18-20-12-44-811.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-04-02-14-34-04-967.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-09-07-10-51-04-386.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-09-11-05-09-09-286.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-31-12-27-41-741.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-22-20-55-04-936.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-07-01-03-27-07-105.evtx

Windows\System32\winevt\Logs\Archive-EMS System-2021-04-09-17-04-07-509.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-09-07-12-48-391.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-10-11-00-11-12-292.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-15-12-21-57-907.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-11-03-08-03-17-087.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-22-18-13-07-673.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-08-27-17-53-57-312.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-09-30-01-16-19-620.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-20-19-51-20-073.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-13-02-24-44-262.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-15-23-57-17-682.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-12-08-20-31-15-022.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-12-09-21-34-01-652.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-02-02-16-11-00-907.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-12-10-17-14-18-337.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-17-16-31-18-634.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-12-09-17-35-55-190.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-12-10-17-26-50-697.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-12-08-21-07-21-169.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2020-12-09-22-01-49-473.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-18-15-41-56-864.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-02-09-17-14-16-397.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-02-09-17-25-20-742.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-02-09-17-33-50-215.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-02-09-17-46-57-607.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-16-17-20-24-507.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-14-07-23-24-216.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-12-14-51-41-139.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-18-04-06-37-355.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-18-09-54-24-839.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-19-14-52-50-172.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-22-06-37-33-489.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-12-03-16-22-000.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-21-19-02-14-166.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-22-00-50-01-529.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-14-01-35-37-340.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-13-14-00-17-879.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-23-17-23-46-597.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-12-20-37-42-553.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-12-09-03-54-186.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-17-10-43-31-360.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-17-04-55-44-339.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-17-22-18-50-801.evtx

Windows\System32\winevt\Logs\Archive-EMS System-2021-05-13-19-47-50-347.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-15-06-34-10-708.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-20-02-28-13-043.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-18-21-29-43-807.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-15-00-46-23-325.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-15-18-09-30-390.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-20-14-03-47-942.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-23-11-36-14-332.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-16-11-32-36-872.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-20-08-16-00-636.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-23-05-48-27-189.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2021-05-22-12-25-20-731.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-UniversalTelemetryClient%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-NetworkProfile%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-StoreMgr%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SMBServer%4Operational.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-07-19-09-27-36-681.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SMBServer%4Security.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-07-23-03-48-10-012.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SMBServer%4Connectivity.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-08-14-17-50-17-089.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-08-22-06-31-22-632.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-08-03-10-49-41-423.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Dhcp-Server%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Dhcp-Server%4FilterNotifications.evtx
Windows\System32\winevt\Logs\DhcpAdminEvents.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-08-26-00-51-55-728.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-08-07-05-09-14-598.evtx
Windows\System32\winevt\Logs\DNS Server.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-DNSServer%4Audit.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-08-10-23-29-48-856.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-08-18-12-10-49-482.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-09-02-13-32-58-546.evtx
Windows\System32\winevt\Logs\Archive-EMS System-2019-08-29-19-12-25-021.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SMBServer%4Audit.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-TaskScheduler%4Maintenance.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-VDRVROOT%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-VHDMP-Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-AppXDeploymentServer%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-AppXDeploymentServer%4Restricted.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-International%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-MUI%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-MUI%4Admin.evtx

Windows\System32\winevt\Logs\Microsoft-Windows-Windows Firewall With Advanced Security%4ConnectionSe
Windows\System32\winevt\Logs\Microsoft-Windows-Iphlpsvc%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-WMI-Activity%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-AppReadiness%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-AppReadiness%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-ApplicationResourceManagementSystem%4Operational.ev
Windows\System32\winevt\Logs\Microsoft-Client-Licensing-Platform%4Admin.evtx
Windows\System32\winevt\Logs\System.evtx
Windows\System32\winevt\Logs\Application.evtx
Windows\System32\winevt\Logs\Security.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Dhcpv6-Client%4Admin.evtx
Windows\System32\winevt\Logs\Windows PowerShell.evtx
Windows\System32\winevt\Logs\Key Management Service.evtx
Windows\System32\winevt\Logs\Internet Explorer.evtx
Windows\System32\winevt\Logs\HardwareEvents.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Wcmsvc%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Application-Experience%4Program-Compatibility-Assistant.
Windows\System32\winevt\Logs\Microsoft-Windows-Program-Compatibility-Assistant%4CompatAfterUpgrade.e
Windows\System32\winevt\Logs\Microsoft-Windows-Dhcp-Client%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-LocalSessionManager%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-WinRM%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Windows Defender%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Windows Defender%4WHC.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-UserPnp%4DeviceInstall.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-UserPnp%4ActionCenter.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider%4Adr
Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-Power%4Thermal-Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-Boot%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-StateRepository%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-StateRepository%4Restricted.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-PnP%4Configuration.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-ShimEngine%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Ntfs%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Ntfs%4WHC.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-VolumeSnapshot-Driver%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-IO%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-CodeIntegrity%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-WHEA%4Errors.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-WHEA%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Windows Firewall With Advanced Security%4Firewall.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-NCSI%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-WinINet-Config%4ProxyConfigChanged.evtx

Windows\System32\winevt\Logs\Microsoft-Windows-Crypto-DPAPI%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Crypto-DPAPI%4BackUpKeySvc.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-DeviceSetupManager%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-DeviceSetupManager%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Containers-Wcifs%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Containers-Wcnfs%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-GroupPolicy%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-User Profile Service%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-LiveId%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SMBClient%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SmbClient%4Connectivity.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SmbClient%4Security.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Winlogon%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SettingSync%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SettingSync%4Debug.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-DateTimeControlPanel%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Known Folders API Service.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Shell-Core%4ActionCenter.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Shell-Core%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Shell-Core%4AppDefaults.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Shell-Core%4LogonTasksChannel.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-AppXDeployment%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-AppModel-Runtime%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-ServerManager-DeploymentProvider%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Store%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-PushNotification-Platform%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-PushNotification-Platform%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-TWinUI%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Application-Experience%4Program-Telemetry.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Application-Experience%4Program-Inventory.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Application-Experience%4Steps-Recorder.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Application-Experience%4Program-Compatibility-Troublesh
Windows\System32\winevt\Logs\Microsoft-Windows-Security-SPP-UX-Notifications%4ActionCenter.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-BackgroundTaskInfrastructure%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-ServerManager-MultiMachine%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-ServerManager-MultiMachine%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Forwarding%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-ServerManager-MgmtProvider%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Resource-Exhaustion-Detector%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-DataIntegrityScan%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-DataIntegrityScan%4CrashRecovery.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Diagnosis-Scheduled%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-HomeGroup Control Panel%4Operational.evtx

Windows\System32\winevt\Logs\Microsoft-Windows-Shell-ConnectedAccountState%4ActionCenter.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-StorageSpaces-ManagementAgent%4WHC.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Diagnosis-DPS%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Storage-ClassPnP%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-RestartManager%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Diagnosis-PLA%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-CAPI2%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-WindowsUpdateClient%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-NlaSvc%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-PowerShell%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-PowerShell%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Diagnosis-PCW%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Storage-Storport%4Operational.evtx
Windows\System32\winevt\Logs\EMS System.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Application Server-Applications%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Application Server-Applications%4Operational.evtx
Windows\System32\winevt\Logs\DVS Adjudication.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-CloudStorageWizard%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-AppV-Client%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-AppV-Client%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-AppV-Client%4Virtual Applications.evtx
Windows\System32\winevt\Logs\Microsoft-Rdms-UI%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Rdms-UI%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-User Experience Virtualization-Agent Driver%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-User Experience Virtualization-App Agent%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-User Experience Virtualization-IPC%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-User Experience Virtualization-SQM Uploader%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-AAD%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-All-User-Install-Agent%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-AllJoyn%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-AppHost%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-AppID%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-ApplicabilityEngine%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-AppLocker%4EXE and DLL.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-AppLocker%4MSI and Script.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-AppLocker%4Packaged app-Deployment.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-AppLocker%4Packaged app-Execution.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-AppxPackaging%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-AssignedAccess%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-AssignedAccessBroker%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Audio%4CaptureMonitor.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Audio%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Audio%4PlaybackManager.evtx

Windows\System32\winevt\Logs\Microsoft-Windows-Authentication User Interface%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Backup.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-BestPractices%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Biometrics%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Bits-Client%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Bluetooth-BthLEPrepairing%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Bluetooth-MTPEnum%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-BranchCacheSMB%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-CertificateServices-Deployment%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-CertificateServicesClient-Lifecycle-System%4Operational.ev
Windows\System32\winevt\Logs\Microsoft-Windows-CertificateServicesClient-Lifecycle-User%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Compat-Appraiser%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-CoreApplication%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-CorruptedFileRecovery-Client%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-CorruptedFileRecovery-Server%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-DAL-Provider%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-DeviceGuard%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Devices-Background%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-DeviceSync%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Diagnosis-Scripted%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Diagnosis-Scripted%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Diagnosis-ScriptedDiagnosticsProvider%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Diagnostics-Networking%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-DirectoryServices-Deployment%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-DiskDiagnostic%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-DiskDiagnosticDataCollector%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-DiskDiagnosticResolver%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-DSC%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-DSC%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-EapHost%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-EapMethods-RasChap%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-EapMethods-RasTls%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-EapMethods-Sim%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-EapMethods-Ttls%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-EDP-Audit-Regular%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-EDP-Audit-TCB%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-EmbeddedAppLauncher%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-EnrollmentPolicyWebService%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-EnrollmentWebService%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-EventCollector%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Fault-Tolerant-Heap%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-FederationServices-Deployment%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-FileServices-ServerManager-EventProvider%4Admin.evtx

Windows\System32\winevt\Logs\Microsoft-Windows-FileServices-ServerManager-EventProvider%4Operational.e
Windows\System32\winevt\Logs\Microsoft-Windows-FileShareShadowCopyProvider%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-FMS%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Folder Redirection%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-NdisImPlatform%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-GenericRoaming%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Help%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Hyper-V-Guest-Drivers%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-IdCtrls%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-IKE%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-International-RegionalOptionsControlPanel%4Operational.
Windows\System32\winevt\Logs\Microsoft-Windows-KdsSvc%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-ApphelpCache%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-EventTracing%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Kernel-WDI%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-LanguagePackSetup%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-ManagementTools-RegistryProvider%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-ManagementTools-TaskManagerProvider%4Operational.ev
Windows\System32\winevt\Logs\Microsoft-Windows-MemoryDiagnostics-Results%4Debug.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-MiStreamProvider%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Mobile-Broadband-Experience-Parser-Task%4Operational.
Windows\System32\winevt\Logs\Microsoft-Windows-Mobile-Broadband-Experience-SmsRouter%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Mprddm%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-MsLbfoProvider%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-NetworkLocationWizard%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-NetworkProvider%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-NTLM%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-OfflineFiles%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-OneBackup%4Debug.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-OOBE-Machine-DUI%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-PackageStateRoaming%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Partition%4Diagnostic.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-PerceptionRuntime%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-PerceptionSensorDataService%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-User Control Panel%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Policy%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-PowerShell-DesiredStateConfiguration-FileDownloadMana
Windows\System32\winevt\Logs\Microsoft-Windows-PrintBRM%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-PrintService%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-ReadyBoost%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-ReFS%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Regsvr32%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-RemoteApp and Desktop Connections%4Admin.evtx

Windows\System32\winevt\Logs\Microsoft-Windows-RemoteApp and Desktop Connections%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-RemoteDesktopServices-RdpCoreTS%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-RemoteDesktopServices-RdpCoreTS%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-RemoteDesktopServices-RemoteFX-Synth3dvsc%4Admin.e
Windows\System32\winevt\Logs\Microsoft-Windows-RemoteDesktopServices-SessionServices%4Operational.evt
Windows\System32\winevt\Logs\Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-ScmBus%4Certification.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-ScmDisk0101%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SearchUI%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Security-Audit-Configuration-Client%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Security-EnterpriseData-FileRevocationManager%4Operati
Windows\System32\winevt\Logs\Microsoft-Windows-Security-Netlogon%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Security-SPP-UX-GenuineCenter-Logging%4Operational.evt
Windows\System32\winevt\Logs\Microsoft-Windows-Security-UserConsentVerifier%4Audit.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-ServerEssentials-Deployment%4Deploy.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-ServerManager-ConfigureSMRemoting%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SettingSync-Azure%4Debug.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SettingSync-Azure%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SilProvider%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SmartCard-Audit%4Authentication.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SmartCard-DeviceEnum%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SmartCard-TPM-VCard-Module%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SmartCard-TPM-VCard-Module%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SMBDirect%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SMBWitnessClient%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SMBWitnessClient%4Informational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Storage-Tiering%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-StorageManagement%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-StorageSpaces-Driver%4Diagnostic.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-StorageSpaces-Driver%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-StorageSpaces-SpaceManager%4Diagnostic.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-StorageSpaces-SpaceManager%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-SystemSettingsThreshold%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-TCPIP%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-ClientUSBDevices%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-ClientUSBDevices%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-PnPDevices%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-PnPDevices%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-Printers%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-Printers%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-RDPClient%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Admin.evt
Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operation

Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-ServerUSBDevices%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-ServerUSBDevices%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-SessionBroker-Client%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-SessionBroker-Client%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-TZSync%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-TZUtil%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-UAC%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-UAC-FileVirtualization%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-User Device Registration%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-User-Loader%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-VerifyHardwareSecurity%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Volume%4Diagnostic.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-VPN-Client%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-VPN%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-WFP%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Win32k%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-WindowsSystemAssessmentTool%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Winsock-WS2HELP%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Wired-AutoConfig%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-Workplace Join%4Admin.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-WPD-ClassInstaller%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-WPD-CompositeClassDriver%4Operational.evtx
Windows\System32\winevt\Logs\Microsoft-Windows-WPD-MTPClassDriver%4Operational.evtx
Windows\System32\winevt\Logs\SMSApi.evtx

APPENDIX D. LIST OF FIGURES

Figure 1 – EMS Server (5.11-CO) Image Attributes Before 8
Figure 2 - EMS Server (5.13) Image Attributes After 9
Figure 3 – EMS Server (5.11-CO) System Data Sources Before 11
Figure 4 - EMS Server (5.13) System Data Sources After 11
Figure 5 - EMSSERVER (5.11-CO) Disk Partition Structure Before 12
Figure 6- EMSSERVER (5.13) Disk Partition Structure After 12
Figure 7 - Server Disk Partition and Directory Changes 13
Figure 8 - EMS Server (5.11-CO) Web Server Log Files Before 15
Figure 9 - EMS Server (5.13) Web Server Log Files After 15
Figure 10 - EMS Server (5.11-CO) MS SQL Server Installation Log Files Before 17
Figure 11 - EMS Server (5.13) MS SQL Server Installation Log Files After 17
Figure 12 - Example of Log File Content from EMS Server (5.11-CO) Before 18
Figure 13 - EMS Server (5.11-CO) SQL Server Log Files Before 19
Figure 14 - EMS Server (5.13) SQL Server Log Files After 19
Figure 15 - EMS Server (5.11-CO) Dell Server Update Files Before 20
Figure 16 - EMS Server (5.13) Dell Server Update Files After 20
Figure 17 - EMS Server (5.11-CO) Administrator WebCache Log Files Before 22
Figure 18 - EMS Server (5.13) Administrator WebCache Log Files After 22
Figure 19 - EMS Server (5.11-CO) "emsadmin" WebCache Log Files Before 23
Figure 20 -EMS Server (5.13) "emsadmin" WebCache Log Files After 23
Figure 21 - EMS Server (5.11-CO) Webcache Log File Content Before 24
Figure 22 - EMS SErver (5.11-CO) Webcache Log File Content Before - II 24
Figure 23 - EMS Server (5.11-CO) SSMS Log Files Before 25
Figure 24 - EMS Server (5.13) SSMS Log Files After 25
Figure 25 - EMS Server (5.11-CO) CBS Log Files Before 26
Figure 26 - EMS Server (5.13) CBS Log Files After 26
Figure 27 - EMS Server (5.11-CO) Election Databases Before 27
Figure 28 - EMS Server (5.13) Election Databases After 27
Figure 29 - EMS Server (5.11-CO) DHCP Log Files Before 28
Figure 30 - EMS Server (5.13) DHCP Log Files After 28
Figure 31 - EMS Server (5.11-CO) Event Logs Before 29
Figure 32 - EMS Server (5.13) Event Logs After 29
Figure 33 - Examples of Election Data Missing After Update 30
Figure 34 - EMS Server (5.11-CO) System Users Before 31
Figure 35 - EMS Server (5.13) System Users After 31
Figure 36 - EMS Server (5.11-CO) Virtual Directory Log Files Before 32
Figure 37 - EMS Server (5.13) Virtual Directory Log Files After 32
Figure 38 - EMS Server (5.11-CO) Windows Defender Log Files Before Dominion Update: 33
Figure 39 - EMS Server (5.13) Windows Defender Log Files After 33
Figure 40 - EMS Server Before/After .log File Comparison List 34
Figure 41 - EMS Server (5.11-CO) List of .evtx Event Log Files Before 36
Figure 42 - EMS Server (5.13) List of .evtx Event Log Files After 36

APPENDIX E. 2002 VOTING SYSTEMS STANDARDS (VSS)

The 2002 VSS explicitly states:

> "2.2.4.1 Common Standards
>
> To ensure system integrity, all system shall:
>
> ...
> g. Record and report the date and time of normal and abnormal events;
>
> h. Maintain a permanent record of all original audit data that cannot be modified or overridden but may be augmented by designated authorized officials in order to adjust for errors or omissions (e.g. during the canvassing process.)
>
> i. Detect and record every event, including the occurrence of an error condition that the system cannot overcome, and time-dependent or programmed events that occur without the intervention of the voter or a polling place operator; and
>
> J. Include built-in measurement, self-test, and diagnostic software and hardware for detecting and reporting the system's status and degree of operability.

Furthermore, in 2.2.5.3, COTS (Commercial Off-The-Shelf) General Purpose Computer System Requirements, the 2002 VSS states:

> Further requirements must be applied to COTS operating systems to ensure completeness and integrity of audit data for election software. These operating systems are capable of executing multiple application programs simultaneously. These systems include both servers and workstations (or "PCs"), including the many varieties of UNIX and Linux, and those offered by Microsoft and Apple. Election software running on these COTS systems is vulnerable to unintended effects from other user sessions, applications, and utilities, executing on the same platform at the same time as the election software.
>
> "Simultaneous processes" of concern include unauthorized network connections, unplanned user logins, and unintended execution or termination of operating system processes. An unauthorized network connection or unplanned user login can host unintended processes and user actions, such as the termination of operating system audit, the termination of electon software processes, or the deletion of election software audit and logging data. The execution of an operating system process could be a full system scan at a time when that process would adversely affect the election software processes. Operating system processes improperly terminated could be system audit or malicious code detection.
>
> To counter these vulnerabilities, three operating system protections are required on all such systems on which election software is hosted.
>
> First, authentication shall be configured on the local terminal (display screen and keyboard) and on all external connection devices ("network cards" and "ports"). This ensures that only authorized and identified users affect the system while election software is running.
>
> Second, operating system audit shall be enabled for all session openings and closings, for all connection openings and closings, for all process executions and terminations, and for the alteration or deletion of any memory or file object. This ensures the accuracy and completeness of election data stored on the system. It also ensures the existence of an audit record of any person or process altering or deleting system data or election data.

Third, the system shall be configured to execute only intended and necessary processes during the execution of election software. The system shall also be configured to halt election software processes upon the termination of any critical system process (such as system audit) during the execution of election software.

And, in 4.3 Data and Document Retention, the 2002 VSS states:

All systems shall:

a. Maintain the integrity of voting and audit data during an election, and for at least 22 months thereafter, a time sufficient in which to resolve most contested elections and support other activities related to the reconstruction and investigation of a contested election; and

b. Protect against the failure of any data input or storage device at a location controlled by the jurisdiction or its contractors, and against any attempt at improper data entry or retrieval.

And the 2002 VSS states, in 4.4.3 In-Process Audit Records:

In-process audit records document system operations during diagnostic routines and the casting and tallying of ballots. At a minimum, the in-process audit records shall contain:

a. Machine generated error and exception messages to demonstrate successful recovery. Examples include, but are not necessarily limited to:

1) The source and disposition of system interrupts resulting in entry into exception handling routines;

2) All messages generated by exception handlers;

3) The identification code and number of occurrences for each hardware and software error or failure;

4) Notification of system login or access errors, file access errors, and physical violations of security as they occur, and a summary record of these events after processing;

5) Other exception events such as power failures, failure of critical hardware components, data transmission errors, or other type of operating anomaly;

b. Critical system status messages other than informational messages displayed by the system during the course of normal operations. These items include, but are not limited to: Diagnostic and status messages upon startup;

2) The "zero totals" check conducted before opening the polling place or counting a precinct centrally;

3) For paper-based systems, the initiation or termination of card reader and communications equipment operation; and

4) For DRE machines at controlled voting locations, the event (and time, if available) of activating and casting each ballot (i.e., each voter's transaction as an event). This data can be compared with the publiccounter for reconciliation purposes;

c. Non-critical status messages that are generated by the machine's data quality monitor or by software and hardware condition monitors; and

d. System generated log of all normal process activity and system events that require operator intervention, so that each operator access can be monitored and access sequence can be constructed.

And secton 6.5.5, Shared Operating Environment, in the the 2002 VSS states:

> Ballot recording and vote counting can be performed in either a dedicated or nondedicated environment. If ballot recording and vote counting operations are performed in an environment that is shared with other data processing functions, both hardware and software features shall be present to protect the integrity of vote counting and of vote data. Systems that use a shared operating environment shall:
>
> a. Use security procedures and logging records to control access to system functions;
>
> b. Partition or compartmentalize voting system functions from other concurrent functions at least logically, and preferably physically as well;:
>
> c. Controlled system access by means of passwords, and restriction of account access to necessary functions only; and
>
> d. Have capabilities in place to control the flow of information, precluding data leakage through shared system resources.

The foregoing Forensic Examination and Report was prepared by me and I am responsible for its content.

This *15th* day of September, 2021.

DOUG GOULD
Chief Technical Officer

CyberTeamUS

Doug Gould Biography

Doug Gould is an expert in Cyber Security with more than 40 years' experience in the field. Doug retired from AT&T after 31 years, where he served as Chief Cyber Security Strategist. He currently serves as Chief Technical Officer at CyberTeamUS.

Doug began at AT&T with Bell Laboratories, serving in the Semiconductor Laser Development department and later in the Bell Lab's Security Group, as a delegate to the Bell Labs' Unix Systems Subcommittee, was an early pioneer in the field of Computer Forensics and won a Bell Labs Innovation Award. At AT&T he designed the security architecture for one of the largest states in the US, consulted with cabinets of the nations' largest corporations and designed the first healthcare network fully compliant with Healthcare Information Exchange standards. Outside AT&T, he has overseen security for a US Government Agency and has solved major cases for the FBI and Secret Service; he has served as an Officer of the Court as a forensic expert and has been an expert witness in landmark cybersecurity cases. He designed security architectures for DoD networks including some of the most sensitive areas of the Government. Doug has owned and led several professional services firms in the Information Security field. He served on the NC Council for Entrepreneurial Development and has consulted with many companies about the complex integration of business and technology.

Doug is the past president of Eastern North Carolina InfraGard, the public-private partnership between the nation's critical infrastructure operators and the US Intelligence community.

Doug's background is at the Master's level in Electrical Engineering, Computer Science, Computer Security and Business Administration.

He is a subject matter expert in:

- Strategic Enterprise Security
- Security Architecture & Design (including network Micro-Segmentation)
- Security Governance
- Risk Management
- Security Device Technologies (Firewalls, IDS/IPS, DLP, SIEMs, Encryption, VPNs, Unified Threat Management, etc., Enterprise, Remote and Cloud)
- Information Forensics (Computer & Network Forensics)
- Public Key Infrastructures
- Identity and Access Management
- Authentication, Authorization and Access Control (incl Biometrics)
- Regulatory Compliance
- Physical Security (Threat Assessment/Risk Analysis, TSCM, Access Control, Counterterrorism & Counterintelligence, facility and site protection)
- Business Continuity & Disaster Recovery Planning
- Response & Recovery Strategy
- Threat Intelligence
- Intelligence Analysis

Doug served as Chief Information Security Officer at the World Institute for Security Enhancement, has written advanced security courses, developed advanced security methodologies and has taught government, private sector professionals and law enforcement agents information security, computer forensics, advanced computer forensic sciences and Technical Surveillance Countermeasures (TSCM).

Doug holds numerous certifications in security including the CISSP and Certified Anti-Terrorism Specialist (CAS), as well as numerous instructor certifications in security.

Doug currently serves as Chief Technical Officer at CyberTeamUS.

He is a Vietnam-era US Navy Veteran where he worked in Electronic Warfare and Electronic Intelligence.

Doug is an invited conference speaker.

EXHIBIT

G

Doug Gould Forensic Addendum

Major Forensic Cases

- 1986 – Disclosure of National Security Information
  Discovered a leak of highly classified information and was able to identify the perpetrator within a group of 15 people. The FBI and US Naval Investigative Service brought this to resolution.
- Early 1990's – US Secret Service investigation, "Mothers of Doom" hacker case
  At USSS Evidence Lab, in response to a request for assistance from USS SA Jack Lewis, performed evidence recovery and identified 800 pages of evidence, invalidating immunity of a suspect's testimony in a proffer session.
- Late 1990's – Interpath, a North Carolina Internet Service Provider (ISP)
  This ISP was a tier-1 (top level) provider infected with Stacheldraht malware. Investigated the live (running) server and identified that all evidence on disc had been deleted. The only remaining evidence was a running program in memory, which was recovered. This case changed the Best Practice in Forensics – no longer is the first step necessarily removing the power. Had that been done no evidence would remain in this case.
- Late 1990's – As senior security administrator for the US EPA, investigated a complaint from the White House of computer intrusions and discovered an international attack involving 4 countries. Wrote monitoring and tracking software to capture the perpetrator online, brought together the FBI, Royal Canadian Mounted Police (RCMP), Scotland Yard and Deutche Bundespost in a live investigation tracking the intruder resulting in an arrest in Germany.
- South Carolina – A Public Works supervisor accused of violation of county policy was fired and brought countersuit. Forensic investigation recovered 4 3" thick binders of evidence showing sexual misconduct. Countersuit dismissed.
- Discovered Al Qaida attack plans targeting US Soil. Working with the FBI, the perpetrator, who was a foreign citizen in the US. Arrest made within 48 hours and the attack was thwarted.
- Mid-2000's – Florida vs. Rabinowicz – in a case where possession of contraband was the only element of proof, stipulated that the contraband was authentic and present. I proved forensically that the defendant was not technically in possession of the evidence and that evidence was planted. Qualified as an expert witness and provided expert testimony in this case.
- Mid-2000's – Identified a leak of national security from Oak Ridge National Laboratory involving chemical weapon information using forensic analysis and was able to identify the perpetrator. DSS responded and resolved the case.
- Mid-2000's – Investigated sabotage of a health industry contractor. The systems administrator had been fired and sabotaged the system. Solved the case and the administrator went to prison.

Instructor of Forensics

- Taught Forensics and Advance Forensic Techniques to State Law Enforcement, Military and major corporate customers at the World Institute for Security Enhancement.
- Taught Technical Surveillance Countermeasures (TSCM) course for government and industry at the World Institute for Security Enhancement.
- Wrote the entire course and taught the entire CISSP curriculum at Able Information Systems.

MESA COUNTY
CLERK & RECORDER

Hon. Janet Rowland

Board of County Commissioners

544 Rood Ave. Grand Junction, CO

March 1, 2022

RE: Forensic Report No. 2 on EMS Server Images

Dear Commissioner Rowland:

Enclosed is the second report, in electronic and hard copy form, from the cybersecurity experts who have continued to analyze the forensic images of the drive of the DVS Democracy Suite Election Management System in my office which we used for the management of the 2020 general election and the 2021 City Council Election. As you know, I had these images taken to preserve election records and help determine whether the county should continue to utilize the equipment from this vendor. Because the enclosed report reveals shocking vulnerabilities and defects in the current system, placing my office and other county clerks in legal jeopardy, I am forwarding this to the county attorney and to you so that the county may assess its legal position appropriately. Then, the public must know that its voting systems are fundamentally flawed, illegal, and inherently unreliable.

From my initial review of the report, it appears that our county's voting system was illegally certified and illegally configured in such a way that "vote totals can be easily changed." We have been assured for years that external intrusions are impossible because these systems are "air gapped," contain no modems, and cannot be accessed over the internet. It turns out that these assurances were false. In fact, the Mesa County voting system alone was found to contain thirty-six (36) wireless devices, and the system was configured to allow "any computer in the world" to connect to our EMS server. For this and other reasons—for example, the experts found uncertified software that had been illegally installed on the EMS server—our system violates the federal Voting System Standards that are mandated by Colorado law.

As the county officer elected to manage our elections in accordance with the law, I cannot hide behind the Secretary of State's certification of the Democracy Suite system and ignore the numerous and profound deficiencies revealed in this report. As the experts point out, the Secretary of State's certification itself was unlawful, based as it was on testing performed by an unaccredited lab, a lab that missed 100% of the security issues that render the system unusable, uncertifiable, and illegal. The county must reassess its recently-renewed lease agreement and consider its legal options immediately. We cannot continue to use this equipment. Please respond once you have read the enclosed report.

Very truly yours

Tina M. Peters

Tina M. Peters

Mesa County Clerk & Recorder

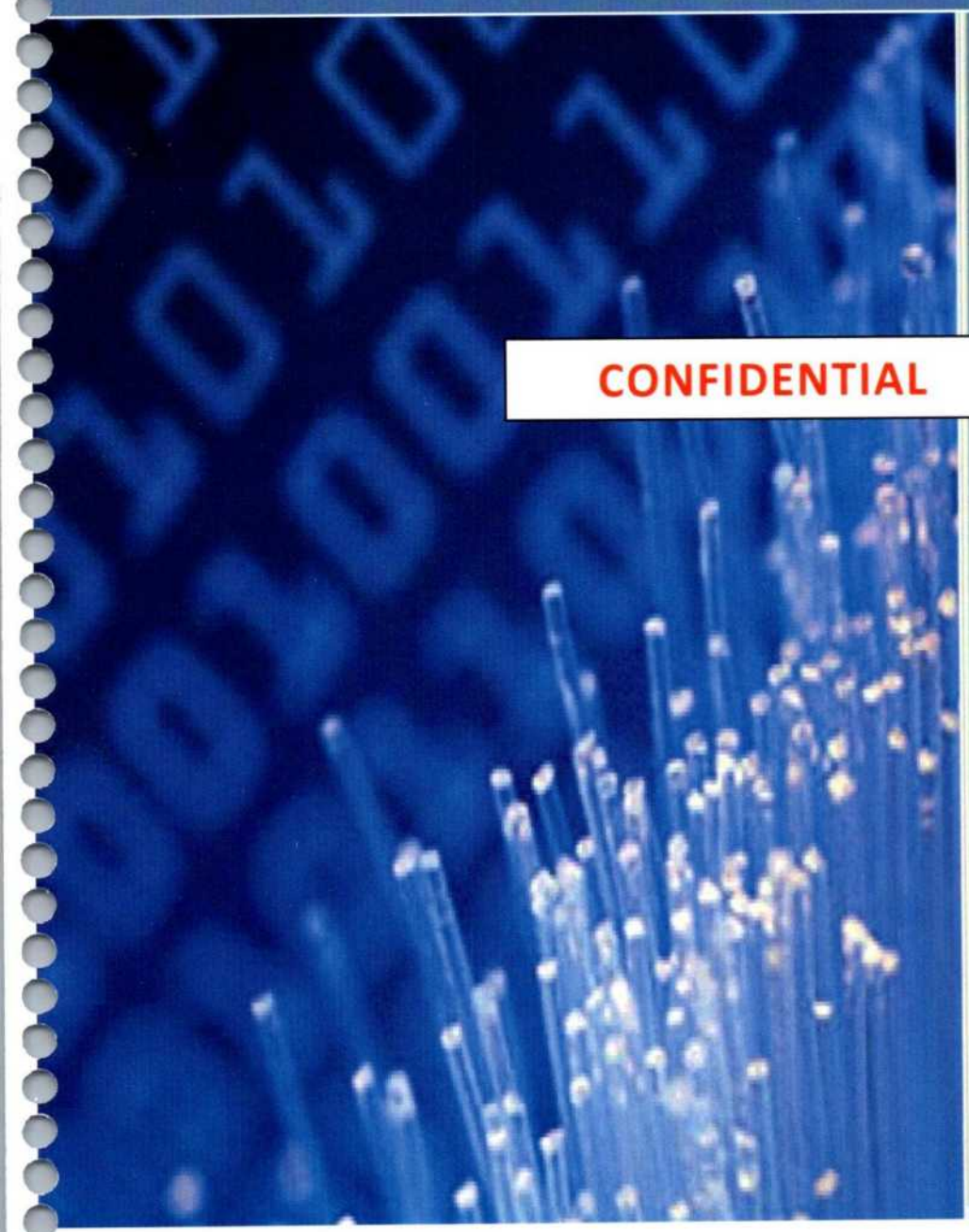
200 S. Spruce Street | Grand Junction, CO 81501

Tina.Peters@MesaCounty.US Office(970) 244-1714 Cell (970) 812-2610

Mesa County Colorado Voting System

Report #2

Forensic Examination and Analysis Report



CONFIDENTIAL

February 28, 2022

Table of Contents

Executive Summary 1
Critical Discoveries 1
Most Significant Findings: The Voting System is Not Secure, Violates Security Standards Required By State and Federal Law 2
"Back-Door" found in Voting System; Uncertified Software Invalidates Voting System Certification ... 2
Capability to Easily "Flip" Election Results Demonstrated 3
Voting System Components Manufactured and Assembled in China and Mexico 3
Voting System Presents an Immediate threat and is Dangerous to use in the upcoming 2022 election 3
Key Findings 5
Analysis Summary: Compliance of Mesa County, Colorado, DVS D-Suite systems with the law 7
Examination Methodology 15
FORENSIC ANALYSIS 19
System identification 19
Authenticity 21
Chain of Custody 21
Tools Used 22
TEST PREPARATION 22
Finding 1: 25
EXAMINATION OBJECTIVE 1: 34
Finding 2: 51
Finding 3: 52
Finding 4: 52
EXAMINATION RESULT 1 52
EXAMINATION OBJECTIVE 2: 53
Finding 5: 68
Finding 6: 75
EXAMINATION RESULT 2: 75
EXAMINATION OBJECTIVE 3: 76
EXAMINATION RESULT 3: 89
Conclusion 92
Appendix A. Compliance Requirements 96
Federal Election Commission 2002 Voting Systems Standards (VSS) 96
APPLICABILITY 96
VSS V1, 1.6, page 1-13: 96
VSS V1, 2.1, page 2-19: 97
VSS V1, 2.2, page 2-20: 97

DATA RETENTION 98
Election Record Definition, Scope and Content 98
VSS V1, 4.4.3, page 4-84: 98
Security Requirements for Voting Systems 100
VSS V1, 6.1, page 6-93: 100
VSS V1, 6.2, page 6-96: 101
VSS V1, 6.2.2, page 6-97: 101
Appendix B. Database Fundamentals 104
Appendix C. IP ADDRESSING FUNDAMENTALS 107
Appendix D. Nation-State Cyber Attack Capabilities 109
Introduction 109
Moonlight Maze 110
Stuxnet 110
Operation Titan-Rain 111
Operation Aurora 111
2020 US Government Attack 112
Summary 112
Appendix E. Security Considerations for SQL Server InstallationS 113
Appendix F. C.R.S. 1-5-608.5 115
Appendix G. C.R.S. 1-5-615 117
Appendix H. Man in the middle attack 119
Appendix J. Forensic Imaging Technology 121
Appendix K. Accessing a Computer Without a Password 126
Finding a password 126
Cracking a password 126
Rainbow Tables 127
Bypassing a password 127
Exploitation of Services 127
Intel Active Management Technology (AMT) and Management Engine (ME) 128
Dell Integrated Remote Access Controller (iDRAC) 129
Strengthening Access Security 129
APPENDIX L. Supply Chain Security Threat and Foreign Manufacturing 131
Appendix M. Colorado Secretary of State Press Release 133
Doug Gould Biography 137

Table of Figures

Figure 1 - SSMS Installation Date on Mesa County EMS server 12
Figure 2 - Mesa County, Colorado EMS server (5.11-CO) Forensic Image Attributes 20
Figure 3 - Test Workstation and Dominion EMS server 23
Figure 4 - Installed Microsoft Software 25
Figure 5 - SQL Server 2016 Configuration Manager 26
Figure 6 - SQL Server 2016 Configuration Manager – Network Protocols enabled 27
Figure 7 - TCP/IP Properties 30
Figure 8 - TCP/IP Properties of SQL Server, attached to port 1433 the standard (default) port. 31
Figure 9 - SQL Server Properties 32
Figure 10 - Encryption is enabled but No Encryption Certificate is configured 33
Figure 11 - SQL Server Management Studio (SSMS) software showing in the EMS server Start Menu 34
Figure 12 - SSMS is installed and starting on the EMS server system. 35
Figure 13 - Logging in to the SQL Server using SQL Server Management Studio 36
Figure 14 - SSMS enables direct access to the internal databases to anyone logged in to the EMS server. 37
Figure 15 - Databases from many prior elections are fully accessible 38
Figure 16 - Additional databases used in previous elections 39
Figure 17 - Internal database tables, including ones with counted votes are accessible 40
Figure 18 - Menu Option to Select the Top 1000 rows 41
Figure 19 - Accessing the Ballot Choice database table 42
Figure 20 - Test to determine if the Ballot Choice Table can be edited to easily flip the votes 43
Figure 21 - Candidate settings for Trump 44
Figure 22 - Candidate settings for Biden 45
Figure 23 - Pulling up the results report prior to attempting the alteration 46
Figure 24 - Run Stored Procedure to pull up a report of Presidential Electors 47
Figure 25 - Retrieved Vote Totals 48
Figure 26 - Candidate number for Trump modified 49
Figure 27 - Candidate number for Biden modified 50
Figure 28 - Vote totals retrieved again after modification. 51
Figure 29 - Accessing port 1433 with Telnet 53
Figure 30 - The EMS server network interface appears to answer a connection to port 1433 54
Figure 31 - EMS server has the 'Windows Firewall' enabled 55
Figure 32 - Windows Firewall Custom SQL entry is enabled 57
Figure 33 - SQL port 1433 is allowed. 58
Figure 34 - Access to the SQL database standard port is allowed from ANY IP ADDRESS worldwide. 59
Figure 35 - No additional IP address restrictions or permissions 60

Figure 36 - Test Workstation, 192.168.100.150, and EMS, 192.168.100.10, are on the same subnet 61
Figure 37 - Mesa EMS server is responding to network ping test..... 62
Figure 38 - Telnet connectivity test from separate computer not part of the Dominion system 63
Figure 39 - Telnet to EMS server port 1433 (SQL) succeeds 64
Figure 40 - SSMS access test from separate computer not part of the DVS D-Suite system..... 65
Figure 41 - Log In to the server..... 66
Figure 42 - From a separate Windows 10 computer EMS server database access has been obtained..... 67
Figure 43 - From a separate Windows computer, the databases can be accessed and reports run..... 68
Figure 44 - SSMS permits database Edit..... 69
Figure 45 - EMS server Database view from a separate computer not part of the DVS D-Suite system 70
Figure 46 - SSMS permits us to edit the databases..... 71
Figure 47 - "internalMachineId" for Trump is now changed back to a 2. 72
Figure 48 - Candidate data for Biden from previous change 73
Figure 49 - Candidate data for Biden changed back to original 74
Figure 50 - The vote choice was remotely changed back to its original state 75
Figure 51 - Network scanner installed on cellphone..... 76
Figure 52 - IP address for the EMS server found via wireless connection and iPhone app..... 77
Figure 53 - Scanner Results..... 78
Figure 54 - SQL Access Functionality 79
Figure 55 - SQL Pro Capabilities..... 80
Figure 56 - Making an SQL Connection..... 81
Figure 57 - iPhone Connection to Dominion EMS Database 82
Figure 58 - Databases listing, Continued 83
Figure 59 - Database Table Listing..... 84
Figure 60 - Database Access 85
Figure 61 - Executing a Database Query..... 86
Figure 62 - Table Data..... 87
Figure 63 - A script to change the vote data 88
Figure 64 - Script Results 89
Figure 65 - Small Wireless Device Surreptitiously Installed (internally) on a Computer Motherboard 90
Figure 66 - DVS Compliance Statement..... 102
Figure 67 - Man In The Middle Attack 119
Figure 68 - Illustrative Hard Disk Components..... 121
Figure 69 - Disk Track and Sector illustration 122

EXECUTIVE SUMMARY

This report documents findings in an ongoing forensic examination of images of the hard drives[1] of the Dominion Voting System (DVS) Democracy Suite (D-Suite) version 5.11-CO Election Management System (EMS) server of Mesa County, Colorado. The DVS D-Suite EMS server in that configuration was used for all elections held in 2020 and through May 2021, including the November, 2020 General Election, and the April, 2021 Grand Junction Municipal Election. This voting system represents a portion of the overall election system infrastructure in Mesa County and the State of Colorado. This report is limited to a subset of the findings of an ongoing investigation. Report #1 is incorporated by reference.[2] The findings in this report were prepared by me as a consultant to the legal team representing Tina Peters, the Mesa County Clerk and Recorder, pursuant to her statutory duties as Mesa County's Chief Election Official.

Critical Discoveries

This report details the following critical discoveries regarding Mesa County's voting system:

- **Uncertified software installed, rendering the voting system unlawful for use in elections.**
- **Does not meet statutorily mandated Voting System Standards (VSS) and could not have been lawfully certified for purchase or use.**
- **Suffered systematic deletion of election records (audit log files required by Federal and State law to be generated and maintained), which, in combination with other issues revealed in this report, creates an unauditable "back door" into the election system.**
- **Violates Voting Systems Standards ("VSS") which expressly mandate prevention of the ability to "change calculated vote totals." This report documents this non-compliance from the logged-in EMS server, from a non-DVS computer with network access, and from a cell phone (which may be possible if any of the 36 internal wireless devices in voting system components are deliberately or accidentally enabled and a password is obtained).**
- **Mandatory VSS "System Auditability" required features are disabled.**
- **Is configured with 36 wireless devices, which represent an extreme and unnecessary vulnerability, and which may be exploited to obtain unauthorized access from external devices, networks, and the Internet.**
- **Is configured through firewall settings to allow any computer in the world to connect to the Election Management System (EMS) server.**
- **Uses only a Windows password with generic userIDs to restrict and control access.**
- **Contains user accounts with administrative access that share passwords, subverting VSS-required user accountability and action traceability controls.**
- **Uses a self-signed encryption certificate which exposes the system to the risk of undetected compromise or alteration.**

---

[1] A forensic image of a hard drive is a bit-for-bit copy of the user accessible data storage area residing on the data storage mechanism used by the computer system; it is every byte of data accessible to the computer or user. For a complete discussion of this definition, see Appendix J.

[2] Report No.1 was issued on September 15, 2021 and can be downloaded at https://standwithtina.org/.

Most Significant Findings: The Voting System is Not Secure, Violates Security Standards Required By State and Federal Law

The most significant findings include the conclusive determination, based on testing, that the voting system is not secure and protections have not been implemented in accordance with the requirements of the Federal Election Commission's 2002 Voting System Standards (VSS) (see Appendix A). Those Standards constitute a mandatory minimum requirement for a voting system to be certified and used under Colorado law. Given the fundamental flaws in the security design and configuration of this system, there is no conceivable interpretation under which this voting system could be considered secure.[3] The fact that it was tested and certified for use vitiates claims of competency and trustworthiness of the entire regime of testing and certification being used, of truthfulness of testing and certification statements, of competency of the Colorado Secretary of State's office, and of the validity of any election results obtained from the voting system as used in any jurisdiction.

"Back-Door" found in Voting System; Uncertified Software Invalidates Voting System Certification

The combination of unauthorized software installed in the EMS server in 2017 (still present in violation of law in 2021), the failure to employ security mechanisms already built into the system and required by VSS, and the obliteration of mandatory audit logs (destruction of both election records and evidence of access to the EMS server) that Federal and State law require be preserved, create a "back-door" to the EMS server that is only partially protected by a simple password, with no preserved audit records. The existence of uncertified software violates the certification of the voting system and makes the use of the voting system in an election illegal. Indeed, University of Michigan Professor J. Alex Halderman,[4] a recognized computer science expert on electronic voting systems, testified under oath[5] that components of this Dominion Voting System ("DVS") are highly vulnerable to attack and that the system he examined is used in 16 other states, including Colorado. In his declaration he states under oath that this vulnerability in the Dominion voting system can be used to "steal votes", and requests the federal court allow him to give the Critical Infrastructure Security Agency (CISA) immediate access to his report detailing his findings.[6] The findings in this report agree with Professor Halderman's finding that the system can be used to steal elections.

---

[3] Even the Center for Internet Security (CIS) recognizes the need for these controls in their Handbook for Election Infrastructure Security: https://www.cisecurity.org/wp-content/uploads/2018/02/CIS-Elections-eBook-15-Feb.pdf. The National Institute of Standards and Technology (NIST), which chaired the development of the Voting Systems Standards extensively recommends the fundamental security principle of "Least Privilege" that has been ignored in the configuration of the EMS.

[4] Professor of Computer Science & Engineering, University of Michigan, Director, University of Michigan Center for Computer Science and Society, Director, Michigan CSE Systems Lab, https://jhalderm.com/.

[5] Declaration of J. Alex Halderman, *Curling et al. v. Raffensperger et al.*, 1:17-cv-02989-AT, Docket No. 1177-1, (ND Ga.).

[6] *Id.*

A password was not necessary to access this EMS server.[7] There are many mechanisms by which a server can be exploited and administrative access obtained without a password; the National Institute of Standards and Technology (NIST) National Vulnerability Database (NVD) has identified over eight hundred of these admin-access vulnerabilities[8] (among hundreds of thousands of other vulnerabilities) since its inception in 2005, and the Common Vulnerabilities and Exposures (CVE) program operated by MITRE Corp. lists nearly 170,000 computer vulnerabilities[9] that are *publicly known* since its inception in 1999.

Capability to Easily "Flip" Election Results Demonstrated

Tests demonstrate the vote totals can be easily changed, commonly known as "flipping the election,"[10] in this critical Election Management System server. The VSS directs voting systems vendors, like DVS, to address this specific risk[11] but based on the software contained on the EMS that was analyzed, the vendor has not done so here. Further, the obliteration of audit trails (logs) on the EMS server makes it extraordinarily difficult (and maybe impossible) to forensically determine whether any external connection allowing unauthorized access to the voting system, wireless or wired, occurred before, during or after the elections.

This report describes the absence of legally required security features on the voting system and then demonstrates only a few examples of the many possible methods by which it is possible to change calculated vote totals and alter the results of an election as consequence of those security failures.

Voting System Components Manufactured and Assembled in China and Mexico

The Mesa County EMS server used through May 2021 (serial number 4NV1V52) was assembled in Mexico, and its motherboard was manufactured in China. It is well understood that foreign manufacture or assembly exposes the components to the risk of compromise through the installation of foreign-controlled access devices during manufacture in the reported supply-chain attack.[12]

Voting System Presents an Immediate threat and is Dangerous to use in the upcoming 2022 election

The tests conducted in this report demonstrate and document three test intrusions into the DVS Election Management System server using popular, commercially available software that allows easy access to vulnerable election records. Given even momentary access, a person with only moderate computer skills

---

[7] The Mesa County Co. DVS D-Suite 5.11-CO server was forensically restored in a virtual environment, and a common password reset/bypass technique was used. See Appendix K. Also see www.gaverifiedvoting.org/pdf-litigation/20200819-785_2-Declaration-Alex-Halderman.pdf

[8] https://nvd.nist.gov/vuln/search/results?form_type=Basic&results_type=overview&query=administrative+access&search_type=all&isCpeNameSearch=false

[9] https://www.cve.org/

[10] The switching of calculated vote totals in an election has been identified in 2 other jurisdictions: Fulton County, Pennsylvania, and Antrim County, Michigan. *See* https://rumble.com/embed/vjr2u6/?pub=dw7pn which documents testimony of the Fulton County finding.

[11] "Changing the calculated vote totals," VSS, Volume 1, section 6.1, page 6-93. See Appendix A.

[12] https://www.bloomberg.com/news/features/2018-10-04/the-big-hack-how-china-used-a-tiny-chip-to-infiltrate-america-s-top-companies; See Appendix L for discussion.

can perform such an intrusion. It is not possible to reconcile these massive security failures with the obvious requirements for such an important piece of critical infrastructure. In combination with mandatory audit records being deleted in violation of state and federal laws that require their preservation, and in violation of evidence preservation orders for active legal cases [13], this EMS server presents an immediate threat to election integrity, with potential grave consequence to Colorado and the Nation by allowing the unauthorized alteration of election results.

The threat is immediate because 2022 election processes are already underway with primary elections imminent, and many jurisdictions will use these systems, and citizens' electoral franchise will be at risk, if citizens and public officials are not warned.

The initial installation and continued presence of uncertified software (Microsoft SQL Server Management Studio) in the Mesa County EMS Server is a violation of law. However, the tests conducted for this report clearly demonstrate that it is not the SSMS software alone that enabled illegal access to and modification of election databases and scanned ballot images. The state certifying this software on a chronically insecure system does not remedy the system's chronic insecurity – it only obfuscates one problem (insecurity) with another (improper testing and certification).

In contrast to the testing and certification of DVS D-Suite 5.11-CO, the current certification in Colorado of DVS D-Suite 5.13 includes SSMS, but tests conducted in this examination demonstrate conclusively that the EMS system is insecure both with, and without, SSMS.

[13] Log files and other auditable records of normal and abnormal activity on computer-based voting systems are not only election records which must be preserved for 22 months according to Federal law, and 25 months according to Colorado law, they also represent evidence that is subject to document preservation requirements in existing civil litigation and, forseeably, for future civil and criminal cases.

Key Findings

Six Key Findings in this report are:

1. The Mesa County EMS server used in the 2020 General Election had Microsoft SQL Server Management Studio 17 installed in May 2017. This software is not listed on the official test and certification report nor on the vendor's application to the Colorado Secretary of State for certification of DVS D-Suite version 5.11-CO signed by "Nick Ikonomakis," VP, Engineering [Dominion Voting Systems], dated 6/6/2019. As it was not listed, tested, or certified, the unauthorized installation of this software violates and renders illegal the certification of the election system, and its use in an election.

2. The inclusion of unauthorized and uncertified Microsoft SQL Server Management Studio software, as configured, allows the bypassing of Dominion Voting Systems' software and enables any data in the vote databases to be changed. For example, using the uncertified Microsoft SQL Server Management Studio software, it is a quick and simple task to "flip" the vote (change calculated vote totals, demonstrated herein by changing only two values in the database to flip tens of thousands of votes).

3. With the addition of a wireless access device (added to the test to emulate the presence of multiple wireless devices that exist on Mesa County's DVS hardware), the insecure configuration of the Mesa County EMS server allowed the editing and changing of the calculated vote totals using a standard iPhone. Wireless access, whether enabled accidentally or enabled/added deliberately (even in secret) to a voting system network, enables intrusion, attack, and compromise of any electronic voting system. The security configuration of the EMS server was wholly inadequate to prevent such intrusions. Thirty-six wireless access devices were identified built-in to the Mesa County DVS D-Suite system components, as documented by Dell and the Secretary of State's equipment inventory.

But, due to the DVS-specified configuration of the EMS, and the Secretary of State-approved procedures that overwrite audit records[14] – by mandating that the EMS server "overwrite" log files "as needed," and further, during the Secretary of State's so-called "Trusted Build" update which overwrote the EMS server, both in violation of federal and state laws - it is at best, extremely difficult to determine from EMS server audit log data how or even whether the wireless connections were used during or affecting Mesa County's elections.

4. The exceptionally poor security configuration of the EMS server's operating system, firewall, and the improper and inadequate configuration of the SQL Server database management system

[14] Approved, by certifying vendor supplied information. CRS-1-5-620 states that the vendor provides documentation including manuals to the Secretary of State, and any information not on file with and approved by the Secretary of State shall not be used in an election.

(DBMS) enabled access to the election databases and the alteration of vote totals using freely available, non-DVS and non-Microsoft database app downloaded and installed onto on a cell phone.

5. The Colorado Secretary of State's certification of DVS D-Suite version 5.11-CO for use throughout the state of Colorado was illegal,[15] given the overwhelming number of VSS compliance violations found within the EMS server, which undermine the credibility of the claimed testing, technical competency of the testing lab, and the Secretary of State's certification.

6. The Mesa County, Colorado EMS server as used in elections including the 2020 General Election, and the April 2021 Grand Junction Municipal Election, has been shown to be insecure and grossly misconfigured such that it could not prevent unauthorized access to the election database or, as explicitly required by the VSS, prevent "changing the calculated vote totals" (demonstrated using an exact forensic replica of the system). This constitutes a material violation of the VSS requirements. It was possible to access the EMS server and change only 2 numbers in the database to completely reverse the Mesa County election 2020 Presidential election results stored on the EMS server. If this was done during the election, the EMS server would have then reported the changed vote totals as its authentic result.

[15] The Colorado Secretary of State's certification of both DVS D-Suite 5.11-CO and 5.13 were also apparently illegal under state law, given that testing by a federally accredited testing lab is prerequisite for certification under Colorado law, and the Secretary's certifications both relied upon testing by an unaccredited voting system testing lab.

Analysis Summary: Compliance of Mesa County, Colorado, DVS D-Suite systems with the law

Four Key Objectives for this assessment are:

1. To determine whether implemented security capabilities comply with the 2002 Voting System Standards (VSS), mandatory under Colorado law;
2. To determine whether the results of an election stored on the EMS server can be altered by any person with physical access to the logged-in EMS server,
3. To determine whether the results of an election stored on the EMS server can be altered by any person using even a non-Dominion computer directly or indirectly connected to the EMS server network, and
4. To determine whether the results of an election stored on the EMS server can be altered by any person using a device such as a cell phone wirelessly connected to the EMS server network.

It is recommended that this report be viewed on a computer. Some of the screen images may be difficult to read when printed on paper, but viewed on a computer they can be expanded (zoomed in) and are easily read.

Documented in this report is a series of tests conducted as part of the examination to evaluate a few aspects of the security compliance[16] of the Mesa County, Colorado DVS D-Suite version 5.11-CO EMS server, and the findings from that examination. These tests were limited to the EMS server. The EMS server receives and stores ballots in the form of electronic ballot images and cast vote records (CVR) from each ballot optically scanned into ImageCast Central (ICC) scanning/tabulation machines, and tabulates the results of the election. The images, CVRs, tabulated results and all system log files that document every aspect of system state, access, and operation are critical election records. The EMS server is one of the most critical components of the voting system and the security of its election records is of paramount importance.

The examination began with no pre-conceived assumptions about vulnerabilities and security. An identical copy of the Mesa County EMS server hard drive image[17] was mounted and tested to exactly replicate the conditions of use during elections conducted between the installation of version 5.11-CO in 2019 and its replacement on May 25, 2021. The identified uncertified SSMS software component was installed earlier and very likely presented this same security weakness since its installation in 2017, but the scope of the tests in this report only addresses the 2019-2021 period. The computer-based voting system is extraordinarily complex and requires skill, knowledge, and diligence to configure securely. Despite being custom-ordered and then configured by the vendor, the critical nature of voting systems and the extreme importance of securely configuring these computer-based systems requires that voting systems be tested by competent cybersecurity professionals to determine their vulnerability. Colorado law requires only that

[16] The evaluation identified critical weaknesses in the system and this report documents those findings. A comprehensive evaluation of every possible defect is beyond the scope of this report; the investigation is ongoing.

[17] An identical copy of the Logical drive image, mounted within an Oracle VirtualBox virtual environment.

they be tested by a laboratory accredited by the U.S. Election Assistance Commission (EAC) and the results certified by the Colorado Secretary of State.

The DVS application to the Colorado Secretary of State for certification of DVS D-Suite 5.11-CO represents that this system "meets the requirements of the Colorado Secretary of State Election Rules (8 CCR 1505-1)" (which specify that all voting systems in Colorado must meet the requirements of the 2002 VSS).[18] This includes documentation of the "minimum services needed for the successful, secure and hardened operation of the voting system" and "contains security measures for all systems, software, devices (upload, download, and other programming devices) that act as connectors and any additional recommended security measures." While this provision of law addresses documentation to be provided, it is also necessarily required that the documentation be truthful and accurate. A forensic examination of this system, and tests performed in this examination, clearly show that these requirements are not met; the system is not secure and certainly not hardened against unauthorized access.

Testing confirmed that an outside party could use a separate computer as well as a cell phone, with publicly available and widely used free software (none of which were part of the DVS D-Suite), to easily change election results. The obliteration of audit trails on the EMS server by DVS and the Secretary of State personnel during the "trusted build" process diminished the ability to forensically determine whether any network connections (including wireless connections or intrusions) were made to the EMS server. Thirty-five wireless devices were identified on the DVS D-Suite system, including the ImageCast Voter Activation (ICVA) computer, serial number 2DX0Z52, ordered on August 16, 2015 by DVS for use in Mesa County. It was ordered by DVS configured with a Dell Wireless 1560 internal wireless adapter, providing both 2.4GHz and 5GHz (dual band) Wi-Fi and Bluetooth connectivity to and through that ICVA computer. In total, Mesa County was provided thirty-five D-Suite components with wireless capability installed: Dell Latitude 7450 computers providing ICVA functionality, serial nos. 8GX0Z52, 8JX0Z52, BCX0Z52 with Dell Wireless 1560 modules, and Dell Optiplex 9030 ImageCast Central (ICC) systems, serial nos. H4B4T52, H4G0T52, H4JBT52, and H4L9T52 with Dell Wireless modules. A Dell E310DW wireless printer was configured as the EMS server's default printer, with IP address 192.168.100.11, bringing the total number of wireless devices to thirty-six. Wireless device encryption can be easily broken,[19] and the vulnerabilities are online and in the Computer Vulnerabilities and Exposures (CVE) database.[20] A demonstration video of this intrusion is also available.[21] Twenty-eight (28) tablets, provided by DVS as ICX devices in the D-Suite system, include

---

[18] https://web.archive.org/web/20201018013640/https://www.sos.state.co.us/pubs/rule_making/CurrentRules/8CCR1505-1/Rule21.pdf

[19] Vulnerability: http://www.dell.com/support/kbdoc/en-us/000125799/wi-fi-security-protocol-key-re-installation-attack-krack-impact-status-on-dell-products; Published and freely available code to implement the attack: https://www.joe0.com/2017/11/11/kali-linux-virtualbox-instructions-for-testing-wi-fi-devices-against-wpa2-key-reinstallation-attack-krack-attack/

[20] http://cve.mitre.org/ : CVE-2017-13077, CVE-2017-13078, CVE-2017-13079, CVE-2017-13080, CVE-2017-13081, CVE-2017-13082, CVE-2017-13084, CVE-2017-13086, CVE-2017-13087, CVE-2017-13088. This attack is against the WPA2 encryption protocol and all wireless devices, regardless of manufacturer, are impacted.

[21] http://www.krackattacks.com/, Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2, Vanhoef and Piessens, https://papers.mathyvanhoef.com/ccs2017.pdf

wireless capability. The prior expert analysis and testimony of Professor Halderman further confirms the vulnerability of these Dominion ICX components to malicious attack and compromise by an outside party.[22]

Because of the extraordinary nature of the "back-door" identified and because internal wireless devices were included as part of the DVS D-Suite system used in Mesa County, I added a wireless access device to the server network during testing to properly replicate the actual hardware used in Mesa County. This enabled determination of whether the system vulnerabilities could be exploited with the more limited capabilities of a mobile device. This report describes testing that demonstrates how easily the design and configuration of this voting system allows this type of exploitation.[23]

The tests in this report first demonstrate that any person with physical access to the logged-in EMS system can change the election database results (calculated vote totals), with[24] or without[25] a userID and password, on the Mesa County EMS before, during, or after the election by using a few mouse clicks. By itself, the ability of any user to modify election database totals illustrates the voting system's non-compliance with VSS and Colorado law. The tests also demonstrate that if the voting system has any external connection for even a moment, a person anywhere in the world can change the election database results on the EMS server with a few mouse clicks. This is an extraordinary danger to election integrity.

The protection offered by use of passwords is further weakened by the fact that different userIDs created on the EMS server share the same password.[26] Shared passwords were also reported in the Maricopa, Arizona forensic audit.[27] Rudimentary security protocol demands that each userID must have its own unique password. The sharing of password across accounts renders ineffective individual accountability for actions by a user (each assigned a specific userID, required for access control mandated by VSS and the ability of audit trails to identify fraudulent activity). This renders the system noncompliant with VSS requirements. VSS mandates, among other things, that the system: (1) "establish and maintain controls that can ensure that accidents, inadvertent mistakes, and errors are minimized; (2) protect the system from intentional manipulation and fraud, and from malicious mischief"; and (3) identify fraudulent or erroneous changes to the system."[28] Other jurisdictions have learned that they do not have control of their voting systems but the vendor, Dominion Voting Systems, has the administrative passwords and, therefore,

---

[22] www.gaverifiedvoting.org/pdf-litigation/20200819-785_2-Declaration-Alex-Halderman.pdf

[23] The VSS expressly identifies the prevention of this type of manipulation in its security objectives for voting systems, VSS Volume 1, section 6.1, page 6-93, excerpted in Appendix A.

[24] I accessed the EMS server with and without a password. I was able to guess the password, and separately used a well-known password bypass technique, both methods were successful and I gained access to a copy of the EMS server in an Oracle VirtualBox environment.

[25] Passwords are easily bypassed, and knowledge of a specific password is not required, since access can be obtained without a password. See Appendix K.

[26] Thirty different userIDs on the Mesa County EMS server were found to share an identical password. Two of those accounts were enabled and active.

[27] Maricopa County Forensic Election Audit, Volume III, section 6.5.2.1.3

[28] (VSS V1, 6.1, page 6-93, see Appendix A).

control.[29] Mesa County's DVS EMS server has an administrator account installed specifically for Dominion Voting Systems' use.[30] In light of the legal and security responsibilities in the administration of elections, allowing a vendor (in this case DVS) to maintain administrator access to the voting system is inexplicable, as is the exclusion of local election officials from control over their own elections.

The names of account userIDs on Mesa County's EMS server, created during the installation of DVS D-Suite 5.11-CO, are generic. Generic account userIDs were also found in the Maricopa, Arizona audit.[31] This finding in Arizona strongly suggests that it is a DVS practice to use generic userIDs and the same userIDs are likely used on *every* DVS election system in the USA. As one of the two components of required authentication (userID and password), this is an extraordinary compromise of security, as it is likely that once a userID from one state is known, it may be known for *all* states.

The examination found that the EMS server network was active and in use; the Ethernet network interface was found to be enabled, an IP address was found to be assigned, and election databases and ballot images were found to be stored on the EMS 'NAS' disk drive. The drive was shared to the connected network.[32] Any representation that the EMS server was not connected to a network is false. The transmission control protocol / internet protocol (TCP/IP) port that supports direct back-end database access on the EMS server was found to be unprotected by anything other than Windows authentication (a common userID and a shared password) and any person who gains unauthorized access will have full access to ballot images and the tabulated vote databases, in violation of the 2002 VSS.

The tests conducted in this examination found the system to be insecure and also ensured that no protections that might otherwise have secured the system were overlooked by the examination process. No advanced security penetration techniques were needed; the initial access to the operating system (i.e., "login") was performed both by guessing the password as well as by using well-known and easy to find password bypass techniques. The unauthorized and uncertified Microsoft SQL Server Management Studio software[33] ("SSMS") on the EMS server was run and access to the SQL server databases on the EMS server, which should be highly restricted, was granted without restriction or challenge. This same access has been found in other forensic examinations of virtually identical DVS D-Suite voting systems used in at least two other states.[34] A non-Microsoft, non-DVS software application that supports SQL database access was also used (from an iPhone) and access to Mesa County EMS server election databases was obtained, allowing

---

[29] Maricopa County Forensic Election Audit, Volume III, section 6.5.3.1.3. *See also* https://www.westernjournal.com/az-audit-exclusive-election-systems-password-hasnt-changed-2-years-shared-time/.

[30] Account names are withheld in this report to protect the security of the system, since an account name and a password are literally the only things protecting this system.

[31] Maricopa County Forensic Election Audit, Volume III, section 6.5.2.1.3

[32] Dominion misleadingly refers to this as "NAS." It is not. NAS stands for Network-Attached Storage. This storage was found not to be network-attached, but instead, "direct-attached," and is thus a DAS instead of a NAS.

[33] D:\Program Files (x86)\Microsoft SQL Server\140\Tools\Binn\ManagementStudio\Ssms.exe.

[34] Analysis of the Antrim County, Michigan November 2020 Election Incident, J. Alex Halderman, March 26,2021, p.10; September 24, 2021, Presentation of Ben Cotton entitled *Arizona Senate Audit, Digital Findings*, slide 13.

changes to the calculated vote totals. Testing shows conclusively that the voting system was not secure and that protections required by law were not enabled.

Report #1 documented the destruction of system log files that voting systems are required to generate and preserve in order to comply with federal and Colorado law.[35]) Those critical election records would be necessary to allow a forensic examiner to identify whether any changes to the election databases were made, and when and how they occurred. This system did not preserve those election records,[36] in violation of federal and Colorado law. This failure was a direct result of the system configurations and technical guidance as directed by Dominion and mandated by the Colorado Secretary of State for all counties using D-Suite version 5.11-CO EMS servers. The installation of the voting system software update (called the "Trusted Build") by the Secretary of State, assisted by DVS personnel, in all DVS-equipped Colorado counties further overwrote and eradicated most records necessary to perform a forensic audit of the affected elections.

As a direct result of the destruction of those election records (in the form of log files that provide an audit trail required by law to be preserved), any examiner, much less a non-expert public official, will find it difficult if not impossible to determine conclusively that the voting systems have not been tampered with or operated in an unauthorized manner. Destruction of those election records prevents detection and/or confirmation that the vulnerabilities identified in this report were not exploited to alter election results.

A full, independent forensic audit should be conducted in any jurisdiction that used this system, given the extraordinary insecurity and non-compliance of this voting system with both legal standards and industry-recognized best practices and the failure of the existing testing and certification regime to detect those conditions,. Such an audit should include every component of the voting system, all electronic logs, removable media, and escrowed source code. Cast paper ballots should be examined for authenticity and then recounted in order to have confidence that the tabulated vote count matches the paper ballots. Because of the obliteration of audit trail data, audit techniques which rely upon small, statistical sampling of results (so-called "risk-limiting audits") are not reliable. No person can trust any result obtained from this system in any election in which it was used due to the extreme insecurity of this voting system.

Although this examination addresses the local Mesa County, Colorado election results stored on the Mesa County EMS server, similar destruction of election records and the security weaknesses that enabled it are

---

[35] Appendix A, VSS, Retention Requirement

[36] If not for the action of the Mesa County Clerk, who forensically preserved the Mesa County election records by backup of EMS server hard drive, the auditable record of the partial EMS server log files that remained from the November 2020 General Election and the April 2021 Grand Junction Municipal Election would have been destroyed by the Secretary of State's action and direction. That destruction of election records by DVS and the Secretary of State would have precluded a forensic audit of those elections and prevented the exposure of the voting system vulnerabilities as they existed in the November 2020 general election and the April 2021 Grand Junction Municipal Election. Failure to meet statutory-security compliance requirements would have been hidden from both public officials and the public. Neither the Secretary of State nor DVS instructed election officials to properly preserve these critical electronic records prior to these destructive "updates" and instead instructed them only to preserve ballot images and related election project files.

highly likely to have occurred across Colorado and possibly other jurisdictions. The configuration of the system is required to be tested by EAC-accredited testing labs, controlled through certification by the Colorado Secretary of State, and specified by Dominion Voting Systems (DVS), so it is almost certain this system is used throughout Colorado, and it is likely very similar, if not identical to systems used in other states.

Examination of the EMS server found that unauthorized Microsoft SQL Server Management Studio software[37] ("SSMS") was installed on 5/17/2017 at 06:49:44 AM. Given that the "trusted build" process was used in 2019 and overwrote all previous data on the Mesa County EMS server, SSMS must have been installed by DVS on its golden image of the D-Suite system; if it were installed by Mesa County staff, the installation date could not have preceded the DVS installation date of D-Suite 5.11-CO in 2019. SSMS remained installed on Mesa County's EMS server through the backup imaging conducted in May 2021. That software was present on the 5.11-CO EMS server but not listed on the Certification Application or testing report for the DVS D-Suite 5.11-CO system. This failure of the manufacturer to meet, the voting system testing lab to verify, and the Colorado Secretary of State to ensure that minimum Federal Voting System Standards were met, as required by law, is inexcusable and grossly violates industry standards. Only after this software was noted in an expert report, dated December 13, 2020, and submitted in connection with a widely publicized vote switching controversy in Antrim County Michigan involving DVS D-Suite systems, did DVS submit an application for certification for version 5.13-CO, dated Jan. 13, 2021 which listed SSMS as an installed software component.[38]

Name	File Ext	Logical Size	Category	File Created
Ssms.exe	exe	720,632	Executable	05/17/17 06:49:44 AM (-4:00 Eastern Daylight Time)

Figure 1 - SSMS Installation Date on Mesa County EMS server

The Colorado Secretary of State should have been aware that this separate software component (a completely separate download from Microsoft) was required to be listed on the application for certification, tested by a federally-accredited lab, and certified. The addition of MS SQL Server Management Studio is not necessary to the election process, and allows any party with access to the EMS server to alter cast ballots, tallies, databases, ballots, and audit records with up to full administrative permission.

Examination revealed fundamental flaws within the security configuration of the Mesa County Election Management System (EMS) server used in the November 2020 general election and the April 2021 Grand Junction municipal election that show conclusively that this voting system and its software, as delivered by Dominion Voting Systems and certified by the Colorado Secretary of State, is uncertifiable under Colorado law because it contains unauthorized, untested and uncertified software in violation of the law, is configured in a manner that violates mandatory VSS and industry best-practice security standards, allows "intentional manipulation and fraud" that the VSS standard prohibits, and fails to log system events and preserve audit trails required by VSS in a manner that makes determination of election integrity extremely difficult, and maybe impossible.

---

[37] D:\Program Files (x86)\Microsoft SQL Server\140\Tools\Binn\ManagementStudio\Ssms.exe.

[38] See Antrim Michigan Forensics Report, Allied Security Operations Group, December 13, 2020.

Nationwide, various election officials have denied qualified third-party investigators the access to election system equipment including logs, network and security equipment configurations, and network diagrams, that might allow the detection of unauthorized access and operation of voting systems. This report demonstrates why this is a dangerous development because the denial of access prevents the discovery of the full extent of the failure of election security and election records integrity.

The techniques used in this report employ basic network troubleshooting techniques that can readily be executed by persons with minimal skills. In fact, software found to be already installed on the EMS server (Microsoft SQL Server Management Studio was downloaded and installed on the test workstation, while Fing and SQL Pro from the Apple App Store were installed on an iPhone). In each instance, the software was launched and access was granted. It was so simple that calling the test an "attack" is almost inappropriate, since standard publicly-available software was used without modification and connection was made in an industry standard manner to the default port assigned for SQL databases.[39] The server had no security implemented other than userID and password, and even that is easily bypassed.[40] In this case it was not a smart examiner but the exceptionally insecure configuration of the voting system that was at fault in failing to meet the requirements of law. That exceptionally insecure configuration is an open invitation to the average hacker, and indeed almost anyone with basic skills, to be able to change election results.

But it is not the average "hacker" or even cyber-criminals that provide the greatest threat to election integrity. While it has been stressed that these *relatively simple* intrusions could be done by anyone with a reasonable understanding of networks, the fact is that nation-state adversaries have long attacked and subverted the critical infrastructure of the United States,[41] as documented in Appendix D. The extreme sophistication of these nation-state actors' cyber threat capabilities has persisted for decades, evolved far beyond the knowledge of the average citizen, and the history of publicly-known attacks document it beyond question. Malicious actors, including foreign nation-states, our most capable and persistent adversaries, already know how to subvert insecure systems, like this election infrastructure.

The evidence of foreign interest in our voting systems is too important to bury in a footnote: four (4) Korean students, at 2 different Korean universities, authored the paper A Study of Vulnerabilities in E-Voting System, Xing Shu Li, Hyang ran Lee, Malrey Lee and Jae-young Choi, *Advanced Science and Technology Letters Vol.95 (**CIA 2015**)*, pp.136-139, https://www.researchgate.net/publication/315040247_A_Study_of_Vulnerabilities_in_E-Voting_System. Section 2 discusses "hybrid election systems" that are exactly what the Dominion Democracy Suite elections systems are.

Continued suppression of the knowledge of this system's extreme security failures, long known to foreign nation-states and others, does not further the security of critical infrastructure election systems – indeed, elections have taken place and are ongoing while these known security failures have been left unaddressed.

---

[39] The standard port for SQL database access is 1433. When this port is found open, it is obvious that it provides access to a database system. The port number can and should be reassigned to another number to improve security, making the discovery of database access more difficult, and is an example of multi-layered "Defense in Depth."

[40] Appendix K.

[41] https://www.whitehatsec.com/blog/2020-election-security-the-urgent-need-to-address-vulnerabilities-in-voting-systems/

For example, in his September 21, 2021 Declaration, Professor Halderman attached an email string with CISA dated August 18-19, 2021, wherein he requested that the federal district court allow him to immediately provide his sealed expert report to CISA because of the threat posed to the election systems in sixteen states—including Colorado—by DVS machines with ICX software that can be used to "steal votes." In that August, 2021, exchange, CISA agreed to receive Halderman's expert report detailing these security failures. However, even though Professor Halderman testified in his Declaration that this threat was "urgent," and that it would take "months" to fix these "critical vulnerabilities," CISA inexplicably waited to even seek Prof. Halderman's report until more than five months had passed—to January 21, 2022.[42] The voting systems Halderman described as critically vulnerable were used in the November, 2021, elections in the U.S., including in Colorado. Thus, the suppression of knowledge of security failures has indeed harmed election security and faciitates continued malfeasance.

The security and configuration of the equipment images examined to date leaves no doubt that our voting systems are dangerously insecure, and renders absurd any claim of election integrity.

This examination has demonstrated the ability for any individual to change the calculated vote totals in the internal database tables used in an actual election, bypassing any Dominion Voting System software security and access controls, with no record preserved in log files that are meant to comprise an audit trail of election records. It demonstrates how trivially election results data can be tampered with and even changed completely by someone with physical access to the EMS server, or by using a non-DVS computer attached to the network, or even by using a cell phone or mobile device if wireless access has by any means been enabled on the network.

[42] Statement of Interest [by CISA], *Curling et al. v. Raffensperger et al.*, 1:17-cv-02989-AT, Docket No. 1269-1 (filed February 10, 2022), (ND Ga.).

EXAMINATION METHODOLOGY

Description of the Examined System

The voting systems used in Mesa County, Colorado, like other systems used across the state and the nation, are made by Dominion Voting Systems (DVS). Many of these voting systems are comprised of an industry-standard computer[43] that uses a Microsoft operating system and a combination of proprietary Dominion application software and non-proprietary, commercially available software. This provides a foundation for election-related functions including creating election projects, defining ballots, capturing and storing the election data in a secure database management system, tabulating and counting the votes, and reporting election results.

The Mesa County Election Management System (EMS) server runs on the Microsoft (MS) Windows Server 2016 operating system, and it employs a database management system known as Microsoft SQL Server (SQL Server). The security of the server depends largely upon the proper configuration of the operating system, network, and the SQL Server.

The design of the voting system includes the functional capability to adjudicate ballots that the computer cannot accurately interpret. Adjudication, in this regard, means nominally, that a person sits in front of a computer terminal, a ballot image is shown on the screen, and this person chooses the option that they feel the voter intended to choose. Adjudication is facilitated by a software application that runs on the EMS system (part of the DVS software) and, normally on one or more Adjudication workstations. If unauthorized code is executed on the EMS system, including on Adjudication workstations or other DVS workstations authorized to be connected to the EMS server, or if an unauthorized user is accessing or has accessed an Adjudication workstation, the adjudication function may be executed to adjudicate ballots without the intervention or knowledge of any authorized operator.

This process requires that the EMS server (which stores and provides access to the election databases and ballot images) be connected to a network. While necessary for the adjudication function to work in the present design of the voting system, this design requirement significantly raises the risk of abuse, especially considering the failure to implement required security.

The Mesa County election director at the time reported that the D-Suite 5.11-CO network consisted of a single network switch connecting only specifically-designated components of the voting system, including the EMS server, adjudication workstations, an EMS server client workstation hosting the Election Event Designer (EED) software, and a Network Attached Storage (NAS) file server.[44] DVS documents the connection of these systems in their manuals. Therefore, while the EMS server may not have been directly connected to the Internet (it is impossible to rule out, without access to all logs which should have been generated and preserved), it was connected to other computers via a network to allow specific voting system devices to communicate with each other. These other computers must be fully examined to assure

---

[43] An "industry-standard" computer is comprised of common components (motherboard, bus, memory, processors, communications, input/output ports) in a common architecture, e.g., the type of computers one purchase in big box stores and find in use in a home-use or business setting, running office productivity and web-browsing software.

[44] The term Network Attached File Server is, in this case, a misnomer. DVS uses the term NAS, however it is a shared disk drive on the EMS server itself. In this report, I may use the term synonymously, but there is a difference that will be noted where relevant.

that no connection to external devices or networks (including the Internet) occurred, because connection to other computers exposes the EMS server to a common "Island-Hopping attack,[45]" which is where every device attached to the EMS network may have a direct or indirect path to and from a device or network outside of the election network, providing a path for an attacker's movement through networked devices to the target. For example, the computers in a home are typically all connected to each other via a wired and/or wireless network, and because the home router is connected to the internet, all devices in that home also have a path to the internet.

The voting system network (based on DVS manuals, EMS server image information, and election official input) was reproduced, both with a virtual network environment and again with a physical Ethernet network composed of cables and a small desktop network switch, to allow the network connection of a Test Workstation used in this report. This configuration was used to test access to the EMS server by a person sitting in front of the EMS server, and again to test access to the EMS server by even a non-Dominion computer that connects to this network. To test whether access from a device with more limited capability such as a mobile phone was possible, a wireless access device was added to the network to simulate the hardware used in Mesa County and the enabling, through misconfiguration or malicious action, of one or more of these wireless devices to provide access, even temporarily. Because I did not physically see or examine the original setup of the voting system network in the Mesa County facility, and due to the destruction of log data by both improper configuration and the overwriting of log files, it is not possible to provide conclusive forensic verification that the voting system was not connected to unauthorized external networks or devices, including wireless devices.[46] It should be noted that seven internal wireless adapters, and twenty-eight wireless-equipped ICX devices, were ordered as components of the Mesa County DVS D-Suite system, as supplied by DVS. In addition, a Dell E310DW wireless-capable network printer was configured as the default printer on the Mesa County EMS server. This brings the total number of wireless access devices to a total of thirty-six devices.

The EMS server has a software firewall. The purpose of having a firewall is to address the risk of access to the EMS server from all unauthorized devices, users, networks, methods, ports, Internet Protocol (IP) addresses or groups of addresses, and during specific time periods. However, a firewall must be specifically configured (programmed) to perform these functions. One risk of a software firewall is that all users with administrative access can change its programming because it resides on the EMS server; a separate hardware firewall device with its own non-shared password mitigates this risk. Per the VSS and required

---

[45] In an Island-Hopping attack, a threat actor gains access to a target computer remotely, through other, connected computers or devices. E.g., a target computer (which we'll call "A") is connected to computer or device "B" (e.g., a network printer). Computer or device "B" is connected to computer or device "C" and computer/device "C" is connected to computer/device "D". It is not necessary that they all be connected in a single physical network. In fact, most modern computers have one or more wireless communications devices; such a wireless capability could allow the access that enables an Island-hopping attack. It is not necessary that the connection be of long duration. The attacker might enter and compromise computer "D" from the global Internet over a wireless connection, determine that computer "C" is connected, break-in to computer "C, move through its connection to computer "B," and finally to computer "A" (which is may be particularly vulnerable if there is an assumed trusted relationship/connection between computers "B" and "A." This chain of connection and intrusions ultimately allows the complete compromise of the target computer.

[46] More detail will be provided in a subsequent forensic report.

by Colorado Law,[47] risks that must be addressed by a voting system include "Preventing access to vote data, including individual votes and vote totals, to unauthorized individuals." The EMS server firewall was found to be programmed specifically to permit access to back-end database services, enabling access to vote data and vote totals[48] on the Mesa County EMS server from ANY IP-address, globally, at any time. This configuration fails to meet requirements in the law, as well as every industry best practice recommendation for firewall rule configuration.

SQL Server, a database management system (DBMS), installed and used on the EMS server (which stores and manages the election databases) is accessible using any software tool supporting connection to SQL Server, employing Windows Authentication. One of the most common and freely available tools is known as Microsoft SQL Server Management Studio ("SSMS"). SSMS is free and available to download from Microsoft from any internet connection. In this examination it was downloaded from Microsoft, installed on the test workstation, and in a matter of minutes, used to easily and directly access the back-end election database and change any data in it. Searching the internet for 'how to install SQL server management studio,' the first result was: https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15, which walks anyone through installing the software while other readily-accessible online videos walk even a novice through the installation.

But even that is not required for anyone with physical access to the EMS server, because SSMS software was found already installed on the Mesa County EMS server image. This software is not on the list of certified software for DVS D-Suite 5.11-CO nor reasonably expected on a voting system, due to the vulnerability it introduces. This addition in itself violates the stated certification of the voting system.

