IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF ARIZONA

Kari Lake, *et al.*,	)	
Plaintiffs,	)	
v.	)	No. 2:22-cv-00677-JJT
Katie Hobbs, Arizona Secretary of State, *et al.*,	)	
Defendants.	)	

Declaration of Douglas Logan

Pursuant to 28 U.S.C §1746, I, Douglas Logan, make the following declaration.

1. I am over the age of 21 years, and I am under no legal disability which would prevent me from giving this declaration.
2. I reside in Sarasota County, Florida.
3. I have over 18 years of experience generically within the Information Technology discipline, and over 10 years of experience specifically focused in Cyber Security in the area of Application Security.
4. I have held the certifications, Certified Information Systems Security Professional (CISSP), GIAC Certified Incident Handler (GCIH), and GIAC Web Application Penetration Tester (GWAPT).
5. I have developed cyber security programs and lead cyber security related services for the federal government, as well as Fortune 500 companies including JPMorgan Chase and Bank of America in areas including ethical hacking, malicious code detection, code review, and threat modeling.

6. I have authored training material in the areas of Web Application Penetration Testing, and Mobile Penetration Testing and have taught classes at Fortune 500 companies on these topics, in addition to teaching these topics to hundreds of students in the US Cyber Challenge program.
7. I have served as the Chief Technologist for the US Cyber Challenge, where I was responsible for helping design and manage a program to help identify and grow the next generation of Cyber Security experts for the United States.
8. I have personally overseen or conducted application vulnerability assessments on over 2,000 software applications. These applications represent all major industries, as well as the federal government and include web, mobile, and embedded applications.
9. I have evaluated Dominion voting machine software deployments in Antrim, Michigan, Maricopa County, Arizona, and Georgia. All versions of the software utilized were substantially similar, with no easily discernable visible differences between the versions of software utilized.
10. I was the primary author of the Maricopa County Election Audit report[1] commissioned by the Arizona State Senate, which included voting machine analysis and findings that I personally contributed.
11. I have worked with law enforcement related to election audit matters, including helping present data to a judge to argue for the issuance of a criminal search warrant.
12. I have talked with election workers across the country to determine common problems encountered, and normal practices.

[1] https://www.azsenaterepublicans.com/audit

13. I have worked with state politicians in multiple states to advocate election reform and correct misconceptions about the security of electronic voting machines.

14. My declaration highlights the long history of significant weaknesses in voting machines in the United States that continues through today, and the danger that using this technology poses.

Term Definitions & Background

15. In broad terms, "electronic voting machines", "electronic voting systems", "electronic election equipment" and "electronic election management systems" refer to any computerized devices or equipment utilized to cast, print, count, tabulate, process, and/or store ballot images and/or election results. These are generically referred to within this declaration as "voting machines", the "election management system" or by the vendor's specific name for the device and defined within its context.

16. "Source code" or generically "code" refers to the human discernable instructions written in a programming language by a programmer that, when deployed as a "computer program", "firmware" or "operating system", tells a computerized device, such as an electronic voting machine, how to operate, "think" and process data.

17. A "computer program", or more generically a "program" is a distinct set of computer instructions created from the source code that tells a computerized device how to perform a given function or set of capabilities. In the case of compiled programming languages, the "computer program" is the output of running the source code through a compiler to change the human discernable source code into a format which the computer can more directly understand. That output is what is deployed on the computerized system. In the case of interpretive programming languages, the source code itself is the "computer program" and is deployed to the computerized device and is translated to computer instructions at run-time by another program called an "interpreter". All "computer programs" associated with "electronic voting machines" that I've seen have been from compiled languages, and there is not any "source code" on the devices.

18. "firmware" or an "operating system" is a specialized "computer program" designed to allow a computerized device to turn on, and potentially run other "computer programs".

19. "erroneous code" is "source code" which when run as a "computer program" does not perform what could be reasonably be determined as the expected behavior and intention of the program. This can be caused by a "bug", or due to "malicious code".

20. A "bug" is when a programmer makes one or more unintentional mistakes when writing the "source code" which causes an adverse behavior in the program. This could be because the programmer had a typo, logic error, or otherwise failed to account for a situation that would occur when the program was deployed and running.

