Overview of Red Team Reports

Matt Bishop, Principle Investigator, University of California, Davis

1.0. 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. The red teams demonstrated that, under these conditions, the technology and security of all three systems could be compromised.

2.0 Goals

In May 2007, the California Secretary of State began a study of all electronic voting systems currently certified in California. This "top to bottom review" (TTBR) was to determine whether the systems currently certified should be left alone, or specific procedures required to provide additional protections for their use, or the machines simply decertified and banned from use. As part of this study, the Secretary contracted with the University of California to conduct a "red team" review of the systems. The specific goal of the Red Team study was "to identify and document vulnerabilities, if any, to 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." ([1], p. 5).

A red team study, also called a penetration study, examines a system from the point of view of an attacker, and analyzes the system to determine how secure it is against an attack. Such a study requires establishing several parameters:

- The specific goals of the system: what is it to do?
- The threat model: with whom or what are the testers concerned?
- The information to be made available to the testers: how much do they know at the start?
- The environment in which the system is used: what policies and procedures are to be applied?
- The specific "rules of engagement": what are the team members allowed to do?

For this TTBR, the specific goals of each system are to record, tabulate, tally, and report votes correctly and to prevent critical election data and system audit data from being altered without authorization. The threats were taken to be both insiders (those with complete knowledge of the system and various degrees of access to the system) and outsiders (those with limited access to the systems). As a result, *all* information available to the Secretary of State was made available to the testers. The testers were told to assume that the environments in which the systems were used would vary, and that the

testers could do whatever they thought necessary to test the machines. The testers therefore assumed the attackers would include anyone coming in contact with the voting systems at some point in the process – voters, poll workers, election officials, vendor employees, and others with varying degrees of access [18].

In developing attack scenarios, the red teams made no assumptions about constraints on the attackers. We recommend that future Red Teams should adopt a similar attitude.

The testers did *not* evaluate the likelihood of any attack being feasible. Instead, they described the conditions necessary for an attacker to succeed. This approach had several benefits:

- The testers could focus on the technology rather than on the policies procedures, and laws intended to compensate for any technological shortcomings.
- In California, specific procedures for controlling access to the election systems and for setting up, using, and storing the election systems is a local matter. As there are 58 different counties, there are at least 58 different sets of procedures. It was impractical for the red team testers to evaluate them.
- If a problem is discovered, the people who know the law and election policies and procedures can modify their policies and procedures appropriately to attempt to address the problem.
- Finally, the effectiveness of the policies and procedures used to control and protect the election systems depends on their implementation. Policies and procedures that look effective on paper may be implemented poorly, rendering them ineffective. It was impractical to evaluate this aspect of the policies and procedures.

Therefore, the results of this study must be evaluated in light of the context in which these election systems are used. This emphasizes a key point often overlooked in the discussion of the benefits and drawbacks of electronic voting systems: those systems are part of a process, the election process; and the key question is whether the election process, taken as a whole, meets the requirements of an election as defined by the body politic.

The participants in this study hope our work contributes in some measure to answering that question.

2.1 Systems Examined

Three systems were reviewed in this study.

Diebold. The Diebold GEMS 1.18.24/AccuVote consisted of the following components:

- GEMS software, version 1.18.24
- AccuVote-TSX with AccuView Printer Module and Ballot Station firmware version 4.6.4
- AccuVote-OS (Model D) with firmware version 1.96.6
- AccuVote-OS Central Count with firmware version 2.0.12

- AccuFeed
- Vote Card Encoder, version 1.3.2
- Key Card Tool software, version 4.6.1
- VC Programmer software, version 4.6.1

Hart Intercivic. The Hart Intercivic System 6.2.1 consisted of the following components:

- Ballot Now software, version 3.3.11
- BOSS software, version 4.3.13
- Rally software, version 2.3.7
- Tally software, version 4.3.10
- SERVO, version 4.2.10
- JBC, version 4.3.1
- eSlate/DAU, version 4.2.13
- eScan, version 1.3.14
- VBO, version 1.8.3
- eCM Manager, version 1.1.7

Sequoia. The Sequoia WinEDS version 3.1.012/Edge/Insight/400-C 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

2.2 Team Organization

Two red teams were organized. One team, led by Robert P. Abbott, was based in Sacramento at the Secretary of State's secure facility. The second team, led by Giovanni Vigna and Richard Kemmerer, was based at the University of California, Santa Barbara, and came to Sacramento as needed to use the system. The first team examined the Diebold and Hart systems; the second, the Sequoia system.

