Security Evaluation of the Sequoia Voting System
Public Report

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

Executive Summary

The California Secretary of State entered into a contract with the University of California to test the security of three electronic voting systems as part of her top to bottom review. Each "red team" was to try to compromise the accuracy, security, and integrity of the voting systems without making assumptions about compensating controls or procedural mitigation measures that vendors, the Secretary of State, or individual counties may have adopted.

This report presents the security analysis of the Sequoia voting system, as performed by the Security Group of UC Santa Barbara. The Security Group was lead by Giovanni Vigna and Richard Kemmerer and included Davide Balzarotti, Greg Banks, Marco Cova, Viktoria Felmetsger, 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 are part of the Sequoia 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." [5]

The "Overview of the Red Team Reports" [1] discusses the goals, context, and threat models used.

During the testing, the Red Team tried a limited number of attacks, due to the time constraints. The testing started on June 12 and terminated on July 10. This is a very limited amount of time for the testing of a complex system such as the Sequoia voting system.

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.

1 Introduction

The use of computers in performing voting and tallying introduces serious concerns about the integrity and confidentiality of the voting process. A number of systems have been proposed that use a combination of electronic vote casting, tallying, and paper trails.

In this report, we present the security analysis of one such system. The analysis, which was sponsored by the California Secretary of State, was performed by the Security Group of UC Santa Barbara.

The threat model for this analysis was described as follows in the "rules of engagement" that were given to us: "The testing assumes two classes of threats: insiders and outsiders. 'Insiders' are those who have physical access to all components of the voting system, including the election management system. 'Outsiders' are those with restricted physical access to the systems, such as voters, poll workers, and observers. Where system security relies upon proper application of procedures, it may be appropriate to examine the consequences of any failure to follow procedures. For example, if the systems are capable of networking, one might examine what are the consequences if they should be connected to a network in spite of procedural requirements to the contrary; the obvious question is what could happen if someone erroneously connects the system to a network. Our threat model also assumes that an attacker has access to all details of the system, including source code, and knows exactly how they are used. This provides for those attackers who acquire knowledge over a long period of time, or who have inside access to the system design and implementation (for example, because documents or source code are accidentally posted to the Internet)." [2]

The "Overview of the Red Team Reports" [1] discusses the context of this threat model, as well as the specific "rules of engagement".

The rest of this report documents our findings. More precisely, in Section 2, we present a high-level view of the Sequoia system under evaluation. In Section 3 we describe a number of security issues that were known *before* our testing started. Then, in Section 4, we present the results of our evaluation. In Section 6, we combine the different attacks we discovered to describe how the results of an election could be modified. Finally, in Section 7 we present our concluding remarks. The confidential report contains a detailed list of the attacks.

2 The Sequoia Voting System

The Sequoia voting system is composed of a number of hardware and software components. The following paragraphs describe the major components in order to provide some background information to the reader.

In the description that follows we will refer to the configuration of the system that was provided to us by Sequoia. The configuration presented was confirmed to be representative of how the Sequoia voting system infrastructure would be deployed in the field. All components listed in the following sections were present for the duration of the review and evaluated by the red team as fully as possible given the time constraints of the review.

The specific system we tested consisted of the following components:

- WinEDS, version 3.1.012
- AVC Edge Model I, firmware version 5.0.24
- AVC Edge Model II, firmware version 5.0.24
- VeriVote Printer
- Optech 400-C/WinETP firmware version 1.12.4

- Optech Insight, APX K2.10, HPX K1.42
- Optech Insight Plus, APX K2.10, HPX K1.42
- Card Activator, version 5.0.21
- HAAT Model 50, version 1.0.69L
- Memory Pack Reader (MPR), firmware version 2.15
- Various removable media, including:
  - Results Cartridges
  - USB flash drives
  - Voter SmartCards
  - Memory packs

2.1 WinEDS

WinEDS is the software application responsible for the management of the election process. This includes both programming and configuring the election as well as tabulating and reporting the results. Through WinEDS, the election managers can initialize all the hardware components involved (e.g., the Edge and HAAT hardware) and count the votes collected by the machines.

WinEDS is composed of a Windows client application and Microsoft SQL Server 2000 database hosted on a Windows XP SP2 machine. The database server can run on a separate machine, but in the setup that we analyzed both the WinEDS client application and the database ran on the same laptop.