21. “malicious code” is “source code” purposely written to cause some adverse behavior by a “malicious actor”. “Malicious code” can be inserted into an otherwise legitimate program to make it a “malicious program” by modifying the underlying “source code” or program; or it can exist as a stand-alone “malicious program” on the computerized device.

22. A “malicious program” is a “computer program” that is created or otherwise contains “malicious code”, and therefore performs some adverse behavior. The “malicious program” can be written to cause the device to perform tasks immediately, or at a conditional time in the future

23. A “malicious program” can perform any action that a computer program on the computerized device has access to do. This includes, but is not limited to, modifying other programs or data, deleting other programs or data, preventing programs or the computerized device from working properly, or exfiltrating data on the device. A “malicious program” can also be written to delete traces it ever ran, including putting things back to the state they were before it ran, or deleting itself.

24. A malicious program can be added to a computerized device directly by a malicious actor who gains temporary access to the device; through a legitimate program that has been made malicious before it was installed on the computerized system; or through another malicious program that accesses the system via a computer network, such as either a local network or the internet, or storage media, such as a USB device.

25. A “secure baseline” is a set of initial configurations for a computerized device or application that defaults to a defined standard to protect the confidentiality, integrity, and availability of the device. These typically includes guidance for implementing all known best practices for the technology including, but not limited to: password complexity, multi-factor authentication, user account management, administrator account management, log level, log management, log retention, patch management, application run levels, and other secure configurations including the disabling of features or functionality that could increase the attack surface. There are a number of defined industry standard baselines that are frequently used including, but not limited to the Center for Internet Security (CIS) Benchmarks, the Defense Information Systems Agency (DISA) Security Technology Implementation Guides (STIG), and Microsoft Security Baselines.

26. There was no indication that any system I reviewed, which were primarily Dominion systems, had implemented any known “secure baseline”.

27. While all aspects of a "secure baseline" are important for ensuring the security of the device, log level, log management and log retention are critical for "incident response". A log is a file that records data about access to or changes made to a computerized system, such as when files were opened, changed, or deleted, and what user accessed the system to view, change, or delete data. Key concepts for log management typically implemented across all the baselines include increasing the default log level so that critical actions are properly logged, configuring logs with larger maximums before they roll-over and auto-delete,[2] and storing logs outside of the computerized device where they are generated so a compromise of that device cannot delete the logs. Failing to do so can result in a situation where the data needed to determine if any type of breach occurred does not exist.

28. None of the systems I reviewed, which were primarily Dominion systems, appeared to implement any of the best practices associated with maintaining logs. As an example, this meant that when files were deleted on a Windows system, there was no log entry generated within the Windows Event Logs; the logs on the system could and were easily rolled over (meaning data from one time period was overwritten and destroyed by data from a later time period); and an individual with "administrative access" to the system had a large window of time to delete the only copy of the logs.

[2] A computer will often limit the amount of data that a log file can store, to prevent the log file from consuming too much of the computer's storage. When the log file reaches the maximum permitted data size, it typically "rolls over" by automatically deleting the oldest data in the file to make room for a new entry.

29. “incident response” is the formal process of investigation that is conducted in response to a breach or a suspected breach of a computerized device or series of computerized devices. It involves evaluating all of the available data to determine if a breach actually occurred; how the breach occurred; what happened during the breach; and when possible, who conducted the breach. Proper incident response is only possible when the appropriate data exists. Without the appropriate data to review, most notably log files, it can be impossible to determine conclusively if a breach even occurred.

30. “Cybersecurity” is the practice of ensuring the confidentiality, availability, and integrity of computerized devices and the data that resides on them. This includes preventing changes being made to the computer programs or data on a device by any person who is not authorized to made changes by the owner of the device, and to detect/remediate any unauthorized changes that are made.

31. “Application Security” is the area of Cybersecurity that specifically addresses “source code” and “computer programs”.

32. “Hacking” is the process by which the misconfiguration of a “computerized device” or “erroneous code” on the device is exploited to cause some adverse behavior that impacts the confidentiality, integrity or availability of the computerized device. This can be performed ethically with the permission of the computerized device owner, or by a malicious actor with malicious intent.