Software (SSMS) that allows direct access to the back end of the election results database and allows changing vote totals was found installed and functional on the Mesa County EMS server. The software firewall, that could have severely restricted access, was programmed instead to allow access from anywhere in the world. Although the VSS does not specifically address firewall configuration, it does specify addressing this kind of risk, and the firewall, supplied by Microsoft as part of the computer operating system could have and should have been programmed to limit access, at a minimum to only those Mesa County devices required to connect to the EMS server (the few other DVS D-Suite computers and devices necessary could be restricted by their specific IP addresses, for example). Such a configuration would also prevent the wireless access demonstrated by my tests and documented in this report, by disallowing its connection, had the firewall been used to control this database access (port 1433, or an alternate port, explained later in this document). However, given the presence of internal wireless devices as part of the DVS D-Suite system, a properly configured firewall rule on the EMS server that restricted access from only other Dominion devices on that network still may not prevent unauthorized access from occurring through the individually-authorized yet wireless-capable devices.[49] Possibly most alarming, I found a firewall rule that allows global (from anywhere in the world) access, is not supplied by Microsoft, and must have been explicitly created. Allowing global access is extraordinarily irresponsible, particularly given that SSMS

[47] See VSS Volume 1, section 6.1.

[48] This firewall could have prevented access but instead specifically allowed it.

[49] This means that the security implemented on every one of these connected devices must be as strong as that of the server that holds and tabulates ballots.

enables direct access to the vote data. This dangerous combination constitutes what is commonly known as a "back door" into the voting system, and together with deleted audit trails presents an undetectable path for unauthorized access to, and illegal manipulation of, election data. The failure of the software firewall is not the only access control that was misconfigured. Access control mechanisms in the DBMS itself failed to prevent the access demonstrated in these relatively simple tests.

It must be emphasized that this test was done on a virtual replica of the Mesa County EMS server, created from an image of that EMS server's hard drive, and not on the actual in-use election system.[50]

For all practical purposes, the term "Mesa County EMS server" is used to mean the logical image[51] of the Mesa County EMS server recreated from the forensic, integrity-controlled Encase Forensic Archive of the actual Mesa County EMS server. The original forensic image of the system was obtained using Access Data's Forensic Tool Kit Forensic Imaging software. Access Data is an industry-standard forensic software vendor. I had no access to the actual Mesa County EMS server hardware and have relied upon forensic images of that server furnished by legal counsel to create a virtual replica of the EMS server.

Access was attempted and established to the (replica) EMS server to determine the degree to which the EMS server was secured in accordance with legally-mandated VSS standards. The results were alarming. It was found that the SQL Server databases on the Mesa County EMS server were unprotected, beyond a simple password that can be bypassed.[52] While many potential security restrictions were possible, it was found that surprisingly few were implemented. The SQL Server software on the EMS server was set up with a Windows Firewall with Advanced Security features, however, an explicit firewall rule on the EMS server allowed access directly to the SQL election databases back-end from any IP address in the world.

Security settings relevant to the SQL Server and access to the databases were examined. A subsequent report will address the comprehensive security implementation. This report focuses upon the EMS server's failure to protect the election databases and the ease with which they can be accessed by any bad actor to change election results.

[50] A forensic image of a hard drive is a bit-for-bit copy of the user accessible data storage area residing on the data storage mechanism used by the computer system. For a complete discussion of this definition, see Appendix J.

[51] The exact view of disk storage data as seen by the EMS server computer.

[52] Appendix K.

FORENSIC ANALYSIS

SYSTEM IDENTIFICATION

The Mesa County, Colorado EMS server analyzed in this report is capable of operating on a local area network (LAN). The network consists of several systems, including servers and workstations. The server that was evaluated was named EMSSERVER. It is running the Microsoft Windows Server 2016 operating system.

The forensic evaluation and reviews were based upon a forensic image[53] archive collected from the Mesa County EMS server. The forensic image of the EMS server examined in this work was collected on May 23, 2021, before the Secretary of State staff, assisted by DVS personnel, installed their "Trusted Build" software update, as documented below. The serial number of the hard drive shown in the collection data set verifies the data origin to be the physical device.

The backup image was obtained, using forensic imaging methods (an AccessData FTK Imager), from the DVS D-Suite EMS Standard Server, version 5.11-CO, in Mesa County, Colorado, as used in the November, 2020 election. The acquisition data are presented in Figure 2.[54]

---

[53] A forensic image (forensic copy) is a bit-by-bit, sector-by-sector duplicate of a physical storage device's user accessible storage area using specialized hardware and software. To be technically accurate, hard drives contain a "service area" that is not accessible by the user or the Operating system, nor by forensic software; this service area is accessed by the drive's internal controller. The service area is used by the firmware in the disk drive to identify defects in the media introduced during manufacture as well as those identified during operation. Making a perfect magnetic storage platter would be prohibitively expensive thus they are made to be fault tolerant, and the defective areas are simply skipped by using a defect-map. Forensic imaging is a much more comprehensive representation of the state and configuration of the imaged system than could be obtained using simple file backup methods. Forensic Imaging copies data from the subject data storage media without altering the original data in any way. The image includes all files, folders, and unallocated, free, and slack space as well as copies of internal Microsoft files that are protected from access during a normal backup (including the MS "Registry database" and other protected files). These forensic images include not only all the files visible to the server operating system but also deleted files and fragments of files left in the slack and free space as well as every digital bit of data present on the storage medium. When multiple disks are configured into a Redundant Array of Independent Disk (RAID) array, the RAID controller provides a "logical view" of every bit on the media to provide a sector-by-sector bit-for-bit copy of the storage medium; this permits, for example, the use of two identical disk storage devices to provide double the space of a single device, or two devices configured as mirror images of each other to provide failure redundancy. While there are many different configurations for RAID subsystems, a RAID subsystem provides the exact same view of the storage medium and data access to a forensic imaging process as it does to the computer in which it is installed.

[54] To the extent that personal identifying information was identified in Figure 2, it has been removed. This in no way affects the accuracy of the findings in this report or the evidence.

```
Created By AccessData® FTK® Imager 4.2.0.13

Case Information:
Acquired using: ADI4.2.0.13
Case Number: 052321
Evidence Number: 00003
Unique description: EMSSERVER

---------------------------------------------------------------

Information for F:\EMSSERVER\EMSSERVER:

Physical Evidentiary Item (Source) Information:
[Device Info]
 Source Type: Physical
[Drive Geometry]
 Cylinders: 121,534
 Tracks per Cylinder: 255
 Sectors per Track: 63
 Bytes per Sector: 512
 Sector Count: 1,952,448,512
[Physical Drive Information]
 Drive Model: DELL PERC H730 Adp SCSI Disk Device
 Drive Serial Number: 00222e64128c016e1d004fc54220844a
 Drive Interface Type: SCSI
 Removable drive: False
 Source data size: 953344 MB
 Sector count:    1952448512
[Computed Hashes]
 MD5 checksum:    3d7cf05ca6e4 2db765bf5c15220c097d
 SHA1 checksum:   eab06a7ea23586de2746b9142461717e075f5c9f

Image Information:
 Acquisition finished:  Sun May 23 2021
```

Figure 2 - Mesa County, Colorado EMS server (5.11-CO) Forensic Image Attributes

AUTHENTICITY

When forensic images are acquired, a hash function[55] is computed. This hash function is far more than a checksum, despite the "checksum" reference in **Error! Reference source not found.**. The mathematical complexity of the hash function is sufficient such that there is only an infinitesimally small probability that any two different source files can produce the same resultant hash.[56] This hash can be used at any time to validate the integrity of the image to ensure that it has not been edited, modified, or changed in any way. The hash function result from the acquisition of data appears in the text above but also appears inside each respective archive and authenticates the data by demonstrating it has not changed since it was acquired. Moreover, two different hash functions (MD5 and SHA-1) are in the image and have never been shown to be simultaneously compromised in the same attack.

The hash function results were compared and match the data from the original collection of the forensic image. This provides the greatest mathematical assurance possible that the data in the forensic image examined is a true, authentic and unaltered copy of the original disk data.

Further confirmation that these are genuine images from the Mesa County EMS server has been provided by the Colorado Secretary of State's office. See:

https://www.sos.state.co.us/pubs/newsRoom/pressReleases/2021/PR20210817MesaCounty.html[57]

Chain of Custody

Digital chain of custody is the record of preservation of digital evidence from collection to presentation in the court of law. This is an essential part of the digital investigation process. The chain of custody is probative that the digital evidence presented to the court remains as originally collected, without tampering. The image analyzed in this report was obtained through AccessData FTK Imager 4.2.0.13.

---

[55] A hash function is a mathematical algorithm that converts an input (e.g., the bits of a file, or all the files on the hard drive) of arbitrary or variable length into an encrypted output of a fixed length. The purpose of the hash in this case, is to create a "signature" for the file or hard drive, such that any other party at any other time, can compute the hash of the file, files or hard drive and confirm that they are identical, because the hash outputs match.

[56] While the SHA-1 128-bit algorithm has been found possible to compromise, the attack required 9,223,372,036,854,775,808 computations of the algorithm. This is the equivalent of 6,500 years of single-CPU computations or 110 years using today's modern Graphics Processing Units (as used in mining cryptocurrency). This attack required the use of two specifically-designed different files that produce the same hash, created by expert mathematicians explicitly for this purpose. Such an attack may be within the capability of a Nation-State or by spending an enormous amount on cloud computing. In its application as a sophisticated checksum, the effort to change an original dataset into a specific altered dataset with the same hash would present astronomical difficulty much greater than the 9.2 quintillion (quintillion means x 10^{18}) computations in the attack referenced here, would require extraordinary resources, financing and would be exceptionally difficult to conceal. The likelihood of this occurring is infinitesimally small. The likelihood of this occurring undetectably is virtually zero. The probability of two different message digest algorithms being simultaneously fooled is nearly impossible and has never been shown to be possible.

[57] Reproduced in Appendix M.

I have reviewed the documented chain of custody for the image and have determined that the chain of custody is complete from the forensic operator utilizing FTK Imager through the source from which I directly received these images. (Because of the pending civil litigation and criminal investigation, the written documentation remains in the custody of counsel for later introduction in court proceedings and thus is not included as part of this report.)

Tools Used

The initial forensic image was acquired using Access Data FTK Imager. Once acquired, Encase Forensic was used to maintain forensic integrity of the archive. Autopsy, Encase Forensic, FTK Imager and Oracle VirtualBox were used to analyze the image. All findings were verified with Encase Forensic examination of the integrity-controlled forensic image.

TEST PREPARATION

The Mesa County EMS server forensic Image was used to recreate a complete and exact replica of the Mesa County EMS server's software, operating system, and even boot code, which was then launched in an Oracle VirtualBox[58] virtual computer environment for the examination. This technology is commonly used in software development and testing. This exact replica was used for this examination.

The image was evaluated to gather technical information, including the integrity of the data stored on the system. No effort was made in this analysis to reverse-design, de-compile, or reverse-engineer the compiled binary Dominion Voting System software. Operating system configuration relevant to the operation of the system as well as DBMS configuration was examined. Results relevant to this investigation are documented.

Screenshots are presented that can be used to review and verify these findings and provide step-by-step instructions to reproduce and validate these results. The security of the system has been compromised by the vendor, the Voting System Testing Lab and the Secretary of State's unlawful certification that the system meets all the requirements in law, and exacerbated by false statements that voting systems are safe, secure and have strong integrity. These test results verify the fact. These screenshots were obtained from the mounted forensic images of the EMS server. These results can be reproduced by anyone.

While many of the EMS server settings can be determined from operating system configuration records, it is much easier and far more understandable to view the same information with the Microsoft applications designed for this purpose. The software that serves as the host for the DVS D-Suite voting system applications is the intellectual property of Microsoft, e.g., Windows, SQL Server, and SSMS. The configuration values, or "settings," are determined by the end user, in this case DVS or the Secretary of State of Colorado, but are not proprietary. These are the settings that must be examined, as part of a comprehensive examination, when a voting system is tested for certification.

---

[58] The VirtualBox environment provides all of the resources that a server provides, including central processing units (CPUs) and network interfaces. Virtual means that many of the functions normally executed by dedicated computer hardware are instead performed in software, and the interfaces present on the original server are emulated by the host computer's interfaces. None the less, a virtual environment allows us to operate an operating system and application programs *as though* they were running on the actual server hardware.

The security of the entire voting system depends on the totality of all the hardware and software, *combined with* the configuration settings and records of system activity preserved in system log files. Similarly, the security of a home depends not just on having 3 doors and 21 windows, but also whether each of them are locked, as well as whether each of them are monitored on video (equivalently, access being logged) and whether they are each monitored by an alarm system.

The design of the system can be more secure or less secure, inherently, just as a house with 1 door and 1 window is more secure than a house with 10 doors and 20 windows. But voting system testing labs (VSTL) are explicitly required to check and verify these critical settings.

Below are presented screenshots from two different computers used in the testing environment. Each step is explained in detail so that one can easily follow along.

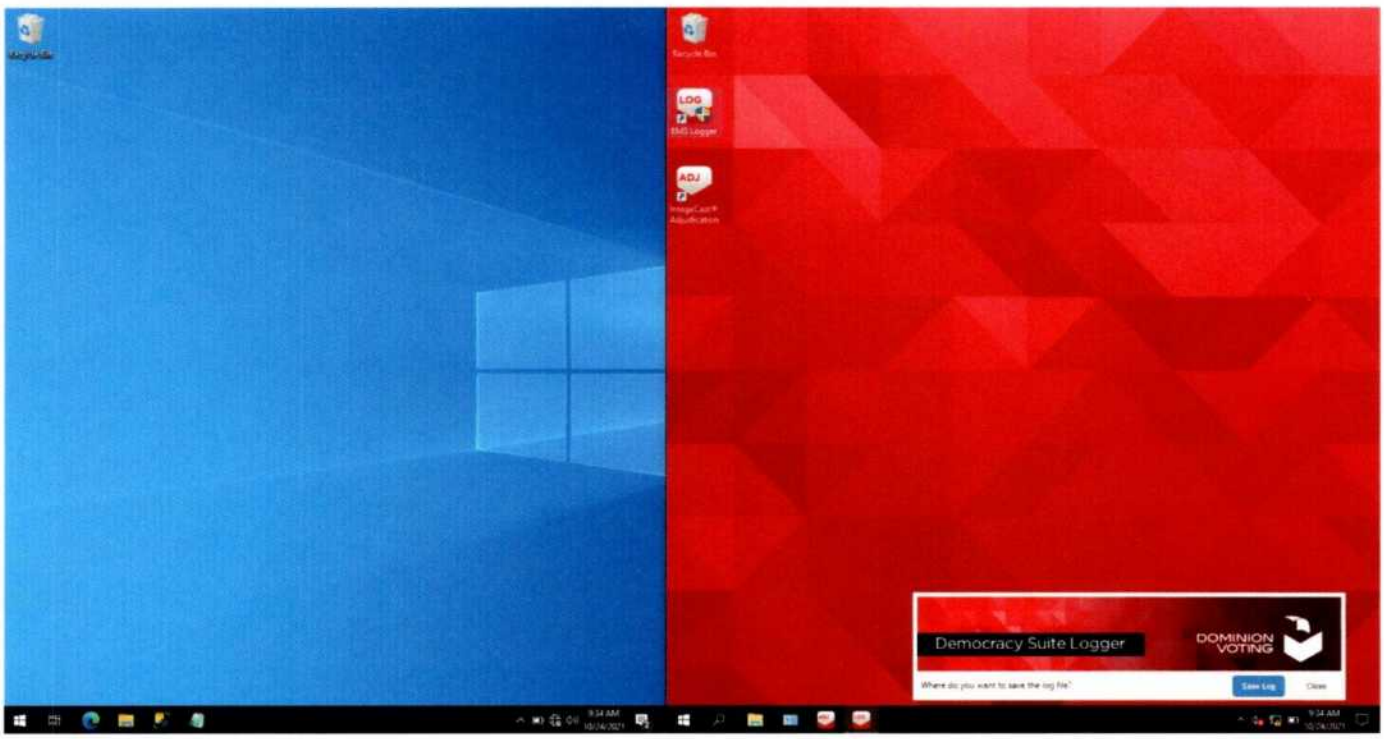


Figure 3 - Test Workstation and Dominion EMS server

On the left side in blue is the Test Workstation running the Microsoft Windows 10 operating system that was used as part of testing. On the right side in red, the emulated Mesa County EMS server, from the EMS Server image, is displayed. The EMS server operating system is Windows Server 2016 and is configured exactly as it was when the image was taken on May 23, 2021. These computers are connected to the same network[59] for testing.

[59] The EMS server has its IP address assigned as 192.168.100.10, just as it was while in operation in Mesa County. The Windows 10 computer is also set up on the same 192.168.100.0/24 network just as any device could have been connected at Mesa County. The figures shown in this report are taken from two "virtually" connected virtual

Both systems are hosted in Oracle VirtualBox virtual environments on an isolated virtual network (emulated within VirtualBox) for the first test – these two computers[60] are the only computing devices connected to this virtual network.

The tests were repeated a second time using a physical network connection from a stand-alone test workstation with Windows 10 (within a separate Oracle VirtualBox instance, for forensic sterility) connected by Ethernet cable to a Netgear GS108 gigabit network switch, and then to the VirtualBox instance of the Mesa County EMS server's host computer.

This implementation, and testing with a physical network, together, exactly mimics the functionality of the Mesa County EMS server because it is running the exact operating system and application software, identically configured because it is an exact copy created from the integrity-controlled forensic image. Thus, its response and security controls are identical and well-suited for examination in this manner.

The Mesa County EMS network was connected to other components of the EMS D-Suite, but these components neither participate in, nor could prevent the accesses demonstrated in this test (if not compromised and exploited). They are, with respect to the conclusions of these tests, irrelevant, notwithstanding the possible additional data paths to external networks they may offer in either direction.

---

environments on a single computer, but the results were verified and duplicated using two different computers and a physical network and network switch, i.e., the test's connection between the two systems made no difference on the results obtained.

[60] The reference to "Computers" in this paragraph specifically refers to the operational system comprised of electrical computing devices which perform identical functions and the software installed and configured to operate those devices. For example, an Intel i7 Central Processing Unit (CPU) performs identically on every computer motherboard provided that all of its features are properly included in the electrical design of the motherboard. The main characteristic of a computer is determined by the Operating System, its configuration, and the application software and its configuration. Thus it is entirely appropriate to examine the Operating system, application software and their respective configurations to understand the computer system's operational capability and function. The reference to the software as "computers" is intended to describe the software's purpose, capability and functionality as used in Mesa County as a computer system, not to a specific device.

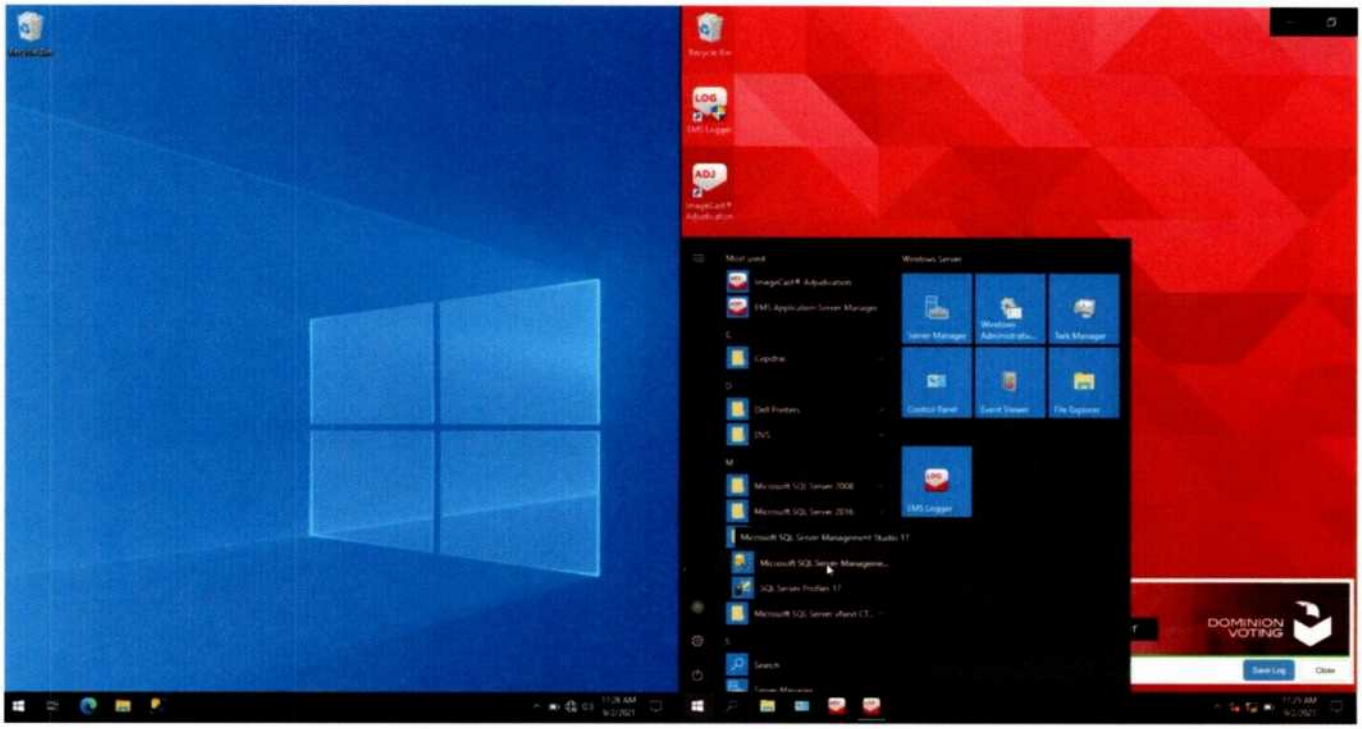


Figure 4 - Installed Microsoft Software

As the Dominion EMS server was examined, the installed Microsoft SSMS software was found listed on the Start Menu.

The presence of SSMS software on the EMS server was unexpected because it enables direct access to the EMS server databases, bypassing the DVS application software. Properly-designed software developed with security in mind would strictly require all database access of any kind (including backup and maintenance) to go through security/tracking/auditing components as part of the design.

The very dangerous side effect of having or allowing Microsoft SSMS software on a voting system is that it can enable surreptitious access to the voting database and is a concern if it is configured to allow such access. Therefore, it is necessary to examine the EMS server's entire software configuration.

Finding 1: The Mesa County EMS system used in the 2020 General Election had Microsoft SQL Server Management Studio 17 installed as configured by Dominion Voting Systems. This software is not listed on the official test report or application for certification. As it was not tested, the unauthorized installation of this software violates and renders illegal the certification of the voting system for use in an election.

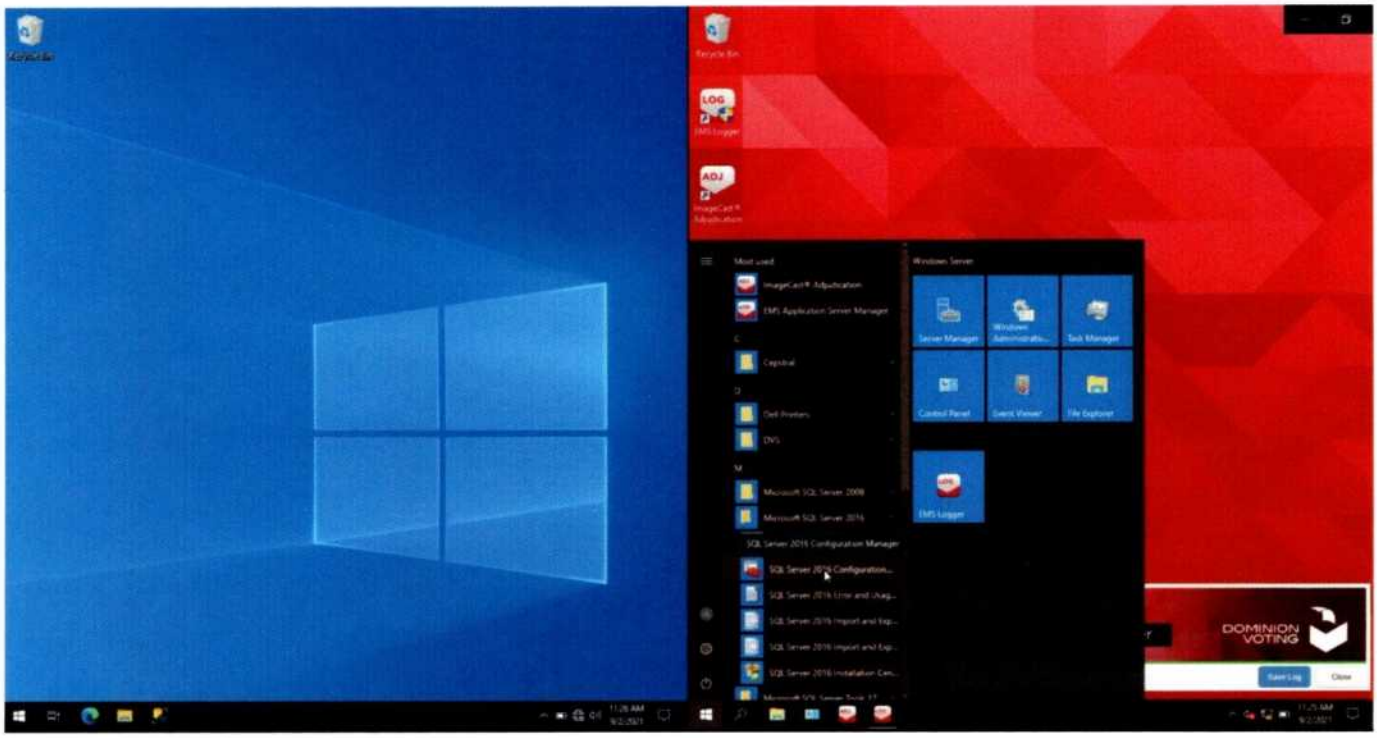


Figure 5 - SQL Server 2016 Configuration Manager

To determine how the SQL Server is configured and whether unfiltered and uncontrolled access is permitted, I examined its configuration through the software application provided by Microsoft entitled "SQL Server 2016 Configuration Manager" as shown in Figure 5.

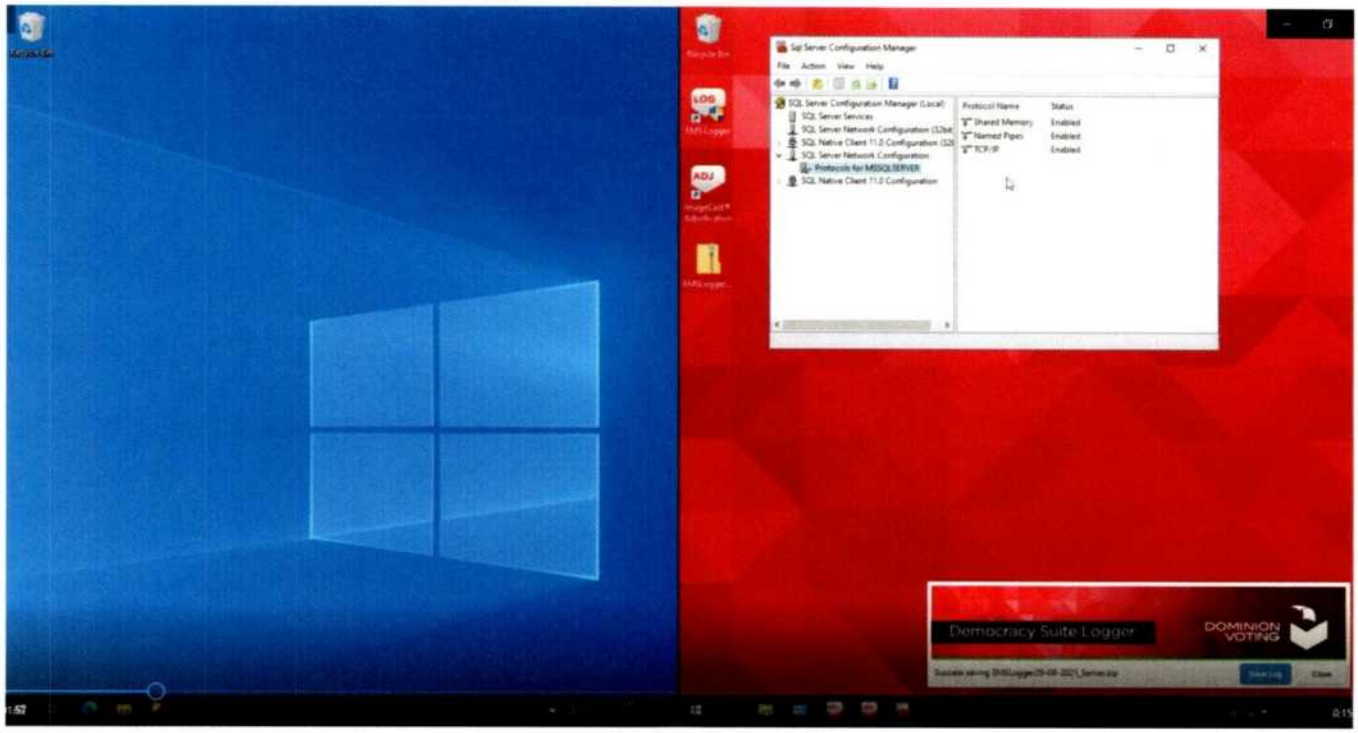


Figure 6 - SQL Server 2016 Configuration Manager – Network Protocols enabled

All three of three possible SQL server protocols were left "Enabled," providing pathways to the database above what are required for operation. These extra pathways can severely reduce system security.

Under the SQL Server "Network Configuration" the menu item is selected titled "Protocols for MSSQLSERVER" that shows that more protocols are enabled than should be, especially for a "secure" system. While one of these may be necessary, all three being enabled presents an unwarranted risk.

Protocol Name	Status
Shared Memory	Enabled
Named Pipes	Enabled
TCP/IP	Enabled

Microsoft states, in its SQL server documentation[61] that:

"To enhance security, SQL Server disables network connectivity for some new installations. Network connectivity using TCP/IP is not disabled if you are using the Enterprise, Standard, Evaluation, or Workgroup

[61] https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/default-sql-server-network-protocol-configuration?view=sql-server-ver15

edition, or if a previous installation of SQL Server is present. For all installations, shared memory protocol is enabled to allow local connections to the server."

For an election management system, it is entirely inappropriate and irresponsible to enable Shared Memory or TCP/IP access over an unsecured network connection, and particularly careless and irresponsible to enable these together with "Named Pipes." Shared Memory access permits an intruder to install malicious software and to execute arbitrary commands with full administrative privileges if exploited. Given the exceptionally minimal protection implemented on this server, if any connection were made to a network that provides a path to the Internet[62] by the EMS system, any other computer connected to the Ethernet network would be granted access to the TCP/IP ports[63] enabled by the EMS server and a hostile party would be able to penetrate and alter the EMS server.[64] In the examined state of the EMS server, if this network *or any computer connected to this network* were connected to the internet either directly or indirectly, by wire or wireless, a hostile party *anywhere in the world* would be able to penetrate and alter the EMS server, including altering actual election records, like tabulated vote databases.

A computer system configured in this manner should never be used in any critical infrastructure or high security environment and, as a voting system, should be immediately decertified and those responsible for creating and selling such system investigated.

While multiple security mechanisms exist within a Microsoft Windows 2016 server, including the Microsoft Windows Defender firewall, SQL database permission restrictions, Operating System security Policy, Group Security Policy, file access control lists, and much more,[65] some were configured not to protect the server but instead to allow all "local" and "remote" access. Tests conducted in this examination demonstrate that not only are those explicit programmed settings misconfigured, but that no other security mechanisms within the installed hardware and software prevented the ability to access and change election data, or even to provide any warning of such drastic and consequential access.

---

[62] Given the exceptionally large number of wireless devices in this election infrastructure (thirty-six), particularly in the context of the plethora of improper security configuration mistakes made in this installation, examination of every device in the infrastructure including the wireless printer must be undertaken before the network can be considered secure; absent appropriate systems log data, such a determination might not be possible.

[63] TCP/IP networks identify computer systems by their IP (Internet Protocol) address. TCP/IP further identifies the specific service (email, file transfer, database access, etc.) to be used on the destination computer using a port number transmitted within the beginning of the packet (in its header). Standards identify the assignment of port numbers to specific services, for example, web browsing uses port 80, encrypted web brmwsing uses port 443, email uses port 25, and database access using the Structured Query Language (SQL) uses port 1433. There are 65,536 available port numbers. Ports 0 through 1,023 are assigned to commonly used services/protocols, 1,024 through 49,151 are sometimes registered to a specific service, and those remaining are available for dynamic use (e.g., as needed). One can conceptually think of these ports in the same way we think of channels on cable TV – each is associated with specific content.

[64] For example, see CVE 2018-8273, CVE 2021-1656, CVE 2020-0618 at http://cve.mitre.org and Microsoft Knowledgebase KB 4073225 regarding the "Meltdown" and "Spectre" vulnerabilities presented by the "management engine" back door in every CPU manufactured since 2007 whether Intel, AMD or ARM processors.

[65] See the US Department of Defense Security Technology Implementation Guides (STIGs), at http://public.cyber.mil

There is a great misunderstanding about intrusion into computer systems. Many people conceive of it as depicted by Hollywood, where an intrusion takes several minutes or significantly longer. While this makes for good drama, it is not realistic at all. In the real world, malicious actors – particularly hostile nation-states, e.g., China, Russia, North Korea and Iran to name a few, have extremely sophisticated cyberwar capabilities. They are capable of intruding and *altering data* in a matter of less than a few seconds and they engage in persistent cyber operations to penetrate and compromise supply chain, industrial base, trusted vendors, academia, and government offices which might someday afford access.

Intrusion can be accomplished without a direct connection to the target computer. In the case of a voting system, using the example of an Adjudication Workstation connected via wired Ethernet to the EMS, if the Adjudication workstation has a wireless (Wi-Fi) interface, such a connection might be automatically connected to external devices and networks without the EMS or Adjudication workstation operator ever noticing it, especially since all laptops today have both wired Ethernet and Wi-Fi interfaces which might enable an Island-Hopping attack. Thirty-six (36) wireless devices were identified in the Mesa County DVS D-Suite system (e.g., on the DVS D-Suite ICVA computers and ICX tablets and one Dell E310DW wireless printer, with IP address 192.168.100.11, set as the default printer on the EMS server). Any other connected device, including a printer like the one installed on the Mesa EMS infrastructure,[66] creates an increase in this risk exposure. This is why an Internet connection in any device or computer, even several connections removed, is so extremely dangerous to critical systems. To mitigate this risk, the US Department of Defense (DoD) maintains special closed networks for sensitive information, which are forbidden to have internet connections or connection to any system with an internet connection.

Appendix D lists some of the more notable nation-state cyber-attacks as well as a link to an online video of one cyberattack that completely destroyed a power generation facility. Adversaries constantly scan and probe every computer on the internet, and through those computers, other devices and computers not directly connected to the internet, to identify weakness well in advance of the need for an attack. Today's attacks occur very quickly, in a matter of seconds.

[66] At Bell Laboratories in the 1980's, printers that used the Postscript language were exploited (to leverage their computational power) in this manner because they were the first to have a bi-directional communication connection (e.g., able to talk back to the host computer, over a network). Today's printers all have this capability and present a risk of being a component of an Island-Hopping attack.

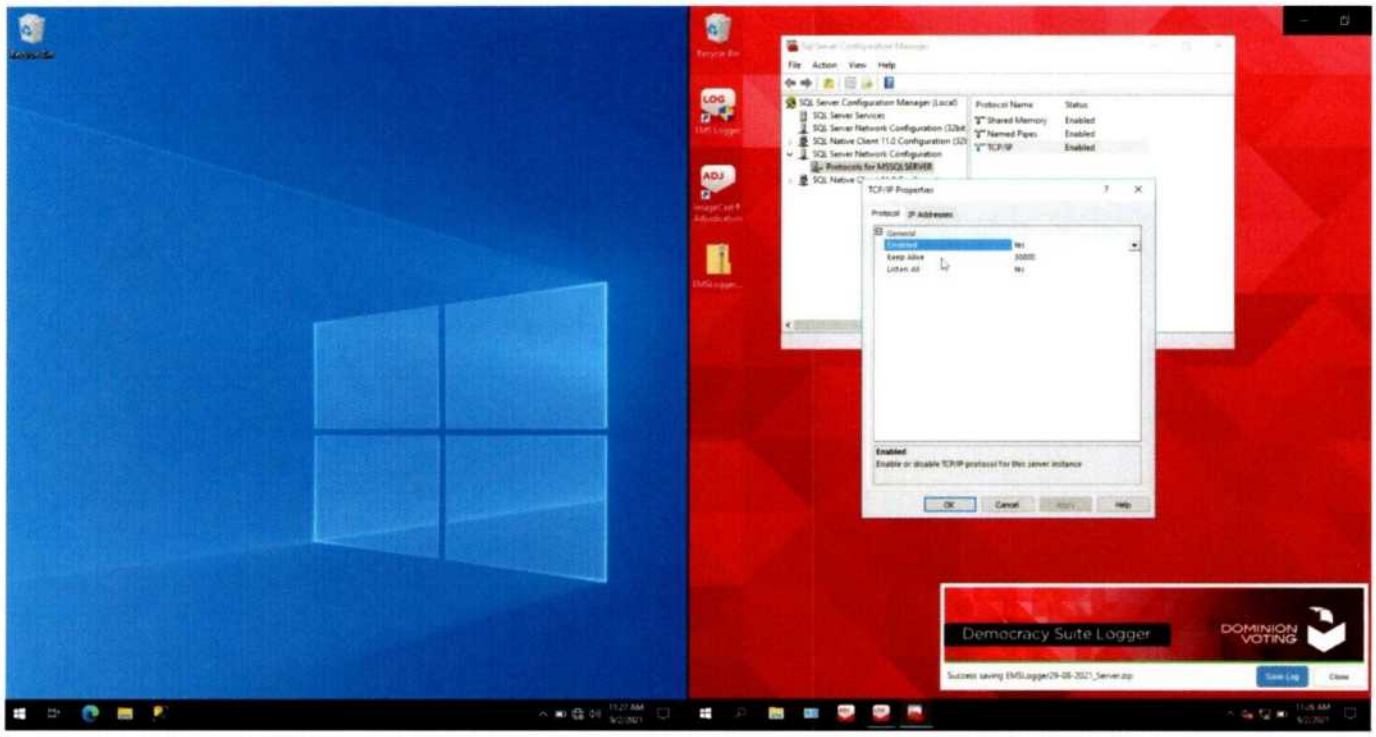


Figure 7 - TCP/IP Properties

The TCP/IP protocol setting in Figure 7 has "Enabled" set to "Yes" on Mesa County's EMS Server, and the configuration setting above has the parameter "Listen All" set to "yes" indicating that the SQL Server will listen on every network connection. More detail for the TCP/IP protocol is in Figure 8.

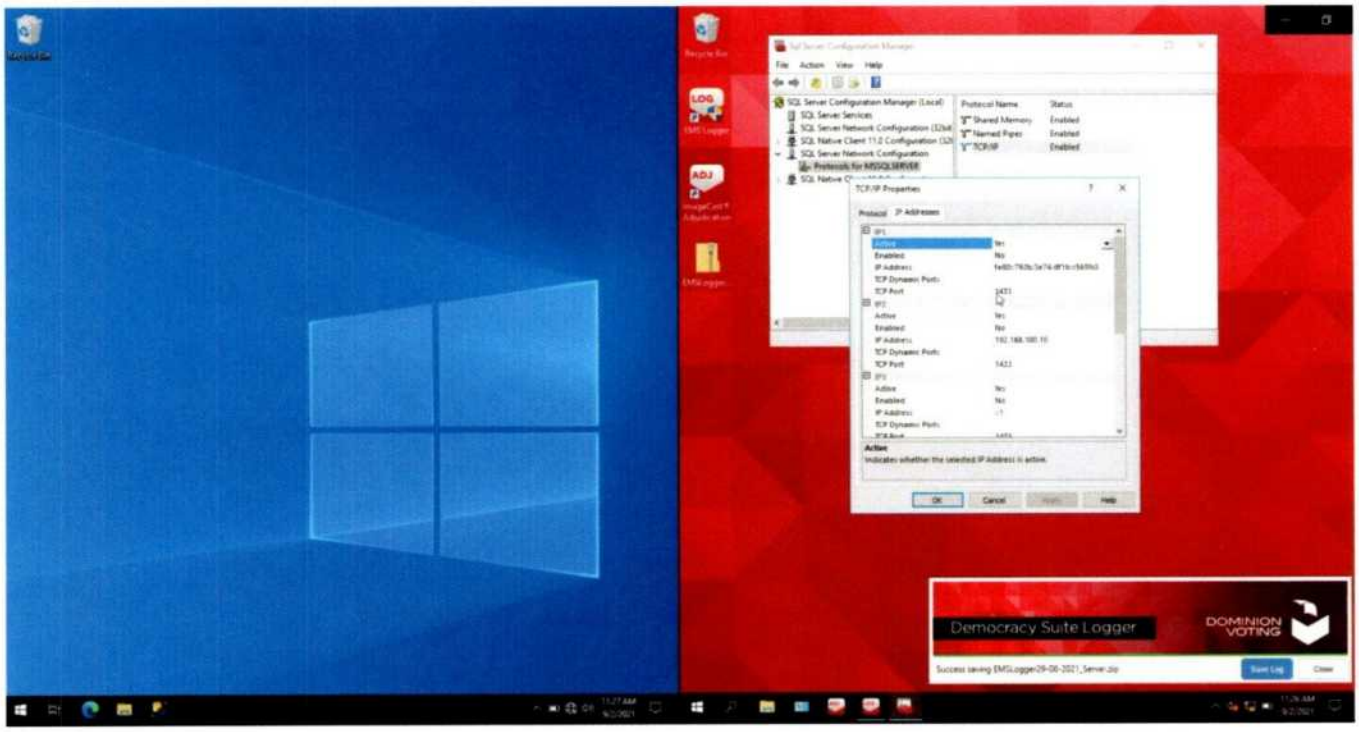


Figure 8 - TCP/IP Properties of SQL Server, attached to port 1433 the standard (default) port.

Figure 8 shows the SQL server is bound to and active on all Ethernet interfaces. This allows multiple electronic pathways to the server over multiple network connections should someone connect a cable into that jack. Also important to note is the default port number 1433 being used, instead of a more secure alternate port.

IP2 shows the IPv4 address 192.168.100.10, an IP address assigned to be used by the Mesa County EMS server. For a discussion of IP addressing fundamentals, see Appendix C. IP Addressing Fundamentals.

The Mesa County EMS server is a Dell PowerEdge T630 server, serial number 4NV1V52, and has 3 Ethernet interfaces (or Network Interface Cards (NICs)) – 2 of them assigned to the computer itself and one assigned to a separate controller (the iDRAC, Integrated Dell Remote Access Controller) which can be used to allow remote control of the computer including power-on, power-off and privileged access to the computer, via this integrated remote access controller (iDRAC). The interfaces accessed via the Server Configuration Manager (shown in these Figures) are those IP addresses assigned to the computer and do not include the interface assigned to the iDRAC.

A conclusive determination that these IP addresses had a connection to another network, even the Internet, is not possible without examining the physical system, as well every other device connected to the network. Most network firewall/router devices use translation (network address translation, NAT, or port address translation, PAT) and most computers/devices with multiple network interfaces (Wi-Fi, and wired Ethernet, for example) can be compromised to implement an Island-Hopping attack (using malicious software that provides translation, even though standards may prohibit it).

Absent a full forensic examination of all network and computing devices, it can be challenging to factually conclude that connection to the global Internet was, or was not, present and in operation. Given that

network systems are designed to support Internet connectivity, other evidence (including the alteration, addition or exclusion of votes, or data in log files, for example – See Report #1) must be considered, may be the only artifacts that enable detection or conclusive determination, and may indicate a probability that such a connection may have been in use.

I was told that when this exact copy (forensic image) of the Mesa County EMS server was taken, the Mesa County EMS server was connected to a (wired) computer network via its Ethernet interface. Configuration data forensically extracted from the EMS server, including some log remnants and registry configuration data validate this information about the connection to a network.

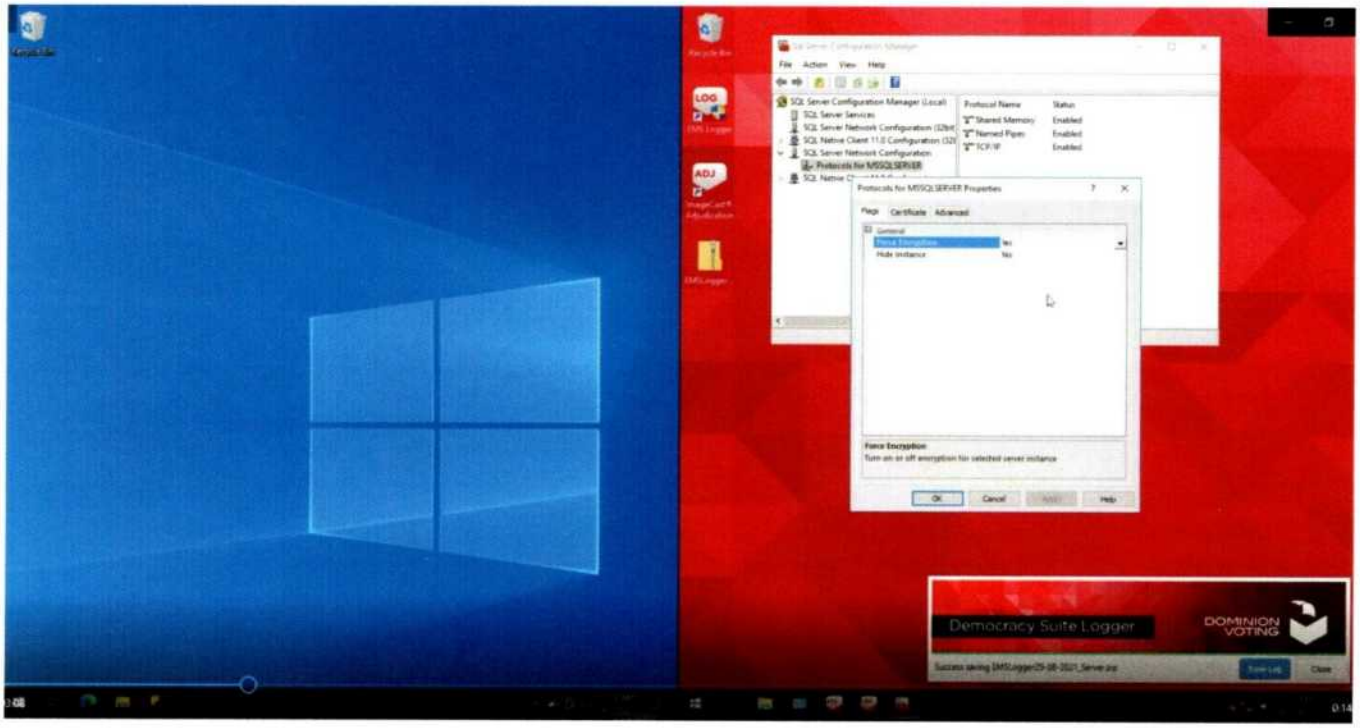


Figure 9 - SQL Server Properties

The SQL Server service is configured to force network communication to be encrypted. This is an expected configuration; however, it is crippled by what was found next.

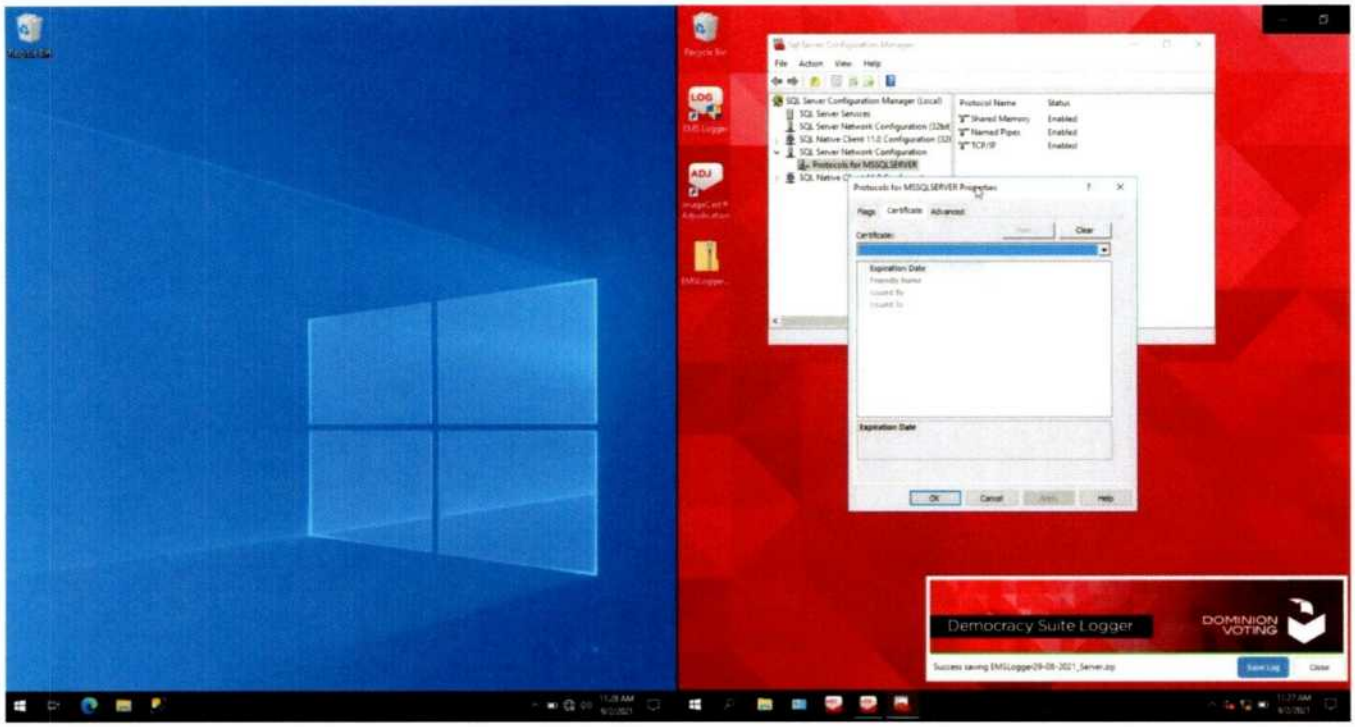


Figure 10 - Encryption is enabled but No Encryption Certificate is configured

No encryption certificate is configured, which causes the server to use a 'self-signed' certificate that is extremely vulnerable to a common man-in-the-middle attack. This means that the communication to and from the voting database itself could be intercepted, viewed, and changed, without detection.

A man-in-the-middle attack is explained in Appendix H.

The SQL Server Documentation directly provided by Microsoft clearly states "Self-signed certificates do not guarantee security. The encrypted handshake is based on NT LAN Manager (NTLM). It is highly recommended that you provision a verifiable certificate on SQL Server for secure connectivity. Transport Security Layer (TLS) can be made secure only with certificate validation." (https://docs.microsoft.com/en-us/sql/relational-databases/native-client/features/using-encryption-without-validation?view=sql-server-2016)

EXAMINATION OBJECTIVE 1:

Determine whether calculated vote totals can be altered by anyone with physical access to the logged-in EMS server.

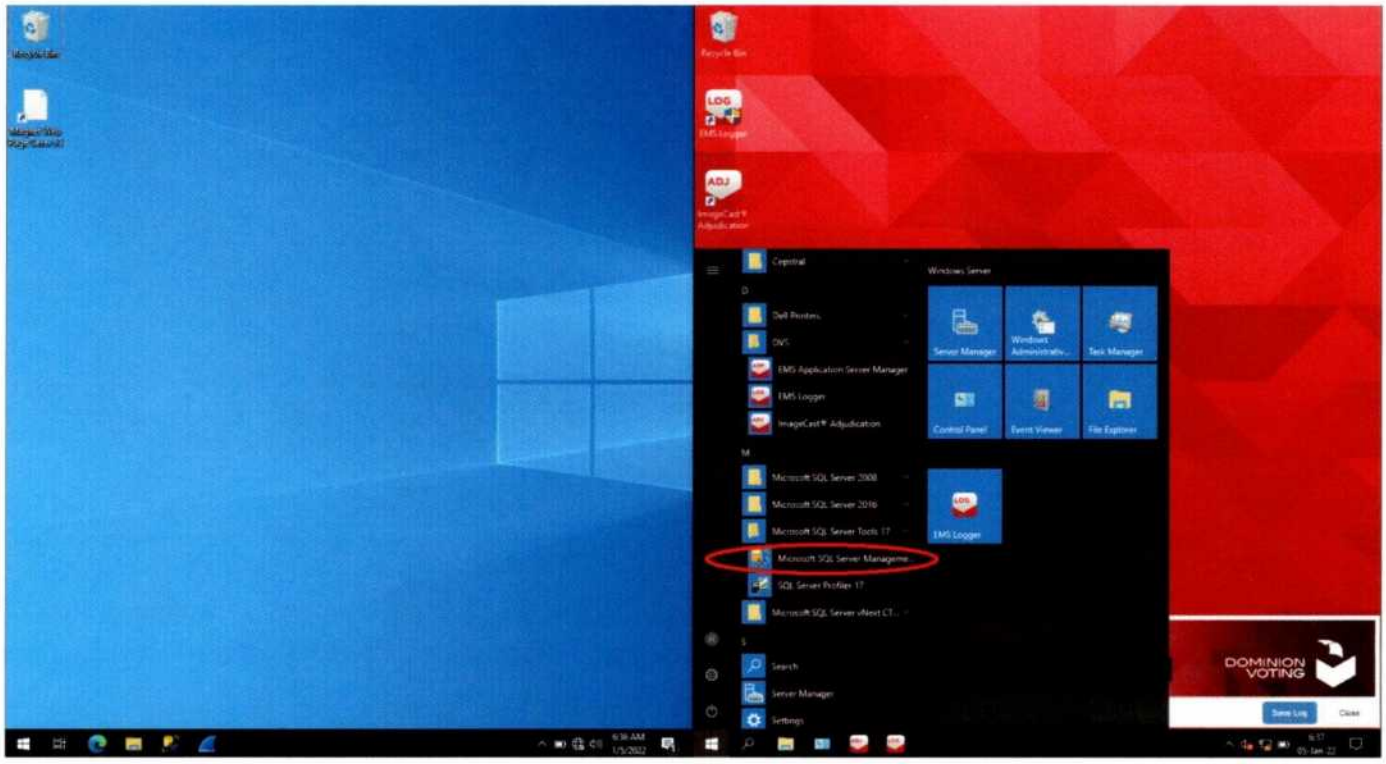


Figure 11 - SQL Server Management Studio (SSMS) software showing in the EMS server Start Menu

Microsoft SQL Server Management Studio (SSMS) allows direct back-end access to and manipulation of SQL Server databases. Figure 12 shows this software is found already installed on the EMS Server.

The VSS explicitly prohibits voting systems from allowing any users to change calculated vote totals, or an individual vote, or to compromise ballot security; the VSS also mandates the retention of all audit trails for 22 months specifically to enable detection of civil rights violations or intentional manipulation and fraud, and to support litigation and prosecution. SSMS enables that prohibited ability, as demonstrated in this test.

The Mesa County EMS was protected by only a (Windows authentication) password, as this test demonstrates. The use of a password alone is not secure; this fact is taught routinely in training for the board certification "Certified Internet Systems Security Professional" (CISSP), emphasizing the principle of "Defense in Depth," e.g., multiple layers of security.

Passwords are compromised often.[67] As early as 1985, the US Government published, in its "rainbow series" of security publications from the DoD, the "Green book[68]" guide to password management. While the password management recommendations in the guide are considered obsolete today, its appendices explain the mathematical calculation for the probability that a password can be guessed based on the complexity of the password, how often the password is changed, and the speed with which a computer can execute those guesses. Today's computer processor execution speed (CPU clock rate) is 5,000 times faster than computers were in 1985. Today's gaming home computers are 5 times faster than the fastest computer in the world was in 1985,[69] and systems used for crypto-mining may be as much as 100 times faster than that fastest 1985 computer.

Password insecurity alone presents an extreme and unacceptable risk.

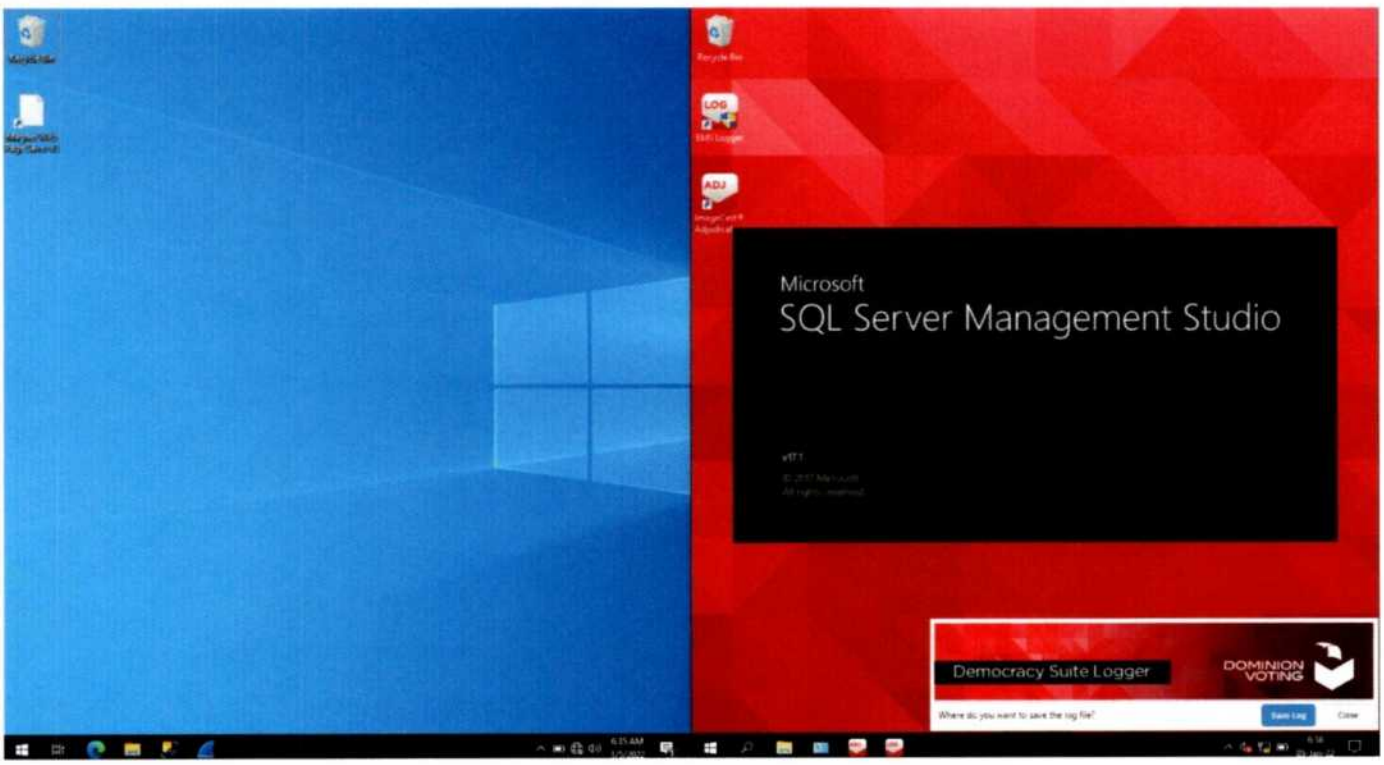


Figure 12 - SSMS is installed and starting on the EMS server system.

The SSMS starts up without any problem or warning when a user clicks on it.

[67] Accounts in public media support this fact. These are only several of many such references: https://www.westernjournal.com/az-audit-exclusive-election-systems-password-hasnt-changed-2-years-shared-time/ and https://www.csoonline.com/article/3266607/1-4b-stolen-passwords-are-free-for-the-taking-what-we-know-now.html

[68] https://csrc.nist.gov/CSRC/media/Publications/white-paper/1985/12/26/dod-rainbow-series/final/documents/std002.txt

[69] A Cray X/MP supercomputer operated at a clock speed of 1 GHz, or 1 billion clock cycles per second in 1985, while the first home PC clock speed was typically 1MHz.

Not only can SSMS be used on a separate computer, not part of the DVS system, to directly access the back-end server databases, it can be used directly by any person with physical access to the logged in server itself (screen, keyboard, and mouse), such as rogue election staff, cleaning staff, etc.

In addition to bad-actors from outside the election staff, any individual election staff worker that has access to a logged-in EMS server also is allowed the ability to go directly into the back-end of the database and add votes, change votes, delete votes, swap votes, and countless other alterations, bypassing all DVS application software. Even an honest individual could accidentally allow data to be changed without their knowledge in a matter of seconds by innocently attaching a USB flash drive with hidden programming/malware on it.

Anyone with unrestricted physical access and knowledge of the userID can make similar changes without even a password, if the standard user account is left logged-in. Someone with advanced security knowledge can access the system without a password, as I was easily able to do.

In this test the Microsoft SQL Server Management Studio is used to demonstrate unauthorized access to the election databases. However, the use of Microsoft SSMS is not even required – a popular piece of software manufactured by SQL Pro (e.g., non-Microsoft software) is shown in the third test in this report, to provide the same access from the more limited computing power of a mobile phone.

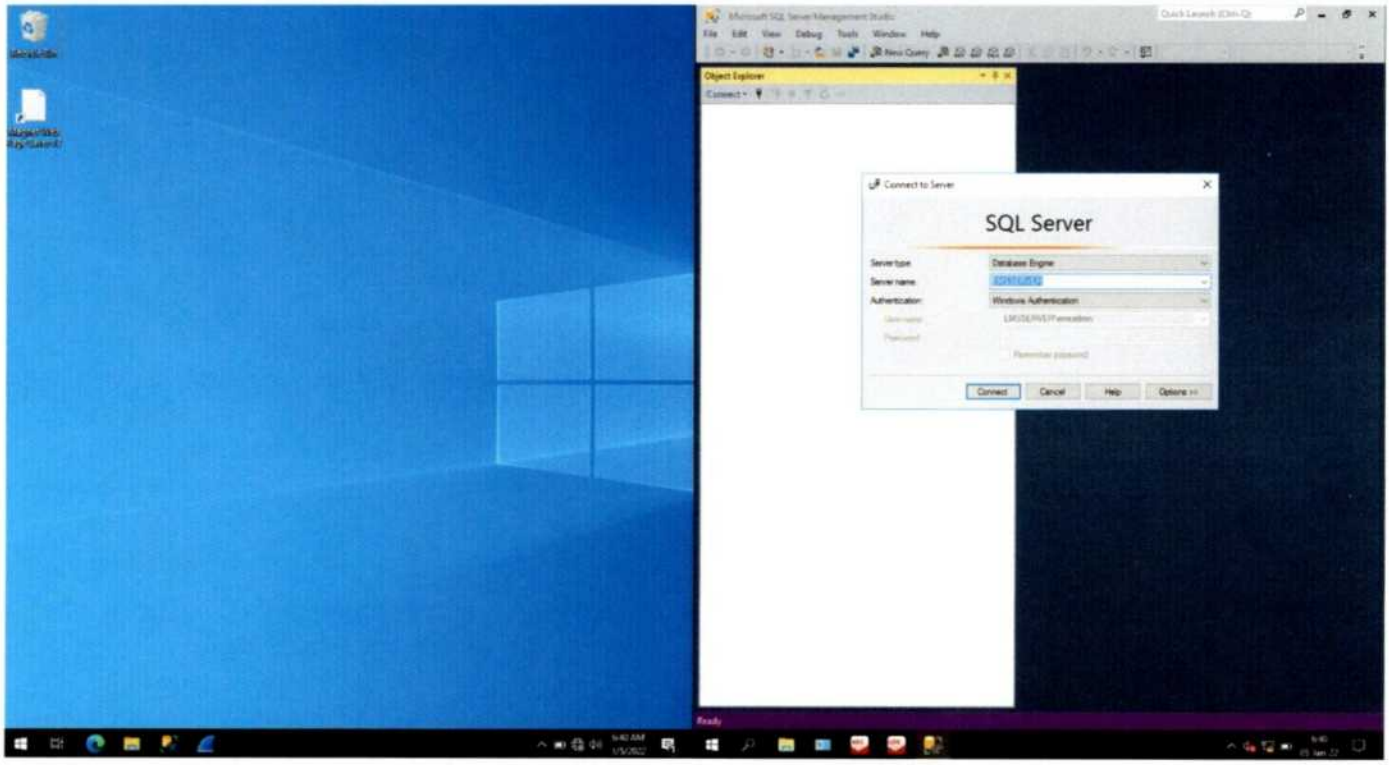


Figure 13 - Logging in to the SQL Server using SQL Server Management Studio

When SQL Server Management Studio (SSMS) first starts, connection entries are already pre-filled-out. The user doesn't isn't required to type a username or password, and needs only to click the 'Connect' button to get into the back-end databases.

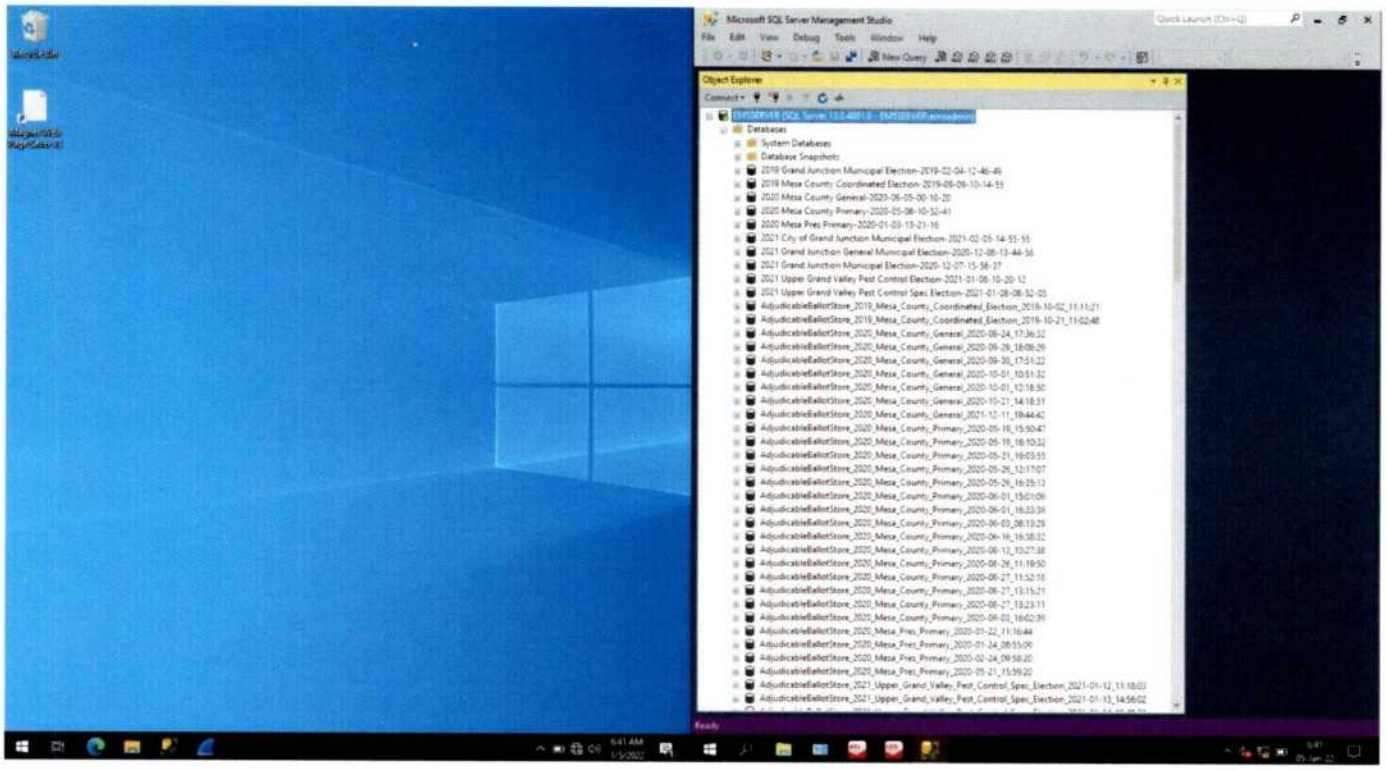


Figure 14 - SSMS enables direct access to the internal databases to anyone logged in to the EMS server

After clicking 'Connect,' and then the '+' sign next to 'Databases' all the internal databases are shown to be accessible. It took only four clicks of the mouse to get here into the back-end of the voting databases.

One of the many election databases that are shown is from the 2020 US General Election. The US Presidential Primary of 2020, among many others, can also be seen.

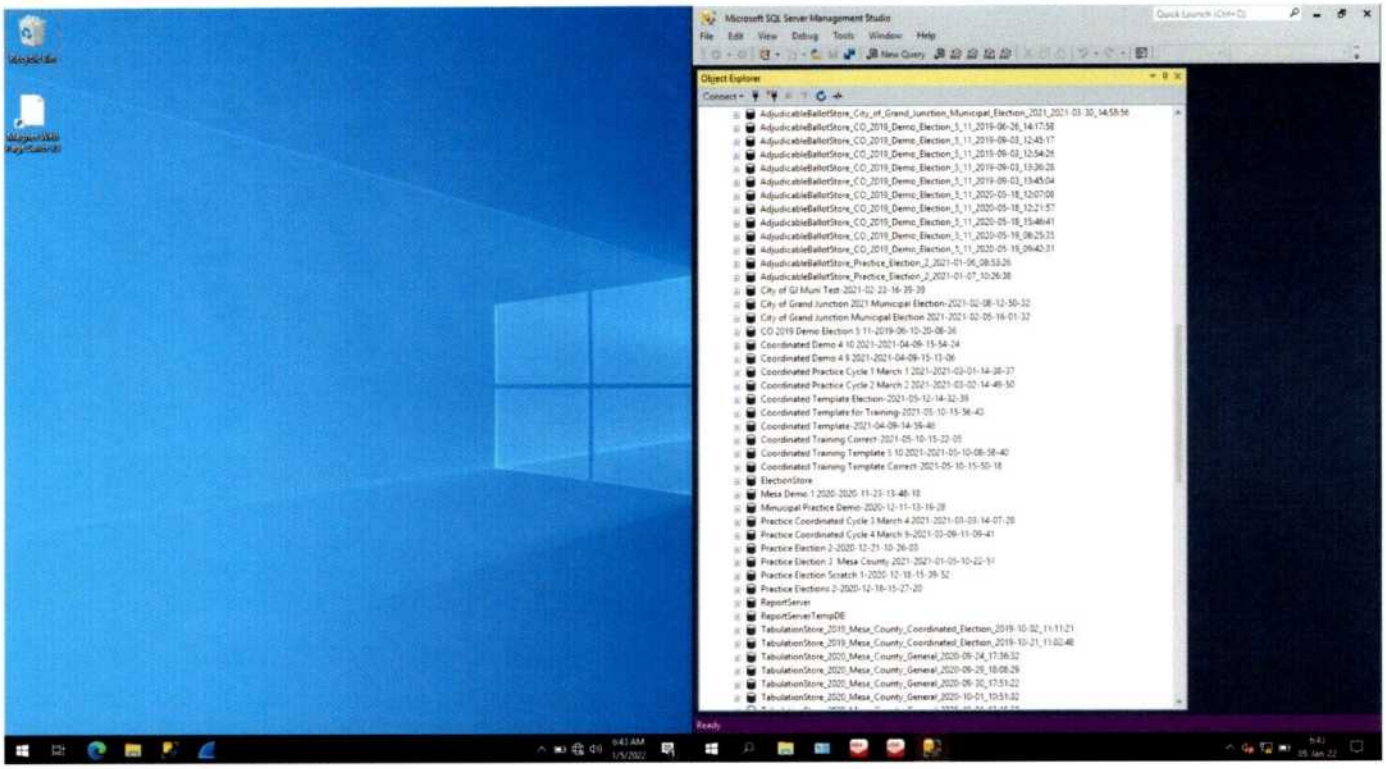


Figure 15 - Databases from many prior elections are fully accessible

Here can be seen accessible many elections from the City of Grand Junction, Mesa County, as well as adjudication and tabulation databases from many of these elections.

The presence of databases from previous elections on the EMS server, provide a rich library of information that can be used to understand and identify potential vulnerabilities in the EMS. While these records are required to be retained, they should be maintained off- system, securely archived, inaccessible to the EMS or any user.

The presence of prior election databases on the EMS server also offers an extensive and convenient repository for copy and paste modifications of election data, not only for the 2020 election but for any prior listed election as well.

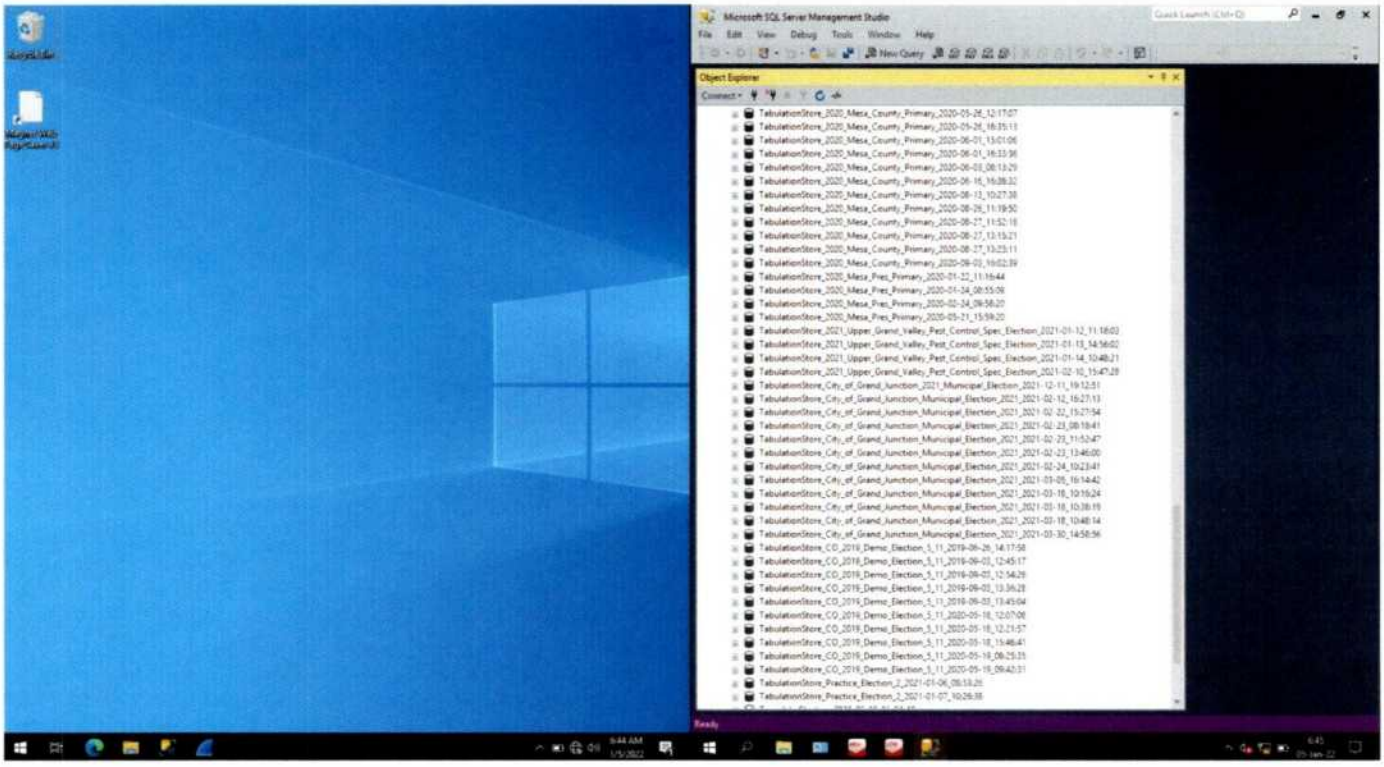


Figure 16 - Additional databases used in previous elections

Many TabulationStore databases are shown here, including even a TabulationStore for the Upper Grand Valley Pest Control Special Election.

Figure 16 is a continuation of the list in Figure 15, demonstrating that far more than one screen of databases are accessible.

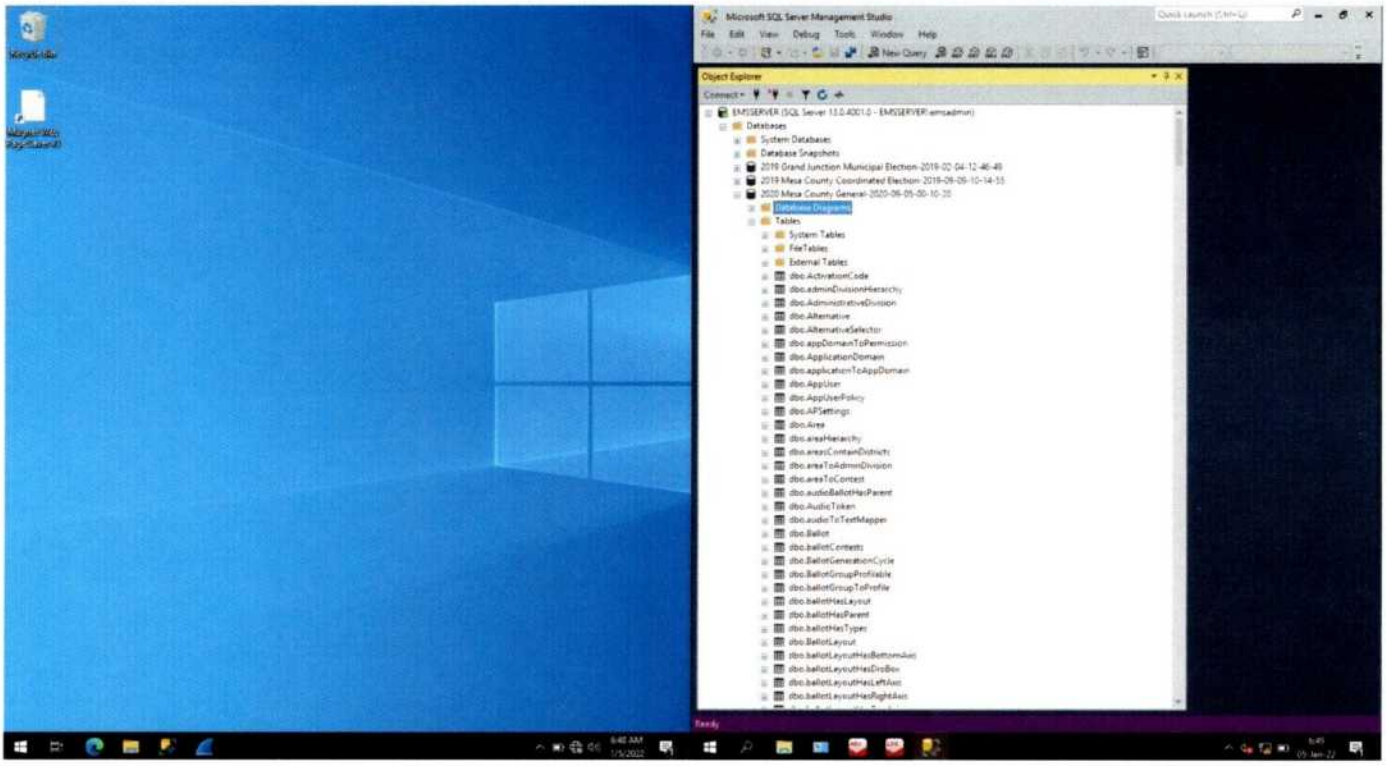


Figure 17 - Internal database tables, including ones with counted votes are accessible

The '+' sign next to the 2020 Mesa County General database was selected, followed by the '+' sign next to 'Tables.' A list of all internal database tables for the 2020 Mesa County General database is now shown. Nothing has stopped me from accessing this. Not a single warning has shown on screen.