3.0 Context

Two contexts are relevant: that of the election process, and that of the system certification.

3.1 Computers as Part of a Process

It is commonly accepted that no computer or computer-based system, called an *information technology system*, can be made completely secure. It is also commonly accepted that the managers of an information technology system have a responsibility to

develop sufficient controls in and around a system to the point that continued operation of the system meets the requirements of the organization. "Organizations should satisfy the quality, fiduciary and security requirements for their information, as for all assets." ([2], p. 5) "A high level of security management may have to be focused only on the most critical enterprise systems." ([2], p. 20)

Electronic voting systems are special purpose computer systems. As such, they require compensating controls just like any other computer system. Protecting computer systems embodies several topics, which, taken together, constitute an Information Technology Security Program. There are many reference guides toward establishing such a program, such as FIPS PUB 200 [7].[1]

An Information Technology Security Plan includes three topics of particular interest to owners of electronic voting systems:

1. *Physical security*. Electronic voting machines must be protected against unauthorized physical and electronic access while: a) in storage; b) at polling locations; c) at the central election center; and d) while in transit between storage, the polls, and the election center.
2. *Security training of staff*. Election officials and poll workers must be acquainted with the concepts of information technology security as well as procedures to invoke when security rules are violated. Training must also address the concept of social engineering, which is a collection of techniques used to manipulate people to perform actions that are forbidden, or divulge information that should remain confidential. An example is pretexting, a technique in which an investigator obtains information about someone's records by pretending to be that person. In 2006, investigators used this technique to try to determine the source of leaks from the HP Board of Directors [17].
3. *Contingency planning*. Plans must be developed to handle the situation in which a polling place or a voting station is rendered inoperative. Every contingency must be thought of and thought through in advance. A work-around process or procedure must be developed and tested for each contingency.

Many, but not all, of the attack scenarios contained in these reports would be mitigated by fully addressing these three topics. The feasibility of developing policies and procedures that can be effectively implemented, what those policies and procedures should be, and how they should be implemented, is a matter that lies within the knowledge and experience of election officials and the California Secretary of State.

Security traditionally relies on layers of mechanisms; this is called *defense in depth*, *layered defense*, or *separation of privilege*. The idea is to force an attacker to breach several security mechanisms to compromise the system, rather than one. Procedures form

---

[1] Other examples are ISO 9001:2000 [3], the CMMI [4], PRINCE2 [5], and PMBOK [6]. Organizations such as SANS (http://www.sans.org) and (ISC)^2 (http://www.isc2.org) provide training and education on information security practices, also.

some of these layers of defensive mechanisms. Proper system configuration and implementation form additional layers of defensive mechanisms. Security plans should *always* rely on multiple layers. In particular, that procedures could mitigate or block the attack scenarios in this report in no way relieves vendors of their responsibility to locate, repair, and fix the vulnerabilities in their products that these attacks exploit.

Finally, no security should *ever* rely solely on secrecy of defensive mechanisms and countermeasures.[2] While not publishing details of security mechanisms is perfectly acceptable as one security mechanism, it is perhaps the one most easily breached, especially in this age of widespread information dissemination. Worse, it provides a false sense of security. Dumpster diving, corporate espionage, outright bribery, and other techniques can discover secrets that companies and organizations wish to keep hidden; indeed, in many cases, organizations are unaware of their own leaking of information. A perhaps classic example occurred when lawyers for the DVD Copyright Control Association sued to prevent the release of code that would decipher any DVD movie file. They filed a declaration containing the source code of the algorithm. One day later, they asked the court to seal the declaration from public view—but the declaration had been posted to several Internet web sites, including one that had over 21,000 downloads of the declaration! [9] More recently, Fox News reported that information posing "a direct threat to U.S. troops … was posted carelessly to file servers by government agencies and contractors, accessible to anyone online" [8], and thefts of credit card numbers and identities are reported weekly and growing in number. Thus, the statement that attackers could not replicate what red team testers do, because the red team testers have access to information that other attackers would not have, profoundly underestimates the ability and the knowledge of attackers, and profoundly overestimates the infallibility of organizations and human nature.