33. “Administrative access” to a computer or computer system means a person knows the necessary passwords to access critical functions of the software and make authorized changes to the system or software. “Administrative access” gives a person control over the computer or system without needing to hack it to do so.

34. Any person who gains sufficient access to add or update a program on electronic equipment that is part of an electronic voting system, has the ability to control the behavior of that equipment – such as the ability to cause the equipment to change, delete, or fabricate votes.

35. Malicious actors who wish to control the outcome of an election without regard to the actual votes cast by voters can create and save to the memory or storage of an electronic device, a malicious program that instructs the device to report that a particular candidate received a majority of the votes, or to report that votes cast for one candidate were instead votes cast for another candidate.

36. Malicious programs can be configured to be extremely subtle choosing not to alter all votes, or to only alter votes from specific precincts, on specific times or on specific days. They could even be configured to only be triggered after a certain type of ballot comes through, or a certain set of ballots in sequence. These types of subtle triggers would be impossible to detect in a Logic and Accuracy test.

37. To prevent electronic devices from manipulating votes, the devices must be absolutely secured against the introduction of any malicious programs, including programs from the voting machine vendor that could potentially be built on “malicious code” and therefore are now “malicious program(s)”.

38. This means that voting machine programs must go through the proper cybersecurity testing and computerized devices must be configured to Cybersecurity best practices so that access is controlled, systems are up-to-date with the latest patched versions of computer programs, and all actions on the system are properly logged so actions on the computerized device can be validated.

39. Proper cybersecurity testing includes all voting machine programs going through regular, independent 3rd party source code review as well as ethical hacking or penetration testing engagements by qualified personnel where real-world deployments are tested for exploitable security vulnerabilities. Current vulnerabilities found across voting machine vendors, as can be seen through evidence referenced in this declaration, heavily suggest that this practice is not happening.

40. Even in a well-designed and properly secured computer system, the factor of human error can lead to cybersecurity breaches.

41. Ultimately it is individual employees or officials who must choose secure passwords, keep their passwords secret, refrain from activating malware by opening email attachments or clicking on unsafe internet links, refrain from connecting computer hardware to portable computer memory media or computer networks, maintain software up-to-date, and a host of other mundane cybersecurity practices – including remembering what cybersecurity practices must be observed.

42. Experience has shown that humans err on these practices, through ignorance, forgetfulness, neglect, and even intention, simply because it is less demanding to ignore the proper procedure.

Context: Cyber Security Conference Shows Vulnerable Voting Machines

43. Since 2017, a Cyber Security conference, "DEFCON", has hosted a "Voting Machine Hacking Village" where voting machines from major vendors have been displayed and made available for participants to attempt to compromise. Conference participants had a maximum of 2.5 calendar days to work with these machines, mixed in with the rest of their conference activities. A report issued by conference organizers after the 2017 conference stated, "By the end of the conference, every piece of equipment in the Voting Village was effectively breached in some manner. Participants with little prior knowledge and only limited tools and resources were quite capable of undermining the confidentiality, integrity, and availability of these systems." A copy of the DEFCON 2017 report is attached as Exhibit A.

44. The statement "undermining the confidentiality, integrity, and availability of these systems" means the participants found methods by which votes could potentially be changed on such systems.

45. The DEFCON Report from the Voting Machine Hacking Village from 2018 shows that significant vulnerabilities remained, and that vulnerabilities that had been reported prior had not been fixed. The summary of the 2018 report stated,

- A voting tabulator that is currently used in 23 states is vulnerable to be remotely hacked via a network attack. Because the device in question is a high-speed unit designed to process a high volume of ballots for an entire county, **hacking just one of these machines could enable an attacker to flip the Electoral College and determine the outcome of a presidential election.**
- A second critical **vulnerability in the same machine was disclosed to the vendor a decade ago**, yet that machine, which was used into 2016, still contains the flaw.
- Another machine used in 18 states was able to be hacked in only two minutes, while it takes the average voter six minutes to vote. **This indicates one could realistically hack a voting machine in the polling place on Election Day within the time it takes to vote.**
- Hackers had the ability to **wirelessly reprogram, via mobile phone, a type of electronic card used by millions of Americans to activate the voting terminal to cast their ballots**. This vulnerability could be exploited to take over the voting machine on which they vote and cast as many votes as the voter wanted.

