Voting System Security Review

Hart InterCivic eSlate
Diebold TSx/GEMS
AutoMARK/ES&S 100

An Evaluation

Prepared for
The Secretary of the Commonwealth of Massachusetts

by

Michael Ian Shamos, Ph.D., J.D.
September 28, 2006

Summary

This report contains the findings of a consultant engaged by the Secretary of the Commonwealth of Massachusetts to examine the security aspects of three electronic voting systems intended for use by disabled voters.

For the reasons given in detail in this report, under the administrative procedures recommended herein, all three systems are sufficiently secure for use.

I. Introduction

Massachusetts conducts voting primarily on optical scan voting equipment. Section 301(a)(3) of the Help America Vote Act (HAVA) (42 U.S.C. §15472) requires that "The voting system shall — (A) be accessible for individuals with disabilities, including nonvisual accessibility for the blind and visually impaired, in a manner that provides the same opportunity for access and participation (including privacy and independence) as for other voters; (B) satisfy the requirement of subparagraph (A) through the use of at least one direct recording electronic voting system or other voting system equipped for individuals with disabilities at each polling place." Because this provision takes effect in 2006, Massachusetts is required to provide at least disabled-accessible voting system for each precinct. Ordinary optical scan systems are not compliant because a voter with visual disabilities is not able to mark an optical scan ballot without assistance.

Before an electronic voting system can be used in Massachusetts, it must be approved by the Secretary of the Commonwealth. 54 M.G.L. §32. The Massachusetts statutes are not specific about security requirements for voting systems. Accordingly, the Secretary has provided regulations relating to approval of such systems. 950 CMR §50.02(2) states that, "Equipment shall be designed so as to maximize accuracy and prevent fraud." The emphasis in this report is on prevention and detection of fraud.

Because the security of electronic voting systems has been questioned in various forums and published reports, the Secretary of the Commonwealth of Massachusetts, its chief election officer, engaged me as a consultant to perform an independent security review of three proposed systems, the Hart InterCivic ("Hart") eSlate, Election Systems & Software ("ES&S") AutoMARK and Diebold Election Systems ("Diebold") TSx with GEMS. The vendors furnished documentation of their systems in advance and appeared with their equipment for review at the offices of the Secretary of the Commonwealth. Hart was reviewed on August 2, 2006, ES&S on August 3, 2006 and Diebold on August 7, 2006. The reviewing process was recorded on videotape and transferred to DVD. The reviews were confined to security matters and did not concern compliance with other aspects of Massachusetts law. The reviews did not constitute certification examinations.

In conducting the reviews I have considered the risks and scenarios presented in various recent published reports, including, the CRS Report[1], the Compuware Report[2], the Carrier article[3], GAO-05-956[4], the Hart Ohio Security Assessment[5], the California

[1] "Election Reform and Electronic Voting Systems (DREs): Analysis of Security Issues." Congressional Research Service (Nov. 4, 2003).

[2] "Direct Recording Electronic (DRE) Technical Security Assessment Report," Compuware, Inc. (Nov. 21, 2003), commissioned by the Ohio Secretary of State.

[3] Michael A. Carrier, "Vote Counting, Technology, and Unintended Consequences," 79 St. John's L. Rev. 645 (2005).

[4] "Elections: Federal Efforts to Improve Security and Reliability of Electronic Voting Systems Are Under Way, but Key Activities Need to be Completed." (Sept. 2005).

Secretary of State's Diebold staff report[6], the Common Cause Report[7], the California Consultant's Reports (eSlate)[8], and the Brennan Center Report[9]. I have also read and considered the documents listed in Appendix A concerning the systems under review. Because these reports have provoked fear and misunderstanding even among the educated public, it is necessary to deal with all of their allegations head-on.

It makes no sense to brand a particular voting system (or any type of system, for that matter) as "secure" or "not secure." Security is meaningful only with respect to a fully articulated catalog of threats. Once the threats and countermeasures are enumerated, determining whether a system is sufficiently secure against those threats becomes a matter of risk assessment. Different states may choose to assign differing probabilities and downsides to various successful threats. The probabilities are never zero, though they may be negligibly small. To insist that a voting system reduce the probability of success of a set of threats to zero is to rule out the use of voting systems entirely, as no system ever built by man has been entirely impervious to intrusion.

The Brennan Center Report is fairly thorough in detailing not only attack modes but "points of attack," namely events or places in the process at which an intruder might be able to gain access to, or introduce malware into, a voting system.

The voting system security threats considered in this report fall into these categories:

- **Machine failure.** Failure of a voting machine that might result in misrecording of votes or the loss of votes already cast. There is great voter concern when a voting machine ceases to operate during the election because of the fear that votes already recorded on the machine might be lost or altered as a result of the failure. The VVPAT mechanism is a safeguard against that type of failure since the paper record exists for the votes previously cast. In the case where the machine begins misrecording subsequent to the failure, the VVPAT is effective only if voters actually review the VVPAT for correctness.
- **Software errors.** Bugs in any component of the software used to set up and conduct an election that might result in misrecording or mistabulation of votes. Software errors are contrasted with "malware," below.
- **Malware.** As used in the report, "malware" means software (or firmware) that has been deliberately created or modified to perform in a manner different from its documented function, and includes other pieces of software (and

[5] "Technical Security Reassessment Report: Hart InterCivic Direct Recording Electronic (DRE) Device," Compuware Corp., Sept. 16, 2005, commissioned by the Secretary of State of Ohio, labeled as confidential but freely available over the Internet.

[6] "Diebold Election Systems, Inc. … [long list of system components] … Staff Review and Analysis" (Nov. 14, 2005).

[7] "Election Reform: Malfunction and Malfeasance – A Report on the Voting Machine Debacle." (2006)

[8] "California Secretary of State Consultant's Report on Hart Intercivic," by Paul Craft (Feb. 25 2006), and "California Secretary of State Consultant's Report on Hart Intercivic System 6.2," by Paul W. Craft and Kathleen A. McGregor (Aug. 4, 2006).

[9] "The Machinery of Democracy: Protecting Elections in an Electronic World," The Brennan Center for Justice (2006).

firmware), such as viruses and Trojans, that modify election software or cause it to perform in a manner other than its documented behavior. In this regard, malware is understood differently from software errors, which are not deliberate. Malware and its creator may take conscious steps to conceal the malware and thus evade or reduce the probability of detection. The creation or insertion of malware might conceivably occur at many different stages, e.g. at the vendor (known or unknown to vendor's management), at the ITA, in the warehouse, at the jurisdiction, at the polling place, etc., and we must evaluate what, if anything, the system does to resist or reveal such intrusions. For example, if someone attempted to patch an .exe file, would that exploit be detected?

- **Calibration errors.** Touchscreens and optical scanners must be calibrated. The touchscreen must be set to recognize properly the physical location of a touch. Poor touchscreen calibration may lead to a touch for one candidate being mistaken as a touch for a different candidate. An optical scanner must be able to set to recognize marks in specific places and at certain light intensities. Poor calibration can lead to misreading of marks, hence counting of votes for the wrong candidate.
- **Tampering with ballot setup information.** For each type of tampering attack, it must be considered separately whether attack could be performed by insiders (that is, parties with special access privileges, such as the original vendor, maintenance personnel, the jurisdiction's IT director, poll workers, etc.), and whether it could be mounted by knowledgeable outsiders (such as hackers or voters).
- **Tampering with uncast ballots.** An election outcome can be affected by various forms of tampering that fall short of modifying software. For example, if the list of candidates presented to the voter is incorrect or incomplete, the voter is not given a meaningful choice. While steps are taken to ensure that slates are complete and correct, it is possible that the slates may be modified after such checking but before the election.
- **Tampering with cast ballots.** Most electronic voting machines make internal electronic records on non-volatile memory of ballot images of votes cast by voters. These are generically referred to as Cast Vote Records (CVRs). Some systems do not record vote totals, but compute them fresh each time they are requested by processing all of the CVRs. This means that if it were possible to alter the CVRs after they had been recorded, the totals would be affected. A VVPAT is a CVR on a physical medium. Clearly having a VVPAT increases the probability that alteration to just the electronic CVRs would be detected. However, this is true only if something is actually done with the VVPAT other than to store it away in a container.
- **Tampering with vote totals.** Vote totals produced by individual voting machines are printed out at the close of polls and signed by the appropriate poll officials. Copies are posted at the polling place and also sent to the jurisdiction for tabulation. In most cases these become the official record of the election, and the original signed documents are used in the canvass to determine the winner of the election. Reporting that proceeds on election

night is generally unofficial only and is there to provide the press and public with a quick assessment of who has won. However, the election night totals are never final. Generally absentee, military and provisional ballots have yet to be counted, so the election night results are unofficial in the sense that that they are not used to determine winners. Nevertheless, the public becomes concerned when the official and unofficial results differ, especially as to outcome. Therefore it is necessary to prevent even unofficial vote totals from being manipulated.

- **Attacks directed at assistive mechanisms.** These are attacks that depend on the fact that disabled voters are often unable to take advantage of various protective mechanisms afforded to voters without disabilities. For example, unsighted voters are unable to read the voter-verified paper trail or the touchscreen display. Therefore, a potential attack would be to print a false paper trail for any voter who is using an audio ballot. The audio information would be played properly, but the vote would be recorded incorrectly in the machine and a corresponding false paper trail could be written, which the voter would be unable to verify. If the voter attempted to verify the ballot through audio means, no discrepancy would be observed.
- **Attacks on auditing mechanisms.** Most voting systems accumulate administrative data that can later be used to pinpoint election irregularities. These include event logs, timestamps indicating each time the machine was activated for voting, maintenance logs and logs of manual changes to election data. Some types of tampering can readily be detected if the log records are maintained faithfully. Therefore, the success of certain attacks depends on the attacker being able to modify or erase evidence of his attack.
- **Privacy leaks.** Determining how voters voted through electronic emissions or other inferences from data provided by the election system. For example, the sequential voter-verified paper trail offered by Diebold and ES&S has the potential to expose the vote of every voter at a polling place, and active measures must be taken to ensure that it will not be possible after an election to reconstruct the order in which voters cast their ballots. Otherwise, that ordering could be matched up against the paper trail to determine each voter's choices.
- **Denial of service attacks.** Efforts to stall or disable voting entirely at selected polling places. Examples include physical attacks on the voting machines to render them inoperative, or trapdoors in the voting software to cause the machines to stop working at a particular time, or software that accepts data from a voter having inside knowledge, whose result is to halt the machine.

Test mode

All voting system offer a variety of test modes in which system functions can be tested and verified without casting official ballots. The role of these test modes is often misunderstood. Test mode never serves to defend against malware. Obviously, if someone has modified election software and wants to avoid getting caught, he will ensure that the system works properly when in test node, but not in election mode. Thus the purpose of test mode cannot be to detect the presence of altered software. Test mode is offered to verify ballot setup and to ensure that normal system functions are operational. Testing for malware must be performed in a different way.

Logic and Accuracy Testing has great value in uncovering errors, as opposed to malware, however, and must be fully utilized, particularly to verify accessible ballots. It can be effective in locating irregularities inserted by malware in ballot setup software, the effective of which is to corrupt static files (as opposed to executables.) The reason it is effective is that these static files cannot be selectively enabled or disabled during an election unless the system firmware has been tampered with. Therefore, there is no way for the system to behave differently in LAT than it will during an election if only static files are altered.

Daisy chaining

Daisy chaining is the act of connecting multiple voting machines, typically in serial fashion, via a single bus cable that passes through all of them. This is done in some cases to provide electric power to all units without the need for multiple wall outlets or power strips. It may also be done to allow data to be routed from each machine to some central device for accumulation or tabulation. The practice has been decried by some security commentators on the grounds that (1) it increases the risk of manipulation of multiple machines from a single place; (2) it presents a privacy risk if vote records are transmitted across a wire since a different voting machine might record the data and/or an eavesdropper might be able to pick up inductive signals from the wire; and (3) it fosters the suspicion that different voting machines might be engaging in unsafe activities made possible through communication. It is true that the first and last of these would be eliminated by outlawing daisy chaining, but this hardly seems warranted in the case of VVPAT systems or ones that can be parallel tested.

Election definition

Voting machines cannot present a slate to the voter unless they are informed of all the offices and candidates. Generating all the ballot styles needed for all polling places in a jurisdiction is known as election definition or ballot setup. It is also often referred to by the misnomer “ballot programming,” which is incorrect since no computer programming is involved. Ballot definition involves setting up geographic boundaries for a jurisdiction, defining precincts, and then listing all candidates and issues on which voters in each precinct are entitled to vote. It is a laborious process sometimes performed by vendors under contract to the jurisdiction. The fact that employees of private corporations have a role in the conduct of elections has given rise to the fear that the corporations control U.S. elections in some way.

The charge is unfounded. Because election setup involves only static data (and not computer programs), and the static data is completely proofread and verified during Logic and Accuracy Testing (LAT), which is a public event at which representatives of political parties verify that all candidates are present and in their proper positions on the ballot.

Voting System Testing

Voting systems are tested by the vendor, by the Independent Testing authority, by the state during certification, by hired consultants, by the jurisdiction at acceptance, by warehouse employees before each election, and at pre-election LAT. They may also be tested during the election by a method called Parallel Testing (discussed later) and post-election Logic and Accuracy Testing. In the even of a claim of irregularity, the machines can be subjected to forensic examination after the fact.

Some of the tests (such as ITA testing) are intended to detect malware inserted by a party other than the tester. It is of course possible to hypothesize that all persons ever involved in the testing process at whatever level were colluding to conceal malware. There is no technological response to such an allegation. Even making the system source code public would not serve to put the charge to rest[10]. Therefore, each jurisdiction must decide for itself whether it is willing to rely on the administrative separation of responsibility to guard against such collusion.

COTS Software

Most voting systems depend on or make use of commercial off-the-shelf (COTS) software. For example, the Windows operating system is COTS, as is often the Basic Input-Output System (BIOS) of a computer, programs for viewing documents, such as Adobe Acrobat Reader, etc. COTS software under Microsoft Windows, Diebold does not own or control the Windows source code[11]. As a general matter under the 2002 FEC standards, COTS is exempt from ITA source code review. Part of the logic behind this policy is the unavailability of the source to the ITA, but also the view that if COTS software is truly off-the-shelf, then it will not contain any malware specifically directed to voting system generally, and particularly not toward any specific voting system, which may change at frequent intervals. Whether these assumptions behind COTS software are realistic is again part of the risk assessment process.