3.2 Certification

The California Secretary of State must certify any electronic voting system before it can be used in California elections. One of the requirements is that the system be federally certified to meet the 2002 Voting System Standards (VSS) [10]. Independent testing authorities (ITAs) test the electronic voting system to certify compliance with these standards. All three systems in this study were so certified [11,12,13].

The quality of the 2002 standards is inadequate (see Barr *et al*. [14] for an analysis of the 2002 standards and their successor, the 2005 Voluntary Voting System Guidelines [15]). Further, questions have been raised about the effectiveness of the testing. For example, Ciber, Inc., an ITA, has been denied interim accreditation for testing voting systems by the Federal Election Assistance Commission after finding that Ciber "was not following its quality-control procedures and could not document that it was conducting all the required tests" [16].

[2] This is often called "security through obscurity".

4.0 Limits and Problems of the Study

The major problem with this study was time. Although the study did not start until mid-June[3], the end date was set at July 20, and the Secretary of State stated that under no circumstances would it be extended. This left approximately 5 weeks to examine the three systems. For budgetary reasons, the UCSB team (which was examining 1 system) planned to conclude its examination by July 10. In order to do as much as possible, it was decided to examine the Hart and Diebold systems simultaneously, rather than allocate 2.5 weeks to each. The examination of both these systems concluded on July 20.

The short time allocated to this study has several implications. The key one is that ***the results presented in this study should be seen as a "lower bound"; all team members felt that they lacked sufficient time to conduct a thorough examination, and consequently may have missed other serious vulnerabilities***. In particular, Abbott's team reported that it believed it was close to finding several other problems, but stopped in order to prepare and deliver the required reports on time. These unexplored avenues are presented in the reports, so that others may pursue them. Vigna's and Kemmerer's team also reported that they were confident further testing would reveal additional security issues.

The second problem was a lack of information. In particular, various documents did not become available until July 13, too late to be of any value to the red teams, and the red teams did not have several security-related documents.[4] Further, some software that would have materially helped the study was never made available. As a specific example, when installing the system initially, the Hart personnel used a program to upgrade firmware on their system. The red and source code team members present asked for a copy of that program, because it would enable the testers to determine whether anyone could upgrade the firmware[5]. Otherwise, the teams would have to discover the protocol used for upgrading the firmware and write programs to do it themselves. The person doing the installation stated that the program was proprietary and would not be released to the Secretary of State and the teams. The teams asked the Secretary of State to obtain the program from Hart. The request was repeated in a phone call with Hart engineers on July 16, and Hart said they would have to discuss it among themselves. The software was never supplied. The red team and source code review team for Hart worked together and created their own upgrade program that performed suitably for the purposes of testing, but the time they spent doing this could have been spent analyzing other aspects of the system.

---

[3] The Diebold system was set up for use by the testers on June 14, the Sequoia system on June 19, and the Hart system on June 22. The testers were able to photograph the Sequoia system on June 14, but the system was not yet set up for use.

[4] See the reports of the document review teams for details of missing documents and documents that arrived after July 12.

[5] More specifically, if the firmware images were digitally signed, an attacker could not "reflash" (i.e., install) new firmware without having access to the private key. However, if the firmware images were *not* digitally signed, all one would need is access to the system to reflash the firmware and compromise the system—a considerably easier task.

Despite these problems, the red team testing was successful, in that it provided results that are reproducible and speak to the vulnerability of all three systems tested.

5.0 Example Threats

An election system consists of three components: the ballot preparation system, the voting mechanisms, and the tallying systems. The red teams were given sample elections and used them in the elections that they tried to subvert.

The voting mechanisms are either Direct Recording Electronic machines (DREs) with Voter Verified Paper Audit Trails (VVPATs) or optical scan systems. They each store the votes that voters cast in various ways. If they can be compromised, the votes stored on those systems may not reflect the votes actually cast by the voters.

As an example, the ability to execute arbitrary programs on one of these systems can cause votes to be misrecorded even when there is a VVPAT. The specific attack relies on the belief that many voters will not check the VVPAT. An attacker creates a new version of the firmware that will misrecord a vote. The incorrect vote will be printed on the VVPAT. If the voter notices and declines to cast the vote by returning to an earlier screen, the malicious firmware will then record the vote correctly. Thus, there will be no discrepancy between the votes as recorded on the VVPAT and on the electronic media.