A copy of the DEFCON 2018 report is attached as Exhibit B.

46. The DEFCON 2019 report repeats similar high-level bullets, showing no noticeable improvements from prior years.:

- Commercially available voting system hardware used in the U.S. remains vulnerable to attack.
- There is an urgent need for paper ballots and risk-limiting audits.
- New Ballot Marking Device (BMD) products are vulnerable.

A copy of the DEFCON 2019 report is attached as Exhibit C.

47. All three reports highlight that there are a variety of ways that voting machines are compromised, even new systems are vulnerable when built, individuals with limited or no experience were able to compromise these systems, and those individuals were able to find these vulnerabilities in three days or less. This suggests that voting machine vendors are not undergoing the same standard security testing that is common and normal in the United States, for example, in the financial services industry.

Antrim, MI: Dominion Configuration Allocates Votes to the Wrong Candidate

48. Antrim County, Michigan initially misreported its vote totals for the 2020 presidential election. Antrim County later issued a revised count that was dramatically different, with the vote total for one candidate increasing significantly and the total for another candidate decreasing significantly.

49. In early 2021, the Michigan Secretary of State engaged J. Alex Halderman, a professor of computer science at the University of Michigan, to provide a technical explanation of what happened in Antrim. Professor Halderman's report[3] outlines a number of failures in the way the election was conducted, but specifically highlights some very alarming problems with the way that Antrim's electronic Election Management System, supplied by Dominion Voting Systems, tallies results.

50. Halderman stated that the Dominion software used numerical values to identify each candidate on the tabulators, and also used numerical values to identify the candidates on the Election Management System (EMS) Server where vote totals are aggregated for the official results. These numbers are supposed to match so that when the tabulator counts the votes for Candidate A, those results are attributed to Candidate A on the EMS Server. Halderman stated that in a number of precincts in Antrim County, MI the numbers had become misaligned due to improper device configurations. As a result, candidate Jorgenson's votes in those precincts were initially attributed to candidate Trump, Trump's votes attributed to candidate Biden, and Biden's votes marked as undervotes.

[3] https://web.archive.org/web/20210327143045/https://content.govdelivery.com/attachments/MISOS/2021/03/26/file_attachments/1736734/Antrim.pdf

51. Halderman attributed this issue to "user error," since the county clerk employee creating the configuration files for the tabulators did not follow Dominion's recommended procedure for implementing ballot configuration changes. Specifically, after making a small ballot change in one precinct, the county employee remade the tabulator configuration files for that one precinct, but did not remake the configuration files for every other tabulator in every other precinct. The county employee presumptively assumed that since nothing changed on the ballots associated with those precincts, there was no reason to update the configuration files for the tabulator. However, Halderman's analysis was that this ballot change created a mismatch in any precinct that showed up in the Dominion software after the precinct that had the ballot change, so that votes for each candidate were misattributed. Halderman viewed this as a "human error" while acknowledging, "The election software also could have done more to help election staff avoid making mistakes that could lead to erroneous results." [4]

52. It is my professional opinion that Halderman's statement that more could have been done by the election software is a gross understatement. As a comparison, I typically see more validation implemented in commercial inventory control software. The primary purpose of election software is to make sure votes for a candidate are properly counted and attributed to the intended candidate. Failing to do any type of validation on the importing of results to be sure that the intended candidate receives the proper tallies at best constitutes gross negligence.

[4] J. Alex Halderman, Analysis of the Antrim County, Michigan November 2020 Election Incident at 3 (March 26, 2021) available at https://web.archive.org/web/20210327143045/https://content.govdelivery.com/attachments/MISOS/2021/03/26/file_attachments/1736734/Antrim.pdf

53. Subsequent analysis of the Dominion system has shown how exploitable the design flaw identified by Halderman is. Working with Dominion equipment, expert Jeff Lenberg was able to alter configuration files for a tabulator or settings on the EMS to intentionally cause the system to wrongly attribute votes to a favored candidate. In his testing Lenberg further caused the poll tape to output the manipulated results, thereby leaving no clear indication anything had been tampered with.