The fact that there may not be a source code review of COTS software does not mean that it never gets tested or evaluated. The system is stress-tested at ITA with a huge number of ballots, is tested at certification and, under ideal conditions, is tested in parallel during the election (to defeat time-sensitive code that causes the system to behave properly at all times except when a real election is in progress.) Whether such testing is sufficient is also part of risk assessment.

[10] It may be alleged, for example, that the object code actually used in the machine may not correspond to the publicly released source code.

[11] The situation is somewhat different for Windows CE, which is used in the Diebold TSx touchscreen machine. In that case the COTS is actually customized based on input from Diebold for the particular platform on which it runs. This issue is discussed later in connection with the TSx review.

COTS Hardware

All voting systems depend on or make use of commercial off-the-shelf (COTS) hardware. For example, Window-based software runs on IBM-compatible PCs using standard, commercially available processors. It is hypothetically possible that an Intel engineer responsible for the design of the Pentium IV chip has inserted rogue components designed to interfere only with elections. The fact that I personally find such a prospect to be ludicrous is not sufficient reason to ignore it. The question is, if such an act occurred, would it be detected. The answer is yes because of the VVPAT and parallel testing, both discussed below.

VVPAT

The Voter-Verified Paper Audit Trail provides the voter with evidence that the machine has correctly understood and recorded the voter's choices. If the VVPAT is used in a recount, the theory is that any flaw, intentional or otherwise, in the software will not interfere with the jurisdiction's ability to count the votes as they were cast by the voters. The success of the VVPAT, assuming that it complies with other election law requirements, depends on voters actually verifying it, maintaining a complete and reliable chain of custody over the paper records, and providing a reliable means of counting the VVPAT should it be necessary. When properly deployed, the VVPAT serves as a check against many tampering threats that have been articulated in the literature. A lot must be read into the phrase "properly deployed." If it is assumed that visually impaired voters cannot check the paper trail, then a mode of attack is to record the ballots of those voters incorrectly both in the electronic record and the paper record (while playing the "correct" names to voter via audio), and relying on the fact that the alteration will not be caught since the records cannot be checked. Thus the VVPAT does not provide the same protection to disabled voters as it does to regular voters.

I believe this to be a distinction without a difference, since whether the VVPAT is operating properly can be determined by testing, and, if it is working, it can be relied on by voters even without verification. However, the scenario has been proposed that the electronic records might be altered only on Election Day, and thus would evade any testing performed before or after the election. This problem is addressed by parallel testing, below.

Parallel Testing

Parallel testing means testing a voting system on Election Day, at the same time that real voting is taking place. The main purpose of parallel testing is to detect malware that takes effect only during the election but is dormant at all time before and after the election. Such malware is hypothetically possible on machines having an onboard clock. The resident software, by interrogating the clock, can determine whether a real election is in progress. If a machine is tested during the election, the malware is unable to determine that the machine is really under test, and so the effects of the malware will be observed.

A problem is that on most systems it is not possible to cast test ballots on a voting machine during the election, since the test ballots would be counted as regular ballots.

Parallel testing in its ideal form consists in having officials appear unannounced at randomly selected polling places and sequestering in each one a voting machine that will not be used for real voters, but will be used by test volunteers who will cast votes on the machines all day long from randomly-generated scripts while being videotaped. The results that should be produced by the scripts are known, and the results actually obtained by the machine can be compared with the known results. The most likely cause of any discrepancy is mechanical error by the volunteer, which can be caught and corrected from the video tape. Any remaining discrepancy reveals some problem with the voting system, which may result from malfunction, software error or malware.

It is critical in conducting parallel testing to treat the machine being tested exactly the same way as normal voting machines. That is, the machine should not be moved after being turned on, the behavior and tempo of volunteer voters must match that of regular voters, all procedures for activating the machines, inserting voter cards, closing the polls, etc., must be performed in the same manner. Otherwise, the process is open to the criticism that clever malware could have detected the fact that the system was being parallel tested. Whether such malware can exist is matter for risk assessment. Since know one knows how voters behave in practice (because of a gross lack of data), it is extremely unlikely that anyone could build software to mimic or recognize that behavior, much less maintain a demographic database hidden in the system of how voters in the 200,000 polling places in the United States behave, but once again this a matter for each jurisdiction’s risk assessment.

Malware that operates by switching votes from one candidate to another, if present on the machine being parallel tested, will be detected. On pure ballot marking systems, such as AutoMark, since the marking machine performs no tabulation, parallel testing is easy. All one need do is, at several random times on Election Day, have a pair of pollworkers mark ballots on the machine. The testers can verify that the correct slate of candidates was presented and that the ballots are marked properly. These test ballots can be marked “TEST” or “VOID” and treated as spoiled ballots without affecting the outcome of the election.

For DRE systems, the question in parallel testing is how many machines need be tested in an effective parallel test. The answer depends on the resumed threat model. If it is believed that malware has been inserted at the vendor and is therefore present in every machine in the state, it is sufficient to test one machine to reveal the exploit. If it is believed that only one machine in the entire state has been tampered with, then parallel testing would only guarantee to uncover the intrusion if every device in the state were tested. This is an impossibility, since we must allocate sufficient machines for voting, and these cannot be subjected to parallel testing.

Software/Firmware Upgrades

Even assuming that a voting system is sufficiently secure for elections, the system must be upgraded periodically. That is, components of its software and firmware must be replaced by new versions. This is necessary because of COTS operating system changes, after-discovered security vulnerabilities, bugs, new features and changes to election law. A good example is the HAVA requirement to add assistive interfaces by 2006. This could not be done without upgrading both software and firmware. However, what controls are there to prevent the upgrade process from being used to introduce unauthorized code or malware?

Unfortunately, the process of performing a legitimate upgrade must be efficient. If it took just one hour to upgrade a voting machine, then about 50 man-years of effort would be needed to install one upgrade nationwide[12]. Since machines are typically upgraded three times per year, the process would be prohibitively slow and expensive. The question is how to perform authorized upgrades quickly without opening a security hole. In general, this is done by connecting a laptop containing the new software/firmware to a voting machine (or inserting removable media into the machine) and initiating an authorization process requiring credentials, then invoking the upgrade mechanism. In a proper system, these activities are logged electronically and digitally signed to prevent alteration of the audit log. Even so, how are we to know that the upgraded software is properly certified and is not simply a Trojan?

Some jurisdictions distribute authorized upgrades by making copies from a standard release sent by the ITA. However, this procedure just results in relocating points of trust. We must rely on the person performing the installation to use the official release, and the upgrade should be witnessed by disinterested observes. Digitally signing the release media is effective if the voting machine has not been corrupted to eliminate verification of digital signatures. Checking MD5 hashes against the National Software Reference Library would be effective if there were a trustworthy way to obtain the hashes from the installation media and, preferably, from the voting machine after installation has been performed. As a general matter, control of voting system software upgrades is weak from the security viewpoint. This means that indirect verification methods, such as parallel testing, rise in importance as a means of verifying upgrades.

Direct verification, such as by reliable export of software/firmware for independent checking, would reduce dependence on indirect methods. However, the software/firmware cannot be relied upon to export itself (in case it has been corrupted), so a hardware mechanism should be provided.

The COTS software upgrade problem is somewhat worse. Upgrades to the Windows operating system, sometimes made necessary by newly discovered security vulnerabilities, are often performed over the Internet. This requires connecting a dedicated election laptop to the public Internet, a thoroughly unwise idea because of the possibility of malware infection. COTS software upgrades should only be made from

[12] Assuming that there are only 100,000 voting machines in the United States, probably a low estimate.

removable media supplied by the ITA from digitally signed copies furnished by the vendor and examined by the ITA.

Rogue Compilers

It has been pointed out many times that perfect, clean voting system source code is not a guarantee of freedom from malware if the code is compiled on corrupted compilers. That is, the compiler may alter or insert object code that does not correspond to the source code on which it is operating. The complier might be “alerted’ to the fact that this code is to be corrupted by the presence of an innocuous character string or otherwise unremarkable sequence of source code statements. This is surely hypothetically possible. The question is how the corrupt compiler might have been created or introduced into the process.

It is true that vendor might employ a rogue compiler of its own design. It might then feel confident in exposing its pristine source code for all to see, yet manufacture corrupted object code inside its factory. This scenario will not be successful. The ITA uses compilers of its own and creates a “witness build” of the voting system on its own premises. An MD5 hash of the object code and the object code itself is maintained by the ITA and can be used for comparison purposes in the event of an alleged irregularity.

It has been floated about that possibly all compilers, or the main commercial ones, such as Borland C, might have been corrupted by programmers at Borland, and therefore the compiler used for the witness build at the ITA might be as corrupt as the one used by the vendor, and anyone else, for that matter, who might want to verify the object code. Without commenting on the likelihood of this attack, I point out that its effects would be detected in parallel testing.

Physical security

As an overall matter, physical security in voting systems is illusory. While various devices may impress voters psychologically, such as locks, seals and tamper-evident tape, each of these is relatively easy to foil and no election should depend solely on physical security for its integrity. The security of desktops and laptops used for ballot setup and tabulation is usually worse, since such machines do not provide for locks or seals.

Cryptographic tokens, passwords and other authentication mechanisms restrict access by outside intruders, but neither they nor physical security are effective against insiders. The tamper-evident seal is a good example of a hurdle but not a complete bar. This is a roll of adhesive tape having sections bearing non-repeating serial numbers. The manufacturer guarantees that no serial numbers are duplicated, even over its entire manufacturing run of the product. The seal is “tamper-evident” in that if an attempt is made to remove it, a warning message is left behind to indicate that intrusion has occurred. Such seals can be forged, and can even be removed in ways that do not cause the warning to appear[13]. It is unlikely that in an election setting anyone would check whether a forged seal was in use, though it is expected that routine checks would at least confirm serial numbers. This is not to argue against the use of seals, but merely a plea that their limits be appreciated.

Even if the seals are genuine and intact, they do not protect against insider threats. What we have, therefore, is a collection of checks, certifications, tests, physical perimeters, human witnesses and administrative procedures that collectively present either obstacles to intrusion or means for detecting intrusion. It is up to each jurisdiction to determine whether it considers the entire set of security measures adequate in any given implementation.

The Review Process

The reviews were conducted in a conference room in the McCormack Office Building at One Ashburton Place, Boston. Each vendor brought and set up its equipment in those premises. The reviews were attended by vendor representatives and staff of the Secretary of State and, on occasion, other state offices. The exams in total lasted just over 15 hours for the three systems.

This security review does not pretend to be a complete security analysis of any of the three systems examined. Such reviews have been performed by other organizations, and are referenced in Appendix A. The starting premise in this case was that all three systems are voter-verified (two by paper trails and one by virtue of an optical ballot), and thus interest was confined to security threats that would not be discovered in a voter-verified system as used in Massachusetts. Massachusetts does not utilize modems or Internet transmission of even unofficial vote totals, and the election night results tallied on electronic tabulation systems are not used in the canvass prior to certification of winners. This means that attacks on jurisdiction-wide systems (i.e. city and town systems in Massachusetts), while undesirable, would not result in any loser being declared a winner. The significant security attacks in this setting would have to be directed to the voting machines themselves, rather than the jurisdiction's central setup or tabulation software. Thus attention was paid to the origin of the software, how to authenticate that it corresponds to properly certified software, and ways the software or firmware in a voting machine might be replaced, either with authorized or unauthorized versions.

Each vendor was given an opportunity to make any presentation or demonstration it wished. We then set about to list all the system components, what software or firmware they contain, how that software is generated and distributed, and how it came to be present in the machine. I was particularly interested in the provenance of each item of software and what protection it might afford against substitution or tempering. I then made various attempts, as appropriate for each type of system, to corrupt the software in

[13] See Chapter 12, "Security Printing and Seals," in Prof. Ross Anderson's excellent and eye-opening book, "Security Engineering," available at http://www.cl.cam.ac.uk/~rja14/book.html.

the machine and any election data, including vote totals. Extensive discussion was held with each vendor as to how various threats would be detected and parried.

Following the examinations, I was provided with DVDs, which I reviewed in their entirety in preparing this report.

II. Hart eSlate

This section is based on the security review performed on August 2, 2006 and Hart InterCivic's response dated August 15, 2005 to a request from the Commonwealth of Massachusetts for a "Voting System Equipped for Accessibility."

System structure

eSlate is a generic term for a comprehensive system that includes ballot definition hardware and software, voting machines, assistive interfaces and tally hardware and software. (eSlate is also the name of the DRE voting terminal on which the voter votes.) Below is an inventory of system components and the security issues they raise.

eSlate™ 4.2.13. This is the DRE voting terminal with which the voter interacts. It is not a touchscreen but presents instead a physical button and wheel interface as well as a color display screen. The regular voter votes by turning the wheel and pressing buttons to make selections. The disabled voter votes using one or more assistive interfaces (possibly also including the butt and wheel interface), as described under "Disabled Access Unit." below.

The result of voting on eSlate is that a CVR is created and stored in three separate places, flash memory in the eSlate, flash memory in the JBC and on the MBB inserted in the JBC. All three of these records must be identical, or the system will not continue in operation. After an election, each of these records can be extracted. Therefore, any exploit that only alters one or two of them can be detected. An exploit that successfully changes all three must be exposed in a different manner.

eSlate is not configured as a general-purpose computer but is an embedded system based on the Motorola Coldfire 5307 processor running the Precise MQX 32-bit real-time operating system. The source code to the operating system is in Hart's possession and the OS and eSlate software are compiled together to yield a single integrated file. No other programs can run on eSlate unless the firmware is modified. The OS is not multithreaded and the device has no hard disk. All the software runs from firmware. While the software is operating, it is continuously performing a cyclic redundancy check to detect whether there has been any alteration to the firmware. This is not done specifically as a defense against tampering but to ensure that firmware device errors do not affect the election process. Each eSlate has an electronic serial number that is installed at manufacture.

Each eSlate is provided with a thermal VVPAT printer housed in a separate compartment next to the eSlate box. Both of these components fit into a tray on a stand to raise them to a convenient height for voting and a black privacy curtain is provided to prevent other from observing the activities at the eSlate. This voter-verified capability is referred to by Hart as its Verifiable Ballot Option (VBO) 1.8.3.

eSlate has two communication interfaces: (1) a serial port driven through a stripped-down version of the RS-485 protocol, allowing communication with the Judge's Booth Controller, and (2) an interface to the VVPAT printer to which a connection is made automatically when the eSlate in installed in the voting booth.. There are no modem or LAN ports.

