

Rebuttal of Sequoia Voting Systems' Response to the UCSB Red Team Report

Computer Security Group
Department of Computer Science
University of California, Santa Barbara
seclab@cs.ucsb.edu

July 31, 2007

On July 27th, the California Secretary of State made public a report that presents a security analysis of the Sequoia voting system as performed by the Security Group of UC Santa Barbara. The Security Group was led by Giovanni Vigna and Richard Kemmerer and included Davide Balzarotti, Greg Banks, Marco Cova, Viktoria Felmetzger, William Robertson, and Fredrik Valeur.

The Security Group acted as a “red team” and performed a series of security tests of both the hardware and the software that comprise the Sequoia voting system to identify possible security problems that could lead to a compromise. A “compromise” is defined as “tampering or error that could cause incorrect recording, tabulation, tallying or reporting of votes or that could alter critical election data such as election definition or system audit data.” [6]

Our testing identified a number of security issues that are of great concern. In our tests, we were able to bypass both the *physical* and the *software* security protections of the Sequoia system. As a result, we were able to disrupt normal voting processes and modify the results of an election.

After the results of our evaluation were published, Sequoia published a response [8]. In the following, we respond to several Sequoia statements that we believe require clarification or are incomplete and incorrect.

1 A Realistic Evaluation

Sequoia’s response states that the evaluation was “an unrealistic worst case scenario,” and, therefore, its findings do not “measure the severity of the actual threats in any meaningful way.” We do not agree with this conclusion.

1.1 System Setup

The system we evaluated was set up *by Sequoia’s employees*. They were requested to set up the system as if it would have been used in a real election. During the setup, which included a complete demo including security procedures recommended by Sequoia, we asked several times if the configuration they set up was a realistic one and received an affirmative answer each time. Contrary to what was stated in Sequoia’s

response, the equipment tested was taken through the prescribed pre-election logic and accuracy testing (pre-LAT) and preparation.

The hardware and the software used in the setup were provided, installed, configured, and verified by Sequoia’s employees. All the components certified for use in California elections were set up and configured. To make the election process as realistic as possible, the ballot information was based on data from a previous, real election.

In addition, when designing and developing the attacks that we presented in our report, we took into account technical protective measures, such as tamper-evident seals, the voter verified paper audit trail (VVPAT), and cryptographic techniques, in addition to the security procedures recommended by Sequoia in their documentation. Our report also explained why these measures do not prevent our attacks (see, for example, Sections 4.9, 4.11, and 4.12).

1.2 Lack of “Blue Team”

In its response to our report, Sequoia claims that red team testing without a corresponding “blue team” is unrealistic. It is, however, not common practice to utilize a “blue team” during red team or penetration testing. Two common standards exist for penetration testing: the Open Source Security Testing Methodology Manual (OSSTMM) [4] and NIST Special Publication 800-42 [10]. The OSSTMM lists six different kinds of penetration test types, none of which involve a “blue team”. The NIST publication does not mention a “blue team” either.

It is our contention that the inclusion of a “blue team” whose sole purpose is to thwart any of the red team’s attacks creates unrealistic test conditions. The “blue team” would be aware that a security test is in progress and of the identities of the red team. The “blue team” would, therefore, be suspicious of any action performed by a red team member, which in a normal situation would not be considered suspicious. This is why “blue teams” are not traditionally used during penetration testing.

The only way to create a completely realistic environment for a penetration test of a voting system is to perform the testing during an actual election without the knowledge of poll-workers, election officials, or voters. This is of course not feasible, since the test could void the official election.

Therefore, this effort represented a security evaluation in a realistic environment, which was set up by the vendor itself. The system was not tested “in isolation”; instead, the complete set of hardware and software components were tested in an integrated way.

2 Access to Hardware and Software

Sequoia’s response states the following regarding our evaluation:

“This was not a security risk evaluation but an unrealistic worst case scenario evaluation limited to malicious tests, studies and analysis performed in a laboratory environment by computer security experts with unfettered access to the machines and software over several weeks. This is not a real-world scenario [...]”

Unfortunately, this statement is misleading for several reasons. First, we presented a number of attacks

that required different levels of knowledge and access to hardware components: single voters, poll workers, and “sleepover” custodians. Not all the attacks and scenarios we discussed require “unfettered access”. In particular, even though we were given access to the source code, none of our attacks require the source code in order to be planned and successfully executed.