Even if there is a discrepancy between the VVPAT and the electronically recorded votes, that discrepancy must be discovered. Typically, this would occur during the 1% audit or a recount. For our purposes, we considered such a discrepancy a valid attack, because the way in which such a discrepancy is to be handled is unclear, especially when the VVPAT is damaged or hard to read.

The election management system consists of software running on a commercial platform. Typically this platform is some form of Microsoft® Windows. The application software consists of a database program and other software. A client program, the database application, or both control access to the election data. This platform may also be used to initialize memory cards or other media to transfer information to the voting machines. The platform may also contain other programs not supplied by the voting system vendor.

For example, all three vendors' election management software runs on platforms with the Windows operating system. The configuration of the Windows system provides a layer of protection against an attacker compromising the software. The strength of this layer depends directly on the security of the underlying operating system. As Windows is known to be vulnerable to many forms of attack, vendors should ensure that the underlying Windows system is locked down sufficiently[6] to counter these threats.

If an attacker can gain privileged access to the underlying operating system, they can control the election management system. This is why election management systems

[6] For example, one step in this procedure would be to disable all unnecessary services.

should be locked down tightly and be kept in a physically secure area: so that attackers have limited to no access to the system. As noted above, physical access is simply one layer of security defense. Minimizing privileges and taking other basic precautions in configuring the underlying operating system provide additional layers.

As an example, if an attacker can insert an untrusted medium (like a U3 USB memory stick) containing a malicious program called a "Trojan horse", and the system is configured so that autorun is turned on, the Trojan horse can be loaded into memory. At that point, it can detect the insertion and removal of media, including media intended to load information onto the voting machine. It can load malicious firmware onto that media. It can modify any local files, and completely control the underlying system.

The results presented in the next section show that the above attacks, and many like them, can be realized.

6.0 Results and Interpretations

This section presents a very high-level overview of the test results, and their technical interpretation. It is up to the Secretary of State, and other election officials, to interpret these findings in light of the relevant laws, regulations, policies, and procedures.

The results are documented in three reports, one for each system. Each report consists of a public portion and a confidential portion. Our goal in the public portion is to provide information about the vulnerabilities of the systems to the public to allow intelligent and reasoned discourse on the effects of those vulnerabilities and on the role of these electronic voting systems in the California election process without providing a step-by-step guide to attacking the systems, and without revealing the vendors' proprietary information. The confidential portion details the attacks that were successful, discusses those attacks that were tried but that failed, and also provides suggestions for attacks that we were unable to try. We hope, and intend, that this material provide guidance to future testers.

We request that the Secretary of State provide the public and confidential reports to the respective vendors. We believe the vendors want to close any vulnerabilities found. We believe that our reports can help them identify those vulnerabilities, how they might be exploited, and how they might be mitigated or eliminated. With their intimate knowledge of their systems, this should be enough to enable them to determine, and take, appropriate corrective action.

6.1 Sequoia

The red team analyzing the Sequoia system identified several issues. They fall into several classes:

1. **Physical Security**. The testers were able to gain access to the internals of the systems by, for example, unscrewing screws to bypass locks. The screws were not protected by seals. Similarly, plastic covers that were protected by seals could be pried open enough to insert tools that could manipulate the protected buttons without damaging

the seals or leaving any evidence that the security of the system had been compromised.

2. **Overwriting Firmware**. The testers discovered numerous ways to overwrite the firmware of the Sequoia Edge system, using (for example) malformed font files and doctored update cartridges. The general approach was to write a program into memory and use that to write the corrupt firmware onto disk. At the next reboot, the boot loader loaded the malicious firmware. At this point, the attackers controlled the machine, and could manipulate the results of the election. No source code access was required or used for this attack, and a feature of the proprietary operating system on the Edge made the attack easier than if a commercial operating system had been used.
3. **Overwriting the Boot Loader**. Just as the testers could overwrite firmware on the disk, they could overwrite the boot loader and replace it with a malicious boot loader. This program could then corrupt anything it loaded, including previously uncorrupted firmware.
4. **Detecting Election Mode**. The firmware can determine whether the system is in test mode (LAT) or not. This means malicious firmware can respond correctly to the pre-election testing and incorrectly to the voters on Election Day.
5. **Election Management System**. The testers were able to bypass the Sequoia WinEDS client controlling access to the election database, and access the database directly. They were able to execute system commands on the host computer with access only to the database. Further, the testers were able to exploit the use of the autorun feature to insert a malicious program onto the system running the Sequoia WinEDS client; this program would be able to detect the insertion of an election cartridge and configure it to launch the above attacks when inserted into an Edge.
6. **Presence of an Interpreter**. A shell-like scripting language interpreted by the Edge includes commands that set the protective counter, the machine's serial number, modify the firmware, and modify the audit trail.
7. **Forging materials**. Both the update cartridges and voter cards could be forged.