54. Halderman's and Lenberg's work show that to alter election results in the Dominion system, all that would be required is a small change on the media used in the tabulator. This could be accomplished through a malicious program, or by a person with temporary access to these cards. The only evidence that tampering had occurred would potentially be a copy of the media used in the tabulator or a full hand count of the ballots in an affected batch.

55. Many counties reuse the media utilized for configuring the tabulator from one election to the next. Such reuse means that at the time of the next election the evidence of previous tampering would be destroyed.

56. In Antrim County, MI this problem was identified because the reported results were so significantly different from the expected results based on historical trends, prompting an investigation. It is highly likely that this problem would go undetected in a county where the normal margins between candidates was less predictable; and it is in fact very likely this *has* happened elsewhere and gone undetected. The assumption that the county employee made in only creating the configuration files for the tabulator where a change happened is a reasonable assumption to make. It is unlikely this is the first time someone made this mistake, and yet it is the first time a mistake was publicly detected.

Voting Machine Software Fails to Implement Best Practices

57. I was given access to the Antrim County Dominion election equipment in connection with post-election litigation for which I provided expert witness services. After analyzing the equipment, I concluded it exhibited a large number of failures in implementing secure coding practices, application security design principles, and cyber security best practices. Specifically:

   a. The Dominion Election Management System software did not follow application security best practices known for over 15 years, and widely adopted across industries.

   b. Operating System credentials used generic accounts that multiple people knew and did not change from year to year. These credentials provided more than enough access and the tooling for any of the individuals who possessed them to change election results.

   c. The systems had not been configured to any type of secure baseline. Lack of a secure baseline increases the chances the systems could be compromised and also means that the appropriate logs required to respond to a suspected breach would not exist on the system, making it nearly impossible to determine if election results were tampered with just by looking at the election equipment.

d. The system had Microsoft SQL Server Management Studio installed, which allows the direct editing of data in the election results database. This software is installed separately from Microsoft SQL Server and is not listed on the Election Assistance Commission's list of approved software. It was unclear what business purpose it could serve on this particular system. The Dominion software already has capabilities within it that could handle any normal expected use case for Microsoft SQL Server Management Studio.

58. These failings are indicative of an immature cyber security program that has not implemented the necessary steps to ensure secure software is created and cannot ensure the integrity of the votes tallied on the systems.

Maricopa County, AZ: Failure to Follow Best Practices & Failure to Preserve Logs and Data

59. The audit of the 2020 election results in Maricopa County, Arizona was the largest and most comprehensive election audit conducted to date. In addition to reviewing the actual voting machine systems used during the election, it involved a full hand-counting of all 2.1 million ballots, and forensic imaging and a forensic paper analysis of all the ballots. I served as the project lead who ran this audit.

60. Maricopa County used electronic election equipment supplied and supported by Dominion Voting Systems.

61. The audit's findings related to the Dominion electronic equipment used to case and tabulate ballots in the election included the following:

a. Operating System credentials used generic accounts that multiple people knew and did not change from year to year. These credentials provided more than enough access and the tooling for any of the individuals who possessed them to change election results.

b. The systems had not been configured to any type of secure baseline. Lack of a secure baseline increases the chances the systems could be compromised and also means that the appropriate logs required to respond to a suspected breach would not exist on the system, making it nearly impossible to determine if election results were tampered with just by looking at the election equipment.

c. Computer logs related to the equipment's performance during the election were lost, and files were deleted. In most cases it was not possible to determine who performed the actions because the required information to do so did not exist.

d. Information was withheld by the County and never provided to allow an analysis of this data. This included routers and log data, in addition to other requested election equipment.

62. While the Splunk logs and router data were later reviewed by Special Master John Shadegg's team, the scope of their review did not cover the time period where the audit detected files deleted, nor did it appear the County stored the necessary net flow data required to make any determination related to internet connectivity. This did not stop their report from stating there was no indication of deletion of files or internet connections. All participants besides Shadegg were prevented from saying anything about the analysis done due to a non-disclosure agreement.