Secondly, it is important to distinguish between the access and knowledge level required to *design* an attack and the level required to *perform* it. The design of many of the attacks we presented does require an in-depth knowledge of the internals of the tested machines. However, the execution of the attacks can be performed with minimal or no knowledge of the machines. In other words, one security failure may be sufficient to design and develop attacks that can be performed in an election day scenario.

Finally, history has demonstrated that it is possible for outsiders to acquire “unfettered access” to Sequoia hardware components and their firmware. In fact, there have been cases where these systems have been regularly bought online, outright stolen, or otherwise accessed by outsiders. Though each of these approaches requires some financial outlay, an attacker who has the goal of modifying the results of an election will not be stopped by the relatively small sums required.

For example, in 2007, Prof. Andrew W. Appel from Princeton University purchased five Sequoia AVC Advantage machines (a model not included in the current review) from an Internet auction site and examined their internals, including the firmware. He notes “I did not have to present any credentials other than my name, address, e-mail, and telephone number. No other questions were asked of me by govdeals.com [the on-line auction site] or by Buncombe county [the auctioning county]. The government had no information about me or my motives in obtaining the voting machines at any time before or after the auction and delivery of the voting machines to me.” [1].

As another example, in 2003, software used in the Sequoia AVC Edge system was found unprotected on a publicly available server. It was found that “the data [was] unencrypted and unprotected [...] the FTP server allowed anyone to access it anonymously.” [11].

3 Point-by-point Answers

3.1 Testing Mode

Sequoia’s response states:

“In Section 4.3 of the Red Team Report – ‘Accuracy Testing Mode Detection’ – the investigators could determine if a voting machine was in test mode or in Election Day mode. This is not surprising and is true of any system that provides a test mode of any sort.”

This statement is not correct. Two of the other Sequoia voting components that we examined, the Optech Insight and the Optech 400-C optical scanners, in fact do use the same operating mode for both testing and election phases. In this case, it is much more difficult for a malicious firmware to determine whether the machine is currently in testing, when it should exhibit the expected behavior, or deployed in an election, when it should execute an attack.

Sequoia further states:

“This opportunity to attack the system has been anticipated by both the vendor community and governments for many years and is the reason for Parallel Testing as required by the State of California. Parallel testing disables this attack [...]”

Parallel testing is the practice of selecting voting machines at random and testing them as realistically as possible during the period that votes are being cast. While parallel testing could detect malicious firmware during an election, the attack would be far from being “disabled.” In fact, the attack, and the consequent analysis required to determine the cause of the discrepancies found during testing, would still cause significant disruption to the election process. If the cause of discrepancies was found to be a widespread deployment of malicious firmware, it is not clear how to recover the correct election results without performing another election. In particular, a report from the Brennan Center for Justice at NYU concludes that “[beyond impounding a faulting machine], even California does not appear to have a clear protocol in place” [2].

3.2 Use and Configuration of Microsoft SQL Server

With respect to our findings regarding Sequoia’s use of Microsoft SQL Server, their response states:

“Section 4.8 of the Red Team Report – ‘Security of the MS SQL Server’ – points to the need for personnel security by the customer jurisdictions. [...] Persons with access to the central count server should undergo background checks commensurate with the valuable data that they maintain. Windows audit logging must be enabled, the allowable log size maximized, and the logs secured against accidental or intentional alteration or deletion.”

Section 4.8 of our report does not point to the need for personnel security by the customer jurisdictions; rather, it presents a number of security problems that we found in the WinEDS software. While background checks of persons with access to the central count server might help to secure the election process, they certainly are not sufficient to fix the technical problems that we identified. In particular, the only technical solution offered by the Sequoia response (the use of Windows logging and audit facility) is rendered completely ineffective by the second vulnerability described in our report, specifically the ability for a user to execute arbitrary commands on the MS SQL Server host. In fact, an attacker could simply use this vulnerability to disable, erase, or otherwise manipulate system logs in order to conceal his or her actions on the system. Furthermore, procedural countermeasures, such as the recommended use of background checks, might also be ineffective, since attacks may be launched without the direct knowledge or participation of persons with access to the central count server. As an example, this could be accomplished through the use of forged memory cartridges (see, for example, Section 4.7, 4.13, 5.1).