The report presents several scenarios in which these weaknesses could be exploited to affect the correct recording, reporting, and tallying of votes.

6.2 Diebold

The team investigating the Diebold system identified several issues. They fall into several classes:

1. **Election Management System**. The testers were able to penetrate the GEMS server system by exploiting vulnerabilities in the Windows operating system as delivered and installed by Diebold. Once this access was obtained, they were able to bypass the GEMS server to access the data directly. Further, the testers were able to take security-related actions that the GEMS server did not record in its audit logs. Finally, with this level of access, the testers were able to manipulate several components networked to the GEMS server, including loading wireless drivers onto the GEMS server that could then be used to access a wireless device plugged surreptitiously into the back of the GEMS server.

2. **Physical Security**. The testers were able to bypass the physical controls on the AccuVote Optical Scanner using ordinary objects. The attack caused the AV-OS unit to close the polls, meaning the machine could not tally ballots at the precinct or inform voters whether they had "over-voted" their ballot. Similarly, the testers were able to compromise the AccuVote TSx completely by bypassing the locks and other aspects of physical security using ordinary objects. They found an attack that will disable the printer used to produce the VVPAT in such a way that no reminders to check the printed record will be issued to voters.
3. **AccuVote TSx**. The testers found numerous ways to overwrite the firmware in the AccuVote TSx. These attacks could change vote totals, among other results. The testers were able to escalate privileges from those of a voter to those of a poll worker or central count administrator. This enabled them to reset an election, issue unauthorized voter cards, and close polls. No knowledge of the security keys was needed.
4. **Security Keys for Cryptography**. The testers discovered that a well-known static security key was used by default.

The report presents several scenarios in which these weaknesses could be exploited to affect the correct recording, reporting, and tallying of votes.

6.3 Hart

The team investigating the Hart system identified several issues. They fall into several classes:

1. **Election Management System**. The testers did not test the Windows systems on which the Hart election management software was installed because Hart does not configure the operating system or provide a default configuration. Hart software security settings provide a restricted, Hart-defined environment that the testers bypassed, allowing them to run the Hart software in a standard Windows environment. They also found an undisclosed account on the Hart software that an attacker who penetrated the host operating system could exploit to gain unauthorized access to the Hart election management database.
2. **eScan**. The testers were able to overwrite the eScan firmware. The team also accessed menus that should have been locked with passwords. Other attacks allowed the team to alter vote totals; these attacks used ordinary objects. The team, in cooperation with the source code review team, was able to issue administrative commands to the eScan.
3. **JBC**. The team developed a surreptitious device that caused the JBC to authorize access codes without poll worker intervention. The team verified that the mobile ballot box (MBB) card can be altered during an election. The team also found that post-election safeguards to prevent the altered data on a tampered MBB card from being counted can be easily bypassed.
4. **eSlate**. The testers were able to remotely capture the audio from a voting session on an eSlate with audio enabled, thereby providing an attack that violates voter privacy. The team was also able to force an eSlate to produce multiple barcodes after printing

"BALLOT ACCEPTED" on the VVPAT records. This could cause a county that used bar code readers to read the VVPAT to produce erroneous vote totals.

The report presents several scenarios in which these weaknesses could be exploited to affect the correct recording, reporting, and tallying of votes.

6.4 Discussion

The red teams demonstrated that the security mechanisms provided for all systems analyzed were inadequate to ensure accuracy and integrity of the election results and of the systems that provide those results.

Electronic voting systems are critical to the successful conduct of elections in those jurisdictions where they are used. Given the importance of voting and elections in the governing of the State of California, one may safely say that these systems are "mission critical". Such systems need to be of the highest assurance in order to ensure they perform as required. Techniques for developing such systems are well known[7] but, sadly, not widely used. Vendors would do well to adopt them for electronic voting systems.