Each eSlate has an onboard battery capable of powering the unit for 18 hours of use, if properly charged. The VVPAT printer is separately powered. At a polling location, the eSlates are daisy-chained both for AC power and for communications with the JBC. However, the data connection is a pass-through – votes from individual eSlates are not read or processed by other eSlates, and the daisy chain is not interrupted if one or more individual eSlates fail during an election. The only effective data connection, therefore, is between an eSlate and the JBC.

The principal eSlate security issues concern the origin of the software and firmware, the physical and electrical security of the device prior to and during an election, the proper correspondence between recorded votes and the paper trail, the permanence of the audit log, faithful display of ballot sent by the JBC and the alterability of Cast Vote records.

eSlate contains no dip switches or wireless components. It is assembled by Suntron in Sugar Land, Texas under contract to Hart. Because the manufacturing process is essentially unauditable, we must rely on testing methods to verify the behavior of the hardware. Suntron subjects its employees to security checks and implements secure procedures in its factory, but there is no way to tell in a specific instance whether a machine has been assembled from genuine components. The ultimate check is through the VVPAT and parallel testing.

Disabled Access Unit™ (DAU). This is a physical unit that, when installed in a standard eSlate, provides alternative access features for disabled voters, including an audio ballot function, jelly switches and sip-and-puff interface. The DAU allows insertion of a PCMCIA memory card (MBB Card) containing audio ballot information. This is separate from the MBB inserted into the JBC.

The principal DAU security issues concern the origin of its software and firmware, the correspondence among the ballot, the displayed candidate names and the spoken candidate names, and the accurate audio summary of the ballot for review by the voter before casting a vote.

Judge's Booth Controller™ (JBC). This is a console installed at the polling place that allows an election judge to manage up to 12 eSlates. If a polling location requires more than 12 eSlates, additional JBCs will be needed. The function of the JBC is to activate a selected eSlate for a particular voter, generate and print out a temporary Access Code enabling the voter to vote, store voting, status and audit information, and to record Cast Vote Records (CVRs) on the Mobile Ballot Box (MBB) PCMCIA inserted in the device.

When a voter votes, a Cast Vote Record produced by the eSlate (and also stored on the eSlate in non-volatile memory) is transmitted to the JBC. The JBC produces totals reports at the close of polls by adding up the individual CVRs.

The JBC has a serial port originally designed for use with a modem (now disabled), a parallel port for a printer and/or firmware burner, and a serial port to connect to daisy-chained eSlates. It also carries on onboard battery that can power the unit for 18 hours. Each JBC has an electronic serial number installed at manufacture.

The principal JBC security issues concern the origin of its software and firmware, the physical and electrical security of the device prior to and during an election, proper generation of Access Codes, correct communication of ballot data to the eSlates, secure receipt and storage of CVRs and reliable tabulation.

The JBC contains no dip switches or wireless components.

Mobile Ballot Box™ (MBB). This is Hart's generic term for computer PCMCIA memory card that has several uses, including holding the election database and formatted ballots for use by the JBC, holding cast vote records and audit data from a polling location, and holding audio files to be played for visually impaired voters on eSlate.

The principal MBB security issues concern the origin of its data, whether rogue information or program may reside on it, the degree to which its contents may be altered before or after an election, and physical handling procedures to defend against substitution of MBBs.

Ballot Origination Software System™ (BOSS) 4.3. This is a Windows software application that enables jurisdictions to build election databases, format ballots, and electronically write multiple ballot types to the MBBs. It runs on a desktop computer or equivalent, recommended to be standalone.

The principal Boss security issues concern the origin of the software, operating system and BIOS, the degree to which the data contents can be altered inside of BOSS or "out of band" (outside of BOSS), the ability to verify the integrity of the software and files, and resistance to intrusions such as viruses and Trojans.

Boss and several other eSlate applications run on desktop/laptops. The laptops supplied by Hart (which are not mandatory) are Dell machines with hardware (including BIOS) as shipped by Dell. Dell is aware of the disposition of these machines because they are purchased through Hart’s commercial account. It is hypothetically possible that someone at Dell could arrange for these laptops to be configured differently (e.g. with a different BIOS) from standard machines. Even if such an exploit were successful, it would be caught by the VVPAT and parallel testing.

In 98% of the cases the jurisdiction uses a Dell computer supplied by Hart instead of one of its own. Hart modifies the machine by locking down Windows to prevent running other applications. For example, the “Start” bar, allowing a user to choose what program to run, is absent. This protects against casual intruders but could be circumvented by an insider.

Boss and the other desktop/laptop programs make use of the Sybase database engine, which is supplied in OEM form to Hart for inclusion in its systems. The Boss user interface is written in PowerBuilder, a high-level Sybase product designed for creating database application programs. The Boss ballot generation portion is written in C. A user with a separate copy of Sybase may be able to read and alter database files. However, they are password-protected. This is no barrier to an insider, but as discussed below, alteration of database files before an election will be detected. The Sybase files are not encrypted. Examining them is Notepad revealed large amounts of election data in plain text.

Tally™ **4.3.** This is the software application that tabulates and generates reports from CVRs on the MBBs. It runs on a desktop or equivalent under Windows. While the output of Tally I unofficial, it is relied upon by the public and the press and therefore must resist manipulation.

Virtually all voting tabulation programs allow manual adjustment of vote totals. While this feature appears horrifying to the uninitiated, it is necessary in an environment in which regular ballots, absentee ballots, military ballots and provisional ballots are counted at different times and on different equipment. The issue is not whether vote totals can be changed, but whether there is an indelible audit trail recording who made each change and what the change was. Tally produces such an audit log, which is stored in a password-protected database. The log entries contain the username of the person making the alteration, the nature of the alteration and the data that was changed.

The principal Tally security issues concern the origin of the software, operating system and BIOS on the machine running the software, the physical and electrical security of the machine, the ability to verify the integrity of the software and files and whether totals reports can be altered after they are generated.

Rally™ **2.3.** This is a laptop application used to run satellite data collection sites at which CVRs can be read from MBBs brought from multiple polling places for transmission to Tally at a centralized location. It would be used sparingly in Massachusetts, if at all, as it is designed for large, geographically dispersed jurisdictions for which driving time is a factor in assembling results for tabulation.

eCM Manager 1.1. This is laptop software necessary to support Hart's removable crypto tokens that are used for two-factor authentication to gain access to certain restricted election functions.

System for Election Records and Verification Operations™ (SERVO) 4.2. This is a laptop-based election records and asset management system that maintains equipment history and election records. SERVO is used to recover data from equipment in the event an MBB is lost or damaged. It also has an administrative role in maintaining eSlate equipment and MBBs, keeping inventories and reading audit logs, and if it is corrupted there will not be sufficient evidence to resolve claims of irregularity. Therefore, the security of SERVO needs to be evaluated.

Trans. Software used for managing translation of ballot information into multiple languages, the key issue being to ensure that all translations of the ballot have exactly the same candidates in the same positions. Trans is used for both alternative language ballots and audio ballots. The relevant security questions are: (1) does Trans maintain the association between candidates on ballots in different languages properly; and (2) how secure is the output of Trans from later manipulation?

Firmware burn utility. This is software running on a laptop that is used to flash new firmware into the JBC and eSlate. Obviously anyone who has access to this software and obtains physical access to the devices has the potential to corrupt all of the firmware in a jurisdiction's machines. While it is normally available only to Hart employees, there is no way to know whether any copies are circulating outside Hart. It also means that Hart employees could potentially be the source of an insider threat. The interesting question, raised several times during the review, is how can a jurisdiction determine exactly what firmware is resident in the eSlate and JBC? If it cannot, then what defenses are there to the introduction of malware? Fortunately, the VVPAT and parallel testing would reveal any intrusion in all but the most unlikely scenarios.

Cryptographic tokens

The security mechanisms used in eSlate were designed by @stake, a computer security firm which was subsequently acquired by Symantec, Inc. Various (not all) files are digitally signed using keys that are set by eCM Manager and installed on physical USB tokens that must be inserted in particular computers and counter-authenticated with a user-entered PIN for digital signing to occur. The signed data on the MBBs includes ballot definitions and CVRs. For example, if an attempt is made to load an altered MBB into a JBC, it will not succeed because the digital signature will not be correct. The possibility of an outsider (one who does not have access to the keys and PINs) modifying an MBB is virtually zero in any reasonable amount of time (e.g. less than a year) Any rational model of a threat addressed to MBBs can therefore be confined to insiders.

If an insider tries to create an MBB by running Boss on the same computer used to create the original election MBBs, he will be unsuccessful. This is because the act of creating MBBs is logged by Boss. If the insider attempts to use a different computer, the time-dependent data written to the MBB will not match the corresponding data in the election database on which tabulation will occur, and the attempt will fail.

There are two important additional reasons that MBB tampering will be detected: the VVPAT and parallel testing.

Malware

It is often alleged incorrectly that it is easy to introduce malware into a voting system. It is especially difficult in the case of eSlate. In this section we examine the various locations in which malware might be inserted, and discuss the corresponding risks. Because the logic associated with ballot presentation and capture is resident in eSlate, that is the most likely target of a software program attack.

An objection frequently raised against DREs is the possibility of substituting a Trojan for the legitimate ballot presentation and capture software, in this case the eSlate firmware. The question is how the Trojan would evade detection. The method usually cited, using an onboard clock to determine when the actual election is in progress, will not work on eSlate because it possesses no onboard clock and does not receive timestamp data from the JBC. While the Trojan could certainly behave differently at LAT than when the unit is in open polls mode, the firmware would be unable to tell whether the unit was being tested in open polls mode. Therefore, routine examination in that mode would reveal the Trojan's presence, as would the VVPAT and parallel testing. It is true that the Trojan might lie in wait, behaving properly, until a certain number of votes, say 150 had been cast, and thus resist detection in routine testing. However, the VVPAT and parallel testing would remain effective.

Alteration of the DAU circuitry could interfere with the rendering of voice or swap inputs from the jelly switches or sip-and-puff interface. Neither the DAU nor the eSlate possess a real-time clock, however, so there is no practical way for the DAU to behave differently at LAT than it does in an election. Substituting a new DAU after LAT is prevented through tamper-evident tape and physical security. Even if a DAU attack is successful, the exploit would be discovered by parallel testing and/or by verifying the assistive interfaces during the election.

It does virtually no good to tamper with the JBC firmware. The reason is that the JBC only downloads static ballot data to the eSlate units. Altering the appearance of the ballots would immediately be noticed by knowledgeable voters, who will see that candidates or issues are missing from the display. A counterattack is to have election workers verify the ballot during the election on each eSlate.

A potential attack on the JBC would be to program it to pre-authorize a large set of Access Codes that could be used indiscriminately by voters to vote multiple times. Not only will this be noted at the close of polls because of a discrepancy between votes cast and voter who appeared at the polls, but the act of voting is indicated on the JBC. Thus a judge who is monitoring the JBC will observe that a voter in a booth is voting more than once.

MBB alteration is impractical, even for an insider, for the reasons discussed above.

All of the eSlate software that runs on laptops is much more vulnerable than any of the voting firmware. The reason is that the Windows environment is fundamentally insecure, especially against insider attack. The system administrator can install, delete or modify any software he wishes. Hart has published in the National Software Reference Library (NSRL) the MD5 hash of each file (including object code) that is not modified in an election cycle. If there is an effective way to computer the hashes of these files as installed on a laptop, then the hashes can be checked against the library values to learn whether even a single bit has been altered. The question remains how a trusted MD5 computation can be performed. If the administrator has substituted a Trojan MD5 computation, then this check will not reveal any alterations. Also, if the modifications are confined to changing files, the MD5 test if no use at all.

Hart has begun a pilot program to distribute code to check hashes. This will work, if uncorrupted, for the laptop application software, but it is not useful for the JBC and eSlate firmware.

One can imagine an independent testing process (not performed by an insider), in which files are taken from the laptop and subjected to a separate MD5 computation. This might be desirable if effective logistics could be worked out, which seems unlikely. The corrupt administrator would replace the corrupted files with legitimate ones before the test and would switch the bad ones back afterward. The real protection, however, is apparent by considering in turn what would happen if any or all of the laptop software were actually Trojans. While significant mischief could be perpetrated, the outcome of the election would not be affected, as discussed in the following paragraphs.

The inputs to Boss are a variety of files, some encrypted, some not, some password-protected and some not. Some of the files can be altered meaningfully outside Boss, even using standard Windows tools. Others require more sophisticated manipulation. However, even corrupting Boss or its files would not result in any change to the outcome of an election, and would be caught in normal pre-election proofing and testing. The reason is that Boss outputs cannot alter the eSlate or JBC firmware.

Altering Boss could certainly result in defective election setup being prepared. Ineligible candidates could be added to precincts, party affiliations swapped, test of propositions altered, etc. Various options, such as allowing write0ins, could appear to be enabled yet actually be disabled. However, the output of Boss is static. It consists of files which cannot behave differently while under test than they do in an actual election. There is nothing that can be done in Boss that alters the logic of eSlate. Therefore, the Lat, VVPAT and parallel testing will reveal any corruption due to Boss.

Altering Tally would create greater havoc, but would ultimately be detected. Vote totals are produced at each polling place on the JBC, printed out, then both posted publicly and physically sent to the jurisdiction. If Tally contained corrupted code that miscounted CVRs, the results reported at the jurisdiction would not match the printed records from the polling locations. This would be detected quickly by poll workers and the public and in any event would be observed at the canvass. The risk is that incorrect unofficial results might be reported to the press on election night, creating public skepticism and misunderstanding. It is therefore important, though not absolutely indispensable, to protect Tally and its records from tampering. Common administrative methods, such as password protection, physical security, and dedication of the Tally laptop to election functions only can make it difficult for all but a very small number of people to access it. If there were a way to verify the MD5 hash independently, at least the integrity of the object code could be assured. However, this would not serve to check the data processed by the software.

Running Tally requires a crypto token. The election database is supplied to Tally via a CD generated when the original election setup was performed. If anyone has altered the first election database since then, the alteration will have no effect since Tally will never see it. If an election database is used that is different from the one from which the MBBs were created, Tally will detect the discrepancy.