3.3 Physical Security

Sequoia's response to the findings of the red team regarding bypassing tamper-evident seals misconstrues the nature of the vulnerability. The response states:

"While seals can be removed, as is their intended use, they cannot be removed undetectably. [...] Tamper evident seals have been used in military environments for many decades [...]"

First, that the tamper-evident seals could be removed without detection by election officials was not an assertion made by the red team. To the contrary, the red team was able to access critical voting machine hardware (e.g., results cartridge ports, polls open/closed buttons) that are intended to be protected by tamper-evident seals without disturbing the seals themselves; that is, the seals can be bypassed without any physical evidence that an intrusion has occurred. Moreover, the ability to bypass the seals is due not to their design, but is rather a consequence of the physical construction of the voting machine enclosures themselves.

Secondly, regardless of the effectiveness of the attack itself, the reference to the use of tamper-evident seals in military environments should not be construed as a testament to their efficacy. In fact, there exists a large body of work that demonstrates the ease in which many commercially-available tamper-evident seals, including those provided as real-world examples to the red team, can be bypassed without physical evidence. [5]

Sequoia's response also speculates as to the usage of tamper-evident seals:

"[Tamper-evident seals] consist of adhesive tapes with unique identifiers, which can not be removed without breaking them. They could be placed on every access point, including access covers and chassis screws and a record kept of the numbers. Jurisdiction procedures would log that the unique identifiers on the tamper evident seals match established records to ensure that no equipment tampering had occurred."

The seals recommended by Sequoia's documentation and provided by their representatives were not of the adhesive tape type described in their response. Instead, the seals we examined were of two types, either a) a small piece of plastic with a snap closure that, in theory, cannot easily be removed once in place, and b) a small metal block and stranded wire cable that can be inserted into a channel within the block and subsequently is, again theoretically, not easily removed. At no time was the red team made aware of any adhesive tape-based seals for use with the Sequoia voting machine hardware. As a side note, both types of seals were found during testing to be bypassable themselves without physical evidence to that effect.

Furthermore, while applying adhesive tape-based seals to access covers and chassis screws as described in the following sentences would, in principle, comprise an improvement to the current state of affairs, to our knowledge no such procedure is recommended by Sequoia. Also, the red team cannot endorse such a mitigation strategy without further information, as implementation details directly affect the security of such a strategy. As an example, many tamper-evident seals are readily available for purchase, which could potentially allow attackers to remove existing seals, perform an attack, and replace the seals with identical copies. Furthermore, a critical part of the voting machine has a "fuzzy cloth" lining that would prevent an adhesive tape seal from properly sticking to the machine.

3.4 Forged Update Cards and Voter Cards

In response to the red team’s findings regarding the forging of update cards and voter cards, Sequoia states:

“[Forging of update cards and voter cards] is mitigated through physically securing the voting machines, election specific information on the voter card, and traditional and current poll-worker training. This scenario requires that attackers gain access to the voting machines and could successfully extract and utilize information regarding voter card programming. Not only this static information needs to be extracted, but the ballot style for a particular precinct would need to be known to the attacker in advance.”

There statement is misleading in several ways. First, it does not address the issue of forged firmware update cartridges, but rather conflates the issue of forged firmware update cartridges with that of forged voter cards. To summarize the former, an attacker can load a malicious firmware on to an AVC Edge in maintenance mode using a forged cartridge that bypasses both file integrity checks and password authentication procedures. At no time is election-specific information or a poll-worker involved in this scenario, and while “physically securing the voting machines” could be considered an effective mitigation of this vulnerability, in practice voting machines are often left exposed to tampering between elections.

Secondly, the assertion that an attacker would need to “extract and utilize information regarding voter card programming” and possess advance knowledge of “the ballot style for a particular precinct” is inaccurate. In reality, this attack can be performed through the collusion of two or more registered voters during the election. As suggested in Section 5.6 of the red team’s report, one plausible scenario is:

1. The first malicious voter receives a voter card from a poll-worker, and surreptitiously copies its contents using a small, concealed smart card reader.
2. The malicious voter casts a ballot using the original voter card, returns the card, and leaves the polling station.
3. The malicious voter then uses his copy of the original voter card to construct many legitimate voter cards using the forged voter card vulnerability. This is possible due to the fact that smart cards used as voter cards in the Sequoia voting system infrastructure are easily acquired through various channels.
4. Subsequent malicious voters enter the polling station with a number of valid forged voter cards concealed on their person. Though they receive only one voter card from a poll-worker, they are able to cast multiple ballots using the forged voter cards.