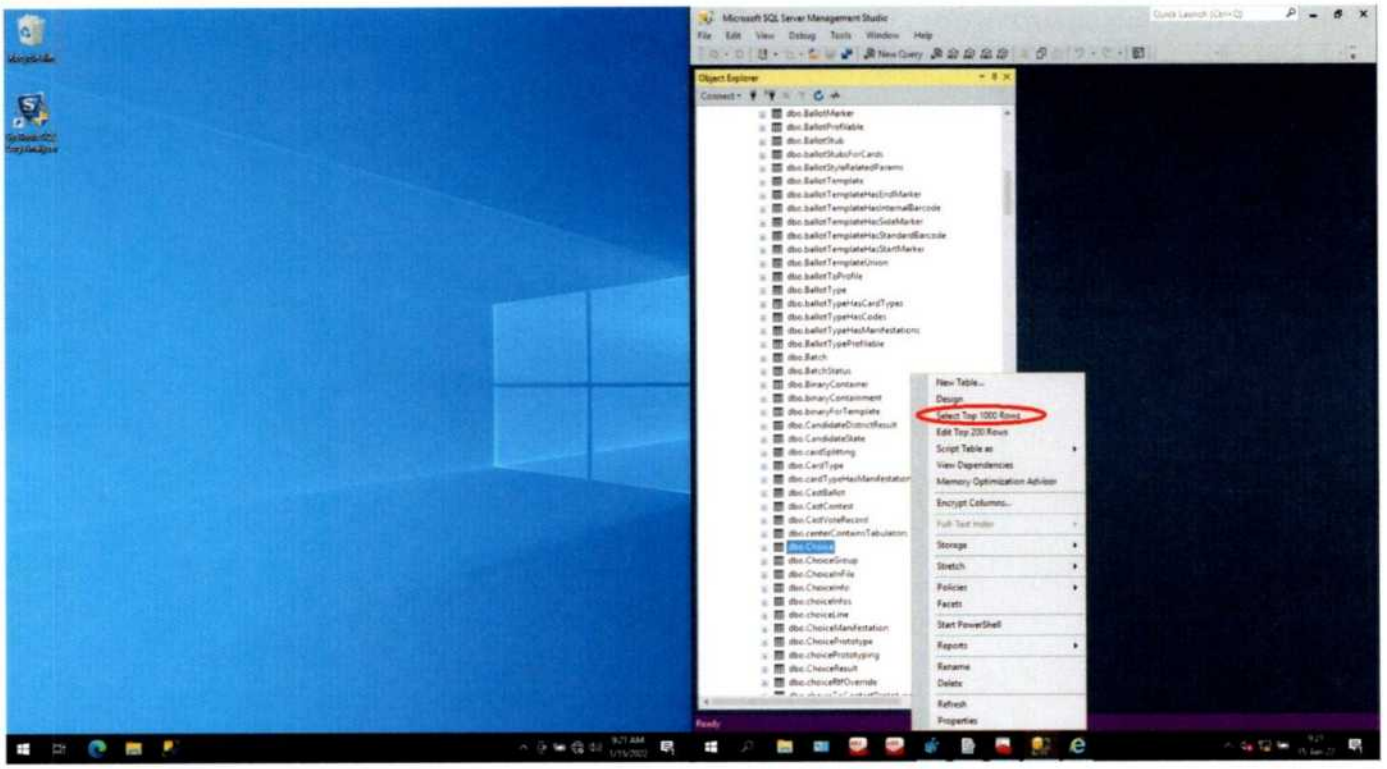


Figure 18 - Menu Option to Select the Top 1000 rows

As an example, one of the tables, 'dbo.Choice,' was selected by scrolling down and right-clicking, then choosing 'Select Top 1000 Rows' by clicking on that option. This instructs the database server to show me the top 1000 rows in the database table.

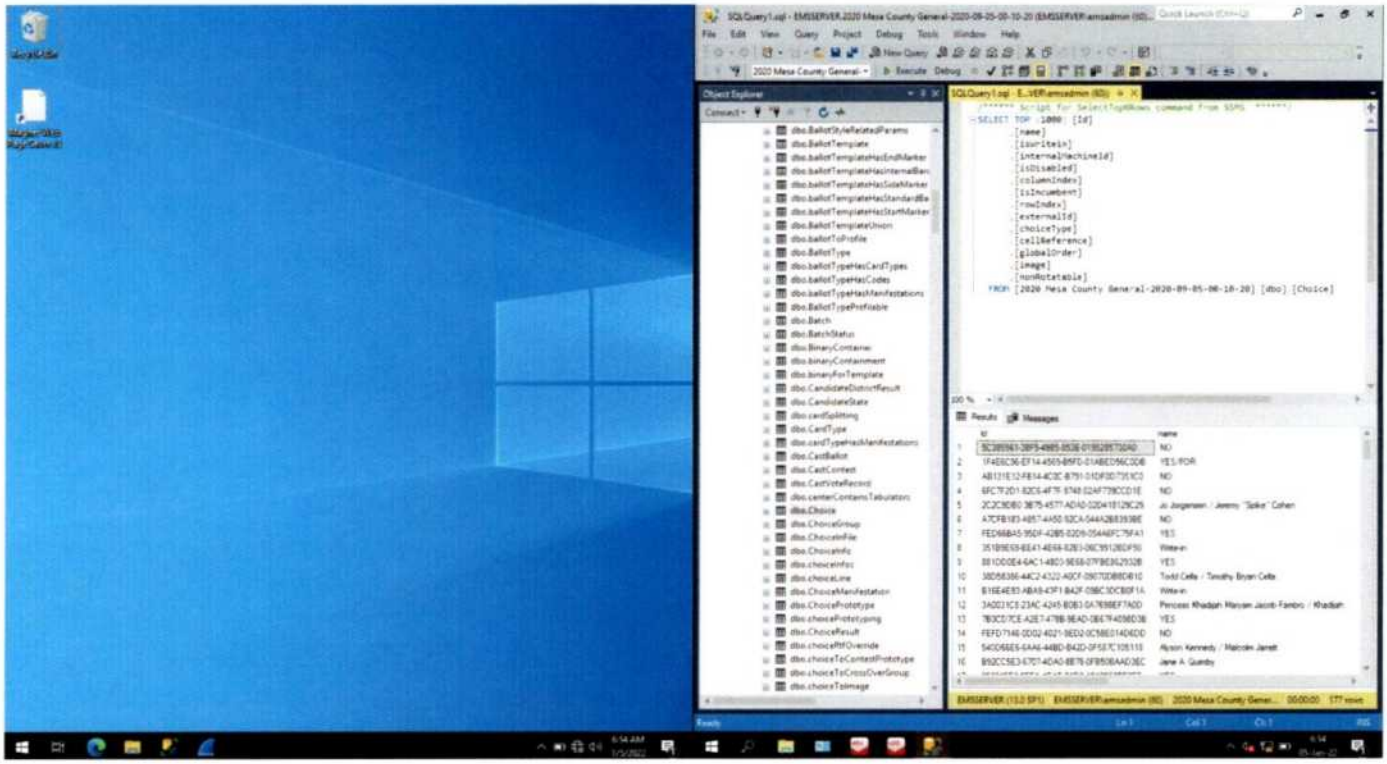


Figure 19 - Accessing the Ballot Choice database table

I was able to easily open the Ballot Choice database table. The computer retrieved all 177 rows of data from this table in the database. This corresponds to 177 different ballot choices in the election. I have still not been blocked, nor has the system provided any warning that anyone is directly accessing the voting database.

Each election "contest" is defined, together with candidates and the rules for voting, e.g., "pick one, pick two, pick three, etc.," depending on the specific item, for example, commissioners of a town, and the number of seats open in this specific election.

On the right side of the screen in the upper right pane is displayed the SQL Query (SQL program script) that is automatically filled-out by SSMS. The user merely just needs to know how to click the mouse button. The automated query shown is used to retrieve data (the top 1000 rows), and the data columns listed that will be retrieved are also shown. On the bottom right pane the response from the request is shown. The first two columns display on screen ('Id' and 'name') but the scroll bar allows one to scroll to the right to see the remaining 12 columns.

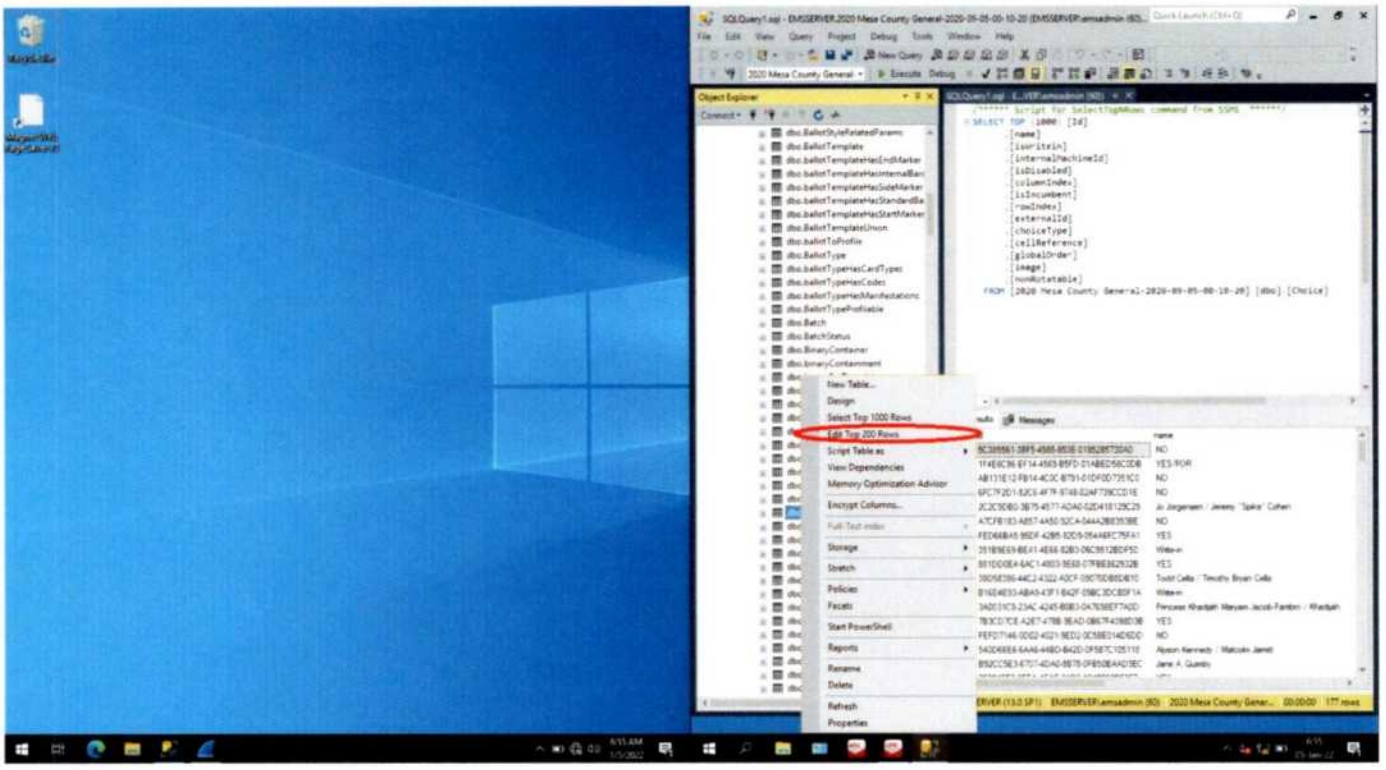


Figure 20 - Test to determine if the Ballot Choice Table can be edited to easily flip the votes

I now right-click the table again and select the menu option to Edit the Top 200 rows of the database to determine if it will also allow me to directly alter the data.

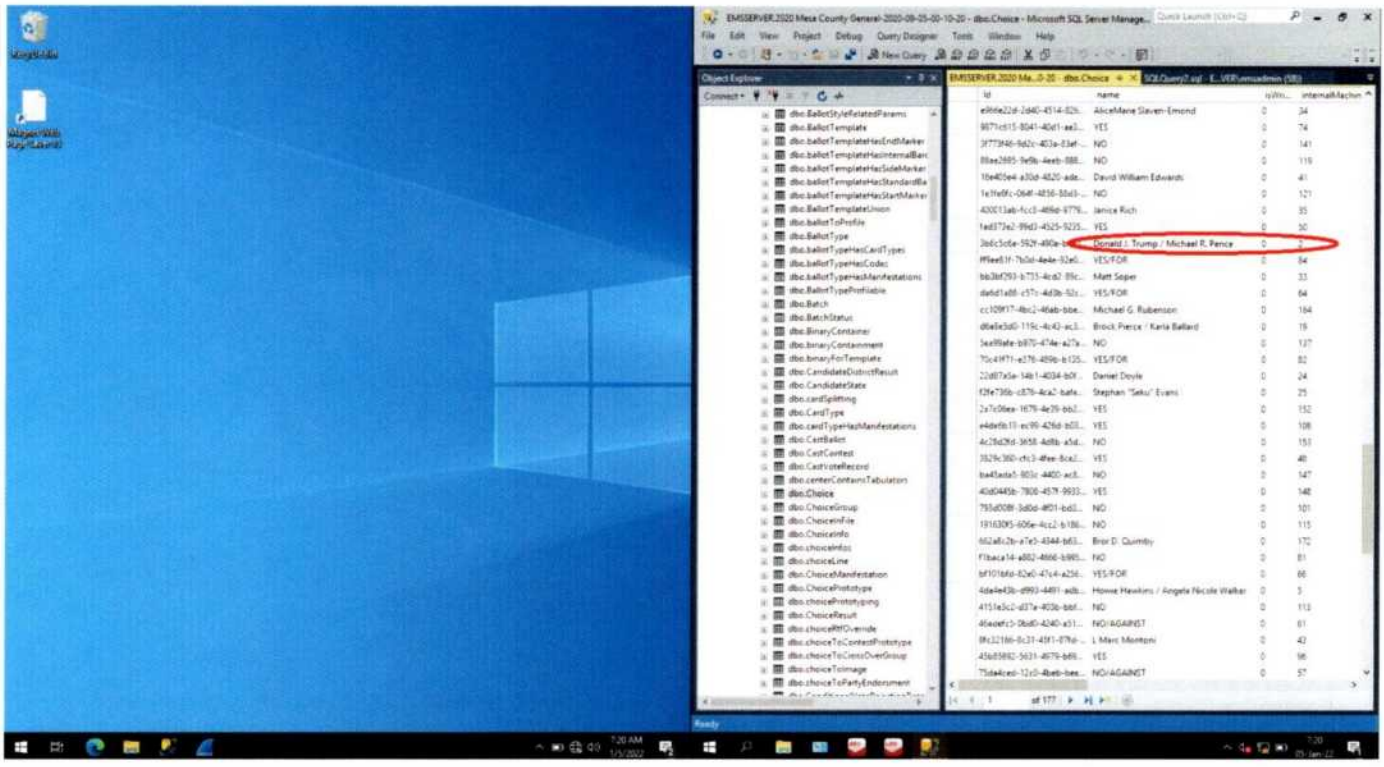


Figure 21 - Candidate settings for Trump

The computer responds to the request and shows all 177 rows of this Choice table in a spreadsheet-like display. Note here that the Choice 'Donald J. Trump / Michael R. Pence' has an internalMachineId of '2'.

Note the first four columns are:

- Id – A unique identifier to identify the particular choice.
- Name – The 'title' of the choice on the ballot.
- isWritein – Possibly used to signify if a particular choice is a write-in field.
- internalMachineId – Another unique identifier to identify a particular choice used to produce reports.

The internalMachineId parameter is an indirect reference to the counted vote for candidates. Because the reference is indirect (i.e., a number rather than a key index that is common to the candidate's identity throughout the database), the reference can be easily changed, flipping the vote, and is extraordinarily difficult to identify. In database design, this is an example of bad design practice that breaks the "referential integrity" of the database and enables the potentially malicious action demonstrated here.

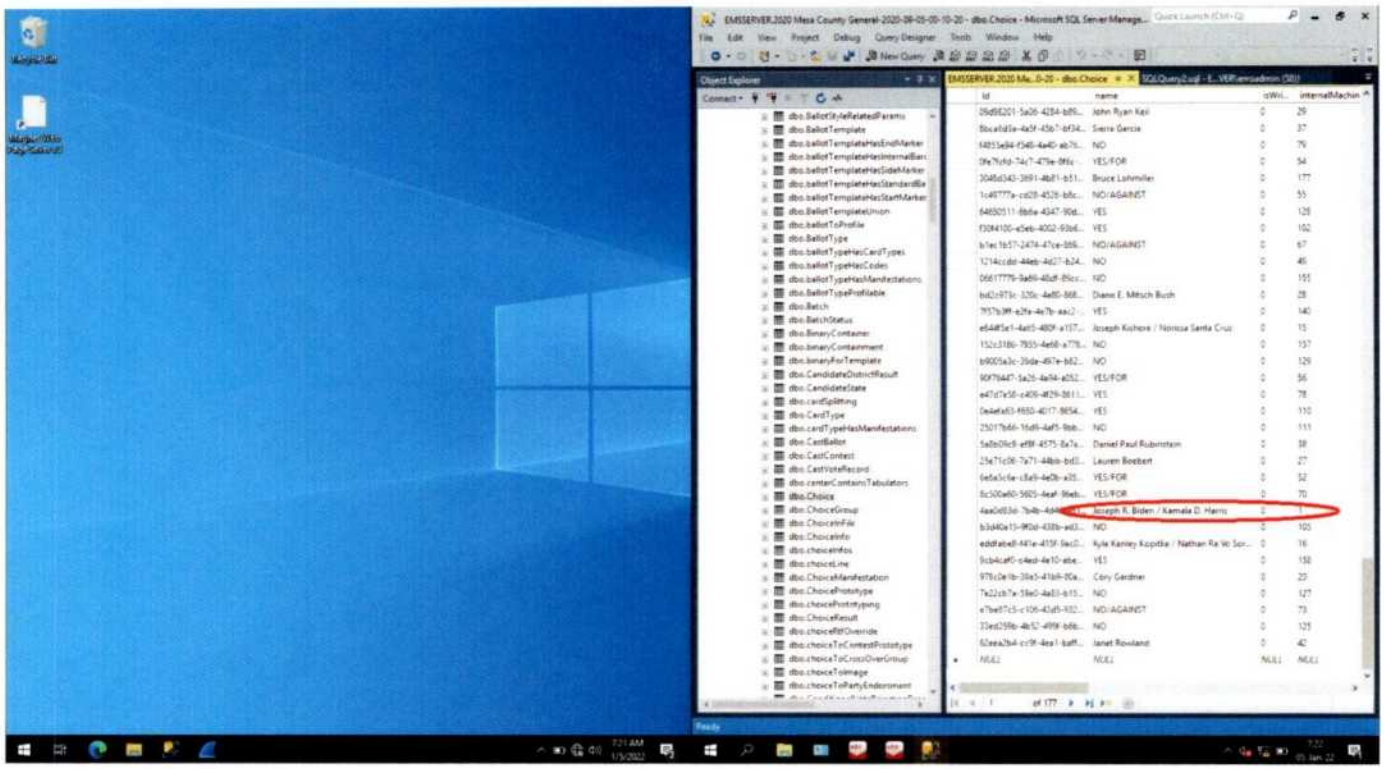


Figure 22 - Candidate settings for Biden

The 'Joseph R. Biden / Kamala D. Harris' choice has an internalMachineId of '1.'

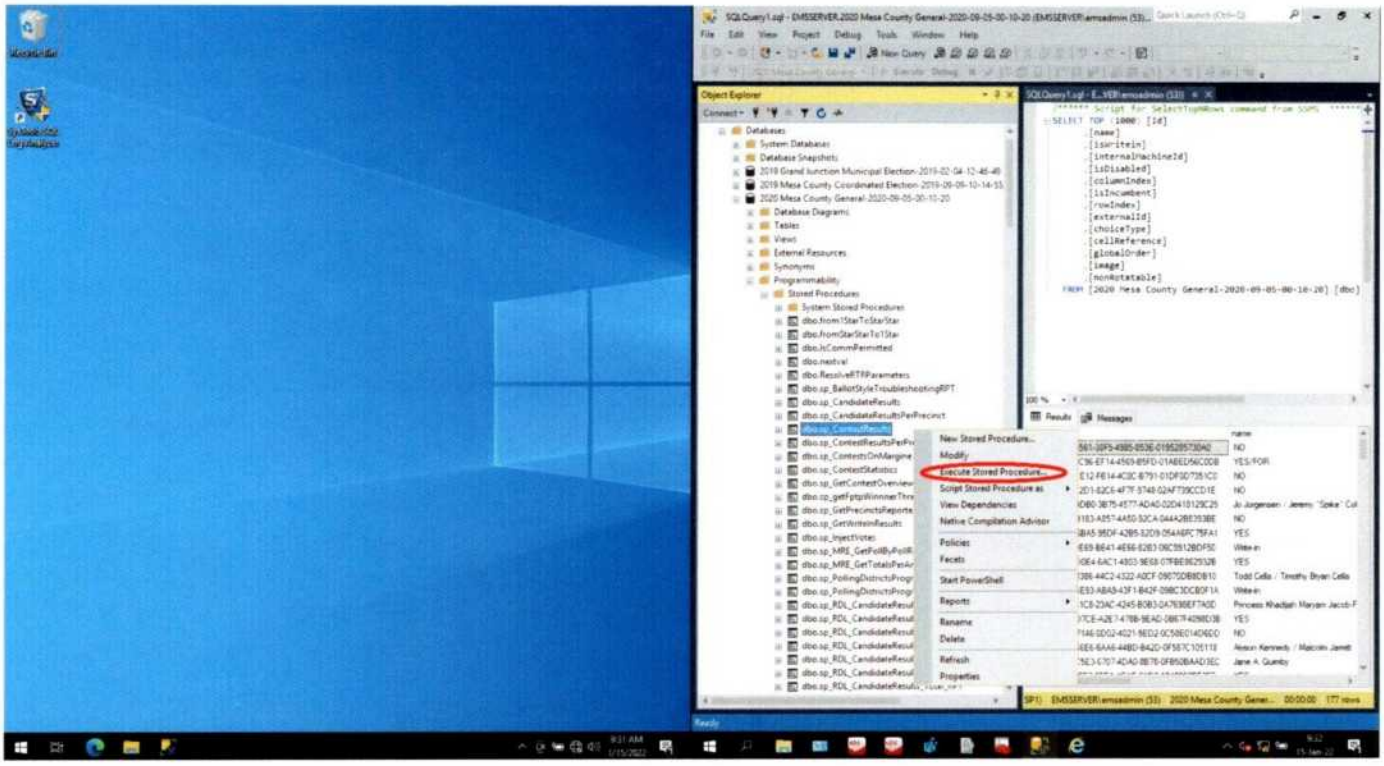


Figure 23 - Pulling up the results report prior to attempting the alteration

Prior to attempting to make a direct change that would alter the results of the election, the Stored Procedure 'dbo.sp_ContestResults' is executed to query the current contest results. These steps involve only a few clicks of the mouse.

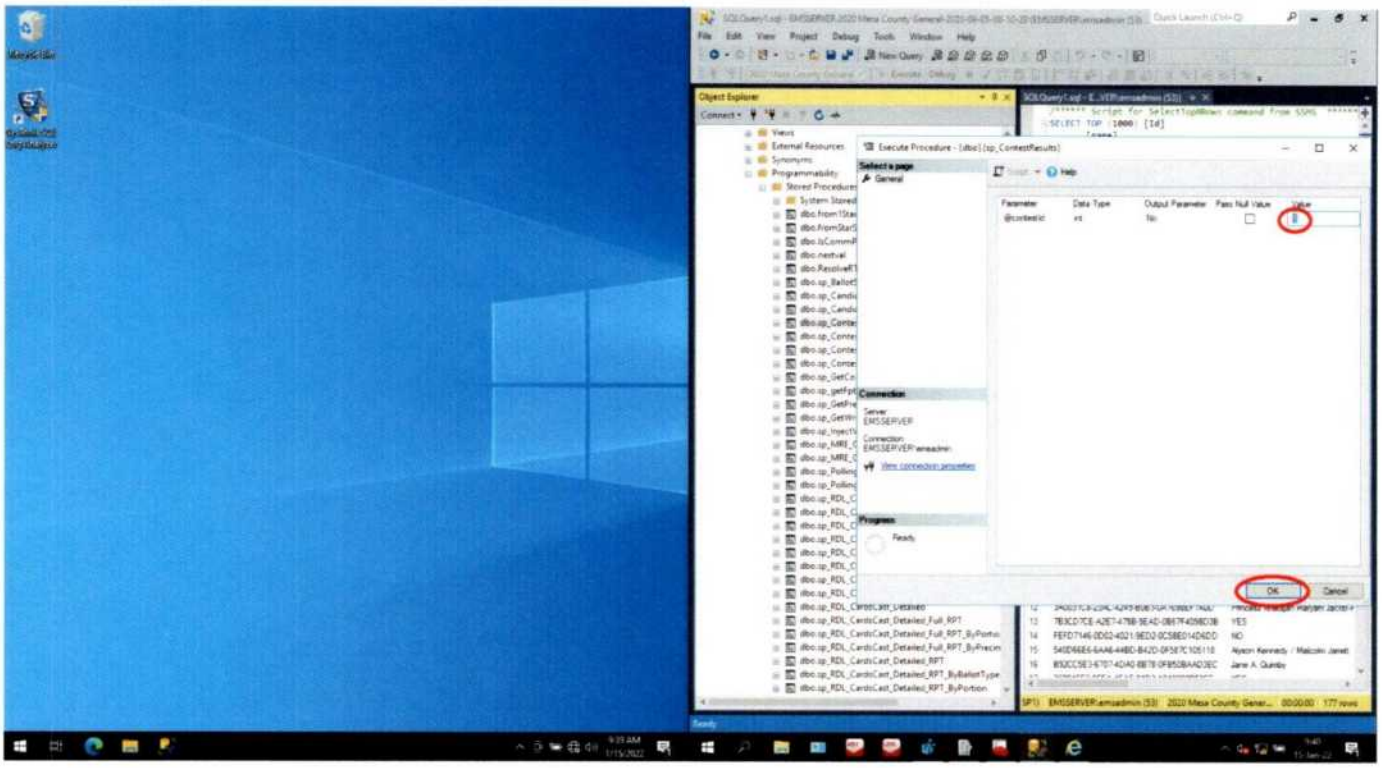


Figure 24 - Run Stored Procedure to pull up a report of Presidential Electors

The computer then prompts for which ContestId to query. A '1' to signify the Presidential Electors is entered, then 'OK' is clicked.

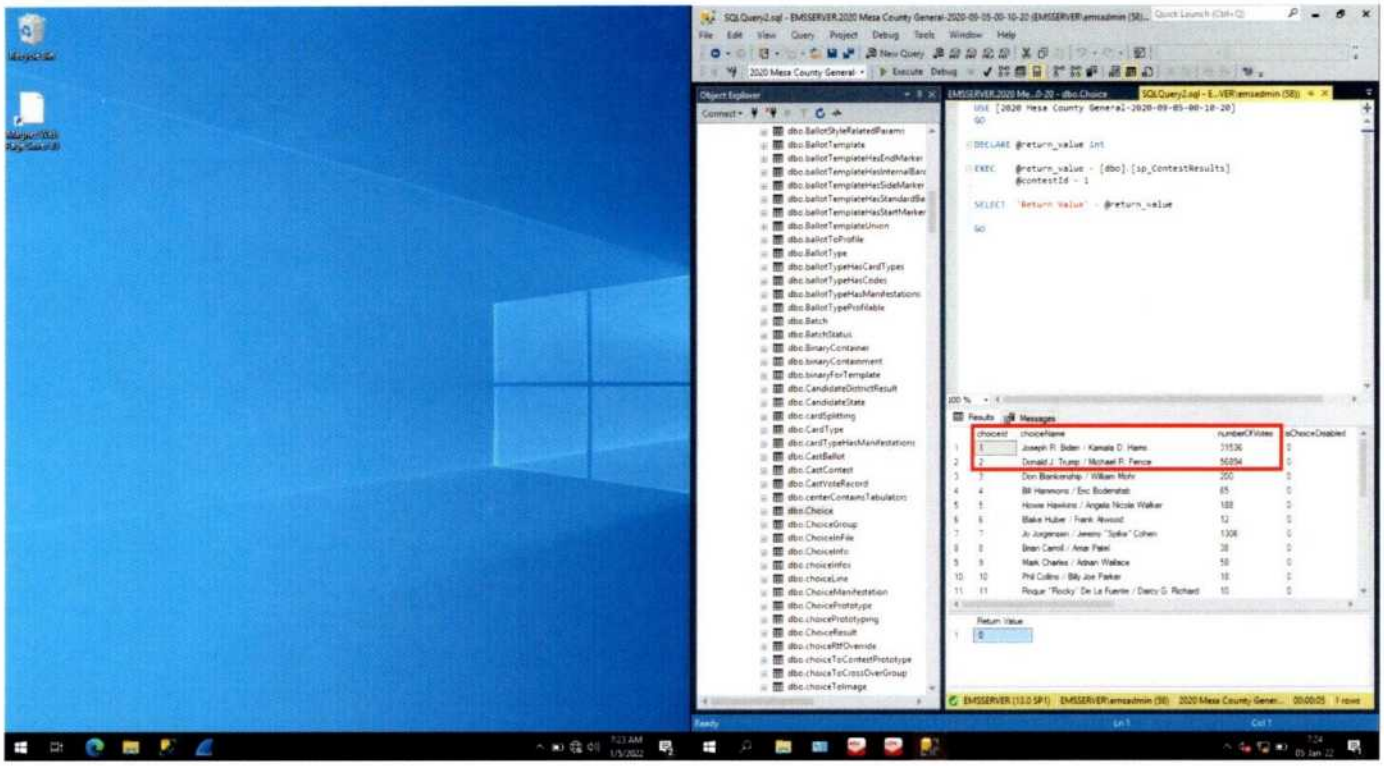


Figure 25 - Retrieved Vote Totals

This report shows the total number of votes for the Presidential contest:

'Joseph R. Biden / Kamala D. Harris' as having 31,536 votes, and

'Donald J. Trump / Michael R. Pence' as having 56,894 votes.

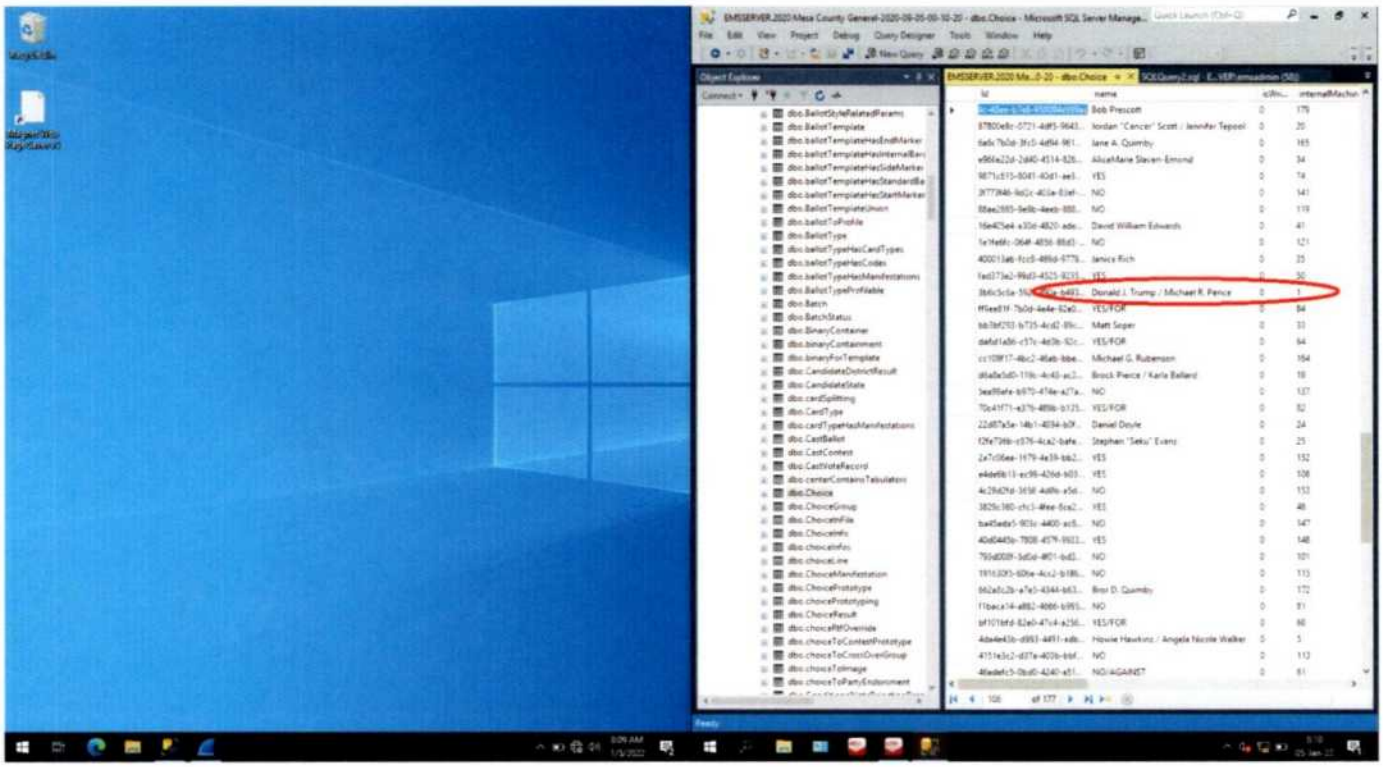


Figure 26 - Candidate number for Trump modified

Here, I change the Trump 'internalMachineId' from a '2' to a '1.' The SQL Server Management Studio allows the change without any hesitation or warning that a crucial piece of data was changed. The lack of good design and very poor referential integrity allows this.

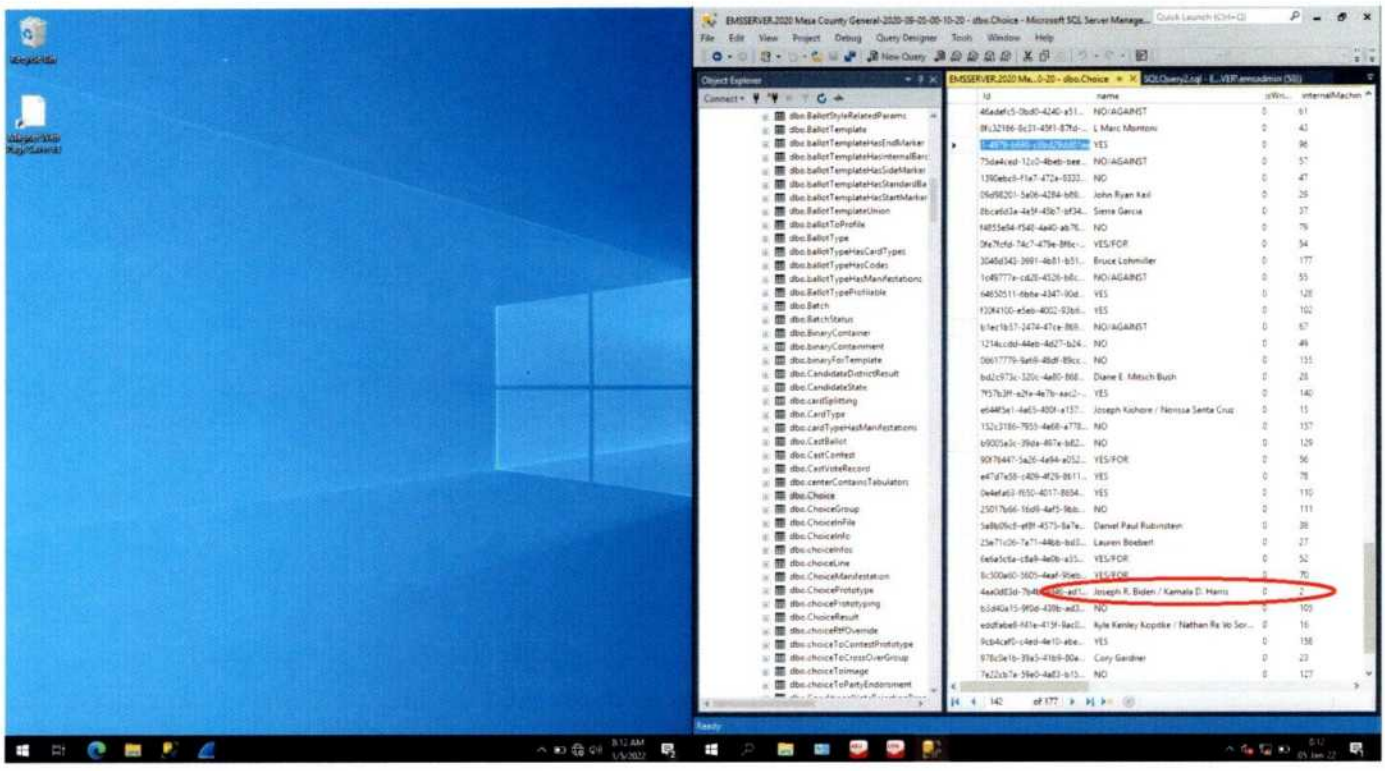


Figure 27 - Candidate number for Biden modified

Next, I change the Biden 'internalMachineId' from a '1' to a '2.' Again, there is no error message or warning given by the system.

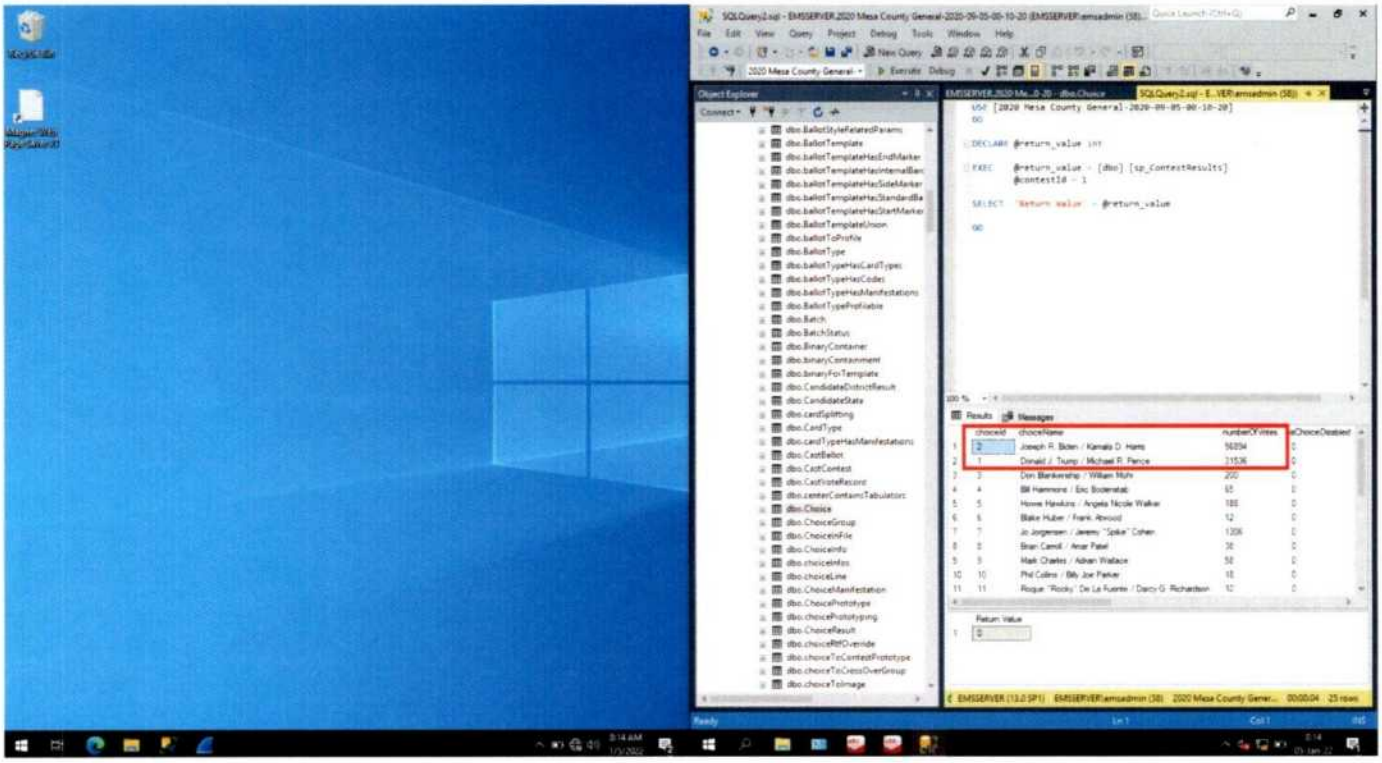


Figure 28 - Vote totals retrieved again after modification.

Making only these two small changes, which can be done in under a minute by an individual sitting in front of the voting system server, resulted in a flip of 25,358 votes. This demonstrates the ease with which someone can completely alter the results of the election on this EMS server with only a few mouse clicks and 2 keypresses on the keyboard with the software that is built-in to this voting system. This is only one in countless ways election data could be altered.

When the stored procedure is executed to retrieve the vote totals again, the vote totals for Biden now show 56,894 and the total for Trump shows 31,536.

By changing only two values in the election database in less than a minute, I have flipped 25,358 votes, completely changing the vote total results in the election database. The change was made using Microsoft SSMS software already residing on the EMS server, without needing to enter any additional password, and without a warning about the risk of changing this information.

Finding 2: The existence and use of unauthorized and uncertified Microsoft SQL Server Management Studio (found on the EMS server in Mesa Co. and in other counties around the country), allows and facilitates the bypass of Dominion Voting Systems' software to alter calculated vote totals in the election database by anyone with physical access to the logged-in EMS server.

It is important to understand how easily this was done, and therefore how quickly such a change can be made. It was not necessary to change the 88,430 votes in the database, but rather only two index values, the internalMachineId values, to completely flip the result of this county's votes.

Finding 3: It is a simple task to flip votes and therefore very easy to do quickly.

Finding 4: The insecurity of the Mesa County EMS server, in concert with unauthorized, uncertified software, allowed the alteration of the election result, flipping the vote from one candidate to another, with trivial difficulty.

Let us also distinguish the claim being made here:

It is not asserted in these findings that this 'Vote Flipping' was performed on this server during the 2020 election, but rather the design and configuration of the system permits it, and due to the extraordinary lack of security and the unauthorized, uncertified software installed on the system, the voting system itself was, and is, completely uncertifiable and wholly unsafe to use for any election.

To be explicitly clear, this demonstration is about the lack of security and the access that insecurity and unauthorized software allows, and it is explicitly not about the vote totals in any election from this server. The lack of efficient logging and the destruction of the required log files prevent any assertion to the contrary in this analysis.

Whether votes were 'flipped' using this process, or the countless other ways that could be used, requires examination of computer system logs and database logs, and other data, and will be separately addressed. In this finding, it is demonstrated that it is possible, and that the defects in the security and certification of the system are extraordinary and far beyond simple errors and omissions.

Examination Result #1

Vote totals can be altered by anyone with physical access to the logged-in EMS server.

EXAMINATION OBJECTIVE 2:

Determine whether the calculated vote totals can be altered by any person using a non-Dominion computer directly or indirectly connected to the EMS server network.

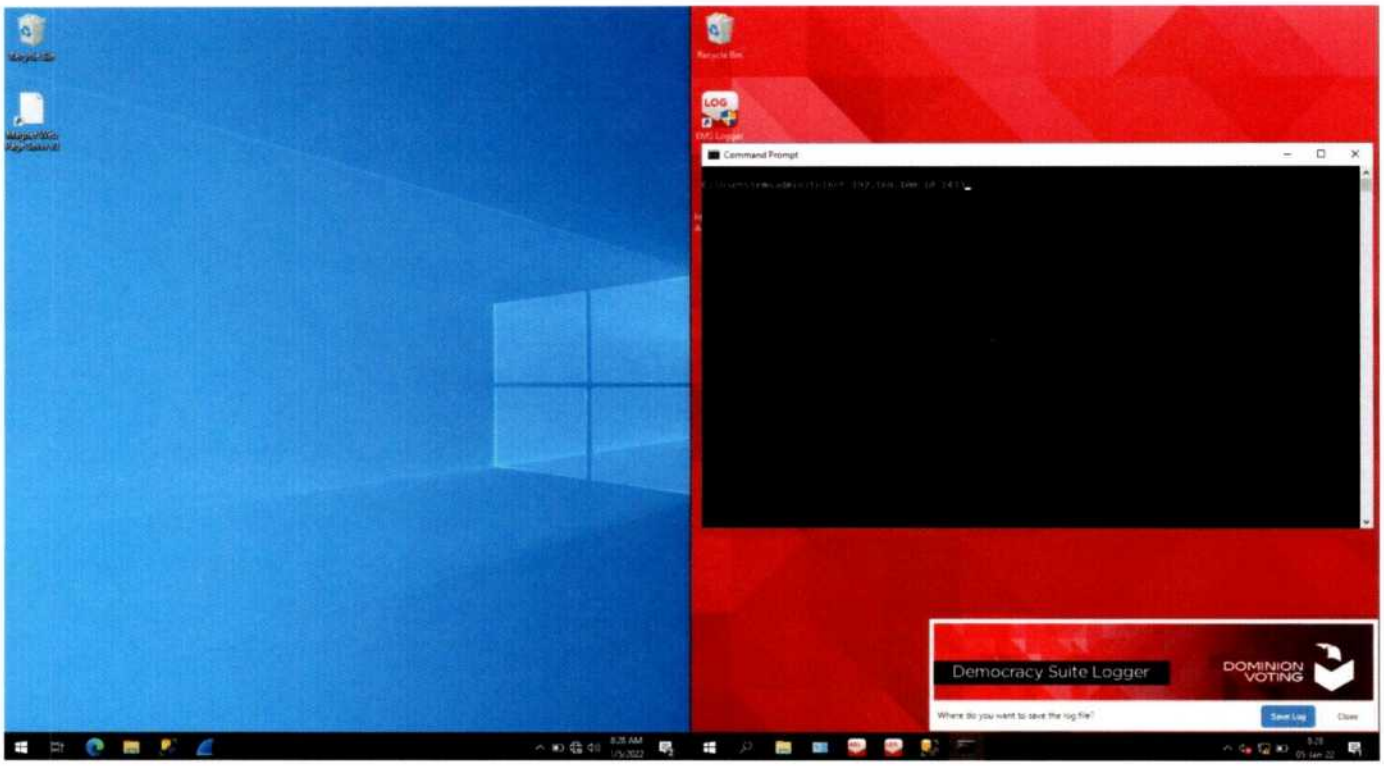


Figure 29 - Accessing port 1433 with Telnet

The telnet command is used to test to see if direct network connection to the database port is possible.

'Telnet' is a common network diagnostic tool used by IT and Cybersecurity professionals for communicating with a telnet server, and other text-based TCP services.

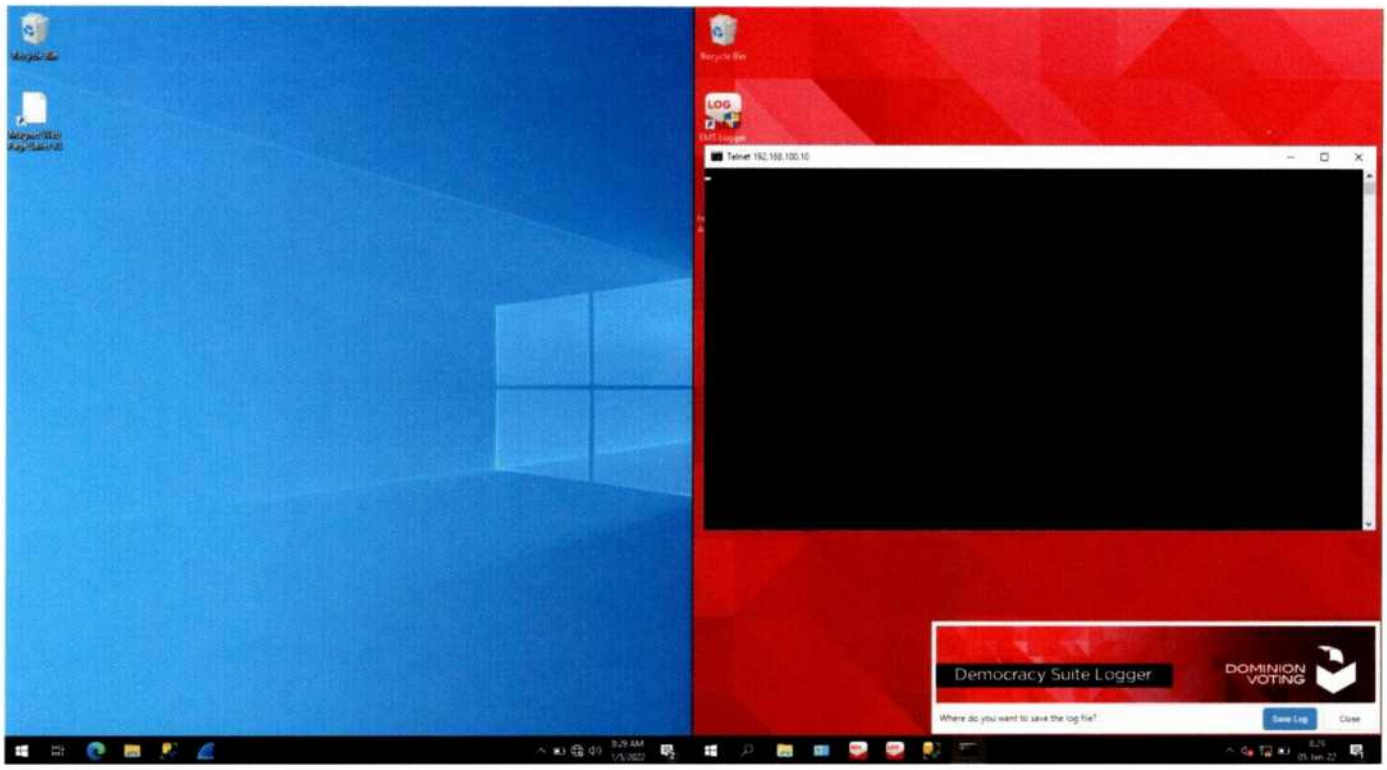


Figure 30 - The EMS server network interface appears to answer a connection to port 1433

The blank window with the cursor in the top left indicates that the connection was indeed successful, and the database service is now waiting for input.

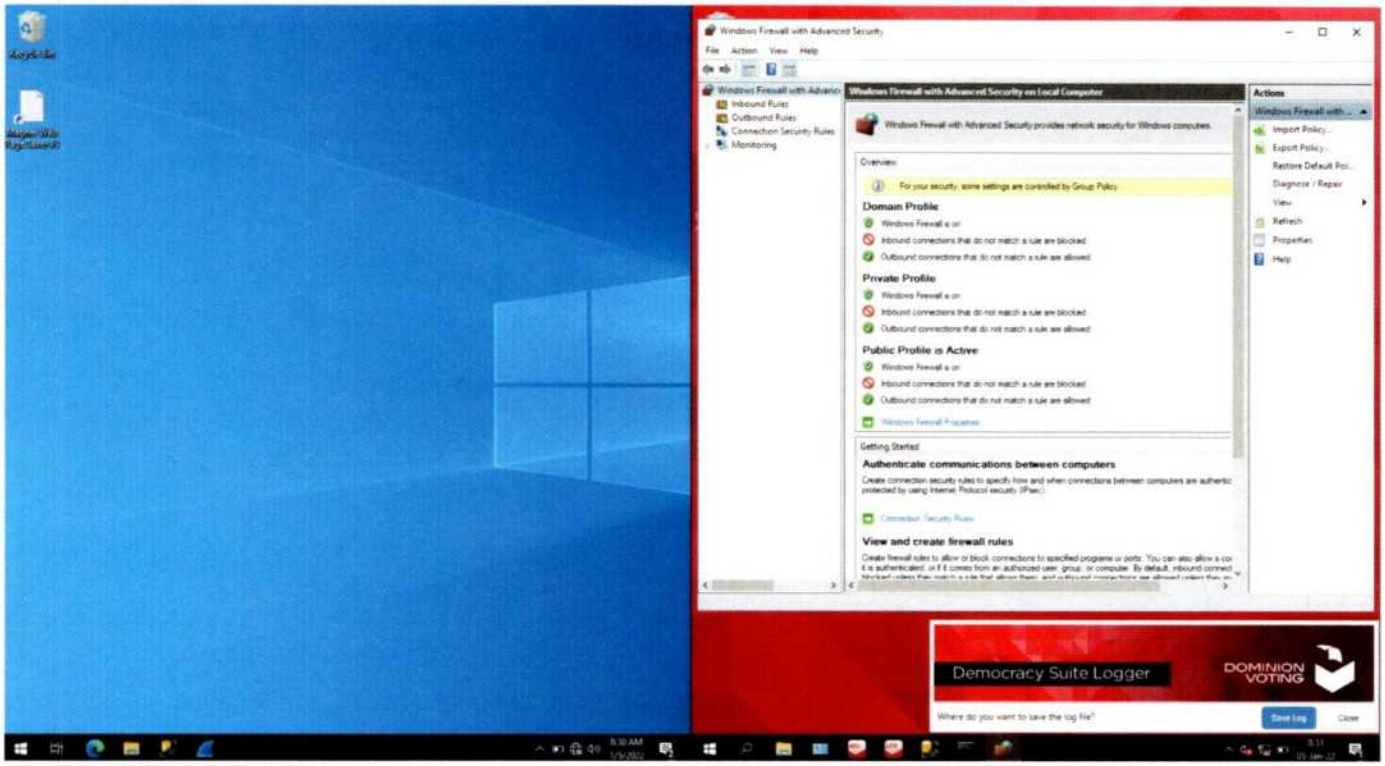


Figure 31 - EMS server has the 'Windows Firewall' enabled

Because it was trivial to connect directly to the database server on port 1433, the firewall was then checked to see if it was enabled on the server. This figure shows that the Windows Firewall with Advanced Security is installed and enabled, however the configuration of the firewall must now be examined to see why it allowed this activity.

The Mesa County EMS server contained firewall software, but it is the specific configuration of the firewall that is unsafe. In this screenshot, the firewall is shown to be enabled. For each profile ("Domain," "Private," and "Public"), the settings are the same:

- Windows Firewall is on. **<- GOOD**
- Inbound connections that do not match a rule are blocked **<- GOOD, but requires further inspection.**
- Outbound connections that do not match a rule are all allowed. **<- RECKLESS FOR A 'SECURE' SYSTEM**

Before going further, it is important to understand what a Firewall is and how it operates. A Firewall is a device that evaluates computer traffic on a network, and based on rules, allows or denies each specific connection. The rules in most common firewalls contain:

- the source IP address,
- source port number,
- Internet Protocol number,
- destination IP address,
- destination port number,

- (Some firewall rules may contain dates and times, for example Monday to Friday 8 am to 5 pm),
- the action to Allow the connection,
- Block the connection,
- Drop the connection, and
- whether to log the connection.

Typically, the rule base is evaluated from top to bottom in order, and the first rule that matches the connection is applied (and the rest of the rule base is skipped). For ANY connection that did not match previously – it is blocked by the Firewall.

It is notable that outbound connections that do not match a rule are set as "Allowed" in this EMS server. For a critical infrastructure voting system, such a configuration is completely reckless. Per VSS[70] and industry best practices systems that require connection should be explicitly specified, and no other outbound connections should be allowed. One of the reasons for such a requirement is that many internet addresses contain malicious software that can be downloaded and installed, sometimes automatically, depending on how they are accessed. The existence of such malicious software has given rise to an entire Anti-Virus and Anti-Malware industry.

[70] VSS Volume 1, sections 6.4 and 6.4.2

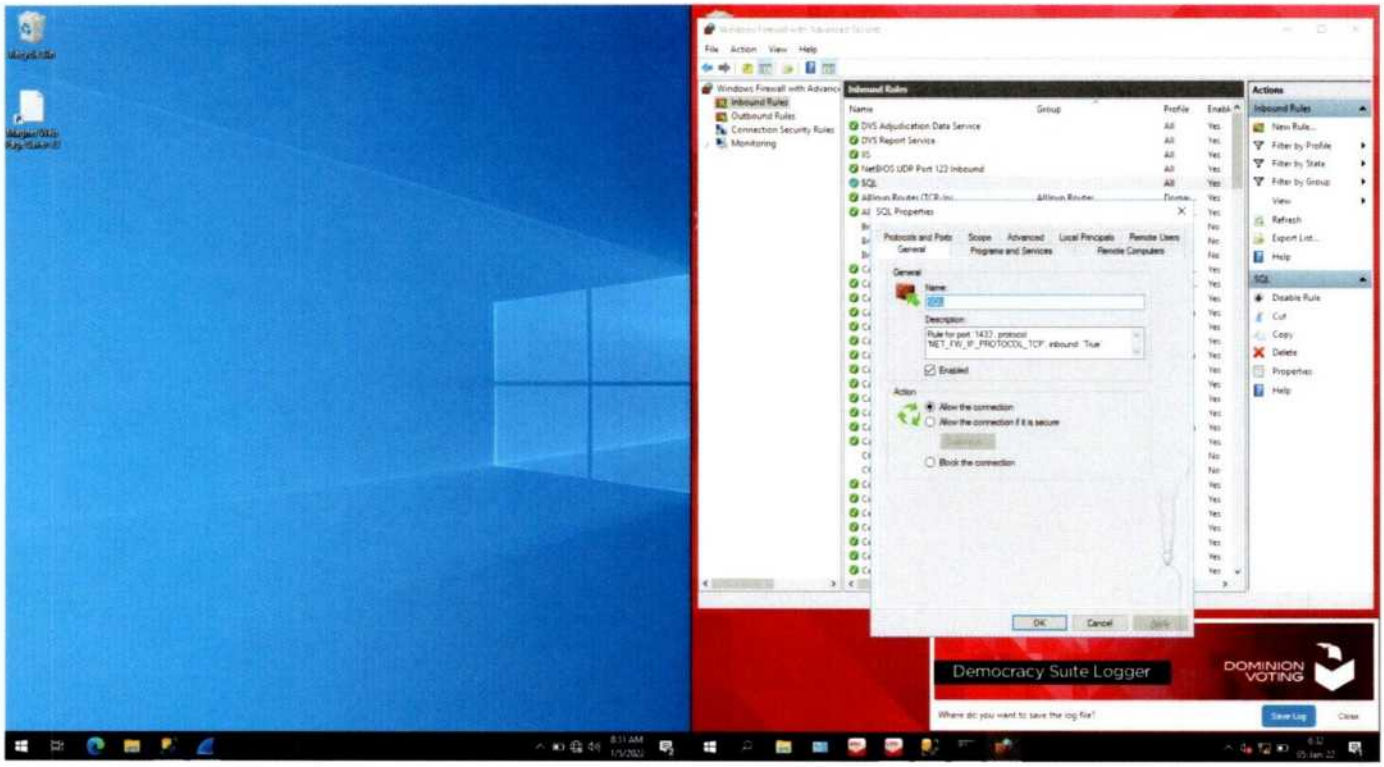


Figure 32 - Windows Firewall Custom SQL entry is enabled

Within the Windows Firewall, a custom firewall rule was found for the SQL service. This rule is not created by Microsoft; it must have been created by another means. The content of the 'SQL' rule is examined and shows the rule is "Enabled," and set to "Allow the connections". Note, the option titled 'Allow the connection if it is secure' just below the chosen option is available however not selected. This means again, the vendor had the option and opportunity to make the system configuration more secure, and neglected to or chose not to, and the individuals involved in the certification either did not check or ignored the vulnerability.

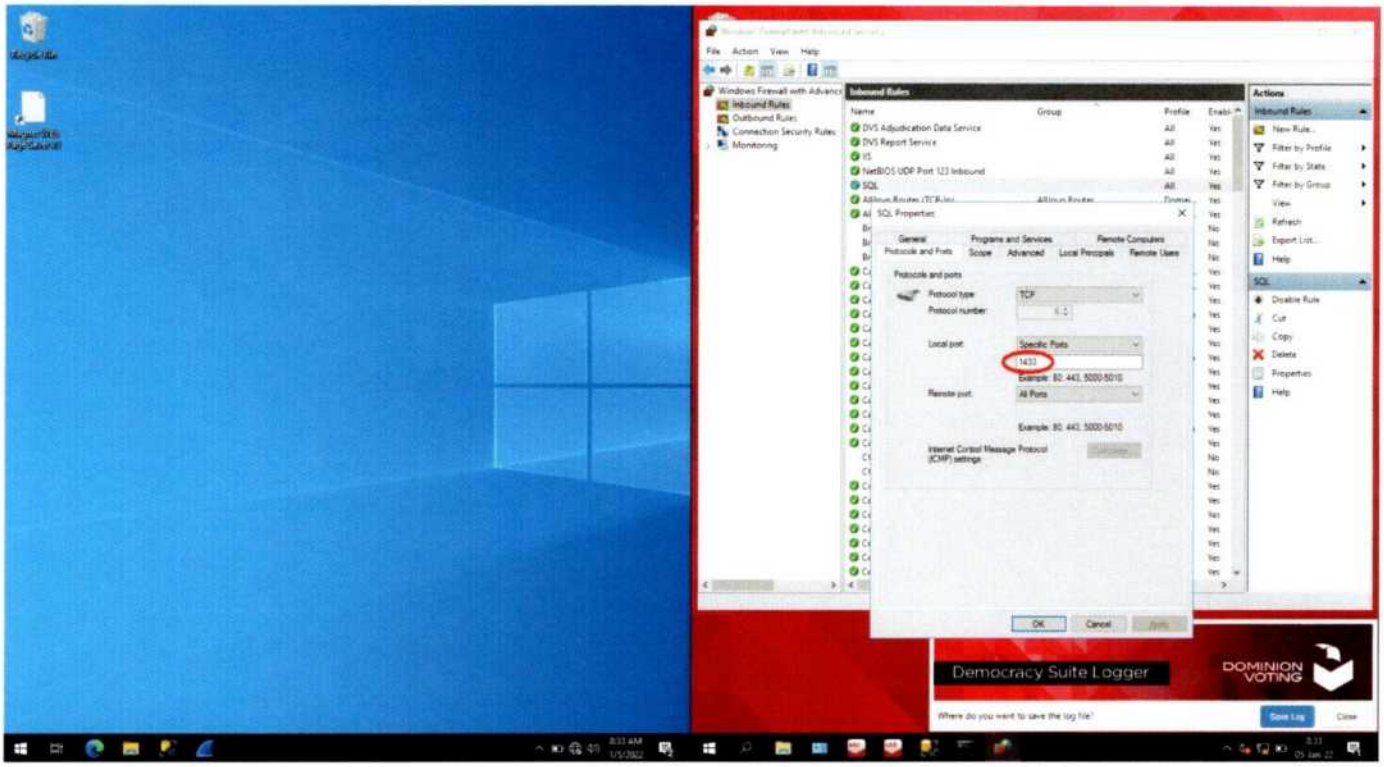


Figure 33 - SQL port 1433 is allowed.

The commonly-known default SQL Service, TCP port 1433 is specifically allowed by this firewall rule.

The port number selected for SQL database access could have been changed so that probing of the computer implicitly revealed less information. This is a recommended technique for high security networks where it is intended that the discovery of systems be disallowed; there are many other recommendations to be followed to truly harden the security of an operating system and its applications.

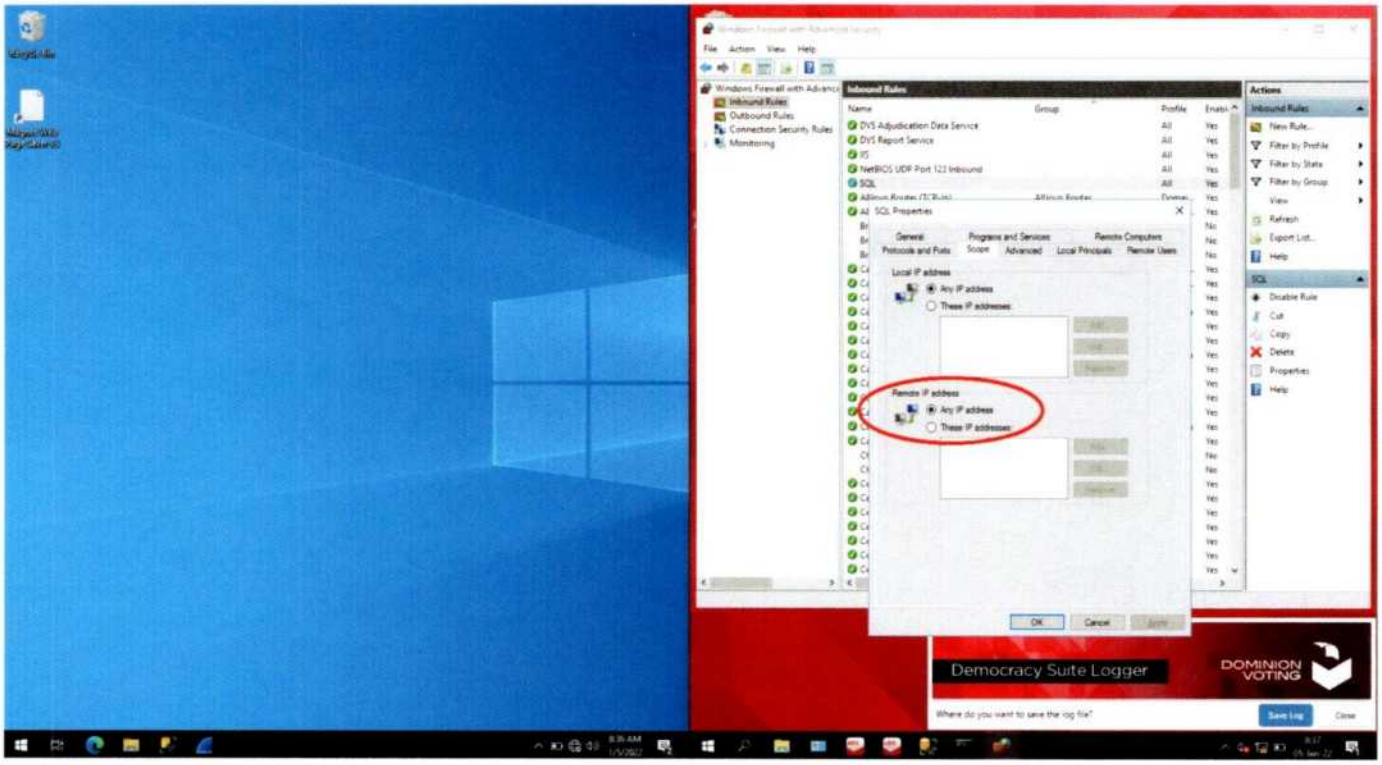


Figure 34 - Access to the SQL database standard port is allowed from ANY IP ADDRESS worldwide.

The IP address of the requesting computer is shown here as 'Remote IP address.' This rule is programmed to allow 'Any IP address' to connect to this port. Any IP address applies to any IP address anywhere in the world.

The ability to make the server more secure has been included by Microsoft and made easy to implement in the graphical user interface (GUI), specifically by allowing for the specification of Remote IP addresses to be accepted (which would exclude all those not explicitly listed). Microsoft documentation states:

"**Any computer (including computers on the Internet)**: Not recommended. Any computer that can address your computer to connect to the specified program or port. This setting might be necessary to allow information to be presented to anonymous users on the internet, but increases your exposure to malicious users. Enabling this setting can allow Network Address Translation (NAT) traversal."

The option to specify a list of IP addresses is present in the GUI, "These IP addresses:" but is not selected.

Again, DVS had the option and opportunity to make the system configuration more secure, and neglected to or chose not to, and the individuals involved in the testing and certification either did not check or ignored the vulnerability.

Instead, they configured the option that Microsoft states is "Not recommended" and "increases your exposure to malicious users."

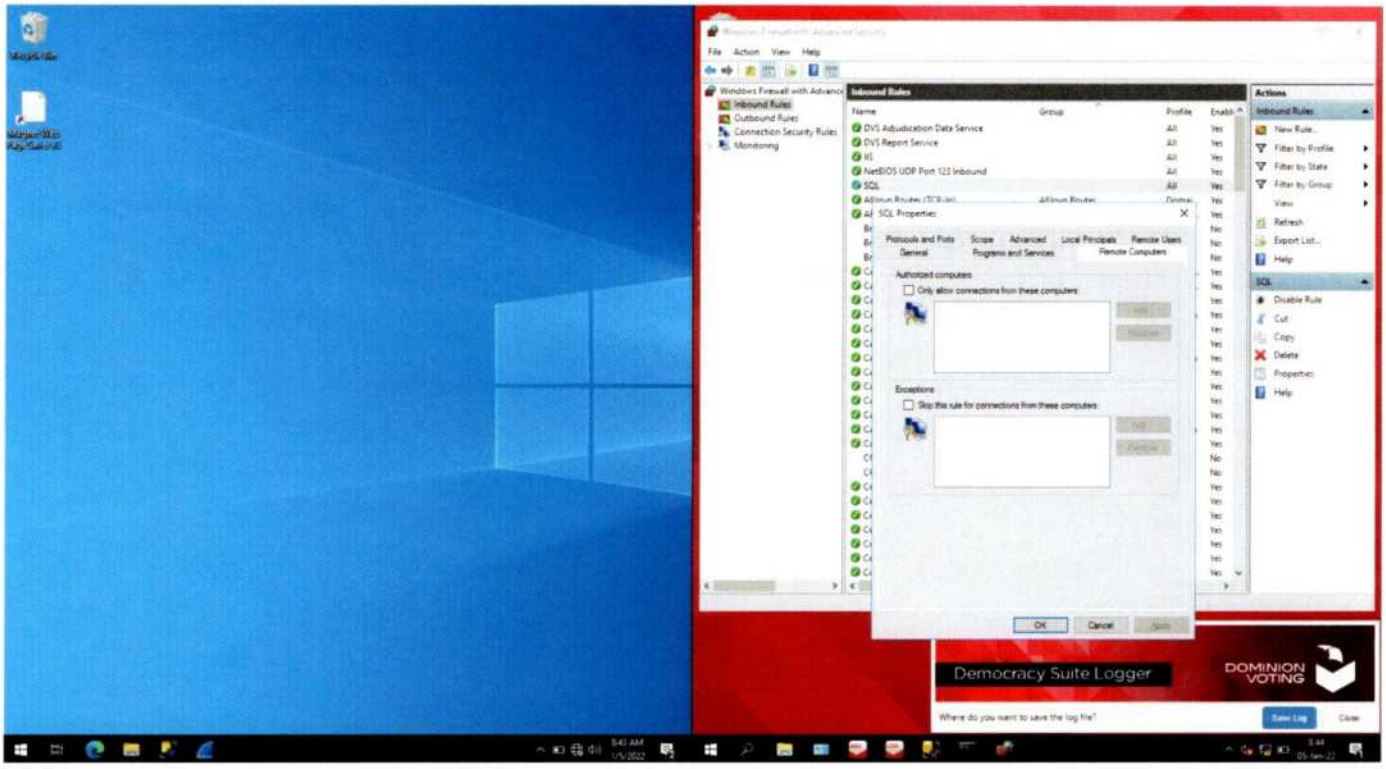


Figure 35 - No additional IP address restrictions or permissions

No restrictions are in place on the firewall that require authentication or integrity-protected communication on the network. The vendor could have specified as "Authorized computers" only those computers and devices deployed within the DVS D-Suite 5.11-CO voting system configuration in Mesa County, and excluded any and all other computers and devices in the world. But the vendor does not restrict that communication and, again, neither the voting system testing lab nor the Secretary of State staff took note or action regarding that neglect of a required security setting. For such a 'secure' critical system ("critical infrastructure," according to the U.S. Government), there is no excuse for this lack of security to help guarantee integrity of each citizen's vote.

It is possible to restrict access to a designated set of computers and even ensure that the connections are authenticated and integrity-protected. The functionality for this is built-in to the operating system, had the voting system vendor chosen to configure it. This safeguard of network traffic authentication and integrity-protection is available, but unused by DVS in this image of the Mesa County EMS server configuration.

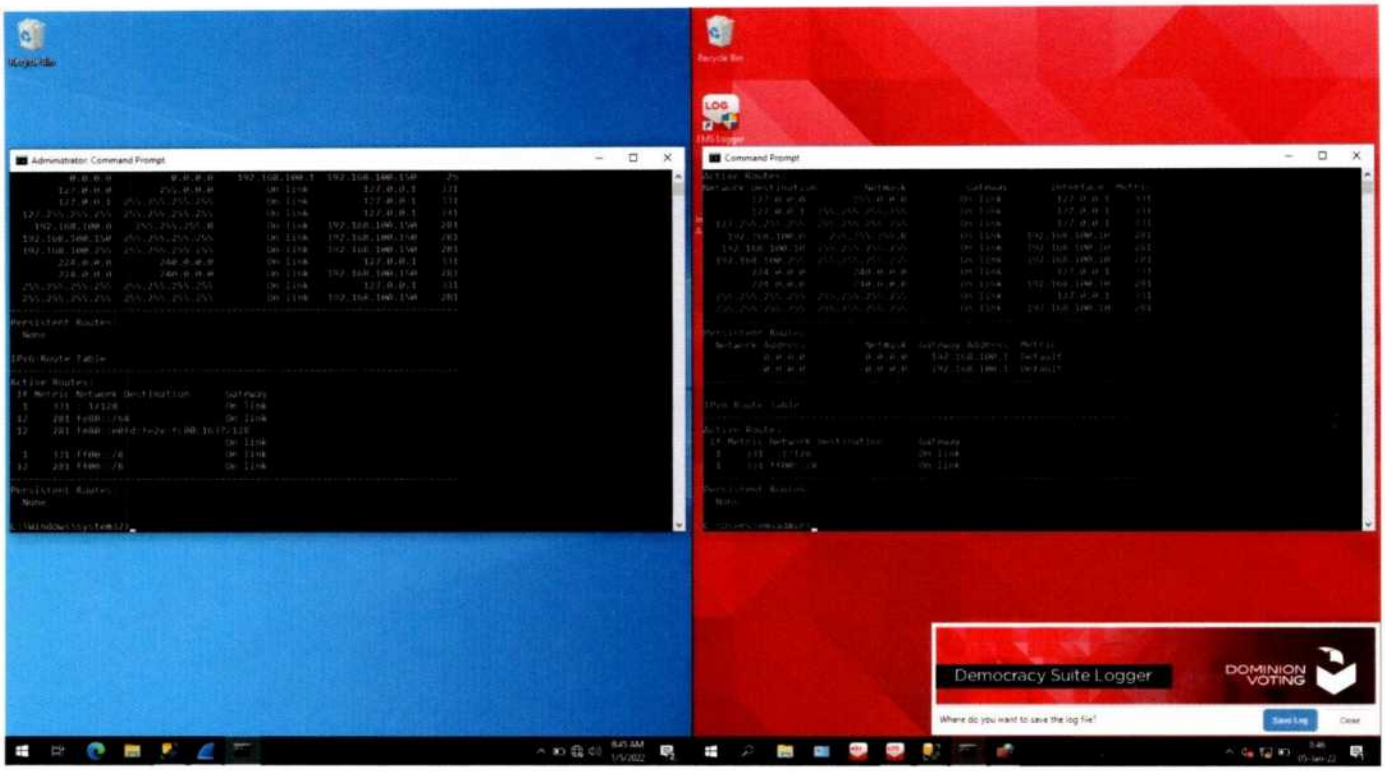


Figure 36 - Test Workstation, 192.168.100.150, and EMS, 192.168.100.10, are on the same subnet

This is demonstrating that the IP address for the Test Workstation on the left is on the same subnet as the IP address for the EMS Server on the right.

This address configuration shows that the test workstation and the EMS server are configured on the same subnetwork, i.e., "subnet," e.g., they should be able to connect to each other if there is not something restricting them from doing so. If they were not on the same subnetwork, a router would be required but is unnecessary in this examination for the finding demonstrated here.

Testing the connection from an external Test Workstation tests the totality of the EMS server configuration and assures that claims of being able to connect from a separate computer not part of the DVS system are valid. Specifically, this test assures that no additional countermeasures or configuration of the EMS server are overlooked in arriving at this conclusion.

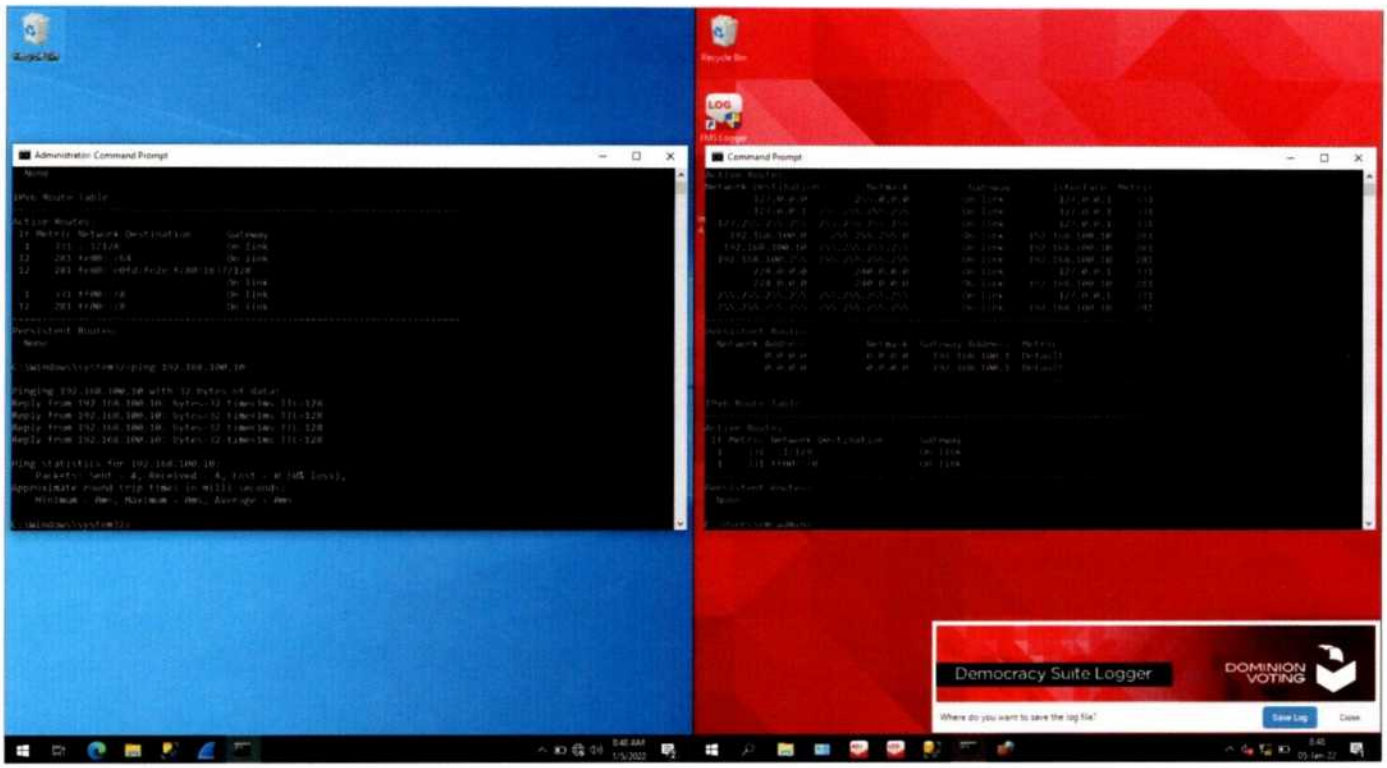


Figure 37 - Mesa EMS server is responding to network ping test.

'Ping' is another common diagnostic utility being used to determine if the EMS server on the right responds to the request from the Test Workstation on the left. All 4 responses were received by the Test Workstation from the EMS server, in response to the 4 requests sent by the Test Workstation.

In a properly highly secured network, one would expect the Internet Control Message Protocol (ICMP) request to be disallowed on the EMS server, in order to help prevent the unauthorized or malicious discovery of the DVS D-Suite network structure of devices and addresses.

This test demonstrates the lack of such restriction: the EMS server responded to the request.

The ping test uses Internet Control Message Protocol (ICMP) and transmits an "echo request" to the echo service on a remote computer. The remote computer responds and the original computer records the time it took to return the request. This is commonly used to determine if a device with a particular IP address is present on a network. This test demonstrates that the Test Workstation is connected to the EMS server across the network.

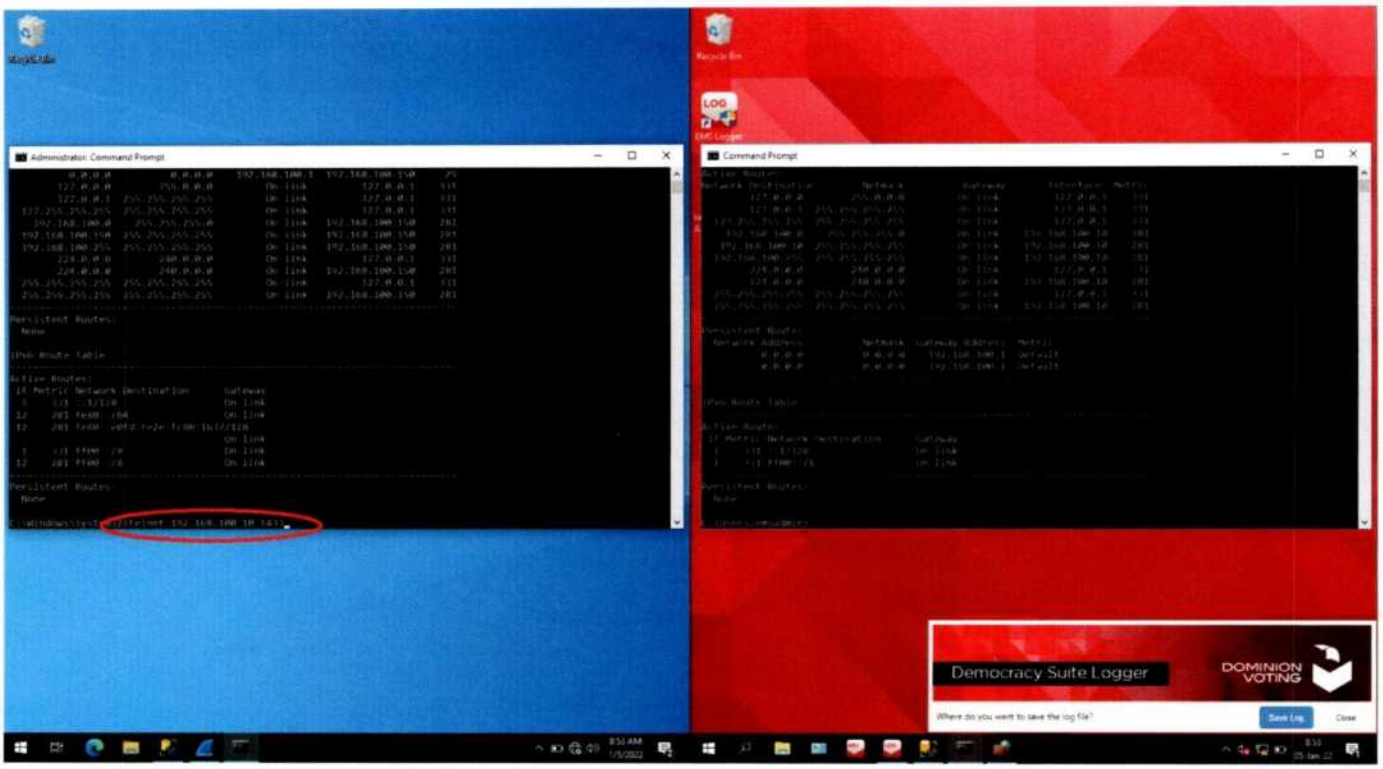


Figure 38 - Telnet connectivity test from separate computer not part of the Dominion system

The same 'Telnet' command (as in Figure 28) is used to see if the commonly-known default configured SQL Server port of 1433 on the EMS server at 192.168.100.10 can be connected to this alternate non-DVS D-Suite system.