The risks to Rally are similar to those for Tally, except that modem transfer creates additional avenues of attack. I do not recommend modem transfer of election results. If its must be done for administrative reasons, the MBBs that are read at satellite locations for speed should still be transported to a single central location for retallying. This will serve as a check on the integrity of the modem count.

eCM Manager. Altering eCM Manager could allow indiscriminate creation of key tokens with insecure or identical keys. This could in turn permit unauthorized people to gain access to election data and records. However, these keys cannot be used to alter an eSlate firmware and thus the intrusion might cause inconvenience but would ultimately be detected by the other methods discussed above. In short, the eCM Manager itself does not control any election results, but could be used as a tool to allow more people to access system components.

SERVO. It is possible that a modification to SERVO would render it impossible to obtain reliable results from eSlate machines if MBBs were damaged or lost. However, the printed election results from each JBC and the VVPAT cannot be altered by SERVO. In the event that a machine failed and no totals tape could be obtained, a corrupted SERVO could report phony results from the eSlates. The defense against this is to use a different copy of SERVO running on a different laptop to obtain the CVRs from the eSlates a second time for a cross-check.

Trans. This program is freely distributed and therefore susceptible to the creation of Trojans. However, its effect is only to create ballots that are proofed and checked by other means, so it is not a useful program to tamper with.

The eSlate system contains no interpreted or self-modifying code.

Paper trail

The Hart VVPAT prints out an unverifiable barcode on the paper tape that is supposed to correspond to the CVR written on the eSlate for that voter. It is unverifiable because humans cannot read barcodes. The reason the barcode is printed is ostensibly to speed up a recount, which would otherwise have to be performed manually. An exploit, therefore, is to print the VVPAT correctly, but create a false CVR and print a corresponding false barcode on the paper tape. This will be caught only in a manual recount that does not rely on the barcode.

The VVPAT is retained internally in a canister in the onboard printer. The printer is not designed to allow paper replacement during the election, so it can be sealed safely with tamper-evident tape and a physical seal before being used. This is an important measure since there is a tendency, supported by statute in most jurisdictions, to regard the VVPAT as authoritative, and is irrebuttably presumed to correct regardless of any electronic evidence to the contrary. Because of this, it is essential to protect the integrity of the VVPAT after the election, as a substitution would very likely be taken as authentic in the event of a recount.

The Hart VVPAT does not show full candidate names because of a discrepancy between the screen ballot and the maximum number of characters that can be printed on the paper tape. This is deficiency of most VVPATs, which seemingly could be remedied by wrapping long text onto a subsequent line.

Passwords

Passwords are used in many places in the eSlate system. To the extent they are well-managed, that is, changed frequently, not repeated among users, not easily guessed, kept secret, etc., they provide a degree of protection against outsiders.

The Boss password can be reset by reinstalling Boss. The installation disks normally remain under Hart’s control, but one cannot dismiss the possibility that someone outside Hart has copies of these disks. This means that certain password protections are illusory, but are still satisfactory to resist outsiders.

Physical security

The physical security of a voting system component must be evaluated based on its intended uses and applicable threat models. With few exceptions, the physical security of eSlate and the JBC during an election is adequate. (One except is that there is a hinged panel directly above the eSlate and VVPAT this is supposed to conceal various wires connected to those units. This panel has no lock, and thus could be tampered with by a voter on Election Day. A remedy is discussed below.) In general, physical security is useless against insiders who not only have access to the equipment but are also provided with physical keys.

Various efforts were made during the review to interfere with the physical integrity of system components. It was of interest what a voter might be able to do to an eSlate during an election. In normal use, warnings of various anomalous conditions, such as an attempt to disconnect an eSlate from the JBC, are displayed on the JBC. To test this, I pulled the daisy chain cable from the eSlate. No warning appeared on the JBC. The reason, it turned out, was that the red and green annunciator lights, indicating the status of each eSlate, were burned out. Such cascading phenomena, in which an event fails to be noticed because of a defect in the warning mechanism, are not uncommon. In this case, it took some time even for the vendor to determine what was wrong, rendering it unlikely that the condition would ever be diagnosed correctly by a poll worker.

However, even if a voter were to disconnect an eSlate or attempt to tamper with it, he could no nothing harmful other than malicious destruction. If the eSlate is disconnected from the JPB, further voting on it is not possible, so no votes would be lost. The voter has no access to the firmware, and removing the audio MBB requires physical disassembly of the unit with tools, which is not realistically possible during an election. Assuming that such a thing occurred, however, it would be caught by having poll works verify the audio ballot at intervals throughout the day.

Election Day procedures

The JBCs and eSlates are set up in advance with MBBs and sealed prior to delivery to the polling place. Each one is configured with ballot styles for a particular polling location and must be delivered to the right place. (Before being sealed with MBBs, the machines are generic and could be used in any location.) Part of the processing of opening the polls is to assign a “booth number” to each eSlate. After this is done, a zero report can be printed at the JBC, which verifies that there are no CVRs present, hence no votes.

When the polls are open, the JBC is used to issue “Access Codes” to voters. These are five-digit numbers which, when entered into an eSlate, activate it for voting with the correct ballot style for that voter. The Access Code is printed on a piece of paper, which is torn off and handed to the voter. The code can only be used once, and must be used within a time period that is settable through Boss. A 30-minute lifetime is typical. A visually impaired voter is told his Access Code orally. No mischief is possible, since if the code is invalid, the voter will learn that fact when attempting to vote.

Various polling place procedures require knowledge of different passwords, but no crypto tokens are used.. When the system is delivered from the manufacturer, the JBC recognizes a default password that is set at the factory and is the same for all JBCs in the world. It recommended by the manufacturer that the default password be changed the first time the JBC is turned on.

Audit records

Vital system functions performed on various components of the system are logged in log files. The eSlate and JBC are stateful machines; that is, they may exist in one of a number of states and are aware of which state they are in. State changes on these devices are logged. For example, on the eSlate the entry of an access code, rejection of a VVPAT ballot and cast vote operations are logged for each voter. The eSlate log can be dumped from unit via SERVO. The JBC log is written to the MBB that is in the JBC and can be examined through Tally.

Logging of events on the laptops (Boss, Tally, etc.) is somewhat more problematical. The reason is that these are Windows machines whose files can be manipulated outside of the Hart applications. In some cases insider information, such as a database password, would be needed to perform the manipulation, but insider threat is what must be guarded against. The saving grace is that manipulation of Boss files is not useful because of subsequent proofing and verification activities, and manipulation of Tally files, discussed above, does not affect official results.

When we attempted to review an audit log, the Boss application kept crashing with an “unknown software exception.” This was anomalous behavior that turned out to have resulted from a “bad” install of Boss. While this is probably a correct explanation, it raises the question how a user might able to verify that he has a “good” installation of Boss. I say this because Boss was not completely disabled. It was still able to perform many of its normal functions.

Audio and accessible ballots

eSlate does not offer any text-to-speech (TTS) capability. Therefore, all audio used in an election must be recorded human voice, typically in the form of .wav audio files. These are created by speaking into a microphone connected to a laptop running Trans and, typically, Boss. Data produced by Trans is stored in the Sybase election database. The files on the MBB must have identification that matches the database, so it would be difficult, but not impossible, to modify the audio files on the MBB after they have been written. However, there is no way for any component of the system to verify that the words spoken on the audio correspond to the text on the eSlate screen. It is also possible that an insider could construct a different but apparently legitimate audio card. Therefore, the correspondence must be verified manually. To check that no errors have been made, verifying the audio ballot at LAT is sufficient. However, this will not reveal time-sensitive malware. Therefore, the audio function should be one of the subjects of parallel testing. It can also be verified by an election worker once an eSlate has been activated for voting but before the voter begins voting.

The audio card itself can be sealed into the eSlate at LAT. Both a seal and numbered tamper-evident tape should be applied. Unfortunately, the security mechanisms designed to prevent tampering with an election MBB have not been implemented on the audio MBB. This is a deficiency that should ultimately be corrected by the vendor. In the meantime, administrative procedures, recommended below, can negate the threat of audio MBB substitution or tampering.

The write-in function is not supported for accessible ballots to the same degree it is for regular ballots. While it is possible to enter a write-in, a visually impaired voter has no way of determining after the fact whose name has been written in, short of deselecting the write-in and entering it again[14]. The ballot review page announces that a write-in has been cast, but does not spell the candidate name. The process of entering a write-in is fairly cumbersome for all voters because of the need to turn the wheel repeatedly to select letters. If a disabled voter must also enter the address of the write-in candidate, it will be onerous to have to do more than once because of the lack of review capability.

Software Updates

We must examine how the software/firmware on eSlate systems is upgraded or modified. The eSlate firmware can be replaced by connecting a device to the printer port and using the firmware burn utility, which requires a user name and password. Authentication by crypto token is also required, but no logging occurs. These are no protection against an insider who somehow has possession of an eSlate Trojan. The defense to this threat is via indirect methods, the VVPAT and parallel testing. In the future, Hart plans to build firmware flash capability into SERVO, where logging will be possible.

The question, therefore, is how to detect malware that may have been introduced into the eSlate. The answer is through a combination of the VVPAT and parallel testing, which will reveal any systematic intrusion. Unless voters verify the VVPAT, intrusion that is not systematic (i.e. that is not present in a large number of machines) will not be detected.

Intrusion attempts

During the review, I attempted to alter a .pbd file using Microsoft Notepad. This caused file corruption so extensive that recovery was not feasible without reinstalling the entire Boss application. I also modified executable and database files and these intrusions were also detected. Modifying files on the MBB results in their being rejected by Tally. I conclude that file modification is not an effective threat against eSlate. It would be necessary to substitute Trojans or the equivalent to make any difference, and that exploit would be revealed by the VVPAT and parallel testing.

Parallel testing

[14] It is questionable whether failing to allow a visually impaired voter to review a write-in complies with Sections 301(a)(1)(A)(i) and 301(a)(3)(A) of HAVA, but this is not a computer security matter and thus not part of this review.

The eSlate architecture presents a certain barrier to parallel testing. Because the machines are daisy-chained, and all CVRs are sent to the JBC from all eSlates that are connected to it, there is no practical way to sequester a single eSlate and wall it off from the regular election for parallel testing. The only way to do this would be to connect the eSlate(s) under test to a separate JBC so the votes cast on the machines would not be counted in the regular election. Unless a polling location can spare more than one eSlate for parallel testing, rogue code might assume that if the eSlate's both number is 1, then it is a solo eSlate and would behave normally, thus evading parallel test. This strategy would cause rogue code to remain hidden in polling places with multiple eSlates. However, the code would have to behave properly in polling places with only one eSlate.

eSlate Conclusions

In my opinion, the critical components of the eSlate system are safe against credible attempts at tampering by outsiders. (An outsider is someone who does not have privileges or confidential documentation.) The reason is extensive use of passwords and crypto tokens and the fact that the eSlate and JBC are not configured as general-purpose computers and do not expose network connections. Even if an outsider is able to gain access to an eSlate, JBC or MBB, he will not be able to perform useful manipulations, i.e. those that would not be detected by the system itself.

Certain kinds of insider intrusions are possible because the insider has the tools necessary to generate and modify cryptographic keys and gain access to all the applications. However, use of the VVPAT and parallel testing negates this risk. Intrusions to the eSlate can be detected in pre-election LAT since without an onboard clock eSlate is unable to determine whether a real election is in progress.

III. Diebold AccuVote TSx

This section is based on the security review performed on August 7, 2006 and Diebold Elections Systems' response entitled "Voting System Equipment for Accessibility: Elections RFR" dated August 15, 2005 to a request from the Commonwealth of Massachusetts for a "Voting System Equipped for Accessibility." The Massachusetts distributor of Diebold voting system is LHS Associates, Inc. of Methuen, MA, which also maintains the equipment and, under contract to jurisdictions, provides ballot setup services.

AccuVote is a comprehensive system for integrating results from various different types of Diebold voting systems, including DRE and optical scan. It includes the following components:

AccuVote TSx 4.6,4 with AVPM (AccuView Printer Module). This is a touchscreen DRE with VVPAT printer and assistive interfaces. It consists of an Intel PXA255 process, typically used for embedded applications, 64MB of non-volatile flash memory, 64MB of RAM, a graphics controller, smart card reader, touchscreen and various communication ports. It connects to an onboard VVPAT printer through an RS232 interface, the touchscreen via a serial interface, a keypad for the visually impaired through a serial interface, and includes a modem port with telephone jack. There is also an analog audio output for headphones and a sip-and-puff interface. It has two PCMCIA slots for memory cards and/or a local network interface card used to connect to GEMS via direct cable.

More than one TSx can be daisy chained to others, but this is for electrical power only. No election information is transmitted via this connection. The TSx contains no wireless devices.

TSx Flash memory contains three types of information:

- Boot loader. This occupies the first 256KB of flash memory and is loaded into RAM on startup for execution.
- Windows CE image. This is a copy of the operating system and its registry, which is also loaded into RAM on startup.
- A file system containing election data, an event log and election archives. It also holds the executable version of BallotStation, the application that interacts with the voter on the TSx.

The file system contains a redundant copy of information loaded onto the memory card. If an attempt is made to tamper with either one, the copies will not coincide, and an irregularity will be detected. TSx has no hard disk drive.

PCMCIA Memory Card. This is a 64 or 128MB removable memory card used to provide ballot display information to the TSx. Cards are loaded using a TSx unit while the unit is connected to a GEMS server via a cable. The connected TSx can be used to load many memory cards. Once loaded, the cards are inserted in the jurisdiction’s TSx units. When the units are powered on, they read the ballot setup information the memory cards and set themselves for an election.

The memory card can hold several distinguishable types of content:

- Election information. These are files defining the races, parties, candidates, and questions for each ballot style to be presented to voters. Audio and rich text files hold information for disabled voters, including the contents of .wav or .mp3 files with spoken test to be played to the voter. Also included is an election database that will be used to hold CVRs from the TSx in which the card is inserted. An election archive is maintained of past elections, which is purged selectively as memory is used up, and log file of administrative operations is maintained.
- AccuBasic object files. Diebold uses a proprietary report definition language called AccuBasic to define customized report formats. More than just a format specifier, it allows executable code to be defined that will be run when it is necessary for TSx to produce a printed report. The nature of AccuBasic will be discussed in detail below, but the memory card must contain the compiled AccuBasic (.abo) files needed to generate the necessary reports on the TSx.