2.2 HAAT

The HAAT (Hybrid Activator, Accumulator and Transmitter) is a device that is used to initialize the cards used by the voter when using the Edge component. We tested the HAAT Model 50, which runs Windows CE Embedded version 5.0.

From the documentation [11]: "[The] HAAT50 is the component that serves as the voter's access to the Edge2 direct-record electronic touch-screen voting machines through activation of a SmartCard interface and at the precinct level acts as an accumulator for consolidating and tallying results. Unlike the HAAT100, the HAAT50 is not able to print or transmit these results, since the HAAT50 does not use any printer or internal modem.

In the SmartCard activation function, on Election Day, after establishing the voter's identity and party affiliation, the poll worker inserts a voter activation card (SmartCard) into the HAAT50 Unit, and follows the proper procedures to activate it. After the card is activated, the poll worker hands the activated SmartCard to the voter who then uses the card to access the voting machine. After the voter completes voting, the voter returns the SmartCard to the poll worker. The HAAT50 can also create an USB Backup Cartridge to preserve all HAAT information. Among the information this Backup Cartridge contains, there is information about the HAAT identification, precinct information and error and audit logs."

The technical documentation [11] presents a detailed description of the HAAT component.

2.3 Card Activator

The Card Activator is a device similar to the HAAT. The Card Activator has an x86 processor running MS-DOS.

The technical manual [7] states that:

The Card Activator (CA) is a component of the AVC Edge, and serves as the voter's access to the AVC Edge direct-record electronic touch-screen voting system. After establishing the voter's identity and party affiliation the poll worker inserts a voter Activation Card into the Card Activator, presses the appropriate number on the Card Activator keypad that designates the voter's party. After the card is activated, the poll worker hands the activated Activation Card to the voter who then uses the card to access the AVC Edge voting system.

2.4 AVC Edge

The AVC Edge is a Direct-Record Electronic (DRE) voting machine. The device is responsible for loading and validating ballot definitions created by WinEDS, presenting ballots to voters, and recording votes cast by voters. The actual voting procedure is performed through a touch screen interface. Each voter receives a SmartCard activated by either a Card Activator or a HAAT device in order to access the voting procedure. The AVC Edge is also equipped with a printer that allows the voter to verify the vote he/she cast.

Votes are stored both internally and on a removable Results Cartridge. When the polls are closed, the cartridge is removed and the contents processed by WinEDS for tallying and reporting purposes.

The AVC Edge can also be equipped with an optional accessibility package. This package consists of a special audio unit that allows sight-impaired voters to cast ballots.

The documentation [6] presents a detailed description of the AVC Edge Model I and II, both of which were tested by the red team. Though each AVC Edge model differs somewhat in physical construction and internal architecture, both variants are for all intents and purposes functionally identical. In particular, both models are capable of being controlled by the same system firmware image.

2.5 Optech 400-C

The Optech 400-C is a large, high-speed device used to count "mark-sense ballots" and tabulate the results. The device is composed of an optical reader attached to a PC that contains the software necessary for the operation of the machine. The PC is protected from physical access by means of a metal door and a lock.

The technical documentation [8] presents a detailed description of the Optech 400-C.

2.6 Optech Insight Plus

The Optech Insight Plus is a precinct-based optical reader for "mark-sense ballots". The system is used to count and tabulate ballots. The device is configured by means of a memory pack prepared by WinEDS which contains the ballot definition and is also used to store the results.

The technical documentation [9] presents the technical details of the Optech Insight Plus.

2.7 Removable Media

The Sequoia voting infrastructure as configured for review by the red team utilizes a set of removable media devices to transfer election data between the WinEDS console, AVC Edge DREs, Optech 400-C, Optech Insight, Card Activator, and HAAT.

The first category of removable media is the Results Cartridge, which is simply a PCMCIA memory card that is initialized by WinEDS with an election definition. A Results Cartridge can then be loaded into a Card Activator to prepare it for generating valid voter SmartCards for the defined election. A Results Cartridge that has been initialized by WinEDS is also used to load an election definition into an AVC Edge or update the firmware of an Edge.

Generic USB flash drives comprise the next category of removable media. These are utilized in several roles, the first of which is to transfer an election definition from WinEDS to a HAAT. In this capacity, a USB flash drive serves an analogous purpose to that of a Results Cartridge for a Card Activator. A USB drive, however, is also used to transfer an election definition to the Optech 400-C/WinETP console.