Having established that the test workstation can connect to the server IP address, the Telnet command is used to test the connection to the EMS server's SQL service. Previously this connection was attempted from the EMS server to itself. The connection from the Test Workstation, a separate computer not part of the DVS D-Suite system, is attempted here.

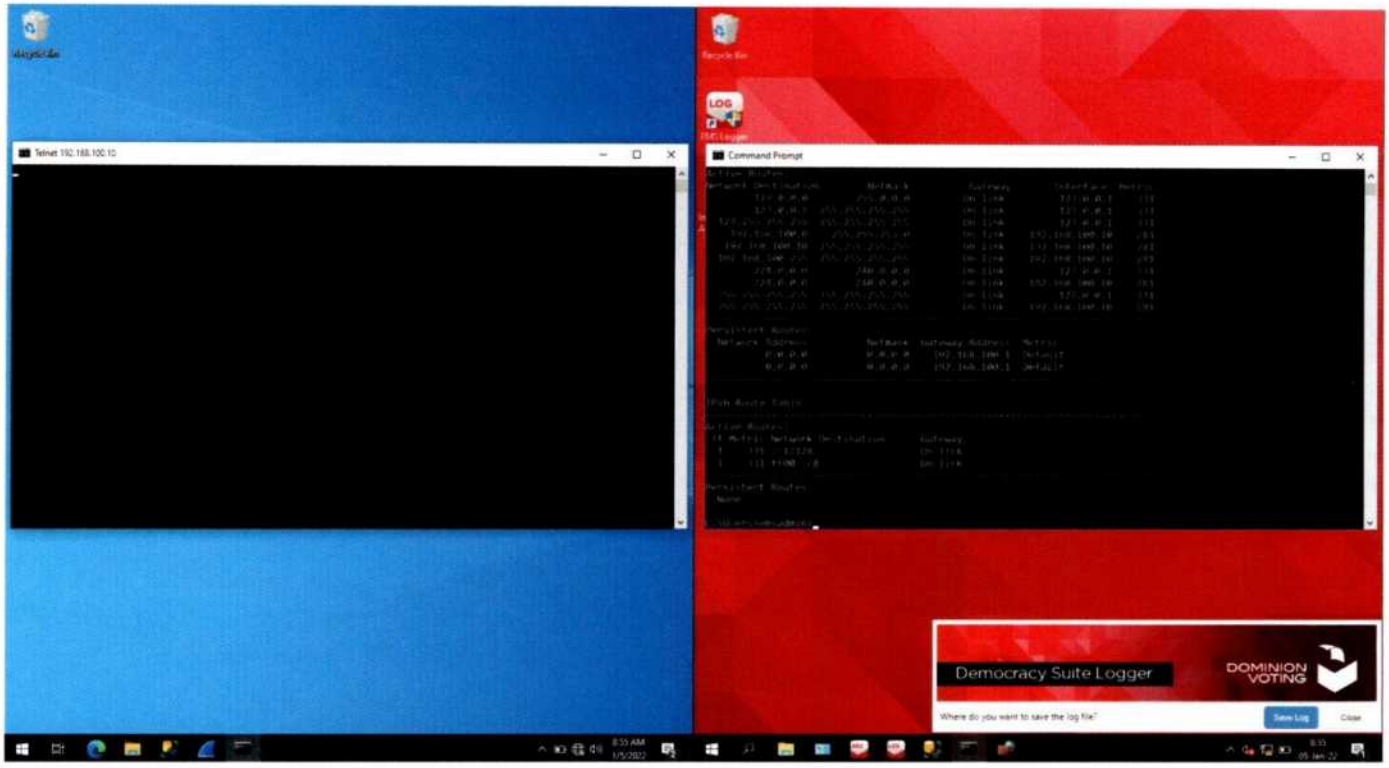


Figure 39 - Telnet to EMS server port 1433 (SQL) succeeds

Just as when this same test was run on the EMS server itself, the connection to the SQL Server port 1433 on the EMS server is successful from the Test Workstation.

The Telnet utility from the Test Workstation is able to connect to the EMS server showing, as in the Telnet test from the server to itself, that the SQL database service port is operating and listening for connections, and accessible from a non-DVS D-Suite computer.

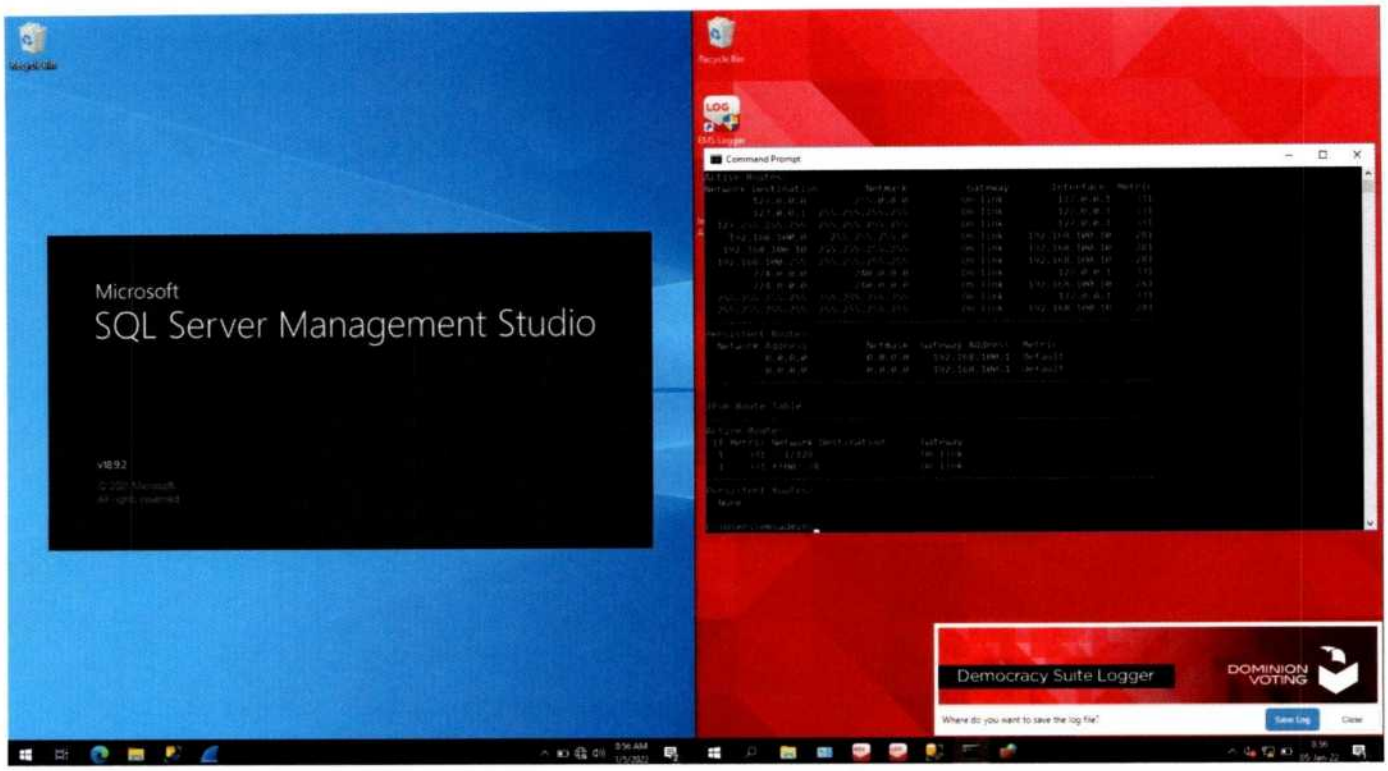


Figure 40 - SSMS access test from separate computer not part of the DVS D-Suite system

SSMS is downloaded from Microsoft and installed on the Test Workstation. Here, it is started, just as it was on the EMS server previously.

Anyone could do this by following the simple directions found with an Internet search for 'how to download SQL server management studio.' There are also many videos on the internet that walk even a novice through doing so.

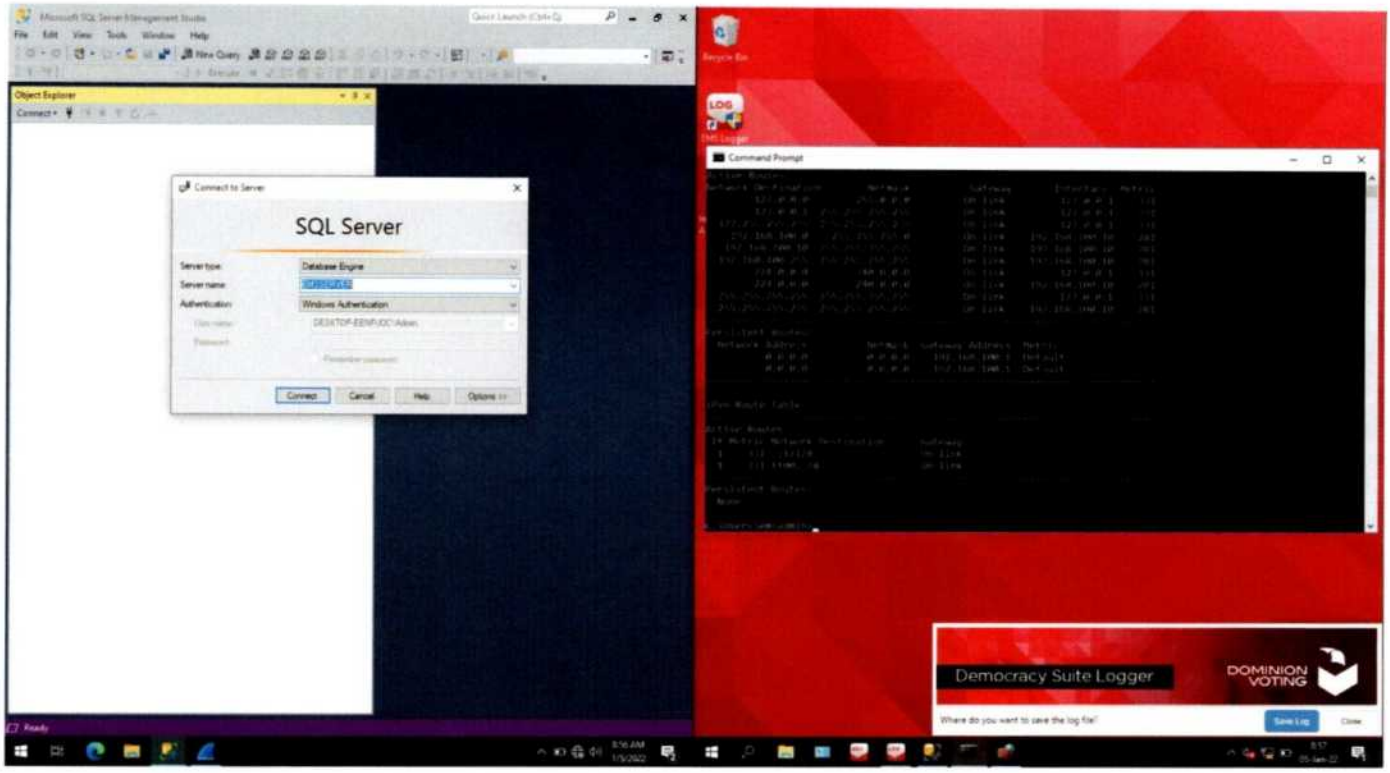


Figure 41 - Log In to the server

A user account was created on the Test Workstation using the same username and password that was used to log in to the EMS server on the right. SQL Server Management Studio was started and the same computer name 'EMSSERVER' was typed into the 'Server name' field on the Test Workstation.

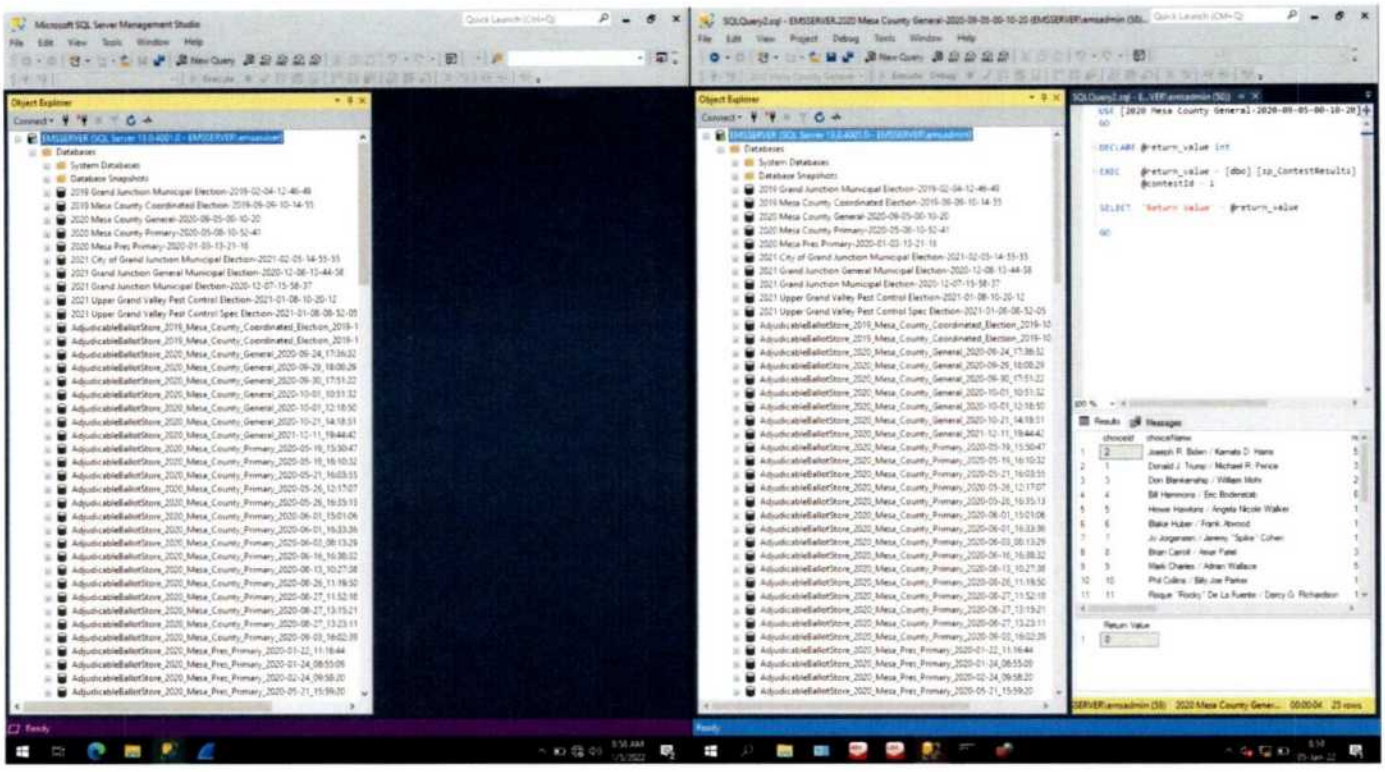


Figure 42 - From a separate Windows 10 computer EMS server database access has been obtained.

After clicking 'Connect,' SQL Server Management Studio connected successfully without so much as a warning. Clicking on the '+' next to Databases reveals the same list of databases available on the EMS server itself, accessible from the Test Workstation.

In Figure 42 I have obtained access to the EMS server from a separate computer not part of the Dominion system and can see election databases. On the left side of the screenshot, the display from the test workstation is shown and on the right side of the screenshot the display from the EMS server is shown. Both systems show the same databases listed. Remote access (i.e., from a separate computer not part of the Dominion system) to the database has been obtained by the Test Workstation.

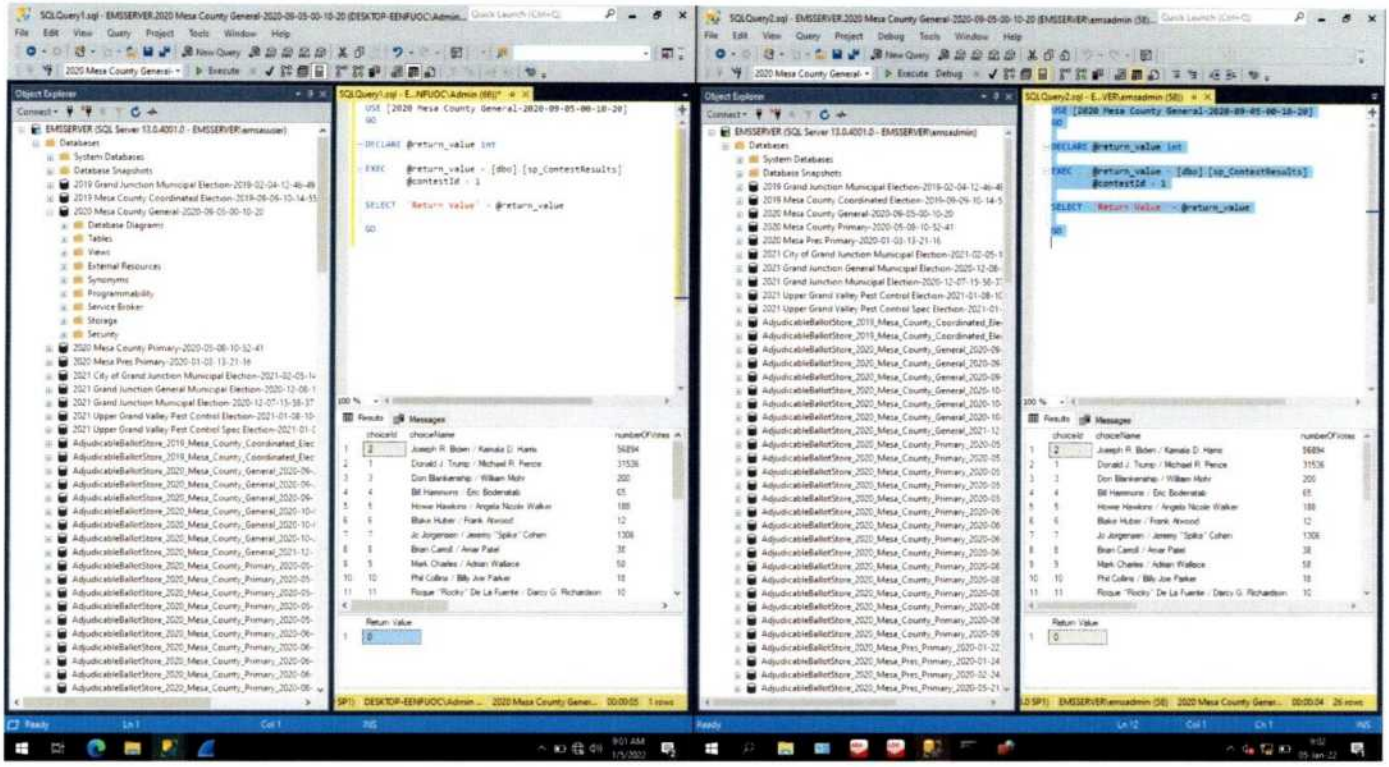


Figure 43 - From a separate Windows computer, the databases can be accessed and reports run.

To confirm this is the data directly from the EMS server, the same report is run on both systems. They both report identical information from the database.

The results display the database in the altered state in which it was left, showing the flipped 56,894 votes for Biden and 31,536 for Trump from the test illustrated in Figure 28.

Finding 5: The security configuration of the Mesa County EMS server permitted access to election data and records from a separate computer not part of the DVS D-Suite system.

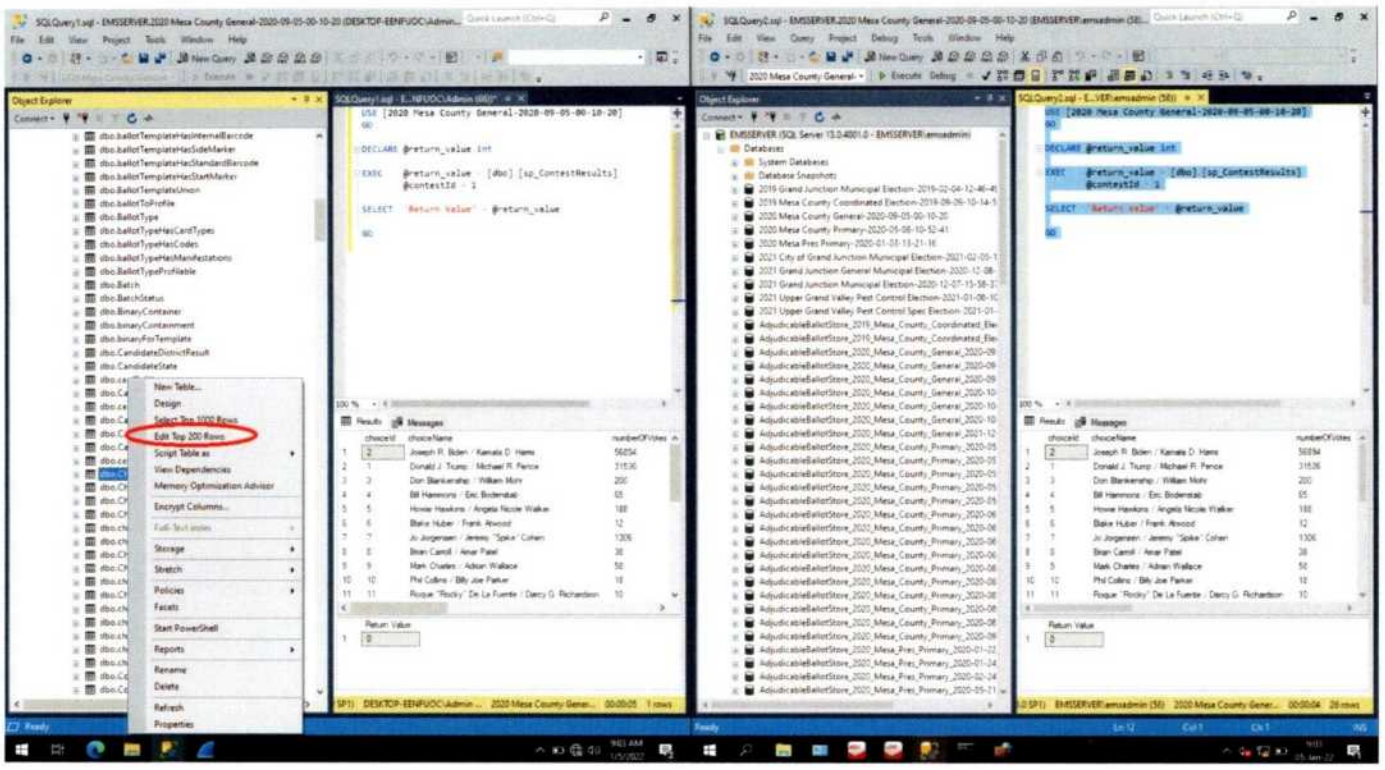


Figure 44 - SSMS permits database Edit

I again right-click on the 'dbo.Choice' table and then select 'Edit Top 200 Rows'.

As previously shown via the EMS server itself, using Microsoft SSMS on a separate computer, not part of the DVS system, access was gained to the same data and the same operations performed as if it was done on the EMS server itself.

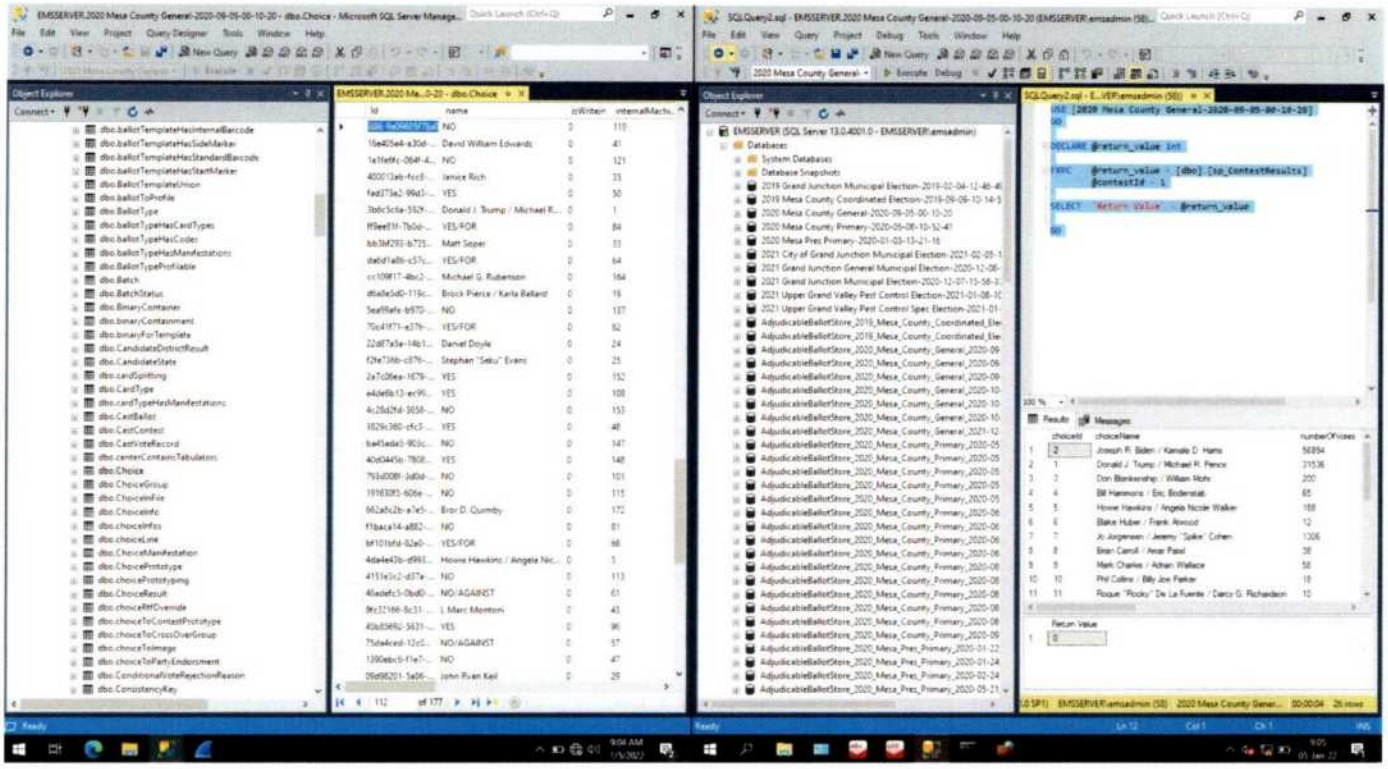


Figure 45 - EMS server Database view from a separate computer not part of the DVS D-Suite system

SSMS shows the same table in the same format as it did on the EMS server.

In Figure 45 the top 200 rows of the election database are available for editing using SSMS running on the Test Workstation to access the Mesa County EMS server across the network. The internalMachineId for Biden is still '2' and for Trump it is still '1' from the previous alteration in Examination Objective 1 (Figure 26).

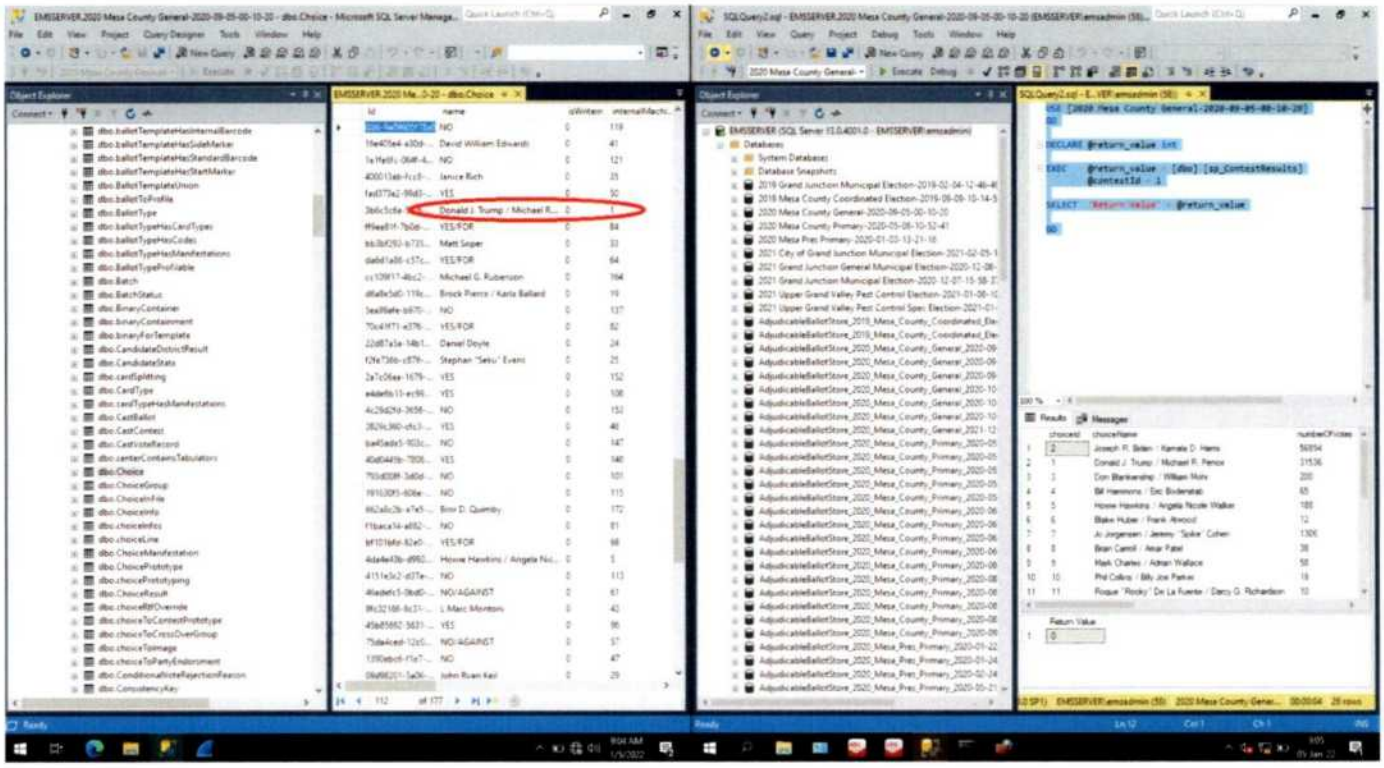


Figure 46 - SSMS permits us to edit the databases

A successful attempt to edit the election database on the EMS server, from the Test Workstation, is made to reverse the changes made earlier, thereby altering them back to the original results. Note the current setting of internalMachineId for Trump is '1.'

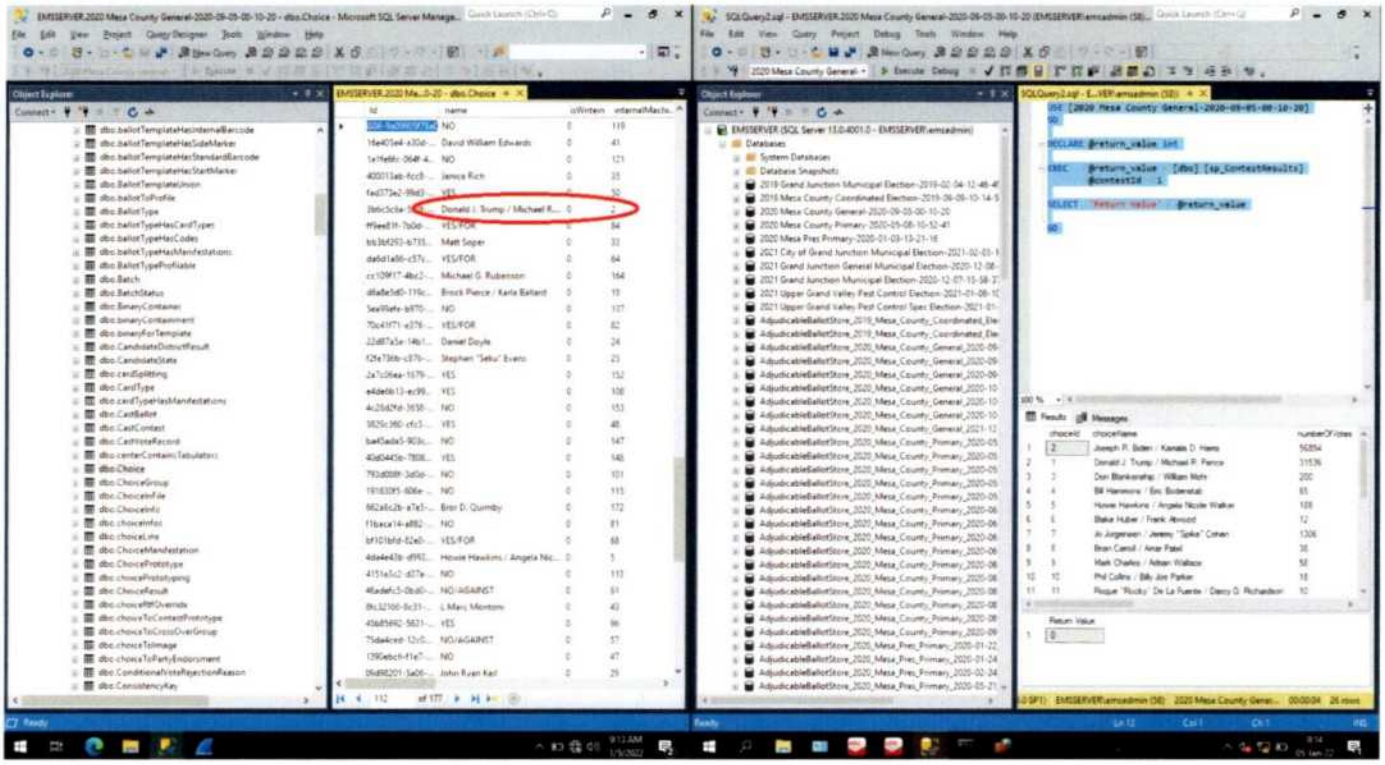


Figure 47 - "internalMachineId" for Trump is now changed back to a 2.

The "internalMachineId" for Trump is changed back to "2." The database server allows this alteration from the Test Workstation without any error or warning.

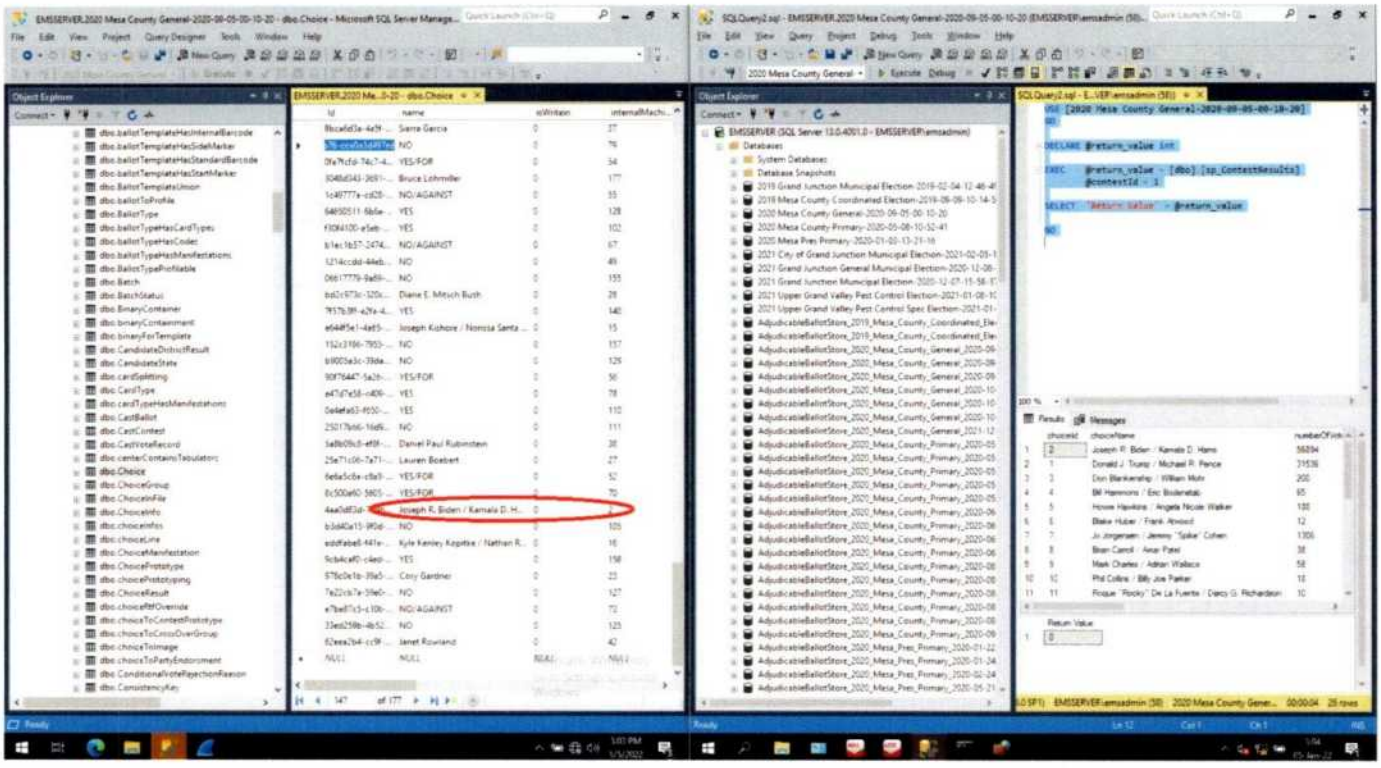


Figure 48 - Candidate data for Biden from previous change

The current "internalMachineId" for Biden is still "2", in the election database on the EMS server, as changed earlier from the EMS server.

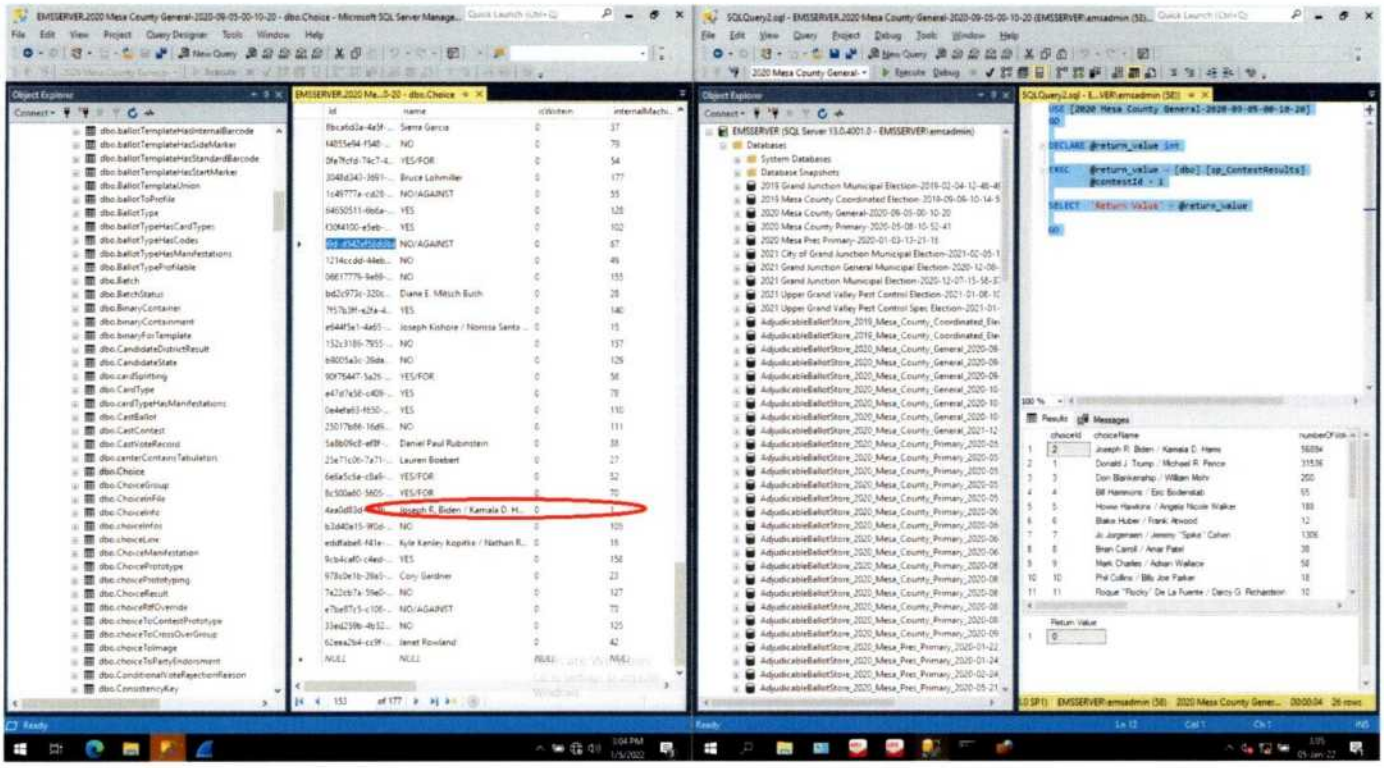


Figure 49 - Candidate data for Biden changed back to original

I next change, from the Test Workstation, the "internalMachineId" for Biden in the election database on the EMS server back to "1", its original value. There is again no error or warning given.

As one can see, this alteration of the voting database was also successful. The system has been restored to the state in which it was found prior to making the first alteration of the voting system database.

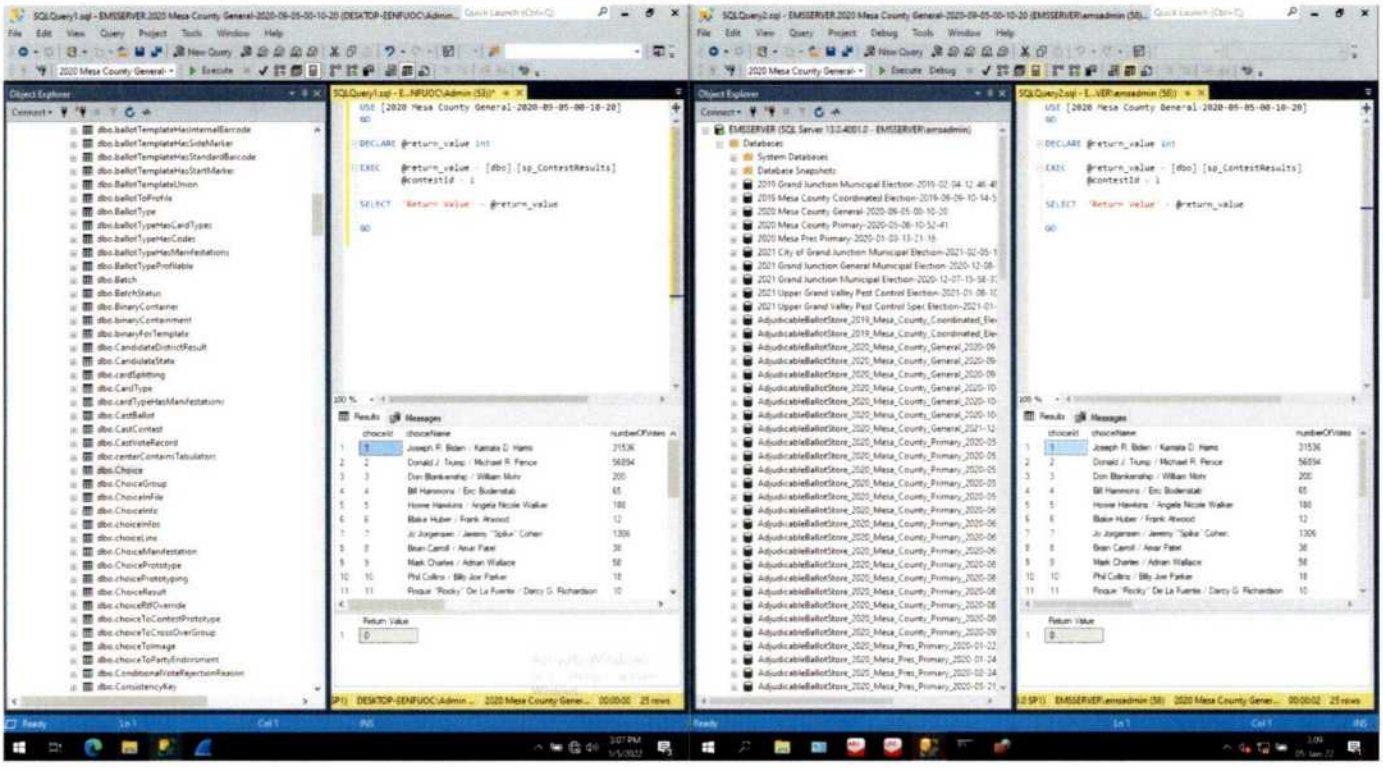


Figure 50 - The vote choice was remotely changed back to its original state

The alterations of the vote totals in the election database on the EMS server also succeeded from a separate computer not part of the DVS D-Suite system. Queries were executed both from the Test Workstation and on the EMS server, and both results again show that it is possible by anyone with physical access to a Dominion Computer or any part of the voting system network to alter the entire election result on the EMS server by changing only two values, with knowledge nearly anyone could attain by using Google and watching one or more YouTube videos.

The query is run on both systems to show that the database results have changed back.

Finding 6: The Mesa County EMS server containing the 2020 General Election vote results has been shown to be insecure and grossly misconfigured such that it allows unrestricted access to the election database and enables changing calculated vote totals from a separate computer not part of the DVS D-Suite system with nothing more than the knowledge of a password. It was possible to access the EMS server and, by changing only 2 numbers in the database, completely alter the election results in Mesa County for the 2020 Presidential election.

EXAMINATION RESULT 2:

The election results database CAN be altered by any person using a non-DVS D-Suite computer directly or indirectly connected to the EMS server network.

EXAMINATION OBJECTIVE 3:

Determine whether the calculated vote totals of an election can be altered by any person using a cell phone or mobile device wirelessly connected to the EMS server network.

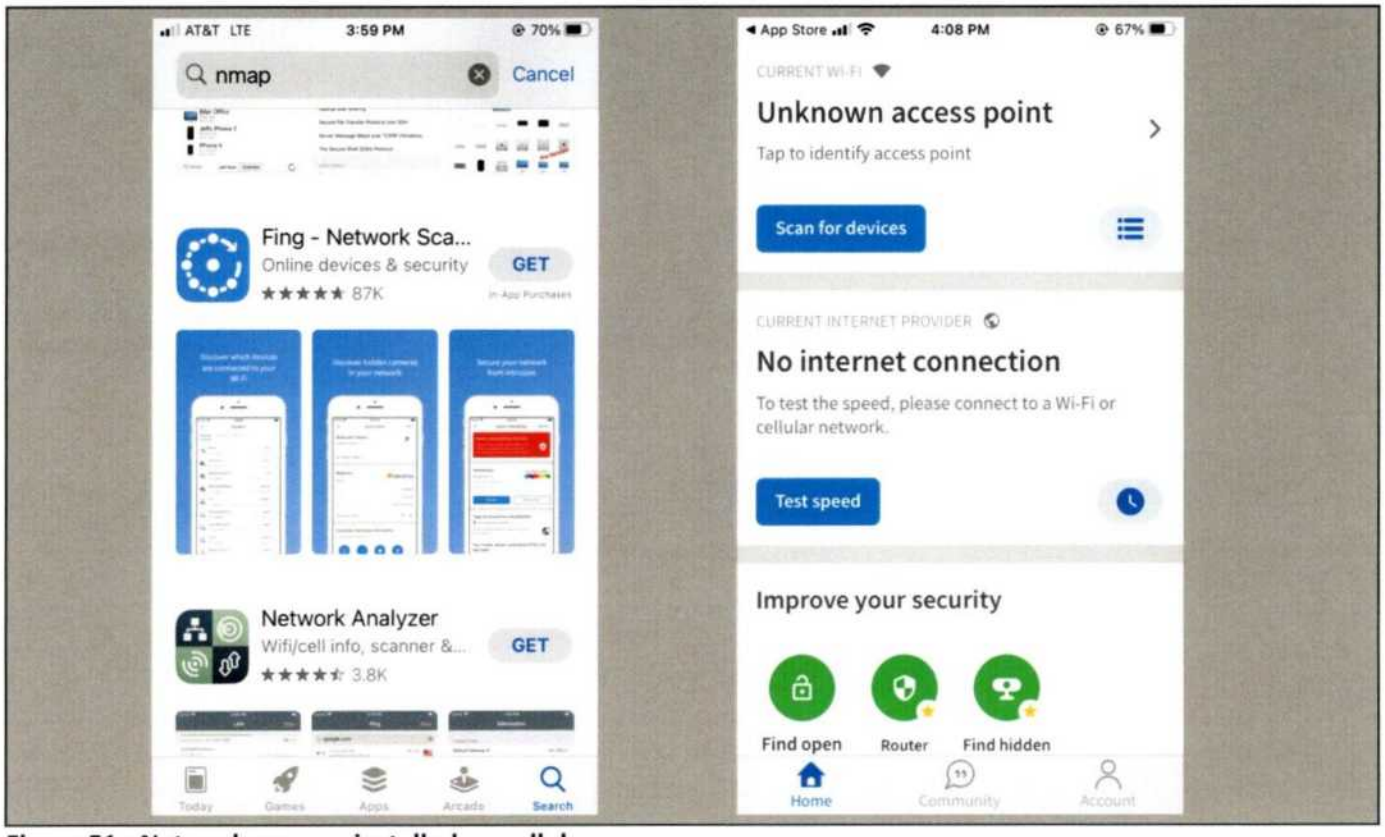


Figure 51 - Network scanner installed on cellphone

An iPhone was connected to the same network, wirelessly, using a common wireless router purchased at a retail store. A router such as this could be plugged in and hidden anywhere on the DVS D-Suite network, or the same functionality could be inserted electronically via common hacking into any device on the network with a wireless card, including network printers and network scanners. As discussed earlier, thirty-five (35) devices of the existing DVS-supplied equipment already had a built-in wireless card or device installed, as well as a wireless-capable printer, so this could have easily been done without attaching any devices outside the system components. The Apple App Store was searched and a common network scanner 'Fing' was easily found. As one can see, 'Fing' has already been downloaded over 87,000 times. In the image on the right, 'Fing' was run and the option 'Scan for Devices' was selected.

Previously an Island-Hopping attack was described. For such an attack to occur, a connection to a different network is used.

This part of the examination was carried out to determine whether the system could have been accessed wirelessly using the more limited capabilities of a mobile device (a cell phone in this test). Thirty-five (35) wireless devices were identified within the Mesa County DVS D-Suite system. In order to perform this part

of the examination it was necessary to mimic the actual MESA hardware, so a wireless access point was connected to the VirtualBox test system that was running the actual software of the Mesa County EMS server via a host-based network interface card.

If any wireless device gains access to any device connected to the EMS infrastructure (as was demonstrated here), including the inadvertent enabling of even a laptop wireless interface (typically performed by a single button press on the keyboard of a laptop, or by preprogrammed, triggered activation of internal code on the device, or by remote command from an actor with access to the device), such an attack could easily occur.

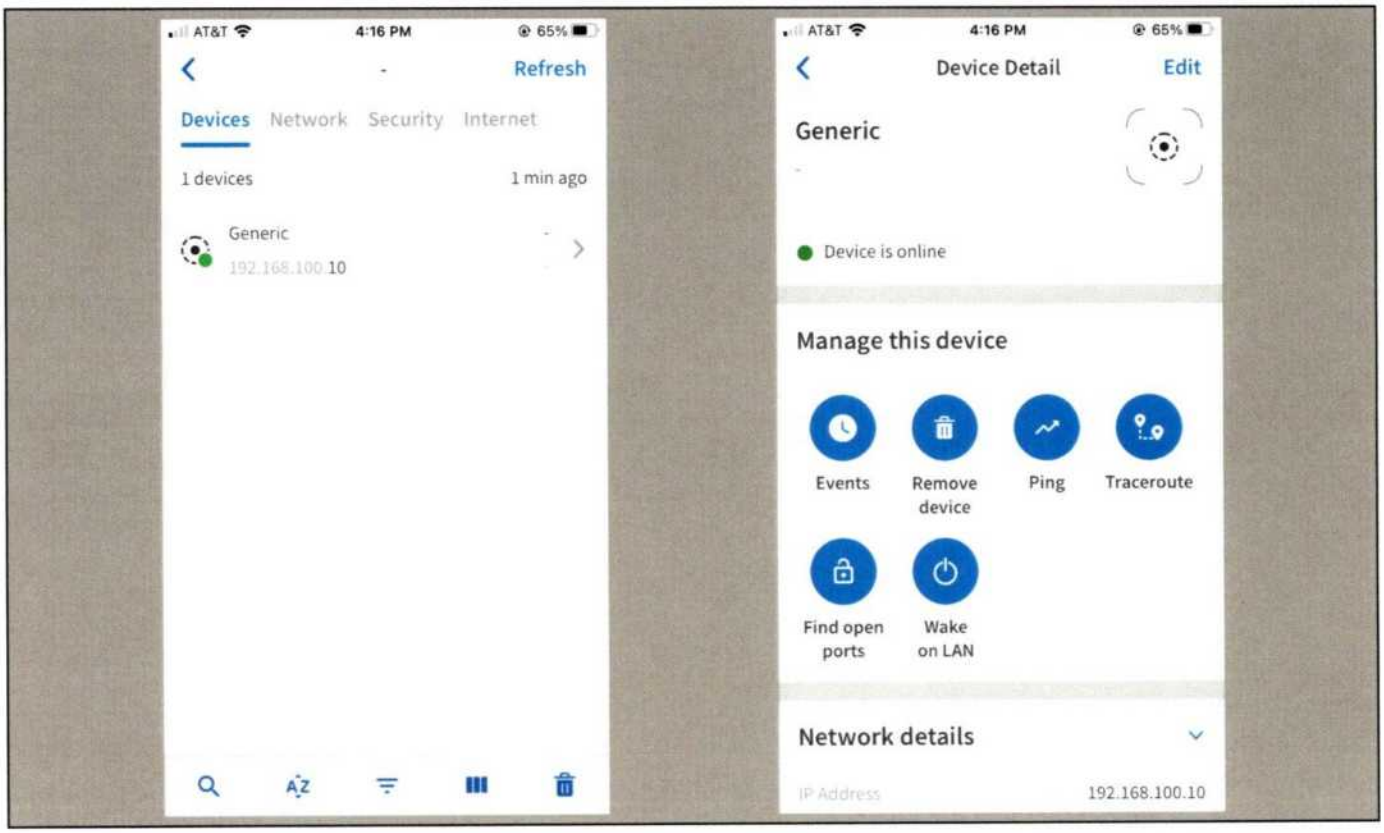


Figure 52 - IP address for the EMS server found via wireless connection and iPhone app

On the left, the network scanner immediately finds the IP address for the EMS Server and displays the IP address (192.168.100.10). The device is selected, and on the right, the phone app presents more options. I then selected "Find open ports."

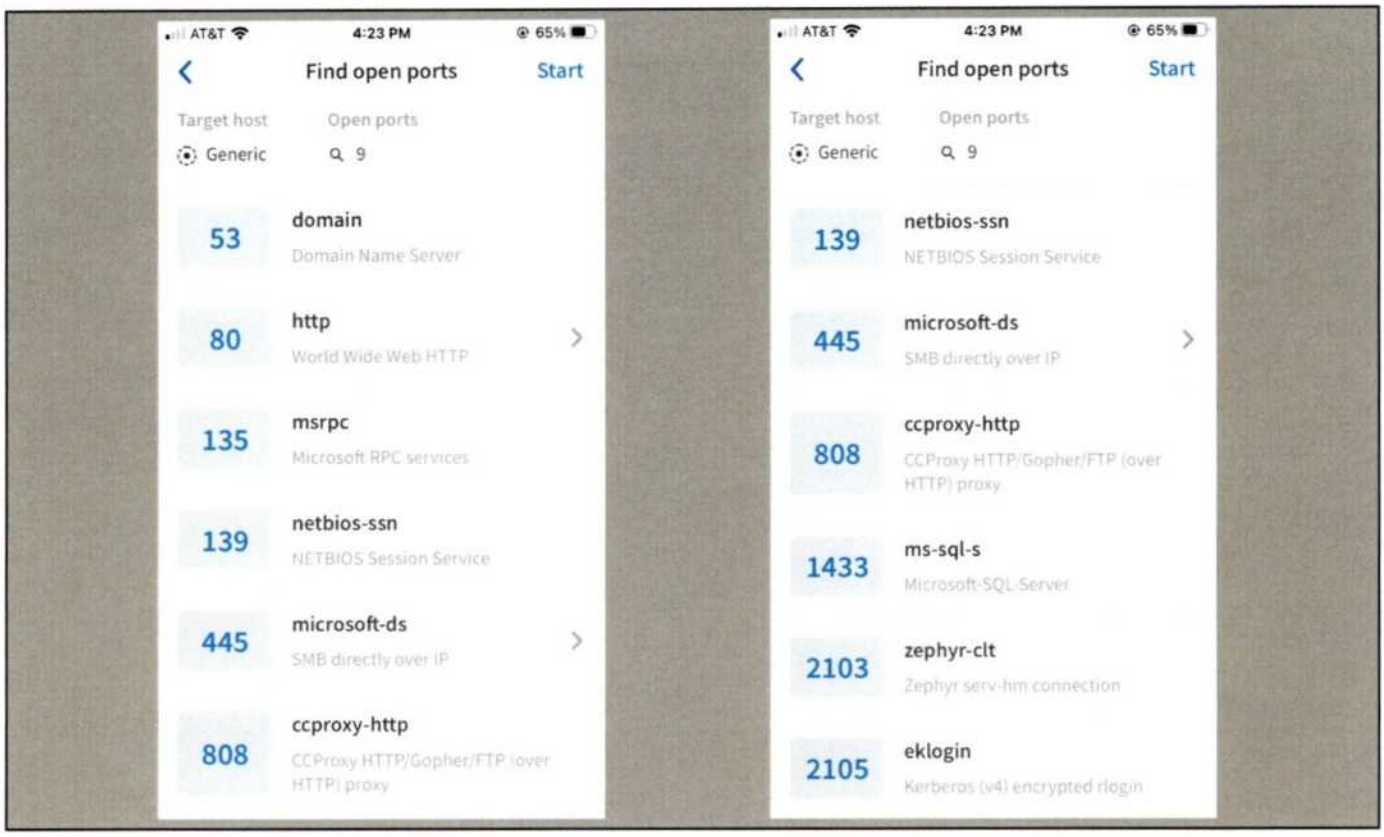


Figure 53 - Scanner Results

The iPhone app lists all the ports that it sees open on the EMS server. Port "1433", which the app indicates is associated with "Microsoft-SQL-Server," Is immediately detected.

In Figure 53, left, one can see the first 6 of the 9 open ports on the EMS server with a wireless access point connected. On the right, scrolling down the screen reveals the remainder of the 9 open ports identified. The SQL service port, 1433, has been identified as operating and configured on this device.

Using the method recommended by CIS (Nmap[71]), a device that offers the Microsoft SQL Service has been identified. This uses standard networking software that many IT professionals and most IT Security professionals are very familiar with.

Whether such an exploitation of technology is performed with the single-response ping command or by using a more powerful tool like Nmap, the discovery of a network connected device on the same network segment has been accomplished.

[71] Network Mapper (Nmap) is a tool for network exploration or security auditing, frequently used by cybersecurity penetration testers to find live / operating devices and hosts on networks, perform port scanning, detect operating systems and versions in use, and ping networks and subnetworks to diagram potential and available communication paths.

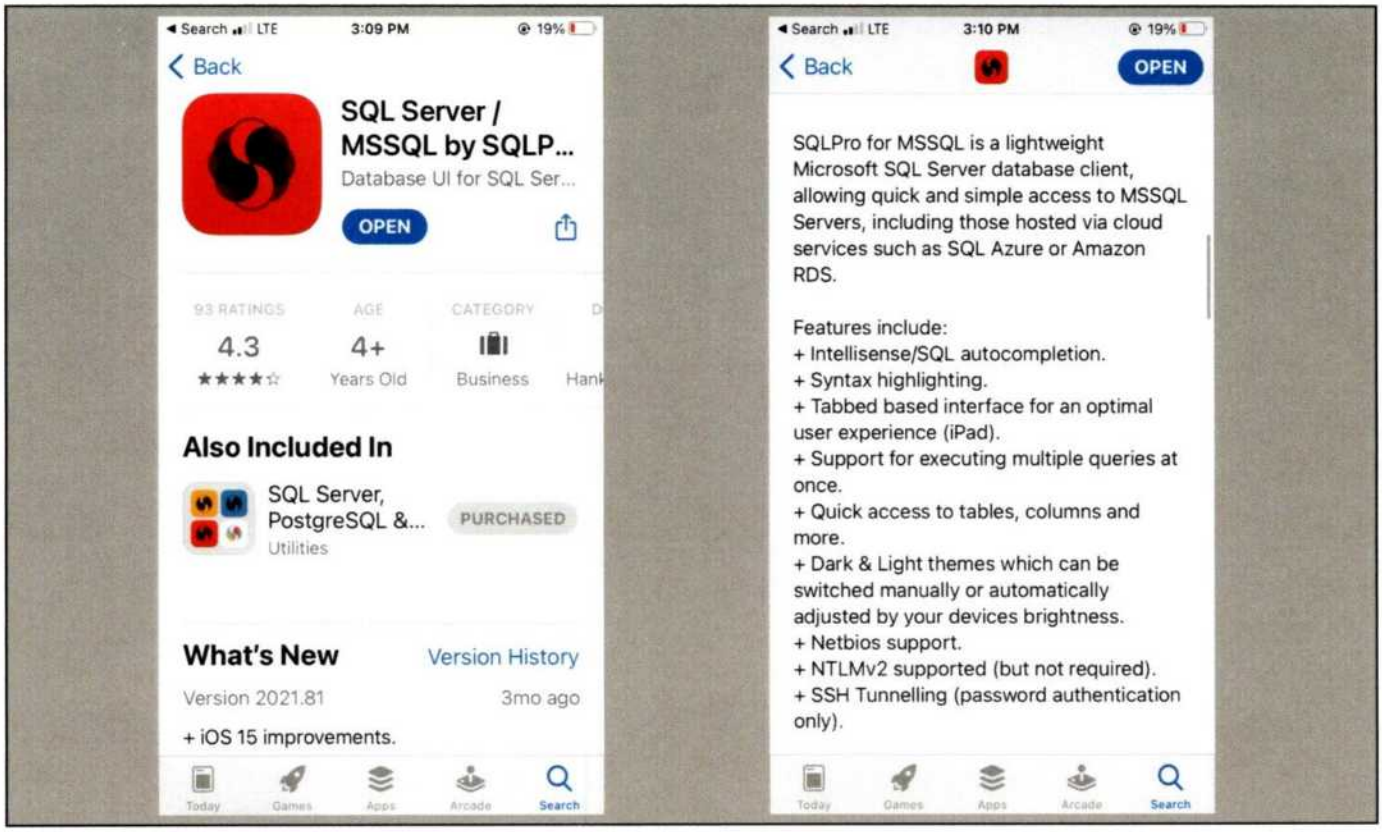


Figure 54 - SQL Access Functionality

Returning to the Apple App Store, a search for 'SQL Server' finds another app, 'SQL Server by SQLPro'. The description shows that it is a Microsoft SQL Server database client.

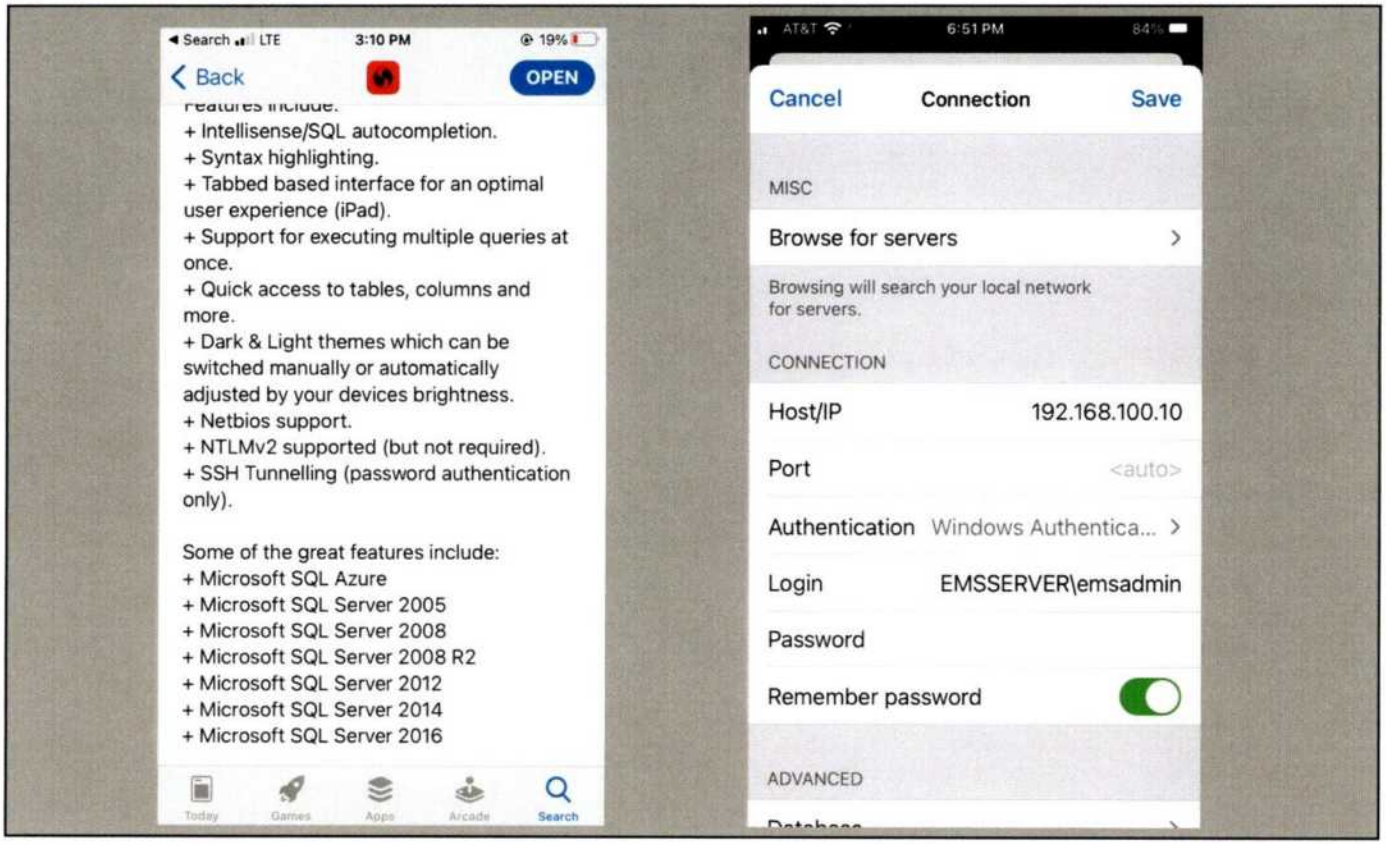


Figure 55 - SQL Pro Capabilities

On the left, the app description shows that it supports Microsoft SQL Server 2016, which is the exact version used by the EMS server. On the right, we use the same IP address, username, and password applied from the iPhone app as previously used to access the EMS server, physically sitting in front of its screen.

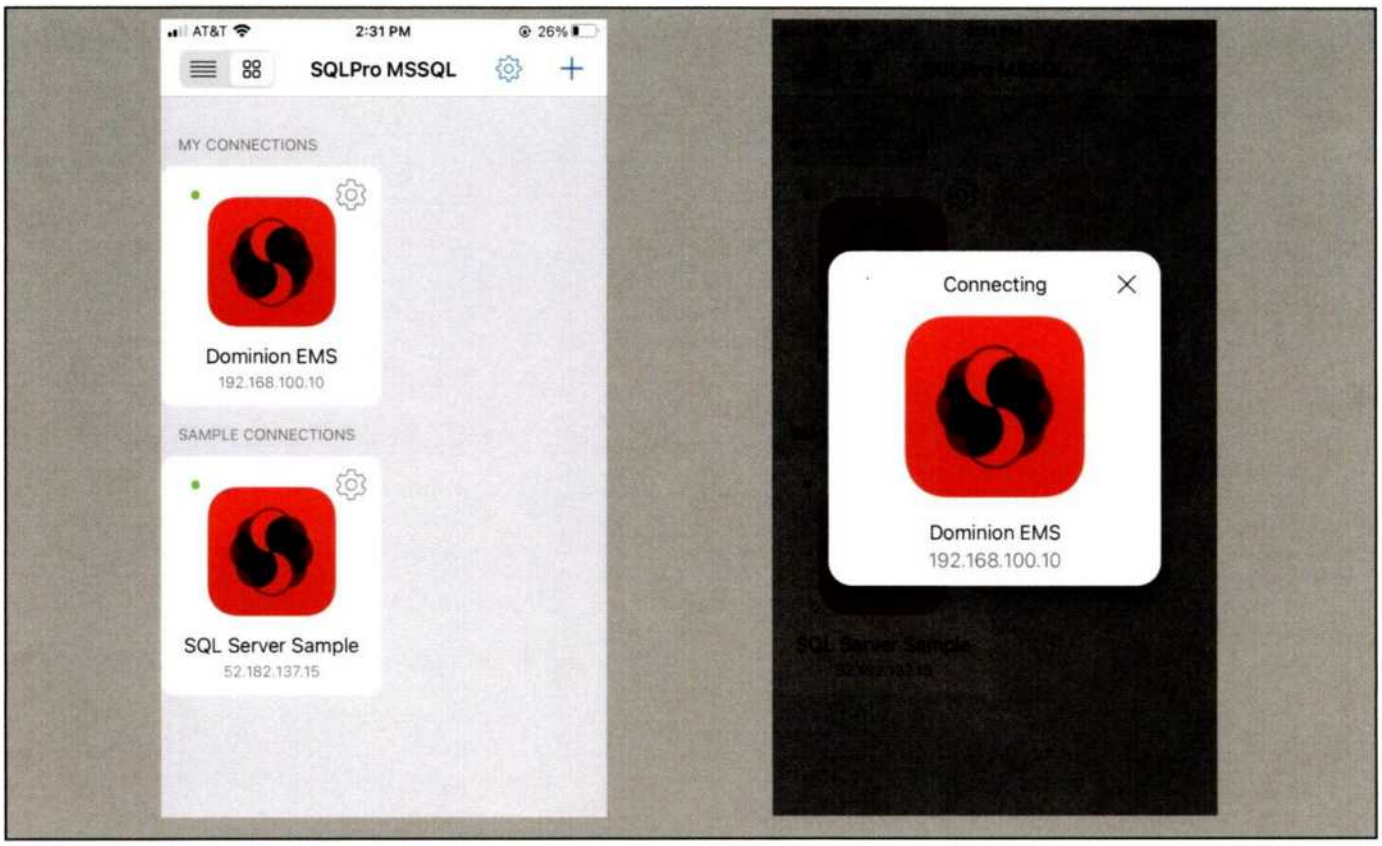


Figure 56 - Making an SQL Connection

The left image shows the configured connection to the EMS server. The right image shows the iPhone connecting directly to the database on the EMS server.

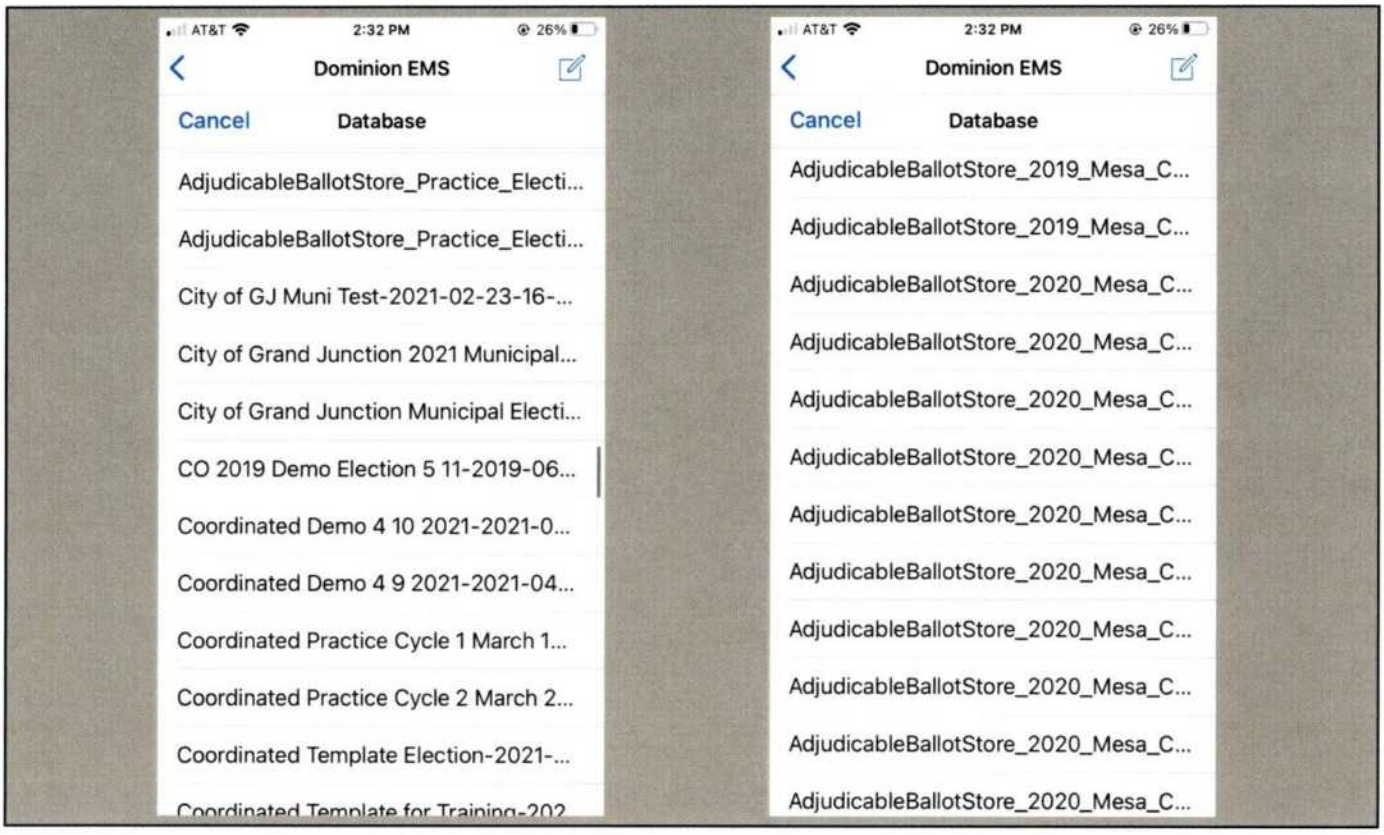


Figure 57 - iPhone Connection to Dominion EMS Database

After a second, the app lists all the voting system databases, just like it did on both the EMS server and on the Test Workstation.

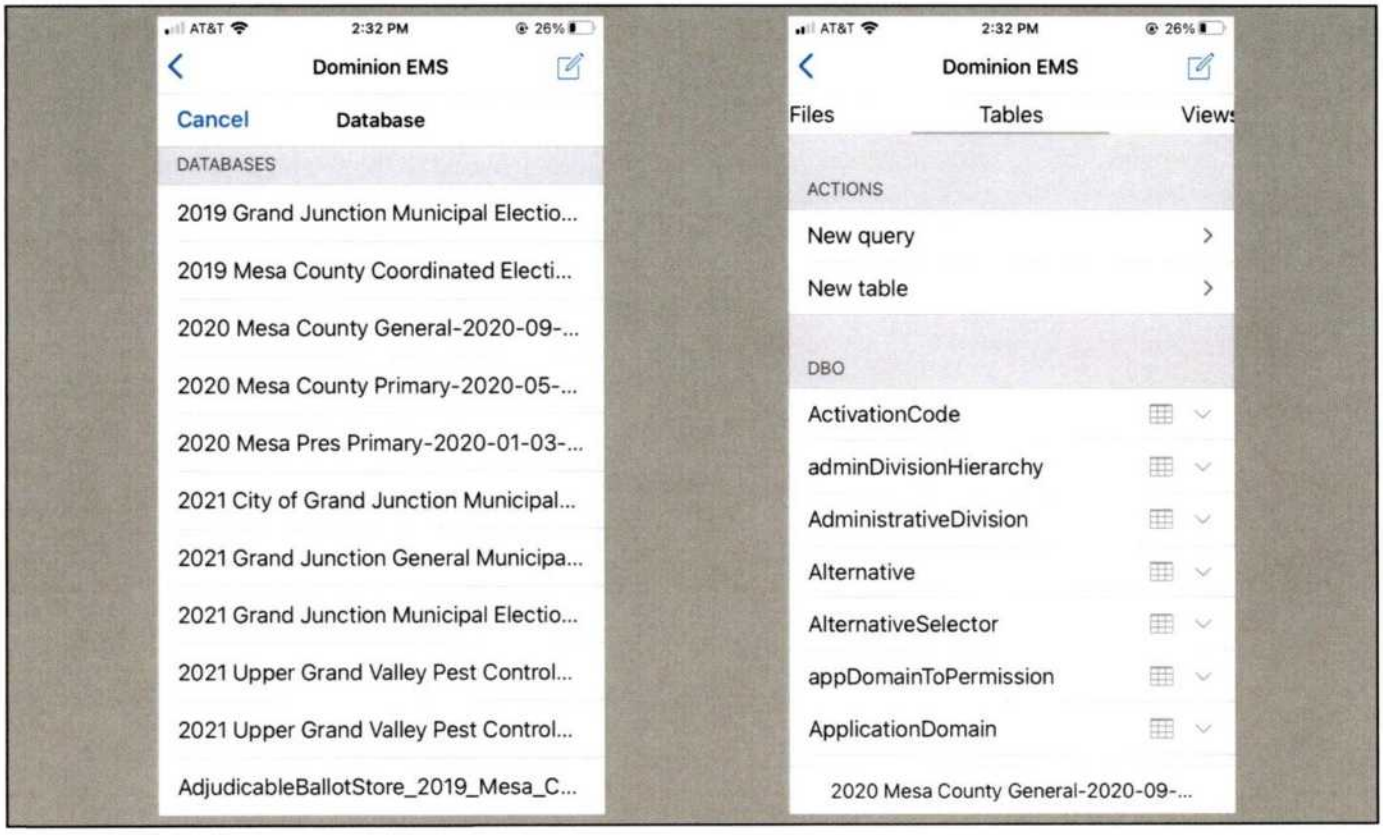


Figure 58 - Databases listing, Continued

Multiple Tabulation Store databases are shown on the left. Next, the 2020 Mesa County General election was chosen from the top of the list, and the image on the right shows the resulting screen, listing the tables in that particular database. So far, the examiner has not been denied access or even experienced a warning of any kind.

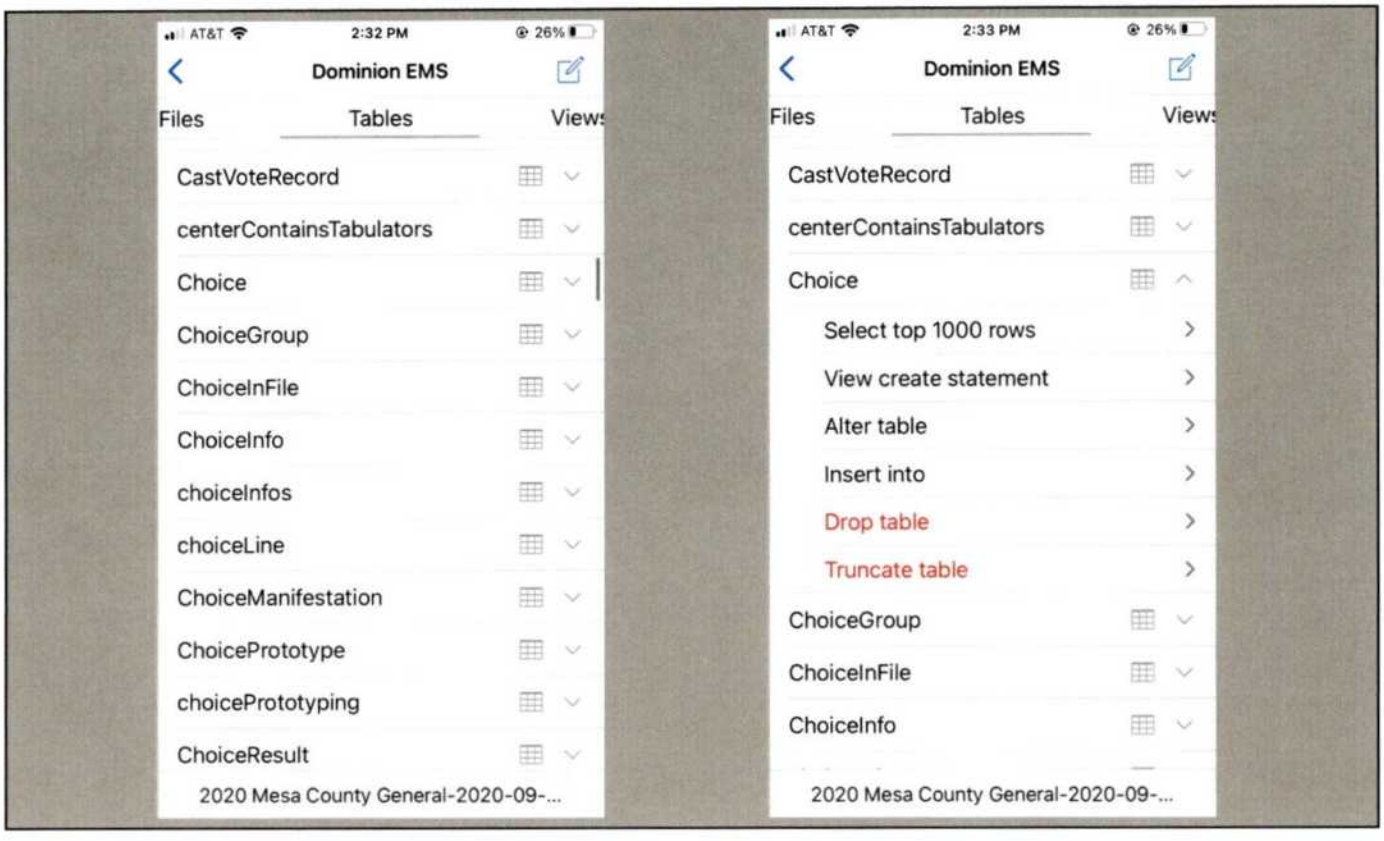


Figure 59 - Database Table Listing

On the left, one sees the same 'Choice' table as was seen on the EMS server and Test Workstation (where it was called 'dbo.Choice'). On the right, 'Choice' table is selected resulting in the options as shown. I selected 'Select top 1000 rows'.