63. The events in Maricopa County illustrate the unsecure practices surrounding electronic voting machines and voting equipment that are standard in Arizona and across the country that could result in the purposeful or accidental altering of election results. In Arizona, as in other states, after-the-fact review found that generic user accounts were used in the electronic election equipment, poor passwords were implemented, and there was insufficient logging to tell what actions had been done and by whom. Specifically in Arizona, every single Windows account on every system analyzed had the exact same password, including for users who could alter results. Election data had been purged and files deleted, and it was not easily attributable to any individual. This setup would make it difficult to detect, let alone respond to, any type of election result problems or mistakes.

64. While this and other aspects of the audit have been claimed by media to have been refuted, the rebuttal of Maricopa County's response has never been fully addressed. A copy of Maricopa County's response to the audit is attached as Exhibit D. The rebuttal to this response can be found as Exhibit E.

Williamson County, TN: Insufficient Technical Oversight Over Voting Machine Code & Changes

65. On November 3, 2021, the federal Election Assistance Commission (EAC) received a report from the Tennessee Secretary of State related to an anomaly from the October 26, 2021 municipal elections in Williamson County, Tennessee. The Tennessee report stated that the close poll reports from 7 of the 18 ICP tabulators used during the election did not match the number of ballots scanned. The election results were validated via a hand-count of ballots, but the EAC launched an investigation into what had caused the tabulating problem, and later issued a report related to the investigation.[5]

66. The equipment used in Williamson County, Tennessee, and investigated by the EAC was part of the D-Suite 5.5-B system supplied by Dominion Voting Systems.

67. The EAC report stated that the first step in the investigation was to validate that the proper version of all files was included on the system. This same process is typically part of the pre-election certification of the machines for use in an election. However, even though these systems had presumptively gone through the certification process, the EAC accredited voting system vendors Pro V&V and SLI Compliance hired to perform the analysis determined that two configuration files used by Williamson County were not the correct versions and were from a prior version. This was later determined to not be significant for the cause of the issue, but shows that the certification processes missed identifying differences that it should have found.

[5] https://www.eac.gov/sites/default/files/voting_system/files/EAC_Report_of_Investigation_Dominion_DSuite_5.5_B.pdf

68. Pro V&V and SLI Compliance were able to reproduce the erroneous results and found that the anomaly corresponded with either the error in the log, "QR code signature mismatch" or "Ballot format or id is unrecognizable."

69. Despite the work being conducted by two EAC accredited vendors, Pro V&V and SLI Compliance, and with representatives present from Dominion, the EAC, the Tennessee Secretary of State, and Williamson County staff, they were unable to determine the cause of this erroneous behavior.

70. The manufacturer, Dominion, submitted a Root Cause Analysis to the EAC stating "erroneous code is present in the EAC certified D-Suite 5.5-B and D-Suite 5.5-C systems." Dominion stated that when a certain part of a QR code was misread, the ICP interpreted the ballot as provisional and thereafter marked **all** ballots subsequently scanned as provisional, leaving these ballots out of the close poll report totals. Dominion's solution was to submit revised code that would reset the provisional flag within the tabulator after a ballot was scanned as provisional, so that subsequent ballots would not automatically be flagged as provisional.

71. A QR code is a defined standard for matrix barcodes which is capable of storing a significant amount of data within a small amount of space. This standard is very resilient and has error correction built into the standard[6]. Depending on the level of error correction implemented, a QR code can have from 7% to 30% of the QR code fully obstructed or damaged and the QR code can still be read properly. This is because a QR code essentially holds the data in it twice. It is in the QR code once in its normal format, and one other time as a polynomial representation of the data[7]. This error correction is robust enough that even at 7% error correction a QR code should fail to be read and error out rather than ever have values within it misread.

72. Dominion utilizes QR codes on ballots printed out of their ICX Ballot Marking Device (BMD) to represent all of the choices the voter made. In some states such as Georgia and Tennessee these ICX BMDs are utilized by all voters. In other states most voters will fill out a standard bubble fill ballot by hand and the ICX BMD will only be utilized for accessibility voters. In all cases the filled out ballot, whether by hand or by the BMD, will subsequently be fed through the ICP tabulator in order to be official cast and tallied.