The next category of removable media is the voter SmartCard. In the context of the Sequoia voting system, SmartCards are simple, memory-constrained devices utilized as hardware tokens. They are initialized by a Card Activator or HAAT for use by a voter and are intended to 1) authenticate a voter to an AVC Edge, and 2) authorize the voter to cast a single ballot.

Finally, the Sequoia voting system utilizes "Memory Packs" to transfer an election definition to an Optech Insight optical scanner. These devices are constructed from a combination of flash memory and EPROMs housed in a protective plastic enclosure, and are programmed by WinEDS through the Memory Pack Reader through a proprietary connector.

3 Known Issues

There are a number of known issues with the Sequoia voting system. These issues have been highlighted by previous evaluations or as "incidents" in official elections. The Electronic Frontier Foundation (EFF) published a list of known problems associated with the Sequoia voting system [3].

3.1 The Alameda County Evaluation

The Sequoia voting system was evaluated by Pacific Design Engineering for Alameda County [4]. That evaluation states that "No practical, realizable vulnerabilities were uncovered that could not be eliminated through appropriate countermeasures involving additional software and data validation or improved physical process countermeasures." and also that "From a technology perspective, the Sequoia Electronic Voting System acquired by Alameda County, along with the processes and countermeasures planned by Alameda County for Election Day, can be considered secure."

The problems found by Pacific Design Engineering can be summarized as follows:

- The WinEDS and the other servers use non-encrypted text passwords when communicating.
- The Edge uses constant hashes and DES encryption keys that can be discovered if somebody has physical access to a machine.

- The Edge's memory cartridge results are not bound together cryptographically, and therefore the content of one cartridge could be copied onto another.
- The WinEDS system uses Windows and therefore inherits the vulnerabilities associated with that operating system.

The Pacific Design Engineering report provides some guidelines about how to ameliorate these situations by changing some procedures and updating/extending the software. The ultimate conclusion of that report is that "The Sequoia Electronic Voting System selected by Alameda County to conduct election operations is inherently secure. The relatively low risk vulnerabilities that do exist in the Sequoia Electronic Voting System components are readily remedied by network security and human process countermeasures, which have been adopted by Alameda County. This positive analysis is based on the fact that Sequoia precinct equipment, although microprocessor-based, does not have an underlying operating system that can be obtained in the public domain, inspected, analyzed, and vulnerability-exploiting tools created, as is true with Windows, Linux, and most every standard operating system."

3.2 Multiple Votes Attack

An attack enabling a voter to vote multiple times without the need for an activated SmartCard has been reported. We verified that this attack is still applicable to the Edge devices that we tested.

4 Findings

In this section we summarize the most important security issues that we identified in our tests. Due to time constraints we were not able to evaluate every single component thoroughly. Therefore, the findings described hereinafter are to be considered an incomplete evaluation of the system.

4.1 Arbitrary Code Execution

We have developed a working exploit that allows an attacker to overwrite an AVC Edge firmware with a malicious version. The exploit is delivered by constructing a Results Cartridge that contains a malicious election definition. During loading of the election into the Edge, a vulnerability is exploited that results in arbitrary code execution. An associated payload was also developed that replaces the system firmware without any user intervention or discernable notification.

The development of the exploit was made easier by the fact that the Edge runs a proprietary OS. The simplified memory layout of the OS made it easier to predict memory addresses necessary for successful exploitation.

4.2 File Overwriting

The AVC Edge firmware is vulnerable to a directory traversal attack that can name, and hence overwrite, the files containing the boot loader and the system firmware. Furthermore, a vulnerability in how firmware update cartridges are handled allows an attacker to add files to the cartridge and thereby overwrite files on the disk in the Edge machine.

4.3 Accuracy Testing Mode Detection

We found that, in the case of the Edge, the pre-election correctness test is performed by switching the machine to a specific "Logic and Accuracy Test (LAT) mode". We created a malicious firmware that detects the current operating mode and performs different functions accordingly. For example, the malicious firmware could perform a correct count during the LAT procedure and then introduce errors during the actual voting procedure.

4.4 Execution of Modified Firmware

There is no way to determine which version of the firmware is running on an Edge device. The Sequoia documentation states that the firmware is stored in ROM and that checksum-based mechanisms are used to determine if the firmware has been modified maliciously. However, in reality there is no secure, hardware-based mechanism to ensure that no corrupted firmware gets loaded and executed. In addition, the Edge firmware is stored on a flash memory card and can be easily overwritten. Hardware support for trusted software execution and the use of non-writable memory would protect the Edge device from a large range of attacks from both insiders and outsiders.