Note that the "drop table" command would delete the table entirely, while the "truncate table" command would shorten the table, and if applied to a table containing actual vote data, would delete some of those votes.

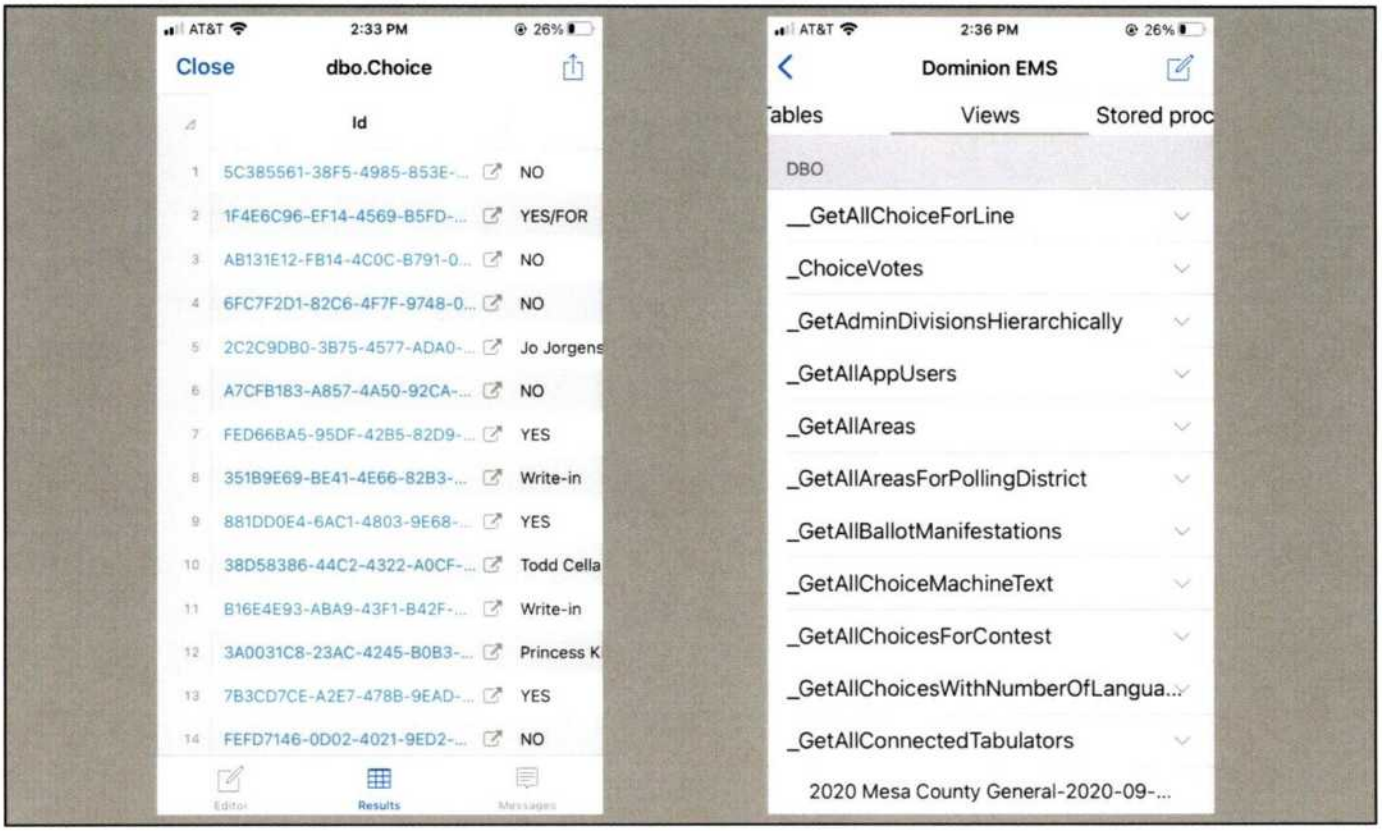


Figure 60 - Database Access

On the left, one sees the top 177 rows in the 2020 Mesa County General database, along with the choices listed as shown by both the EMS server and the Test Workstation. One the right, 'Views' at the top menu was then selected to pull up the database views from the EMS server.

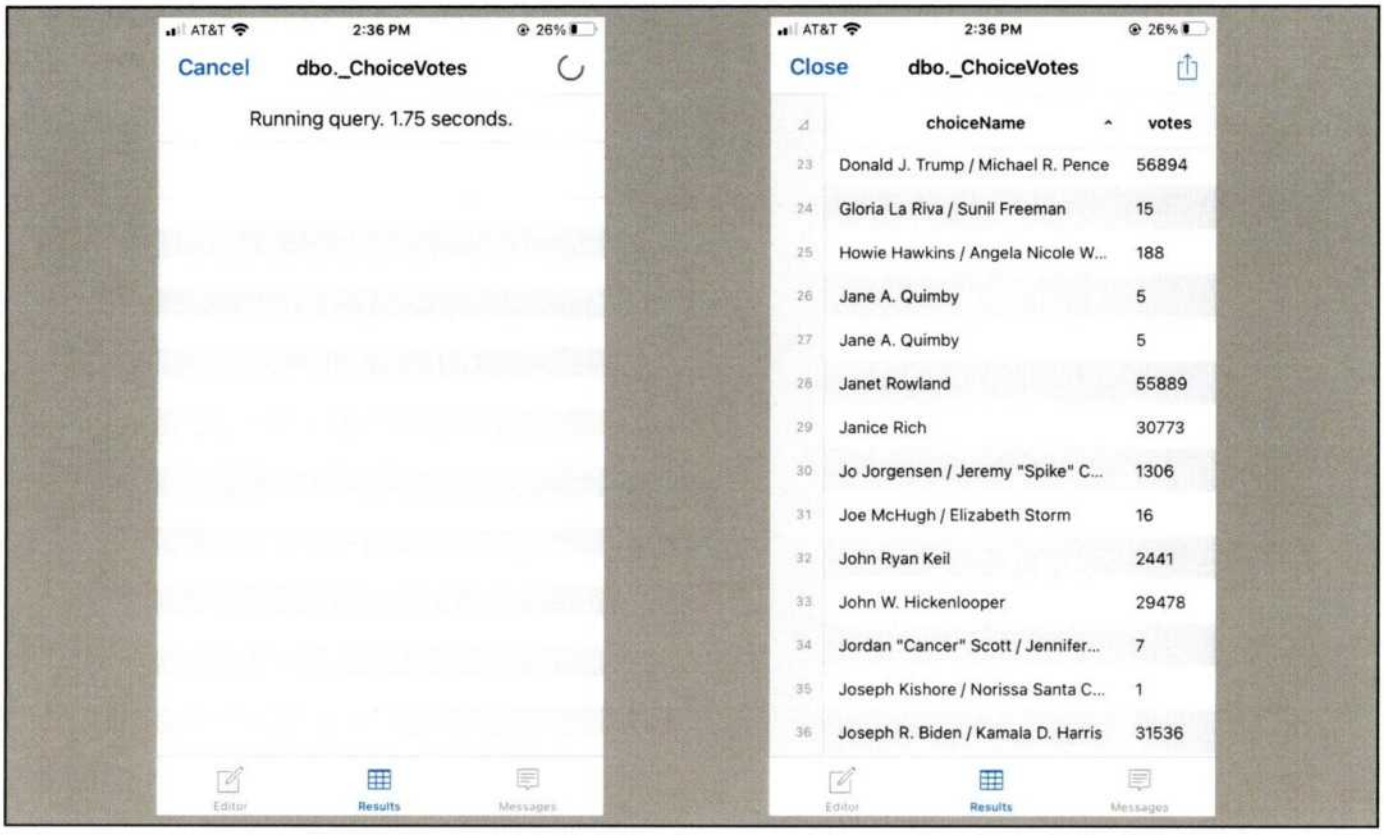


Figure 61 - Executing a Database Query

The _ChoiceVotes view was selected. On the left, one sees that it took 1.75 seconds to pull up all the votes for each choice in the election. The result of that query is shown on the right.

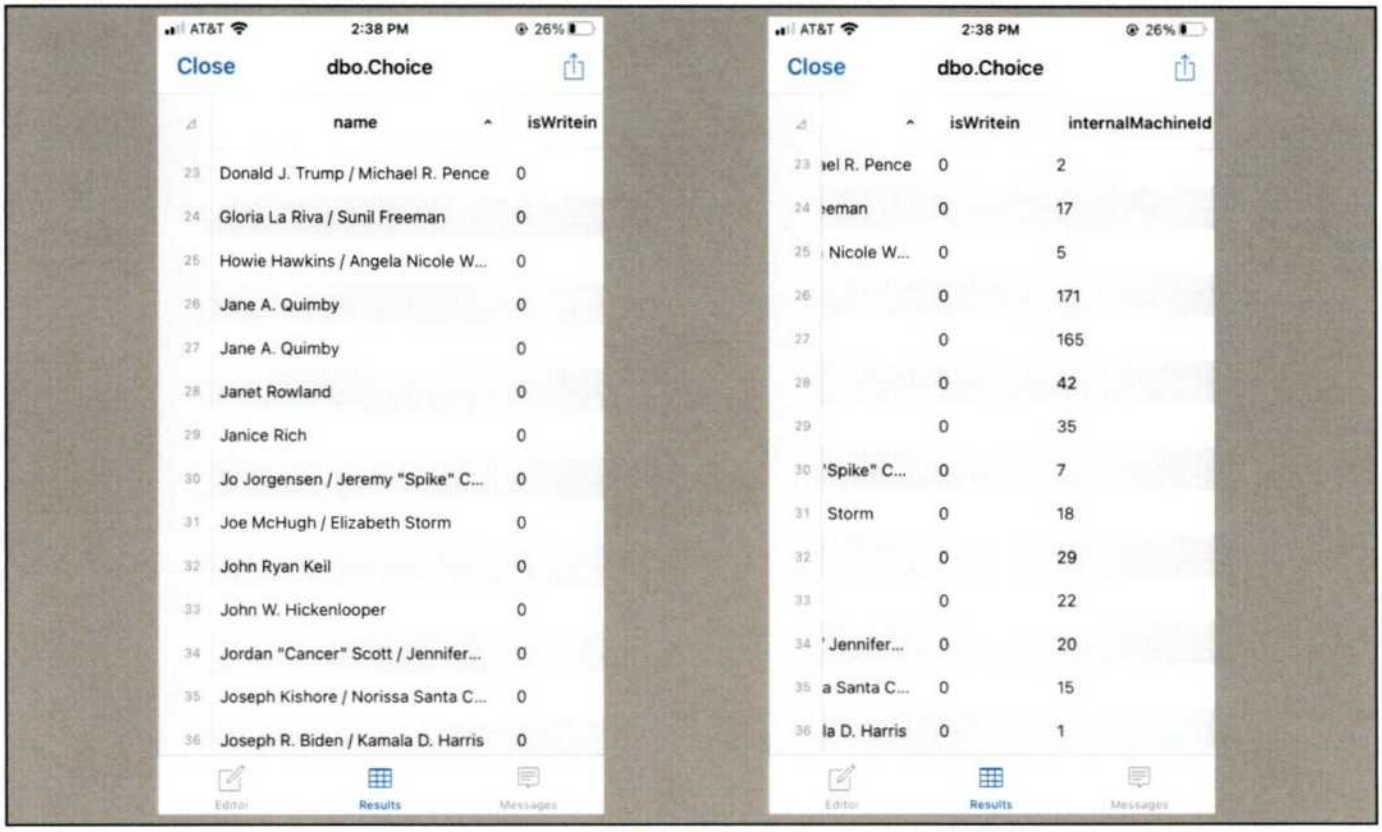


Figure 62 - Table Data

The left and right images demonstrate the effect of scrolling to the right, to display all the columns. All the columns in this table can be viewed without being denied or without any type of warning.

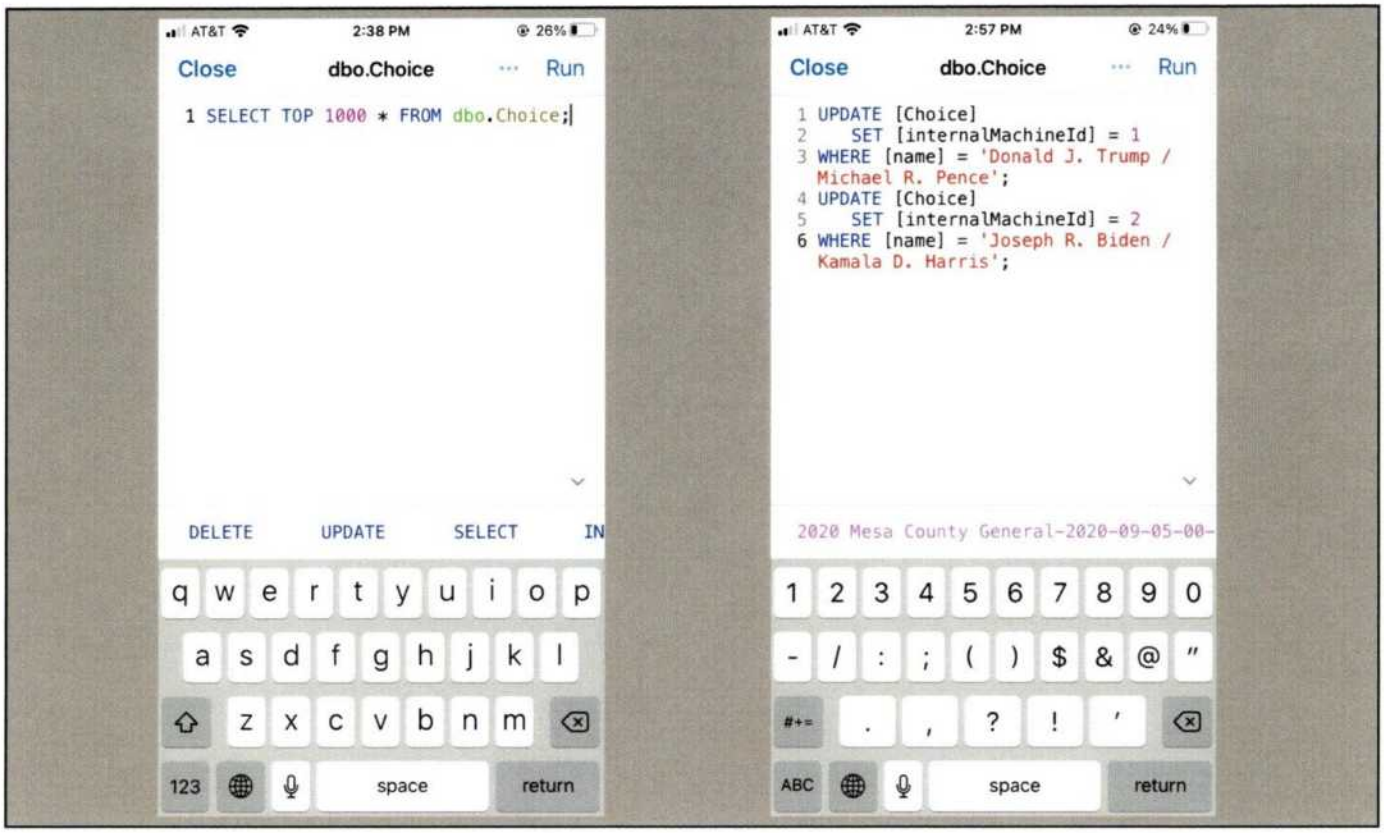


Figure 63 - A script to change the vote data

The left image shows the default query that asks for the SQL Server to send the top 1000 rows from the dbo.Choice table. The instructions on the image on the right were then typed in. What they do is very simple: They update the Choice table by setting the internalMachineId to '1' for 'Donald J. Trump / Michael R. Pence,' and setting the internalMachineId to '2' for the entry with 'Joseph R. Biden / Kamala D. Harris' in it. This is the same type of change that was made by hand on both the EMS server and the Test Workstation earlier in this report.

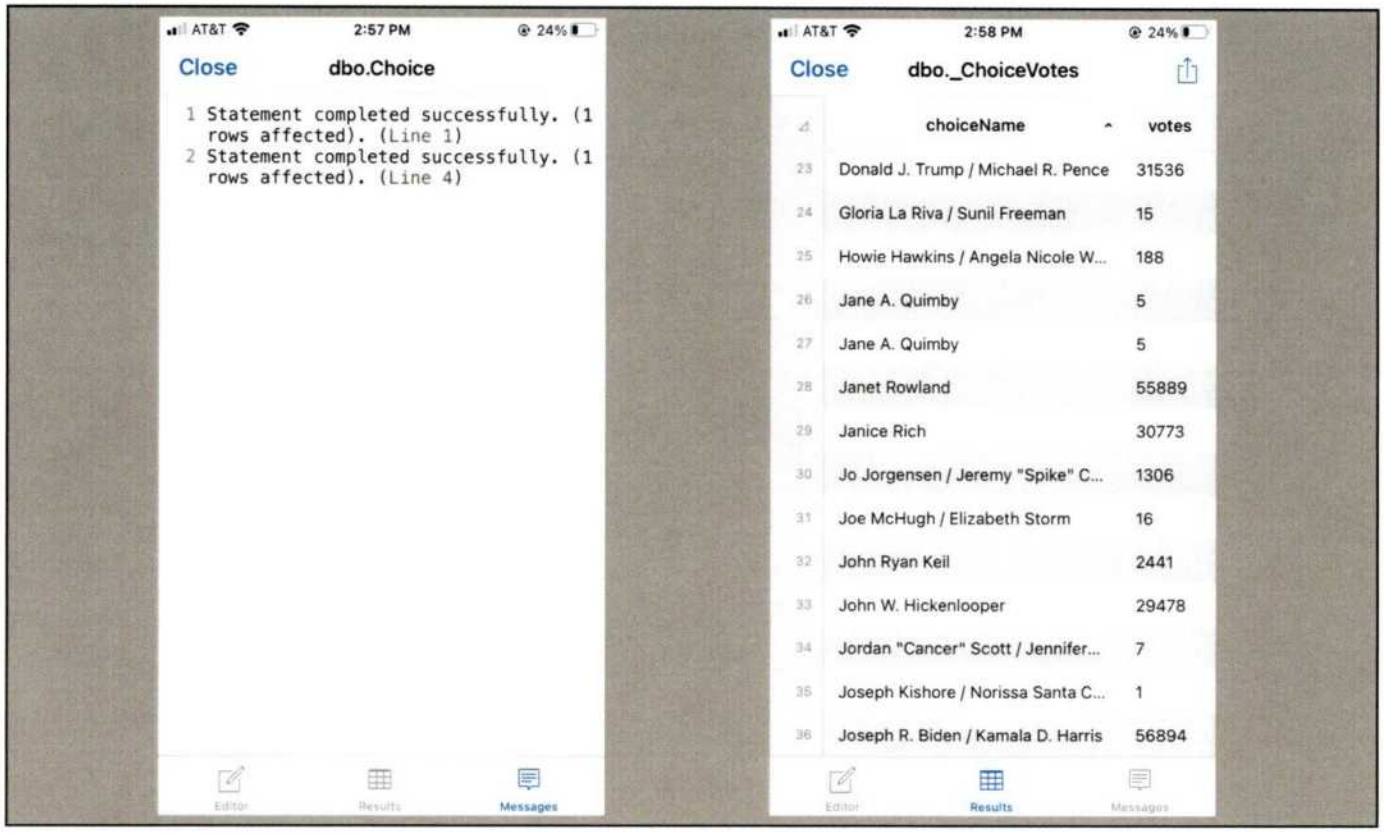


Figure 64 - Script Results

The image on the left shows the typed instructions were executed and the EMS server reported that each instruction was completed successfully, affecting one row each. On the right, the _ChoiceVotes view is run again to see that once again the election results were flipped from Trump to Biden, using a basic iPhone with an app downloaded from the App Store that anyone could install and use.

EXAMINATION RESULT 3:

The calculated vote totals in an EMS server database can be altered by any person using the more limited capabilities of a mobile device wirelessly connected to the EMS server network.

For the iPhone test, while a wireless device was added to the network to allow this demonstration to occur, it's alarming that's all it took to accomplish this, especially since thirty-six devices in the Mesa DVS hardware had wireless cards installed. Anyone could purchase a wireless device like this online or at most computer or office supply stores, attach it inside the voting center, and use one of the easy to guess or well-known passwords on the system (or obtain it from the Darkweb,[72] or access the iDRAC remote control server, or use DVS-published default passwords, etc.), could sit out in the parking lot and change any part of the database before, during, or after an election. More dangerous, since thirty-six devices in the DVS D-Suite System were configured with a wireless card, the same abuse could be committed by someone with basic computer networking skills,[73] given wireless access to the EMS server is completely insecure, exposed to access, protected by only a Windows password, despite many additional protections being available. As an example, a Dell Wireless 1560 internal wireless adapter *was* identified in the specified configuration on the DVS D-Suite ImageCast Voter Activation (ICVA) computer that is part of the Mesa County DVS D-Suite system. A skilled individual could easily get away with this same unauthorized access and much more with almost any modern cell Phone, iPhone or Android, Mac, or PC. Wireless capability is very small today, easily fitting inside a small USB device, which could even be inserted in an internal port, invisible to County officials, allowing for the surreptitious connection of the capability in such a manner that only highly trained specialists would be able to find it. Figure 65 depicts such a miniaturized wireless USB device, which could be installed without notice on a motherboard of the type used by D-Suite EMS servers (shown).

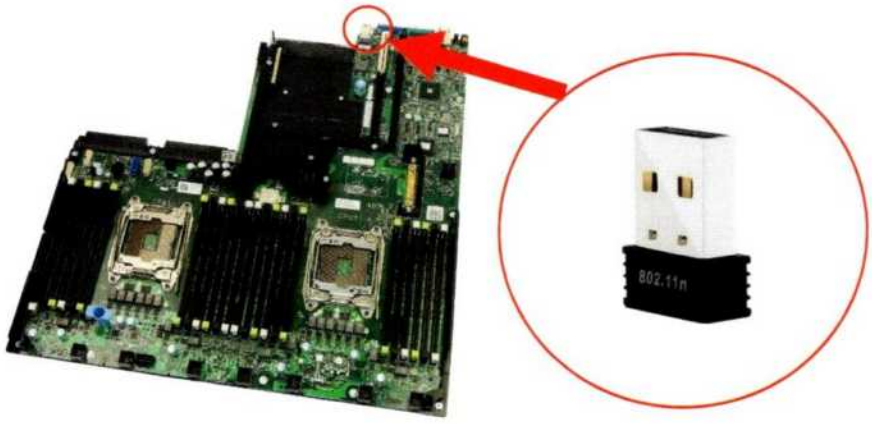


Figure 65 - Small Wireless Device Surreptitiously Installed (internally) on a Computer Motherboard

The result of this examination demonstrates that an attack is possible using a wireless device connected in any one of a multitude of ways. It was possible to perform network scanning using industry standard tools on a common Apple iPhone.

---

[72] The Darkweb is a clandestine, encrypted, anonymized webserver infrastructure characterized by extensive criminal activity including trafficking in computer access credentials (passwords) as well as many other criminal activities. Access to the Darkweb is only available through use of The Onion Router (TOR) which hides and renders untraceable (to most searches and searchers) the IP address and location of its users. Content on the Darkweb is hidden from the general Internet to facilitate criminal activity.

[73] https://papers.mathyvanhoef.com/ccs2017.pdf, http://www.krackattacks.org/

It must be noted that the methods used here are well described in publicly-available, commonly-known literature. Cybersecurity industry guides [74] describe the Nmap application specifically to identify connections to the network. Nmap uses the ICMP 'echo request' command, just as our previous check did using the 'ping' command. Nmap is capable of executing many echo requests in parallel to more rapidly identify the devices connected to a network than the single-request ping program can, and Nmap tests all of the port numbers configured to be tested, for each IP address being tested.

The report in this examination does not reveal any secret tradecraft, or compromise election security; the techniques used in this examination are common among IT professionals. Unfortunately, there is very little security in this voting system, to begin with.

This examination demonstrates how the use of wireless networking can be easily exploited and documents the risk presented using one example. Given the ease with which it can be implemented if wireless devices are enabled (e.g., by an accidental button press on a laptop), it is important to acknowledge the risk so that future elections can be properly protected. To assure integrity of the infrastructure, computing devices with wireless network capability must not be used because wireless networking can be easily enabled by accident (or maliciously). Additionally, certification under VSS absolutely required steps to have been taken within the voting system design and implementation that secure the system from accidental or malicious connections to other networks. The fact that these steps were not taken casts doubt on the credibility and competence of the vendor, the certification authority, the certification testing lab, and the institutions responsible for the testing lab accreditation program.

Making use of the broadband modem inside the cell phone, it may be possible to create a connection from the internet directly into the electronic voting system, bypassing all County firewalls and security, allowing someone to command and control it from anywhere in the world.

This would be completely undetectable by election officials, and most, if not all forensic experts.

While critics may assert that it has not yet been proven that any wireless device was connected to the Mesa County systems and operating prior, during, or after the election, the fact is that wireless devices were installed in Mesa County DVS systems, and critics cannot prove those devices were not operating and exploited. The required compliance standards were created explicitly to provide such proof, yet the features that enable compliance were disabled. Due to the illegal disabling of logging mechanisms, configured overwriting of logs, and the failure to preserve the log data (in violation of the law) that would either show tampering and fraud or support claims of the integrity of the election, it cannot be proven that the election was free of intrusion and tampering (See Report #1).

[74] https://attack.mitre.org/techniques/T1046/

CONCLUSION

An ongoing forensic examination of the Mesa County EMS server, version 5.11-CO, provided by DVS revealed the overwriting of critical log data and election records, the misconfiguration of logging functions, and the failure to preserve required election records in Report #1.

In this Report #2, the examination has conclusively shown and demonstrated the ability to access election records from a separate computer, not part of the DVS D-Suite system, the ability to edit the election database, and the ability to change calculated vote totals to alter the election results on the Mesa County EMS server entirely, "flipping" the winner of an election contest in the jurisdiction from one candidate to another.

The Key Objectives for this report were answered by this examination:

1. To determine whether D-Suite-implemented security requirements comply with the 2002 Voting System Standards (VSS)
    a. Uncertified software was used on the system rendering the certification of the entire system and all elections conducted with it, Invalid.
    b. Security protections required by law were almost completely absent
        i. Other than a userID and an easily guessed or bypassed password, no authentication was required
        ii. The firewall rule for access to the election database, ballots and results was unrestricted to any IP address in the world
        iii. Together with the firewall rule, Microsoft SQL Server Management Studio (SSMS) enabled complete access to the entire election databases – not just to the 2020 election but to the elections of June 2019 through May 25, 2021.
        iv. A self-signed encryption certificate was used introducing the potential for a man-in-the-middle attack
        v. Thirty-five wireless devices (802.11, Wi-Fi) were installed inside election equipment and an additional wireless device was identified in a connected printer
        vi. Any or all of these wireless devices could have connected to the Internet via the building wireless facilities
        vii. "Purging" (deletion) of critical Audit Log data, as specified by DVS and directed by the Secretary of State[75], destroyed all records of connection to the Internet or elsewhere, all record of user activity, including programs run by these users, errors, and any record of the addition or deletion of votes and the alteration of election results.
    c. **EACH of the compliance failures identified in 1.a. and 1.b. above are clear violations of the law.**

[75] The TDP associated with the "trusted build" process is promulgated by the Secretary of State. CRS 1-5-620 States that the vendor provides manuals and documentation and that any information not on file with and approved by the Secretary of State shall not be used in an election.

2. To determine whether the results of an election, stored on the EMS server, can be altered by any person with physical access to the logged-in EMS server,
    a. **Any person with physical access to the logged-in EMS server can change the calculated vote totals on the EMS server.**
3. To determine whether the results of an election stored on the EMS server, can be altered by any person using even a non-Dominion computer directly or indirectly connected to the EMS server network.
    a. **Any person using even a non-DVS computer directly or indirectly connected to the EMS server network can change the calculated vote totals on the EMS server.**
4. To determine whether the results of an election stored on the EMS server, can be altered by any person using a device such as a cell phone wirelessly connected to the EMS server network.
    a. **Any person using a device such as a cell phone wirelessly connected to the EMS server network can change the calculated vote totals on the EMS server.**

Examination of wireless vulnerability required that a wireless device be connected to the EMS server network and demonstrated that such a device when connected is capable of allowing uncontrolled access to and alteration of an election database on the EMS server.

The purpose and the finding of Key Objective 4 demonstrates that if such a wireless device were connected to the EMS server network, the election results can be accessed and altered surreptitiously. The ease with which wireless technology can be enabled, even by accident, presents an unacceptable risk to critical infrastructure voting systems, especially when combined with the egregious violations of the VSS and the multiple security failures found in this examination. **Wireless encryption is easy to break,[76] has been broken, documented and demonstrated online.[77]**

The disabling and mis-configuration of numerous security measures as found in this Examination renders this EMS election system unsafe and utterly insecure. Unauthorized software, multiple violations of VSS and consequently Colorado law and the use of an un-accredited testing laboratory made the certification of this system, and its subsequent use in elections, illegal.

The on-going examination found that security provisions on the election equipment were not restricted by IP address but rather the firewall configuration was programmed to allow any IP address from anywhere in the entire World to access the election records with no more than a single and relatively simple password to protect it.

There is nothing secret or novel about the techniques used to demonstrate direct access, access by a non-DVS computer or iPhone access to the election databases. Software accessible to hundreds of millions of people and openly advertised for free download and use was used to demonstrate the extreme insecurity of the voting system.

---

[76] http://cve.mitre.org/, supra note 18

[77] https://papers.mathyvanhoef.com/ccs2017.pdf, http://www.krackattacks.org/

The reason for the insecure configuration of these critical infrastructure-designated voting systems, in contradiction to the vendor's claims[78] and the Secretary of State's certification, should be determined through appropriate investigation.

The law requires the retention of election records including system logs but this election system is grossly out of compliance with the law. Combined with the overwriting of log files, the systematic disabling of critical logging and numerous security elements disabled or bypassed, creating a "back-door" for malicious actors, this configuration of the Mesa County, Colorado voting system assures that may not be possible to prove the integrity of any election in which this equipment was used. This voting system is not compliant with the law, should never have been used in an election, and cannot be trusted to provide authenticated, reliable election data in any election.

Nearly every point of examination has revealed the most serious deficiencies in both security and configuration.

The claim that "election systems were not connected to the Internet" has been made, however the use of removable media devices, presence of wireless networking components within DVS components, use of the internet for election results reporting and other functions, and the destruction of and non-retention of critical logs prevent the verification that the system was not connected to the internet. The configuration of logging to ensure overwriting of log data resulted in operating system logs not being retained that may have shown any improper activity, had it occurred. Because of this it is not possible, on the basis of election systems log files (that are required to be retained), to prove election tampering or election integrity.

This failure of the voting system to retain log files that could prove election integrity is a most serious violation of certification requirements. The voting system, having not met election certification requirements, could not have been legally authorized for use in an election.

This report has detailed the following critical discoveries in Mesa County's voting system:

- **Uncertified software installed, rendering the voting system unlawful for use in elections.**
- **Does not meet statutorily mandated Voting System Standards (VSS) and could not have been lawfully certified for purchase or use.**
- **Suffered systematic deletion of election records (audit log files required by Federal and State law to be generated and maintained), which, in combination with other issues revealed in this report, creates an unauditable "back door" into the election system.**
- **Violates Voting Systems Standards ("VSS") which expressly mandate prevention of the ability to "change calculated vote totals." This report documents this non-compliance from the logged-in EMS server, from a non-DVS computer with network access, and from a cell phone (which may be possible if any of the 36 internal wireless devices in voting system components are deliberately or accidentally enabled and a password is obtained).**
- **Mandatory VSS "System auditability" required features are disabled.**
- **Is configured with 36 wireless devices, which represent an extreme and unnecessary vulnerability, and which may be exploited to obtain unauthorized access from external devices, networks, and the Internet.**

[78] See Appendix A. Compliance Requirements.

- **Is configured through firewall settings to allow any computer in the world to connect to the EMS server.**
- **Uses only a Windows password with generic userIDs to restrict and control access.**
- **contains user accounts with administrative access that share passwords, subverting VSS-required user accountability and action traceability controls.**
- **Uses a self-signed encryption certificate which exposes the system to the risk of undetected compromise or alteration.**

This report does not compromise state secrets or election integrity – that has already been done by these multiple violations of law, multiple failures of the vendor, the Voting System Testing Lab and the Secretary of State's improper certification. Nation-state adversaries already know these vulnerabilities exist; it is only the American people that are unaware. No new vulnerabilities are discovered or disclosed in this report; all of them are previously well known in the industry and to professionals.

Immediately pending elections and the complete lack of election integrity presented by this voting system present an extreme danger to our constitutional republic. With elections beginning on a large scale very soon, with the massive security vulnerabilities, the weakness presented by this uncertifiable Voting System, the abject failure of the Voting System Testing Laboratory with expired accreditation and lack of proper oversight by authorities, remediation of these issues before pending elections is not possible.

This DVS election system has been shown non-compliant with the law and has been shown to be uncertifiable. The use of this system in an election was itself a breach of law, and more importantly a breach of public trust with reckless disregard for the right of a free people to choose their government.

APPENDIX A. COMPLIANCE REQUIREMENTS

Standards for election systems are provided by the Federal Election Commission Voting Systems Standards (VSS) and in Colorado, compliance is required with this standard.

The VSS requires access control to prevent or detect access to election systems, ensure that system functions are executable only in the intended manner and order, provide safeguards to prevent tampering, record and report the date and time of normal and abnormal events, maintain a permanent record of all audit data that cannot be modified or overridden, detect and record every event including an error condition that the system cannot overcome, time-dependent or programmed events that occur without the intervention of the voter or a polling place operator, and to protect the system from intentional manipulation and fraud, among many other requirements.

Federal Election Commission 2002 Voting Systems Standards (VSS)

Specific compliance requirements from the 2002 Voting Systems Standards (VSS) documentation are excerpted in this section. The Standards are contained in 2 volumes which together are several hundred pages long, and are published on the Federal Election Commission website as two PDF documents.

Excerpts in this Appendix are cited by VSS Volume, Section and Page number for reference in the first line of each box, followed by text of the VSS. Discussion of these standards follows outside each text box as appropriate.

APPLICABILITY

VSS V1, 1.6, page 1-13:
The Standards apply to all system hardware, software, telecommunications, and documentation intended for use to:

- Prepare the voting system for use in an election;
- Produce the appropriate ballot formats;
- Test that the voting system and ballot materials have been properly prepared and are ready for use;
- Record and count votes;
- Consolidate and report votes;
- Display results on-site or remotely; and
- Maintain and produce all audit information.

In general, the Standards define functional requirements and performance characteristics that can be assessed by a series of defined tests. Standards are mandatory requirements and are designated by use of the term "shall".

All of these functional requirements are important. In this report we focus on aspects of recording and counting votes. Determination of whether the election management system performed with the accuracy and integrity required by these standards requires the audit information be maintained and preserved in accordance with law. The VSS is applicable the DVS D-Suite systems examined and reported upon in this document and in Report #1.

VSS V1, 2.1, page 2-19:

This section contains standards detailing the functional capabilities required of a voting system.

[...]

- Overall Capabilities: These functional capabilities apply throughout the election process. They include Security, accuracy, integrity, system auditability, election management system, vote tabulation, ballot counters, telecommunication and data retention.

The VSS is written specifying capabilities required at a high level. Detailed implementation methods are not specified but it is clear, for example, that these topics are not to be ignored.

VSS V1, 2.2, page 2-20:

This section defines required functional capabilities that are system-wide in nature and not unique to pre-voting, voting, and post-voting operations. All voting systems shall provide the following functional capabilities:

- Security;
- Accuracy;
- Error Recovery;
- Integrity;
- System auditability;
- Election management system;
- Accessibility;
- Vote tabulating;

The emphasis on all of these functional capabilities together indicates the serious nature of the requirement in this standard. The declaration by the U.S. Government that these systems are part of the national critical infrastructure further reinforces the importance of these capabilities. "Shall provide" indicates the mandatory nature of the requirement. The implementation of a functional security capability does not mean to apply the weakest possible implementation of security, for example.

DATA RETENTION

VSS V1, 2.2.11, page 2-34:

United States Code Title 42, Sections 1974 through 1974e, states that election administrators shall preserve for 22 months "all records and paper that come into (their) possession relating to an application, registration, payment of poll tax, or other act requisite to voting." This retention requirement applies to systems that will be used at anytime for voting of candidates for Federal offices (e.g., Member of Congress, United States Senator, and/or Presidential Elector). Therefore, all systems shall provide for maintaining the integrity of voting and audit data during an election and for a period of 22 months thereafter.

[...]

The appropriate state or local authority must preserve all records that may be relevant to the detection and prosecution of federal civil rights or election crimes for the 22-month federal retention period, if the records were generated in connection with an election that was held in whole or in part to select federal candidates.

This requirement is clear. In discussion of retention of "all records that come into their possession" the burden of understanding what a record is, falls on election administrators. In particular this standard specifies that state or local authority must perform the preservation of all records.

Election Record Definition, Scope and Content

VSS V1, 4.4.3, page 4-84:
In-process audit records document system operations during diagnostic routines and the casting and tallying of ballots. At a minimum, the in-process audit records shall contain:

a. Machine generated error and exception messages to demonstrate successful recovery. Examples include, but are not necessarily limited to:
   1) The source and disposition of system interrupts resulting in entry into exception handling routines;
   2) All messages generated by exception handlers;
   3) The identification code and number of occurrences for each hardware and software error or failure;
   4) Notification of system login or access errors, file access errors, and physical violations of security as they occur, and a summary record of these events after processing.

Other exception events such as power failures, failure of critical hardware component, data transmission errors, or other type of operating anomaly;

Documenting computer interrupts is a very detailed requirement, from a computer science perspective it is considered extreme. In normal operation, logs of computer activity typically do not include this level of detail unless the generation of records (logging) is set to the most verbose level for software debugging,

because the volume of log data generated can be extreme. The specification that these records are generated during diagnostic routines as well as during the counting and tallying of the vote, in the same sentence, is illuminating and indicates that the intention of the VSS is that this most extreme level of record be generated especially in the 4th example listed in this standard.

It is instructive to note that this standard specifically enumerates these requirements within the definition of a record, rather than in the section that specifically addresses security:

- System login;
- System access errors;
- File access errors; and
- Physical violations of security as they occur,

One reason that file access errors are included in this definition is that programming and operational errors can result in the creation of errors in stored data (that manifest in file access errors). Another reason is that intruders were well known at the time this standard was written and before, to attempt to destroy evidence of their activities by deleting audit trail records that might tend to incriminate them. Title 18, Sec. 1030 makes unauthorized access to such a computer system a felony.

In other election cases such as the Antrim, Michigan case it is notable that while records of previous elections were preserved and still on the election system, the audit records from the 2020 election were missing; the fact that records were generated and preserved previously but suddenly stopped during a specific event where malfeasance is suspected is significant and indicative of the practice by intruders to delete any record of their activity.

Astronomer Cliff Stoll became famous as an early computer crime investigator and published a book entitled "The Cuckoo's Egg" in which he recognized that computers don't make mistakes – programmers do. As a consequence, he looked at the very records regarding exception handling and errors that are required in this standard, because accounting software on the computer he managed as a grad student reported a 25-cent error in accounting data. Cliff's curiosity and persistence resulted in the discovery of a computer attack where the intruder tried to delete audit records that resulted in the error. The investigation ultimately revealed international espionage and attacks against the US Government that would have gone unnoticed without his analytical search for what he initially assumed was a programming error. As a pattern of evidentiary finding, this history is very useful in understanding computer crime and criminal behavior.

This inclusion of these security-specific requirements in this basic but over-arching definition indicates their importance and that the intent of the standard is for great detail in the generation of these specific security audit records.

Security Requirements for Voting Systems

VSS V1, 6.1, page 6-93:

[...]

Ultimately, the objectives of the security standards for voting systems are:

- To establish and maintain controls that can ensure that accidents, inadvertent mistakes, and errors are minimized;
- To protect the system from intentional manipulation and fraud, and from malicious mischief;
- To identify fraudulent or erroneous changes to the system; and
- To protect secrecy in the voting process.

The Standards are intended to address a broad range of risks to the integrity of a voting system. While it is not possible to identify all potential risks, the Standards identify several types of risk that must be addressed by a voting system. These include:

- Unauthorized changes to system capabilities for:
  - Defining ballot formats;
  - Casting and recording votes;
  - Calculating vote totals consistent with defined ballot formats; and
  - Reporting vote totals;
- Alteration of voting system audit trails;
- Changing, or preventing the recording of, a vote;
- Introducing data for a vote not cast by a registered voter;
- Changing calculated vote totals;
- Preventing access to vote data, including individual votes and vote totals, to unauthorized individuals; and
- Preventing access to voter identification data and data for votes cast by the voter such that an individual can determine the content of specific votes cast by the voter.

This standard is also clear. The first three bullets in the list of objectives are related as previously explained, because intentional manipulation, fraud, malicious mischief and fraudulent or erroneous changes to the system often manifest in records that appear initially to have been accidents, inadvertent mistakes and errors.

The failure of security identified in this report specifically permitted unauthorized changes to the recording of votes in a database, as components of the database that should have been protected were allowed to be altered. A more difficult to find alteration might involve the changing of ballot formats so that a vote for one candidate appeared as a vote for a different candidate, but the access granted by the failure of security access controls allowed full administrative access to the database. The changing of calculated vote totals was specifically demonstrated by the tests in this examination. The data values changed essentially mean "Trump's votes are stored here -> X" and "Biden's votes are here -> Y" and the test switched X and Y.

As presented in Report #1, audit trails were altered (deleted) because the specifically enumerated risk was not addressed as required by this standard.

VSS V1, 6.2, page 6-96:
Access controls are procedures and system capabilities that detect or limit access to system components in order to guard against loss of system integrity, availability confidentiality and accountability. Access controls provide reasonable assurance that system resources such as data files, application programs, and computer-related facilities and equipment are protected against unauthorized operation, modification, disclosure, loss, or impairment. Unauthorized operations include modification of compiled or interpreted code, run-time alteration of flow control logic or of data, and abstraction of raw or processed voting data in any form other than a standard output report by an authorized operator.

Access control capability was built into the EMS server operating system and into the SQL DBMS but not programmed to be secure and one most egregious finding was that the EMS server was specifically configured to be insecure in defiance to the requirements in this standard and every known industry, government and security best practice, the standards of the National Institute of Standards and Technology (which chaired the committee that produced the VSS), and the DoD Security Technology Implementation Guides.

VSS V1, 6.2.2, page 6-97:
Vendors shall provide a detailed description of all access control measures designed to permit authorized access to the system and prevent unauthorized access. Examples include:

a. Use of data and user authorization;
b. Program unit ownership and other regional boundaries;
c. One-end or two-end port protection devices;
d. Security kernels;
e. Computer-generated password keys;
f. Special protocols;
g. Message encryption; and
h. Controlled access security.

Vendors shall also define and provide a detailed description of the methods used to prevent unauthorized access to the access control capabilities of the system itself.

This standard requires a detailed description to be provided by the voting system vendor, but clearly expects these functional protections to be implemented if the measures are to be documented.

DVS stated on their website[79] that they are compliant with voting systems standards, including the Voluntary Voting Systems Guidelines (VVSG) as shown in Figure 66. A review of the VSTL test-related documents reveals that the standards tested against were the VVSG standards. By comparing the test plans and reports to the requirements in the VVSG, this is easily assessed.

Figure 66 - DVS Compliance Statement

The Voluntary Voting Systems Guidelines (VVSG) contain even more explicit and precise definitions of the logging required than do the VSS, and although these are Guidelines that are not explicitly required under Colorado law, DVS makes the claim on their website that they are compliant with them. The 2005 VVSG were a defacto standard for the security of election systems and have been revised several times. The 2005 VVSG specifically requires in section 2.1.5.1 that a number of safeguards and operational requirements be applied. The VVSG excerpt below is *only a small partial list of those requirements*, but for this examination, the finding of key compliance issues is noted in Red following each requirement:

a. Voting system equipment shall record activities through an event logging mechanism.
FAIL. Log mechanism does exist and records some, but not all activities, even though it overwrites and destroys those records frequently. Logging is not only incomplete but is wholly inadequate.
b. Voting system equipment shall enable file integrity protection for stored log files as part of the default configuration.
FAIL. Not only have log files not been preserved, but they have been overwritten as indicated in Report #1. Further, the log file size has been set to a very small limit such that the log data is NOT preserved and cannot be recovered historically. Integrity Protection for these log files is not implemented.
c. The voting system equipment logs shall not contain information that, if published, would violate ballot secrecy or voter privacy or that would compromise voting system security in any way.
FAIL. The log files that remain contain very little information of value in determining the integrity of the election at all; no information was found in the logs that can violate the secrecy of ballots or voter privacy, or that would compromise voting system security, but critical Audit Log data has been deleted (overwritten and in some cases its collection disabled) that is required for an Audit of the system's security, integrity, accuracy, that would identify errors, malicious actions, illegal tampering with ballots and vote totals, intrusions, what programs

[79] This statement was present on Dominion Voting Systems' website in September, 2020 and has since been removed, however the claim that they comply with voluntary VVSG standards brings this into relevance.

were run, by whom, and their results. Contrary to the law, this is not in compliance – it is just the opposite: voting system security is compromised by the inability to detect malicious activity.

d. The voting system equipment shall log at a minimum the following data characteristics for each type of event: 1) system ID; 2) unique event ID and/or type; 3) timestamp; 4) success or failure of event, if applicable; 5) User ID trigger the event, if applicable; 6) Resources requested, if applicable.
FAIL. The EMS system does not record this information and in most cases has been configured by the Manufacturer to not log this information.

e. Voting system equipment shall log all events, including abnormal events.
FAIL. The disabling of logging and the overwriting of log files above a certain size prevent the logging of all events.

f. Voting system equipment shall ensure that event logging cannot be disabled. Voting system equipment shall implement default settings for secure log management activities, including log generation, transmission, storage, analysis and disposal.
FAIL. The design and configuration of this voting system provides exactly the opposite. Logging has been disabled by design and by the misconfiguration of the operating system such that the required and necessary records are NOT stored.

g. Voting system equipment shall log clearing of logs and log rotation.
FAIL. The EMS system does not log the clearing of logs or log rotation, nor the overwriting of files (an act of "clearing the logs"). No record of log rotation could be found. In Report #1, the vendor DVS not only overwrote the operating systems and all log data with its "Trusted Build" installation, it designed the installation process to re-format and re-partition the hard disk ensuring that this occurred.

Of particular importance are sections b, d, e, f and g above. Had they been implemented properly and in accordance with the standards as Dominion claims and Customers expect, these log data would have supported conclusions regarding the integrity or the lack of integrity of the election. In both Antrim and Maricopa investigations, the DVS software did not log each modification to each record. Per the VSS, this detail of logging should be not only performed, but retained for 22 months (25 months in Colorado).

Even the Center for Internet Security (CIS) recognizes the need for these controls, among many others, in their Handbook for Election Infrastructure Security.[80]

Given the failure to implement these required and recommended controls, the DVS Democracy Suite version 5.11-CO as provided to the State of Colorado **does not possess the required integrity controls as claimed by DVS and required by law. From the evidence presented in this report, this failure of integrity safeguards means that elections held in Colorado using this equipment do not possess the integrity to protect the vote from tampering, or to record access to or modification of the vote.**

[80] https://www.cisecurity.org/wp-content/uploads/2018/02/CIS-Elections-eBook-15-Feb.pdf

APPENDIX B. DATABASE FUNDAMENTALS

This report addresses computerized databases. This Appendix provides a basic understanding of the terms and technology involved to support the reader's understanding of findings in this report.

The voting systems used in Mesa County, Colorado are made by DVS. Many of these voting systems are comprised of an industry-standard computer that uses a Microsoft operating system and Dominion application software that provides a foundation for election related functions including capturing and storing the election data in a database management system, tabulating and counting the vote.

The Mesa County Election Management System (EMS) server runs on the Microsoft Windows Server 2016 operating system, and it employs a database management system known as Microsoft SQL Server. The security of the system depends largely upon the proper configuration of the Operating System and the SQL Server.

There are several types of databases, including relational, non-relational, and object-oriented databases. This discussion will be limited to relational databases because this is the type of database used in the Dominion voting system that is the subject of this forensic examination.

Microsoft SQL Server is a Database Management System (DBMS). A DBMS can contain many databases. Within this Mesa County, Colorado EMS server DBMS are many databases from prior elections, in addition to the 2020 General Election. Each database consists of many tables that can have different purposes. Some are administrative (access permissions for example), some are necessary for the DBMS to function (such as the database of databases, necessary because a DBMS can have multiple databases), and some have operational content related to the purpose of the database. This information is contained in multiple Tables consisting of multiple columns (and multiple rows if not empty). The database of databases (referred to as the DBDB) identifies the users, access permissions, the identity of each table that is contained within each respective database, among other items.

The fundamental components of a relational database are Tables, Rows and Columns. Data are organized in tables. Columns within a table contain specific data types, for example, first name, last name, street address, city, state, etc. Rows within a table each contain an instance of the data, referred to in database science as a tuple. The database is called a relational database because the various tables are *Related* by what is known as a KEY value. The Key value exists in multiple tables and is the item that links or *relates* the data in one table to the data in another table. For example, in a voter database, one reasonable KEY value might be Ballot Number – it would exist in all the associated tables and it becomes possible to retrieve ALL the data about a particular ballot by searching for every row where ballot number equals, for example, 300.

One primary purpose of a database is to return data in response to a request for that data, called a Query. One of the most common computer languages used in modern relational databases is the Structured Query Language (SQL). Structured query language is intended to be readable and understood easily.

An example of an SQL-like query might be to find the address of a person in a database table called "Addresses". Such a query might look like:

> RETRIEVE(Addresses) address.street.address, address.city, address.state where first.name="John" and last.name="Smith"

IF the database table has an entry for John Smith, the above query would return the Street Address, City and State for him, *provided that* the user of this database had permission to read this specific Addresses table. While there is a specific order (syntax) for the components of the database command (e.g., a format), the commands are not difficult to understand, and the example here, while similar, is simplified to make it very understandable.

A DBMS implemented in software known as *Microsoft SQL Server* is addressed in this document because it is the DBMS installed and used in the Mesa County Colorado Election Management System server. The function of a DBMS is to organize its tables and rows, to provide a very granular set of permissions to the users of the database, to provide the integrity of the data – specifically to ensure that data cannot be inappropriately altered or deleted, and to control the four basic functions of the database. Four basic functions are implemented in relational databases, with respect to the data contained in its table-rows. Those basic functions are read, write, modify, and delete. The DBMS also supports various types of calculations based on the data in its tables.

One of the features of a DBMS is to very granularly control the permissions within a database. For example, a user might have permission to change the street address within a row, but not be allowed to change the city or state. Normal computer system permissions *without* a DBMS give the user permission to access the entire data set (for example, within a spreadsheet). Thus, the permission settings (e.g., configuration) of a DBMS are critical to its proper functioning and the ability to maintain integrity of the data within the database. These permission settings control who can perform which transactions.

Permissions within a well-controlled database specify which users can read which tables, which users can add data to the table, which users can modify (or update) data in the table, and which users can delete data in a table. Most commonly only the DBMS administrator has all four permissions for any table. It is common for an average user to be able to read and perhaps add data to a table, while changing or deleting data requires a supervisor to perform the task. A computer program (or task) can be assigned permission in the same manner that a user can be, sometimes by creating a user-id that is used only by the program.

There are special tables within a database that are highly restricted. These special tables include the DBDB within the DBMS, the User table within each database, the permissions for each user to each database and database table, and in some cases, the permissions for each user to the columns within each table. These special tables define how the DBMS operates.

It is required that a particular user within a DBMS only be able to alter the data with good reason. One example might be the case of a changed vote. Let's consider, for example, a hand-marked ballot, for simplicity, identified as ballot #300. The identity of the voter is not associated with the ballot number so it is accessed only by its number. The ballot contains circles or squares to be marked to indicate a vote. However, if the ballot marking is not within the lines (within limits), the ballot is marked for adjudication so that a human can then take steps to determine the voter's intent and then store that entry in the database. In this example, the original vote (and the photographic image of the ballot) might be stored in a database table called PendingAdjudication (the table name is an example to illustrate the technology). The Adjudication user should be able to read the data in the PendingAdjudication table, but not change or delete it. The user looks at the ballot image and makes their determination of voter intent and the results are written to a separate table called AdjudicatedVote. The user then has permission to change the value of ONE COLUMN within the PendingAdjudication table (for the specific data row) to indicate Adjudicated or

NotAdjudicated. The point of the example is that even in this case, the original data is not deleted, and a separate database table is used to compile the data. The Adjudication user in this example NEVER changes the original data, but the vote that is counted is in the AdjudicatedVote table. Thus, an audit of the complete voter database should show that there is one, and only one, entry for ballot #300, and the decision of the auditor should be available for review and the actions taken should be traceable. A more complex design may even use a separate table all-together to track which items are adjudicated or not.

The design of the database must make sense. In the example above, if the Adjudicator were to be permitted to change the original vote in the PendingAdjudication table, the ability to review their decision would be lost and there would be no way to audit the change, without seeing the before- and after- results. Thus, not only must the configuration of permissions enable those necessary changes but it must protect the integrity of the data and support the ability of the system to be auditable.

There is much not discussed here. For example, the DBMS in a voting application would be expected to check the PendingAdjudication table to make sure that every ballot that was sent to be adjudicated HAD BEEN processed, and that there were no rows with NotAdjudicated remaining, before the tabulation and count of votes had been finalized.

The design of the database and its permissions are only part of the logic required to make such a system work properly. As with the check above to ensure that all votes were adjudicated, there is much additional logic, which should be found within the database processing workflow, to ensure the proper calculations and integrity are maintained throughout the entire voting process.

APPENDIX C. IP ADDRESSING FUNDAMENTALS

There are two versions of Internet Protocol addressing seen in this data. The legacy version of addressing is expressed by four one- to three-digit numbers separated by periods – "X.X.X.X," where X is an 8-bit number (e.g., has a value of 0-255). Because industry and users throughout the world have exceeded the number of available address numbers, a new address scheme was developed. The legacy address scheme is known as IP version 4 (IPv4) and is 32 binary bits long, while the new scheme is known as IP version 6 (IPv6) and is 128 binary bits long, represented as 8 groups of 4 hexadecimal values (0-9 and A-F) separated by periods (A.B.C.D.E.F.G.H). This solves the problem of running out of IPv4 addresses and provides, by one estimate, more than 1,500 IP addresses for every square meter of Earth's surface. This explanation is provided because both types of addresses are present in this forensic analysis and it is necessary for the reader to understand the data being presented.

In Figure 8, IP2 shows the IPv4 address 192.168.100.10, the address assigned to be used by the Mesa County EMS server. IP1 shows the IPv6 address FE80::792B:3E74:DF1B:C565%5. This translates to FE80:0000:0000:0000:792B:3E74:DF1B:C565 (the double colon stands for repeated 0 address values), and "zone" 5 (%5) which is essentially the identifier that indicates which IP Network Interface Card (NIC) the address is tied to. While these data reflect the interface capability of the Oracle VirtualBox environment, the IP Address 192.168.100.10 is configured in the stored operating system and when launched here, automatically assumed the same IP address. IPv6 is addressed here for completeness.

The IPv4 address used (192.168.x.x) is a "Private Network" address per Internet Standard RFC-1918 and is NOT directly routable across the Internet. However, firewalls, routers and other network devices use a service called Network Address Translation (NAT) or Port Address Translation (PAT) to convert these private addresses to publicly routable addresses and allow them to be transmitted over the larger Internet. Thus, the use of a private network address assigned to a particular Ethernet interface does not in itself, prevent the computer from accessing the Internet – it becomes necessary to examine all routers, firewalls and other networking equipment to determine whether the computer is capable of *direct* connection to the Internet via a translation mechanism such as NAT or PAT.

For every IPv4 address, the number is split into two parts – the first part of the number is the Network Address and the second part of the number is the Device Address. This is defined by the number of bits assigned to the network address and follows the IP address and a slash "/." "192.168.100.0/24" indicates the first 24 bits of this binary number constitutes the Network Address and the remaining 8 bits constitute the Device Address. This set of Device Addresses is referred to as a Subnet. For data to leave a subnet, the subnet must have a Default Gateway assigned. When a computing device sends data to an address that is outside the Subnet group of addresses, it sends that data to the Default Gateway address which then *routes* the data onward to its destination.

There are two special Device Addresses: the first value in the Device Address is used to specify the Network Address while the last address in the subnet range is defined as a Broadcast Address and is used to send data to every device in the Subnet. In the address example "192.168.100.0/24," the first address is 0 and is the Network Address is 255; a broadcast to all 254 device addresses possible on this subnet would be sent to "192.168.100.255." The first usable address of this subnet is "192.168.100.1," which is typically used for the Default Gateway address.

The IPv6 address used (FE80:x:x:x:x:x:x:x) is a *link-local* address, which means that it is also not routable across the Internet. The concept of NAT and PAT are not used in IPv6, *with the single exception of using it to translate IPv6 addresses to IPv4 addresses and vice versa* because not all network equipment is capable of using IPv6 (yet). Some legacy network equipment widely in use today is not capable of transporting IPv6.

This link-local (FE80) address is not routable and is not supposed to be translatable from IPv6 to IPv4 and vice versa, however this depends on whether a particular network device vendor has followed the standard when implementing their software. While most vendors have designed their devices properly (network devices would not work properly otherwise), from a scientific and evidentiary perspective, it still remains necessary to forensically examine all connected network devices to ensure that these addresses cannot reach the Internet.

APPENDIX D. NATION-STATE CYBER ATTACK CAPABILITIES

Introduction

The mere idea of advanced Nation-State cyberwarfare capabilities at first blush seems like fantasy straight out of a James Bond film. Yet these attack capabilities are the most sophisticated on the planet. Most countries, including the USA, consider their defensive and offensive cyberwar capabilities to be highly classified. In the USA these are implemented by the National Security Agency, specifically in its Tailored Access Operations (TAO) group according to numerous reports, and in the UK, by the CGHQ. In this appendix, a short synopsis (and bibliography) of several of the more sophisticated cyberattacks are presented, *in particular in support of statements made elsewhere in this document* –specifically, that attacks occur *extremely quickly,* that a USB Thumb Drive can be infected with malicious software which can then infect other computer systems, and that cyberattacks can cause considerable damage. This is a very small sampling of some of the more sophisticated attacks but is illustrative of the advanced sophistication and the pervasive nature of vulnerabilities.

Security experts in the USA also understand and have documented issues with Voting Systems security, in *this report* https://archive.org/details/6432002-Voting-Village-Report-defcon27/page/n15/mode/2up. This security conference (Defcon 2019) is often billed as a "hacker" conference, however some of the most renowned security professionals in the world attend it, and the "Voting Village" at Defcon, in the referenced report, is co-chaired by Matt Blaze, Professor of Law and McDevitt Chair for the Department of Computer Science, Georgetown University (and author of many books on the subject). Christopher Krebs, Director of the US Critical Infrastructure Security Agency (CISA) also attended.

In 1984, while working at Bell Telephone Laboratories, I witnessed one of the very first destructive computer viruses. In that era, computer monitors used standard NTSC television signals to present video on a large cathode ray tube "tv screen". The monitor used a very high voltage (tens of thousands of volts) to cause the electron beam to display a picture. To generate the high voltage, the monitor used a "flyback" transformer, a specific type of high-frequency transformer commonly found in televisions, that took advantage of the 15,575 hertz horizontal scan signal that is part of the NTSC standard video signal. This signal was amplified and fed the primary winding of the transformer. It was found that the video driver circuit card in primitive 'PCs' of that era allowed the frequency of the horizontal scan signal to be programmed. When that frequency was programmed to 0 hertz, the electric current through the primary winding of the transformer changed from a rapidly varying signal to a constant "on" state. Since this state exceeded the capability of the transformer, it burned the transformer out, destroying the monitor.

In 2007, DHS and the Idaho National Laboratory ran the Aurora Generator Test to demonstrate vulnerabilities in the electric power grid in the USA.[81] A leaked video [82] of the attack is widely available on the Internet and shows the complete destruction of a 27 ton, 2.25MW generator by a cyberattack. In this attack, the attackers (part of the US Military) opened the relays of the generator (by remote computer control) long enough for the generator to slip out of synchronization with the power grid, and then reconnected the relays, causing a catastrophic mechanical jolt to the generator. This is the equivalent of driving your car at 70 mph, and while moving at that speed, placing your car into reverse gear. They did

[81] https://en.wikipedia.org/wiki/Aurora_Generator_Test
[82] https://youtu.be/LM8kLaJ2NDU

this three times, as is apparent from the video. The third time was "the charm" as the generator's diesel engine self-destructs and the room as well as the external exhaust pipe fill with black smoke. The article cited[83] includes both the video of the test showing destruction of the generator as well as the original DHS report, released under FOIA.

Adversaries constantly scan and probe every computer on the internet to identify weakness well in advance of the need for an attack. A commercial (i.e., unclassified) example of this scanning is demonstrated by the company Lumeta. During the first Gulf War, noted security expert Bill Cheswick, co-founder of Lumeta, used a common troubleshooting tool (ping) and was able to perform real-time battle-damage assessment by detecting computers that went offline due to active bombing campaigns. Adversaries have discovered their targets well in advance and have pre-programmed attacks ready to launch.

Moonlight Maze

"Moonlight Maze was a 1999 US government investigation into a massive data breach of classified information. It started in 1996 and affected NASA, the Pentagon, military contractors, civilian academics, the DOE, and numerous other American government agencies. By the end of 1999, the Moonlight Maze task force was composed of forty specialists from law enforcement, military, and government. The investigators claimed that if all the information stolen was printed out and stacked, it would be three times the height of the Washington Monument, which is 555 ft (169 m) tall. The Russian government was blamed for the attacks, although there was initially little hard evidence to back up the US accusations besides a Russian IP address that was traced to the hack. Moonlight Maze represents one of the first widely known cyber espionage campaigns in world history. It was even classified as an Advanced Persistent Threat (a very serious designation for stealthy computer network threat actors, typically a nation state or state-sponsored group) after two years of constant assault. Although Moonlight Maze was regarded as an isolated attack for many years, unrelated investigations revealed that the threat actor involved in the attack continued to be active and employ similar methods until as recently as 2016."[84]

Stuxnet

Stuxnet was an offensive operation, believed to be conducted by the USA and Israel,[85] to destroy nuclear enrichment centrifuges at Iran's Natanz enrichment facility,[86] About 1,000 centrifuges were involved in the enrichment of 'yellow cake' uranium from "fuel grade" for commercial power reactors to "weapons grade" to create nuclear weapons (bombs/missiles).

"Stuxnet was a 500-kilobyte computer worm that infected the software of at least 14 industrial sites in Iran. This worm was an unprecedentedly masterful and malicious piece of code that attacked in three phases. First, it targeted Microsoft Windows machines and networks, repeatedly replicating itself. Then it sought out Siemens Step7 software, which is also Windows-based and used to program industrial control systems that operate equipment, such as centrifuges. Finally, it compromised the programmable logic controllers. The worm's authors could thus spy on the industrial systems and even cause the fast-spinning centrifuges to tear themselves apart, unbeknownst to the human operators at the plant."

---

[83] https://www.muckrock.com/news/archives/2016/nov/14/aurora-generator-test-homeland-security/
[84] https://en.wikipedia.org/wiki/Moonlight_Maze
[85] https://www.jpost.com/International/Snowden-US-Israel-created-virus-to-destroy-Iran-nukes-319226
[86] https://www.wired.com/2010/11/stuxnet-sabotage-centrifuges/

"Stuxnet could spread stealthily between computers running Windows—even those not connected to the Internet. If a worker stuck a USB thumb drive into an infected machine, Stuxnet could "worm" its way onto it, then spread onto the next machine that read that USB drive. Because someone could unsuspectingly infect a machine this way, letting the worm proliferate over local area networks, experts feared that the malware had perhaps gone wild across the world."

"In October 2012, U.S. defense secretary Leon Panetta warned that the United States was vulnerable to a "cyber–Pearl Harbor" that could derail trains, poison water supplies, and cripple power grids. The next month, Chevron confirmed the speculation by becoming the first U.S. corporation to admit that Stuxnet had spread across its machines." [87]

Operation Titan-Rain

Titan Rain was a series of coordinated computer attacks[88] on the United States that began in 2003 and originated from Guangdong, China. The attacks are believed to have come from the People's Liberation Army unit 61398, located at the Lingshui Signals Intelligence Unit on Hainan Island, one of China's largest military facilities in the South China Sea. This is the same unit responsible for the attack on the Wall Street Journal, which cyber forensics company Mandiant identified as APT-1 (Advanced Persistent Threat–1)[89].

"An **advanced persistent threat (APT)** is a stealthy threat actor, typically a nation state or state-sponsored group, which gains unauthorized access to a computer network and remains undetected for an extended period. In recent times, the term may also refer to non-state-sponsored groups conducting large-scale targeted intrusions for specific goals."[90]

Titan Rain is rumored to have stolen as much as 40 Terabytes of US Government secrets. This attack persisted for many years.

Operation Aurora

Operation Aurora was conducted by the People's Liberation Army of China from mid-2009 through December, 2009.[91]

It was a very large scale attack that affected numerous commercial entities including Google, Morgan-Stanley, Adobe Systems, Akamai Technologies, Juniper Networks, and Rackspace who have publicly confirmed that they were targeted. According to reports, Yahoo, Symantec, Northrop Grumman, Morgan Stanley, and Dow Chemical were also among the targets. The unit which conducted the attack has been named APT-17.

"The attack was named 'Operation Aurora' by Dmitri Alperovitch, Vice President of Threat Research at cybersecurity company McAfee. Research by McAfee Labs discovered that 'Aurora' was part of the file path on the attacker's machine that was included in two of the malware binaries McAfee said were associated

---

[87] https://spectrum.ieee.org/the-real-story-of-stuxnet
[88] https://en.wikipedia.org/wiki/Titan_Rain
[89] https://www.lawfareblog.com/mandiant-report-apt1
[90] https://en.wikipedia.org/wiki/Advanced_persistent_threat
[91] https://en.wikipedia.org/wiki/Operation_Aurora

with the attack. "We believe the name was the internal name the attacker(s) gave to this operation," McAfee Chief Technology Officer George Kurtz said in a blog post."

"According to McAfee, the primary goal of the attack was to gain access to and potentially modify source code repositories at these high-tech, security, and defense contractor companies. '[The software configuration management systems] were wide open,' says Alperovitch. 'No one ever thought about securing them, yet these were the crown jewels of most of these companies in many ways—much more valuable than any financial or personally identifiable data that they may have and spend so much time and effort protecting.' "

2020 US Government Attack

In 2020, a massive nation-state attack against many companies and US Government organizations took place.[92] Initially only the Treasury department and the NTIA were thought to have been attacked. But it turned out that many of the US Government operations including the IRS and even the US Administrative Office of the Courts (which relies heavily on the software SolarWinds) were compromised.

This attack was a supply-chain attack. SolarWinds, a network management system, as many software firms do, periodically releases updates to its software. SolarWinds was broken into and one of its update programs was infected with malware. Because SolarWinds was inappropriately assigned too much trust by its customers, their software updates were white-listed (allowed through the firewall, unchallenged). The attack was in the update.

This is widely regarded as one of the worst attacks in US history for the length of time it lasted (9 months) before detection as well as the impact it had upon affected organizations.

Summary

Nation-States including Russia, China, North Korea, Malaysia, Iran and many others seek to attack the USA's national security, economic, industrial, communications, and financial systems. These attackers are extremely sophisticated and well trained. For example, North Korea has an institute in Pyongyang that teaches cyberwarfare and has been turning out more than 100 graduates every month for well over 15 years. Other Nation-States, including Iran, have sent students to North Korea's school.

This brief history has documented the sophistication of advanced cybersecurity attacks.

Multiple references show that sophisticated attacks can occur by transfer through USB drives, without being detected by the end user.

This history shows how unprotected system configurations have enabled advanced cyberattacks, and how software updates can infiltrate a company's IT operations and take control.

[92] https://en.wikipedia.org/wiki/2020_United_States_federal_government_data_breach.

APPENDIX E. SECURITY CONSIDERATIONS FOR SQL SERVER INSTALLATIONS

The following information was taken directly from Microsoft documentation and is provided here to be a reference to basic security considerations related to installations of Microsoft SQL Server. This is relevant as Microsoft SQL Server is the product used in the Dominion EMS.

From Microsoft SQL Server Documentation:

Security is important for every product and every business. By following simple best practices, many security vulnerabilities can be avoided. Below are some security best practices that should be considered both before installing SQL Server and after SQL Server has been installed. Security guidance for specific features is included in Microsoft reference articles for those features.

Before Installing SQL Server:

- Follow these best practices when setting up the server environment:
- Enhance physical security
- Use firewalls
- Isolate services
- Configure a secure file system
- Disable NetBIOS and server message block

Details about these items are provided below.

Enhance Physical Security

Physical and logical isolation make up the foundation of SQL Server security. To enhance the physical security of the SQL Server installation, do the following tasks:

- Place the server in a room accessible only to authorized persons.
- Place computers that host a database in a physically protected location, ideally a locked computer room with monitored flood detection and fire detection or suppression systems.
- Install databases in the secure zone of the corporate intranet and do not connect your SQL Servers directly to the Internet.
- Back up all data regularly and secure the backups in an off-site location.

Use Firewalls

Firewalls are important to help secure the SQL Server installation. Firewalls will be most effective by following these guidelines:

- Put a firewall between the server and the Internet. Enable your firewall. If your firewall is turned off, turn it on. If your firewall is turned on, do not turn it off.
- Divide the network into security zones separated by firewalls. Block all traffic, and then selectively admit only what is required.
- In a multi-tier environment, use multiple firewalls to create screened subnets.
- When you are installing the server inside a Windows domain, configure interior firewalls to allow Windows Authentication.

Isolate Services

Isolating services reduces the risk that one compromised service could be used to compromise others. To isolate services, consider the following guidelines:

- Run separate SQL Server services under separate Windows accounts. Whenever possible, use separate, low-rights Windows or Local user accounts for each SQL Server service.

Configure a Secure File System

Using the correct file system increases security. For SQL Server installations, you should do the following tasks:

- Use the NTFS file system (NTFS). NTFS is the preferred file system for installations of SQL Server because it is more stable and recoverable than FAT file systems. NTFS also enables security options like file and directory access control lists (ACLs) and Encrypting File System (EFS) file encryption. During installation, SQL Server will set appropriate ACLs on registry keys and files if it detects NTFS. These permissions should not be changed. Future releases of SQL Server might not support installation on computers with FAT file systems.
- Use a redundant array of independent disks (RAID) for critical data files.

Disable NetBIOS and Server Message Block

Servers in the perimeter network should have all unnecessary protocols disabled, including NetBIOS and server message block (SMB).

NetBIOS uses the following ports:

- UDP/137 (NetBIOS name service)
- UDP/138 (NetBIOS datagram service)
- TCP/139 (NetBIOS session service)

SMB uses the following ports:

- TCP/139
- TCP/445

During or After Installation of SQL Server

After installation, you can enhance the security of the SQL Server installation by following these best practices regarding accounts and authentication modes:

Service accounts

- Run SQL Server services by using the lowest possible permissions.
- Associate SQL Server services with low privileged Windows local user accounts, or domain user accounts.

Authentication mode

- Require Windows Authentication for connections to SQL Server.
- Use Kerberos authentication.

Strong passwords

- Always assign a strong password to the sa [system administrator] account.
- Always enable password policy checking for password strength and expiration.
- Always use strong passwords for all SQL Server logins.

References:

https://docs.microsoft.com/en-us/sql/sql-server/install/security-considerations-for-a-sql-server-installation?view=sql-server-ver15

https://docs.microsoft.com/en-us/sql/sql-server/install/security-considerations-for-a-sql-server-installation?view=sql-server-2016

APPENDIX F. C.R.S. 1-5-608.5

1-5-608.5. Electronic and electromechanical voting systems - testing by federally accredited labs - certification and approval of purchasing of electronic and electromechanical voting systems by secretary of state - conditions of use by secretary of state - testing.

(1) A federally accredited laboratory may test, approve, and qualify electronic and electromechanical voting systems for sale and use in the state of Colorado.

(2) (Deleted by amendment, L. 2009, (HB 09-1335), ch. 260, p. 1190, § 4, effective May 15, 2009.)

(3)

(a) If the electronic and electromechanical voting systems tested pursuant to this section satisfy the requirements of this part 6, the secretary of state shall certify such systems and approve the purchase, installation, and use of such systems by political subdivisions and establish standards for certification.

(b) The secretary of state may promulgate conditions of use in connection with the use by political subdivisions of electronic and electromechanical voting systems as may be appropriate to mitigate deficiencies identified in the certification process.

(c) In undertaking the certification required by this section, the secretary of state may consider either procedures used or adopted by county clerk and recorders or best practices recommended by equipment vendors.

(3.5)

(a) [Editor's note: Subsection (3.5) is effective July 1, 2022.] On and after December 31, 2022, if an electronic and electromechanical voting system tested pursuant to this section satisfies the requirements of this part 6 related to the use of the system in an election using instant runoff voting and the rules established by the secretary of state pursuant to section 1-5-616 (1.5), the secretary of state shall certify such system and approve the purchase, installation, and use of such system by political subdivisions in an election using instant runoff voting.

(b) The secretary of state may promulgate conditions of use in connection with the use by political subdivisions of an electronic and electromechanical voting system in an election using instant runoff voting as may be appropriate to mitigate deficiencies identified in the certification process.

(c) In undertaking the certification required by this section, the secretary of state may consider procedures used or adopted by county clerk and recorders or best practices recommended by equipment vendors.

(4) In undertaking the certification required by this section, the secretary of state may request a federally accredited laboratory to undertake the testing of an electronic or electromechanical voting system or may use and rely upon the testing of an electronic or electromechanical voting system already performed by another state or a federally accredited laboratory upon satisfaction of the following conditions:

(a) The secretary of state has complete access to any documentation, data, reports, or similar information on which the other state or laboratory relied in performing its testing and will make such information available to the public subject to any redaction required by law; and

(b) The secretary of state makes written findings and certifies that he or she reviewed the information specified in paragraph (a) of this subsection (4) and determines that the testing:

(I) Was conducted in accordance with appropriate engineering standards in use as of the time the testing is undertaken; and

(II) Satisfies the requirements of sections 1-5-615 and 1-5-616 and all rules promulgated thereunder.

(5) In undertaking the certification required by this section, the secretary of state may conduct joint testing with an agency of another state or with a federally accredited laboratory.

History

Source: L. 93:Entire section added, p. 1414, § 57, effective July 1. L. 2004:Entire section amended, p. 1346, § 13, effective May 28. L. 2009:Entire section amended,(HB 09-1335), ch. 260, p. 1190, § 4, effective May 15. L. 2021:(3.5) added,(HB 21-1071), ch. 367, p. 2416, § 3, effective July 1, 2022.

Research References & Practice Aids

Hierarchy Notes:

C.R.S. Title 1

C.R.S. Title 1, Art. 5

State Notes

Research References & Practice Aids

Cross references:

For the legislative declaration contained in the 2004 act amending this section, see section 1 of chapter 334, Session Laws of Colorado 2004.

APPENDIX G. C.R.S. 1-5-615

1-5-615. Electronic and electromechanical voting systems - requirements.

(1) The secretary of state shall not certify any electronic or electromechanical voting system unless such system:

(a) Provides for voting in secrecy;

(b) Permits each elector to vote for all offices for which the elector is lawfully entitled to vote and no others, to vote for as many candidates for an office as the elector is entitled to vote for, and to vote for or against any ballot question or ballot issue on which the elector is entitled to vote;

(c) Permits each elector to verify his or her votes privately and independently before the ballot is cast;

(d) Permits each elector privately and independently to change the ballot or correct any error before the ballot is cast, including by voting a replacement ballot if the elector is otherwise unable to change the ballot or correct an error;

(e) If the elector overvotes:

(I) Notifies the elector before the ballot is cast that the elector has overvoted;

(II) Notifies the elector before the vote is cast that an overvote for any office, ballot question, or ballot issue will not be counted; and

(III) Gives the elector the opportunity to correct the ballot before the ballot is cast;

(f) Does not record a vote for any office, ballot question, or ballot issue that is overvoted on a ballot cast by an elector;

(g) For electronic and electromechanical voting systems using ballot cards, accepts an overvoted or undervoted ballot if the elector chooses to cast the ballot, but it does not record a vote for any office, ballot question, or ballot issue that has been overvoted;

(h) In a primary election, permits each elector to vote only for a candidate seeking nomination by the political party with which the elector is affiliated;

(i) In a presidential election, permits each elector to vote by a single operation for all presidential electors of a pair of candidates for president and vice president;

(j) Does not use a device for the piercing of ballots by the elector;

(k) Provides a method for write-in voting;

(l) Counts votes correctly;

(m) Can tabulate the total number of votes for each candidate for each office and the total number of votes for and against each ballot question and ballot issue for the polling location;

(n) Can tabulate votes from ballots of different political parties at the same voter service and polling center in a primary election;

(o) Can automatically produce vote totals for the polling location in printed form; and

(p) Saves and produces the records necessary to audit the operation of the electronic or electromechanical voting system, including a permanent paper record with a manual audit capacity.

(1.5) [Editor's note: Subsection (1.5) is effective July 1, 2022.] The secretary of state shall not certify any electronic or electromechanical voting system for use in an election using instant runoff voting unless, in addition to meeting the requirements of subsection (1) of this section, the system meets the requirements and performs the functions required by section 1-7-1003.

(2) The permanent paper record produced by the electronic or electromechanical voting system shall be available as an official record for any recount conducted for any election in which the system was used.

History

Source: L. 2004:Entire section added, p. 1347, § 14, effective May 28. L. 2013:IP(1), (1)(m), (1)(n), and (1)(o) amended,(HB 13-1303), ch. 185, p. 713, § 49, effective May 10. L. 2021:(1.5) added,(HB 21-1071), ch. 367, p. 2417, § 6, effective July 1, 2022.

Research References & Practice Aids

Hierarchy Notes:

C.R.S. Title 1

C.R.S. Title 1, Art. 5

State Notes

Research References & Practice Aids

Cross references:

(1) For the legislative declaration contained in the 2004 act enacting this section, see section 1 of chapter 334, Session Laws of Colorado 2004.

(2) In 2013, the introductory portion to subsection (1) and subsections (1)(m), (1)(n), and (1)(o) were amended by the "Voter Access and Modernized Elections Act". For the short title and the legislative declaration, see sections 1 and 2 of chapter 185, Session Laws of Colorado 2013.

APPENDIX H. MAN IN THE MIDDLE ATTACK

In Figure 10, an encryption certificate is not visible. This is due to the fact that an encryption certificate had not been created and assigned. This alone does not indicate the lack of a security encryption certificate, because SQL Server will create a self-signed certificate automatically, as it has done in this case. However, self-signed certificates are known to be insecure and susceptible to common man-in-the-middle attacks. On a voting system, where security should be paramount, this is wholly irresponsible at best.

Despite the direct connection to the back-end of the SQL server is set to be encrypted even in this sub-par fashion, any device with Microsoft SQL Server Management Studio or any other SQL Server client installed that supports the Windows Authentication method can connect to the server provided they have some type of connection (directly or indirectly) to any part of the voting system network, can find the server IP address, a userID and a password. Microsoft SQL Server Management Studio is a free download from Microsoft and does not require any special licensing – anyone can obtain it and use it without restriction. There are also many other SQL Clients that exist for Windows, OS X, iPhone, Android, and others, many that are free to download and use.

The SQL Server Management Studio (SSMS) software used on the Expert's client computer was downloaded directly from Microsoft, and that Expert's client computer had no prior encryption configuration, encryption keys or certificates containing encryption keys – the only things supplied to make the connection to the EMS server were a userID, password, and the IP address of the server.

Detail:

A "Man-In-The-Middle" attack (MITM) is an attack where an eavesdropper intercepts a communication between two parties, and makes each party believe he (or she) is the person they intended to communicate with by impersonating them.

In Figure 67 below, Person A would normally communicate with Person B directly. The attack involves intercepting the communication and impersonating the other party as illustrated by the red arrows and Person C.