[6] https://www.qrcode.com/en/about/error_correction.html

[7] https://blog.qrstuff.com/2011/12/14/qr-code-error-correction

73. The QR codes utilized by Dominion utilizes Level M of error correction, which offers 15% of redundancy. This can be seen in the photo to the right, which is a Dominion QR code from a Georgia ballot. The hockey puck like black box that has been highlighted by a red box is the indicator that this level of error correction is in the QR code[8].

74. At 15% error correction it would be extremely unlikely for a QR code to have a value in it misread. Credit cards utilize a single digit in the credit card number as a checksum to make sure the credit card was read properly. This is simple, yet sufficient to make sure when you swipe your credit card it either errors out and needs to be reswiped, or your account is charged and not someone else's. The level of validation in a QR code is significantly more complicated and sophisticated. Dominion's ICP tabulators are also fully capable of spitting a ballot back out and forcing the user to refeed the ballot in the case where something is not read correctly. As a result, Dominion's explanation that the QR code was misread is insufficient to adequately explain what occurred.

75. Even taking that into the consideration, per the description of the Engineering Change Order, Dominion's code change did not do anything to fix the situation which caused a ballot to be misread. The revised code simply reset the provisional flag for subsequent ballots so the error code would not impact everything afterwards

[8] https://blog.qrstuff.com/2011/12/14/qr-code-error-correction

76. The EAC report states, “Modified ICP source code was submitted by Dominion that resets the provisional flag following each voting session. The ECO analysis included source code review to confirm the change to both systems and to ensure no other code is changed.” This appears to mean that the original ballot that was misread would still be misread and may be erroneously flagged as provisional, and not counted.

77. Yet the EAC report clearly states that testing after the change was implemented showed that the fix resulted in all ballots being properly counted and no errors in the log: “This software was then tested for accuracy by processing two thousand ballots printed by an ICX, utilizing the same election definition used in Williamson County, TN on October 26, 2021. The analysis and testing of the ECOs has demonstrated that the anomaly was successfully fixed. No instance of the anomaly or the associated error or warning messages in the ICP audit logs were observed during the testing.” This either means that ballots that were utilized to test the fix never would have triggered the error condition in the first place, or it means that the misreading of a ballot did not impact the ballot itself but only impacted subsequent ballots.

78. If ballots were utilized to test the fix that never would have triggered the error condition, it means that it is unknown if this issue is actually fixed. More importantly, it suggests either gross technical incompetence by whoever performed the testing in not realizing that the test case did not validate the fix; or a knowing and willful misrepresentation to the EAC that the code changes were validated. In either case, by accepting the fixes from Dominion as resolving the issue, the EAC demonstrated a lack of technical competence to properly oversee the necessary actions needed to ensure the integrity of electronic voting machines.

79. If ballots were in fact used in the test case to validate the fix that would have created this error condition, but all the results came back properly and validated that the code change did fix the issue, this would indicate that misreading of a QR code ONLY impacted subsequent ballots and not the current ballot being processed. This would be an extremely abnormal programming error to accidentally occur and highly suggests intentional malicious code. The source code needs to be thoroughly reviewed by qualified personnel to determine the origins of this erroneous code and the reason it was put in place.

80. In summary, Dominion machines in Williamson County, TN failed to properly tally votes in a significant way. This ended up in an EAC investigation where they stated that "erroneous code" was in the Dominion software that had caused that issue. This same code has been utilized in elections across the country for quite some time with an unknown impact. The EAC accepted an explanation from the voting machine Dominion that doesn't make technical sense and let the voting machine vendor define the root cause and create the code to fix the issue. This code was either insufficiently tested to validate the issue was fixed, or the original code was a strong indication of a subversion of the voting equipment. In either case, the EAC demonstrated inadequate oversight to properly ensure the integrity of votes tallied in electronic voting machines.

The Insufficiency of Air Gaps

81. It is commonly argued that deficient cyber security practices concerning electronic election equipment are insignificant because these systems are "air-gapped" and not connected to any type of external computer networks.