4.5 Availability of an Interpreter in Violation of Guidelines

Even though Section 4.2.2 of the 2002 Voting Systems Standards says that "*Self-modifying, dynamically loaded, or interpreted code is prohibited, except under the security provisions outlined in section 6.4.e*," the Edge firmware was discovered to include a shell-like scripting language interpreter. This language includes, among others, several interesting commands:

- A command to set the protective counter of the machine, which was described to us by the Sequoia representatives as tamper-proof.
- A command to set the machine's serial number.
- A command that can be used to overwrite arbitrary files on the internal compact flash drive, including the system firmware or audit trail.
- Commands to reboot the machine at will.

The interpreter was tested by the red team and was found to be fully enabled. In particular, the ease in which the interpreter could be coerced into performing malicious actions led to its use in several exploits developed by the red team, although those exploits could have been written without using the interpreter.

4.6 Arbitrary Directory Creation Through Traversal Attack

The AVC Edge voting machine ballot loading logic is vulnerable to a directory traversal attack that leads to a denial of service. The particular vulnerability arises because there is insufficient validation performed by the Edge firmware of the data stored on a Results Cartridge. This allowed an attacker to create directory structures on the internal compact flash. It is not known whether this vulnerability can lead to the execution

of arbitrary code, but the vulnerability allows an attacker to perform a denial of service that prevents a ballot from being loaded. This is accomplished by constructing a directory in the place of a critical system file.

4.7 Automatic Execution of Code

The WinEDS host operating system provided and configured by Sequoia is configured so that it will execute an "autorun" file whenever removable media is inserted

Recently, a new type of USB flash drive, called U3, has been developed. The U3 has two separate partitions, with one of these partitions emulating a CD-ROM. This feature allows one to create a malicious USB flash drive that executes code upon insertion when autorun is enabled for CD-ROMs. Using this feature, we created a malicious USB flash drive that will silently install a Trojan on any Windows machine it is inserted into. This Trojan is set to run automatically at boot-time, and simply monitors all device change events through the Windows shell API and a special U3-device-aware API. When the Trojan notices a Results Cartridge being written, for instance, it has the opportunity to rewrite some of the files after they have been written by WinEDS and before they are inserted into the Edge machine. In addition to modifying ballot definitions and results, the Trojan program could easily spread to other USB flash drives by re-infecting any U3 device that is subsequently inserted or by attempting to take advantage of any remote vulnerabilities present on machines connected over the local network.

4.8 Security of the MS SQL Server

The WinEDS SQL Server is supposed to be a secured, stripped down machine. In the documentation ([10], p. 3-1), it is stated that: "WinEDS currently does NOT utilize code outside of MS SQL Server and no connections or permissions are required on the server (besides SQL Client.) The lack of server access by individual users provides the application with a secure client-server environment. The election data stored on the server can only be modified by authorized users only through the application."

Unfortunately, this is not true. In fact, it is possible to connect to the database and completely compromise the MS SQL server host without using the WinEDS application. This is achieved by exploiting two security problems. First of all, the WinEDS access control procedures can be bypassed. Second, the MS SQL server delivered with the Sequoia system enables users to execute arbitrary commands.

4.9 Votes Encrypted Using Static Key

The contents of the Results Cartridge are not protected by any cryptographic signatures, and can easily be modified. The vote data is protected using DES encryption, but the key is stored in the firmware and can easily be recovered by an attacker. If the key is compromised, it is not possible to change the key without changing the firmware. As a consequence the system would have to go through the voting system certification process again.

4.10 Possible Unsafe OS Choices

The WinEDS documentation [10] states that Windows 98 or Me could be used for the WinEDS client machine. This is a problem, because those Windows versions provide no user-level security.

4.11 Physical Security

There are serious concerns about the physical security of the different hardware components. Even though the Edge devices can be protected by seals, these protections as set up by the Sequoia representatives were easily bypassed. Other security-critical parts of the system were accessed by simply unscrewing a few screws. All components (Optech 400-C, Edge, HAAT and Card Activator, Insight Optical Scanner, and Memory Packs) are vulnerable to these attacks.

4.12 Reversible Password Hash

The password stored on the update cartridge is not stored as a password hash (that is, as the result of a one-way hash function such as SHA-1). Instead, the password is simply DES encrypted with a key that is stored in the firmware image. Therefore, someone with access to an update cartridge could decrypt the password if he/she knows the encryption key. If this password were used to protect components other than just the update cartridge (e.g., the database or the windows accounts, etc.), then the attacker could also gain access to those components.