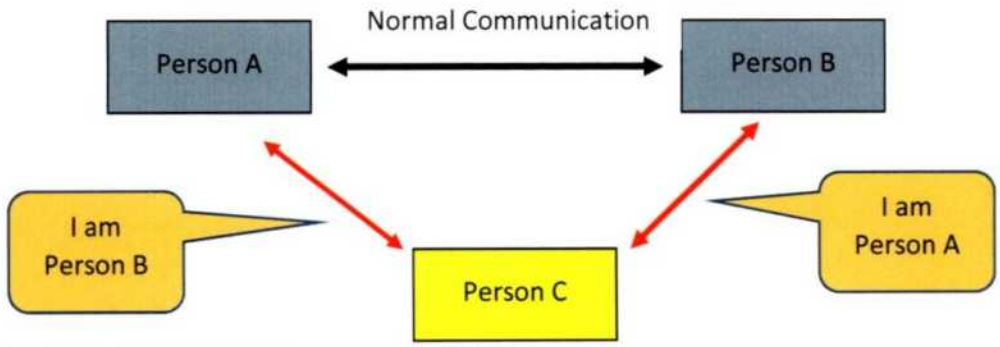


Figure 67 - Man In The Middle Attack

In the MITM attack, Person C can eavesdrop undetected, and can also alter or insert data that the other parties are unaware of. This is often used to steal passwords as well as change information, when the communication is unencrypted.

When the communication is encrypted with an encryption certificate, the certificate must be checked to be sure it is authentic and valid. If these checks are not properly performed, the MITM attack becomes possible.

A public Certificate Authority (a commercial service that can be purchased) usually guides the user through the proper certificate checking process when setting up the service. Alternatively, encryption may be setup using a Self-Signed Certificate, however the user is dependent upon their own knowledge and experience, thus Self-Signed Certificates are more prone to human error, oversight, or lack of knowledge of the proper process. If the checks are not properly setup, either method may be subject to this attack method.

While this seems complicated to setup for the average user, devices that perform MITM attacks are commonly available (see the Wi-Fi Pineapple, https://shop.hak5.org/products/Wi-Fi-pineapple). Tools such as these are used by cybersecurity professionals to check for the kind of misconfiguration that would allow an MITM attack, with the goal of helping the client fix those security problems, once identified. However, the devices are available for purchase by anyone.

APPENDIX J. FORENSIC IMAGING TECHNOLOGY

In the forensic community, forensic imaging is often referred to as producing a bit-for-bit image of a data storage medium, most of which historically have been hard disk drives. The statement is not quite so simple – as this Appendix explains.

In the figure below, internal components of a hard disk drive are illustrated. The blue disks are the actual 'disk platters', each of which have an upper magnetic media surface and a lower magnetic media surface. Each disk platter is mounted on a center shaft, called a 'spindle' which is connected to a motor that rotates the disks. For each media surface (i.e., where data can be stored) there is an armature (illustrated in black on the right) with a read/write head (in red, at the end of the armature). In this illustration, there are 4 platters with 2 media surfaces each, for a total of 8 surfaces where data can be stored.

As the disk spins, the read/write heads (similar to the heads in a magnetic tape recorder) move over the data and can read and write new data by magnetizing the disk media. These heads actually aerodynamically fly, a micron or so above the disk platter.

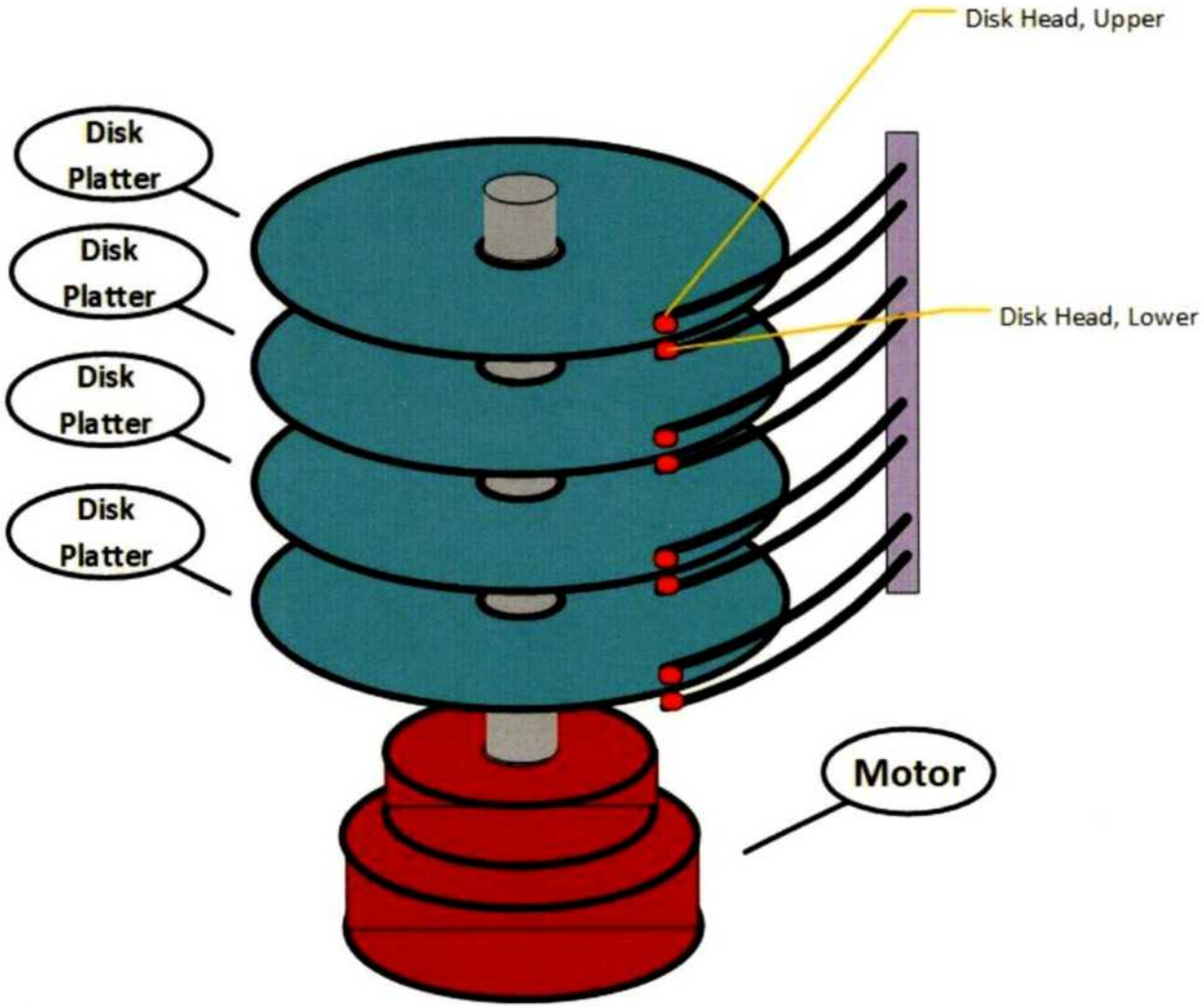


Figure 68 - Illustrative Hard Disk Components

Much like a pizza, each platter surface is divided into sectors (nearly triangular, just as pizza slices are). The surface is further divided into tracks – concentric rings that are smaller and smaller as they move toward the center of the disk. This organization is illustrated in a highly simplified illustration in Figure 67.

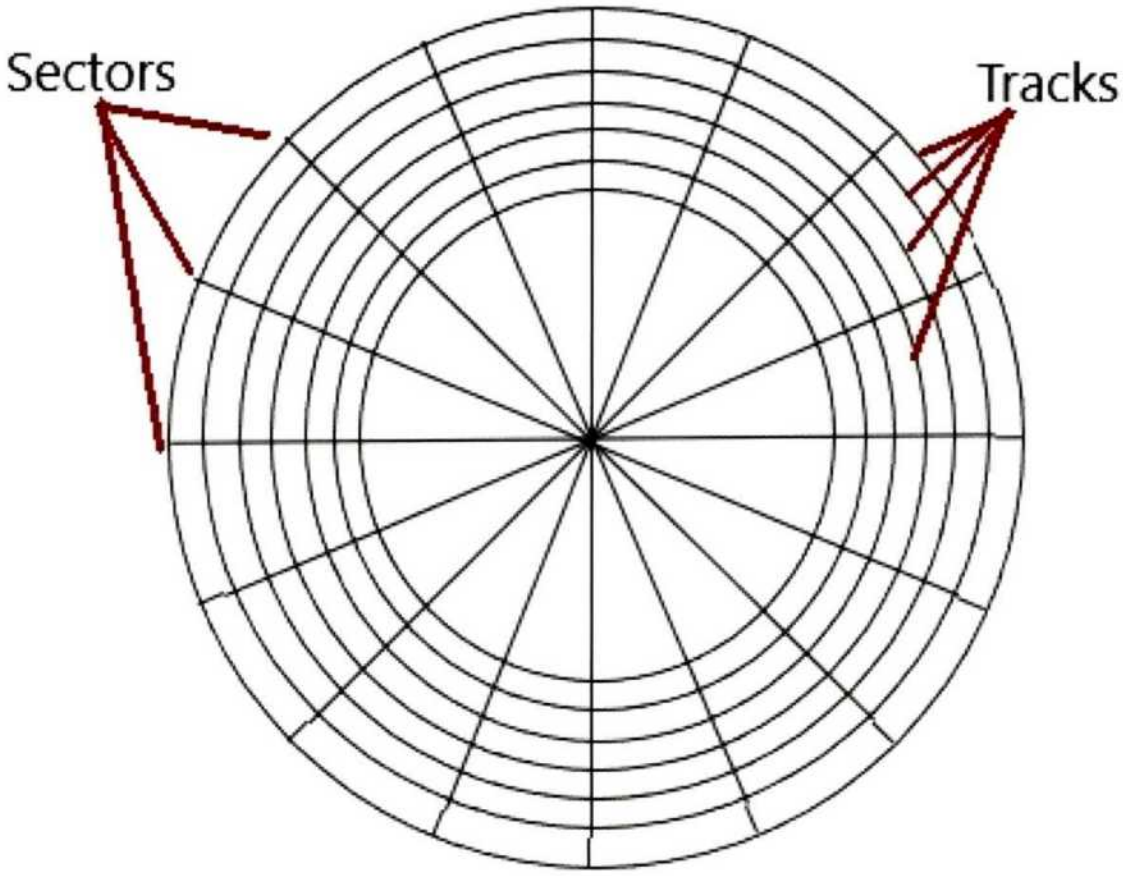


Figure 69 - Disk Track and Sector illustration

In the 1970's, magnetic media was manufactured to have a defect-free surface, but this was prohibitively expensive. Winchester disk technology provided a solution to the high expense. Rather than manufacture a disk media surface that was 100% usable, the manufacture of disk media with a 98% usable surface provided the ability to reduce cost very significantly. This allowed for defective areas on the disk – sectors in which data could not be reliably stored. But this required a scheme to identify these bad areas and ignore them. A map of the disk was developed, from the first sector to the last. As the disk was manufactured, the surface was tested for defects and those sectors with defects were added to the Permanent Defect list, today referred to as the p-list. When the disk is formatted, the disk controller (contained in the disk itself, on its circuit card) will access each physical sector on the disk that is not contained in the p-list, and label that sector with a sequential sector number known as a Logical Block Address (LBA). Obviously the LBA will skip over those sectors in the Defect list. To accommodate the growth of future defects, a list of new bad sectors (to be discovered later in the life of the device) would be added to a Growth List, known as a G-list.

Disk drives are manufactured with more capacity than the end user can access. For example, a 500Gb disk may actually have 580Gb of media storage available. This extra area is known as the Service Area of the disk, and is inaccessible except to the physical disk controller (circuit card that is part of the disk drive itself).

The p-list may be stored in a read-only memory (ROM) on the physical disk controller, or it may be stored in the service area. The g-list is empty at manufacture time and cannot be stored in a ROM but is rather stored in the service area of the disk. The remainder of the service area consists of spare sectors – unused sectors. When a new bad sector is discovered (i.e., a new disk failure) special disk access commands in the disk driver software instruct the disk that a specific logical block is bad and that block is added to the g-list, together with the identity of a spare sector used to replace that sector. The physical disk controller may have replaced physical sector 3921 (LBA 3921) with spare sector 616416, but the new physical storage sector is still addressed by the host computer controller as LBA 3921 because of this mapping. This permits the disk to continue to be used without the computer (and consequently its software) being aware of the replacement sector. If data was unreadable from the damaged sector, the data (file) stored in that location may be damaged and have to be replaced but the disk device still appears, to the computer, to work normally.

Because the sectors in the p-list were defective and never used after manufacture at all, and the g-list sectors were determined after manufacture to be defective, they cannot be read at all. The physical disk controller (built into the drive) has made these p-list and g-list sectors no longer accessible. Spare sectors are also not accessible in the service area of the disk as they are intended to be used as future replacements for active data storage. Finally some physical disk controllers store disk firmware in the service area of the disk but this is neither accessible nor usable to the end user or to the host computer system, buy ONLY to the physical disk controller itself.

Thus, there exist data storage areas on a hard drive that have a list of bad sectors, the actual bad sectors themselves that cannot be read, and spare sectors used to repair the drive (and sometimes disk controller firmware). These data storage areas are protected from access to ensure that the drive can be used even though some defects are present from manufacture and others may develop during the lifetime of the drive.

This detail is provided to explain from a scientific perspective that the statement that "every physical bit on a hard drive is accessible and preserved in a forensic image" is true because the logical hard drive, i.e., the total user accessible data area, is what the computer itself and the user are able to access and every bit of data is preserved exactly as it existed at the time of imaging the data. These data in the service area of the data storage system are not accessible to the computer or any user, are not able to be read by forensic software, and they are not copied as part of a forensic image, but they are also not relevant to a forensic analysis of the computer system as none of the data in this service area can be read, written or manipulated without special equipment used by the manufacturer to create the storage device.

The unreadable service area on the drive is not accessible by the computer and does not contain any user accessible data. Even when a bad sector is added to the g-list, the computer does not access the protected service area; it sends a command to the physical disk controller which adds the sector to the g-list and remaps a spare sector in its place.

The remainder of the disk is known as 'user accessible data area' and is accessible by the computer system. This user accessible data area is formatted by the computer operating system, Microsoft Windows Server 2016 standard in the case of the Mesa County EMS server, and the data components necessary to create a file system are added to the drive (Master Boot Record, Partition Table, list of free data blocks / sectors, directories and ultimately files containing program and user data). Data in the user accessible data area

can be created, modified, or overwritten. When a file is "deleted" by the operating system, the directory entry is marked indicating that the directory slot is now available to be reused and the sector numbers previously occupied by the file are added back to the list of free data blocks (the free list). The data is not physically deleted from the drive – the drive area is simply marked as available for reuse. When the sectors previously occupied (by for example, data from a photographic image, 1 megabyte in size) are reused by a smaller file, for example, 10,000 bytes of data, the remainder of the original file is still present on the drive and these 990,000 bytes of the photo image in this example can be recovered. Forensic practitioners call this "carving" data from the unallocated disk data, because the boundaries of the previous data are no longer defined and must be discovered by the practitioner to successfully recover the data. These data are fragments of previous files, and while recoverable, are incomplete and sometimes present the forensic analyst with difficulty even determining what kind of data it previously was. Data that has been partially overwritten is not likely recoverable, but the remainder of the data that was not overwritten is able to be recovered. Absent context it may not be possible to draw a conclusion from the data so recovered, however sometimes enough information persists that it supports a conclusion alone or in combination with other data recovered.

All data, and every bit stored in the user accessible data area on the disk drive are captured by a forensic image of the entire disk system and are accessible to the forensic analyst in the forensic image. Thus, for all practical purposes, every possible bit and byte of data on the storage device that is accessible is captured and its integrity preserved such that any modification or alteration of the forensic image is detectable.

The data storage device may be a spinning magnetic disk storage device (hard drive), or it may include Solid State Disks (SSD) or other storage devices and may be in a Redundant Array of Independent Disk (RAID) configuration, in which case the data captured in a forensic image will include every bit of data in the logical hard drive exactly as presented to the computer by the data mass storage subsystem. From an evidentiary point of view, the forensic image captures and preserves every bit and byte of data in the logical view of the physical disk. The forensic imaging software copies all the data that can be accessed by the computer system regardless of whether it is partitioned and formatted or not.

Data that has been completely overwritten is not likely recoverable. "Completely overwritten" means that a sector containing 512 bytes of data is overwritten with 512 bytes of new data (random data in the case of "drive wiping" software). The US Department of Defense considers a file containing classified information (up to the Secret classification) to be adequately destroyed and unrecoverable when overwritten with random data 7 times.

In this examination, the term "hard drive image" refers to this exact data set presented to and operated upon by the computer system. It is a complete set of all data accessible to the computer or computer operator and is an accurate reproduction of ALL of the data on the disk system that can be accessed by the computer under examination.

The original data in the integrity-protected forensic archive cannot be altered, and preserves forensic chain of custody, because this examination used an exact copy of from the original preserved in the forensic archive.

In this Appendix the capability of a forensic image has been explained, with the technical detail of hard drive technology to aid in the understanding that the statement that "every bit and byte of data in the hard

drive is captured and preserved", made with reference to the logical view of the data storage medium is technically accurate, and that "every bit and byte of data that can be accessed by a human or a computer operating system IS captured and preserved", integrity controlled and evidentiarily a complete set of all possible data is preserved and presented in the examination.

APPENDIX K. ACCESSING A COMPUTER WITHOUT A PASSWORD

It is a common belief that a password provides safety as a security mechanism.

In this Appendix I discuss some of the many methods by which password can be bypassed, at a high level. Step-by-step instruction is not provided here. Many books have been published[93] and many professional instruction courses and certifications[94] exist for those in the field who need or desire it and it is not my purpose to repeat that content here.

Finding a password

Many resources exist on the Darkweb[95] to obtain passwords that have been broken by criminals and are either offered for free or for sale. The article cited discusses 1.4 billion passwords available for free on the Darkweb. US Title 18, section 1029 makes trafficking in passwords or access devices a crime. I did not search the Darkweb for these passwords because trafficking in passwords is a crime, the Darkweb is also full of criminal content, some of which the mere possession of without any intent, is a crime, as well as malware and ransomware, often disguised in innocent-looking webpages. Venturing onto the Darkweb is a good way to lose all your computer data as a consequence of encountering these subversive "traps".

Method #1 is simply looking up the password. Despite the risk of computer infection or damage, many people do use the Darkweb and this content is available in many cases for free. This risk is so prolific that many services monitor this for you, Norton LifeLock and Identity Force among them, by searching for your credentials on the Darkweb and providing notification if your access has been compromised.

Cracking a password

Passwords, when entered, are encrypted and only the encrypted form of the password is stored. When a person enters a password to login, it is again encrypted and the result is compared to the stored encrypted password. The two encrypted passwords are compared and if they match, access is granted. The encrypted, stored password is *never* decrypted in the process of granting access.

It is possible, once the encrypted stored passwords are obtained, to run various "password cracking" software that tries all conceivable combinations of letters, numbers and symbols until a match between the encrypted stored password and the result under test. The password "cracker" outputs the unencrypted password, once found.

---

[93] https://www.goodreads.com/shelf/show/penetration-testing
https://computingforgeeks.com/best-penetration-testing-books-to-buy

[94] Certified Ethical Hacker (CEH) Certification, GPEN, Certified Penetration Tester (CPT), PenTest+, ECSA- EC Council Certified Security Analyst, Certified Expert Penetration Tester (CEPT), Licensed Penetration Tester (LPT), OSCP – Offensive Security Certified Professional, OSCE – Offensive Security Certified Expert
https://alpinesecurity.com/blog/top-penetration-testing-certifications/

[95] https://www.csoonline.com/article/3266607/1-4b-stolen-passwords-are-free-for-the-taking-what-we-know-now.html

Rainbow Tables

Encrypting every possible password (called a "brute force" method) requires an extensive amount of computing power and is remarkably slow. To speed this process up, "rainbow tables" have been created. These are tables of encrypted passwords and the corresponding plaintext password allowing the application to simply search the list for a matching entry rather than encrypting every possible combination until a match is found.

Many sources of rainbow tables and the software that uses them exist on the Internet and are readily available.

Bypassing a password

It is possible to bypass a password requirement altogether by using special software on a CD, USB thumb drive or other media or installed by one of many access methods. Security professionals use capabilities like password bypass when a password is forgotten and must be recovered. Microsoft operating systems even include the option to create such a bypass mechanism when the operating system is installed (a password recovery disk). There are many password recovery methods identified on the Internet that perform this function across many different operating systems and are readily available on demand including, specifically, for Microsoft Windows Server 2016 Standard.[96]

Exploitation of Services

Often, in the programming of a computer service, for example, a web server, mistakes and oversights are made in the programming process that leave opportunities for a malicious person to obtain unauthorized access. One such example is the inclusion of "non-printable" characters in an input value (meaning that the included data does not show on a screen). This technique fools the receiving computer into accepting part of the input value as a command that it should execute (a command that means "send me your password file," for example). There are many different ways to do this, each with their own deep technical explanation (buffer overflow, cross-site scripting, code injection, manipulation of software timing, etc.). There are many penetration testing textbooks that explain the deep technical process and teach how to do this.

These types of mistakes and oversights account for nearly 170,000 identified weaknesses that allow a computer to be attacked. The CVE[97] system operated by Mitre Corp. has identified 169,169 publicly disclosed vulnerabilities to date. The National Vulnerability Database (NVD[98]) is provided by the National Institute of Technology and Standards (NIST) and contains 808 vulnerabilities that provide full administrative access (between 2005 and the time of this writing). Computer vulnerabilities (weaknesses) are identified nearly daily, and are reported and validated before being published in the CVE or NVD repositories. There exist more vulnerabilities than are publicly known; many are under investigation, not yet validated, while others are known to the US military and intelligence communities and are classified. From these 808 publicly known vulnerabilities, many could be applied to the Mesa County EMS server to grant the type of access demonstrated in this report.

---

[96] https://www.top-password.com/blog/reset-forgotten-windows-server-2016-password/

[97] https://www.cve.org/

[98] https://nvd.nist.gov/vuln/search/results?form_type=Basic&results_type=overview&query=administrative+access&search_type=all&isCpeNameSearch=false

There are entire suites of software that simplify and automate the capability. Manually performing an exploitation may be a difficult process that requires deep technical knowledge but these automated suites simplify the task making it accessible to a larger population of people. For example, Metasploit can obtain access to a system and return to the user a fully logged-in session with administrative access, allowing the malicious user to do whatever they want to with the system, including stealing or altering data. Kali Linux is an operating system (intended for security professionals to test the security of systems) that contains Metasploit and many dozens of other security tools that can be used to exploit a computer system.

Even passwords (and encryption keys) specific to Dominion Voting Systems have been revealed on the Internet, by no less than the U.S. Election Assistance Commission, and are available online at the time of this writing. One such report with the actual system passwords and encryption keys was published more than 10 years ago and is still available online.

Intel Active Management Technology (AMT) and Management Engine (ME)

Every processor made by Intel since 2008, as well as processors made by AMD and others, incorporate a form of this Management Engine (ME) technology.[99] This has not been popularized broadly but is a serious concern for all computer systems.

Embedded in the silicon of microprocessors is an independent processor with its own operating system. This processor runs even when the power is off (as long as there is power to the motherboard), and is accessible via the computer's network interface. It provides its own IP address and MAC address and is capable of bypassing the operating system.

Vulnerabilities identified in 2017 were identified as critical.[100] Researchers indicated that it was possible to read passwords from memory (among other things) and completely bypass the Operating System of today's computers. While no exploitation of this capability has been identified that we know of, Nation-States (including our own) would consider the ability to be highly classified – to the point – we would not know about it.

The vulnerabilities are known as Meltdown and Spectre. They are side-channel attacks against systems.[101]

These vulnerabilities if exploited could provide complete access, undetectably, to a system, even with the computer in a "shutdown" state, as long as the system is plugged in (i.e., power is supplied to the motherboard). This continuous power to the motherboard has long been a feature in modern computer systems and is how the "Wake on LAN" feature is able to function ... it is not that the computer has no power, it just has very low power applied.

---

[99] https://en.wikipedia.org/wiki/Intel_Management_Engine

[100] https://www.intel.com/content/www/us/en/support/articles/000025619/software.html

[101] https://www.intel.com/content/www/us/en/architecture-and-technology/side-channel-variants-1-2-3.html
https://www.intel.com/content/www/us/en/architecture-and-technology/side-channel-variants-3a-4.html
https://www.intel.com/content/www/us/en/architecture-and-technology/l1tf.html
https://www.intel.com/content/www/us/en/architecture-and-technology/mds.html

Dell Integrated Remote Access Controller (iDRAC)

Dell offers a capability known as iDRAC on its servers.[102] It is a completely separate processor with its own Ethernet interface, IP and MAC addresses. It is intended to be used on a highly restricted network for "out of band" management of the server, and allows an administrator (or anyone with access[103]) to reboot the system, access and change the BIOS, and alter the system without the motherboard's processor being able to detect this activity. If you have a server in a data center 30 miles (or more) from your office that needs to be rebooted, and you don't have staff at this remote location, driving an hour or more just to reboot the system is an impediment to productivity – the iDRAC is intended to provide remote control for just this reason.

The primary computer has no way to detect the use of the iDRAC; if used the primary computer's audit and system logs would not record it. An iDRAC is intended to permit access to the core computer and its files.

Strengthening Access Security

One technique for strengthening access security is multi-factor authentication. This is an industry-standard practice and recommended by the National Institute of Standards and Technology (NIST) among many other technical and professional organizations.

Many readers will recognize this multi-factor authentication as something you have already used, once you understand what it is. Multi-factor authentication requires identification be verified by techniques in two or more of the three categories:

1. Something you know (a password, special code, birthday, or other identifying number not related to the information you are accessing),
2. Something you have (an access token, a calculator that accepts an input number and returns an encrypted response, a cellphone where you receive a message to authorize the access, etc.), and
3. Something you are (biometric information, a fingerprint, retina scan, iris scan, face recognition, etc.).

Systems that send you a verification code via cell phone SMS message are a good example of the use of multi-factor authentication.

Best practice in access security is to apply the principle of "Defense in Depth," which is to apply multiple layers of security such that if one fails another serves to protect the system. A "hardened" system requires Defense in Depth, and the proper implementation of multiple security mechanisms, as specified in the DoD Security Technology Implementation Guides (STIGs).

The US Department of Defense employs thousands of military and contractor staff who work full-time on the problem of maintaining sufficient cybersecurity to (hopefully) stay ahead of the threat. Homeland Security maintains a significant cybersecurity division, as does the National Security Agency (NSA) and other parts of the US intelligence community; the Critical Infrastructure Security Agency (CISA) is dedicated to this mission; NIST maintains an entire division for cybersecurity; the DOJ maintains its own capability for the investigation and prosecution of these High-Tech crimes and the High-Tech Criminal Investigator's Association (HTCIA) provides a public private partnership with their law enforcement counterparts. This is

---

[102] https://www.dell.com/support/kbdoc/en-us/000179517/dell-poweredge-how-to-configure-the-idrac-system-management-options-on-servers

[103] Note that this document identifies the default iDRAC userID and password as "root" and "calvin".

a gross understatement of the problem and the resources allocated to address it. Part of the mission of the FBI InfraGard program is to maintain a public-private partnership with the civilian operators of US national critical infrastructure to thwart cybercrime and cyber threats against the USA. The US Secret Service maintains an Electronic Financial Crimes Task Force (EFCTF) to pursue financial cybercrimes. The budget for these efforts far exceeds several billion dollars annually.

Yet our election security depends on temporary workers with very minimal training and no requirement for cybersecurity knowledge, training or certification. DoD requires thousands of security professionals. Is our election infrastructure less important?

The ability to obtain access to a computer without a password is a persistent problem and will continue to be because computers are programmed by humans; and humans are not perfect, they make mistakes.

Unfortunately, there are enough nefarious people in the world exploiting these weaknesses for their own benefit, that this problem is not likely to ever end.

APPENDIX L. SUPPLY CHAIN SECURITY THREAT AND FOREIGN MANUFACTURING

The United States is a significant target of espionage from foreign adversaries. According to the US Director of National Intelligence in their Supply Chain Risk Management Best Practices[104] document,

> "The U.S. is under systematic assault by Foreign Intelligence Entities (FIEs) who have augmented traditional intelligence operations with nontraditional methods, including economic espionage, supply chain exploitation, and the use of students, scientists, and corporate employees, to collect both classified and unclassified information. The scale of this effort has put entire industries at risk. Specifically, the globalization of supply chains presents a major attack vector, characterized by a complex web of contracts and subcontracts for component parts, services, and manufacturing. FIEs use this complexity to obfuscate efforts to penetrate sensitive research and development programs, steal vast amounts of personally identifiable information (PII) and intellectual property (IP), and insert malware into critical components. Supply chain exploitation, especially when executed in concert with cyber intrusions, malicious insiders, and economic espionage, threatens the integrity of key U.S. economic, critical infrastructure, and research/development sectors."

With the growth of global competition, industry in the US is driven to source materials, components, and finished goods from other countries where costs are significantly lower. However, FIEs continue to insert operatives into these foreign supply chains to the USA where they might be strategically positioned to infiltrate supplies using espionage techniques, including inserting surveillance devices into manufactured goods.

This activity includes the contamination of manufactured electronic components with surveillance devices that record and retransmit audio, video and computer data to their foreign controllers.

Presidential Executive Orders 13959[105] signed by President Trump declared a National Emergency (Addressing the Threat From Securities Investments That Finance Communist Chinese Military Companies) and Presidential Executive Order 14032[106] signed by President Biden continued and expanded that National Emergency, banning investment in listed foreign companies. These include manufacturers like Huawei, China Telecom, cellphone manufacturers and electronics manufacturers that have conducted espionage against the US by means of installing covert surveillance devices in equipment during its manufacture.

Infiltration of the supply chain includes the use of hardware and software alterations to systems. The SolarWinds attack on the US Government involved a software infiltration of the supply chain.[107]

---

[104] https://www.dni.gov/files/NCSC/documents/supplychain/20190405-UpdatedSCRM-Best-Practices.pdf

[105] https://www.federalregister.gov/documents/2020/11/17/2020-25459/addressing-the-threat-from-securities-investments-that-finance-communist-chinese-military-companies

[106] https://www.federalregister.gov/documents/2021/06/07/2021-12019/addressing-the-threat-from-securities-investments-that-finance-certain-companies-of-the-peoples

[107] https://www.asisonline.org/security-management-magazine/articles/2021/03/spies-in-the-supply-chain/

These alterations of hardware and software are incredibly sophisticated. The alteration of electronic computer chips to plant malicious circuitry[108] in the design of silicon integrated circuits has been demonstrated at the University of Michigan.[109]

FBI Director Christopher Wray stated that Chinese spying in the U.S. is so widespread the FBI must launch two counterintelligence investigations a day to counter it.[110] China is focused on stealing U.S. technology to increase its capabilities while shortening the research and development time. The FBI currently has over 2,000 active counterintelligence cases related to China.

Bloomberg reported about China's infiltration of the motherboards of Supermicro computers,[111] manufactured outside the United States and how the insertion of a small chip on the motherboard compromised dozens of companies in the US.

The use of components fabricated, assembled and, or manufactured outside the US, whether furnished as individual parts, assemblies or finished goods, exposes them to the risk of foreign exploitation.

As Bloomberg claimed about the exploitation of Supermicro computers, sourcing components from foreign suppliers presents a supply chain risk that can only be avoided by domestic sourcing.

---

[108] https://www.wired.com/2016/06/demonically-clever-backdoor-hides-inside-computer-chip/

[109] https://web.eecs.umich.edu/~taustin/papers/OAKLAND16-a2attack.pdf

[110] https://forwardobserver.com/dailysa-fbi-blown-away-by-chinese-spying/

[111] https://www.bloomberg.com/news/features/2018-10-04/the-big-hack-how-china-used-a-tiny-chip-to-infiltrate-america-s-top-companies

APPENDIX M. COLORADO SECRETARY OF STATE PRESS RELEASE

News Release

Media contact
303-860-6903

Annie Orloff
annie.orloff@sos.state.co.us

Steve Hurlbert
steve.hurlbert@sos.state.co.us

State of Colorado
Department of State
1700 Broadway
Suite 550
Denver, CO 80290

Jena Griswold
Secretary of State

Chris Beall
Deputy Secretary of State

Statement from Colorado Secretary of State's Office Regarding an Official Order to Appoint Sheila Reiner and an Advisory Committee to Supervise Mesa County Elections

Denver, August 17, 2021 - Today, the Colorado Secretary of State's office issued an Order to appoint Mesa County Treasurer Sheila Reiner to supervise all conduct of the Mesa County elections and establish a three-person advisory committee including Representative Janice Rich, Ouray Clerk and Recorder Michelle Nauer, and former Secretary of State Bernie Buescher to advise and assist Reiner in her duties.

"The people of Mesa County deserve safe and secure elections. I am confident that with these appointments, voters in Mesa will be able to exercise their constitutional right to have their voices heard in our democracy. As Secretary of State, my top priority is to ensure all election security protocols are followed and to safeguard Coloradans' right to vote and we will continue to conduct the business required of our office to provide oversight, to ensure the integrity of the state's elections," said Colorado Secretary of State Jena Griswold.

"In light of the ongoing investigation into the chain-of-custody and election security protocol breach in Mesa County, the Colorado County Clerks Association supports the Colorado Secretary of State's designation of an interim election official to conduct and oversee elections in Mesa County until the investigation is complete. While unusual, this important step of placing a top-notch election expert in the office will ensure a safe and secure election

is conducted for the citizens of Mesa County," said Matt Crane, Executive Director of the Colorado County Clerks Association.

While Department of State staff is continuing to conduct analysis and awaiting additional information, as well as the outcome of a criminal investigation, several facts have prompted substantial concern regarding the ability of the Mesa County Clerk and Recorder's office to execute an election in compliance with statute and rule. Of particular concern:

- Mesa County authorized a non-employee, Gerald Wood, to attend the May 25, 2021 trusted build, in clear violation of Election Rule 20.5.4. The Department has confirmed that this individual was present at the May 25, 2021 trusted build event. The Department has determined that Mesa County Clerk and Recorder employees Belinda Knisley and Sandra Brown participated in facilitating the improper presence of this non-employee during the trusted build event by misrepresenting the individual's employment status and role.
- Footage, both video and photos, was posted online showing the BIOS passwords for Mesa County's voting system. The Department knows from the timestamp on the video and from other evidence that it is likely this sensitive information was filmed and collected during the limited access trusted build installation in Mesa County on May 25, 2021. This meeting was limited only to a minimal number of Department of State staff, voting equipment vendor staff, and three individuals approved to attend by Mesa County: Clerk Tina Peters, Sandra Brown, and Gerald Wood.
- Video surveillance of the Mesa County voting equipment was not continuous and cannot confirm chain of custody of voting equipment. The evidence suggests that an individual in the Mesa County Clerk's office directed Mesa County staff to turn off video surveillance of the voting equipment prior to the May 25, 2021 trusted build. The video surveillance cameras were not turned back on until well after the trusted build had been completed, which is inconsistent with the Department's understanding of the normal course of business practice in Mesa County.
- Two hard drive images from Mesa County election servers were released on the internet during the week of August 9, 2021. Analysis confirms that these images belong to Mesa County hard drives and were created before and after the May 25, 2021 trusted build. The only method to make such copies is to physically access the machines.
- One of the hard drive images is believed to have been taken on Sunday, May 23, 2021. The Department has confirmed that Clerk Peters, Sandra Brown, and Gerald Wood accessed the area where election equipment was stored outside of normal work hours on May 23.

At this time, it is clear that the facts uncovered in the Mesa County Clerk and Recorder's office require that the Secretary of State exercise her authority as Colorado's chief election official pursuant to 1-1-107, C.R.S. to supervise all elections occurring under the authority of Title 1 of the Colorado Revised Statutes in order ensure compliance with all election statutes and rules.

Effective immediately and until revoked by the Secretary of State through subsequent order, Sheila Reiner the Mesa County Treasurer and former Mesa County Clerk will supervise all

conduct related to elections occurring under the authority of Title 1 of the Colorado Revised Statutes. The newly formed advisory committee will be responsible for advising and assisting Reiner and will include Representative Janice Rich, Ouray Clerk and Recorder Michelle Nauer, and former Secretary of State Bernie Buescher.

The committee will participate in weekly meetings with Ms. Reiner during the preparation for and execution of an election, unless Ms. Reiner and the committee decide upon another frequency. The committee shall also be permitted to participate in election functions as designated by Ms. Reiner. The Mesa County Clerk and Recorder and staff will take any and all lawful direction from Ms. Reiner and any other Secretary of State designee on any and all election matters.

Given the deadline to purchase, certify, and install trusted build on election equipment before August 31st, a swift appointment was required to ensure safe and secure elections in Mesa County.

# # #

The foregoing Forensic Examination and Report was prepared by me and I am responsible for its content.

This *28th* day of February, 2022.

DOUG GOULD
Chief Technical Officer
CyberTeamUS

Doug Gould Biography

Doug Gould is an expert in Cyber Security with more than 40 years' experience in the field. Doug retired from AT&T after 31 years, where he served as Chief Cyber Security Strategist. He currently serves as Chief Technical Officer at CyberTeamUS.

Doug began at AT&T with Bell Laboratories, serving in the Semiconductor Laser Development department and later in the Bell Lab's Security Group, as a delegate to the Bell Labs' Unix Systems Subcommittee, was an early pioneer in the field of Computer Forensics and won a Bell Labs Innovation Award. At AT&T he designed the security architecture for one of the largest states in the US, consulted with cabinets of the nations' largest corporations and designed the first healthcare network fully compliant with Healthcare Information Exchange standards. Outside AT&T, he has overseen security for a US Government Agency and has solved major cases for the FBI and Secret Service; he has served as an Officer of the Court as a forensic expert and has been an expert witness in landmark cybersecurity cases. He designed security architectures for DoD networks including some of the most sensitive areas of the Government. Doug has owned and led several professional services firms in the Information Security field. He served on the NC Council for Entrepreneurial Development and has consulted with many companies about the complex integration of business and technology.

Doug is the past president of Eastern North Carolina InfraGard, the public-private partnership between the nation's critical infrastructure operators and the US Intelligence community.

Doug's background is at the Master's level in Electrical Engineering, Computer Science, Computer Security and Business Administration.

He is a subject matter expert in:

- Strategic Enterprise Security
- Security Architecture & Design (including network Micro-Segmentation)
- Security Governance
- Risk Management
- Security Device Technologies (Firewalls, IDS/IPS, DLP, SIEMs, Encryption, VPNs, Unified Threat Management, etc., Enterprise, Remote and Cloud)
- Information Forensics (Computer & Network Forensics)
- Public Key Infrastructures
- Identity and Access Management
- Authentication, Authorization and Access Control (incl Biometrics)
- Regulatory Compliance
- Physical Security (Threat Assessment/Risk Analysis, TSCM, Access Control, Counterterrorism & Counterintelligence, facility and site protection)
- Business Continuity & Disaster Recovery Planning
- Response & Recovery Strategy
- Threat Intelligence
- Intelligence Analysis

Doug served as Chief Information Security Officer at the World Institute for Security Enhancement, has written advanced security courses, developed advanced security methodologies and has taught government, private sector professionals and law enforcement agents information security, computer forensics, advanced computer forensic sciences and Technical Surveillance Countermeasures (TSCM).

Doug holds numerous certifications in security including the CISSP and Certified Anti-Terrorism Specialist (CAS), as well as numerous instructor certifications in security.

Doug currently serves as Chief Technical Officer at CyberTeamUS.

He is a Vietnam-era US Navy Veteran where he worked in Electronic Warfare and Electronic Intelligence.

Doug is an invited conference speaker.

Doug Gould Forensic Addendum

MAJOR FORENSIC CASES

- 1986 – Disclosure of National Security Information
  Discovered a leak of highly classified information and was able to identify the perpetrator within a group of 15 people. The FBI and US Naval Investigative Service brought this to resolution.
- Early 1990's – US Secret Service investigation, "Mothers of Doom" hacker case
  At USSS Evidence Lab, in response to a request for assistance from USS SA Jack Lewis, performed evidence recovery and identified 800 pages of evidence, invalidating immunity of a suspect's testimony in a proffer session.
- Late 1990's – Interpath, a North Carolina Internet Service Provider (ISP)
  This ISP was a tier-1 (top level) provider infected with Stacheldraht malware. Investigated the live (running) server and identified that all evidence on disc had been deleted. The only remaining evidence was a running program in memory, which was recovered. This case changed the Best Practice in Forensics – no longer is the first step necessarily removing the power. Had that been done no evidence would remain in this case.
- Late 1990's – As senior security administrator for the US EPA, investigated a complaint from the White House of computer intrusions and discovered an international attack involving 4 countries. Wrote monitoring and tracking software to capture the perpetrator online, brought together the FBI, Royal Canadian Mounted Police (RCMP), Scotland Yard and Deutche Bundespost in a live investigation tracking the intruder resulting in an arrest in Germany.
- South Carolina – A Public Works supervisor accused of violation of county policy was fired and brought countersuit. Forensic investigation recovered 4 3" thick binders of evidence showing sexual misconduct. Countersuit dismissed.
- Discovered Al Qaida attack plans targeting US Soil. Working with the FBI, the perpetrator, who was a foreign citizen in the US. Arrest made within 48 hours and the attack was thwarted.
- Mid-2000's – Florida vs. Rabinowicz – in a case where possession of contraband was the only element of proof, stipulated that the contraband was authentic and present. I proved forensically that the defendant was not technically in possession of the evidence and that evidence was planted. Qualified as an expert witness and provided expert testimony in this case.
- Mid-2000's – Identified a leak of national security from Oak Ridge National Laboratory involving chemical weapon information using forensic analysis and was able to identify the perpetrator. DSS responded and resolved the case.
- Mid-2000's – Investigated sabotage of a health industry contractor. The systems administrator had been fired and sabotaged the system. Solved the case and the administrator went to prison.

INSTRUCTOR OF FORENSICS

- Taught Forensics and Advance Forensic Techniques to State Law Enforcement, Military and major corporate customers at the World Institute for Security Enhancement.
- Taught Technical Surveillance Countermeasures (TSCM) course for government and industry at the World Institute for Security Enhancement.

Wrote the entire course and taught the entire CISSP curriculum at Able Information Systems.

Mesa County Colorado Voting Systems

Report #3
Election Database and Data Process Analysis

March 19 2022

Table of Contents

Executive Summary 3
Introduction 5
Definition of Terms 7
Analysis 9
Discussion 26
Conclusions 28
Appendix A – Batches in Original Adjudication Database 31
Appendix B – Batches in New Adjudication Database 38
Appendix C – EMS User Log Events in November 2020 59
Appendix D – EMS User Log Events in March 2021 68
Reference A – Databases and Tables 73
Reference B – Scanner Speed 75
Reference C – Scanners Used by Mesa County 76
Reference D – Data Movement from Batches to Votes 77

EXECUTIVE SUMMARY

This report documents the findings of an examination of tabulated vote databases based on forensic analysis of the drive image of Mesa County, Colorado's Dominion Voting Systems (DVS) Election Management System (EMS) server. The findings in this report were prepared by the authors as consultants to the legal team representing Tina Peters, the Mesa County Clerk and Recorder, pursuant to her statutory duties as Mesa County's Chief Election Official. The findings provide evidence of potentially unauthorized and illegal manipulation of tabulated vote data during the 2020 General Election and 2021 Grand Junction Municipal Election. Because of this evidence, which led to the vote totals for those elections being impossible to verify, the results and integrity of Mesa County's 2020 General Election and the 2021 Grand Junction Municipal Election are in question.

This analysis was performed using the forensic image of the EMS server, which was backed up before Colorado Secretary of State and DVS overwrote the hard drive with D-Suite version 5.13.

Findings and Implications:

1) There was an unauthorized creation of new election databases during early voting in the 2020 General Election on October 21, 2020, followed by the digital reloading of 20,346 ballot records into the new election databases, making the original voter intent recorded from the ballots unknown. In addition, 5,567 ballots in 58 batches did not have their digital records copied to the new database, although the votes from the ballots in those batches were recorded in the Main election database.

2) The same unauthorized creation of new election databases occurred during the 2021 Grand Junction Municipal Election on March 30, 2021, followed by the digital reloading of 2,974 ballot records, making the original voter intent recorded on those ballots unknown. In addition, 4,458 ballots in 46 batches did not have their digital records copied to the

new database, although the votes from the ballots in those batches were recorded in the Main election database.

3) The absence of secure hash algorithm (.sha) files for each digital ballot image makes the authenticity of each digital ballot image, and the ballot-level record for those ballots, impossible to verify.

4) The true total vote count in Mesa County, Colorado cannot be accurately calculated for the 2020 General Election or the 2021 Grand Junction Municipal Election from records in the databases of the county's voting system.

5) There is no function or feature on the EMS server that could be executed inadvertently or deliberately by a local election official that would cause this combination of events to occur, especially within the time frame that these events occurred. Given the complex sequence of data manipulations and deletions necessary to produce the digital evidence described in this report, this combination of events could not have been the result of either deliberate or inadvertent actions by those officials.

6) Dominion's installation of the Trusted Build update on the EMS in May of 2021, as ordered by the Colorado Secretary of State, destroyed all data on the EMS hard drive, including the batch and ballot records that evidenced the creation of new databases and reprocessing of ballot records described in Findings 1 and 2 above. This destruction of all data by the trusted build is described in the "Mesa County, Colorado Voting Systems Forensic Examination and Analysis Report".

7) The fact that such ballot record manipulation has been shown demonstrates a critical security failure with the DVS EMS wherever it is used. The manipulation would not be identifiable to an election official using the voting systems, nor to an observer or judge overseeing the election conduct, much less to citizens with no access to the voting systems; without both cyber and database management system expertise, and

unfettered access to database records and computer log files (many of which were destroyed by the actions of the Secretary of State) from the EMS server, the manipulation would be undetectable.

INTRODUCTION

The use of computerized election management systems is now nearly universal across counties in the United States. While the use of these systems is touted as "efficient", potentially decreasing manpower costs and time to produce election results, it also greatly reduces the transparency of the election process and exposes our elections to extraordinary vulnerability from both inadvertent and deliberate misconfiguration or misuse. Americans' right to free and fair elections is inalienable, but that right is infringed by lack of transparency, and by whatever lies behind that opaque curtain.

Without free and fair elections and the transparency to see it for themselves, without relying on the assertions of any other person or organization, Americans' consent and the legitimacy of our government, at all levels, is in doubt. If Americans' votes are to be recorded and counted by machines, every aspect of those machines' operation, configuration, and data must be recorded, immediately available at no cost or administrative burden to citizens and their independent examiners and confirmed 100% accurate through that independent verification. The absence or shortfall of any of those three imperatives (recorded, available, and independently verified) should immediately cause the public to distrust both the purported result from those machines, and also anyone who insists that they accept those results.

Numerous Federal and State laws attempt to safeguard our voting rights and the integrity of our elections. Title 52 USC provides for much of the Federal guidance in this area, and Colorado Revised Statute (CRS) Title 1 covers most of the Colorado state guidance.

a) 52 U.S. Code § 10307 prohibits any person acting under color of law to "…willfully fail or refuse to tabulate, count, and report…" the vote of any person entitled to vote.

b) 52 U.S. Code § 10308(a) prescribes penalties for any person depriving or attempting to deprive any person of voting rights under Federal statute.
c) 52 U.S. Code § 10308(c) prescribes penalties for conspiring to violate or interfere with secured voting rights.
d) 52 U.S. Code § 20701 mandates the preservation of all election records for 22 months after an election for Federal offices.[1,2]
e) 52 U.S. Code § 20702 prescribes penalties for theft, destruction, concealment, mutilation, or alteration of § 20701 election records.
f) 52 U.S. Code § 21081 requires that voting systems used in elections for Federal office meet the standards of that section, including that the voting system shall produce a record with an audit capacity for such system, and that "the error rate of the voting system in counting ballots…shall comply with the error rate standards established under section 3.2.1 of…" the Federal Election Commission 2002 Voting System Standards (VSS).[3]
g) CRS §1-5-601.5 requires that voting systems and equipment in Colorado meet 2002 VSS standards, at minimum.
h) CRS §1-7-802 requires the preservation of election records for 25 months after elections.
i) CRS §1-13-111 prescribes penalties for destroying, removing, or delaying delivery of election records.

Title 52 clarifies that the "every officer of election" is responsible for maintaining the election records.

[1] U.S. Department of Justice Publication "Federal Law Constraints on Post-Election 'Audits'," July 28, 2021, states that "The materials covered by Section 301 extend beyond 'papers' to include other 'records.' Jurisdictions must therefore also retain and preserve records created in digital or electronic form."
[2] The Federal Election Commission's 2002 Voting System Standards, the standards of which are mandatory minima for certification of voting systems under Colorado state statute § 1-5-601.5., specifies that a voting system which "…provides access to incomplete election returns and interactive inquiries before the completion of the official count…shall: a. …be designed to provide external access to incomplete election returns only if that access for these purposes is authorized by the statutes and regulations of the using agency…b. Use voting system software and its security environment designed such that data accessible to interactive queries resides in an external file, or database, that is created and maintained by the elections software under the restrictions applying to any other output report, namely, that: 1) The output file or database has no provision for write-access back to the system. 2) Persons whose only authorized access is to the file or database are denied write-access, both to the file or database, and to the system," and states that the Standards are "intended to address…risks to the integrity of a voting system...," including "…Changing calculated vote totals;…" and "Preventing access to vote data, including individual votes and vote totals, to unauthorized individuals;…"
[3] 2002 VSS, para 3.2.1 specifies "d. For central-county systems…: Consolidation of vote selection data from multiple counting devices to generate jurisdiction-wide vote counts, including storage and reporting of the consolidated vote data…a target error rate of no more than one in 10,000,000 ballot positions." A ballot position is each and every choice (e.g. a "bubble" which can be marked or filled-in) on a ballot selectable by a voter to convey their voting choices.

Mesa County, Colorado, uses software and hardware provided by DVS and for the 2020 General Election and the 2021 Grand Junction Municipal Election, specifically used "D-Suite 5.11-CO." The primary voting system EMS server, which contains the raw tabulated vote information used to produce official election reports, makes use of Microsoft SQL Server 2016 databases running on the Microsoft Windows Server 2016 operating system. The forensic image used for the analysis, created on May 23, 2021, has been validated as authentic.

DEFINITION OF TERMS

"Ballot": Mesa County used two-sided paper ballots in the November 2020 General Election and the 2021 Grand Junction Municipal Election. A ballot is a device used to cast votes in an election. In Colorado, ballots are pieces of paper defining races and issues, and reflecting the choices of individual voters from among the options available for each race and issue. A digital image of each paper ballot is created by the DVS D-Suite voting system during the processing of ballots, as described below, and that ballot image is stored on the designated "NAS (Network Attached Storage device)" of the D-Suite voting system.

"Adjudication": A term used to describe the process of determining voter intent from a voter's ballot marks, where ballot markings are ambiguous. According to Dominion's Democracy Suite Use Procedure Manual, adjudication is "the process of examining voted ballots to determine, and, in the judicial sense, adjudicate voter intent". In the DVS D-Suite, adjudication refers to the operation and use of a software component called "EMS Adjudication," and the process of using that software component to manually or automatically interpret voter intent from scanned ballot images, and then to record that interpretation as the record of the vote choices for the affected ballots, in both "result files" and ballot images. Depending on software configuration choices, individual ballot images/result files, entire batches of ballot images/result files, or all ballot images/result files can be subjected to automatic or manual adjudication on the basis of "exceptions" or "outstack conditions" (e.g., "overvotes", where too many choices are marked for a race or issue; or "marginal marks" when ballot choice ovals are not adequately filled in), or by the arbitrary decision of EMS administrators.

“Manual Adjudication”: Either all ballot images, or individual ballot images, or those from particular batches or tabulators, in which voter intent for any race or issue is flagged by the EMS Adjudication software module as not being determinable (or as having “exceptions”), are, in theory, sent to “Manual Adjudication” stations where officials called “Adjudicators” view the digital images of the ballots and decide the voter’s intent. In this Report we sometimes use the terms “manual adjudication” and “machine adjudication” to clearly distinguish the process of human judging of voter intent from the process of the DVS EMS Adjudication software’s determining of voter intent.

“Adjudication database”: The DVS D-Suite version used in Mesa County during the November 2020 and April 2021 elections maintains a separate SQL Server database, called an “AdjudicableBallotStore,” created by DVS software, for each election which contains records of all batches and ballots scanned into the voting system through ImageCast scanning workstations, and any batch and ballot records manually entered. The database maintains critical information about each batch and ballot, most importantly the ballot Adjudication status and the file location of the ballot image. A batch can have any of the following adjudication statuses in the adjudication system: In-Progress, Read Error, Review, Pending Submission, Submitted, or Submission Error.[4] Throughout, “Adjudication database."

“Main election database”: The DVS D-Suite version used in Mesa County during the November 2020 and April 2021 elections maintains a database for each election, called an “ElectionStore” by DVS, which contains information defining an election, including contest, candidate, and ballot definitions as well as aggregated vote information which is used to produce all election reports generated by County officials. Throughout, “Main database.”

“Tabulation database”: The DVS D-Suite version used in Mesa County during the November 2020 and April 2021 elections maintains a database for each election,

[4] In-Progress batches have been acquired by the system (e.g. through scanning at an ICC) and have ballots being served to clients (Adjudication); Read Error batches are those which encounter errors while being loaded into the system; Review are batches with all ballot adjudication complete, including batches with no adjudication required; Pending Submission are batches submitted to tally, but which have not yet completed that transmission to the tally process; Submitted are batches which have completed the transmission to the tally process; Submission Error are batches that were submitted to the tally process, but which were unsuccessfully submitted.

called a “TabulationStore” by DVS, which contains the timestamps and ballot counts for each batch of ballots, which duplicates that information contained in the Adjudication database. It contains other tables which are not used by Mesa County’s elections. Throughout, “Tabulation database.”

“Reprocessed”: For the purposes of this Report, the term “reprocessed” means that one or more data records which had already been created, presumably by scanning of paper ballots through an ImageCast Central (ICC) workstation, though also technically possible through manual entry of records, within the databases associated with an election, were loaded into the system *again* to a different database, and that this re-loading was not performed in connection within any documented, authorized election-related operations procedure or function. A comparison with the log files of the respective ICC workstations might reveal whether the reprocessed paper ballots were, in fact, rescanned at the ICC, but many of those log files have been destroyed by the Secretary of State’s “Trusted Build.”

ANALYSIS

I. Evidence of ballot record data manipulation – November 2020 General Election

Our analysis shows manipulation, which was neither initiated nor authorized by Mesa County election clerks, of the batches and ballots processed during the first three days of ballot processing in the November 2020 General Election.

The following timeline of events, beginning October 19, 2020, when Mesa County began processing ballots in the General Election, demonstrates this manipulation of ballots.

October 19, 2020 – October 21, 2020, 2:14 PM

On these first three days of ballot counting in Mesa County, up until 2:14 PM on October 21, 2020, 267 batches, consisting of 25,913 ballots, were physically processed (physically scanned on DVS ICC scanners with voters’ choices, in the

form of marks on the ballots, scanned and interpreted by software) through three tabulators, internally identified in the Main database as tabulator IDs 4, 7, and 10. Mesa County election clerks reported no unusual activity or errors encountered during the processing of these 267 batches. The Adjudication database used at this time contains records of all batches with a sequential "load order" of 1 to 267, and other tables within it record the information about each ballot, for instance the time it was entered into the database, the tabulator used, and the adjudication status. Those which were selected for Adjudication have the proper status records indicating that the normal adjudication steps occurred.

The initial 10 batches processed through tabulator 10, containing a total of 941 ballots, had timestamps indicating that they were all entered into the database within 47 seconds (total – not 47 seconds per batch, but 47 seconds for 10 batches). (See Appendix A for a list of the batches and their timestamps in the **original** Adjudication database.) The Canon DR-G1130, which according to purchasing documents and Colorado Secretary of State voting equipment inventories is the model of scanners used by Mesa County (see Reference C and the Colorado Secretary of State website[5]), operates at approximately 100 pages per minute (ppm), duplex, meaning that scanning both sides of each ballot would take no less than 0.01 minutes, which is 0.6 seconds, per ballot. 941 ballots at 0.6 seconds per ballot should have taken a minimum of 564 seconds, or slightly under 9 and a half minutes, a significantly longer interval than 47 seconds, which is physically impossible. Mesa County election clerks were unaware of these batch timestamps, or any issue which could explain them.

October 21, 2020 - 2:14 PM

According to the data contained in the EMS SQL Server Database, new Tabulation and Adjudication databases were created on the EMS server at 12:18:50 PM October 01, 2020. These databases initially contained no data records.

[5] CO SecState Voting Equipment Inventory at: https://archive.ph/RQS91

Figure 1. "Before" Screenshot of Databases on the Mesa EMS Server

	name
1	2020 Mesa County General-2020-09-05-00-10-20
2	AdjudicableBallotStore_2020_Mesa_County_General_2020-10-01_12:18:50
3	TabulationStore_2020_Mesa_County_General_2020-10-01_12:18:50

One Adjudication database and one Tabulation database were listed, with creation times before the counting in Mesa County began on October 19, 2020.

Figure 2. "After" Screenshot of Databases on Mesa EMS Server

	name
1	2020 Mesa County General-2020-09-05-00-10-20
2	AdjudicableBallotStore_2020_Mesa_County_General_2020-10-21_14:18:51
3	TabulationStore_2020_Mesa_County_General_2020-10-21_14:18:51
4	AdjudicableBallotStore_2020_Mesa_County_General_2020-10-01_12:18:50
5	TabulationStore_2020_Mesa_County_General_2020-10-01_12:18:50

Two Adjudication databases (“AdjudicableBallotStore”) and two Tabulation databases (“TabulationStore”) are now listed, one set of which had creation times before the date and time ballot scanning and tabulation began in Mesa County on October 19, 2020 and the other set of which the EMS server data indicate were created two and a half days *after* ballot scanning and tabulation began.

It has been observed that a clerk giving the EMS system a command to stop and then restart adjudication in an election again creates new Adjudication and Tabulation databases. Mesa County clerks are very certain that they did not initiate any such action in either the November 2020 or the April 2021 elections. Therefore, it is likely that a procedure internal to the DVS software had to perform a stop and restart of the adjudication services in order to perform the batch and ballot manipulation which occurred later (see below).

There are only a few possibilities which would explain how the database copying process was initiated.

1. **Direct action by Mesa County personnel**
   The client application used by election clerks does give them the ability to stop and restart adjudication, which would create the new databases.

   However, Mesa County personnel are very clear that they did nothing of the sort and explained that they would only do such a thing in an extreme emergency, as the process would have made the production of legally mandated reports very difficult.

2. **Triggered remotely**
   "Report #2, Forensic Examination and Analysis Report" by D. Gould identifies numerous security vulnerabilities in the DVS EMS server. A signal, or external trigger,[6] giving instructions to software inside the EMS server could have been sent to and received through any of the open communication ports, or through the port 80 Web Server port, which has been demonstrated to be open on the server and accepting commands via an application programming interface (API).[7] This signal, along with other information, could have been received via a local network connection (from any device connected to the EMS server's internal network), from a remote network connection (if the EMS server's internal network has been bridged to the external internet), or via an internal cellular modem installed in the EMS server. If the EMS Server was connected to a wireless network, it is feasible that even a cell phone outside of the building, but still within the wireless signal radius, could have been used to trigger the events.

   This option is plausible but infers a degree of external, time-sensitive control over the DVS equipment in use in Mesa County. This control might

[6] E.g., an "external trigger" most people are familiar with is the function whereby their smartphone's wi-fi connection is turned on in response to detecting the proximity of a saved, pre-approved wi-fi network. The external trigger satisfies the criteria of an internal, saved rule for application behavior, and the application then executes the correlated command or function. We likely don't think of "Do Not Disturb" mode on our smartphones as being similarly controlled by an external trigger, but if our smartphones are configured to "use network time," meaning the time signal transmitted by the cellular carrier network, then our smartphones' "Do Not Disturb" mode isn't turned on at the time we set, per se, but when our cellular carrier tells our phone that the specified time is reached.
[7] An API is a specification for interaction which allows computer applications to communicate with, make requests to, and issue commands to other computer applications. I.e., API enables machine-machine communication, coordination, and command and control, depending on the permissions and allowable exchanges of the specific API specification.

be considered undesirable by the perpetrators responsible for manipulating the election data, because it was a possibility that any unauthorized network connections, whether they be via standard ethernet, wireless network connection, or cellular modem, could have been discovered during the election period.

3. **Algorithmically Triggered**
   A software algorithm[8] running inside the DVS computer systems in Mesa County could have made the decision to perform the new database creations and the selected record manipulation which followed based on preprogrammed criteria related to the election results at the time.

   Given that this method requires the least amount of external control and monitoring, this option would seem to be the most likely. The decision to copy the Adjudication and Tabulation databases and re-process the ballot records would be made by software running inside the Dominion EMS (or inside another connected machine running Dominion software) based on unexpected voting patterns.

October 21, 2020, 2:30 PM – 2:34 PM

During this time period, 209 out of the original 267 batches (containing a total of 20,346 ballots) were digitally – not physically – loaded into the new Adjudication and Tabulation databases. Specifically, records for batches with load order 2 through 59 were not reloaded and do not appear in the new Adjudication database in any form. The timestamps of the 209 batch records (load order 1 and load orders 60 through 267) show an impossibly short processing time (approximately 4 seconds each) for these batches to have been physically processed into the newly created Adjudication and Tabulation databases. As described above, because of the minimum scanning time of one

[8] An "algorithm" is simply a set of rules for logical, sequential consideration of inputs (e.g. a contingent variable state, like "the switch is off" or "the switch is on," or the value of field/memory location "X" is "1" or is "Not 1") to produce a consistent, expected output. In this case, a simple, hypothetical algorithm might have been something like "IF ('numberofbatches'>50) AND ("ElectionProjectActive"=TRUE) AND ("EMSAdminUserLoggedIn"=FALSE) AND (VOTETOTAL,"InternalMachineID:01">VOTETOTAL,"InternalMachineID:02") AND (SYSTIME>20201019) AND (SYSTIME<20201103) THEN COPY:BATCHID030010:BATCHID030059 AND INSERTINTO "adjudicableballotstore," etc.

minute per batch for the Canon DR-G1130 scanner-based tabulator, it is not possible for these 20,346 ballots to have been physically rescanned (i.e., the paper ballots were not reloaded into the scanning hardware), but rather the digital batch and ballot records were directly added to the new Adjudication database. This indicates that the batches could only have been loaded into the newly created Adjudication and Tabulation databases by using software code or a script running within the EMS server. See Appendix B for a list of all batches and their timestamps in the new Adjudication database. See Appendix C for a list of all commands executed prior to and after the database copy, which provides a precise timeline of the effects of those commands on the database copy.

It is important to note that this unauthorized procedure only copied the records of selected batches of ballots, indicating that this was an intentional act.

Below is a screenshot of the beginning of the list of batches recorded in the original Adjudication database, sorted by the order that they were loaded:

Figure 3. List of Batches Recorded in the Original Adjudication Database, Sorted by Load Order

	TabulatorId	BatchId	Category	CvrBatchId	Name	LoadOrder	CreationTime	ModificationTime
1	10	4001	0	1	Tabulator 10 - Batch 4001	1	2020-10-19 12:07:40.850	2020-10-19 12:09:58.200
2	10	4002	0	2	Tabulator 10 - Batch 4002	2	2020-10-19 12:07:44.443	2020-10-20 10:43:25.547
3	10	4003	0	3	Tabulator 10 - Batch 4003	3	2020-10-19 12:07:48.257	2020-10-20 10:43:26.437
4	10	4004	0	4	Tabulator 10 - Batch 4004	4	2020-10-19 12:07:50.960	2020-10-20 10:43:27.423
5	10	4005	0	5	Tabulator 10 - Batch 4005	5	2020-10-19 12:07:53.960	2020-10-20 10:43:28.500
6	10	4006	0	6	Tabulator 10 - Batch 4006	6	2020-10-19 12:08:12.123	2020-10-20 10:43:29.907
7	10	4007	0	7	Tabulator 10 - Batch 4007	7	2020-10-19 12:08:16.137	2020-10-20 10:43:30.953
8	10	4008	0	8	Tabulator 10 - Batch 4008	8	2020-10-19 12:08:20.107	2020-10-20 10:43:32.390
9	10	4009	0	9	Tabulator 10 - Batch 4009	9	2020-10-19 12:08:24.170	2020-10-20 10:43:33.673
10	10	4010	0	10	Tabulator 10 - Batch 4010	10	2020-10-19 12:08:28.233	2020-10-20 10:43:34.703
11	4	2001	0	11	Tabulator 4 - Batch 2001	11	2020-10-19 12:23:35.457	2020-10-20 10:42:50.047
12	4	2002	0	12	Tabulator 4 - Batch 2002	12	2020-10-19 12:30:25.763	2020-10-20 10:42:51.343
13	4	2003	0	13	Tabulator 4 - Batch 2003	13	2020-10-19 12:32:30.137	2020-10-20 10:42:52.470
14	4	2004	0	14	Tabulator 4 - Batch 2004	14	2020-10-19 12:36:19.937	2020-10-20 10:42:52.970
15	4	2005	0	15	Tabulator 4 - Batch 2005	15	2020-10-19 12:43:25.387	2020-10-20 10:42:53.797
16	4	2006	0	16	Tabulator 4 - Batch 2006	16	2020-10-19 13:50:28.623	2020-10-20 10:42:48.593
17	4	2007	0	17	Tabulator 4 - Batch 2007	17	2020-10-19 13:54:18.990	2020-10-20 10:42:54.533
18	4	2008	0	18	Tabulator 4 - Batch 2008	18	2020-10-19 13:58:23.777	2020-10-20 10:42:56.080
19	4	2009	0	19	Tabulator 4 - Batch 2009	19	2020-10-19 14:03:28.847	2020-10-20 10:42:57.673
20	4	2010	0	20	Tabulator 4 - Batch 2010	20	2020-10-19 14:06:33.427	2020-10-20 10:42:58.877
21	4	2011	0	21	Tabulator 4 - Batch 2011	21	2020-10-19 14:10:23.157	2020-10-20 10:42:59.563
22	4	2012	0	22	Tabulator 4 - Batch 2012	22	2020-10-19 14:14:28.253	2020-10-20 10:43:00.360
23	4	2013	0	23	Tabulator 4 - Batch 2013	23	2020-10-19 14:18:33.053	2020-10-20 10:43:00.970
24	4	2014	0	24	Tabulator 4 - Batch 2014	24	2020-10-19 14:22:22.753	2020-10-20 10:43:01.533

Note that there is a sequential order with all load order numbers represented.

Below is a screenshot of the same table in the newly created Adjudication database:

Figure 4. List of Batches in Newly Created Adjudication database

	TabulatorId	BatchId	Category	CvrBatchId	Name	LoadOrder	CreationTime	ModificationTime
1	10	4001	0	1	Tabulator 10 - Batch 4001	1	2020-10-21 14:20:07.257	2020-10-22 10:33:50.593
2	10	4025	0	60	Tabulator 10 - Batch 4025	60	2020-10-21 14:20:26.273	2020-10-22 10:33:51.907
3	4	2036	0	61	Tabulator 4 - Batch 2036	61	2020-10-21 14:20:30.477	2020-10-22 10:31:56.047
4	10	4026	0	62	Tabulator 10 - Batch 4026	62	2020-10-21 14:20:34.430	2020-10-22 10:33:53.330
5	4	2037	0	63	Tabulator 4 - Batch 2037	63	2020-10-21 14:20:39.043	2020-10-22 10:31:56.780
6	10	4027	0	64	Tabulator 10 - Batch 4027	64	2020-10-21 14:20:43.107	2020-10-22 10:33:54.423
7	10	4028	0	65	Tabulator 10 - Batch 4028	65	2020-10-21 14:20:47.370	2020-10-22 10:33:55.703
8	4	2038	0	66	Tabulator 4 - Batch 2038	66	2020-10-21 14:20:51.527	2020-10-22 10:31:57.297
9	4	2039	0	67	Tabulator 4 - Batch 2039	67	2020-10-21 14:20:55.887	2020-10-22 10:31:58.187
10	4	2040	0	68	Tabulator 4 - Batch 2040	68	2020-10-21 14:21:00.060	2020-10-22 10:31:58.860
11	4	2041	0	69	Tabulator 4 - Batch 2041	69	2020-10-21 14:21:04.060	2020-10-22 10:31:59.907
12	10	4029	0	70	Tabulator 10 - Batch 4029	70	2020-10-21 14:21:08.260	2020-10-22 10:33:56.843
13	4	2042	0	71	Tabulator 4 - Batch 2042	71	2020-10-21 14:21:12.747	2020-10-22 10:32:00.593
14	10	4030	0	72	Tabulator 10 - Batch 4030	72	2020-10-21 14:21:16.810	2020-10-22 10:33:57.640
15	4	2043	0	73	Tabulator 4 - Batch 2043	73	2020-10-21 14:21:20.747	2020-10-22 10:32:01.217

While the record of the batch with load order 1 was copied, there is a gap of 58 batches before the second line, which is a record of the batch with load order 60. Batch load order numbers 2 through 59 were *not* copied, effectively deleting them in the new Adjudication database.

The data records describing the batches and the ballots contained within them in the new Adjudication database, specifically the time stamps shown in Appendix B as well as statements by Mesa County election officials, indicate that the paper ballots and batches were not physically re-scanned. Therefore, it appears the process of scanning these ballots was simulated, and the records of the batches and the ballots contained within them were electronically transferred from the original Adjudication database into the new Adjudication database.

For example, below is the sequence of events detailing the processing of batch 4024 (whose ballots and records were *not* copied to the new Adjudication database) and batch 4025 (whose ballots and records were copied to the new Adjudication database). This will illustrate the contrast between copied and uncopied batch and ballot records.

Batch 4024 is recorded in the original Adjudication database as being created at 4:09:34 PM on October 19. It contained 100 ballots and was scanned by tabulator 10. Ten of these ballots from batch 4024 were subsequently manually adjudicated. The manually adjudicated ballot numbers in the batch which were manually adjudicated were 4, 8, 13, 14, 30, 48, 63, 87, 88, and 90. Then, the votes contained on all 100 ballots were recorded in the appropriate tables in the Main database (see Reference A for a list of these tables). When the new Adjudication database was created, no records from Batch 4024 were copied to it, and thus there was no reprocessing or physical rescanning of the ballots. Adjudication history for the 10 ballots which were manually adjudicated was no longer available to the Mesa County clerks, and the original voter intent of these ballots is unknown.

In contrast, Batch 4025 is recorded in the original Adjudication database as being processed at 4:12:23 PM on October 19. This batch contained 99 ballots and was also scanned by tabulator 10. Fourteen of these ballots were subsequently manually adjudicated. The ballot numbers in the batch which were manually adjudicated were 3, 10, 13, 21, 22, 23, 34, 40, 49, 59, 66, 79, 97, and 99. Then, the votes contained on all 99 ballots were recorded in the appropriate tables in the Main database.

After the new Adjudication database was created, a record of Batch 4025 appeared in its tables at 2:20:26 PM on October 21. It is still listed as having 99 ballots and from tabulator 10. In the new Adjudication database, however, only 6 of the batch 4025 ballots (8 less than the first time these batches were entered into the original Adjudication database), were *again* manually adjudicated. The individual ballot numbers were 3, 21, 22, 40, 59, and 66. At this point, the vote records from at least those 6 ballots and possibly all 99 would have been recorded in the appropriate tables in the Main database, replacing the votes which were already in that database from those ballots. Adjudication history for the 14 ballots which were manually adjudicated was no longer available to the Mesa County clerks, and the original voter intent of these ballots is unknown.

The selected batches in the new Adjudication database (batch 1 and batches 60 through 267) appeared in the same serial order that they were loaded into the original Adjudication database, with the same batch numbers, ballot counts, and load order numbers (compare Appendix A and Appendix B).

October 21, 2020, shortly after 2:34 PM

At this point, as reported by Mesa County election officials, some Mesa County adjudication officials began noticing that they were being asked to look at ballots that they had already adjudicated. This is consistent with these ballots and batches being reprocessed in the new Adjudication database. When the new Adjudication database was created, and the selected records described above were copied and reprocessed, there were outstanding ballots from the last set of batches scanned before the event. As some of these ballots were sent to manual adjudication again after the batches were reprocessed, this caused a situation where the same ballot was in the manual adjudication process twice. This caused confusion among the election staff who were assigned the duty of manual adjudication, since when a ballot was adjudicated the second time the master count of adjudicated ballots, which is displayed by the Dominion system and is used by the election clerks to track the overall adjudication process, did not change. This caused the Adjudication officials to assume that there had been an error and, in some cases, to attempt additional manual adjudications of the same ballot with the same unsatisfactory result.

According to several Mesa County election officials, DVS support was contacted at approximately 4PM on the 21st of October, and while the support representative claimed to not have a solution for the issue Mesa County was seeing, that issue ceased soon afterwards. This indicates that DVS may have performed or caused to be performed an operation unknown to Mesa County election officials (and outside of their control) to address this problem which manifested after the unauthorized database manipulation.

Of the 209 batches which were processed twice (batches 1 and 60 through 267), the ballot counts match between the old and new Adjudication database. However, DVS software marked 2,166 ballots for manual adjudication the first time they were processed in the original Adjudication database, but when reprocessed in the new Adjudication database the software marked only 965 ballots for manual adjudication.

The same ballots run through the same hardware and evaluated by the same software should have had the same resulting ballots marked for adjudication, but they did not. This leads to the logical critical conclusion that not all the ballots in the batches processed after the database copy were the same and had the same votes as the ballots in the same batches processed before the database copy. There is no record remaining of the votes originally recorded from the ballots, and therefore there can be no certainty that the votes now recorded are the same. In essence, the chain of custody has been broken for these votes in the database.

The 58 batches which were *not* duplicated in the new Adjudication database must also be seen as suspect, as their chain of custody has also been broken via the fact that no record of them or their adjudication exists in the Adjudication database in use at the end of the election. A clerk wishing to view the adjudication status of a ballot in any of the 58 batches would be unable to do so, as no information about those batches exists in the new Adjudication database.

Thus, all 25,931 ballot records processed before 2:14 PM on October 21, 2020, comprising over 25% of the County's total over the entire election, cannot be verified and should not have been counted.

II. Evidence of Ballot Manipulation – April 2021 Grand Junction Municipal Election

Our analysis shows a nearly identical manipulation of the batches and ballots processed during the first six days of ballot processing in the April 2021 Municipal Election in Grand Junction, Colorado.

The timeline of events beginning March 24, 2021, when Mesa County Election clerks began processing ballots in the 2021 Grand Junction Municipal Election, follows.

March 24, 2021 – March 30, 2021, 2:43 PM

On these first seven days of counting, up until 2:43 PM on March 30, 2021, 88 batches of ballots, consisting of 8,540 ballots, were processed. County Election clerks report no unusual activity or errors encountered *at any time* during the election counting process. The Adjudication database used at this time contains records of all batches with a sequential load order of 1 to 88, and other tables within it record each ballot. Those which were selected for Manual Adjudication (339 in total) have the proper status records indicating the normal adjudication steps occurred.

March 30, 2021, 2:58 PM

According to the data contained in the EMS server, new Adjudication and Tabulation databases were created and registered within the DVS system as the associated databases for the election. As in the circumstance previously described in the early voting period for the November 2020 election, these two databases initially contained no data.

See Appendix D for a list of all commands executed prior to and after the database creation in the April 2021 Municipal Election, which provides a precise timeline of the effects of creating the new databases and copying the batch and ballot records.

It is important to note that this unauthorized procedure copied the records of only selected batches of ballots, indicating that this was an intentional act.

Below is a screenshot of the beginning of the list of batches recorded in the original Adjudication database, sorted by the order that they were loaded:

Figure 5. List of Batches Recorded in the Original Adjudication Database, Sorted by Load Order

TabulatorId	BatchId	Category	CvrBatchId	Name	LoadOrder	CreationTime	ModificationTime
30	3000	0	2	Tabulator 30 - Batch 3000	1	2021-03-24 14:52:58.350	2021-03-29 14:07:57.213
30	3001	0	3	Tabulator 30 - Batch 3001	2	2021-03-24 15:09:05.203	2021-03-29 14:07:57.557
30	3002	0	4	Tabulator 30 - Batch 3002	3	2021-03-24 15:10:53.607	2021-03-29 14:07:57.917
30	3003	0	5	Tabulator 30 - Batch 3003	4	2021-03-24 15:13:57.637	2021-03-29 14:07:58.307
30	3004	0	6	Tabulator 30 - Batch 3004	5	2021-03-24 15:17:01.500	2021-03-29 14:07:58.587
30	3005	0	7	Tabulator 30 - Batch 3005	6	2021-03-24 15:21:03.903	2021-03-29 14:07:58.837
30	3006	0	8	Tabulator 30 - Batch 3006	7	2021-03-24 15:24:05.780	2021-03-29 14:07:56.837
30	3007	0	9	Tabulator 30 - Batch 3007	8	2021-03-24 15:28:55.220	2021-03-29 14:07:59.273
30	3008	0	10	Tabulator 30 - Batch 3008	9	2021-03-24 15:33:59.600	2021-03-29 14:07:59.650
30	3009	0	11	Tabulator 30 - Batch 3009	10	2021-03-24 15:37:03.750	2021-03-29 14:08:00.010
30	3010	0	12	Tabulator 30 - Batch 3010	11	2021-03-29 11:06:24.310	2021-03-29 14:08:00.323
30	3011	0	13	Tabulator 30 - Batch 3011	12	2021-03-29 11:38:04.150	2021-03-29 14:08:00.667
30	3012	0	14	Tabulator 30 - Batch 3012	13	2021-03-29 11:42:52.697	2021-03-29 14:08:01.040
30	3013	0	15	Tabulator 30 - Batch 3013	14	2021-03-29 11:45:56.630	2021-03-29 14:08:01.383
30	3014	0	16	Tabulator 30 - Batch 3014	15	2021-03-29 11:48:00.360	2021-03-29 14:08:01.713
30	3015	0	17	Tabulator 30 - Batch 3015	16	2021-03-29 11:51:03.987	2021-03-29 14:08:02.040
30	3016	0	18	Tabulator 30 - Batch 3016	17	2021-03-29 13:05:52.907	2021-03-29 14:08:02.400
30	3017	0	19	Tabulator 30 - Batch 3017	18	2021-03-29 13:08:56.587	2021-03-29 14:08:02.743
30	3018	0	20	Tabulator 30 - Batch 3018	19	2021-03-29 13:12:00.483	2021-03-29 14:08:03.073
30	3019	0	21	Tabulator 30 - Batch 3019	20	2021-03-29 13:14:04.207	2021-03-29 14:08:03.400
30	3020	0	22	Tabulator 30 - Batch 3020	21	2021-03-29 13:19:53.690	2021-03-29 14:08:03.727
30	3021	0	23	Tabulator 30 - Batch 3021	22	2021-03-29 13:22:57.557	2021-03-29 14:08:04.103
30	3022	0	24	Tabulator 30 - Batch 3022	23	2021-03-29 13:26:01.097	2021-03-29 14:08:04.430
30	3023	0	25	Tabulator 30 - Batch 3023	24	2021-03-29 13:28:04.527	2021-03-29 14:08:04.790
30	3024	0	26	Tabulator 30 - Batch 3024	25	2021-03-29 13:30:53.410	2021-03-29 14:08:05.133
30	3025	0	27	Tabulator 30 - Batch 3025	26	2021-03-29 13:32:57.050	2021-03-29 14:08:05.540
30	3026	0	28	Tabulator 30 - Batch 3026	27	2021-03-29 13:37:01.250	2021-03-29 14:08:05.900
30	3027	0	29	Tabulator 30 - Batch 3027	28	2021-03-29 13:40:05.090	2021-03-29 14:08:06.227
30	3028	0	30	Tabulator 30 - Batch 3028	29	2021-03-29 13:41:53.687	2021-03-29 14:08:06.573
30	3029	0	31	Tabulator 30 - Batch 3029	30	2021-03-29 13:44:57.640	2021-03-29 14:08:06.900

Below is a screenshot of the same table in the newly created Adjudication database, sorted by creation time:

Figure 6. List of Batches in the Newly Created Adjudication Database, Sorted by Creation Time

TabulatorId	BatchId	Category	CvrBatchId	Name	LoadOrder	CreationTime	ModificationTime
30	3047	0	49	Tabulator 30 - Batch 3047	48	2021-03-30 15:00:14.560	2021-03-30 15:25:33.373
30	3048	0	50	Tabulator 30 - Batch 3048	49	2021-03-30 15:00:17.950	2021-03-30 15:25:34.090
30	3050	0	52	Tabulator 30 - Batch 3050	51	2021-03-30 15:00:36.577	2021-03-30 15:25:33.733
30	3054	0	56	Tabulator 30 - Batch 3054	55	2021-03-30 15:00:50.780	2021-03-30 15:03:21.940
20	2000	0	59	Tabulator 20 - Batch 2000	58	2021-03-30 15:01:01.250	2021-03-30 15:03:19.520
20	2002	0	61	Tabulator 20 - Batch 2002	60	2021-03-30 15:01:07.983	2021-03-30 15:03:19.847
20	2003	0	62	Tabulator 20 - Batch 2003	61	2021-03-30 15:01:11.467	2021-03-30 15:03:20.050
20	2004	0	63	Tabulator 20 - Batch 2004	62	2021-03-30 15:01:15.123	2021-03-30 15:03:19.127
20	2005	0	64	Tabulator 20 - Batch 2005	63	2021-03-30 15:01:18.687	2021-03-30 15:03:20.237
20	2006	0	65	Tabulator 20 - Batch 2006	64	2021-03-30 15:01:22.203	2021-03-30 15:03:20.410
20	2008	0	67	Tabulator 20 - Batch 2008	66	2021-03-30 15:01:29.203	2021-03-30 15:03:20.627
20	2009	0	68	Tabulator 20 - Batch 2009	67	2021-03-30 15:01:32.953	2021-03-30 15:03:20.847
20	2010	0	69	Tabulator 20 - Batch 2010	68	2021-03-30 15:01:36.467	2021-03-30 15:03:21.033
20	2011	0	70	Tabulator 20 - Batch 2011	69	2021-03-30 15:01:40.437	2021-03-30 15:03:21.237
20	2012	0	71	Tabulator 20 - Batch 2012	70	2021-03-30 15:01:44.513	2021-03-30 15:03:21.410
20	2013	0	72	Tabulator 20 - Batch 2013	71	2021-03-30 15:01:48.063	2021-03-30 15:03:21.567
20	2015	0	74	Tabulator 20 - Batch 2015	73	2021-03-30 15:01:55.170	2021-03-30 15:03:21.723
20	2016	0	75	Tabulator 20 - Batch 2016	74	2021-03-30 15:01:58.827	2021-03-30 15:03:39.753
20	2018	0	77	Tabulator 20 - Batch 2018	76	2021-03-30 15:02:05.920	2021-03-30 15:03:47.913
20	2019	0	78	Tabulator 20 - Batch 2019	77	2021-03-30 15:02:09.407	2021-03-30 15:03:48.210
20	2020	0	79	Tabulator 20 - Batch 2020	78	2021-03-30 15:02:13.140	2021-03-30 15:03:47.613
20	2021	0	80	Tabulator 20 - Batch 2021	79	2021-03-30 15:02:16.017	2021-03-30 15:03:51.317
20	2023	0	82	Tabulator 20 - Batch 2023	81	2021-03-30 15:02:22.517	2021-03-30 15:04:13.667
20	2024	0	83	Tabulator 20 - Batch 2024	82	2021-03-30 15:02:26.050	2021-03-30 15:04:18.870
20	2025	0	84	Tabulator 20 - Batch 2025	83	2021-03-30 15:02:29.580	2021-03-30 15:04:18.527
20	2026	0	85	Tabulator 20 - Batch 2026	84	2021-03-30 15:02:33.597	2021-03-30 15:04:20.793
10	1001	0	87	Tabulator 10 - Batch 1001	86	2021-03-30 15:02:40.610	2021-03-30 15:04:48.777
10	1002	0	88	Tabulator 10 - Batch 1002	87	2021-03-30 15:02:44.050	2021-03-30 15:04:49.120
10	1003	0	89	Tabulator 10 - Batch 1003	88	2021-03-30 15:02:47.423	2021-03-30 15:04:48.417
10	1000	0	96	Tabulator 10 - Batch 1000	114	2021-03-30 16:04:55.357	2021-03-30 16:17:49.813

March 30, 2021, 3:00 PM – 3:03 PM

During this three-minute time period, records of 42 batches and the 4,082 ballots contained within them, previously processed into the original Adjudication database, were copied into the new Adjudication database. According to the time stamps, the records of the batches appeared in the new Adjudication database in intervals of a fraction of a second between them, much too quickly for the ballots contained in the batches to have been physically scanned (per the maximum scanning speeds discussed above). Mesa County election clerks state that they did not take any action to reprocess or re-scan any batches on that day, nor did they at any time stop and restart the Adjudication software process. Only 39 ballots in these 42 batches went through manual adjudication after being copied to the new database, and database records indicate that the adjudication process was completed successfully on those 39 ballots.