Sequoia further states:

“Poll-workers are responsible for ensuring that only voters that have just received voter cards from them approach the machines. It is unreasonable to believe that a person or persons could approach the line of voting machines in a precinct without having been credentialed, and especially that an attacker or group of attackers could do so repeatedly.”

In the attack scenario outlined above, no attacker enters the polling station more than once, and all attackers approaching the voting machines have been authenticated by a poll-worker. Regardless, Sequoia’s assertion that “it is unreasonable to believe that a person or persons could approach the line of voting machines

in a precinct without having been credentialed” is directly contradicted by numerous real-world examples of the potential for just such an attack (e.g., polling stations located in crowded shopping malls [9]).

3.5 Other Considerations

In response to the red team’s attack scenario 1, Sequoia states:

“Attack scenario 1 (insert a malicious HAAT USB stick into the initialization process) relies on two assumptions: that there is a pool of HAAT USB sticks for initialization such that a malicious HAAT USB stick could be inserted into that pool; and autorun on the WinEDS computer is allowed. HAAT USB sticks are specific to each precinct or polling location, thus it would be extremely unlikely that a malicious USB stick could be inserted into the jurisdiction’s HAAT initialization process.”

Under the assumption that one of the insiders has been corrupted, there are many ways in which malicious code can be introduced into the system. USB sticks are only one example. Any removable device can be used for that purpose, including the results cartridges.

“As stated above, autorun features should be disabled on all computers performing election related tasks.

The autorun feature was enabled *on the machine provided to us by the vendor*, and, in the documentation provided to us by Sequoia, we did not find any advisory saying that this feature should be turned off. Furthermore, using this feature to load the malicious software is again just an example. The insider can simply double click on the program icon, operations that require only a couple of seconds.

“Likewise, the assumption that a large number of voters do not check their vote on the paper record, when it scrolls in front of them (providing both visual and audible cues as to its existence) and when the voter is forced to interact with the voting machine to produce the record, is also false.”

The risk that people do not check the paper trail is supported by at least two studies. In [3], the author reports that over 60% of the voters she tested did not notice their votes had been changed in the review screen. It is reasonable to expect that a similar result would be obtained by changing the vote on the paper record.

In another report, Selker and Cohen present a study in which errors were intentionally inserted into the VVPAT [7]. No voters reported the errors during voting, and only 8% agreed that there were errors in the VVPAT when asked.

“U3 flash drives should not be permitted in the secure area and should never be used on the system.”

Distinguishing a regular USB stick from a U3 USB stick is non-trivial. U3 sticks are traditional USB sticks with a specific internal setup, and they cannot be distinguished simply by visual examination of the physical media. It is, therefore, unrealistic to suppose that an election official can easily distinguish between

the two types of flash drives in order to prevent the latter from entering a secure area.

“Even with these precautions, it is possible for malicious software to find its way into the network via results cartridges or other mobile data storage devices that may be used with the computers on the network. This is mitigated by ensuring a strict virus and spyware detection regime is implemented on the system, including ensuring the most recent updates are utilized.”

Here the vendor agrees that *it is actually possible* for a malicious software to find its way into the system. However, they claim that this can be prevented by deploying up-to-date antivirus and spyware detection software. Unfortunately, these programs are almost universally based on a set of signatures that are used to uniquely identify the presence of *known* malicious code. It is unrealistic to believe that they would be able to detect our malicious program, especially because our software does not exhibit traditional virus behavior and can be installed by the user just as any other Windows application.

The WinEDS machine should be secured to prevent any software from being installed on the computer. Relying on the presence of antivirus or spyware detection software would not foil the attack in this context.

“Attack scenario 2 (same as attack scenario 1, but with a fleeing voter that did not review their paper ballot) is likewise implausible. How would the malicious software know that the voter had actually fled? The interaction with the voter and a poll-worker is the same regardless of which one actually completes the ballot casting process. Poll-workers need to keep the voting machines open, so fleeing voters’ ballots are typically cast quickly after the voter leaves the voting machine, so time intervals would not aid the malicious software in determining when it could successfully change a voter’s ballot choices.”