4.13 Forging Update Cards and Voter Cards

Forging update cartridges is possible for a number of reasons. First of all, the password used to protect the updating process from abuse is stored on the cartridge itself. Therefore, it is possible to create a valid update cartridge with a known password. In addition, the integrity check on the files is achieved through a very simple checksum, which can be easily re-computed after the files are modified. Finally, sometimes the checksum verification process can be bypassed altogether.

It is possible to arbitrarily forge voter SmartCards because the SmartCards are DES-encrypted using a static key. Therefore, one can recover the key from the binary image of the Edge firmware and both decrypt legitimate SmartCards and encrypt forged SmartCards. In addition, the contents of a SmartCard are "checksummed" and the expected hash is stored on the SmartCard itself. Therefore, once the contents are modified it is trivial to re-compute the correct checksum, making the card appear legitimate.

Finally, each Edge machine records the SmartCards that have been used on it (to prevent replay attacks). The Edge identifies a card by the serial number of the card's programmer and the time the card was issued. Both pieces of information are part of the SmartCard's contents, and, therefore, they can be easily modified.

5 Successful Attack Scenarios

An important question that needs to be answered in a clear way is "what happens if the number of votes returned by a machine does not match the number of votes recorded by the voting officials?" For example, what happens if an Edge system returns a vote count of 5400 but only 5000 people showed up to vote?

Our understanding is that it is not possible to tell which votes are the ones surreptitiously added and which votes are the legitimate ones. Therefore, any attack that adds additional votes should be considered an effective attack that will affect the outcome of an election, regardless of the fact that the attack will be detected (by a simple count check or as the result of a partial manual check of the voting results.)

We implemented and tested all the attack scenarios in this section.

5.1 Attack Scenario 1

In this scenario, we assume that an attacker is able to drop a USB flash drive in the pool of USB drives used to initialize the HAAT systems. This drive contains a Trojan application that is invoked as part of the autorun procedure. When the drive is inserted in the computer on which WinEDS is running, the Trojan is executed and becomes silently active in the background. No noticeable event is produced. Even if the autorun feature were turned off, it is possible to execute a trojan application by several means. This can for instance be done by explicitly double clicking on the application or by constructing a worm that can be injected into the county intranet.

At this point, the Trojan monitors the insertion and removal of other removable media and infects them as they are inserted in the WinEDS machine. In addition, the Trojan detects whenever a Results Cartridge is inserted and modifies it so that it exploits the integer overflow vulnerability and installs the malicious firmware on the AVC Edges.

Before the election, the cartridge is inserted in an Edge machine to load the ballots. After the cartridge is inserted, the exploit is automatically triggered, and, as a result, a malicious firmware is installed. The malicious firmware behaves normally during the pre-election LATs, but it starts to actively modify the election results as soon as the actual election starts.

The malicious firmware monitors the votes being cast and modifies the ballot to give advantage to a certain candidate. The resulting ballot is then printed and presented to the voter. If the voter actually checks the printed output of his/her votes and detects that an error has been made, then the malicious firmware lets the voter re-cast his/her vote and stops modifying the voted values for a while, to avoid being detected.

If one assumes that a large portion of voters does not check their printed results, this attack scenario can modify the results of an election, and it cannot be detected by a manual audit.

5.2 Attack Scenario 2

In this scenario, the Edge machine is infected in the same way as described in the previous scenario. However, this time the malicious firmware takes advantage of "fleeing" voters. These are voters that leave the voting station *before* having completed their voting (this is not uncommon).

When this happens, a poll worker has to press a button and follow a specific procedure to complete the casting of the ballot. For privacy reasons, the poll worker has no access to the content of the ballot and therefore he/she cannot verify that what is being recorded faithfully represents the voter's choices. Therefore, the malicious firmware detects that the poll worker has initiated the procedure to manually complete the ballot casting and the firmware records a modified vote.

5.3 Attack Scenario 3

In this scenario, the Edge machine is compromised in the same way as described for the first scenario. However, in this case the firmware prints a copy of the voter's actual choices. Then, the firmware asks the voter to confirm his/her choices.