Unlike what was found in the November 2020 General Election records described above, the records for these 42 batches which were copied to the new database do not appear in the new Adjudication database in exactly the same order as they had originally been loaded; 12 batch records are out of order when the records in the original Adjudication database and the new Adjudication database are compared.

No further anomalies are shown in the Adjudication database records during the Election counting process, nor did Mesa County election clerks encounter any unexpected issues.

Of the 42 batches which were processed twice (batches 45 through 49 and 51 through 88), the ballot counts (total number of ballots) match between the old and new Adjudication databases. However, DVS software sent 339 ballots to manual adjudication the first time they were processed in the original Adjudication database, but when reprocessed in the new Adjudication database the software sent just 39 ballots to manual adjudication.

The same ballots run through the same hardware and evaluated by the same software should have had the same resulting ballots marked for adjudication, but they did not. This leads to the logical critical conclusion that not all the ballots in the batches processed after the database copy were the same and had the same votes as the ballots in the same batches processed before the database copy. There is no record remaining of the votes originally recorded from the ballots, and therefore there can be no certainty that the votes now recorded are the same. In essence, the chain of custody has been broken for these votes in the database.

The 46 batches which were *not* duplicated in the new Adjudication database must also be seen as suspect, as their chain of custody has also been broken via the fact that no record of them or their adjudication exists in the Adjudication database in use at the end of the election. A clerk wishing to view the adjudication status of a ballot in any of these 46 batches would be unable to do

so, as no information about these batches exists in the new Adjudication database.

Thus, all 8,540 ballots processed before 2:58 PM on March 30, 2021, comprising over 49% of the total votes in the entire 2021 Grand Junction Municipal Election, cannot be verified and should not have been counted. These 8,540 ballots represent more than twice the winning margin in any of the four City Council races that occurred in this election.

III. Comparison of the November 2020 General Election Findings and the April 2021 Grand Junction Municipal Election Findings

Comparing the above findings for the two elections shows numerous similarities and also critical differences.

Similarities:

- In both elections, a software process running within the DVS system performed an unauthorized creation of new Adjudication and Tabulation databases.
- In both elections, database records of selected batches and of the ballots within those batches were copied into the new databases and were reprocessed.
- In both elections, selected batches were not copied to the new Adjudication and Tabulation databases, making adjudication information invisible to the Mesa County election clerks.

Differences:

- In the November 2020 General Election, records of a sequential series of batches and the ballots contained within them were copied from the original Adjudication and Tabulation databases to the new Adjudication and Tabulation databases, and the batches were copied in the same order as in the original databases. In the April 2021 Grand Junction Municipal Election, records of a non-sequential series of batches and the ballots contained within

them were copied, and they appear in the new Adjudication database in a different order than in the original database.
- In the April 2021 Grand Junction Municipal Election, the EMS User Logs (which show events and commands which were executed) show reference to a batch 89. As there were 88 batches in the original Adjudication database, this would have logically been the next batch received from the scanners. However, no record of a batch with the load order '89' exists in either Adjudication database, and there are missing load orders between 88 and 114 as well.

The similarities lead to the conclusion that the same method was used to alter the database records in both elections.

The differences lead to the conclusion that there is a degree of control in the method used to alter the database records which used parameters unique to each election.

IV. Summary Impact of Above Findings

This manipulation of batch and ballot records described above is significant for three reasons.

First, when the ballots were reprocessed as described above, including re-adjudication, it is logical to conclude that whatever votes had been initially recorded could well have been replaced by the reprocessed votes in the Main election database. The differences in the Manual Adjudication numbers certainly supports this possible conclusion. Thus, this procedure could change votes in the Main database without leaving any evidence to indicate changes had been made, or any way to determine the nature of the changes or what the original vote data was.

Second, the adjudication status (including the timestamps of adjudication events, the results of the adjudication, and the user who performed the

adjudication) of any ballots in the batches not copied to the new Adjudication database would not be viewable through the DVS client software applications.

Third, an examination of the EMS server which was less rigorous than ours would not likely have caught the fact that the Adjudication and Tabulation databases used at the end of the elections were not the same, nor did they contain the same records, as the databases used at the beginning of the elections. This leads to the possible conclusion that some batches and ballots were excluded from the new databases so as to inhibit the possibility of their being audited or examined.

V. Lack of Referential Integrity in DVS Database Tables

Most modern database designs include a concept called “referential integrity.” For example, if you have one table of data that has information about “people,” and another table that has information about “colleges,” you might have a field in an individual record in the “people’s” table that can contain an id, or pointer, to the college he or she attended. Referential integrity, in this case, would mean that if “John Smith’s” record had a pointer to the “University of Pittsburgh”, the system should give an error if you try to remove the item “University of Pittsburgh” from the “colleges” table. It would not allow you to do this action because a field in “John Smith’s” record refers to the college “University of Pittsburgh” and deleting that entry in the “colleges” table makes “John Smith’s” record invalid.

However, some of the DVS Election Management System data structures have no such referential integrity built into them. Therefore, batch records in one database could be deleted without any consequence to records that point to that batch in another database, and without any detection of the error. This lack of referential integrity means that vote or ballot information could easily be added or removed from one part of the database without any warnings or errors occurring in other parts of the database.

It is, for example, possible to change the fields with vote counts in one table of the Main election database without having that change affect any other tables

or cause a referential integrity violation. This is a fundamental and critical breach of sound database design, particularly considering the importance of chain-of-custody and audit trail evidence for the provenance of ballot record and tabulated vote information in a voting system.

Please see Reference D for an example of how the batch and ballot data moves through the various databases and tables in the Dominion EMS.

VI. Digital Ballot Images are Obfuscated and Unverifiable

An attempt was made to investigate the conditions of the digital ballot images to corroborate the findings above. This avenue of research is greatly hindered because the ballot IDs or sequence numbers in the batches are not relatable to their images, not even within the DVS databases themselves. This is an additional example of a lack of referential integrity within the system.

Additionally, the digital ballot images do not have the accompanying ".sha" files which are meant to prove the authenticity of the ballots. Therefore, any findings, including the integrity and authenticity of ballot images, related to the digital ballot images cannot be absolutely validated because there is no proof that the images are the ones created at the time the ballots were first processed.

Finally, code running within the EMS server that has the system access rights to create and alter SQL Server database records could be used to alter the stored digital ballot images themselves. The EMS Adjudication module software already has the capability of altering scanned images and legitimately does so for each manually adjudicated ballot.

DISCUSSION

The events described above show a significant manipulation of a large number of batch, ballot, and vote records in the DVS EMS Database in Mesa County, and there are only a few possible explanations for the manipulation.

1. Human Error

Extensive questioning of Mesa County election clerks has ruled out human error as the reason for the unauthorized creation of election databases on October 21, 2020, followed by reprocessing of 20,346 ballots. These personnel have a strong recollection of the events of October 21, 2020, and because of the timelines established both by their recollection and corresponding database time stamps, it is evident that any and all unusual actions they might have taken on that day were in response to the new database's creation having already occurred, and batch records being copied into the new database, which affected their ability to complete adjudication on some in-process ballots. Similarly, Mesa County election officials have a strong recollection of the events of March 30, 2021. They state that they did not take any steps that would have given rise to the unauthorized creation of new election databases during the 2021 Grand Junction Municipal Election on that day, followed by the reprocessing of 2,974 ballots,.

2. Software Failure

While an error or failure in the DVS EMS server is a possibility, it strains credulity that any error could cause the numerous specific events which are documented above. In particular, the non-sequential reloading of the batches during the 2021 Grand Junction Municipal Election, when compared with the sequential reloading in the November 2020 General Election, makes it inherently impossible for the same error to have caused both chains of events.

However, as noted in the section above labelled "Algorithmically Triggered", the DVS EMS server (or another connected machine running Dominion software) could have been preprogrammed to perform the unauthorized new database creations and the selected record manipulation which followed based on preprogrammed criteria related to the election results at the time. This would be the result of advance planning in the deliberate

design of the software to alter outcomes when unexpected voting patterns were detected.

3. **Network Breach or Pre-Installation of Manipulating Software or Algorithm**

   A device external to the DVS D-Suite network could have connected to the DVS D-Suite and to the EMS server, using the open SQL Server port 1433, open Web Services port 80, or through any other open port directly into the DVS Software. As outlined in "Report #2, Forensic Examination and Analysis Report" by D. Gould, there are numerous flaws in the security of the server, many of which could provide an outside entity with direct access to the SQL Server Database or the Application itself. The DVS D-Suite makes use of "SOAP" messaging protocol API calls through its web server, so malicious procedures could be triggered by simple port 80 access.

   As *all* Windows log files which would show these accesses are configured, as specified by DVS manuals published by the Colorado Secretary of State as mandatory technical procedures for County election officials, to keep only a small amount of log entries before they are overwritten, no record of external access to the DVS D-Suite is available in system logs.

   Regardless of whether the voting system was connected to an external network or device, even momentarily, or whether a pre-installed software or algorithm was triggered by an external command or complex set of variable conditions, the execution of manipulating software or algorithm could plausibly be responsible for the results described in our findings.

CONCLUSIONS

1. Unauthorized creation of new Tabulation and Adjudication databases occurred during the counting of the November 2020 General Election, along with the selective copying of batch and ballot records from the original databases to the new ones. This manipulation places all 25,913 initial ballots counted into a state where they cannot be validated – some because

it is possible that their vote information was changed, and unverifiable that it was not, and the rest because their “chain of evidence” has been intentionally obfuscated. Even if the count and content of ballot images match the numbers and counts reported by the database, there is no method to validate those ballot images due to missing “.sha” files, which are intended to provide such validation.

2. Unauthorized creation of new Tabulation and Adjudication databases occurred during the 2021 Grand Junction Municipal Election, along with the selective copying of batch and ballot records from the original databases to the new ones. This places all 8,540 initial ballots counted into a state where they cannot be validated – some because it is possible that their vote information was changed and unverifiable that it was not, and some because their “chain of evidence” has been intentionally obfuscated.

3. As we have found evidence that thousands of ballot records have had their validity placed in serious question, none of the election results from the 2020 General or 2021 Grand Junction Municipal Elections in Mesa County can be considered trustworthy. If Mesa County has preserved the respective paper ballots, as they are required to do by law, and those ballots were forensically authenticated with confirmed chain-of-custody from eligible electors to sworn county election officials (not possible retrospectively, nor under current election procedures in Colorado), then a hand-count of paper ballots might support a verifiable, trustworthy conclusion about the county-level results of these two elections.

4. Because the unauthorized methods used to alter batch and ballot-level information described above are available within the DVS EMS server, this system cannot be considered reliable for use in any election. An investigation, involving all physical and cyber evidence, including a source code audit of the exact, verifiable version of all DVS-supplied executable and library files, is necessary to identify the exact software methods used to produce the manipulation and to determine other potential unauthorized actions that the code is able to cause or enable.

5. The Dominion Voting System's database structure stores actual vote information in only one table, in aggregated form, so alterations made to vote counts or candidates in just that table, create a single point of attack or failure for the entire vote reporting process (see Reference D).

APPENDIX A – BATCHES - ORIGINAL ADJUDICATION DATABASE (WITH TIME BETWEEN BATCHES) – NOVEMBER 2020 ELECTION

TabulatorId	BatchId	LoadOrder	Creation Date	Creation Time	Difference from Prior	Ballots
10	4001	1	10/19/2020	12:07:41 PM		100
10	4002	2	10/19/2020	12:07:44 PM	0:00:04	42
10	4003	3	10/19/2020	12:07:48 PM	0:00:04	100
10	4004	4	10/19/2020	12:07:51 PM	0:00:03	100
10	4005	5	10/19/2020	12:07:54 PM	0:00:03	100
10	4006	6	10/19/2020	12:08:12 PM	0:00:18	100
10	4007	7	10/19/2020	12:08:16 PM	0:00:04	100
10	4008	8	10/19/2020	12:08:20 PM	0:00:04	99
10	4009	9	10/19/2020	12:08:24 PM	0:00:04	100
10	4010	10	10/19/2020	12:08:28 PM	0:00:04	100
4	2001	11	10/19/2020	12:23:35 PM	0:15:07	98
4	2002	12	10/19/2020	12:30:26 PM	0:06:50	100
4	2003	13	10/19/2020	12:32:30 PM	0:02:04	100
4	2004	14	10/19/2020	12:36:20 PM	0:03:50	100
4	2005	15	10/19/2020	12:43:25 PM	0:07:05	100
4	2006	16	10/19/2020	1:50:29 PM	1:07:03	100
4	2007	17	10/19/2020	1:54:19 PM	0:03:50	100
4	2008	18	10/19/2020	1:58:24 PM	0:04:05	100
4	2009	19	10/19/2020	2:03:29 PM	0:05:05	100
4	2010	20	10/19/2020	2:06:33 PM	0:03:05	100
4	2011	21	10/19/2020	2:10:23 PM	0:03:50	99
4	2012	22	10/19/2020	2:14:28 PM	0:04:05	100
4	2013	23	10/19/2020	2:18:33 PM	0:04:05	100
4	2014	24	10/19/2020	2:22:23 PM	0:03:50	100
4	2015	25	10/19/2020	2:26:27 PM	0:04:05	100
4	2016	26	10/19/2020	2:30:32 PM	0:04:05	100
4	2017	27	10/19/2020	2:34:23 PM	0:03:51	100
4	2018	28	10/19/2020	2:36:27 PM	0:02:04	33
4	2019	29	10/19/2020	2:40:30 PM	0:04:03	100
4	2020	30	10/19/2020	2:44:20 PM	0:03:51	100
4	2021	31	10/19/2020	2:48:25 PM	0:04:05	99
4	2022	32	10/19/2020	2:51:29 PM	0:03:04	100
4	2023	33	10/19/2020	2:57:21 PM	0:05:52	99
4	2024	34	10/19/2020	2:59:26 PM	0:02:04	100
10	4011	35	10/19/2020	3:05:31 PM	0:06:05	100
4	2025	36	10/19/2020	3:06:20 PM	0:00:49	100
10	4012	37	10/19/2020	3:09:24 PM	0:03:05	100
4	2026	38	10/19/2020	3:10:28 PM	0:01:04	99
10	4013	39	10/19/2020	3:12:33 PM	0:02:04	100

4	2027	40	10/19/2020	3:15:22 PM	0:02:49	100
10	4014	41	10/19/2020	3:16:26 PM	0:01:04	100
4	2028	42	10/19/2020	3:17:33 PM	0:01:07	100
4	2029	43	10/19/2020	3:19:22 PM	0:01:49	25
4	2030	44	10/19/2020	3:22:24 PM	0:03:02	100
10	4015	45	10/19/2020	3:24:29 PM	0:02:04	99
4	2031	46	10/19/2020	3:29:34 PM	0:05:05	100
10	4016	47	10/19/2020	3:30:23 PM	0:00:49	100
4	2032	48	10/19/2020	3:34:28 PM	0:04:05	99
10	4017	49	10/19/2020	3:34:32 PM	0:00:04	100
10	4018	50	10/19/2020	3:40:22 PM	0:05:50	100
10	4019	51	10/19/2020	3:44:26 PM	0:04:05	100
10	4020	52	10/19/2020	3:48:31 PM	0:04:05	99
4	2033	53	10/19/2020	4:00:24 PM	0:11:53	100
10	4021	54	10/19/2020	4:00:43 PM	0:00:19	99
4	2034	55	10/19/2020	4:03:33 PM	0:02:49	100
10	4022	56	10/19/2020	4:03:37 PM	0:00:04	100
10	4023	57	10/19/2020	4:05:26 PM	0:01:49	78
4	2035	58	10/19/2020	4:09:30 PM	0:04:04	100
10	4024	59	10/19/2020	4:09:34 PM	0:00:04	100
10	4025	60	10/19/2020	4:12:24 PM	0:02:50	99
4	2036	61	10/19/2020	4:13:28 PM	0:01:04	100
10	4026	62	10/19/2020	4:15:33 PM	0:02:04	100
4	2037	63	10/19/2020	4:16:22 PM	0:00:49	99
10	4027	64	10/19/2020	4:19:26 PM	0:03:04	100
10	4028	65	10/19/2020	4:22:31 PM	0:03:05	100
4	2038	66	10/19/2020	4:24:20 PM	0:01:50	100
4	2039	67	10/19/2020	4:28:25 PM	0:04:05	100
4	2040	68	10/19/2020	4:32:30 PM	0:04:05	100
4	2041	69	10/19/2020	4:35:22 PM	0:02:52	100
10	4029	70	10/19/2020	4:37:26 PM	0:02:04	100
4	2042	71	10/19/2020	4:41:31 PM	0:04:05	99
10	4030	72	10/19/2020	4:41:35 PM	0:00:04	99
4	2043	73	10/19/2020	4:46:25 PM	0:04:50	100
10	4031	74	10/19/2020	4:49:29 PM	0:03:05	97
10	4032	75	10/19/2020	4:53:34 PM	0:04:05	99
10	4033	76	10/19/2020	4:55:23 PM	0:01:49	100
4	2044	77	10/19/2020	4:57:27 PM	0:02:04	100
10	4034	78	10/19/2020	4:58:32 PM	0:01:04	100
4	2045	79	10/19/2020	5:00:22 PM	0:01:51	99
10	4035	80	10/19/2020	5:03:27 PM	0:03:05	100
10	4036	81	10/19/2020	5:06:31 PM	0:03:05	100
4	2046	82	10/19/2020	5:16:22 PM	0:09:51	99
4	2047	83	10/19/2020	5:18:28 PM	0:02:06	100
10	4037	84	10/19/2020	5:18:32 PM	0:00:04	100

4	2048	85	10/19/2020	5:22:22 PM	0:03:51	99
4	2049	86	10/20/2020	10:05:36 AM	16:43:13	100
4	2050	87	10/20/2020	10:07:25 AM	0:01:49	100
4	2051	88	10/20/2020	10:10:30 AM	0:03:05	99
4	2052	89	10/20/2020	10:13:35 AM	0:03:05	100
4	2053	90	10/20/2020	10:17:26 AM	0:03:51	100
4	2054	91	10/20/2020	10:29:33 AM	0:12:06	100
4	2055	92	10/20/2020	10:32:37 AM	0:03:05	100
4	2056	93	10/20/2020	10:40:29 AM	0:07:51	100
4	2057	94	10/20/2020	10:43:33 AM	0:03:05	100
4	2058	95	10/20/2020	10:50:38 AM	0:07:05	99
4	2059	96	10/20/2020	10:53:28 AM	0:02:49	100
4	2060	97	10/20/2020	10:56:32 AM	0:03:05	100
4	2061	98	10/20/2020	10:59:37 AM	0:03:04	100
4	2062	99	10/20/2020	11:02:27 AM	0:02:50	98
4	2063	100	10/20/2020	11:05:31 AM	0:03:04	100
4	2064	101	10/20/2020	11:08:36 AM	0:03:05	100
4	2065	102	10/20/2020	11:11:26 AM	0:02:50	100
4	2066	103	10/20/2020	11:16:31 AM	0:05:05	100
4	2067	104	10/20/2020	11:19:35 AM	0:03:05	100
4	2068	105	10/20/2020	11:22:40 AM	0:03:05	100
4	2069	106	10/20/2020	11:26:29 AM	0:03:50	99
4	2070	107	10/20/2020	11:30:34 AM	0:04:05	94
4	2071	108	10/20/2020	11:33:38 AM	0:03:04	100
4	2072	109	10/20/2020	11:38:28 AM	0:04:50	100
4	2073	110	10/20/2020	11:43:33 AM	0:05:05	100
10	4038	111	10/20/2020	11:43:37 AM	0:00:04	99
10	4039	112	10/20/2020	11:48:28 AM	0:04:50	100
10	4040	113	10/20/2020	11:50:32 AM	0:02:04	99
4	2074	114	10/20/2020	11:52:36 AM	0:02:04	99
10	4041	115	10/20/2020	11:55:28 AM	0:02:51	100
4	2075	116	10/20/2020	11:56:32 AM	0:01:04	99
10	4042	117	10/20/2020	11:58:36 AM	0:02:04	100
10	4043	118	10/20/2020	12:02:26 PM	0:03:50	100
10	4044	119	10/20/2020	12:05:31 PM	0:03:05	100
4	2076	120	10/20/2020	12:07:35 PM	0:02:04	100
4	2077	121	10/20/2020	12:13:40 PM	0:06:05	100
10	4045	122	10/20/2020	12:13:44 PM	0:00:04	100
4	2078	123	10/20/2020	12:16:34 PM	0:02:50	99
10	4046	124	10/20/2020	12:16:38 PM	0:00:04	100
4	2079	125	10/20/2020	12:19:28 PM	0:02:50	99
10	4047	126	10/20/2020	12:19:46 PM	0:00:19	99
4	2080	127	10/20/2020	12:22:36 PM	0:02:49	99
10	4048	128	10/20/2020	12:22:40 PM	0:00:04	100
4	2081	129	10/20/2020	12:25:30 PM	0:02:50	100

10	4049	130	10/20/2020	12:30:35 PM	0:05:05	100
4	2082	131	10/20/2020	1:08:32 PM	0:37:57	100
10	4050	132	10/20/2020	1:11:37 PM	0:03:05	100
4	2083	133	10/20/2020	1:14:29 PM	0:02:52	100
10	4051	134	10/20/2020	1:14:48 PM	0:00:19	100
4	2084	135	10/20/2020	1:16:37 PM	0:01:49	100
10	4052	136	10/20/2020	1:18:26 PM	0:01:49	100
4	2085	137	10/20/2020	1:20:30 PM	0:02:04	100
10	4053	138	10/20/2020	1:21:35 PM	0:01:04	100
4	2086	139	10/20/2020	1:22:39 PM	0:01:04	100
10	4054	140	10/20/2020	1:27:29 PM	0:04:50	100
10	4055	141	10/20/2020	1:31:34 PM	0:04:05	100
4	2087	142	10/20/2020	1:33:38 PM	0:02:04	98
4	2088	143	10/20/2020	1:37:28 PM	0:03:50	100
10	4056	144	10/20/2020	1:47:34 PM	0:10:06	97
4	2089	145	10/20/2020	1:53:39 PM	0:06:05	100
4	2090	146	10/20/2020	1:58:30 PM	0:04:51	99
10	4057	147	10/20/2020	2:02:35 PM	0:04:05	96
4	2091	148	10/20/2020	2:04:39 PM	0:02:04	100
4	2092	149	10/20/2020	2:07:29 PM	0:02:49	100
10	4058	150	10/20/2020	2:07:47 PM	0:00:18	97
4	2093	151	10/20/2020	2:10:37 PM	0:02:49	100
4	2094	152	10/20/2020	2:14:28 PM	0:03:52	100
4	2095	153	10/20/2020	2:19:33 PM	0:05:05	97
4	2096	154	10/20/2020	2:23:38 PM	0:04:05	99
7	3001	155	10/20/2020	2:26:28 PM	0:02:50	100
7	3002	156	10/20/2020	2:29:32 PM	0:03:05	100
4	2097	157	10/20/2020	2:31:37 PM	0:02:04	100
7	3003	158	10/20/2020	2:32:26 PM	0:00:49	100
7	3004	159	10/20/2020	2:36:31 PM	0:04:05	98
4	2098	160	10/20/2020	2:38:36 PM	0:02:05	100
7	3005	161	10/20/2020	2:39:40 PM	0:01:04	98
4	2099	162	10/20/2020	2:42:29 PM	0:02:49	99
7	3006	163	10/20/2020	2:43:33 PM	0:01:04	100
4	2100	164	10/20/2020	2:46:38 PM	0:03:05	100
7	3007	165	10/20/2020	2:47:27 PM	0:00:49	100
4	2101	166	10/20/2020	2:49:32 PM	0:02:05	100
7	3008	167	10/20/2020	2:51:36 PM	0:02:04	100
4	2102	168	10/20/2020	2:57:29 PM	0:05:53	97
7	3009	169	10/20/2020	2:57:47 PM	0:00:18	95
4	2103	170	10/20/2020	3:00:37 PM	0:02:49	99
7	3010	171	10/20/2020	3:01:41 PM	0:01:04	99
4	2104	172	10/20/2020	3:04:33 PM	0:02:53	99
7	3011	173	10/20/2020	3:04:37 PM	0:00:04	98
4	2105	174	10/20/2020	3:07:28 PM	0:02:51	100

7	3012	175	10/20/2020	3:07:47 PM	0:00:19	100
7	3013	176	10/20/2020	3:10:36 PM	0:02:50	98
4	2106	177	10/20/2020	3:12:41 PM	0:02:04	95
7	3014	178	10/20/2020	3:13:30 PM	0:00:49	100
4	2107	179	10/20/2020	3:21:35 PM	0:08:06	95
7	3015	180	10/20/2020	3:23:39 PM	0:02:04	94
4	2108	181	10/20/2020	3:25:30 PM	0:01:50	100
7	3016	182	10/20/2020	3:25:50 PM	0:00:20	100
7	3017	183	10/20/2020	3:29:39 PM	0:03:50	100
4	2109	184	10/20/2020	3:30:29 PM	0:00:50	94
7	3018	185	10/20/2020	3:35:34 PM	0:05:05	97
7	3019	186	10/20/2020	3:59:43 PM	0:24:09	95
2	1001	187	10/20/2020	4:02:32 PM	0:02:49	75
7	3020	188	10/20/2020	4:03:36 PM	0:01:03	99
7	3021	189	10/20/2020	4:06:40 PM	0:03:04	99
7	3022	190	10/20/2020	4:10:30 PM	0:03:50	99
7	3023	191	10/20/2020	4:13:34 PM	0:03:05	100
7	3024	192	10/20/2020	4:22:40 PM	0:09:06	99
4	2110	193	10/20/2020	4:24:30 PM	0:01:50	98
7	3025	194	10/20/2020	4:26:34 PM	0:02:04	100
4	2111	195	10/20/2020	4:28:38 PM	0:02:04	100
7	3026	196	10/20/2020	4:29:30 PM	0:00:51	100
7	3027	197	10/20/2020	4:34:35 PM	0:05:05	100
7	3028	198	10/20/2020	4:38:39 PM	0:04:05	100
2	1002	199	10/20/2020	4:46:29 PM	0:07:50	21
2	1003	200	10/20/2020	4:56:34 PM	0:10:04	75
2	1004	201	10/20/2020	4:57:37 PM	0:01:03	80
4	2112	202	10/21/2020	9:05:41 AM	16:08:04	99
4	2113	203	10/21/2020	9:07:45 AM	0:02:04	95
4	2114	204	10/21/2020	9:10:35 AM	0:02:50	98
4	2115	205	10/21/2020	9:14:40 AM	0:04:05	96
4	2116	206	10/21/2020	9:18:45 AM	0:04:05	99
4	2117	207	10/21/2020	9:20:34 AM	0:01:49	100
4	2118	208	10/21/2020	9:22:38 AM	0:02:04	100
4	2119	209	10/21/2020	9:28:44 AM	0:06:05	100
4	2120	210	10/21/2020	9:33:35 AM	0:04:51	100
4	2121	211	10/21/2020	9:36:39 AM	0:03:04	100
4	2122	212	10/21/2020	9:39:44 AM	0:03:05	100
4	2123	213	10/21/2020	9:42:35 AM	0:02:51	100
2	1005	214	10/21/2020	9:50:40 AM	0:08:05	68
2	1006	215	10/21/2020	9:54:43 AM	0:04:03	37
2	1007	216	10/21/2020	9:56:45 AM	0:02:02	76
2	1008	217	10/21/2020	10:14:37 AM	0:17:51	14
10	4059	218	10/21/2020	10:16:39 AM	0:02:02	100
10	4060	219	10/21/2020	10:25:45 AM	0:09:06	100

10	4061	220	10/21/2020	10:28:34 AM	0:02:50	100
10	4062	221	10/21/2020	10:32:39 AM	0:04:05	98
10	4063	222	10/21/2020	10:34:43 AM	0:02:04	100
10	4064	223	10/21/2020	10:37:35 AM	0:02:51	100
10	4065	224	10/21/2020	10:40:39 AM	0:03:05	100
10	4066	225	10/21/2020	10:43:44 AM	0:03:05	100
10	4067	226	10/21/2020	10:47:34 AM	0:03:51	99
10	4068	227	10/21/2020	10:51:39 AM	0:04:05	100
10	4069	228	10/21/2020	10:56:44 AM	0:05:05	100
10	4070	229	10/21/2020	10:59:33 AM	0:02:49	100
10	4071	230	10/21/2020	11:02:38 AM	0:03:05	99
10	4072	231	10/21/2020	11:05:42 AM	0:03:05	100
10	4073	232	10/21/2020	11:21:37 AM	0:15:54	100
10	4074	233	10/21/2020	11:27:42 AM	0:06:05	100
10	4075	234	10/21/2020	11:35:47 AM	0:08:06	97
10	4076	235	10/21/2020	11:38:37 AM	0:02:49	100
10	4077	236	10/21/2020	11:41:41 AM	0:03:05	100
10	4078	237	10/21/2020	11:46:46 AM	0:05:05	100
10	4079	238	10/21/2020	11:49:36 AM	0:02:50	100
10	4080	239	10/21/2020	11:53:41 AM	0:04:05	100
10	4081	240	10/21/2020	12:00:46 PM	0:07:05	101
10	4082	241	10/21/2020	12:02:36 PM	0:01:49	100
10	4083	242	10/21/2020	12:05:40 PM	0:03:05	100
10	4084	243	10/21/2020	12:08:45 PM	0:03:05	100
10	4085	244	10/21/2020	12:12:35 PM	0:03:50	100
10	4086	245	10/21/2020	12:15:40 PM	0:03:05	98
10	4087	246	10/21/2020	12:50:37 PM	0:34:57	98
10	4088	247	10/21/2020	12:53:41 PM	0:03:04	95
10	4089	248	10/21/2020	12:55:46 PM	0:02:04	97
10	4090	249	10/21/2020	12:58:35 PM	0:02:50	98
10	4091	250	10/21/2020	1:03:42 PM	0:05:06	95
10	4092	251	10/21/2020	1:08:46 PM	0:05:05	98
10	4093	252	10/21/2020	1:10:37 PM	0:01:50	98
10	4094	253	10/21/2020	1:13:41 PM	0:03:04	95
10	4095	254	10/21/2020	1:16:45 PM	0:03:04	97
10	4096	255	10/21/2020	1:19:35 PM	0:02:50	96
10	4097	256	10/21/2020	1:23:41 PM	0:04:05	95
10	4098	257	10/21/2020	1:28:46 PM	0:05:05	100
10	4099	258	10/21/2020	1:29:34 PM	0:00:49	63
10	4100	259	10/21/2020	1:34:38 PM	0:05:04	100
7	3029	260	10/21/2020	1:50:46 PM	0:16:07	100
7	3030	261	10/21/2020	1:53:37 PM	0:02:51	100
7	3031	262	10/21/2020	1:57:42 PM	0:04:05	100
7	3032	263	10/21/2020	2:00:50 PM	0:03:09	100
7	3033	264	10/21/2020	2:05:40 PM	0:04:50	100

7	3034	265	10/21/2020	2:08:46 PM	0:03:05	100
7	3035	266	10/21/2020	2:11:35 PM	0:02:50	100
7	3036	267	10/21/2020	2:14:40 PM	0:03:05	99

APPENDIX B – BATCHES - NEW ADJUDICATION DATABASE (WITH TIME BETWEEN BATCHES) – NOVEMBER 2020 ELECTION

TabulatorId	BatchId	CvrBatchId	LoadOrder	Creation Date	Creation Time	Difference from Prior	Ballots
10	4001	1	1	10/21/2020	2:20:07 PM		100
10	4025	60	60	10/21/2020	2:20:26 PM	0:00:19	99
4	2036	61	61	10/21/2020	2:20:30 PM	0:00:04	100
10	4026	62	62	10/21/2020	2:20:34 PM	0:00:04	100
4	2037	63	63	10/21/2020	2:20:39 PM	0:00:05	99
10	4027	64	64	10/21/2020	2:20:43 PM	0:00:04	100
10	4028	65	65	10/21/2020	2:20:47 PM	0:00:04	100
4	2038	66	66	10/21/2020	2:20:52 PM	0:00:04	100
4	2039	67	67	10/21/2020	2:20:56 PM	0:00:04	100
4	2040	68	68	10/21/2020	2:21:00 PM	0:00:04	100
4	2041	69	69	10/21/2020	2:21:04 PM	0:00:04	100
10	4029	70	70	10/21/2020	2:21:08 PM	0:00:04	100
4	2042	71	71	10/21/2020	2:21:13 PM	0:00:04	99
10	4030	72	72	10/21/2020	2:21:17 PM	0:00:04	99
4	2043	73	73	10/21/2020	2:21:21 PM	0:00:04	100
10	4031	74	74	10/21/2020	2:21:25 PM	0:00:04	97
10	4032	75	75	10/21/2020	2:21:29 PM	0:00:05	99
10	4033	76	76	10/21/2020	2:21:34 PM	0:00:04	100
4	2044	77	77	10/21/2020	2:21:40 PM	0:00:06	100
10	4034	78	78	10/21/2020	2:21:44 PM	0:00:04	100
4	2045	79	79	10/21/2020	2:21:48 PM	0:00:04	99
10	4035	80	80	10/21/2020	2:21:53 PM	0:00:05	100
10	4036	81	81	10/21/2020	2:21:57 PM	0:00:04	100
4	2046	82	82	10/21/2020	2:22:01 PM	0:00:04	99
4	2047	83	83	10/21/2020	2:22:05 PM	0:00:04	100
10	4037	84	84	10/21/2020	2:22:09 PM	0:00:04	100
4	2048	85	85	10/21/2020	2:22:13 PM	0:00:04	99
4	2049	86	86	10/21/2020	2:22:17 PM	0:00:04	100
4	2050	87	87	10/21/2020	2:22:21 PM	0:00:04	100
4	2051	88	88	10/21/2020	2:22:25 PM	0:00:04	99
4	2052	89	89	10/21/2020	2:22:30 PM	0:00:04	100
4	2053	90	90	10/21/2020	2:22:34 PM	0:00:04	100
4	2054	91	91	10/21/2020	2:22:38 PM	0:00:04	100
4	2055	92	92	10/21/2020	2:22:42 PM	0:00:04	100
4	2056	93	93	10/21/2020	2:22:46 PM	0:00:04	100
4	2057	94	94	10/21/2020	2:22:50 PM	0:00:04	100
4	2058	95	95	10/21/2020	2:22:54 PM	0:00:04	99
4	2059	96	96	10/21/2020	2:22:58 PM	0:00:04	100
4	2060	97	97	10/21/2020	2:23:02 PM	0:00:04	100

4	2061	98	98	10/21/2020	2:23:06 PM	0:00:04	100
4	2062	99	99	10/21/2020	2:23:11 PM	0:00:04	98
4	2063	100	100	10/21/2020	2:23:15 PM	0:00:04	100
4	2064	101	101	10/21/2020	2:23:19 PM	0:00:04	100
4	2065	102	102	10/21/2020	2:23:23 PM	0:00:04	100
4	2066	103	103	10/21/2020	2:23:28 PM	0:00:05	100
4	2067	104	104	10/21/2020	2:23:34 PM	0:00:06	100
4	2068	105	105	10/21/2020	2:23:38 PM	0:00:04	100
4	2069	106	106	10/21/2020	2:23:42 PM	0:00:04	99
4	2070	107	107	10/21/2020	2:23:46 PM	0:00:04	94
4	2071	108	108	10/21/2020	2:23:50 PM	0:00:04	100
4	2072	109	109	10/21/2020	2:23:54 PM	0:00:04	100
4	2073	110	110	10/21/2020	2:23:58 PM	0:00:04	100
10	4038	111	111	10/21/2020	2:24:02 PM	0:00:04	99
10	4039	112	112	10/21/2020	2:24:06 PM	0:00:04	100
10	4040	113	113	10/21/2020	2:24:10 PM	0:00:04	99
4	2074	114	114	10/21/2020	2:24:14 PM	0:00:04	99
10	4041	115	115	10/21/2020	2:24:18 PM	0:00:04	100
4	2075	116	116	10/21/2020	2:24:22 PM	0:00:04	99
10	4042	117	117	10/21/2020	2:24:26 PM	0:00:04	100
10	4043	118	118	10/21/2020	2:24:31 PM	0:00:05	100
10	4044	119	119	10/21/2020	2:24:35 PM	0:00:04	100
4	2076	120	120	10/21/2020	2:24:39 PM	0:00:04	100
4	2077	121	121	10/21/2020	2:24:43 PM	0:00:04	100
10	4045	122	122	10/21/2020	2:24:48 PM	0:00:04	100
4	2078	123	123	10/21/2020	2:24:52 PM	0:00:04	99
10	4046	124	124	10/21/2020	2:24:56 PM	0:00:05	100
4	2079	125	125	10/21/2020	2:25:00 PM	0:00:04	99
10	4047	126	126	10/21/2020	2:25:05 PM	0:00:04	99
4	2080	127	127	10/21/2020	2:25:09 PM	0:00:04	99
10	4048	128	128	10/21/2020	2:25:13 PM	0:00:04	100
4	2081	129	129	10/21/2020	2:25:17 PM	0:00:04	100
10	4049	130	130	10/21/2020	2:25:22 PM	0:00:05	100
4	2082	131	131	10/21/2020	2:25:26 PM	0:00:04	100
10	4050	132	132	10/21/2020	2:25:30 PM	0:00:04	100
4	2083	133	133	10/21/2020	2:25:35 PM	0:00:04	100
10	4051	134	134	10/21/2020	2:25:39 PM	0:00:04	100
4	2084	135	135	10/21/2020	2:25:44 PM	0:00:05	100
10	4052	136	136	10/21/2020	2:25:49 PM	0:00:05	100
4	2085	137	137	10/21/2020	2:25:53 PM	0:00:04	100
10	4053	138	138	10/21/2020	2:25:57 PM	0:00:04	100
4	2086	139	139	10/21/2020	2:26:01 PM	0:00:04	100
10	4054	140	140	10/21/2020	2:26:05 PM	0:00:04	100
10	4055	141	141	10/21/2020	2:26:09 PM	0:00:04	100
4	2087	142	142	10/21/2020	2:26:13 PM	0:00:04	98

4	2088	143	143	10/21/2020	2:26:17 PM	0:00:04	100
10	4056	144	144	10/21/2020	2:26:21 PM	0:00:04	97
4	2089	145	145	10/21/2020	2:26:25 PM	0:00:04	100
4	2090	146	146	10/21/2020	2:26:29 PM	0:00:04	99
10	4057	147	147	10/21/2020	2:26:33 PM	0:00:04	96
4	2091	148	148	10/21/2020	2:26:37 PM	0:00:04	100
4	2092	149	149	10/21/2020	2:26:41 PM	0:00:04	100
10	4058	150	150	10/21/2020	2:26:45 PM	0:00:04	97
4	2093	151	151	10/21/2020	2:26:49 PM	0:00:04	100
4	2094	152	152	10/21/2020	2:26:53 PM	0:00:04	100
4	2095	153	153	10/21/2020	2:26:58 PM	0:00:04	97
4	2096	154	154	10/21/2020	2:27:02 PM	0:00:04	99
7	3001	155	155	10/21/2020	2:27:06 PM	0:00:04	100
7	3002	156	156	10/21/2020	2:27:10 PM	0:00:04	100
4	2097	157	157	10/21/2020	2:27:14 PM	0:00:04	100
7	3003	158	158	10/21/2020	2:27:18 PM	0:00:04	100
7	3004	159	159	10/21/2020	2:27:22 PM	0:00:04	98
4	2098	160	160	10/21/2020	2:27:26 PM	0:00:04	100
7	3005	161	161	10/21/2020	2:27:30 PM	0:00:04	98
4	2099	162	162	10/21/2020	2:27:35 PM	0:00:04	99
7	3006	163	163	10/21/2020	2:27:39 PM	0:00:04	100
4	2100	164	164	10/21/2020	2:27:43 PM	0:00:04	100
7	3007	165	165	10/21/2020	2:27:47 PM	0:00:04	100
4	2101	166	166	10/21/2020	2:27:53 PM	0:00:06	100
7	3008	167	167	10/21/2020	2:27:58 PM	0:00:05	100
4	2102	168	168	10/21/2020	2:28:02 PM	0:00:04	97
7	3009	169	169	10/21/2020	2:28:06 PM	0:00:04	95
4	2103	170	170	10/21/2020	2:28:10 PM	0:00:04	99
7	3010	171	171	10/21/2020	2:28:14 PM	0:00:04	99
4	2104	172	172	10/21/2020	2:28:18 PM	0:00:04	99
7	3011	173	173	10/21/2020	2:28:22 PM	0:00:04	98
4	2105	174	174	10/21/2020	2:28:26 PM	0:00:04	100
7	3012	175	175	10/21/2020	2:28:30 PM	0:00:04	100
7	3013	176	176	10/21/2020	2:28:34 PM	0:00:04	98
4	2106	177	177	10/21/2020	2:28:38 PM	0:00:04	95
7	3014	178	178	10/21/2020	2:28:42 PM	0:00:04	100
4	2107	179	179	10/21/2020	2:28:47 PM	0:00:04	95
7	3015	180	180	10/21/2020	2:28:50 PM	0:00:04	94
4	2108	181	181	10/21/2020	2:28:54 PM	0:00:04	100
7	3016	182	182	10/21/2020	2:28:58 PM	0:00:04	100
7	3017	183	183	10/21/2020	2:29:02 PM	0:00:04	100
4	2109	184	184	10/21/2020	2:29:06 PM	0:00:04	94
7	3018	185	185	10/21/2020	2:29:10 PM	0:00:04	97
7	3019	186	186	10/21/2020	2:29:15 PM	0:00:05	95
2	1001	187	187	10/21/2020	2:29:19 PM	0:00:04	75

7	3020	188	188	10/21/2020	2:29:22 PM	0:00:03	99
7	3021	189	189	10/21/2020	2:29:26 PM	0:00:04	99
7	3022	190	190	10/21/2020	2:29:31 PM	0:00:04	99
7	3023	191	191	10/21/2020	2:29:35 PM	0:00:04	100
7	3024	192	192	10/21/2020	2:29:40 PM	0:00:05	99
4	2110	193	193	10/21/2020	2:29:44 PM	0:00:04	98
7	3025	194	194	10/21/2020	2:29:48 PM	0:00:05	100
4	2111	195	195	10/21/2020	2:29:53 PM	0:00:04	100
7	3026	196	196	10/21/2020	2:29:57 PM	0:00:04	100
7	3027	197	197	10/21/2020	2:30:01 PM	0:00:04	100
7	3028	198	198	10/21/2020	2:30:05 PM	0:00:04	100
2	1002	199	199	10/21/2020	2:30:09 PM	0:00:04	21
2	1003	200	200	10/21/2020	2:30:10 PM	0:00:02	75
2	1004	201	201	10/21/2020	2:30:14 PM	0:00:04	80
4	2112	202	202	10/21/2020	2:30:17 PM	0:00:03	99
4	2113	203	203	10/21/2020	2:30:22 PM	0:00:04	95
4	2114	204	204	10/21/2020	2:30:25 PM	0:00:04	98
4	2115	205	205	10/21/2020	2:30:29 PM	0:00:04	96
4	2116	206	206	10/21/2020	2:30:34 PM	0:00:04	99
4	2117	207	207	10/21/2020	2:30:38 PM	0:00:04	100
4	2118	208	208	10/21/2020	2:30:42 PM	0:00:04	100
4	2119	209	209	10/21/2020	2:30:46 PM	0:00:04	100
4	2120	210	210	10/21/2020	2:30:50 PM	0:00:04	100
4	2121	211	211	10/21/2020	2:30:54 PM	0:00:04	100
4	2122	212	212	10/21/2020	2:30:59 PM	0:00:04	100
4	2123	213	213	10/21/2020	2:31:03 PM	0:00:04	100
2	1005	214	214	10/21/2020	2:31:06 PM	0:00:04	68
2	1006	215	215	10/21/2020	2:31:09 PM	0:00:03	37
2	1007	216	216	10/21/2020	2:31:11 PM	0:00:02	76
2	1008	217	217	10/21/2020	2:31:14 PM	0:00:03	14
10	4059	218	218	10/21/2020	2:31:15 PM	0:00:01	100
10	4060	219	219	10/21/2020	2:31:20 PM	0:00:04	100
10	4061	220	220	10/21/2020	2:31:24 PM	0:00:04	100
10	4062	221	221	10/21/2020	2:31:28 PM	0:00:04	98
10	4063	222	222	10/21/2020	2:31:32 PM	0:00:04	100
10	4064	223	223	10/21/2020	2:31:36 PM	0:00:04	100
10	4065	224	224	10/21/2020	2:31:40 PM	0:00:04	100
10	4066	225	225	10/21/2020	2:31:44 PM	0:00:04	100
10	4067	226	226	10/21/2020	2:31:49 PM	0:00:04	99
10	4068	227	227	10/21/2020	2:31:53 PM	0:00:04	100
10	4069	228	228	10/21/2020	2:31:57 PM	0:00:04	100
10	4070	229	229	10/21/2020	2:32:02 PM	0:00:04	100
10	4071	230	230	10/21/2020	2:32:06 PM	0:00:04	99
10	4072	231	231	10/21/2020	2:32:10 PM	0:00:04	100
10	4073	232	232	10/21/2020	2:32:15 PM	0:00:05	100

10	4074	233	233	10/21/2020	2:32:19 PM	0:00:04	100
10	4075	234	234	10/21/2020	2:32:23 PM	0:00:04	97
10	4076	235	235	10/21/2020	2:32:27 PM	0:00:04	100
10	4077	236	236	10/21/2020	2:32:32 PM	0:00:05	100
10	4078	237	237	10/21/2020	2:32:36 PM	0:00:04	100
10	4079	238	238	10/21/2020	2:32:40 PM	0:00:04	100
10	4080	239	239	10/21/2020	2:32:44 PM	0:00:04	100
10	4081	240	240	10/21/2020	2:32:49 PM	0:00:04	101
10	4082	241	241	10/21/2020	2:32:53 PM	0:00:04	100
10	4083	242	242	10/21/2020	2:32:57 PM	0:00:04	100
10	4084	243	243	10/21/2020	2:33:02 PM	0:00:05	100
10	4085	244	244	10/21/2020	2:33:06 PM	0:00:04	100
10	4086	245	245	10/21/2020	2:33:11 PM	0:00:05	98
10	4087	246	246	10/21/2020	2:33:15 PM	0:00:04	98
10	4088	247	247	10/21/2020	2:33:19 PM	0:00:04	95
10	4089	248	248	10/21/2020	2:33:23 PM	0:00:04	97
10	4090	249	249	10/21/2020	2:33:27 PM	0:00:05	98
10	4091	250	250	10/21/2020	2:33:31 PM	0:00:04	95
10	4092	251	251	10/21/2020	2:33:35 PM	0:00:04	98
10	4093	252	252	10/21/2020	2:33:39 PM	0:00:05	98
10	4094	253	253	10/21/2020	2:33:43 PM	0:00:04	95
10	4095	254	254	10/21/2020	2:33:47 PM	0:00:04	97
10	4096	255	255	10/21/2020	2:33:51 PM	0:00:04	96
10	4097	256	256	10/21/2020	2:33:55 PM	0:00:04	95
10	4098	257	257	10/21/2020	2:33:59 PM	0:00:04	100
10	4099	258	258	10/21/2020	2:34:03 PM	0:00:04	63
10	4100	259	259	10/21/2020	2:34:06 PM	0:00:03	100
7	3029	260	260	10/21/2020	2:34:10 PM	0:00:04	100
7	3030	261	261	10/21/2020	2:34:14 PM	0:00:04	100
7	3031	262	262	10/21/2020	2:34:18 PM	0:00:04	100
7	3032	263	263	10/21/2020	2:34:22 PM	0:00:04	100
7	3033	264	264	10/21/2020	2:34:26 PM	0:00:04	100
7	3034	265	265	10/21/2020	2:34:30 PM	0:00:04	100
7	3035	266	266	10/21/2020	2:34:34 PM	0:00:04	100
7	3036	267	267	10/21/2020	2:34:38 PM	0:00:04	99
7	3037	268	268	10/21/2020	2:34:42 PM	0:00:04	100
7	3038	269	269	10/21/2020	2:39:47 PM	0:05:05	100
7	3039	270	270	10/21/2020	2:45:38 PM	0:05:50	100
7	3040	271	271	10/21/2020	2:48:42 PM	0:03:05	99
7	3041	272	272	10/21/2020	2:51:47 PM	0:03:05	100
7	3042	273	273	10/21/2020	2:55:39 PM	0:03:52	100
7	3043	274	274	10/21/2020	3:03:44 PM	0:08:06	99
7	3044	275	275	10/21/2020	3:08:37 PM	0:04:52	100
7	3045	276	276	10/21/2020	3:15:42 PM	0:07:05	100
7	3046	277	277	10/21/2020	3:25:48 PM	0:10:06	99

7	3047	278	278	10/21/2020	3:30:39 PM	0:04:51	100
7	3048	279	279	10/21/2020	3:33:43 PM	0:03:05	100
7	3049	280	280	10/21/2020	3:39:50 PM	0:06:06	99
7	3050	281	281	10/21/2020	3:42:39 PM	0:02:49	98
7	3051	282	282	10/21/2020	3:45:43 PM	0:03:04	99
7	3052	283	283	10/21/2020	3:49:48 PM	0:04:05	100
7	3053	284	284	10/21/2020	3:52:38 PM	0:02:49	99
7	3054	285	285	10/21/2020	3:56:43 PM	0:04:05	100
7	3055	286	286	10/21/2020	3:58:47 PM	0:02:04	100
4	2124	287	287	10/21/2020	4:06:38 PM	0:07:52	100
4	2125	288	288	10/21/2020	4:08:43 PM	0:02:04	100
4	2126	289	289	10/21/2020	4:11:48 PM	0:03:05	100
4	2127	290	290	10/21/2020	4:19:38 PM	0:07:51	98
4	2128	291	291	10/21/2020	4:23:43 PM	0:04:05	100
4	2129	292	292	10/21/2020	4:27:48 PM	0:04:05	100
4	2130	293	293	10/21/2020	4:30:37 PM	0:02:50	100
4	2131	294	294	10/21/2020	4:35:42 PM	0:05:05	99
4	2132	295	295	10/21/2020	4:39:47 PM	0:04:05	100
4	2133	296	296	10/21/2020	4:44:37 PM	0:04:50	99
4	2134	297	297	10/21/2020	4:47:42 PM	0:03:05	99
4	2135	298	298	10/21/2020	4:51:47 PM	0:04:05	100
4	2136	299	299	10/21/2020	4:55:38 PM	0:03:51	100
4	2137	300	300	10/21/2020	5:00:42 PM	0:05:05	100
4	2138	301	301	10/21/2020	5:03:47 PM	0:03:04	100
4	2139	302	302	10/21/2020	5:06:36 PM	0:02:50	100
4	2140	303	303	10/21/2020	5:09:41 PM	0:03:04	98
4	2141	304	304	10/21/2020	5:12:45 PM	0:03:05	100
4	2142	305	305	10/21/2020	5:17:36 PM	0:04:50	100
4	2143	306	306	10/21/2020	5:24:41 PM	0:07:05	99
10	4101	307	307	10/22/2020	8:33:49 AM		100
10	4102	308	308	10/22/2020	8:41:55 AM	0:08:06	96
10	4103	309	309	10/22/2020	8:49:45 AM	0:07:50	98
4	2144	310	310	10/22/2020	8:50:50 AM	0:01:05	100
4	2145	311	311	10/22/2020	8:53:41 AM	0:02:51	100
10	4104	312	312	10/22/2020	8:56:45 AM	0:03:05	100
4	2146	313	313	10/22/2020	9:01:50 AM	0:05:05	99
4	2147	314	314	10/22/2020	9:08:43 AM	0:06:53	96
4	2148	315	315	10/22/2020	9:13:48 AM	0:05:05	100
10	4105	316	316	10/22/2020	9:14:52 AM	0:01:04	100
10	4106	317	317	10/22/2020	9:20:43 AM	0:05:51	100
4	2149	318	318	10/22/2020	9:21:47 AM	0:01:04	96
10	4107	319	319	10/22/2020	9:24:52 AM	0:03:04	100
4	2150	320	320	10/22/2020	9:29:41 AM	0:04:50	99
10	4108	321	321	10/22/2020	9:30:46 AM	0:01:04	100
4	2151	322	322	10/22/2020	9:32:50 AM	0:02:04	100

10	4109	323	323	10/22/2020	9:35:54 AM	0:03:05	100
4	2152	324	324	10/22/2020	9:39:44 AM	0:03:50	100
10	4110	325	325	10/22/2020	9:40:48 AM	0:01:04	100
4	2153	326	326	10/22/2020	9:43:53 AM	0:03:05	100
10	4111	327	327	10/22/2020	9:44:42 AM	0:00:50	100
4	2154	328	328	10/22/2020	9:46:47 AM	0:02:04	100
10	4112	329	329	10/22/2020	9:48:51 AM	0:02:04	100
4	2155	330	330	10/22/2020	9:52:41 AM	0:03:50	100
10	4113	331	331	10/22/2020	9:52:59 AM	0:00:18	100
4	2156	332	332	10/22/2020	9:55:48 AM	0:02:49	99
10	4114	333	333	10/22/2020	9:56:52 AM	0:01:04	100
4	2157	334	334	10/22/2020	9:58:41 AM	0:01:49	100
10	4115	335	335	10/22/2020	10:00:46 AM	0:02:04	100
4	2158	336	336	10/22/2020	10:01:50 AM	0:01:05	100
4	2159	337	337	10/22/2020	10:04:42 AM	0:02:52	100
4	2160	338	338	10/22/2020	10:09:47 AM	0:05:05	100
4	2161	339	339	10/22/2020	10:12:52 AM	0:03:05	100
4	2162	340	340	10/22/2020	10:15:42 AM	0:02:51	100
10	4116	341	341	10/22/2020	10:18:47 AM	0:03:05	99
4	2163	342	342	10/22/2020	10:20:51 AM	0:02:04	100
10	4117	343	343	10/22/2020	10:20:55 AM	0:00:04	100
10	4118	344	344	10/22/2020	10:24:45 AM	0:03:50	100
10	4119	345	345	10/22/2020	10:28:50 AM	0:04:05	100
4	2164	346	346	10/22/2020	10:31:54 AM	0:03:04	99
10	4120	347	347	10/22/2020	10:32:42 AM	0:00:48	100
10	4121	348	348	10/22/2020	10:36:46 AM	0:04:04	100
4	2165	349	349	10/22/2020	10:37:50 AM	0:01:04	92
10	4122	350	350	10/22/2020	10:41:42 AM	0:03:51	100
4	2166	351	351	10/22/2020	10:44:46 AM	0:03:05	95
10	4123	352	352	10/22/2020	10:46:51 AM	0:02:04	100
4	2167	353	353	10/22/2020	10:52:40 AM	0:05:50	46
4	2168	354	354	10/22/2020	10:57:44 AM	0:05:04	65
4	2169	355	355	10/22/2020	11:08:49 AM	0:11:05	85
4	2170	356	356	10/22/2020	11:09:53 AM	0:01:04	85
4	2171	357	357	10/22/2020	11:11:42 AM	0:01:49	78
2	1009	358	358	10/22/2020	11:21:47 AM	0:10:05	26
2	1010	359	359	10/22/2020	11:22:48 AM	0:01:02	66
2	1011	360	360	10/22/2020	11:26:52 AM	0:04:04	100
2	1012	361	361	10/22/2020	11:28:41 AM	0:01:49	31
4	2172	362	362	10/22/2020	12:22:41 PM	0:54:01	60
4	2173	363	363	10/22/2020	3:01:48 PM	2:39:07	100
4	2174	364	364	10/22/2020	3:07:53 PM	0:06:05	99
4	2175	365	365	10/22/2020	3:12:43 PM	0:04:50	100
4	2176	366	366	10/22/2020	3:16:48 PM	0:04:05	100
4	2177	367	367	10/22/2020	3:18:53 PM	0:02:04	100

4	2178	368	368	10/22/2020	3:21:44 PM	0:02:51	100
4	2179	369	369	10/22/2020	3:26:49 PM	0:05:05	100
4	2180	370	370	10/22/2020	3:31:54 PM	0:05:05	100
4	2181	371	371	10/22/2020	3:37:44 PM	0:05:51	99
4	2182	372	372	10/22/2020	3:40:49 PM	0:03:05	100
4	2183	373	373	10/22/2020	3:43:54 PM	0:03:05	100
4	2184	374	374	10/22/2020	3:46:44 PM	0:02:50	100
10	4124	375	375	10/22/2020	4:19:55 PM	0:33:11	99
10	4125	376	376	10/22/2020	4:35:48 PM	0:15:53	85
10	4126	377	377	10/22/2020	4:37:52 PM	0:02:04	87
10	4127	378	378	10/22/2020	4:39:56 PM	0:02:04	100
10	4128	379	379	10/22/2020	4:42:45 PM	0:02:50	100
10	4129	380	380	10/22/2020	4:45:50 PM	0:03:04	100
10	4130	381	381	10/22/2020	4:48:54 PM	0:03:05	100
10	4131	382	382	10/22/2020	4:54:45 PM	0:05:51	100
10	4132	383	383	10/22/2020	4:57:49 PM	0:03:05	100
4	2185	384	384	10/22/2020	5:03:54 PM	0:06:05	98
4	2186	385	385	10/22/2020	5:06:44 PM	0:02:50	100
4	2187	386	386	10/22/2020	5:13:50 PM	0:07:05	100
4	2188	387	387	10/22/2020	5:18:55 PM	0:05:05	100
4	2189	388	388	10/22/2020	5:21:45 PM	0:02:50	100
4	2190	389	389	10/22/2020	5:24:50 PM	0:03:05	100
4	2191	390	390	10/22/2020	5:31:55 PM	0:07:05	99
4	2192	391	391	10/22/2020	5:36:45 PM	0:04:50	88
4	2193	392	392	10/22/2020	5:40:49 PM	0:04:04	100
4	2194	393	393	10/22/2020	5:43:54 PM	0:03:05	100
4	2195	394	394	10/22/2020	5:46:44 PM	0:02:50	100
4	2196	395	395	10/22/2020	5:49:48 PM	0:03:05	100
4	2197	396	396	10/22/2020	6:01:54 PM	0:12:06	24
4	2198	397	397	10/26/2020	10:25:50 AM		100
4	2199	398	398	10/26/2020	10:27:54 AM	0:02:04	100
4	2200	399	399	10/26/2020	10:30:45 AM	0:02:51	100
10	4133	400	400	10/26/2020	10:32:49 AM	0:02:04	99
4	2201	401	401	10/26/2020	10:33:54 AM	0:01:05	100
10	4134	402	402	10/26/2020	10:36:44 AM	0:02:49	99
4	2202	403	403	10/26/2020	10:37:48 AM	0:01:04	99
10	4135	404	404	10/26/2020	10:40:52 AM	0:03:05	100
4	2203	405	405	10/26/2020	10:43:42 AM	0:02:50	100
10	4136	406	406	10/26/2020	10:44:46 AM	0:01:04	100
4	2204	407	407	10/26/2020	10:46:51 AM	0:02:04	100
10	4137	408	408	10/26/2020	10:48:41 AM	0:01:50	100
4	2205	409	409	10/26/2020	10:49:45 AM	0:01:04	100
10	4138	410	410	10/26/2020	10:51:50 AM	0:02:04	100
2	1013	411	411	10/26/2020	10:53:40 AM	0:01:50	66
10	4139	412	412	10/26/2020	10:55:44 AM	0:02:04	100

2	1014	413	413	10/26/2020	10:58:48 AM	0:03:04	59
10	4140	414	414	10/26/2020	10:58:51 AM	0:00:02	55
2	1015	415	415	10/26/2020	11:02:40 AM	0:03:49	57
2	1016	416	416	10/26/2020	11:04:44 AM	0:02:03	87
2	1017	417	417	10/26/2020	11:06:48 AM	0:02:04	95
2	1018	418	418	10/26/2020	11:09:52 AM	0:03:04	97
2	1019	419	419	10/26/2020	11:12:41 AM	0:02:49	43
10	4141	420	420	10/26/2020	1:13:55 PM	2:01:14	99
10	4142	421	421	10/26/2020	1:17:45 PM	0:03:50	99
10	4143	422	422	10/26/2020	1:24:50 PM	0:07:05	100
10	4144	423	423	10/26/2020	1:28:55 PM	0:04:05	100
10	4145	424	424	10/26/2020	1:31:44 PM	0:02:50	100
10	4146	425	425	10/26/2020	1:34:49 PM	0:03:05	100
10	4147	426	426	10/26/2020	1:41:54 PM	0:07:05	100
4	2206	427	427	10/26/2020	1:42:45 PM	0:00:51	98
10	4148	428	428	10/26/2020	1:45:50 PM	0:03:04	99
10	4149	429	429	10/26/2020	1:49:54 PM	0:04:05	100
10	4150	430	430	10/26/2020	1:53:44 PM	0:03:50	99
4	2207	431	431	10/26/2020	1:56:49 PM	0:03:05	99
4	2208	432	432	10/26/2020	1:59:53 PM	0:03:05	100
4	2209	433	433	10/26/2020	2:02:43 PM	0:02:50	99
4	2210	434	434	10/26/2020	2:05:49 PM	0:03:05	100
4	2211	435	435	10/26/2020	2:08:53 PM	0:03:05	100
10	4151	436	436	10/26/2020	2:27:47 PM	0:18:54	100
10	4152	437	437	10/26/2020	2:30:51 PM	0:03:05	99
10	4153	438	438	10/26/2020	2:34:44 PM	0:03:53	100
10	4154	439	439	10/26/2020	2:38:49 PM	0:04:05	99
10	4155	440	440	10/26/2020	2:41:53 PM	0:03:05	100
10	4156	441	441	10/26/2020	2:50:44 PM	0:08:51	96
10	4157	442	442	10/26/2020	2:53:49 PM	0:03:04	100
10	4158	443	443	10/26/2020	2:56:55 PM	0:03:07	100
10	4159	444	444	10/26/2020	3:00:46 PM	0:03:50	100
10	4160	445	445	10/26/2020	3:03:50 PM	0:03:05	100
7	3056	446	446	10/26/2020	3:12:56 PM	0:09:06	100
4	2212	447	447	10/26/2020	3:13:45 PM	0:00:49	99
7	3057	448	448	10/26/2020	3:15:50 PM	0:02:04	100
4	2213	449	449	10/26/2020	3:16:54 PM	0:01:04	100
4	2214	450	450	10/26/2020	3:19:43 PM	0:02:49	37
4	2215	451	451	10/26/2020	3:26:47 PM	0:07:04	99
4	2216	452	452	10/26/2020	4:40:52 PM	1:14:05	100
4	2217	453	453	10/26/2020	4:44:57 PM	0:04:05	100
4	2218	454	454	10/26/2020	4:48:46 PM	0:03:50	99
4	2219	455	455	10/26/2020	4:52:51 PM	0:04:05	100
4	2220	456	456	10/26/2020	4:54:56 PM	0:02:04	100
4	2221	457	457	10/26/2020	4:59:45 PM	0:04:50	100

4	2222	458	458	10/26/2020	5:03:50 PM	0:04:05	100
4	2223	459	459	10/26/2020	5:06:55 PM	0:03:05	100
4	2224	460	460	10/26/2020	5:09:45 PM	0:02:50	100
4	2225	461	461	10/26/2020	5:12:49 PM	0:03:05	100
4	2226	462	462	10/26/2020	5:15:54 PM	0:03:04	98
4	2227	463	463	10/26/2020	5:21:44 PM	0:05:50	100
4	2228	464	464	10/26/2020	5:30:50 PM	0:09:06	100
4	2229	465	465	10/27/2020	12:58:54 PM		100
4	2230	466	466	10/27/2020	1:04:00 PM	0:05:05	98
10	4161	467	467	10/27/2020	1:06:04 PM	0:02:04	100
4	2231	468	468	10/27/2020	1:07:53 PM	0:01:49	100
10	4162	469	469	10/27/2020	1:08:58 PM	0:01:05	100
4	2232	470	470	10/27/2020	1:12:02 PM	0:03:05	100
10	4163	471	471	10/27/2020	1:12:52 PM	0:00:49	99
4	2233	472	472	10/27/2020	1:14:56 PM	0:02:04	100
10	4164	473	473	10/27/2020	1:16:00 PM	0:01:04	82
4	2234	474	474	10/27/2020	1:19:04 PM	0:03:04	100
10	4165	475	475	10/27/2020	1:19:53 PM	0:00:49	99
4	2235	476	476	10/27/2020	1:21:57 PM	0:02:04	100
10	4166	477	477	10/27/2020	1:23:02 PM	0:01:04	100
4	2236	478	478	10/27/2020	1:24:51 PM	0:01:49	99
2	1020	479	479	10/27/2020	1:30:56 PM	0:06:06	84
2	1021	480	480	10/27/2020	1:35:00 PM	0:04:04	34
2	1022	481	481	10/27/2020	1:38:03 PM	0:03:02	74
2	1023	482	482	10/27/2020	1:38:51 PM	0:00:49	71
4	2237	483	483	10/27/2020	1:49:57 PM	0:11:05	100
4	2238	484	484	10/27/2020	1:53:01 PM	0:03:05	100
4	2239	485	485	10/27/2020	1:55:52 PM	0:02:50	100
4	2240	486	486	10/27/2020	2:01:57 PM	0:06:06	98
4	2241	487	487	10/27/2020	2:06:02 PM	0:04:05	100
4	2242	488	488	10/27/2020	2:07:51 PM	0:01:49	100
4	2243	489	489	10/27/2020	2:14:57 PM	0:07:05	100
4	2244	490	490	10/27/2020	2:18:01 PM	0:03:05	100
4	2245	491	491	10/27/2020	2:20:51 PM	0:02:50	100
4	2246	492	492	10/27/2020	2:24:56 PM	0:04:05	100
4	2247	493	493	10/27/2020	2:29:01 PM	0:04:05	96
10	4167	494	494	10/27/2020	2:29:50 PM	0:00:49	68
4	2248	495	495	10/27/2020	2:30:53 PM	0:01:03	99
10	4168	496	496	10/27/2020	2:31:57 PM	0:01:04	100
4	2249	497	497	10/27/2020	2:35:02 PM	0:03:05	100
10	4169	498	498	10/27/2020	2:35:06 PM	0:00:04	100
4	2250	499	499	10/27/2020	2:38:56 PM	0:03:50	100
10	4170	500	500	10/27/2020	2:41:00 PM	0:02:04	99
4	2251	501	501	10/27/2020	2:42:04 PM	0:01:04	100
10	4171	502	502	10/27/2020	2:44:53 PM	0:02:50	100

4	2252	503	503	10/27/2020	2:45:58 PM	0:01:04	100
10	4172	504	504	10/27/2020	2:48:02 PM	0:02:04	100
4	2253	505	505	10/27/2020	2:49:53 PM	0:01:51	100
4	2254	506	506	10/27/2020	2:54:58 PM	0:05:05	98
4	2255	507	507	10/27/2020	2:59:02 PM	0:04:05	100
4	2256	508	508	10/27/2020	3:01:52 PM	0:02:50	100
4	2257	509	509	10/27/2020	3:08:57 PM	0:07:05	100
4	2258	510	510	10/27/2020	3:16:03 PM	0:07:05	100
4	2259	511	511	10/27/2020	3:22:54 PM	0:06:51	100
4	2260	512	512	10/27/2020	3:25:58 PM	0:03:05	100
4	2261	513	513	10/27/2020	3:30:03 PM	0:04:05	100
10	4173	514	514	10/27/2020	3:31:52 PM	0:01:49	100
10	4174	515	515	10/27/2020	3:37:58 PM	0:06:05	99
4	2262	516	516	10/27/2020	3:39:02 PM	0:01:04	100
4	2263	517	517	10/27/2020	3:44:53 PM	0:05:51	100
10	4175	518	518	10/27/2020	3:45:11 PM	0:00:19	100
10	4176	519	519	10/27/2020	4:06:05 PM	0:20:53	98
10	4177	520	520	10/27/2020	4:09:54 PM	0:03:50	98
10	4178	521	521	10/27/2020	4:13:59 PM	0:04:05	100
10	4179	522	522	10/27/2020	4:18:04 PM	0:04:05	100
10	4180	523	523	10/27/2020	4:21:54 PM	0:03:50	100
10	4181	524	524	10/27/2020	4:25:59 PM	0:04:05	100
4	2264	525	525	10/27/2020	4:27:02 PM	0:01:04	100
10	4182	526	526	10/27/2020	4:28:52 PM	0:01:49	100
4	2265	527	527	10/27/2020	4:29:56 PM	0:01:04	100
10	4183	528	528	10/27/2020	4:32:00 PM	0:02:04	100
4	2266	529	529	10/27/2020	4:32:52 PM	0:00:52	100
10	4184	530	530	10/27/2020	4:35:57 PM	0:03:05	98
4	2267	531	531	10/27/2020	4:42:02 PM	0:06:05	99
4	2268	532	532	10/27/2020	4:49:53 PM	0:07:51	99
4	2269	533	533	10/27/2020	4:54:58 PM	0:05:05	90
2	1024	534	534	10/27/2020	6:54:58 PM	2:00:00	57
7	3058	535	535	10/27/2020	7:01:02 PM	0:06:04	99
4	2270	536	536	10/28/2020	2:03:04 PM		10
4	2271	537	537	10/29/2020	12:58:08 PM		100
4	2272	538	538	10/29/2020	1:01:13 PM	0:03:05	100
4	2273	539	539	10/29/2020	1:08:05 PM	0:06:52	100
4	2274	540	540	10/29/2020	1:12:10 PM	0:04:05	100
4	2275	541	541	10/29/2020	1:17:00 PM	0:04:50	98
4	2276	542	542	10/29/2020	1:21:05 PM	0:04:05	100
4	2277	543	543	10/29/2020	1:24:09 PM	0:03:04	100
4	2278	544	544	10/29/2020	1:30:01 PM	0:05:52	100
4	2279	545	545	10/29/2020	1:36:06 PM	0:06:05	100
4	2280	546	546	10/29/2020	1:40:11 PM	0:04:05	99
4	2281	547	547	10/29/2020	1:44:02 PM	0:03:51	100

4	2282	548	548	10/29/2020	1:48:06 PM	0:04:05	99
4	2283	549	549	10/29/2020	1:54:11 PM	0:06:05	100
4	2284	550	550	10/29/2020	1:56:02 PM	0:01:51	100
4	2285	551	551	10/29/2020	1:59:07 PM	0:03:05	100
4	2286	552	552	10/29/2020	2:02:12 PM	0:03:05	100
4	2287	553	553	10/29/2020	2:09:03 PM	0:06:51	100
4	2288	554	554	10/29/2020	2:13:08 PM	0:04:05	100
4	2289	555	555	10/29/2020	2:19:12 PM	0:06:05	100
4	2290	556	556	10/29/2020	2:22:02 PM	0:02:50	100
4	2291	557	557	10/29/2020	2:25:07 PM	0:03:05	100
4	2292	558	558	10/29/2020	2:36:13 PM	0:11:06	100
4	2293	559	559	10/29/2020	2:38:04 PM	0:01:51	100
4	2294	560	560	10/29/2020	2:40:09 PM	0:02:04	100
4	2295	561	561	10/29/2020	2:45:13 PM	0:05:05	97
4	2296	562	562	10/29/2020	2:49:03 PM	0:03:50	99
4	2297	563	563	10/29/2020	2:51:07 PM	0:02:04	100
4	2298	564	564	10/29/2020	2:54:14 PM	0:03:06	100
4	2299	565	565	10/29/2020	2:57:03 PM	0:02:50	100
4	2300	566	566	10/29/2020	3:00:08 PM	0:03:05	100
2	1025	567	567	10/29/2020	3:16:15 PM	0:16:07	72
2	1026	568	568	10/29/2020	3:17:03 PM	0:00:48	61
2	1027	569	569	10/29/2020	3:23:07 PM	0:06:04	83
2	1028	570	570	10/29/2020	3:25:11 PM	0:02:04	82
2	1029	571	571	10/29/2020	3:27:14 PM	0:02:04	59
2	1030	572	572	10/29/2020	3:33:03 PM	0:05:49	82
2	1031	573	573	10/29/2020	3:34:07 PM	0:01:03	81
2	1032	574	574	10/29/2020	3:35:10 PM	0:01:03	76
2	1033	575	575	10/29/2020	3:37:13 PM	0:02:04	73
2	1034	576	576	10/29/2020	3:41:02 PM	0:03:49	52
4	2301	577	577	10/29/2020	3:45:05 PM	0:04:03	100
4	2302	578	578	10/29/2020	3:48:10 PM	0:03:04	99
4	2303	579	579	10/29/2020	3:52:14 PM	0:04:05	100
4	2304	580	580	10/29/2020	3:55:04 PM	0:02:49	100
4	2305	581	581	10/29/2020	3:59:09 PM	0:04:05	99
4	2306	582	582	10/29/2020	4:04:13 PM	0:05:05	99
4	2307	583	583	10/29/2020	4:07:03 PM	0:02:50	100
4	2308	584	584	10/29/2020	4:11:08 PM	0:04:05	100
4	2309	585	585	10/29/2020	4:13:12 PM	0:02:04	100
4	2310	586	586	10/29/2020	4:17:03 PM	0:03:50	98
4	2311	587	587	10/29/2020	4:20:07 PM	0:03:04	100
4	2312	588	588	10/29/2020	4:23:12 PM	0:03:05	100
4	2313	589	589	10/29/2020	4:26:04 PM	0:02:52	99
4	2314	590	590	10/29/2020	4:32:09 PM	0:06:05	98
4	2315	591	591	10/29/2020	4:38:14 PM	0:06:05	100
4	2316	592	592	10/29/2020	4:41:04 PM	0:02:50	100

4	2317	593	593	10/29/2020	4:49:10 PM	0:08:06	100
4	2318	594	594	10/29/2020	4:52:14 PM	0:03:05	99
4	2319	595	595	10/29/2020	4:56:04 PM	0:03:50	100
4	2320	596	596	10/29/2020	4:59:09 PM	0:03:05	99
4	2321	597	597	10/29/2020	5:01:13 PM	0:02:04	100
4	2322	598	598	10/29/2020	5:07:05 PM	0:05:52	99
4	2323	599	599	10/29/2020	5:11:09 PM	0:04:05	99
4	2324	600	600	10/29/2020	5:14:14 PM	0:03:05	100
4	2325	601	601	10/29/2020	5:16:03 PM	0:01:49	100
4	2326	602	602	10/29/2020	5:20:08 PM	0:04:05	100
4	2327	603	603	10/29/2020	5:22:13 PM	0:02:04	100
4	2328	604	604	10/29/2020	5:25:04 PM	0:02:51	100
4	2329	605	605	10/29/2020	5:30:09 PM	0:05:05	100
7	3059	606	606	10/30/2020	12:37:26 PM		97
7	3060	607	607	10/30/2020	12:41:16 PM	0:03:50	100
7	3061	608	608	10/30/2020	12:45:20 PM	0:04:05	100
7	3062	609	609	10/30/2020	12:50:11 PM	0:04:51	100
7	3063	610	610	10/30/2020	12:55:16 PM	0:05:05	99
10	4185	611	611	10/30/2020	1:16:25 PM	0:21:08	100
10	4186	612	612	10/30/2020	1:20:14 PM	0:03:50	100
10	4187	613	613	10/30/2020	1:23:19 PM	0:03:05	100
10	4188	614	614	10/30/2020	1:27:24 PM	0:04:05	99
10	4189	615	615	10/30/2020	1:30:13 PM	0:02:49	100
10	4190	616	616	10/30/2020	1:37:19 PM	0:07:05	100
10	4191	617	617	10/30/2020	1:40:23 PM	0:03:05	100
10	4192	618	618	10/30/2020	1:43:13 PM	0:02:50	100
10	4193	619	619	10/30/2020	1:48:18 PM	0:05:05	100
10	4194	620	620	10/30/2020	1:51:23 PM	0:03:05	100
10	4195	621	621	10/30/2020	1:54:12 PM	0:02:50	100
10	4196	622	622	10/30/2020	2:04:18 PM	0:10:06	99
10	4197	623	623	10/30/2020	2:07:23 PM	0:03:05	100
10	4198	624	624	10/30/2020	2:13:13 PM	0:05:50	100
10	4199	625	625	10/30/2020	2:22:19 PM	0:09:06	96
10	4200	626	626	10/30/2020	2:25:23 PM	0:03:04	100
10	4201	627	627	10/30/2020	2:28:13 PM	0:02:50	100
10	4202	628	628	10/30/2020	2:31:17 PM	0:03:05	100
10	4203	629	629	10/30/2020	2:34:22 PM	0:03:05	100
10	4204	630	630	10/30/2020	2:41:13 PM	0:06:51	98
10	4205	631	631	10/30/2020	2:44:17 PM	0:03:05	100
10	4206	632	632	10/30/2020	2:47:22 PM	0:03:05	100
10	4207	633	633	10/30/2020	2:51:11 PM	0:03:50	100
10	4208	634	634	10/30/2020	2:54:16 PM	0:03:05	100
10	4209	635	635	10/30/2020	2:57:21 PM	0:03:05	100
10	4210	636	636	10/30/2020	3:00:12 PM	0:02:51	100
10	4211	637	637	10/30/2020	3:03:17 PM	0:03:04	100