Similarly, many components of voting systems run on commercial operating systems. A non-secure underlying operating system offers attackers avenues into the software that the operating system runs, in this case the vendors' election management systems. Hence vendors must ensure that whatever underlying operating system their software runs on meets the security requirements that their software meets.

A key idea underlying high assurance techniques is that *security should be part of the design and implementation of the system and not added on "after the fact"*. The reasons for this need not be repeated here. Many of the components tested appear to have been hardened by taking their basic design and adding security features. As a result, the testers were able to exploit inconsistencies between the protective mechanisms and that which they were intended to protect.

Vendors should assume the components of the voting system will be used in untrusted environments in which they cannot be adequately monitored. Thus, their physical protections should be "hardened" to withstand determined attack. The added barrier that such mechanisms create will hamper the ability of attackers to obtain illicit access to the components even if lapses in procedural mechanisms allow them unobserved or unfettered access to the systems.

Of equal importance is the ability to detect when such attacks occur. Again, this speaks to security mechanisms as being "layered"; one must implement mechanisms to prevent compromise, and then add mechanisms (which may be the same as the previous ones) to enable observers to detect compromise should the preventative mechanisms fail.

[7] See for example Elisabeth Sullivan's excellent discussion in [9], chapters 18 and 19.

Because detection requires that people take some action, the security mechanisms require that specific procedures be designed in order to ensure that failure of the preventative mechanisms, and success of the detection mechanisms, are properly handled. An excellent example comes from the realm of physical security. A common belief is that tamperproof tape is sufficient to detect the violation of preventative mechanisms; for example, sealing a bay with tamperproof tape enables one to detect that the bay has been opened. Two problems arise. First, there must be a procedure to check the tamperproof tape. Second, an attacker can often acquire the same tape as is used to protect the systems. The attacker simply removes the tape showing evidence of the tampering, and replaces it with her own tape. Unless the original tamperproof tape has unique serial numbers and the observers check those serial numbers, the detection mechanism is defeated. Unless the customers follow an appropriate procedure (here, checking that the tape is intact and the intact tape has the right serial numbers), the security mechanism is easily defeated.

Finally, the red teams wish again to emphasize the inadequacy of "security through obscurity" as a key defensive mechanism. No security mechanism should ever depend on secrecy. At best, secrecy should be a single security mechanism in a layer of defensive security mechanisms. In this study, when vendors failed to provide software that would have helped the red teams expedite the testing process, the failure became a *motivation* for the red teams to construct equivalent software to carry out the attacks. The only thing lost was time that could have been used for testing. Given the constraints under which the red teams operated, a well-financed team of attackers, with plenty of time to plan attacks between elections, could do considerably better.

7.0 Conclusion

Neither this report nor the individual system public reports count the successful, unsuccessful, and untried attacks. The reason for this is that a comparison of the systems based on raw counts from this study is at best meaningless and at worse misleading. First, judging the vulnerability of a system requires understanding both the nature and the implementation of the policies and procedures under which it is used. A system that has 10 vulnerabilities that can be remediated by proper, realistic procedures can meet a set of requirements better than a system with only one vulnerability that cannot be remediated by realistic procedures. As the red teams ignored compensating controls and mitigations, the raw counts of successful, unsuccessful, and untried attacks do not indicate which would still be successful in the face of compensating controls—and how realistic those compensating controls would be.

Nor does this report characterize the difficulty, or lack thereof, of carrying out the successful attacks. This question has two parts. The first is how difficult it was to develop the attack mechanisms and (when needed) software. The second is how difficult it would be to carry out the actual attack, given the mechanisms and software developed. Consider an attack that replaces the firmware of a voting system with firmware that is malicious. Developing the malicious firmware, and building the software mechanism to install it, requires an expert or team of experts. But carrying out the attack requires only access to a voting system (i.e., someone voting) and not technical expertise. Further, the precise

procedural controls in place affect the difficulty of both phases of the attack, and the red teams focused only on the technical aspects of the systems.

As mentioned earlier, the red teams encountered difficulties in acquiring information and tools that would have allowed the study to go faster, and explore more potential threats and attacks. For the future, we suggest the Secretary of State adopt regulations to make the delivery of documents, software, and other material mandatory *before* certification. Then from the beginning of the review, the testers and reviewers will have all material at hand.