Once the voter confirms the printed copy, the firmware displays a message on screen that says "Please Wait, Recording Vote" for a few seconds. Then, the malicious firmware displays a screen saying "Thank

you, vote recorded." After a few more seconds, during which it is assumed that the voter leaves the station, the machine prints "VOIDED" on the receipt and jumps back to the ballot.

At this point, even though the voter is sure that he/she completed the voting process, the station appears to have been left prematurely (that is, the voter appears to be a "fleeing" voter). Therefore, a poll worker will have to initiate the manual voting procedure. The attack then proceeds as described in Scenario 2.

5.4 Attack Scenario 4

This scenario is a variation on Scenario 3. In this case, after the machine prints "VOIDED", instead of jumping back to the ballot, it completes the voting process by casting (and printing) a modified vote.

5.5 Attack Scenario 5

In this scenario, an Edge is delivered before voting day. During the night, an attacker removes the back of the Edge and replaces the firmware's flashcard with one containing a malicious firmware. After that, the results of the election process can be tampered with using one of the techniques described in the previous scenarios.

5.6 Attack Scenario 6

In this scenario, an attacker has access to a number of blank voter cards. Through a side channel, the attacker obtains access to the static key used to encrypt the voter cards. Using this knowledge, the attacker creates a number of valid voter cards. When the attacker goes to the polling station, she has the opportunity to vote multiple times.

5.7 Attack Scenario 7

In this scenario, someone with limited access to election functionality on a WinEDS workstation directly connects to the MS SQL Server running on a separate WinEDS server machine. Since the WinEDS access control mechanisms are not enforced outside of the WinEDS software, the attacker has full administrative privileges on the MS SQL server. As the next step, the attacker transfers a malicious program to the database, and installs the program on the WinEDS server. The program runs in the background and from time to time connects to the database and changes election data in a consistent way. At this point, this program can be used to change any election data (such as, ballot definitions, election results, etc.) transparently from the local machine.

The installed program can be left on the machine as a Trojan. Assuming that the machine is not re-installed between elections, the program could guarantee the attacker access to the server during the next election.

6 Potential Attack Scenarios

This section contains a plausible, but untested, attack scenario.

6.1 Attack Scenario 8

In this scenario, an authorized user gets physical access to a 400-C machine. The attacker accesses the PC within the enclosure and reboots the PC with a bootable CD containing a different OS. The attacker then proceeds to install a Trojan application on the Windows system installed on the PC. The Trojan application stays silent until a ballot with a specific (and unlikely) selection of options appears. At that point, it will start modifying the vote counters introducing a bias towards a specific candidate. By waiting until a specific ballot has been processed it is possible to hide the malicious behavior from the LAT procedures.

7 Conclusions

Although, we did not have enough time to perform a complete evaluation of the Sequoia voting system, we exposed a number of serious security issues. These vulnerabilities could be exploited by a determined attacker to modify (or invalidate) the results of an election.

All the attacks described in this report can be carried out without any knowledge of the source code. In fact, we were able to extract and analyze the Edge's firmware binary representation. In addition, we were able to extend the firmware by using binary patching. This technique allowed us to create a "debugging" version of the firmware, as well as several different "malicious" versions.

The implementation of the attacks did not require access to the source code.

References

[1] Matt Bishop. Overview of the red team reports, July 2007.

[2] Matt Bishop. Protocol for red team testing, May 2007.

[3] EFF. Electronic Voting Machine Information Sheet, Sequoia Voting Systems AVC Edge, October 2006. Version 1.1.

[4] C. Humphreys and C. Merchant. Sequoia Voting Systems Vulnerability Assessment and Practical Countermeasure Development for Alameda County. Technical report, Pacific Design Engineering, October 2006.

[5] Regents of the University of California. Scope Of Work, June 2007.

[6] Sequoia Voting Systems. *AVC Edge, System Overview*, document version 1.00 edition, May 2005.

[7] Sequoia Voting Systems. *Card Activator: Software Specification, Release 5.0*, document version 1.01 edition, May 2005.

[8] Sequoia Voting Systems. *Optech 400-C, System Overview*, document version 1.02 edition, October 2005.

[9] Sequoia Voting Systems. *Optech Insight Plus, System Overview*, document version 1.01 edition, September 2005.

[10] Sequoia Voting Systems. *WinEDS (Windows Election Database System) For Avc Edge/Advantage, Release 3.1*, document version 1.08 edition, December 2005.

[11] Sequoia Voting Systems. *HAAT System Overview*, document version 1.08 edition, January 2006.