10	4212	638	638	10/30/2020	3:06:21 PM	0:03:05	100
2	1035	639	639	10/30/2020	3:09:11 PM	0:02:50	96
10	4213	640	640	10/30/2020	3:09:30 PM	0:00:18	100
2	1036	641	641	10/30/2020	3:11:19 PM	0:01:49	79
10	4214	642	642	10/30/2020	3:12:22 PM	0:01:04	100
2	1037	643	643	10/30/2020	3:14:12 PM	0:01:49	82
2	1038	644	644	10/30/2020	3:16:15 PM	0:02:03	69
2	1039	645	645	10/30/2020	3:18:18 PM	0:02:03	34
4	2330	646	646	10/30/2020	3:25:22 PM	0:07:03	100
10	4215	647	647	10/30/2020	3:27:12 PM	0:01:50	93
4	2331	648	648	10/30/2020	3:30:16 PM	0:03:04	98
10	4216	649	649	10/30/2020	3:35:23 PM	0:05:07	100
4	2332	650	650	10/30/2020	3:38:13 PM	0:02:49	100
10	4217	651	651	10/30/2020	3:38:31 PM	0:00:18	100
4	2333	652	652	10/30/2020	3:47:22 PM	0:08:51	100
10	4218	653	653	10/30/2020	3:48:11 PM	0:00:49	100
4	2334	654	654	10/30/2020	3:51:16 PM	0:03:04	100
10	4219	655	655	10/30/2020	3:53:20 PM	0:02:05	70
4	2335	656	656	10/30/2020	3:55:24 PM	0:02:03	100
10	4220	657	657	10/30/2020	3:57:13 PM	0:01:49	100
4	2336	658	658	10/30/2020	3:59:17 PM	0:02:04	99
10	4221	659	659	10/30/2020	4:00:21 PM	0:01:04	100
10	4222	660	660	10/30/2020	4:03:11 PM	0:02:50	100
4	2337	661	661	10/30/2020	4:04:15 PM	0:01:04	99
10	4223	662	662	10/30/2020	4:07:20 PM	0:03:05	100
10	4224	663	663	10/30/2020	4:10:25 PM	0:03:05	100
4	2338	664	664	10/30/2020	4:11:14 PM	0:00:49	100
4	2339	665	665	10/30/2020	4:17:19 PM	0:06:05	99
4	2340	666	666	10/30/2020	4:22:24 PM	0:05:05	100
4	2341	667	667	10/30/2020	4:27:14 PM	0:04:50	84
4	2342	668	668	10/30/2020	4:32:18 PM	0:05:04	75
4	2343	669	669	10/30/2020	4:37:22 PM	0:05:04	68
10	4225	670	670	11/1/2020	1:56:35 PM		100
10	4226	671	671	11/1/2020	1:59:40 PM	0:03:05	100
10	4227	672	672	11/1/2020	2:01:29 PM	0:01:49	100
10	4228	673	673	11/1/2020	2:04:34 PM	0:03:05	100
10	4229	674	674	11/1/2020	2:09:39 PM	0:05:05	100
10	4230	675	675	11/1/2020	2:12:43 PM	0:03:05	100
10	4231	676	676	11/1/2020	2:17:33 PM	0:04:50	100
10	4232	677	677	11/1/2020	2:22:38 PM	0:05:05	99
10	4233	678	678	11/1/2020	2:26:43 PM	0:04:05	99
10	4234	679	679	11/1/2020	2:29:32 PM	0:02:49	100
10	4235	680	680	11/1/2020	2:32:37 PM	0:03:05	99
10	4236	681	681	11/1/2020	2:35:41 PM	0:03:05	100
10	4237	682	682	11/1/2020	2:38:31 PM	0:02:49	100

10	4238	683	683	11/1/2020	2:44:36 PM	0:06:05	100
10	4239	684	684	11/1/2020	2:51:42 PM	0:07:05	97
10	4240	685	685	11/1/2020	2:54:31 PM	0:02:49	100
10	4241	686	686	11/1/2020	2:57:36 PM	0:03:05	100
10	4242	687	687	11/1/2020	3:00:40 PM	0:03:05	99
10	4243	688	688	11/1/2020	3:05:31 PM	0:04:51	100
10	4244	689	689	11/1/2020	3:07:35 PM	0:02:04	100
7	3064	690	690	11/1/2020	3:14:42 PM	0:07:07	100
7	3065	691	691	11/1/2020	3:17:32 PM	0:02:50	100
7	3066	692	692	11/1/2020	3:20:36 PM	0:03:05	100
7	3067	693	693	11/1/2020	3:24:41 PM	0:04:05	100
10	4245	694	694	11/1/2020	3:35:32 PM	0:10:51	100
10	4246	695	695	11/1/2020	3:38:37 PM	0:03:05	100
10	4247	696	696	11/1/2020	3:41:41 PM	0:03:05	100
10	4248	697	697	11/1/2020	3:45:31 PM	0:03:50	99
10	4249	698	698	11/1/2020	3:49:36 PM	0:04:05	99
10	4250	699	699	11/1/2020	3:51:40 PM	0:02:04	100
10	4251	700	700	11/1/2020	3:54:30 PM	0:02:50	100
10	4252	701	701	11/1/2020	3:57:35 PM	0:03:05	100
10	4253	702	702	11/1/2020	4:00:39 PM	0:03:05	100
10	4254	703	703	11/1/2020	4:02:44 PM	0:02:04	100
10	4255	704	704	11/1/2020	4:05:33 PM	0:02:49	100
10	4256	705	705	11/1/2020	4:12:39 PM	0:07:05	100
10	4257	706	706	11/1/2020	4:15:43 PM	0:03:05	100
10	4258	707	707	11/1/2020	4:19:33 PM	0:03:50	100
10	4259	708	708	11/1/2020	4:24:38 PM	0:05:05	100
4	2344	709	709	11/1/2020	4:27:42 PM	0:03:04	70
10	4260	710	710	11/1/2020	4:27:45 PM	0:00:03	100
10	4261	711	711	11/1/2020	4:30:35 PM	0:02:50	100
4	2345	712	712	11/1/2020	4:32:39 PM	0:02:04	100
10	4262	713	713	11/1/2020	4:33:43 PM	0:01:04	100
10	4263	714	714	11/1/2020	4:36:33 PM	0:02:50	100
10	4264	715	715	11/1/2020	4:38:37 PM	0:02:04	99
10	4265	716	716	11/1/2020	4:42:42 PM	0:04:05	100
4	2346	717	717	11/1/2020	4:45:31 PM	0:02:49	99
10	4266	718	718	11/1/2020	4:46:36 PM	0:01:04	100
4	2347	719	719	11/1/2020	4:48:39 PM	0:02:04	22
10	4267	720	720	11/1/2020	4:51:42 PM	0:03:02	100
10	4268	721	721	11/1/2020	4:54:31 PM	0:02:50	100
2	1040	722	722	11/2/2020	3:06:33 PM		94
2	1041	723	723	11/2/2020	3:06:36 PM	0:00:03	91
2	1042	724	724	11/2/2020	3:06:39 PM	0:00:03	77
2	1043	725	725	11/2/2020	3:06:58 PM	0:00:19	75
2	1044	726	726	11/2/2020	3:07:01 PM	0:00:03	83
2	1045	727	727	11/2/2020	3:07:04 PM	0:00:03	80

2	1046	728	728	11/2/2020	3:07:07 PM	0:00:03	72
2	1047	729	729	11/2/2020	3:07:10 PM	0:00:03	75
2	1048	730	730	11/2/2020	3:07:13 PM	0:00:03	36
2	1049	731	731	11/2/2020	3:07:14 PM	0:00:02	51
2	1050	732	732	11/2/2020	3:07:17 PM	0:00:02	55
2	1051	733	733	11/2/2020	3:07:19 PM	0:00:02	35
2	1052	734	734	11/2/2020	3:07:21 PM	0:00:02	61
2	1053	735	735	11/2/2020	3:07:23 PM	0:00:03	64
2	1054	736	736	11/2/2020	3:07:26 PM	0:00:03	53
2	1055	737	737	11/2/2020	3:07:29 PM	0:00:03	55
2	1056	738	738	11/2/2020	3:07:47 PM	0:00:18	98
10	4269	739	739	11/2/2020	3:07:51 PM	0:00:04	99
2	1057	740	740	11/2/2020	3:08:26 PM	0:00:35	97
10	4270	741	741	11/2/2020	3:08:45 PM	0:00:19	100
2	1058	742	742	11/2/2020	3:10:34 PM	0:01:49	97
2	1059	743	743	11/2/2020	3:12:23 PM	0:01:49	32
10	4271	744	744	11/2/2020	3:13:26 PM	0:01:03	100
2	1060	745	745	11/2/2020	3:17:30 PM	0:04:05	61
2	1061	746	746	11/2/2020	3:19:33 PM	0:02:03	62
2	1062	747	747	11/2/2020	3:20:36 PM	0:01:03	30
10	4272	748	748	11/2/2020	3:22:38 PM	0:02:02	98
10	4273	749	749	11/2/2020	3:26:28 PM	0:03:50	100
4	2348	750	750	11/2/2020	3:28:32 PM	0:02:04	99
10	4274	751	751	11/2/2020	3:29:36 PM	0:01:04	100
4	2349	752	752	11/2/2020	3:31:26 PM	0:01:50	100
10	4275	753	753	11/2/2020	3:32:31 PM	0:01:04	100
4	2350	754	754	11/2/2020	3:34:35 PM	0:02:04	100
4	2351	755	755	11/2/2020	3:36:26 PM	0:01:51	100
10	4276	756	756	11/2/2020	3:37:31 PM	0:01:04	99
4	2352	757	757	11/2/2020	3:40:35 PM	0:03:05	100
10	4277	758	758	11/2/2020	3:40:39 PM	0:00:04	100
4	2353	759	759	11/2/2020	3:44:29 PM	0:03:50	100
10	4278	760	760	11/2/2020	3:45:33 PM	0:01:04	100
4	2354	761	761	11/2/2020	3:50:25 PM	0:04:51	100
10	4279	762	762	11/2/2020	3:50:45 PM	0:00:20	99
10	4280	763	763	11/2/2020	3:54:34 PM	0:03:50	100
4	2355	764	764	11/2/2020	3:55:39 PM	0:01:04	100
4	2356	765	765	11/2/2020	3:58:28 PM	0:02:49	100
10	4281	766	766	11/2/2020	3:59:32 PM	0:01:04	99
4	2357	767	767	11/2/2020	4:02:37 PM	0:03:05	100
10	4282	768	768	11/2/2020	4:02:41 PM	0:00:04	100
10	4283	769	769	11/2/2020	4:05:30 PM	0:02:50	98
4	2358	770	770	11/2/2020	4:06:34 PM	0:01:04	100
10	4284	771	771	11/2/2020	4:12:27 PM	0:05:52	100
10	4285	772	772	11/2/2020	4:14:31 PM	0:02:04	100

10	4286	773	773	11/2/2020	4:19:37 PM	0:05:06	100
7	3068	774	774	11/2/2020	4:21:26 PM	0:01:49	98
10	4287	775	775	11/2/2020	4:22:30 PM	0:01:04	100
10	4288	776	776	11/2/2020	4:25:35 PM	0:03:05	100
7	3069	777	777	11/2/2020	4:27:25 PM	0:01:50	99
7	3070	778	778	11/2/2020	4:32:30 PM	0:05:05	100
7	3071	779	779	11/2/2020	4:42:36 PM	0:10:06	86
7	3072	780	780	11/2/2020	5:06:30 PM	0:23:54	60
2	1063	781	781	11/3/2020	1:58:38 PM		100
2	1064	782	782	11/3/2020	2:01:42 PM	0:03:05	100
2	1065	783	783	11/3/2020	2:07:33 PM	0:05:50	63
2	1066	784	784	11/3/2020	2:08:35 PM	0:01:03	61
2	1067	785	785	11/3/2020	2:12:39 PM	0:04:03	62
2	1068	786	786	11/3/2020	2:14:42 PM	0:02:03	69
2	1069	787	787	11/3/2020	2:19:45 PM	0:05:04	62
10	4289	788	788	11/3/2020	2:24:49 PM	0:05:04	86
2	1070	789	789	11/3/2020	2:26:38 PM	0:01:49	73
10	4290	790	790	11/3/2020	2:26:41 PM	0:00:03	100
2	1071	791	791	11/3/2020	2:28:45 PM	0:02:04	74
2	1072	792	792	11/3/2020	2:29:34 PM	0:00:49	78
10	4291	793	793	11/3/2020	2:29:52 PM	0:00:18	99
2	1073	794	794	11/3/2020	2:31:42 PM	0:01:49	79
10	4292	795	795	11/3/2020	2:32:45 PM	0:01:03	100
2	1074	796	796	11/3/2020	2:34:34 PM	0:01:49	40
10	4293	797	797	11/3/2020	2:35:36 PM	0:01:02	100
10	4294	798	798	11/3/2020	2:37:41 PM	0:02:04	100
2	1075	799	799	11/3/2020	2:40:45 PM	0:03:04	45
10	4295	800	800	11/3/2020	2:42:48 PM	0:02:03	97
10	4296	801	801	11/3/2020	2:44:37 PM	0:01:49	100
2	1076	802	802	11/3/2020	2:46:41 PM	0:02:04	86
2	1077	803	803	11/3/2020	2:48:45 PM	0:02:04	76
2	1078	804	804	11/3/2020	2:54:34 PM	0:05:49	61
10	4297	805	805	11/3/2020	2:57:38 PM	0:03:03	99
2	1079	806	806	11/3/2020	2:58:42 PM	0:01:04	72
2	1080	807	807	11/3/2020	3:00:45 PM	0:02:03	81
10	4298	808	808	11/3/2020	3:01:36 PM	0:00:51	100
2	1081	809	809	11/3/2020	3:02:40 PM	0:01:04	59
10	4299	810	810	11/3/2020	3:04:43 PM	0:02:03	100
10	4300	811	811	11/3/2020	3:07:48 PM	0:03:05	100
10	4301	812	812	11/3/2020	3:10:37 PM	0:02:49	100
7	3073	813	813	11/3/2020	3:11:41 PM	0:01:04	100
10	4302	814	814	11/3/2020	3:13:46 PM	0:02:04	99
7	3074	815	815	11/3/2020	3:14:35 PM	0:00:49	99
10	4303	816	816	11/3/2020	3:16:39 PM	0:02:04	100
7	3075	817	817	11/3/2020	3:17:43 PM	0:01:04	100

10	4304	818	818	11/3/2020	3:18:48 PM	0:01:04	100
7	3076	819	819	11/3/2020	3:20:37 PM	0:01:49	100
10	4305	820	820	11/3/2020	3:21:42 PM	0:01:05	100
7	3077	821	821	11/3/2020	3:23:47 PM	0:02:05	100
10	4306	822	822	11/3/2020	3:24:36 PM	0:00:49	100
10	4307	823	823	11/3/2020	3:26:40 PM	0:02:04	100
7	3078	824	824	11/3/2020	3:27:45 PM	0:01:04	99
10	4308	825	825	11/3/2020	3:30:36 PM	0:02:51	99
7	3079	826	826	11/3/2020	3:32:40 PM	0:02:04	100
10	4309	827	827	11/3/2020	3:33:44 PM	0:01:04	100
7	3080	828	828	11/3/2020	3:35:48 PM	0:02:04	100
7	3081	829	829	11/3/2020	3:41:38 PM	0:05:50	98
10	4310	830	830	11/3/2020	3:41:42 PM	0:00:04	100
7	3082	831	831	11/3/2020	3:45:47 PM	0:04:05	100
10	4311	832	832	11/3/2020	3:47:37 PM	0:01:50	98
7	3083	833	833	11/3/2020	3:49:41 PM	0:02:04	100
10	4312	834	834	11/3/2020	3:50:45 PM	0:01:04	100
10	4313	835	835	11/3/2020	3:54:35 PM	0:03:50	100
7	3084	836	836	11/3/2020	3:55:39 PM	0:01:04	100
7	3085	837	837	11/3/2020	3:58:44 PM	0:03:05	100
10	4314	838	838	11/3/2020	3:58:48 PM	0:00:04	99
7	3086	839	839	11/3/2020	4:03:38 PM	0:04:50	99
10	4315	840	840	11/3/2020	4:03:57 PM	0:00:19	99
10	4316	841	841	11/3/2020	4:07:46 PM	0:03:50	100
7	3087	842	842	11/3/2020	4:08:36 PM	0:00:49	99
10	4317	843	843	11/3/2020	4:10:40 PM	0:02:04	100
10	4318	844	844	11/3/2020	4:12:45 PM	0:02:05	100
7	3088	845	845	11/3/2020	4:16:35 PM	0:03:50	100
10	4319	846	846	11/3/2020	4:16:53 PM	0:00:18	100
10	4320	847	847	11/3/2020	4:19:43 PM	0:02:49	100
7	3089	848	848	11/3/2020	4:21:49 PM	0:02:06	100
10	4321	849	849	11/3/2020	4:22:40 PM	0:00:52	100
7	3090	850	850	11/3/2020	4:25:45 PM	0:03:05	97
7	3091	851	851	11/3/2020	4:28:37 PM	0:02:52	100
7	3092	852	852	11/3/2020	4:34:43 PM	0:06:05	95
10	4322	853	853	11/3/2020	4:37:47 PM	0:03:04	99
7	3093	854	854	11/3/2020	4:38:36 PM	0:00:49	100
7	3094	855	855	11/3/2020	5:11:47 PM	0:33:11	100
7	3095	856	856	11/3/2020	5:14:37 PM	0:02:50	100
10	4323	857	857	11/3/2020	5:44:47 PM	0:30:10	100
10	4324	858	858	11/3/2020	5:47:37 PM	0:02:50	100
10	4325	859	859	11/3/2020	5:55:43 PM	0:08:06	100
10	4326	860	860	11/3/2020	6:01:49 PM	0:06:06	100
7	3096	861	861	11/3/2020	6:11:40 PM	0:09:51	20
10	4327	862	862	11/3/2020	6:11:41 PM	0:00:01	86

7	3097	863	863	11/3/2020	6:48:53 PM	0:37:12	100
7	3098	864	864	11/3/2020	6:51:42 PM	0:02:50	100
7	3099	865	865	11/3/2020	7:19:52 PM	0:28:10	100
10	4328	866	866	11/3/2020	7:20:41 PM	0:00:49	100
7	3100	867	867	11/3/2020	7:24:46 PM	0:04:05	100
10	4329	868	868	11/3/2020	7:25:37 PM	0:00:51	100
7	3101	869	869	11/3/2020	7:27:42 PM	0:02:05	98
10	4330	870	870	11/3/2020	7:34:48 PM	0:07:05	100
7	3102	871	871	11/3/2020	7:35:37 PM	0:00:49	100
10	4331	872	872	11/3/2020	7:49:44 PM	0:14:07	99
7	3103	873	873	11/3/2020	7:50:48 PM	0:01:05	98
10	4332	874	874	11/3/2020	8:05:41 PM	0:14:52	100
10	4333	875	875	11/3/2020	8:09:46 PM	0:04:05	99
2	1082	876	876	11/3/2020	8:16:36 PM	0:06:51	100
2	1083	877	877	11/3/2020	8:17:41 PM	0:01:04	81
2	1084	878	878	11/3/2020	8:19:44 PM	0:02:04	81
10	4334	879	879	11/3/2020	8:23:35 PM	0:03:51	100
10	4335	880	880	11/3/2020	8:27:39 PM	0:04:04	99
2	1085	881	881	11/3/2020	8:29:43 PM	0:02:04	71
10	4336	882	882	11/3/2020	8:30:47 PM	0:01:03	100
2	1086	883	883	11/3/2020	8:31:36 PM	0:00:49	78
10	4337	884	884	11/3/2020	8:37:42 PM	0:06:06	100
2	1087	885	885	11/3/2020	8:38:46 PM	0:01:04	52
2	1088	886	886	11/3/2020	8:40:48 PM	0:02:03	57
10	4338	887	887	11/3/2020	8:40:51 PM	0:00:03	99
2	1089	888	888	11/3/2020	8:41:40 PM	0:00:49	35
2	1090	889	889	11/3/2020	8:44:42 PM	0:03:03	85
10	4339	890	890	11/3/2020	8:44:46 PM	0:00:03	100
2	1091	891	891	11/3/2020	8:46:36 PM	0:01:50	84
10	4340	892	892	11/3/2020	8:47:40 PM	0:01:04	100
2	1092	893	893	11/3/2020	8:48:44 PM	0:01:04	82
2	1093	894	894	11/3/2020	8:50:47 PM	0:02:04	82
10	4341	895	895	11/3/2020	8:50:51 PM	0:00:03	100
2	1094	896	896	11/3/2020	8:51:40 PM	0:00:49	74
10	4342	897	897	11/3/2020	8:53:43 PM	0:02:04	100
10	4343	898	898	11/3/2020	8:56:48 PM	0:03:04	99
2	1095	899	899	11/3/2020	8:58:37 PM	0:01:49	83
10	4344	900	900	11/3/2020	8:59:41 PM	0:01:04	100
2	1096	901	901	11/3/2020	9:01:45 PM	0:02:04	89
10	4345	902	902	11/3/2020	9:02:49 PM	0:01:04	100
10	4346	903	903	11/3/2020	9:08:39 PM	0:05:50	97
10	4347	904	904	11/3/2020	9:13:44 PM	0:05:05	99
10	4348	905	905	11/3/2020	9:17:48 PM	0:04:05	99
10	4349	906	906	11/3/2020	9:20:38 PM	0:02:50	99
7	3104	907	907	11/3/2020	9:24:43 PM	0:04:05	99

10	4350	908	908	11/3/2020	9:27:47 PM	0:03:05	100
7	3105	909	909	11/3/2020	9:28:37 PM	0:00:49	99
7	3106	910	910	11/3/2020	9:32:41 PM	0:04:05	68
10	4351	911	911	11/3/2020	10:13:38 PM	0:40:57	55
10	4352	912	912	11/3/2020	10:16:42 PM	0:03:03	98
10	4353	913	913	11/3/2020	10:22:47 PM	0:06:05	99
10	4354	914	914	11/3/2020	10:25:38 PM	0:02:51	100
10	4355	915	915	11/3/2020	10:28:43 PM	0:03:05	100
10	4356	916	916	11/3/2020	10:31:47 PM	0:03:05	99
10	4357	917	917	11/3/2020	10:34:38 PM	0:02:51	100
10	4358	918	918	11/3/2020	10:37:43 PM	0:03:05	100
10	4359	919	919	11/3/2020	10:40:47 PM	0:03:05	100
10	4360	920	920	11/3/2020	10:43:39 PM	0:02:52	100
10	4361	921	921	11/3/2020	10:45:44 PM	0:02:04	100
10	4362	922	922	11/3/2020	10:48:48 PM	0:03:05	100
2	1097	923	923	11/3/2020	10:50:37 PM	0:01:49	99
2	1098	924	924	11/3/2020	10:53:42 PM	0:03:04	94
10	4363	925	925	11/3/2020	10:55:46 PM	0:02:04	100
10	4364	926	926	11/3/2020	11:00:38 PM	0:04:52	98
10	4365	927	927	11/3/2020	11:03:43 PM	0:03:04	100
2	1099	928	928	11/3/2020	11:04:47 PM	0:01:04	68
10	4366	929	929	11/3/2020	11:06:51 PM	0:02:04	100
10	4367	930	930	11/3/2020	11:09:40 PM	0:02:50	100
2	1100	931	931	11/3/2020	11:10:44 PM	0:01:04	77
10	4368	932	932	11/3/2020	11:11:48 PM	0:01:03	100
2	1101	933	933	11/3/2020	11:15:37 PM	0:03:50	88
10	4369	934	934	11/3/2020	11:15:56 PM	0:00:19	100
10	4370	935	935	11/3/2020	11:18:45 PM	0:02:50	100
2	1102	936	936	11/3/2020	11:19:50 PM	0:01:04	99
10	4371	937	937	11/3/2020	11:20:39 PM	0:00:49	100
2	1103	938	938	11/3/2020	11:21:43 PM	0:01:04	65
2	1104	939	939	11/3/2020	11:22:45 PM	0:01:03	37
2	1105	940	940	11/3/2020	11:27:48 PM	0:05:03	95
2	1106	941	941	11/3/2020	11:28:37 PM	0:00:49	95
7	3107	942	942	11/3/2020	11:37:42 PM	0:09:05	66
2	1107	943	943	11/10/2020	12:58:43 PM		87
2	1108	944	944	11/10/2020	1:01:32 PM	0:02:49	81
2	1109	945	945	11/10/2020	1:03:36 PM	0:02:04	69
2	1110	946	946	11/10/2020	1:08:40 PM	0:05:04	98
2	1111	947	947	11/10/2020	1:15:45 PM	0:07:05	97
2	1112	948	948	11/10/2020	1:27:36 PM	0:11:51	95
2	1113	949	949	11/10/2020	1:28:40 PM	0:01:04	68
2	1114	950	950	11/10/2020	1:30:43 PM	0:02:03	81
2	1115	951	951	11/10/2020	1:37:33 PM	0:06:50	85
2	1116	952	952	11/10/2020	1:40:37 PM	0:03:04	81

2	1117	953	953	11/10/2020	1:42:41 PM	0:02:04	95
2	1118	954	954	11/10/2020	1:44:31 PM	0:01:50	94
2	1119	955	955	11/10/2020	1:54:37 PM	0:10:06	98
2	1120	956	956	11/10/2020	1:56:41 PM	0:02:04	87
2	1121	957	957	11/10/2020	1:58:31 PM	0:01:50	83
2	1122	958	958	11/10/2020	2:01:35 PM	0:03:04	85
2	1123	959	959	11/10/2020	2:02:39 PM	0:01:03	68
7	3108	960	960	11/10/2020	2:13:44 PM	0:11:05	97
7	3109	961	961	11/10/2020	2:21:34 PM	0:07:51	99
7	3110	962	962	11/10/2020	2:24:39 PM	0:03:04	100
7	3111	963	963	11/10/2020	2:33:45 PM	0:09:06	99
7	3112	964	964	11/10/2020	2:39:35 PM	0:05:50	90
7	3113	965	965	11/10/2020	2:45:40 PM	0:06:05	92
7	3114	966	966	11/10/2020	3:43:41 PM	0:58:01	82
2	1124	967	969	11/13/2020	11:38:15 AM		59
7	3115	968	970	11/13/2020	11:38:17 AM	0:00:02	21
7	3116	969	971	11/13/2020	11:38:18 AM	0:00:01	2
7	3117	970	972	11/13/2020	11:38:18 AM	0:00:00	23
7	3118	971	973	11/13/2020	12:59:07 PM	1:20:49	40
2	1125	972	976	11/20/2020	11:47:01 AM		12
7	3119	973	977	11/20/2020	11:53:03 AM	0:06:02	21

APPENDIX C – LOG ENTRIES FROM EMS USER LOG – OCTOBER 2020

Date and Time	Commands / Comments in red
10/21/2020 14:18:01	SubmitBatchCommand (execution duration: 1250ms): Batch 227 - Successfully synchronized results.
10/21/2020 14:18:02	SubmitBatchCommand (execution duration: 1156ms): Batch 226 - Successfully synchronized results.
10/21/2020 14:18:03	SubmitBatchCommand (execution duration: 1297ms): Batch 228 - Successfully synchronized results.
10/21/2020 14:18:05	SubmitBatchCommand (execution duration: 906ms): Batch 229 - Successfully synchronized results.
	The above four commands are the last 4 batches adjudicated before the database copy being written back to the database
10/21/2020 14:18:14	GetBatchesCommand (execution duration: 16ms): Cast vote record batch list was successfully retrieved (CvrSortOrder greater then 267).
10/21/2020 14:18:33	LoadResultsCommand (execution duration: 2688ms): Result file '1_1_7_3037_DETAIL.DVD' was loaded successfully.
	The above two statements are the Adjudication Module checking for new Batches, and then loading the DVD file from the NAS for the new one it encountered.
	Note that this batch never made it to the old Adjudication database, it eventually made it to the new one. We are at this point 18 seconds from the database copy.
10/21/2020 14:18:39	GetAdjudicationSupportStatusCommand (execution duration: 16ms): Adjudication status retrieved (Adjudication is enabled)
10/21/2020 14:18:52	GetAdjudicationSupportStatusCommand (execution duration: 0ms): Adjudication status retrieved (Adjudication is enabled)
	This is the only time that this command is seen in the log during the election period, and they happened 12 seconds BEFORE the copy and then again 1 second after.
10/21/2020 14:18:57	GetRelevantOutstackConditionsCommand (execution duration: 31ms): Successfully retrieved relevant outstack conditions
10/21/2020 14:19:26	GetRelevantOutstackConditionsCommand (execution duration: 0ms): Successfully retrieved relevant outstack conditions
	Again, the only time this command is seen, and to me it seems that it involves the new database trying to figure out the current adjudication status
10/21/2020 14:20:06	GetBatchesCommand (execution duration: 1156ms): Cast vote record batch list was successfully retrieved (CvrSortOrder greater then 0). Values of CvrSortOrder field for delivered batches are: 1
	This is the Adjudication Module looking for new batches on the NAS drive. As there are currently no batches (the new database was created with no records), it is looking for anything >0
10/21/2020 14:20:07	GetCastVoteRecordsCommand (execution duration: 219ms): Cast vote records for batch '1' successfully retrieved.
	1 Batch was found and retrieved
10/21/2020 14:20:25	GetBatchesCommand (execution duration: 47ms): Cast vote record batch list was successfully retrieved (CvrSortOrder greater then 1). Values of CvrSortOrder field for delivered batches are: 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268
	Now the Adjudication module checks for new batches again. Note that the 1st time it found only one, now it finds ALL the rest from 60 - 267 (no sign of 2-59)
10/21/2020 14:20:26	GetCastVoteRecordsCommand (execution duration: 219ms): Cast vote records for batch '60' successfully retrieved.
10/21/2020 14:20:30	GetCastVoteRecordsCommand (execution duration: 312ms): Cast vote records for batch '61' successfully retrieved.

10/21/2020 14:20:34	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '62' successfully retrieved.
10/21/2020 14:20:38	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '63' successfully retrieved.
10/21/2020 14:20:43	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '64' successfully retrieved.
10/21/2020 14:20:47	GetCastVoteRecordsCommand (execution duration: 344ms): Cast vote records for batch '65' successfully retrieved.
10/21/2020 14:20:51	GetCastVoteRecordsCommand (execution duration: 406ms): Cast vote records for batch '66' successfully retrieved.
10/21/2020 14:20:55	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '67' successfully retrieved.
10/21/2020 14:20:59	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '68' successfully retrieved.
10/21/2020 14:21:04	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '69' successfully retrieved.
10/21/2020 14:21:08	GetCastVoteRecordsCommand (execution duration: 359ms): Cast vote records for batch '70' successfully retrieved.
10/21/2020 14:21:12	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '71' successfully retrieved.
10/21/2020 14:21:16	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '72' successfully retrieved.
10/21/2020 14:21:20	GetCastVoteRecordsCommand (execution duration: 297ms): Cast vote records for batch '73' successfully retrieved.
10/21/2020 14:21:24	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '74' successfully retrieved.
10/21/2020 14:21:28	GetCastVoteRecordsCommand (execution duration: 344ms): Cast vote records for batch '75' successfully retrieved.
	The above log entries are the reloading of the batches 60 - 75
10/21/2020 14:21:29	GetCastVoteRecordImageCommand (execution duration: 0ms): Image for tabulator '10, batch '4001' and session '18' successfully retrieved.
	The Adjudication Module begins processing ballots needing adjudication (4001 = batch 1)
10/21/2020 14:21:33	GetCastVoteRecordsCommand (execution duration: 312ms): Cast vote records for batch '76' successfully retrieved.
10/21/2020 14:21:37	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '77' successfully retrieved.
10/21/2020 14:21:44	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '78' successfully retrieved.
10/21/2020 14:21:48	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '79' successfully retrieved.
10/21/2020 14:21:52	GetCastVoteRecordsCommand (execution duration: 375ms): Cast vote records for batch '80' successfully retrieved.
10/21/2020 14:21:56	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '81' successfully retrieved.
10/21/2020 14:22:00	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '82' successfully retrieved.
10/21/2020 14:22:04	GetCastVoteRecordsCommand (execution duration: 266ms): Cast vote records for batch '83' successfully retrieved.
10/21/2020 14:22:09	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '84' successfully retrieved.
10/21/2020 14:22:13	GetCastVoteRecordsCommand (execution duration: 344ms): Cast vote records for batch '85' successfully retrieved.
10/21/2020 14:22:17	GetCastVoteRecordsCommand (execution duration: 281ms): Cast vote records for batch '86' successfully retrieved.
10/21/2020 14:22:21	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '87' successfully retrieved.
10/21/2020 14:22:25	GetCastVoteRecordsCommand (execution duration: 297ms): Cast vote records for batch '88' successfully retrieved.

10/21/2020 14:22:29	GetCastVoteRecordsCommand (execution duration: 328ms): Cast vote records for batch '89' successfully retrieved.
10/21/2020 14:22:33	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '90' successfully retrieved.
10/21/2020 14:22:37	GetCastVoteRecordsCommand (execution duration: 266ms): Cast vote records for batch '91' successfully retrieved.
10/21/2020 14:22:41	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '92' successfully retrieved.
10/21/2020 14:22:46	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '93' successfully retrieved.
10/21/2020 14:22:50	GetCastVoteRecordsCommand (execution duration: 266ms): Cast vote records for batch '94' successfully retrieved.
10/21/2020 14:22:54	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '95' successfully retrieved.
10/21/2020 14:22:58	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '96' successfully retrieved.
10/21/2020 14:23:02	GetCastVoteRecordsCommand (execution duration: 359ms): Cast vote records for batch '97' successfully retrieved.
10/21/2020 14:23:06	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '98' successfully retrieved.
10/21/2020 14:23:10	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '99' successfully retrieved.
10/21/2020 14:23:14	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '100' successfully retrieved.
10/21/2020 14:23:18	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '101' successfully retrieved.
10/21/2020 14:23:23	GetCastVoteRecordsCommand (execution duration: 344ms): Cast vote records for batch '102' successfully retrieved.
10/21/2020 14:23:27	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '103' successfully retrieved.
10/21/2020 14:23:31	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '104' successfully retrieved.
10/21/2020 14:23:37	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '105' successfully retrieved.
10/21/2020 14:23:41	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '106' successfully retrieved.
10/21/2020 14:23:45	GetCastVoteRecordsCommand (execution duration: 312ms): Cast vote records for batch '107' successfully retrieved.
10/21/2020 14:23:49	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '108' successfully retrieved.
10/21/2020 14:23:53	GetCastVoteRecordsCommand (execution duration: 344ms): Cast vote records for batch '109' successfully retrieved.
10/21/2020 14:23:57	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '110' successfully retrieved.
10/21/2020 14:24:01	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '111' successfully retrieved.
10/21/2020 14:24:06	GetCastVoteRecordsCommand (execution duration: 328ms): Cast vote records for batch '112' successfully retrieved.
10/21/2020 14:24:10	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '113' successfully retrieved.
10/21/2020 14:24:13	GetCastVoteRecordsCommand (execution duration: 281ms): Cast vote records for batch '114' successfully retrieved.
10/21/2020 14:24:18	GetCastVoteRecordsCommand (execution duration: 297ms): Cast vote records for batch '115' successfully retrieved.
10/21/2020 14:24:22	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '116' successfully retrieved.
10/21/2020 14:24:26	GetCastVoteRecordsCommand (execution duration: 281ms): Cast vote records for batch '117' successfully retrieved.

10/21/2020 14:24:30	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '118' successfully retrieved.
10/21/2020 14:24:35	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '119' successfully retrieved.
10/21/2020 14:24:39	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '120' successfully retrieved.
10/21/2020 14:24:43	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '121' successfully retrieved.
10/21/2020 14:24:47	GetCastVoteRecordsCommand (execution duration: 328ms): Cast vote records for batch '122' successfully retrieved.
10/21/2020 14:24:51	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '123' successfully retrieved.
10/21/2020 14:24:55	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '124' successfully retrieved.
10/21/2020 14:25:00	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '125' successfully retrieved.
10/21/2020 14:25:04	GetCastVoteRecordsCommand (execution duration: 297ms): Cast vote records for batch '126' successfully retrieved.
10/21/2020 14:25:08	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '127' successfully retrieved.
10/21/2020 14:25:12	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '128' successfully retrieved.
10/21/2020 14:25:16	GetCastVoteRecordsCommand (execution duration: 297ms): Cast vote records for batch '129' successfully retrieved.
10/21/2020 14:25:20	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '130' successfully retrieved.
10/21/2020 14:25:25	GetCastVoteRecordsCommand (execution duration: 328ms): Cast vote records for batch '131' successfully retrieved.
10/21/2020 14:25:30	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '132' successfully retrieved.
10/21/2020 14:25:34	GetCastVoteRecordsCommand (execution duration: 266ms): Cast vote records for batch '133' successfully retrieved.
10/21/2020 14:25:38	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '134' successfully retrieved.
10/21/2020 14:25:42	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '135' successfully retrieved.
10/21/2020 14:25:47	GetCastVoteRecordsCommand (execution duration: 344ms): Cast vote records for batch '136' successfully retrieved.
10/21/2020 14:25:52	GetCastVoteRecordsCommand (execution duration: 266ms): Cast vote records for batch '137' successfully retrieved.
10/21/2020 14:25:57	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '138' successfully retrieved.
10/21/2020 14:26:01	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '139' successfully retrieved.
10/21/2020 14:26:05	GetCastVoteRecordsCommand (execution duration: 266ms): Cast vote records for batch '140' successfully retrieved.
10/21/2020 14:26:09	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '141' successfully retrieved.
10/21/2020 14:26:13	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '142' successfully retrieved.
10/21/2020 14:26:17	GetCastVoteRecordsCommand (execution duration: 344ms): Cast vote records for batch '143' successfully retrieved.
10/21/2020 14:26:21	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '144' successfully retrieved.
10/21/2020 14:26:25	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '145' successfully retrieved.
10/21/2020 14:26:29	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '146' successfully retrieved.

10/21/2020 14:26:33	GetCastVoteRecordsCommand (execution duration: 219ms): Cast vote records for batch '147' successfully retrieved.
10/21/2020 14:26:37	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '148' successfully retrieved.
10/21/2020 14:26:41	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '149' successfully retrieved.
10/21/2020 14:26:45	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '150' successfully retrieved.
10/21/2020 14:26:49	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '151' successfully retrieved.
10/21/2020 14:26:52	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '152' successfully retrieved.
10/21/2020 14:26:57	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '153' successfully retrieved.
10/21/2020 14:27:01	GetCastVoteRecordsCommand (execution duration: 297ms): Cast vote records for batch '154' successfully retrieved.
10/21/2020 14:27:05	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '155' successfully retrieved.
10/21/2020 14:27:10	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '156' successfully retrieved.
10/21/2020 14:27:14	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '157' successfully retrieved.
10/21/2020 14:27:18	GetCastVoteRecordsCommand (execution duration: 359ms): Cast vote records for batch '158' successfully retrieved.
10/21/2020 14:27:21	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '159' successfully retrieved.
10/21/2020 14:27:26	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '160' successfully retrieved.
10/21/2020 14:27:30	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '161' successfully retrieved.
10/21/2020 14:27:34	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '162' successfully retrieved.
10/21/2020 14:27:38	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '163' successfully retrieved.
10/21/2020 14:27:42	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '164' successfully retrieved.
10/21/2020 14:27:46	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '165' successfully retrieved.
10/21/2020 14:27:50	GetCastVoteRecordsCommand (execution duration: 266ms): Cast vote records for batch '166' successfully retrieved.
10/21/2020 14:27:56	GetCastVoteRecordsCommand (execution duration: 312ms): Cast vote records for batch '167' successfully retrieved.
10/21/2020 14:28:01	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '168' successfully retrieved.
10/21/2020 14:28:05	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '169' successfully retrieved.
10/21/2020 14:28:10	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '170' successfully retrieved.
10/21/2020 14:28:13	GetCastVoteRecordsCommand (execution duration: 266ms): Cast vote records for batch '171' successfully retrieved.
10/21/2020 14:28:18	GetCastVoteRecordsCommand (execution duration: 297ms): Cast vote records for batch '172' successfully retrieved.
10/21/2020 14:28:22	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '173' successfully retrieved.
10/21/2020 14:28:25	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '174' successfully retrieved.
10/21/2020 14:28:30	GetCastVoteRecordsCommand (execution duration: 344ms): Cast vote records for batch '175' successfully retrieved.

10/21/2020 14:28:34	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '176' successfully retrieved.
10/21/2020 14:28:38	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '177' successfully retrieved.
10/21/2020 14:28:42	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '178' successfully retrieved.
10/21/2020 14:28:46	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '179' successfully retrieved.
10/21/2020 14:28:50	GetCastVoteRecordsCommand (execution duration: 313ms): Cast vote records for batch '180' successfully retrieved.
10/21/2020 14:28:54	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '181' successfully retrieved.
10/21/2020 14:28:58	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '182' successfully retrieved.
10/21/2020 14:29:02	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '183' successfully retrieved.
10/21/2020 14:29:06	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '184' successfully retrieved.
10/21/2020 14:29:10	GetCastVoteRecordsCommand (execution duration: 312ms): Cast vote records for batch '185' successfully retrieved.
10/21/2020 14:29:14	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '186' successfully retrieved.
10/21/2020 14:29:18	GetCastVoteRecordsCommand (execution duration: 172ms): Cast vote records for batch '187' successfully retrieved.
10/21/2020 14:29:22	GetCastVoteRecordsCommand (execution duration: 438ms): Cast vote records for batch '188' successfully retrieved.
10/21/2020 14:29:26	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '189' successfully retrieved.
10/21/2020 14:29:30	GetCastVoteRecordsCommand (execution duration: 328ms): Cast vote records for batch '190' successfully retrieved.
10/21/2020 14:29:34	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '191' successfully retrieved.
10/21/2020 14:29:38	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '192' successfully retrieved.
10/21/2020 14:29:43	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '193' successfully retrieved.
10/21/2020 14:29:47	GetCastVoteRecordsCommand (execution duration: 281ms): Cast vote records for batch '194' successfully retrieved.
10/21/2020 14:29:52	GetCastVoteRecordsCommand (execution duration: 297ms): Cast vote records for batch '195' successfully retrieved.
10/21/2020 14:29:56	GetCastVoteRecordsCommand (execution duration: 266ms): Cast vote records for batch '196' successfully retrieved.
10/21/2020 14:30:00	GetCastVoteRecordsCommand (execution duration: 297ms): Cast vote records for batch '197' successfully retrieved.
10/21/2020 14:30:04	GetCastVoteRecordsCommand (execution duration: 281ms): Cast vote records for batch '198' successfully retrieved.
10/21/2020 14:30:08	GetCastVoteRecordsCommand (execution duration: 62ms): Cast vote records for batch '199' successfully retrieved.
10/21/2020 14:30:09	GetCastVoteRecordsCommand (execution duration: 203ms): Cast vote records for batch '200' successfully retrieved.
10/21/2020 14:30:13	GetCastVoteRecordsCommand (execution duration: 203ms): Cast vote records for batch '201' successfully retrieved.
10/21/2020 14:30:17	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '202' successfully retrieved.
10/21/2020 14:30:21	GetCastVoteRecordsCommand (execution duration: 281ms): Cast vote records for batch '203' successfully retrieved.
10/21/2020 14:30:25	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '204' successfully retrieved.

10/21/2020 14:30:29	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '205' successfully retrieved.
10/21/2020 14:30:33	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '206' successfully retrieved.
10/21/2020 14:30:37	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '207' successfully retrieved.
10/21/2020 14:30:41	GetCastVoteRecordsCommand (execution duration: 313ms): Cast vote records for batch '208' successfully retrieved.
10/21/2020 14:30:45	GetCastVoteRecordsCommand (execution duration: 313ms): Cast vote records for batch '209' successfully retrieved.
10/21/2020 14:30:50	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '210' successfully retrieved.
10/21/2020 14:30:54	GetCastVoteRecordsCommand (execution duration: 266ms): Cast vote records for batch '211' successfully retrieved.
10/21/2020 14:30:58	GetCastVoteRecordsCommand (execution duration: 266ms): Cast vote records for batch '212' successfully retrieved.
10/21/2020 14:31:02	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '213' successfully retrieved.
10/21/2020 14:31:06	GetCastVoteRecordsCommand (execution duration: 187ms): Cast vote records for batch '214' successfully retrieved.
10/21/2020 14:31:09	GetCastVoteRecordsCommand (execution duration: 94ms): Cast vote records for batch '215' successfully retrieved.
10/21/2020 14:31:11	GetCastVoteRecordsCommand (execution duration: 266ms): Cast vote records for batch '216' successfully retrieved.
10/21/2020 14:31:14	GetCastVoteRecordsCommand (execution duration: 47ms): Cast vote records for batch '217' successfully retrieved.
10/21/2020 14:31:15	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '218' successfully retrieved.
10/21/2020 14:31:19	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '219' successfully retrieved.
10/21/2020 14:31:23	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '220' successfully retrieved.
10/21/2020 14:31:27	GetCastVoteRecordsCommand (execution duration: 266ms): Cast vote records for batch '221' successfully retrieved.
10/21/2020 14:31:31	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '222' successfully retrieved.
10/21/2020 14:31:35	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '223' successfully retrieved.
10/21/2020 14:31:40	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '224' successfully retrieved.
10/21/2020 14:31:44	GetCastVoteRecordsCommand (execution duration: 328ms): Cast vote records for batch '225' successfully retrieved.
10/21/2020 14:31:48	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '226' successfully retrieved.
10/21/2020 14:31:52	GetCastVoteRecordsCommand (execution duration: 266ms): Cast vote records for batch '227' successfully retrieved.
10/21/2020 14:31:56	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '228' successfully retrieved.
10/21/2020 14:32:01	GetCastVoteRecordsCommand (execution duration: 328ms): Cast vote records for batch '229' successfully retrieved.
10/21/2020 14:32:05	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '230' successfully retrieved.
10/21/2020 14:32:09	GetCastVoteRecordsCommand (execution duration: 219ms): Cast vote records for batch '231' successfully retrieved.
10/21/2020 14:32:14	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '232' successfully retrieved.
10/21/2020 14:32:18	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '233' successfully retrieved.

10/21/2020 14:32:23	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '234' successfully retrieved.
10/21/2020 14:32:27	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '235' successfully retrieved.
10/21/2020 14:32:31	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '236' successfully retrieved.
10/21/2020 14:32:36	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '237' successfully retrieved.
10/21/2020 14:32:40	GetCastVoteRecordsCommand (execution duration: 250ms): Cast vote records for batch '238' successfully retrieved.
10/21/2020 14:32:44	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '239' successfully retrieved.
10/21/2020 14:32:48	GetCastVoteRecordsCommand (execution duration: 266ms): Cast vote records for batch '240' successfully retrieved.
10/21/2020 14:32:52	GetCastVoteRecordsCommand (execution duration: 219ms): Cast vote records for batch '241' successfully retrieved.
10/21/2020 14:32:57	GetCastVoteRecordsCommand (execution duration: 313ms): Cast vote records for batch '242' successfully retrieved.
10/21/2020 14:33:01	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '243' successfully retrieved.
10/21/2020 14:33:06	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '244' successfully retrieved.
10/21/2020 14:33:10	GetCastVoteRecordsCommand (execution duration: 297ms): Cast vote records for batch '245' successfully retrieved.
10/21/2020 14:33:14	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '246' successfully retrieved.
10/21/2020 14:33:18	GetCastVoteRecordsCommand (execution duration: 219ms): Cast vote records for batch '247' successfully retrieved.
10/21/2020 14:33:22	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '248' successfully retrieved.
10/21/2020 14:33:26	GetCastVoteRecordsCommand (execution duration: 344ms): Cast vote records for batch '249' successfully retrieved.
10/21/2020 14:33:31	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '250' successfully retrieved.
10/21/2020 14:33:34	GetCastVoteRecordsCommand (execution duration: 219ms): Cast vote records for batch '251' successfully retrieved.
10/21/2020 14:33:38	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '252' successfully retrieved.
10/21/2020 14:33:43	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '253' successfully retrieved.
10/21/2020 14:33:47	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '254' successfully retrieved.
10/21/2020 14:33:51	GetCastVoteRecordsCommand (execution duration: 281ms): Cast vote records for batch '255' successfully retrieved.
10/21/2020 14:33:55	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '256' successfully retrieved.
10/21/2020 14:33:59	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '257' successfully retrieved.
10/21/2020 14:34:03	GetCastVoteRecordsCommand (execution duration: 141ms): Cast vote records for batch '258' successfully retrieved.
10/21/2020 14:34:05	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '259' successfully retrieved.
10/21/2020 14:34:10	GetCastVoteRecordsCommand (execution duration: 344ms): Cast vote records for batch '260' successfully retrieved.
10/21/2020 14:34:13	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '261' successfully retrieved.
10/21/2020 14:34:17	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '262' successfully retrieved.

10/21/2020 14:34:21	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '263' successfully retrieved.
10/21/2020 14:34:25	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '264' successfully retrieved.
10/21/2020 14:34:30	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '265' successfully retrieved.
	The reload is almost complete
10/21/2020 14:34:33	GetCastVoteRecordImageCommand (execution duration: 16ms): Image for tabulator '10, batch '4001' and session '25' successfully retrieved.
	A second ballot from batch 1 goes to adjudication
10/21/2020 14:34:34	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '266' successfully retrieved.
10/21/2020 14:34:38	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '267' successfully retrieved.
	At this point we have reloaded all of the batches from the original database. Time elapsed since copy event: 11 minutes, 47 seconds
	This is 3 seconds (on average) per copied batch, .03 seconds (on average) per ballot.
10/21/2020 14:34:42	GetCastVoteRecordsCommand (execution duration: 234ms): Cast vote records for batch '268' successfully retrieved.
	This is the system actually loading up batch 3037, the last one that was saved before the database copy. (See line 7 above)
	At this point all but 58 batches and their ballots from the original Adjudication database are now copied to the new database

APPENDIX D – LOG ENTRIES FROM EMS USER LOG – MARCH 2021

Date and Time	Command / Comment
03/30/2021 14:57:16	GetCastVoteRecordImageCommand (execution duration: 16ms): Image for tabulator '30, batch '3044' and session '72' successfully retrieved.
	Adjudication Module requesting an image so that it can be adjudicated
03/30/2021 14:57:17	GetBatchesCommand (execution duration: 0ms): Cast vote record batch list was successfully retrieved (CvrSortOrder greater then 88).
03/30/2021 14:57:32	GetBatchesCommand (execution duration: 16ms): Cast vote record batch list was successfully retrieved (CvrSortOrder greater then 88).
03/30/2021 14:57:47	GetBatchesCommand (execution duration: 16ms): Cast vote record batch list was successfully retrieved (CvrSortOrder greater then 88).
03/30/2021 14:58:02	GetBatchesCommand (execution duration: 0ms): Cast vote record batch list was successfully retrieved (CvrSortOrder greater then 88).
	Normal checks for new batches. We are 54 seconds from database copy event
03/30/2021 14:58:41	GetAdjudicationSupportStatusCommand (execution duration: 16ms): Adjudication status retrieved (Adjudication is enabled)
03/30/2021 14:58:57	GetAdjudicationSupportStatusCommand (execution duration: 16ms): Adjudication status retrieved (Adjudication is enabled)
	Like in the November 2020 election these two commands appear right before and right after the copy event.
03/30/2021 14:58:59	GetRelevantOutstackConditionsCommand (execution duration: 47ms): Successfully retrieved relevant outstack conditions
03/30/2021 14:59:15	GetRelevantOutstackConditionsCommand (execution duration: 0ms): Successfully retrieved relevant outstack conditions
	These two commands also were found just after the database copy event
03/30/2021 14:59:52	GetBatchesCommand (execution duration: 156ms): Cast vote record batch list was successfully retrieved (CvrSortOrder greater then 0). Values of CvrSortOrder field for delivered batches are: 45, 46
03/30/2021 14:59:52	GetCastVoteRecordsCommand (execution duration: 406ms): Cast vote records for batch '46' successfully retrieved.
03/30/2021 14:59:56	GetCastVoteRecordsCommand (execution duration: 109ms): Cast vote records for batch '47' successfully retrieved.
	The select batches from the original Adjudication database begin being encountered, although like November 2020, not all at once.
03/30/2021 15:00:14	GetBatchesCommand (execution duration: 16ms): Cast vote record batch list was successfully retrieved (CvrSortOrder greater then 46). Values of CvrSortOrder field for delivered batches are: 48, 49
03/30/2021 15:00:14	GetCastVoteRecordsCommand (execution duration: 78ms): Cast vote records for batch '49' successfully retrieved.
03/30/2021 15:00:18	GetCastVoteRecordsCommand (execution duration: 109ms): Cast vote records for batch '50' successfully retrieved.

03/30/2021 15:00:36	GetBatchesCommand (execution duration: 31ms): Cast vote record batch list was successfully retrieved (CvrSortOrder greater then 49). Values of CvrSortOrder field for delivered batches are: 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88
03/30/2021 15:00:36	GetCastVoteRecordsCommand (execution duration: 94ms): Cast vote records for batch '52' successfully retrieved.
03/30/2021 15:00:40	GetCastVoteRecordsCommand (execution duration: 78ms): Cast vote records for batch '53' successfully retrieved.
03/30/2021 15:00:43	GetCastVoteRecordsCommand (execution duration: 94ms): Cast vote records for batch '54' successfully retrieved.
03/30/2021 15:00:47	GetCastVoteRecordsCommand (execution duration: 78ms): Cast vote records for batch '55' successfully retrieved.
03/30/2021 15:00:51	GetCastVoteRecordsCommand (execution duration: 78ms): Cast vote records for batch '56' successfully retrieved.
03/30/2021 15:00:54	GetCastVoteRecordsCommand (execution duration: 94ms): Cast vote records for batch '57' successfully retrieved.
03/30/2021 15:00:58	GetCastVoteRecordsCommand (execution duration: 78ms): Cast vote records for batch '58' successfully retrieved.
03/30/2021 15:01:01	GetCastVoteRecordsCommand (execution duration: 125ms): Cast vote records for batch '59' successfully retrieved.
03/30/2021 15:01:05	GetCastVoteRecordsCommand (execution duration: 78ms): Cast vote records for batch '60' successfully retrieved.
03/30/2021 15:01:08	GetCastVoteRecordsCommand (execution duration: 94ms): Cast vote records for batch '61' successfully retrieved.
03/30/2021 15:01:11	GetCastVoteRecordsCommand (execution duration: 94ms): Cast vote records for batch '62' successfully retrieved.
03/30/2021 15:01:15	GetCastVoteRecordsCommand (execution duration: 94ms): Cast vote records for batch '63' successfully retrieved.
03/30/2021 15:01:19	GetCastVoteRecordsCommand (execution duration: 125ms): Cast vote records for batch '64' successfully retrieved.
03/30/2021 15:01:22	GetCastVoteRecordsCommand (execution duration: 78ms): Cast vote records for batch '65' successfully retrieved.
03/30/2021 15:01:25	GetCastVoteRecordsCommand (execution duration: 78ms): Cast vote records for batch '66' successfully retrieved.
03/30/2021 15:01:29	GetCastVoteRecordsCommand (execution duration: 94ms): Cast vote records for batch '67' successfully retrieved.
03/30/2021 15:01:33	GetCastVoteRecordsCommand (execution duration: 94ms): Cast vote records for batch '68' successfully retrieved.
03/30/2021 15:01:36	GetCastVoteRecordsCommand (execution duration: 94ms): Cast vote records for batch '69' successfully retrieved.
03/30/2021 15:01:40	GetCastVoteRecordsCommand (execution duration: 94ms): Cast vote records for batch '70' successfully retrieved.

03/30/2021 15:01:44	GetCastVoteRecordsCommand (execution duration: 109ms): Cast vote records for batch '71' successfully retrieved.
03/30/2021 15:01:48	GetCastVoteRecordsCommand (execution duration: 94ms): Cast vote records for batch '72' successfully retrieved.
03/30/2021 15:01:51	GetCastVoteRecordsCommand (execution duration: 94ms): Cast vote records for batch '73' successfully retrieved.
03/30/2021 15:01:55	GetCastVoteRecordsCommand (execution duration: 78ms): Cast vote records for batch '74' successfully retrieved.
03/30/2021 15:01:59	GetCastVoteRecordsCommand (execution duration: 109ms): Cast vote records for batch '75' successfully retrieved.
03/30/2021 15:02:02	GetCastVoteRecordsCommand (execution duration: 109ms): Cast vote records for batch '76' successfully retrieved.
03/30/2021 15:02:06	GetCastVoteRecordsCommand (execution duration: 94ms): Cast vote records for batch '77' successfully retrieved.
03/30/2021 15:02:09	GetCastVoteRecordsCommand (execution duration: 78ms): Cast vote records for batch '78' successfully retrieved.
03/30/2021 15:02:12	GetCastVoteRecordsCommand (execution duration: 78ms): Cast vote records for batch '79' successfully retrieved.
03/30/2021 15:02:16	GetCastVoteRecordsCommand (execution duration: 63ms): Cast vote records for batch '80' successfully retrieved.
03/30/2021 15:02:19	GetCastVoteRecordsCommand (execution duration: 109ms): Cast vote records for batch '81' successfully retrieved.
03/30/2021 15:02:22	GetCastVoteRecordsCommand (execution duration: 94ms): Cast vote records for batch '82' successfully retrieved.
03/30/2021 15:02:26	GetCastVoteRecordsCommand (execution duration: 94ms): Cast vote records for batch '83' successfully retrieved.
03/30/2021 15:02:29	GetCastVoteRecordsCommand (execution duration: 78ms): Cast vote records for batch '84' successfully retrieved.
03/30/2021 15:02:33	GetCastVoteRecordsCommand (execution duration: 109ms): Cast vote records for batch '85' successfully retrieved.
03/30/2021 15:02:37	GetCastVoteRecordsCommand (execution duration: 78ms): Cast vote records for batch '86' successfully retrieved.
03/30/2021 15:02:40	GetCastVoteRecordsCommand (execution duration: 94ms): Cast vote records for batch '87' successfully retrieved.
03/30/2021 15:02:44	GetCastVoteRecordsCommand (execution duration: 94ms): Cast vote records for batch '88' successfully retrieved.
03/30/2021 15:02:47	GetCastVoteRecordsCommand (execution duration: 47ms): Cast vote records for batch '89' successfully retrieved.
03/30/2021 15:03:04	GetBatchesCommand (execution duration: 16ms): Cast vote record batch list was successfully retrieved (CvrSortOrder greater then 88).

03/30/2021 15:03:19	SubmitBatchCommand (execution duration: 203ms): Batch 63 - Successfully synchronized results.
03/30/2021 15:03:19	GetBatchesCommand (execution duration: 16ms): Cast vote record batch list was successfully retrieved (CvrSortOrder greater then 88).
03/30/2021 15:03:19	SubmitBatchCommand (execution duration: 141ms): Batch 59 - Successfully synchronized results.
03/30/2021 15:03:20	SubmitBatchCommand (execution duration: 125ms): Batch 61 - Successfully synchronized results.
03/30/2021 15:03:20	SubmitBatchCommand (execution duration: 78ms): Batch 62 - Successfully synchronized results.
03/30/2021 15:03:20	SubmitBatchCommand (execution duration: 78ms): Batch 64 - Successfully synchronized results.
03/30/2021 15:03:20	SubmitBatchCommand (execution duration: 62ms): Batch 65 - Successfully synchronized results.
03/30/2021 15:03:21	SubmitBatchCommand (execution duration: 109ms): Batch 67 - Successfully synchronized results.
03/30/2021 15:03:21	SubmitBatchCommand (execution duration: 78ms): Batch 68 - Successfully synchronized results.
03/30/2021 15:03:21	SubmitBatchCommand (execution duration: 62ms): Batch 69 - Successfully synchronized results.
03/30/2021 15:03:21	SubmitBatchCommand (execution duration: 94ms): Batch 70 - Successfully synchronized results.
03/30/2021 15:03:21	SubmitBatchCommand (execution duration: 62ms): Batch 71 - Successfully synchronized results.
03/30/2021 15:03:22	SubmitBatchCommand (execution duration: 63ms): Batch 72 - Successfully synchronized results.
03/30/2021 15:03:22	SubmitBatchCommand (execution duration: 62ms): Batch 74 - Successfully synchronized results.
03/30/2021 15:03:22	SubmitBatchCommand (execution duration: 78ms): Batch 56 - Successfully synchronized results.
03/30/2021 15:03:34	GetBatchesCommand (execution duration: 0ms): Cast vote record batch list was successfully retrieved (CvrSortOrder greater then 88).
03/30/2021 15:03:40	SubmitBatchCommand (execution duration: 188ms): Batch 75 - Successfully synchronized results.
03/30/2021 15:03:48	SubmitBatchCommand (execution duration: 172ms): Batch 79 - Successfully synchronized results.
03/30/2021 15:03:48	SubmitBatchCommand (execution duration: 125ms): Batch 77 - Successfully synchronized results.
03/30/2021 15:03:48	SubmitBatchCommand (execution duration: 109ms): Batch 78 - Successfully synchronized results.

03/30/2021 15:03:49	GetBatchesCommand (execution duration: 0ms): Cast vote record batch list was successfully retrieved (CvrSortOrder greater then 88).
03/30/2021 15:03:51	SubmitBatchCommand (execution duration: 125ms): Batch 80 - Successfully synchronized results.
03/30/2021 15:04:04	GetBatchesCommand (execution duration: 16ms): Cast vote record batch list was successfully retrieved (CvrSortOrder greater then 88).
03/30/2021 15:04:14	SubmitBatchCommand (execution duration: 203ms): Batch 82 - Successfully synchronized results.
03/30/2021 15:04:18	SubmitBatchCommand (execution duration: 156ms): Batch 84 - Successfully synchronized results.
03/30/2021 15:04:19	SubmitBatchCommand (execution duration: 125ms): Batch 83 - Successfully synchronized results.
03/30/2021 15:04:19	GetBatchesCommand (execution duration: 0ms): Cast vote record batch list was successfully retrieved (CvrSortOrder greater then 88).
03/30/2021 15:04:21	SubmitBatchCommand (execution duration: 156ms): Batch 85 - Successfully synchronized results.
03/30/2021 15:04:34	GetBatchesCommand (execution duration: 16ms): Cast vote record batch list was successfully retrieved (CvrSortOrder greater then 88).
03/30/2021 15:04:48	SubmitBatchCommand (execution duration: 109ms): Batch 89 - Successfully synchronized results.
03/30/2021 15:04:49	SubmitBatchCommand (execution duration: 141ms): Batch 87 - Successfully synchronized results.
03/30/2021 15:04:49	SubmitBatchCommand (execution duration: 156ms): Batch 88 - Successfully synchronized results.
	Like November 2020, the rest of the batches from the original Adjudication database are copied and reprocessed.
	The order, however, is not the same, and there is a referenced to a batch 89 which does not have a record in either Adjudication database.

REFERENCE A – DATABASES AND TABLES

In order to assist other researchers, who may wish to examine these findings or perform additional analysis, here are the most important databases and tables which were used in this analysis.

Main election databases:

November 2020 General Election:
[2020 Mesa County General-2020-09-05-00-10-20]

April 2021 Municipal Election:
[City of Grand Junction-Municipal Election 2021-2021-02-05-16-01-32]

Primary Tables (specifically related to vote totals):
ResultContainer: (Batch level raw vote data)
ResultSplitter: (Vote Data by Polling Location
ChoiceResult: (Raw aggregated vote data)
CastVoteRecord: (Raw per-ballot list)
Choice: (All Candidates/Choices)
Contest: (All contests in Election)
Tabulator: (All defined tabulators)

Stored Procedures (useful for checking final results):
GetContestResults: Displays current results of any or all contests
GetContestStatistics: Displays stats for any or all contests, including undervotes and overvotes

Adjudication databases:

November 2020 General Election:
[AdjudicableBallotStore_2020_Mesa_County_General_2020-10-01_12:18:50] (before copy)
[AdjudicableBallotStore_2020_Mesa_County_General_2020-10-21_14:18:51] (after copy)

April 2021 Municipal Election:
[AdjudicableBallotStore_City_Of_Grand_Junction_Municipal_Election_2021_2021-03-18_10:48:14] (before copy)

[AdjudicableBallotStore_City_Of_Grand_Junction_Municipal_Election_2021_2021-03-30_14:58:56] (after copy)

Primary Tables:
Batches: Raw batch information
SerializedAdjudicableBallots: Contains one data record for each ballot received.
BallotStatusEvents: Every ballot with Adjudication status. New records for same ballot whenever any change occurs in the status of the ballot.

REFERENCE B – SCANNER SPEED

4.5 Processing Rate

The central scanning device's processing rate also depends on the handling and poll verification activities.

The number of ballots per minute depends on the width or length of the ballot. The following table documents the approximate scanning speed of the ICC scanners.

Scanner	Ballot Size	Pages per Minute (ppm) Scanned
Canon DR-G1130	8.5" x 11"	Approximately 100 ppm, as per Dominion Voting's Quality Assurance test results.
Canon DR-G2140	8.5" x 22"	Aproximately 70 ppm, as per Dominion Voting's Quality Assurance test results.
Canon DR-M160II	8.5" x 11"	Approximately 60 ppm, as per Dominion Voting's Quality Assurance test results.

https://www.sos.state.co.us/pubs/elections/VotingSystems/DVS-DemocracySuite511/documentation/2-03-ICC-FunctionalityDescription-5-11-CO.pdf

REFERENCE C – SCANNERS USED BY MESA COUNTY

DESCRIPTION	QTY	UNIT PRICE	EXTENSION
Central Scanning Hardware & Software License			
ImageCast Central Includes:	4	$18,500	$74,000
Canon DR-G1130 high speed document scanner.			
- ImageCast® Central Software including third party Kofax VRS 4.5 software.			
- OptiPlex 9020 All-in-One Series with pre-loaded software			
- One (1) iButton Programmer and (1) iButton Key Switch & Cat5 RJ 45			
Cables			
- 12 months Hardware Warranty			
- 12 months Firmware License			

https://onbase.mesacounty.us/OnBaseAgendaOnline/Documents/Downloadfile/Special_Meeting_1018_Agenda_Packet_8_24_2021_1_00_00_PM.pdf?documentType=5&meetingId=1018&isAttachment=True

REFERENCE D – DATA MOVEMENT FROM BATCHES TO VOTES

Below please find an example of how the data moves through the system from the batch to its votes, and how the ballot level vote data is obfuscated in the process. Blank and irrelevant fields are omitted.

When batch 4025 was received in the Mesa County November 2020 election, the following record was created in the *batches* table of the Adjudication database.

Field	Value
TabulatorId	10
BatchId	4025
Name	Tabulator 10 - Batch 4025
LoadOrder	60
CreationTime	10/21/20 2:20 PM
ModificationTime	10/22/20 10:33 AM
BallotCount	99
HasAdjudicatedBallots	1

After all adjudication tasks were complete, a record exists in the Main election database *ResultContainer* table.

Field	Value
Id	60
containerType	CVR
resultState	Published
batchId	4025
fileName	1_1_10_4025_DETAIL.DVD
tabulatorId	10
CvrSortOrder	60
TimeStamp	10/19/20 4:12 PM

This table serves as a record of each individual batch received, and the *batchId* field (4025 in this case) references the *BatchId* of the Adjudication database's *Batches* table, as shown above. This is the first evidence of a break in referential integrity, as there is no database-level relationship between these two tables. In

other words, the *Batches* record in the Adjudication database can be removed or altered without any warning or error being generated by the database.

Of note also is that the number of ballots which exists in each batch is not a part of the *ResultContainer* table. This makes reconciling the data in the Main election database tables much more difficult.

From here, the information goes to three other tables of interest in the Main election database. The first is *CastVoteRecord*, which contains the ballot-level vote data.

Id	ResultContainerId	RecordId	PrecinctPortionId	IsCurrent	OriginalCvrId	OutstackConditions	BallotTypeId	tabulatorId	batchId
5892	60	1	124	1	NULL	1088	5	10	4025
5893	60	2	103	1	NULL	1088	7	10	4025
5894	60	3	66	0	NULL	256	8	10	4025
5895	60	4	44	1	NULL	1088	3	10	4025
5896	60	5	111	1	NULL	1088	7	10	4025
5897	60	6	63	1	NULL	1088	7	10	4025
5898	60	7	79	1	NULL	0	7	10	4025
5899	60	8	98	1	NULL	0	7	10	4025
5900	60	9	22	1	NULL	1088	3	10	4025
5901	60	10	22	0	NULL	256	3	10	4025
5902	60	11	111	1	NULL	0	7	10	4025
5903	60	12	134	1	NULL	0	7	10	4025
5904	60	13	134	0	NULL	256	7	10	4025
5905	60	14	4	1	NULL	1088	1	10	4025
5906	60	15	100	1	NULL	1088	7	10	4025
5907	60	16	98	1	NULL	1088	7	10	4025
5908	60	17	40	1	NULL	0	1	10	4025
5909	60	18	129	1	NULL	1088	7	10	4025
5910	60	19	124	1	NULL	1088	5	10	4025
5911	60	20	108	1	NULL	0	7	10	4025
5912	60	21	131	0	NULL	5	7	10	4025
5913	60	22	41	0	NULL	1344	1	10	4025
5914	60	23	42	0	NULL	256	1	10	4025
5915	60	24	13	1	NULL	1088	2	10	4025
5916	60	25	42	1	NULL	1088	1	10	4025
5917	60	26	2	1	NULL	0	1	10	4025
5918	60	27	42	1	NULL	0	1	10	4025
5919	60	28	60	1	NULL	0	7	10	4025
5920	60	29	95	1	NULL	0	7	10	4025
5921	60	30	100	1	NULL	1088	7	10	4025

5922	60	31	10	1	NULL	1088	1	10	4025
5923	60	32	117	1	NULL	1088	5	10	4025
5924	60	33	101	1	NULL	0	7	10	4025
5925	60	34	10	0	NULL	256	1	10	4025
5926	60	35	102	1	NULL	1088	7	10	4025
5927	60	36	60	1	NULL	0	7	10	4025
5928	60	37	101	1	NULL	1088	7	10	4025
5929	60	38	7	1	NULL	1088	1	10	4025
5930	60	39	41	1	NULL	0	1	10	4025
5931	60	40	101	0	NULL	1344	7	10	4025
5932	60	41	62	1	NULL	1088	8	10	4025
5933	60	42	46	1	NULL	1088	1	10	4025
5934	60	43	70	1	NULL	0	8	10	4025
5935	60	44	63	1	NULL	1088	7	10	4025
5936	60	45	100	1	NULL	0	7	10	4025
5937	60	46	94	1	NULL	1088	7	10	4025
5938	60	47	101	1	NULL	1088	7	10	4025
5939	60	48	79	1	NULL	1088	7	10	4025
5940	60	49	131	0	NULL	1	7	10	4025
5941	60	50	3	1	NULL	0	1	10	4025
5942	60	51	108	1	NULL	0	7	10	4025
5943	60	52	63	1	NULL	0	7	10	4025
5944	60	53	105	1	NULL	0	7	10	4025
5945	60	54	100	1	NULL	0	7	10	4025
5946	60	55	17	1	NULL	0	3	10	4025
5947	60	56	40	1	NULL	0	1	10	4025
5948	60	57	101	1	NULL	0	7	10	4025
5949	60	58	60	1	NULL	1088	7	10	4025
5950	60	59	22	0	NULL	1	3	10	4025
5951	60	60	134	1	NULL	1088	7	10	4025
5952	60	61	103	1	NULL	1088	7	10	4025
5953	60	62	60	1	NULL	0	7	10	4025
5954	60	63	7	1	NULL	1088	1	10	4025
5955	60	64	52	1	NULL	0	3	10	4025
5956	60	65	100	1	NULL	1088	7	10	4025
5957	60	66	100	0	NULL	256	7	10	4025
5958	60	67	101	1	NULL	0	7	10	4025
5959	60	68	79	1	NULL	0	7	10	4025
5960	60	69	100	1	NULL	0	7	10	4025
5961	60	70	129	1	NULL	1088	7	10	4025
5962	60	71	98	1	NULL	0	7	10	4025
5963	60	72	138	1	NULL	0	7	10	4025
5964	60	73	119	1	NULL	0	7	10	4025

5965	60	74	50	1	NULL	1088	1	10	4025
5966	60	75	102	1	NULL	0	7	10	4025
5967	60	76	100	1	NULL	0	7	10	4025
5968	60	77	22	1	NULL	1088	3	10	4025
5969	60	78	60	1	NULL	1088	7	10	4025
5970	60	79	44	0	NULL	1344	3	10	4025
5971	60	80	101	1	NULL	0	7	10	4025
5972	60	81	111	1	NULL	1088	7	10	4025
5973	60	82	129	1	NULL	0	7	10	4025
5974	60	83	98	1	NULL	1088	7	10	4025
5975	60	84	111	1	NULL	1088	7	10	4025
5976	60	85	34	1	NULL	0	1	10	4025
5977	60	86	35	1	NULL	1088	1	10	4025
5978	60	87	17	1	NULL	0	3	10	4025
5979	60	88	50	1	NULL	1088	1	10	4025
5980	60	89	104	1	NULL	0	7	10	4025
5981	60	90	104	1	NULL	0	7	10	4025
5982	60	91	30	1	NULL	0	1	10	4025
5983	60	92	134	1	NULL	0	7	10	4025
5984	60	93	134	1	NULL	1088	7	10	4025
5985	60	94	101	1	NULL	1088	7	10	4025
5986	60	95	105	1	NULL	1088	7	10	4025
5987	60	96	52	1	NULL	1088	3	10	4025
5988	60	97	105	0	NULL	256	7	10	4025
5989	60	98	100	1	NULL	1088	7	10	4025
5990	60	99	98	0	NULL	1344	7	10	4025
9514	60	3	66	1	5894	0	8	10	4025
9515	60	10	22	1	5901	0	3	10	4025
9516	60	13	134	1	5904	0	7	10	4025
9517	60	22	41	1	5913	1088	1	10	4025
9518	60	23	42	1	5914	0	1	10	4025
9519	60	21	131	1	5912	1092	7	10	4025
9620	60	34	10	1	5925	0	1	10	4025
9621	60	49	131	1	5940	1088	7	10	4025
9622	60	40	101	1	5931	1088	7	10	4025
9623	60	59	22	1	5950	1	3	10	4025
9624	60	66	100	1	5957	256	7	10	4025
9625	60	79	44	1	5970	1088	3	10	4025
9626	60	99	98	1	5990	1088	7	10	4025
9728	60	97	105	1	5988	0	7	10	4025

There is one record for each ballot in batch 4025, and then an additional record for each ballot which went through the manual adjudication process. The *IsCurrent* field indicates which of the two ballot records is the latest one. No timestamp exists in this table to be able to determine the time the ballot data was entered or modified.

Unlike a “Cast Vote Record” file, this table contains no vote information.

Next is *ResultSplitter*. Batch 4025 was separated into 42 rows in this table:

Id	numberOfValid	pollingDistrictId	resultContainerId	numberOfWriteIns	tabulatorId	ballotId
2008	1	30	60	0	10	1
2007	2	104	60	0	10	7
2006	1	35	60	0	10	1
2005	1	34	60	0	10	1
2004	2	50	60	0	10	1
2003	1	119	60	0	10	7
2002	1	138	60	0	10	7
2001	2	52	60	0	10	3
2000	2	17	60	0	10	3
1999	3	105	60	0	10	7
1998	1	3	60	0	10	1
1997	1	94	60	0	10	7
1996	1	70	60	0	10	8
1995	1	46	60	0	10	1
1994	1	62	60	0	10	8
1993	2	7	60	0	10	1
1992	2	102	60	0	10	7
1991	8	101	60	0	10	7
1990	1	117	60	0	10	5
1989	2	10	60	0	10	1
1988	1	95	60	0	10	7
1987	5	60	60	0	10	7
1986	1	2	60	0	10	1
1985	1	13	60	0	10	2
1984	3	42	60	0	10	1
1983	2	41	60	0	10	1
1982	2	131	60	0	10	7
1981	2	108	60	0	10	7
1980	3	129	60	0	10	7
1979	2	40	60	0	10	1
1978	9	100	60	0	10	7

1977	1	4	60	0	10	1
1976	5	134	60	0	10	7
1975	4	22	60	1	10	3
1974	5	98	60	0	10	7
1973	3	79	60	0	10	7
1972	3	63	60	0	10	7
1971	4	111	60	0	10	7
1970	2	44	60	0	10	3
1969	1	66	60	0	10	8
1968	2	103	60	0	10	7
1967	2	124	60	0	10	5

The 99 ballots in batch 4025 are segregated here by polling district number. No vote information appears in this table, and this table links back to its corresponding record in the *ResultContainer* table through the *resultContainerId* field. Again, this table contains no specific vote information for the ballots.

Next is the table *ChoiceResult*. Because of how the records are aggregated, there are over 1,600 records for batch 4025. For brevity, only the first 49 records are displayed.

Id	numberOfVotes	isValid	contestResultId	pollingDistrictId	tabulatorId	resultContainerId	choiceId	partyId
72135	2	1	56749	63	10	60	82	0
72136	1	1	56749	63	10	60	83	0
72137	2	1	56750	63	10	60	88	0
72138	1	1	56750	63	10	60	89	0
72139	2	1	56751	79	10	60	2	0
72140	1	1	56751	79	10	60	1	0
72141	2	1	56752	79	10	60	23	5
72142	1	1	56752	79	10	60	22	2
72143	2	1	56753	79	10	60	27	5
72144	1	1	56753	79	10	60	28	2
72145	2	1	56754	79	10	60	32	5
72146	1	1	56754	79	10	60	31	2
72147	2	1	56755	79	10	60	35	5
72148	1	1	56755	79	10	60	36	2
72149	2	1	56756	79	10	60	38	5
72150	2	1	56757	79	10	60	39	5
72151	1	1	56757	79	10	60	40	2
72152	2	1	56758	79	10	60	42	5
72153	1	1	56758	79	10	60	41	2

72154	3	1	56759	79	10	60	44	0
72155	3	1	56760	79	10	60	46	0
72156	3	1	56761	79	10	60	48	0
72157	3	1	56762	79	10	60	50	0
72158	2	1	56763	79	10	60	74	0
72159	1	1	56763	79	10	60	75	0
72160	3	1	56764	79	10	60	76	0
72161	3	1	56765	79	10	60	78	0
72162	3	1	56766	79	10	60	80	0
72163	2	1	56767	79	10	60	53	0
72164	1	1	56767	79	10	60	52	0
72165	3	1	56768	79	10	60	55	0
72166	2	1	56769	79	10	60	56	0
72167	1	1	56769	79	10	60	57	0
72134	3	1	56748	63	10	60	73	0
72133	1	1	56747	63	10	60	71	0
72132	2	1	56747	63	10	60	70	0
72131	3	1	56746	63	10	60	68	0
72130	2	1	56745	63	10	60	66	0
72129	1	1	56745	63	10	60	67	0
72128	2	1	56744	63	10	60	65	0
72127	1	1	56744	63	10	60	64	0
72126	1	1	56743	63	10	60	63	0
72125	2	1	56743	63	10	60	62	0
72124	1	1	56742	63	10	60	60	0
72123	2	1	56742	63	10	60	61	0
72122	2	1	56741	63	10	60	59	0
72121	1	1	56741	63	10	60	58	0
72120	1	1	56740	63	10	60	57	0
72119	2	1	56740	63	10	60	56	0

This table, the only table which actually has a record of the vote totals used to produce reports, aggregates the votes by polling district and candidate or issue choice. As an example, the fifth line of data specifies that there are two votes for Donald Trump (*choiceId* 2, which references the *internalMachineId field* of the table *Choice*) from polling district 3075539035 – GJ (*pollingDistrictId* 79, which references the *internalMachineId* field of the table *PollingDistrict*).

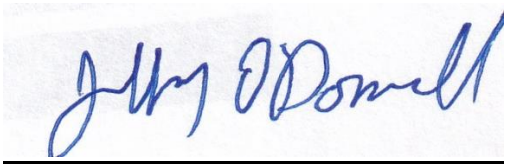
From this table, and the associated tables it links to, all reports are generated. As this is the only table which records vote choices, this is a single point of attack

or failure for the entire vote counting process of the Dominion system. Changes can be made to this table by any process, for instance changing the number of votes or the candidate, would be undetectable as such changes do not affect any other records in any other tables. Nor would such changes require alterations of any other records in any other tables.

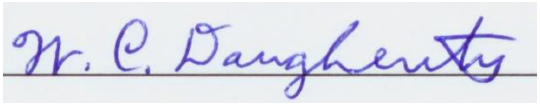
Additionally, there is no way to consistently link a particular vote shown in the *ChoiceResult* table to its original ballot within the batch.

The foregoing Forensic Examination and Report was prepared by us, and we are responsible for its content.

The 19th day of March 2022.



Jeffrey O'Donnell
Chief Information Officer
Ordros Analytics



Walter C. Daugherity
Senior Lecturer Emeritus
Department of Computer Science and Engineering
Texas A&M University

BIOGRAPHY

Jeffrey O'Donnell is a Full Stack software and database developer and analyst. He holds Bachelor's degrees in Computer Science and Mathematics from the University of Pittsburgh.

Over the last 40 years, Mr. O'Donnell has worked and consulted for numerous private sector corporations, including Rockwell International, Westinghouse Electric Nuclear, General Defense, U.S. Steel, Mellon Bank, IOTA 360, and the Penn State Applied Research Laboratory. For several years he also delivered and created computer science curriculum for the Community College of Allegheny County.

For the last two decades, Mr. O'Donnell has developed numerous "big data" analysis systems, including systems to provide short-term stock market investors with new types of research and predictive analytics.

He currently is President of Qest Development, a full-service software consulting and publishing company, and is Chief Information Officer of Ordros Analytics, which specializes in election analytics of all types.

Dr. Walter C. Daugherity is a computer consultant and also Senior Lecturer Emeritus in the Department of Computer Science and Engineering at Texas A&M University. He graduated from Oklahoma Christian University with a degree in mathematics, and then earned master's and doctor's degrees from Harvard University, which he attended on a Prize Fellowship from the National Science Foundation.

As a computer expert he has consulted for major national and international firms, and for government agencies. He helped develop the national computer keyboard standard and invented integrated user training within computer applications as well as various electronic computer interfaces.

As a computer science and engineering teacher and researcher, he has published 26 research articles from over $2.8 million in funded research projects, plus conference papers and other publications. He taught many areas of computer science and engineering for 37 years (32 years at Texas A&M University), including artificial intelligence, quantum computing, programming and software design, and cyber-ethics.

At Harvard he received the Bowdoin Prize and medal for writing, and in 2015 was named a Distinguished Alumnus of Oklahoma Christian University. He is a life member of the Association for Computing Machinery and American MENSA.