- Executable code. The memory card may also contain four additional items of executable code used to update the TSx unit: (1) a boot loader, (2) a customized version of Windows CE, (3) the BallotStation application; and (4) a file to erase the Windows CE registry when new software is uploaded.

The fact that the memory card is a route of introduction of new software/firmware has security implications. Steps must be taken to assure that the mechanism of uploading new software via the memory card is not used as an avenue of intrusion.

GEMS 1.18.24. This is a Windows application, written in C++ using Visual Studio, the Microsoft C compiler, Microsoft Foundation Classes, the Jet database engine and Windows 2000 Server (obtained from an office products store). It also includes third party device drivers, compression utilities and audio code libraries. It is used for election administration on a "central" desktop or laptop, commonly an off-the-shelf machine from Dell. That is, GEMS is not present at polling locations. In practice, because of the local nature of Massachusetts elections, many jurisdictions will not run GEMS at all, but will have election setup preformed by LHS Associates. Some jurisdictions will choose to run GEMS to tabulate unofficial results; others will tally results directly from printouts prepared at polling locations without using GEMS at all. This configuration differs greatly from those in some other states, notably Maryland and Georgia, in which GEMS is installed on a server in each county and in the state capital and unofficial results are transmitted via networks.

Fundamentally, GEMS performs both pre-election and post-election function. Pre-election it is used for ballot setup and to create PCMCIA memory cards for use in the TSx. After the election it is used to tally unofficial results and produce reports.

Any data that needs to be loaded onto a PCMCIA memory card must be present on the GEMS system that writes the card, including election data and the AccuBasic object files. To prepare memory cards, the GEMS server is connected directly to a TSx unit locally by means of a direct cable. The TCP/IP protocol is used to transfer data to the TSx, which then digitally signs the data. Multiple memory cards can be made using this process, enough for the entire jurisdiction.

There are two versions GEMS, having different prices. "Full" GEMS allows ballot setup. "Upload only" GEMS is only useful for unofficial tallies of election results. After the GEMS software is compiled during the "witness build" at the ITA, it is distributed to jurisdictions by the ITA using encrypted CDs.

Access Cards. These are plastic contact smart cards that come in four varieties: (1) Central Administrator; (2) Supervisor; (3) Voter Access; and (4) Key Card. These cards are distinguishable at manufacture and one sort of card cannot be turned into a different variety after that point. The Key Card is used to transport cryptographic keys to units so that only cards that have been prepared for insertion in machines a particular jurisdiction will be recognized. This prevents someone who happens to possess a card of a particular type from using it in an election. The access cards are Diebold-specific. It is not feasible to take a stock smart card and attempt to make it work in a Diebold machine.

Voter Card Encoder (VCE). This is a small hand-held device with buttons, a card slot and a liquid crystal display that places information on Voter Access Cards so they can be used to activate a TSx unit for voting. When a voter appears at a polling place, a poll worker uses the VCE to produce a Voter Access Card designating the correct ballot style for that voter and any other necessary options, such as the activation of an audio or magnified ballot. The VCE can hold up to eight ballot styles. If more ballots styles are used in a polling place, more VCEs are needed.

The VCE learns the necessary keys when it is placed in a certain mode and a Supervisor Card is inserted an the correct PIN entered. A Key Card produced by GEMS can then be inserted and the key information will be uploaded to the encoder in a way that cannot be retrieved by reverse engineering. (That is, it is not feasible to learn the keys even if one has possession of a VCE for a prolonged period.

The Key Card is also used to transfer keys to the TSx units. These must match the keys used on Voter Access Cards and are also used by TSx to digitally sign ballot data and memory cards. Keys should be changed after each election.

Because the VCE can be used to generate valid Voter Access Cards, it poses security issues. For example, if VCE were in the hands of a political party, could it be used to produce extra cards so that voter could vote multiple times?

VC Programmer. This is a Windows application that permits creation of Voter Access Cards for use with TSx o a smart card reader/writer attached to a PC. To do this properly it requires access to a file exported from a GEMS election database. If VC Programmer is corrupted or replaced by a Trojan, it might be used to create erroneous Voter Access Cards that could cause TSx to display the wrong ballot to a voter. It is important for voters to be aware of which candidates and races they should be viewing. However, it is likely that in minor races a manipulation of VC Programmer would be successful, as many voters would not realize that the ballot was incorrect. A method of detecting the problem (after the fact, unfortunately), is to have the VVPAT print the candidates that the voter did not select as well as the ones he did choose. In this way it can be determined later which choices were presented to the voter.

Key Card software tool. This is a Windows application used to load Key Cards with the security keys necessary for the TSx and VCE units in a jurisdiction. This application must be run in a secure, controlled environment, or it will be possible for unauthorized people to obtain privileged cards, enabling them to perform administrative functions on the TSx and VCE.

Structure of the TSx

The TSx is a complex device with sophisticated software and a variety of external interfaces. The TSx motherboard contains no dipswitches, but has debug jumpers, a slot for an SD (secure digital) card and two JTAG ports. Diebold explains that these slots and ports are unpopulated and unsupported, but it is not evident from a physical inspection what use might be made of them. The Hursti II report listed in Appendix A raises the prospect that the JTAG port could be used for an attack on the TSx. Even if such an attack were successful it would be caught by the VVPAT and parallel testing.

There is no useful way to determine what firmware is loaded into the TSx because the firmware cannot readily be dumped. It is a curiosity among vendors that the relatively safe operation of exporting the contents of firmware is nearly impossible, while the dangerous operation of uploading new firmware into the machine is easy.

Encryption

TSx uses a 128-bit AES encryption key (symmetric) for digital signing and a 64-bit DES key on the smart card to encrypt the 128-bit AES key. Thus even if someone finds or steals a Key Card for the election, he will be unable to obtain the symmetric signing key and will not be able to forge or alter the election contents of the memory card.

After a voter votes, a new CVR is added to the memory card and the old digital signature is replaced with a corrected one. If the machine goes down at any time, all of the votes cast so far can be recovered from the flash memory, the memory card and/or the VVPAT record.

The SSL protocol is used for transmission of data to the TSx, even over short distances. The ITA creates MD5 hashes of executable files and places them in NSRL. While this can provide some degree of assurance that GEMS programs are legitimate, since it is easy to acquire an independent MD5 has checker, the hashes are nearly useless for TSx firmware because there is no straightforward way to computer the hashes from the firmware.

GEMS

The vendor recommends that GEMS be run on a dedicated standalone computer with no network connections. This is a good recommendation, which should be made mandatory by regulation. The fact that the GEMS computer as sold by Diebold is “locked down,” however, provides only illusory protection because this mechanism is easily subverted by an insider. It may protect against casual intruders, not insiders.

Software/Firmware Updates

Much has been written about the ease with which the TSx firmware can be replaced. (See, e.g., Hursti II in Appendix A.). Unless defended against, it is a severe security hole for the simple reason that when a TSx unit is prepared for an election, no statement can be made as to what software it might be running because the firmware might have been reflashed at any prior time.

When the TSx is powered up, the firmware boot loader initializes the machine before loading the operating system. One of the first things the boot loader does is look on the memory card for a different version of the boot loader. (It tests for a difference by a cyclic redundancy check.) If a different version is present, the boot loader loads the new version into firmware and reboots the machine. This is done without warning to the user and no credentials or authentication is required. The user might not even be aware that the reflashing is going on or even that there was a new firmware version on the memory card.

Next, the boot loader checks for a new version of Windows CE. If one is present, it immediately replaces the version in firmware without a prompt or warning. Otherwise, the old version is loaded.

When Windows CE starts up, it checks the memory card for a version of the BallotStation application that differs from the one already in firmware. If one exists, the user is asked if he wants the old version replaced.

A recent report, referred to in Appendix A as the "Princeton Report," detailed how it was possible to reflash the firmware in an AccuVote TS unit within a minute and demonstrated the equivalent of a computer virus for spreading malware from one TS unit to another. Without commenting on whether the scenario laid out in the Princeton Repory is realistic, I observe that the viral exploit does not work on the AccuVote TSx units being considered in Massachusetts.

Being able to upgrade voting machines relatively rapidly is necessary for efficient maintenance. However, the process could be much more secure. At the least, authentication and logging should be required. Diebold represents that this process has been made more secure in a new release now in ITA testing. In the meantime, see the "Ballot Setup Procedures" section below for a temporary remediation that will ensure that only certified software is present in the machine. This will negate the possibility that anyone might have used the vulnerability described in the Princeton Report to alter the machines' firmware prior to LAT.

VVPAT

As each voter votes, the VVPAT can be viewed. For sighted voters, the VVPAT is paged so that only the number of lines that fit in the viewing area are printed at once. There are minor difficulties with the mechanics of this process. First, the VVPAT viewing area does not necessarily provide enough space to list all the offices that appear on the touchscreen at a given time, or it may list more candidates than are visible on the screen. This is a function of ballot complexity, race complexity and font size. Therefore, the voter may have to return to the touchscreen and scroll up his review page to compare it fully with all the VVPAT entries. Second, the last candidate printed is not visible (because it is right above the print head) until the next portion of the VVPAT is printed. Third, because of limitations on the width of the VVPAT, long candidate names may not print fully.

The VVPAT prints a bar code with a CVR for possible future automated recount by scanning equipment. Printing the barcode can be disabled. A possible attack on this mechanism is a Trojan that prints the verified ballot properly but prints a barcode with a different CVR. If the ballots are not tabulated manually, this manipulation will not be detected. Therefore, a manual recount of randomly chosen paper trails is essential. The correctness of the barcode can be verified by a manual process that does not require complete tally of an entire paper roll.

The TSx VVPAT is maintained on a sequential roll of paper tape. Therefore, steps must be taken to ensure that no record is maintained of the order in which voters voted.

Ballot Setup Procedures

In Massachusetts the small size of most jurisdictions makes it impractical for them to own GEMS servers and software. Therefore, election setup, which means preparing a GEMS database with geographical units, populating the database with races and candidates, and preparing memory cards for use in the TSx units, is likely to be performed by LHS Associates. For the purpose of this discussion only, and without making any negative suggestion concerning LHS, let us assume that LHS is not a trusted party and everything it does needs to be checked.

As far as the jurisdiction is concerned, everything LHS does that can affect an election is present in the delivered TSx units, on the memory cards furnished to the jurisdiction and the VCEs that will be used at polling places. Let us assume that none of these components can be trusted. Assume that all the software on the TSx consists of malware, the memory cards contain false election setup information, corrupted .abo files and possibly object code whose effect will be to reflash the memory of the TSx. Let us see which, if any, of these exploits would be caught, and when.

Because it is "easy" to reflash a TSx unit from a memory card, a remediation procedure to ensure that malware is not present is to reflash the machine before the election with a memory card obtained from the ITA that has only the certified software (boot loader, Windows CE and BallotStation) on it. The last procedure will be to remove the memory card that is in the machine and replace it with the ITA card. Then power up the TSx and allow all the software components to reflash the machine. At this point the machine has certified software[15]. Then the memory card provided by LHS can be checked to ensure that it has no object code. If this is true, the card can be safely inserted into the TSx and the power can be recycled. This is so even if rogue election setup and .abo files are present. This will be detected readily at LAT.

[15] This was the procedure used in Pennsylvania before the May 16, 2006 primary election after the revelation of the memory card exploit now known as Hursti II to ensure that only certified software was used in TSx units.

Now run an effective LAT, including checking the audio ballot. The Trojans that were on the memory card and the TSx now have no effect, since the machine has been "cleaned" and any object code (except .abo files) has been purged from the card. We now only have to worry about election information and the .abo files. If the .abo files are illegitimate, votes cast at LAT will not be tallied correctly by the TSx and this will be seen immediately. (This is not a matter of subtlety – the .abo file have no way of knowing whether the system is under test, unlike a Trojan of BallotStation, and must engage in the same manipulation of votes every time they are invoked.

Any error, deliberate or otherwise, in the ballot setup will be detected by the parties, who check to see that all of their candidates are listed and associated with the correct parties. This data is static, and cannot change between LAT and the real election as long as the memory card is not removed (an exploit discussed later). Therefore, if ballot setup is correct at LAT, it will remain correct for the election.

This demonstrates that no person or organization, insider or not, need be trusted prior to LAT other than the ITA, who must be relied upon at this stage to supply memory card containing only certified software. Even if the ITA cannot be trusted, this will be learned from the VVPAT and parallel testing.

Polling Place Procedures

By 6:30 a.m. on Election Day, the sealed bags containing the TSx units are delivered by policemen to the polling place. The seal numbers are checked against paperwork separately provided to the poll workers. If everything is in order, the poll workers remove the TSx units from the bags and hand them to public works employees to be set up into the voting booths. To turn on the power to the TSx, a poll worker must open an external panel covering the sealed-in memory card. At this point the integrity of the memory card seal is checked to be sure no substitution has occurred.

When the TSx is powered up, it automatically produces a printer test report to verify that the VVPAT printer is capable of printing in all positions. It then produces a zero report. This is to be torn off, signed, and posted in the polling place before voting begins. A second zero report is then printed and not torn off, but spooled up inside the VVPAT canister, which is then closed and locked in the TSx. There is no need to produce any further printed reports until the close of polls.

Voting proceeds by having a voter obtain a Voter Access Card from a poll worker. The voter is escorted to a TSx unit, where the poll worker watches the voter insert the Voter Access Card into the TSx. The voter votes, hopefully verifying the ballot against the VVPAT, then casts a vote. The Voter Access Card is rendered inactive and expelled from the TSx. An audible sound is made to indicate that a vote has been cast, and the VVPAT spools up into the canister so the next voter cannot see it.

It would do the voter no good to retain the Voter Access Card after voting even he intended to use it in an illegal scheme. Suppose the voter somehow has access to a VCE containing the necessary keys to activate the card again. He would have to gain access to the polling place again, which he cannot do since a record has been made that he has already voted. He might try to hand the activated card outside the polling place to another voter to allow that voter to vote twice, but that voter would be caught since the TSx would make more than one audible sound indicating a vote had been cast while only a single voter was voting. In addition, the public counter on the TSx would advance by more than one for only one voter, providing an indication that the voter voted more than once.

At the close of voting, the polls are closed using a Supervisor Card, which requires entry of a confirming PIN. A totals report is produced and spooled up into the VVPAT canister, which is then signed and sealed. It is returned to the jurisdiction along with any opscan ballots. A second total report is produced, torn off, signed, and posted in the polling location. A third copy may also be sent to the jurisdiction for tally without the need to open the VVPAT canister.