The AVC Edge keeps track of the number of fleeing voters for auditing purposes, and the poll-worker is required to follow a specific procedure for casting a fleeing voter’s ballot. The poll-worker must press the yellow button on the rear of the machine in order to cast the ballot and proceed with the election. A malicious firmware can easily catch this event and act accordingly.

3.6 Voter Verified Paper Audit Trail

In their response, Sequoia claims that:

“[With a VVPAT in place,] none of these attacks described in the Red Team report are capable of success. All would be prevented or detected through use of the VVPAT and legally sufficient audits.”

Unfortunately, this is incorrect. In six out of the seven attack scenarios listed in Section 5 of our report, votes are modified such that the VVPAT, the results cartridge, the internal audit trail of the AVC Edge, and the total vote tally all reflect the same modified vote. Legally mandated audits or recounts would not be able to detect any tampering as a result of these attacks.

4 Conclusions

The Sequoia answer to our report characterizes our evaluation as unrealistic. In this document we have clearly explained why the evaluation was indeed performed in a realistic setting and the attacks we described are not against single components analyzed “in isolation,” rather, they are attacks against the voting process as a whole. In summary, our evaluation was methodologically correct, the threats are credible, and the outlined serious technical problems should be addressed.

Sequoia proposes that the security flaws in the Sequoia voting systems should be solved by their customers through improved security procedures. While some of these procedures might help, not all attacks can be foiled by adopting special procedures, and, in addition, it is unfair to put the burden of security on the customer.

In addition, the Sequoia report states that “*the versions of the hardware, firmware and software systems evaluated were developed several years ago. While it can not be guaranteed that all of the extremely improbable vulnerabilities identified are prevented by subsequent product development and updates, many are specifically addressed.*” Therefore, they admit that they fixed some (unspecified) security problems that are present in the current version of their system. Unfortunately, they never produced any advisory or document that specifies which vulnerabilities were found and how they were fixed. Instead, they suggest that Sequoia’s customers “*should consider purchasing new machines or updates to existing units that meet the 2005 VVSG, and subsequently adopt the 2007 VVSG when available.*” Therefore, Sequoia itself suggests that its customers should get rid of their current systems and adopt different ones.

References

- [1] Andrew W. Appel. Affidavit of Andrew W. Appel for the Superior Court of New Jersey in the New Jersey Voting Machines Lawsuit. <http://www.cs.princeton.edu/~appel/nj-voting-case/Appel-feb07-certif.pdf>, February 2007.
- [2] Brennan Center Voting System Security Task Force. The Machinery of Democracy: Protecting Elections in an Electronic World. Technical report, Brennan Center for Justice, June 2006.
- [3] Sarah P. Everett. *The Usability of Electronic Voting Machines and How Votes Can Be Changed Without Detection*. PhD thesis, Rice University, 2007.
- [4] Institute for Security and Open Methodologies. Open Source Security Testing Methodology Manual. <http://www.isecom.org/osstmm/>, December 2006.
- [5] Roger G. Johnston. Tamper-Indicating Seals. *American Scientist*, 94(6), November-December 2006.
- [6] Regents of the University of California. Scope Of Work, June 2007.
- [7] Ted Selker and Sharon Cohen. An active approach to voting verification. Technical Report 28, Caltech/MIT Voting Technology Project, May 2005.

- [8] Sequoia Voting Systems. Response from Sequoia Voting Systems to the California Secretary of State's Office on the Top-To-Bottom Review of Voting Systems Red Team and Accessibility Principal Investigator Reports on Sequoia's WinEDS version 3.1.012/Edge/Insight/400-C, July 30 2007.
- [9] Brian Todd. E-Voting Vulnerabilities. <http://www.cnn.com/video/#/video/us/2007/07/31/todd.ca.evoting.flaws.cnn>, July 2007.
- [10] J. Wack, M. Tracy, and M. Souppaya. NIST Special Publication 800-42: Guideline on Network Security Testing. <http://csrc.nist.gov/publications/nistpubs/800-42/NIST-SP800-42.pdf>, October 2003.
- [11] Kim Zetter. E-Vote Software Leaked Online. <http://www.wired.com/politics/onlinerights/news/2003/10/61014>, October 2003.