82. There is substantial evidence that many election systems are not actually protected by "air-gapping." In Maricopa County, the analysis of the equipment revealed significant indications that the Election Management network had connections to the internet at times.· Both Dominion[9] and ES&S Software[10] advertise that their systems support modems of various types and modem support is part of Dominion's contract for Michigan.[11]

83. Even a properly "air-gapped" system is not safe from malicious code being copied into it. For example, computers in nuclear reactors are air-gapped to increase their security, yet the Stuxnet malicious code infected computers in a nuclear reactor in Iran, ultimately causing damage to the reactor. To do so, the Stuxnet code spread through computers in Iran until it copied itself to a system connected to a portable USB drive that was later also used on the nuclear reactor computer network. The Stuxnet code copied itself onto the USB drive and then onto the nuclear reactor network computers. Inside the air-gapped network, the Stuxnet code slowed down the nuclear reactor without causing a nuclear meltdown.

84. The same or a similar technique could be used to compromise "air-gapped" electronic voting machines, and in fact, electronic voting systems routinely utilize USB Drives and other portable media to transfer results, configuration files, and ballot images from one system to the next

[9] https://www.michigan.gov/-/media/Project/Websites/sos/01holland/071B7700117_Dominion.pdf

[10] https://web.archive.org/web/20150920041547/https://sos.idaho.gov/elect/Clerk/DS200%20Procedures/ESS_EVS5000_SOP00_DS200_Operator%20Guide.pdf

[11] https://www.michigan.gov/-/media/Project/Websites/sos/01holland/071B7700117_Dominion.pdf

Conclusions

85. There is an extreme lack of cyber security maturity in the election software and equipment industry, which has not implemented best practices that have been known for more than 20 years and have been adopted and applied in most major industries in the United States.

86. The repetitive nature of severe vulnerabilities detected across multiple voting machine vendors over many years indicates that proper cybersecurity testing, such as secure code review and penetration testing are not being performed on a regular basis by competent personnel, and/or what is detected is not being remediated. Without proper cybersecurity testing and the prompt remediation of what is detected it is impossible for voting machine software to be secure.

87. The problem is further complicated by the fact that many election workers operating election systems are older[12] and have inadequate technical knowledge, forcing them to fully rely on the voting machine vendor or their subcontractors to perform the most basic tasks.

   a. Maricopa County has stated that it employs two Dominion individuals full-time in order to run its elections. These individuals do everything from configuring the elections to making backups and taking them offsite.

   b. Michigan uses a company called Election Source to build its elections. Local county officials have little understanding of the machines they use.

[12] https://www.eac.gov/documents/2017/11/15/eavs-deep-dive-poll-workers-and-polling-places

c. In Antrim County, the county clerk was quick to state the county had made a mistake and the problem was human error, even though she could not give any details as to why but merely relied on the vendor's analysis.[13] The technical explanation eventually provided by the Secretary of State, written by Professor Halderman, at best indicated extremely poorly written software.

d. The Williamson County, TN EAC report makes it very clear that the EAC lacks the needed technical knowledge to fully monitor and validate that identified issues are properly remediated by voting machine vendor.

88. Because election abnormalities directly affect partisan political interests, investigations related to electronic election equipment, in my experience, rapidly become heated and toxic. This makes it harder to conduct an adequate investigation, because one side has incentive to keep the investigation from finding anything. The complexity of the electronic equipment and software gives interested parties ample opportunity to obfuscate the investigation.

89. Without an understanding on how computer systems and election systems work or how to detect unauthorized modification of code on a system, elected officials, law enforcement officials, and court officials are not well-equipped to deal with controversies related to electronic election equipment.

[13] https://www.youtube.com/watch?v=S9uMlrnAVwo

90. The end result is that current electronic voting systems (1) are unsecure and can be compromised by individuals with no experience with voting equipment with less than three days of work, (2) are built and deployed without the proper safeguards for accountability to allow a proper investigation to happen, (3) are difficult to audit after the fact because they are too complicated for elected officials, law enforcement, and the courts to easily understand.

91. It is my professional opinion, for all of the reasons outlined, it is impossible for electronic voting machines to be properly secured by the 2022 elections, and that they should not be used. Something so essential to democracy should not be more complicated than the average American, law enforcement, and the courts can easily understand and handle in a timely manner.

I declare under penalty of perjury that the foregoing is true and correct.

Executed on June 7, 2022.

Douglas J Logan