Perhaps wording similar to the following would accomplish this goal:

> "Source code materials shall be submitted to an approved escrow company for placement in the escrow facility. The contents of source code materials shall be in a form, and include the source code, tools and documentation, to allow the complete and successful compilation and installation of a system in its production/operational environment with confirmation by a verification test by qualified personnel using only this content."

The intent is that the vendor should provide everything necessary for the testers to build the system from the escrowed materials, and that those materials include the source code. This would have, for example, required Hart to escrow the uploading program that the red and source code review teams reconstructed, and allowed the teams more time to test potential attacks and problems using that tool.

8.0 Acknowledgments

Many people assisted us on this study. We wish to thank Jason Heyes, Ryan Macias, and Miguel Castillo of the Office of the California Secretary of State for putting up with the odd hours that we worked, and taking care of the electronic voting systems; Chris Maio for helping us set up some equipment; Debbie O'Donoghue and Lowell Finley of the Office of the California Secretary of State for administrative support and, especially, for helping us communicate with the vendors; and all the members of the accessibility review, source code review, and document review teams. Throughout this project, the other teams provided information, suggested possible attacks, helped us analyze results, and in some cases came up to Sacramento to help us analyze systems and try different attacks. Finally, we thank David Wagner for his critical support throughout this project. This was truly a group effort, and it is a pleasure to acknowledge all involved.

9.0 References

[1] *Master Agreement 06158101 between the Secretary of State and the Regents of the University of California*, May 4, 2007

[2] *COBIT 4.1: Framework, Control Objectives, Management Guidelines, Maturity Models*, IT Governance Institute, Rolling Meadows, IL (2007).

[3] *ISO 9001:2000, Quality Management Systems—Requirements*, International Standards Organization, Geneva, Switzerland (2000).

[4] *Capability Maturity Model® Integration (CMMI^{SM})*, Version 1.1, Technical Report CMU/SEI-2002-TR-012, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 15213 (Mar. 2002).
[5] Office of Government Commerce, *Managing Successful Projects with PRINCE2*, The Stationary Office, London, UK (May 2005).
[6] *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*, Third Edition, Project Management Institute, Newtown Square, PA (2004)
[7] *Minimum Security Requirements for Federal Information and Information Systems*, FIPS PUB 200, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899 (Mar. 2006).
[8] "Government Agencies Posting Sensitive 'Need to Know' Material Online", Fox News (July 12, 2007); available at http://www.foxnews.com/story/0,2933,289011,00.html.
[9] M. Bishop, *Computer Security: Art and Science*, Addison-Wesley Professional, Boston, MA (2003).
[10] "Voting System Standards", Federal Election Commission, Washington DC (2002); available at http://www.eac.gov/election_resources/vss.html.
[11] *Approval of Use of Diebold Election Systems, Inc. DRE & Optical Scan Voting System*, Office of the California Secretary of State, Sacramento, CA (2006); available at http://sos.ca.gov/elections/voting_systems/diebold_cert.pdf.
[12] *Approval of Use of Sequoia Voting Systems, Inc. DRE & Optical Scan Voting System*, Office of the California Secretary of State, Sacramento, CA (2006); available at http://sos.ca.gov/elections/voting_systems/sequoia_cert.pdf.
[13] *Approval of Use of Hart InterCivic System 6.2.1 DRE & Optical Scan Voting System*, Office of the California Secretary of State, Sacramento, CA (2006); available at http://sos.ca.gov/elections/voting_systems/2006-09-22_System_6_2_1.pdf.
[14] E. Barr, M. Bishop, and M. Gondree, "Fixing Federal E-Voting Standards," *Communications of the ACM* **50**(3) pp. 19–24 (Mar. 2007).
[15] *Voluntary Voting System Guidelines*, Election Assistance Commission, Washington DC (2005); available at http://www.eac.gov/vvsg_intro.htm.
[16] "Citing Problems, U.S. Bars Lab from Testing Electronic Voting", *New York Times* p. 1 (Jan. 4, 2007).
[17] D. Kawamoto, "SEC Filing Acknowledges 'Pretexting' in HP Board Probe," *CNET News*.com (Sep. 6, 2006); available at http://news.com.com/SEC+filing+acknowledges+pretexting+in+HP+board+probe/2100-1014_3-6112710.html.
[18] M. Bishop, *Protocol for Red Team Testing*. Available at http://www.sos.ca.gov/elections/voting_systems/ttbr/red_team_protocol.pdf.