Accessible Voting

The disabled or partially disabled voter has a choice of interfaces, including any combination of touchscreen, audio, keypad and sip-and-puff. While write-ins are cumbersome when audio is used, the candidate’s name is spelled out for the voter in audio at the review screen, so independent verification of the write-in is possible.

Audio ballots can be created through GEMS by connecting a microphone to the GEMS computer and speaking any required text. The audio is compressed, associated with the correct candidate and language and integrated into the election database that is written to the memory card. TSx has no text-to-speech (synthesized) component.

Because disabled voters represent an identified target of malware exploits, careful review of the assistive mechanism is necessary both at LAT and on Election Day.

Tabulation

It is important to distinguish between unofficial results reported quickly on election night and official results that are eventually certified by the jurisdiction to determine the winners. There are many checks and balances for both processes that have been largely ignored in the vociferous public debate surrounding electronic voting. First, when the results from each TSx are posted in the polling place, they can be recorded by any voter or party worker. It is also common for party workers to communicate the results to a central location by cellphone. Furthermore, a press staffer is often present to send the results to a press pool for early totaling and publication. If any change occurred between the posted totals and the totals later reported by the jurisdiction, many groups would be able to spot it.

The bags containing the TSx units are transported back to the town clerk by policemen. The seals are checked upon receipt, the units are removed for their bags and external seals also checked. The CVRs can be read from the TSx without removing the memory card, which remains sealed in the unit with tamper-evident tape. Therefore, floating, loose or rogue memory cards cannot be used to affect even unofficial totals. The machines are read one-by-one in public view so the CVRs can be fed to GEMS for overall tabulation. Still, the results produced by GEMS are unofficial, though they may be used by the press for rapid reporting.

It is common for towns to maintain websites on which results are posted throughout the evening. This is not done by any direct upload from GEMS over a network. It is done by manually moving data from GEMS onto a flash drive and hand carrying it to another machine to upload it onto a web server.

It is important to detail the ways in which manipulation of the TSx units and/or memory cards might occur after they are voted but before tally. Aside from requiring complicity by the police, it would necessary to make a purposeful modification to a card while it is still sealed into a TSx machine. The only ways to do this would be (1) to attempt to open the machine for voting, which fails because of the stateful nature of the TSx; or (2) to clear out the election on the TSx, set it up again, and cast new votes. The result of this would be that the real election would be archived in flash memory of the TSx and on the very memory card being used for the exploit. Therefore it would fail. A third method would be to actually open the memory card compartment and substitute a voted memory card. This will fail because when the machine is connected to GEMS, it will be determined that the election records in the machine's flash memory do not match those on the card. Another possibility would be to substitute machines (and fabricate duplicate seals). This would be caught at tally because the serial number of the substitute machine would not be recognized.

Another method used at jurisdictions to read memory cards is to prepare one TSx unit for uploading to GEMS and then successively insert memory cards from other machines into the "master" machine. This process is quite rapid, but of course requires individual handling of memory cards. However, it is not feasible to substitute another card, since it will not correspond to any machine in the GEMS database and will not have been digitally signed by a valid TSx. An attempt to upload results from the same TSx or memory card more than once fails because it is recognized by GEMS.

Any effort to simply modify files on the memory card to alter the outcome would fail because of integrity checks on the files, which are digitally signed by the TSx unit. Any attempt to read a corrupted file at GEMS will fail. I therefore conclude that no useful manipulation of the TSx or the memory card can occur while these are in transit to the counting location.

It is conceivable that the jurisdiction is running a GEMS Trojan that is set to report false results. However, this and all of the above manipulations would fail because they would result in discrepancies between the results reported by GEMS and the results posted at the polling place. The same is true if an insider used his access privileges to run GEMS and manually substitute vote totals. Even if the insider were able to disguise his acts by altering the event log files, the results would still not correspond with those obtained and posted at polling locations.

Even assuming that one or more of the foregoing manipulations succeeds, we can still recount the election independently from (1) polling place totals reports; (2) firmware records in the individual TSx units; (3) memory cards in the TSx units; and (4) the VVPAT. I have not heard of any credible exploit that would escape detection by all four of these methods.

AccuBasic

The way in which AccuBasic is used presents a minor threat to voting integrity, but one that is easily remedied. Because the memory card contains AccuBasic object (.abo) files, and these files are used to produce zero reports and totals reports at the polling place, an intruder can hypothetically get TSx to say anything he wants it to on these reports. However, the intruder must be an insider since the .abo files are placed on the memory card by the GEMS server that is used to set up the election. Let us assume a very sophisticated attack in which each memory card is given a different set of .abo files. (Using the same .abo files for all polling locations would immediately be caught, as each location's totals reports would contain the same results.) This exploit would be caught at LAT because the same .abo files are used for LAT reports as for Election Day reports.

Any AccuBasic exploit would also be caught at tally since GEMS makes no use of .abo files in its reporting. It computes totals by reading CVRs from the memory card. Therefore, the totals reported by GEMS would be different from the false ones posted in the polling location. This would point to an irregularity, and a check of the .abo files on the memory card would reveal the intrusion. The CVRs on the memory card would not be affected, so no votes would have been lost. This is also true because of the VVPAT.

While AccuBasic is fairly straightforward to reverse engineer, there are nevertheless some hurdles to overcome. AccuBasic is a source language that must be compiled into .abo files for installation on a memory cards. The AccuBasic source (.abs) files are created by Diebold and the AccuBasic compiler is resident at Diebold. The .abo files are interpreted by code on the TSx to produce printed reports. Much has been made of the apparent prohibition in the 2002 Voluntary Voting System Guidelines (VVSG) against interpreted code, but certification questions are not the subject of this report, and Pennsylvania evaluated AccuBasic and found its use to be permissible under the VVSG.

The AccuBasic exploit known as Hursti I on Diebold precinct count optical scan units does not work on the TSx for several reasons, one of which is that there is no way to store negative vote totals on a TSx memory card for the simple reason that no vote totals at all are stored on the card.

In any event, AccuBasic files are present only on “full” GEMS serves, not on upload-only GEMS, and thus in Massachusetts will only be present on LHS premises. It has already been explained that even if corrupt .abo files are used they would be detected at LAT.

Event (audit) logs

GEMS and TSx provide a variety of event log mechanisms. In general, GEMS logging is so weak that it provides no protection against an insider. However, it has already been explained that there is nothing an insider can do before an election on GEMS to affect the outcome if appropriate administrative steps are followed, and there is nothing he can do afterward, either. The simplest exploit would be to run GEMS, open the results screen, and make manual entries at will to override the correct tabulation. Then the intruder could leave GEMS and manually edit the log files to remove any trace of the change. This is highly undesirable, but the effect would be to alter unofficial results only. The manipulation would never affect the canvass or the declaration of winners, although it could cause great public dismay and mistrust and would provoke an investigation which might well fail for lack of evidence. Therefore, I will recommend below that the insecurity in the GEMS logging mechanism be corrected in future release. Diebold has been aware of this deficiency for at least eight months.

This is an example of a security flaw, exposing the basically patchwork nature of TSx/GEMS security. Certain mechanisms, such as digital signing of memory card files, are effective. But security on GEMS is largely a veneer that might stymie an outsider for a time, but would present no obstacle to a determined insider. For example, the lockout mechanism that prevents a user from invoking the Windows “Start” menu to run programs is just the setting of a Windows registry key. Anyone with administrator privileges can reset the key and allow himself to run any program he desires. For example, he could install and run Microsoft Access to manipulate the database outside of GEMS.

Despite this low level of GEMS security, I have concluded that the system is safe because there is no useful exploit that an insider can perform that would evade detection by other procedures.

TSx security is at a higher level, largely because of its embedded architecture and its ability to digitally sign files, including its election log.

Physical security

My view is that locks, seals and tamper-evident tape are useful to deter casual intruders but are of essentially no use against insiders. That does not mean these mechanisms aren’t useful, but they are no guarantee of security. They may have a useful psychological effect on voters and would-be tamperers.

TSx conclusions

Given the fact that GEMS tabulations are unofficial in Massachusetts, the perceived insecurity of GEMS in a Windows configuration against insider manipulation is largely irrelevant. Though use of the VVPAT and parallel testing, reliable elections can be held that will not only survive scrutiny but will allow trustworthy post-election audit.

IV. AutoMARK

This section is based on the security review performed on August 3, 2006 and Election Systems & Software's response entitled "Enhancing the State of Massachusetts Election Process," dated August 15, 2005 to a request from the Commonwealth of Massachusetts for a "Voting System Equipped for Accessibility."

AutoMARK is simply a ballot marking system. It neither retains nor counts votes, but allows both sighted and disabled voters to mark optical ballots properly and reliably. The ballots themselves must be counted on other certified equipment. AutoMARK itself resembles a DRE machine in operation and appearance. It has a touchscreen and various assistive interfaces, but also a ballot scanner. The voter appears at the polling location and receives an appropriate optical scan ballot. The ballot is pre-printed with information indicating precinct, split, party (for primary elections), etc. The voter may fill the ballot in by hand, or may use the AutoMARK to mark it.

To use the AutoMARK, the voter approaches the machine and inserts the ballot (in any orientation). The machine scans the ballot to determine its ballot style and the locations of the markable areas. (Even though candidate names and office titles appear on the ballot, AutoMARK does not read them.) From the ballot style, AutoMARK is able to display an equivalent of the ballot (that is all races and questions) to the voter in a DRE style. That is, the voter is guided through the ballot, warned of undervotes, forbidden to overvote, allowed to enter write-in names, and review the ballot. The undervote warning is particularly prominent. If a voter attempts to proceed beyond a race in which he has not fully voted, an entire screen is shown warning of the undervote and asking whether the voter wants to return to the undervoted office or proceed. This warning is impossible to ignore. All of these functions are accessible to the disabled also. When the voter is satisfied with his choices, he can ask the machine to mark the ballot. It then applies ink to the correct areas on the ballot, spells out write-ins legibly and marks the required space next to each write-in. It then returns the marked ballot to the voter by ejecting it. At this point the ballot has not been counted and AutoMARK retains no record of the choices.

The voter may examine the ballot at will to determine whether it correctly represents his choices. If not, or if he changes his mind, he may turn the ballot in as spoiled, receive another, and try again. A disabled voter may re-insert the now-marked ballot back into the AutoMARK. It recognizes that the ballot is no longer blank and enters a review mode in which all of the choices on the ballot are read back to the voter through headphones. When the voter (disabled or otherwise) decides that the ballot is correct, he takes the steps necessary to cause it to be counted. In precinct-count jurisdictions, this usually involves having the voter insert the ballot into a counting device[16].

Of course, there must be an exact correspondence among (1) the printed positions and candidate names on the physical ballot; (2) where AutoMARK thinks those positions are and which candidates are associated with them; and (3) where the counting device thinks they are and which candidates are associated with them. Interfering with this correspondence is one mode of attack against ballot marking systems generally.

The AutoMARK device is a product of AutoMARK Technical Systems LLC. AutoMARK is sold and serviced by Election Systems & Software, Inc. (ES&S) and can support ES&S scanners. It is claimed that the scanners of other manufacturers can be used with AutoMARK, but the process is more difficult since the automated tools used to maintain the three correspondences listed above are not readily modified and certain manual steps are required for each election.

It is possible to set up the AutoMARK for ballot scanning by preparing election data using a system called the AutoMARK Information Management (AIMS). This can be done for each ballot style using no more than the printed ballot (or a proof copy). However, setting up AutoMARK this way essentially requires duplicate data entry, since the original ballot formatting would have been done by a different system, usually ES&S Election Data Manager (EDM). Duplicate data would have to be entered into both EDM and AIMS. To avoid the extra work and possibility of error, it is possible to import files produced by EDM directly into AIMS. This is particularly easy of the two system are running on the same computer.

The "natural" environment for AutoMARK, and the one proposed by ES&S to Massachusetts and demonstrated at the review, is to prepare ballots on an ES&S system, prepare optical scan media for insertion in the counting equipment, import the setup information into AIMS, prepare media for AutoMARK, count the ballots on ES&S scanners and transfer the totals from the scanners back to an ES&S system for tabulation. Because the AutoMARK and ES&S products are so tightly integrated during this process, it only makes sense for security purposes to consider them as one comprehensive system comprising these components:

AutoMARK with Firmware 1.1. This is a standalone touchscreen device based on an Intel XScale processor. Its proprietary operating system resides on a flash memory chip that must be physically replaced in order to update the operating system. The ballot marking application resides in a separate flash memory chip that is soldered to the motherboard. The application can be updated from a compact flash card that can be inserted into the machine. The security implications of this process are discussed below.

[16] While the disabled community generally supports AutoMARK because of its extensive assistive features, it has been pointed out that a physically disabled voter may lack the ability to carry a ballot from the AutoMARK to the counting device even if the distance is extremely short. For such voters, a jurisdiction must provide a means to do this for the voter while retaining full ballot secrecy. This can be done through the use of privacy sleeves.

AutoMARK also has volatile RAM, a keypad and an AC97 sound chip, but no hard disk. It includes a ballot scanner, an HP printer for marking ballots and a separate serial interface board to support connecting jelly buttons, and a sip & puff device. It contains no wireless devices or interfaces and no dipswitches.

Software resident on the AutoMARK was written by a subcontractor to specifications provided by AutoMARK. It's graphic user interface is written in VisualBasic.NET, the application itself is in C# and the necessary .dll's are written in C. All of the code is bundled into a single .cab file for uploading to the device via a compact flash card. A hash value is generated as part of this process, which must be known to the person attempting to install the software on the AutoMARK. The update process is discussed below.

The AutoMARK touchscreen occasionally requires recalibration, which is a process whereby a human touches areas of the screen indicated by marks produced by the machine. A threat, therefore, is deliberate miscalibration. It is not sufficient to test the machine at LAT because it could be miscalibrated later. The countermeasure is to print and verify test ballots throughout Election Day.

AutoMARK 1,1 and the AIMS 1.2 software listed below passed ITA testing on April 10, 2006 but has not yet been issued a NASED number.

AutoMARK AIMS 1.2. AIMS is a Windows XP laptop application written in VisualBasic with a Microsoft Access front end to a relational election database. This is the software/firmware with which the voter interacts by using the touchscreen. The database can be populated manually by typing data into the database through AIMS, or data produced by ES&S Unity (described below) can be imported or data can be imported from a different election management system supported by AIMS.

Compact Flash (memory) card. After manufacture, AutoMARK receives all of its software and election data from a compact flash (CF) card. The card is partitioned into separate areas for election data, log files and firmware (which may or may not be present). The election data may be read into RAM in its entirety or may be read in parts, if it is too large to fit at once. The CF card is loaded via AIMS, either by the jurisdiction, a contractor, or AutoMARK.

To reflash the firmware in the AutoMARK, a physical key is required[17]. After boot, the unit is placed in test mode and "System Maintenance" is selected. This brings up a soft keyboard. The user must enter an administrative password. Initially this password is set at the factory but can and should be changed immediately after the machines are delivered. Reflashing can be initiated from the System Maintenance screen by selection "Upload Firmware," but the user is then prompted to enter the hash value that was

[17] This is no barrier to an insider, as all keys for all AutoMARK units are keyed alike.

created when the new firmware was compiled. Even in the event new firmware is present on a memory card, it will not be loaded until the user completes this procedure.

Unity 3.0.1.0. Unity 3.0.1.0 was qualified by NASED on April 14, 2006. It is a suite of Windows XP programs used to set up, manage and tally an election. Its components are written in various languages, including C, C++, VisualBasic, Java[18] and COBOL. The master distribution disk for Unity is created during a witness build procedure by the ITA at ES&S offices in Omaha. It is not necessary to rely on the vendor for copies of the certified software, which can be obtained by authorized parties from the ITA.

Unity for Massachusetts and AutoMARK includes these components. All are Windows applications.

- Election Data Manager (EDM). This is used for election setup and ballot definition.
- Ballot Image Manager. There are several image manager programs to format ballots for different devices. The relevant one for optical ballots is Ballot Image Manager.
- Hardware Programming Manager(HPM). This formats and burns media for use in the Optech.
- Election Reporting Manager (ERM). This is a tabulation program for unofficial results that offloads results from media to obtain vote totals from scanners.
- Audit Manager. A program to inspect event logs.

AIMS and Unity can run on the same computer, in which case files created by Unity are simply read by AIMS. If they are running on different computers, transfer of files is needed.

Optech Eagle ballot scanner. For testing purposes we had the vendor supply an Optech Eagle scanner of the type already certified in Massachusetts. The scanner itself is not part of AutoMARK, but must receive the same ballot setup data as AutoMARK so it may properly interpret marked ballots.

Ballot definition information is provided to Eagle via a “memory pack,” which is created through HPM. The pack contains a small memory chip in a large plastic container. It holds only static data files, with no executable or interpreted code. The files, however, are not encrypted or subject to integrity checks. While this hole should be remedied in the future, it does not present a particular problem here because of the availability of auditing methods discussed below.

Eagle maintains a system log that records significant events, such as the zeroing of totals before an election. Suppose a conspirator contrives to annul an election by zeroing its

[18] The presence of Java, an interpreted language (for some definition of “interpreted”), would seem to violate the 2002 VVSG prohibition against interpreted code. For this reason alone I believe the prohibition to be wildly ill-advised if it is taken to exclude Java, since that language disallows dangerous programming practices that would be possible in C.

results and counting ballots of his own construction in a quantity identical to the number voted on Election Day. This exploit will be detected from the system log, which will show a zeroing that should not have occurred.

Eagle allows modem transfer of results for a polling place to a central collection point, a thoroughly bas idea that ought to be forbidden by regulation. In the event it is allowed, the capability needs to be tested thoroughly in LAT to ensure that the proper dialup number is stored on the memory pack.

Ballot Setup Procedures

Ballots are set up in EDM using familiar methods common to many election administration systems. The reason this is done is that once accomplished, the ballot styles can be exported to different types of equipment, such as DREs and optical scanners, and also to AIMS for use in AutoMARK.

For optical ballots, it is necessary to lay them out using Ballot Image Manager, which enables fine control over fonts, spacing and appearance. The Image Manager, a Unity program, operates on ballot definition files and various text files to compose the ballot, which can be edited graphically. Proofing can be done at the computer or by printing out .pdfs of the ballot and marking them up manually. When ballot have been proofed, .pdfs can be sent to a printing contractor for printing. The same data can be used by Image Manager to produce files for AIMS.

AIMS takes as input a set of six files and produces a CF disk with ballot formats, fonts, text, marking area locations, colors, precinct identifiers, and all other information necessary to recognize and mark a ballot, including audio data. AIMS provides the user with a wizard to assist in ballot definition. The CF card is protected by hash codes that cause alterations to be detected. Thus if the CF card is tampered with before it is used in AutoMARK, it will not be accepted for use. Audio files are also key-hashed with a 8-digit key.

Ballot data is made available to the AutoMARK by inserting the CF card created by AIMS into the AutoMARK and powering it up. After LAT, the card should be sealed into the unit with tamper-evident tape and a seal placed over the hinged access door covering the card.

Accessible ballots

A voter can use the touchscreen, the keypad, jelly buttons, or the sip & puff interface, with or without audio guidance. The text on the screen can differ from that presented in audio. This is necessary since non-disabled voters should not be given instructions for disabled voting. However, careful proofing is required to verify that the audio information is accurate, since this is a potential security exploit.

For five languages, AIMS and AutoMARK provide text-to-speech synthesis via a program known an Eloquence, which is familiar to the visually impaired. Eloquence also accepts phonetic spelling so it is able to pronounce words that sound different from their English spellings. For other languages, such as Chinese, the user must create .wav files, which are input to AIMS and placed on the CF card. When placed on the card, all election-specific data is wrapped into a single file called VALID.CVE, which is protected by a hash code for integrity. It is thus not feasible to tamper with the file. Even if it were, the integrated nature of the file would make it difficult to modify any given portion without detailed knowledge of the layout of the data, which varies with each election.

AutoMARK Security

AutoMARK security is good, although as a pure marking device it probably needs less security than any DRE. There are locks and/or seals on (1) the on/off switch; (2) the CF card compartment; and (3) the printer cartridge compartment. There is another seal lug on the entire unit when it is in closed position.

There is an RJ45 jack on the front side of the AutoMARK, which looks like it allows a network connection. However, this is a dummy jack for future use (curbside voting) and is not connected to anything.

Optech Security

Optech physical security is minimal and would not even serve as an inconvenience to an insider, although it probably prevents members of the public from opportunistic tampering. The locks on Optech all use the same key, including (1) the storage area for ballots, (2) the back door of the counting unit where the memory pack is held; and (3) the lock on the maintenance panel. It is true that if doors are opened during voting an alarm will sound, but this is no protection when the unit is powered off.

Eagle maintain an audit log of significant events, such as the opening and closing of polls, zeroing the election, rejection of a ballot, etc. This log is written to the memory pack at the end of the election and can also be printed locally (recommended). It can be viewed in Unity through Audit Manager. While the memory pack is not easily modified because it is difficult to obtain drivers that recognize it, the logs can be easily altered once they are returned to Unity. See the next section.

Unity Security

Because Unity is a Windows application, it is difficult to maintain it and its data in a secure manner. There is no real control over who may gain access to the laptop on which it runs, although certain physical security measures, such as keeping it locked in a safe when not in use, would help, along with administrative security means such as login passwords.

Possibly for historical reasons, the Unity files vary greatly in their resistance to attack. For example, a result of running EDM is the creation of a file name Candidat.DBF in the election database. While it was not feasible to modify this file using Notepad, it was easy using a copy of Microsoft Access. We opened the file, changed the name of a candidate from "Sherry Smith" to "Sherry Jones," and closed the file. This modification was done outside Unity. The next time Unity was run, "Jones" had replaced "Smith" with no complaint from the software. This could be remedied by password-protecting the database or (better) by encrypting it.

While the ability to modify election database files is disconcerting, it is not fatal. The reason is that the proper marking of ballots is verified at LAT and during the election by poll workers and voters. Any modification of database files before the election will be caught through diligent review. The next question is what night happen in Election Reporting Manager if votes for A are reported as votes for B through an intrusion into the database. The answer is confusion for a time but without permanent effect. The totals produced by ERM would not match the individual results produced at the polling location, and the original optical ballots are available for manual or machine recount.

A continuing problem with Unity, which ES&S has not shown any inclination to correct, is that it offers a plethora of ballot setup options which even the vendor's representatives are unable to explain. If a jurisdiction uses Unity on its own, the possibility of setting up an illegal election is significant. The vendor counters that these operations are generally performed by experts who know what they are doing, but no such person has appeared at any examination I have conducted. The remedy is to allow for prestored configuration files indicating which options are illegal for a given state. At the least, context-sensitive help could be provided. Failing all of that, the vendor, in consultation with the Secretary's office, could provide a printed checklist of valid and invalid options.

Unity's log files are unprotected and can be modified easily using Windows accessories. In Notepad, for example, it was easy to change the login ID of a person performing an operation or delete a log entry entirely. Unprotected logs have some utility but are not effective against deliberate intrusions.

In Unity it is possible to insert or alter unofficial vote totals manually. These operations are logged. However, it is possible to modify the logs to eliminate any trace of the modification, making it impossible to audit the election or explain irregularities. I did not find it possible to alter results files outside of Unity without corrupting the files. Therefore, if the log files were subject to the same level of protection the possibility of an unnoticed alteration would be reduced significantly.

Malware

For testing purposes, a ballot marking device is a completely different animal from a DRE, whatever the outward similarities may be. The reason is that a ballot marker can be tested at will during an election, something that cannot be done with a DRE. If it is suspected that candidates are being dropped from the screen or that ballots are being marked poorly or erroneously, it is only necessary for a poll worker to attempt to mark one or more ballots. After the ballot is examined, and possibly verified on the AutoMARK, it can be marked as a void test ballot and can be stored away with other supplies from the election. This is not possible on a DRE that is being used in an election, since any vote cast on it will be recorded as a vote. It is because AutoMARK does not record or tally votes that this check can be performed.

Unity is distributed to jurisdiction on an encrypted CD from the ITA. Nevertheless, we shall assume that somehow Unity, AIMS and the AutoMARK firmware have been compromised, and the nature of the AutoMARK compromise is sophisticated enough that the rogue code only operates during an election. While the effect of the Unity and AIMS exploits would be discovered at LAT, the AutoMARK intrusion would not be.

Because the effect of AutoMARK is only to mark ballots, the only effect an exploit can have is to cause ballots to be marked incorrectly. There are many ways of achieving this, from using light marks on the ballot, placing the marks incorrectly, compromising the audio ballot so the voter is misled about the candidate choice, etc. In all of these cases, the ballot will not be read as intended by the voter.

The countermeasure to this attack is regular, random testing of the AutoMARK throughout election day by poll workers at all polling locations, including verification of the audio ballot and its instructions. If any ballot is marked incorrectly, the machine can be taken out of service and replaced. The memory card and/or the machine firmware, which will remain in the machine after it is removed from service, can be investigated later.

Correspondence between AutoMARK and scanner

A threat model that must be considered is lack of synchrony between AutoMARK and the optical scanner. That is, AutoMARK and the scanner have different ideas about the layout of the ballot, either by error or design. This can occur through an intrusion into AutoMARK, but this is highly unlikely because of the high probability of detection. More likely is an attack on the ballot scanner that will cause it to behave anomalously, including failing to warn of undervote, switching votes, or simply producing skewed totals at the end of voting. The countermeasure to such attacks, aside from steps taken to make them difficult, is to perform manual recounts religiously according to statute and regulations. Precinct count scanners cannot readily be tested during an election since any proper ballot they read will be included in the vote totals. It is impractical to perform parallel testing by counting ballots on a different system at the precinct, but it is not amiss to recount the ballots later using a central count scanner, for example.

Another known attack on optical scan systems is selective adjustment of read sensitivity so that certain columns of the ballot will not be read if marks are not extremely dark. This exploit must be combined with turning off undervote protection, or voters will immediately be altered to the problem. A countermeasure is to encourage voters to choose a random orientation in which to insert the ballot so no systematic manipulation can succeed, and may in fact have an effect very different from that intended by the intruder. This countermeasure can also be combined with the one described in the preceding paragraph.

The modes of operation of Optech Eagle must be set in EDM and are acquired from the memory pack. It is not possible to change or override them at the polling place. Thus, for example, undervote warning cannot be turned off if it was initially enabled.

Ballot scanning

It is not clear, when considering Massachusetts regulations and HAVA together, exactly how optical ballots should be handled by the scanner at a polling place. Should a completely blank ballot be retuned to the voter? If so, should the scanner inform the poll worker that the ballot is blank (requiring the voter to reveal his vote, or lack of one)? Should the ballot be redstriped? What about an undervoted ballot, warning of which is not required by HAVA, even though AutoMARK warns of undervotes? Does a rejected ballot get redstriped? These questions do not impact security directly, but they do indirectly, since an intruder who interferes with these procedures may allow other exploits to go unnoticed.

AutoMARK conclusions

Because AutoMARK is a ballot marking device that does not tabulate or record votes, any effort to tamper with it will result in mischief only and will not affect the outcome of an election, regardless of the margin, if testing procedures are implemented and followed on Election Day.

V. Recommendations

This section contains recommendations to the Secretary of State concerning procedures for use of the reviewed system to mitigate perceived security risks. I have made no attempt to evaluate the probability that any particular threat will be attempted or the probability that it might succeed. Most of the threats are only practicable for insiders. The problem is that unless proper administrative procedures are adopted, Massachusetts officials will find it difficult to refute charges that tampering has occurred or that the systems are unsafe.

It is divided into two sections. “General Recommendations” apply to all three systems. “Specific Recommendations” are particular to the individual systems.

General Recommendations

This section applies to all three of the systems reviewed.

Voting is a complex process involving large numbers of people, devices and administrative processes. A system deemed sufficiently secure may not remain so if it is not operated in accordance with proper procedures. Certain procedures can remedy what would otherwise be security flaws in a system. This section contains recommendations applicable generally to all three systems reviewed and contains separate sections pertaining to systems individually.

Development of Election Day procedures that can be introduced and implemented smoothly and can still prevent and detect errors and intrusions is not a simple matter. For example, the California Secretary of State recently issued 52 pages of regulations that must be followed just when using Hart eSlate[19].

Software/Firmware Distribution. I believe this to be one of the most vulnerable processes in election management. Assuring that only certified software of known origin is inserted in and running on voting machines and tabulation/management systems is extremely difficult given the number of machines in the field, the number of jurisdictions controlling them, and the low level of security of desktop/laptop systems. Vendors need to provide ways to export and verify code, but have not yet done so. In the meantime, obtaining releases only from the ITA and sealing machines in the interim so unauthorized code cannot be inserted must be relegated to administrative controls. The availability of MD5 hash codes stored at NSRL is of some use, but only if trusted software can be used to computer the hash values of installed software. They are nearly useless for firmware because there is no reliable way currently provided to read election firmware externally.

VVPAT. Recent surveys have shown that in practice only 1-3 percent of voters actually refer to the paper trail before casting their ballots. This means that the only likely effect of a VVPAT would be to deter some potential intruders. An insider who is aware of these statistics might not be put off at all from attempting an exploit. The remedy is to perform manual audits of randomly selected precincts to compare the electronic results with the paper trail. This will reveal the effect of any malware, software error or machine malfunction. VVPATs should list candidate positions not chosen instead of merely the ones that were chosen, so it can be determined which ballot was presented to the voter. Voter should be strongly encouraged to check the VVPAT before casting a vote.

The DRE VVPATs under review maintain sequential records, which can have the effect of revealing voters' ballots in the event of a recount if the order of voters is maintained. Therefore, such records must not be produced or retained.

Opscan ballots. Random manual recount of optical scan ballots is essential, as discussed above with respect to the VVPAT. But it is also needed for another reason – there are a host of errors and manipulations that are possible on an optical scan ballot, including offsetting marking ovals, use of doctored inks, adjusting timing marks, and the

[19] See "Voting System Use Procedures for California," referenced in Appendix A.

like. Innocent errors are also possible, such as using a ballot layout that is not easily read by the scanner, and even atmospheric effects such as high humidity can influence the count. Voter error is another source of miscount. Failure to mark the oval next to a write-in, incorrectly circling a name instead of marking it properly are common conditions that can only be caught through manual inspection of the ballots. To avoid HAVA controversies, it is suggested that Massachusetts develop clear procedures for when an optical scan ballot is to be returned to the voter if an anomalous condition is detected.

Modems. Even though modem transmission of unofficial election data can be made safe, it is difficult to persuade voters that connecting a voting device through a telephone line to a remote computer whose true identity may be unknown is safe. There is a belief that the connection might be used in the opposite direction, and that malware or other corrupt files might be downloaded to the device while it is connected. Another perceived risk is that the ability of a system to receive vote data over telephone lines would allow an impostor who knew the right telephone number to introduce spurious vote totals into the system. While steps are taken to ensure that neither of these scenarios is realistically possible, the difficulty of convincing the public that the process is safe seems not to be worth the small benefit obtained from modem transfer. Therefore I recommend that all use of modems for transfer of vote totals or cast vote records be disallowed. I have made this same recommendation in each state in which I have examined voting systems.

Wireless components. The previous comments concerning modems are even more applicable to wireless transmission. Wireless components have no necessary place in election systems and should be outlawed by regulation. Voter should be precluded by regulation from bringing a picture cellphone into a polling place. Such a regulation exists in Maryland to prevent the voter from making a record of his vote that could be used as part of a vote-buying or coercive scheme.

Internet. No system or machine on which voting software is installed should ever be connected to the Internet, even for a short time, and not even with a browser. If it is necessary to update Windows or any software component, this should be done form approved hard media, not a network connection. Any transfer of results to a jurisdiction’s web server should be done via hand-carried media. No voting system or components should be connected to a jurisdiction’s LAN, since there is no legitimate reason to do so.

Dedicated systems. Given the state of Windows and voting application security, every effort must be made to prevent introduction of software attack programs. For this reason, no applications other than those necessary for voting administration should be permitted to be installed on any machine that runs election software.

Passwords. Management of election system passwords is compromised because manufacturers for reasons of efficiency either code passwords in software or distribute equipment with default passwords. While jurisdictions are told to change the passwords, they often do not do so since the software does not require it. Furthermore, the use of

Windows passwords is not under the control of election vendors at all, and weak methods or none at all are employed. Jurisdictions should be required by regulation to change all passwords after each election.

Remain alert to insider threats. The systems reviewed are protected, in some cases extensively, against external attack, that is, by people who do not have knowledge of or access to key components. The threat model employed by most vendors assume that insiders can be trusted implicitly, but there is no basis for such an assumption. The principle that should be applied is "minimal reliance." An insider should be trusted only to perform those tasks he is assigned, and should be given the minimal privileges necessary for his job. Where it is necessary to grant a high degree of privilege or access (e.g. to system administrators), extrinsic checks and balances should be imposed to minimize the possibility of unauthorized activity.

Logs. All significant election events should be logged electronically in secure (unalterable) logs. Vendors as a general matter have provided only rudimentary security for application logs, or none at all.

Pre-election testing. LAT is a critical step in detecting many of the exploits detailed in this report. While LAT cannot be depended upon to reveal malware, it is very successful at identifying "static" exploits – those that do not involve modifying executable code. All ballot styles must be verified using all assistive interfaces. In particular, all audio instructions and candidate names must be verified. After LAT, all election media must be sealed into their respective machines with tamper-evident seals that cannot be duplicated.

Parallel testing. The fundamental purpose of parallel testing is to detect malware or erroneous code that would likely escape other evaluation methods such as black box testing and code reading. For example, the compiler exploit in which the object code produced by compilation does not correspond to the source code would not be detected by a code read and could contain clever mechanisms using the onboard clock to evade discovery before an election.

The theory of parallel testing holds that to influence a statewide or national race manipulation of the software of a significant number of machines is required. Since the attacker cannot know in advance which machines might be subject to parallel test, the probability of detection can be increased by increasing the number of machines to be tested. Election malware is useless unless it manifests itself during an election, which means that well-designed parallel testing can detect many exploits previously thought to be undetectable.

Parallel testing of DREs must be done carefully because it cannot be interrupted during Election Day and partial totals cannot be produced. A rigorous parallel testing scheme is needed in which no one can know in advance which polling locations or machines will be tested and effort is required to ensure that there is no discernible difference in the voting procedures used for machines under parallel test.

Encryption. Voting systems should not store or transmit election information in the clear but they should always be encrypted. When vendors speak of encrypted files they often do not mean literally that the files are encrypted but that some protective mechanism such as a password or hash code is being employed. That is not enough.

From the above comments it can be seen that I agree with the following recommendations of the Brennan Center Report to the extent they are applicable to Massachusetts. These are:

1. Conduct Automatic Routine Audits comparing the VVPAT to the electronic records following every election.
2. Perform "Parallel Testing" on Election Day by sequestering one machine of each type in a central location and testing it using a real ballot setup during the hours of the election. The test should consist of casting at least as many ballots as would be cast on a typical machine in Massachusetts. If malware is present that activates only during the normal hours of voting, it will b detected. This is not necessary for AutoMark if the procedure described under "Parallel Testing" is followed.
3. Ban use of voting machines with wireless components.
4. Use a transparent and random selection process for all auditing procedures. For any auditing to be effective (and to ensure that the public is confident in such procedures), jurisdictions must develop and implement transparent and random selection procedures.
5. Ensure decentralized Programming and Voting System administration. Where a single entity, such as a vendor or state or national consultant, performs key tasks for multiple jurisdictions, attacks against statewide elections become easier.
6. Institute clear and effective procedures for addressing evidence of fraud or error. Both Automatic Routine Audits and Parallel Testing are of questionable security value without effective procedures for action where evidence of machine malfunction or fraud is discovered. Detection of fraud without an appropriate response will not prevent attacks from succeeding.

Specific Recommendations

Hart eSlate

Audio MBB. In the present release, there is no protection for files on an audio MBB. They are neither encrypted nor integrity checked. A temporary measure is to ensure that they are thoroughly verified at LAT then sealed into the eSlates, but there is no reason the same level of protection should not be accorded to the audio MBB that is provided for the regular MBB.

Audio ballots. A voter should be able to have an entered write-in name spelled out on audio for verification.

Voter tampering. It is not desirable to have voters attempting to open a voting unit, tamper with its cabling or experiment with door and panels. Even if there is nothing useful a voter might accomplish, the mere possibility that some sort of physical intrusion is possible is disconcerting to the public. The snap panel above the eSlate requires a lock and/or seal. If a voter lifts it, he will see removable cabling and might be motivated to experiment.

Election Day testing. The eSlate and all of its assistive interfaces should be checked randomly during idle periods throughout the day during an election. This does not involve casting votes, but activating the machine for voting and performing testing before a voter votes or activating a machine, testing it, and performing a “fleeing voter” cancellation so not vote is registered. The purpose of such testing is to detect malware that is inoperative at the beginning of voting, and to ensure that no exploit has been performed during the election.

Diebold

Firmware upgrades. The current process for uploading new firmware to the TSx (as disclosed in Hursti II) is far too loose and unauditable. At a minimum, anyone changing the firmware should be required to authenticate himself and a secure log record of the event must be made. Until the vendor provides a more secure method of performing upgrades, the remediation procedure described in the “Ballot Setup Procedures” section above under “Diebold” should be followed.

Audit log security. It is possible to edit GEMS logs outside of GEMS in a way that GEMS cannot detect, Therefore, these logs are useless for detecting intrusions, although they have legitimate other applications, such as pinpointing certain types of procedural errors. Some mechanism must be introduced to maintain the integrity of log files.

Keys and passwords. All software keys and passwords should be changed after each election.

AutoMARK/Unity

Illegal EDM setup options. The vendor must provide a mechanism to prevent users from selecting illegal or inconsistent ballot setup options which could lead to an invalid election. The reason I make this point so strenuously is that it was obvious during the review that the probability of such an event is high.

Optech physical security. Optech is of old design and affords only minimal protection from physical intrusion. Improvement is required. It is not longer acceptable in election systems to allow th same physical key to be used on multiple machines.

Unity security. Unity provides insufficient security for election and log files. they are too easy to modify outside Unity using Windows.

Conclusions

It is my opinion that all three systems reviewed can be used safely in Massachusetts if the above recommendations are followed because I have not found any credible threat that would not be detected by the VVPAT and parallel testing with routine manual audits.

Respectfully submitted,

Michael Ian Shamos

Michael Ian Shamos, Ph.D., J.D.
Consultant
Pittsburgh, PA
September 28, 2006

Appendix A
Additional Materials Reviewed

General

"Cuyahoga Election Review Panel, Cuyahoga County, OH Final Report (July 20, 2006), available at http://www.cuyahogacounty.us/BOCC/GSC/pdf/elections/CERP_Final_Report_20060720.pdf.

"Developing an Analysis of Threats to Voting Systems: Workshop Summary," October 7, 2005, published by NIST. Available at http://vote.nist.gov/threats/workshop_summary.pdf.

"Cuyahoga Election Review Panel, Cuyahoga County, OH Final Report (July 20, 2006), available at

Hart

"California Secretary of State Consultant's Report on HART INTERCIVIC SYSTEM 6.2," prepared August 4, 2006 by Paul W. Craft and Kathleen A. McGregor

Letter from Molly Terry to Michelle Tassinari re: *Request for Response for Voting System Equipped for Accessibility,* Follow-up from Security Review," dated August 18, 2006.

Securing the eSlate Electronic Voting System, by Brad Arkin, Symantec.com (2004)

"Security, Accuracy and Reliability of the Hart InterCivic eSlate Voting System purchased by Fort Bend County," by Steve Raborn, Fort Bend County Elections Administrator, updated October 7, 2005.

"Technical Security Reassessment Report Hart InterCivic Direct Recording Electronic (DRE) Device," Ohio Secretary of State, September 16, 2005

"Voting System Examination Hart InterCivic, Prepared for the Secretary of State of Texas
James Sneeringer, Ph.D., Designee of the Attorney General," June 26, 2006.

"Voting System Use Procedures for California: Hart Voting System 6.2," Regulations of the California Secretary of State.

Diebold

"Certification Test for the Diebold Election Systems, Inc. (DESI) GEMS 1.18.24, AV-OS 1.96.6, AV-TSX 4.6.4 with AccuView Printer Module, and Voter Access Card utilities: Executive summary," by Bruce McDannold, Election Systems Division, Office of the Secretary of State of California, November 11, 2005

"Commonwealth of Pennsylvania Department of State Amended Certification of the Diebold Election Systems' AccuVote TSx Direct Recording Electronic Voting System and Certification of the AccuVote OS Optical Scan Central Count Reader CC 2.0.12." dated January 17, 2006.

"Diebold Election Systems," report of Thomas Watson, an examiner for the State of Texas, dated January 19, 2006.

"Diebold Election Systems, Inc. AccuVote OS, AccuVote TSX, GEMS Election Management System and Accessories -- An Evaluation Prepared for the Secretary of the Commonwealth of Pennsylvania." by Michael Ian Shamos, December 2005.

"Diebold Election Systems, Inc. GEMS Version 1.18.24 – Staff Review and Analysis," California Secretary of State Elections Division, November 15, 2005.

"Diebold TSx Evaluation SECURITY ALERT: May 11, 2006 Critical Security Issues with Diebold TSx," by Harri Hursti ("Hursti II").

Directive Concerning the Installation of Files Regarding the Diebold AccuVote-TSx Electronic Voting System Issued by the Secretary of the Commonwealth (PA), May 2, 2006.

"Security Analysis of the Diebold AccuVote-TS Voting Machine," by Ariel J. Feldman, J. Alex Halderman and Edward W. Felten (Sept. 13, 2006) (the "Princeton Report"). Available at http://itpolicy.princeton.edu/voting.

"The Black Box Report SECURITY ALERT: July 4, 2005 Critical Security Issues with Diebold Optical Scan Design," by Harri Hursti ("Hursti I")

AutoMARK

California Election Procedures Manual for the ES&S AutoMARK Voter Assist Terminal, published by AutoMARK Technical Systems, LLC (2003).

"Certification of ES&S AutoMARK Voting System Version 1.0 Introducing the AutoMARK Technical Services' (ATS) AutoMARK Voter Assist Terminal (VAT) Release Ver. 1.0 In Conjunction with the ES&S Unity 2.4.3 Election Management System and Selected Optical Scanners," by Bruce McDannold, Election Systems Division, Office of the Secretary of State of California, June 7, 2005.

“Election Systems & Software AutoMARK Voter Assist Terminal, version 1.0, AutoMARK Information Management System, v 1.0, MDB, version 1.0.40: Staff Review and Analysis,” prepared by the Secretary of State Elections Division, June 9, 2005.

“ES&S AutoMARK™ Election Day Checklist,” by Election Systems and Software