Systemic Issues in the Hart InterCivic and Premier Voting Systems: Reflections Following Project EVEREST*

Kevin Butler, William Enck, Harri Hursti†, Stephen McLaughlin, Patrick Traynor, and Patrick McDaniel
Systems and Internet Infrastructure Security Laboratory
Department of Computer Science and Engineering
The Pennsylvania State University
{butler,enck,smclaugh,mcdaniel,traynor}@cse.psu.edu, †hursti@hursti.net

Abstract

In response to growing concern about the security and integrity of elections in the state of Ohio, Secretary of State Jennifer Brunner set in motion a comprehensive study of the electronic voting equipment used throughout the state. Known as Project EVEREST (Evaluation and Validation of Election Related Equipment, Standards and Testing), this study attempted to assess the risks associated with Ohio's current voting systems. In this paper, we discuss the systemic vulnerabilities and weaknesses discovered during the academic team's evaluation of the Hart InterCivic and Premier Elections Solutions (formerly Diebold) hardware and software. We begin by describing a methodology for identifying and confirming vulnerabilities aimed at preventing vendor deniability so prevelant in voting systems analysis.

Both systems' studies began with an independent analysis of known vulnerabilities and quickly expanded. The Hart analysis expanded on previous findings and discovered 27 new vulnerabilities. Most notably, we discovered a large swath of undocumented functionality within the Hart system that could be highly dangerous in an election environment. Like previous evaluations, our analysis of the Premier system notes that the platform is plagued by systemic security issues; however, our evaluation goes further.

We observe that even in the presence of over half a decade of evaluation, these systems do not appear to be improving—in some cases reintroduce failed designs indentified by past studies. Weaknesses in both systems make the security of elections almost entirely reliant upon the universal and consistent enforcement of physical procedures - a difficult process at best. Most critically, this study, the results of which this paper reflects upon, demonstrates that such problems are not solvable by simple patches; rather, they are the result of fundamental misunderstandings about the use of proper security practices in systems.

*The comments made in this paper reflect the observations made during the course of our investigation and do not necessarily represent the opinions of the Secretary of State or the State of Ohio.

1 Introduction

The Help America Vote Act (HAVA) of 2002 mandated the widespread use of electronic voting machines across the United States. As a result, the technology supporting elections nationwide changed nearly overnight. However, concerns about the security and integrity of elections conducted using available products arose nearly as quickly. As an increasing body of independent reports painted a bleak portrait of such systems, a number of state and federal officials began sanctioning official evaluations. Responding to declining public confidence in electronic voting machine technology the state of Ohio initiated an investigation of the risks associated with systems used across the state. Project EVEREST (Evaluation and Validation of Election Related Equipment, Standards and Testing) [23] brought together teams from academia and industry to develop a comprehensive understanding of the vulnerabilities and risks in the three systems used in Ohio: Premier Elections Solutions (formerly Diebold), Hart InterCivic, and Election Systems and Software (ES&S). This paper focuses on the experiences of the academic team at the Pennsylvania State University, who evaluated the Hart InterCivic and Premier systems and discovered systemic weaknesses in both.

One of the critical discoveries in the Hart InterCivic portion of the study is that the full functionality of the Hart system is currently unknown. We found numerous functions and system configurations that were not documented and not described in previous studies, and whose purpose was non-obvious. The vast majority of these remain unstudied for lack of reviewer time, but the functionality we discovered allows for exploitation of the system in novel ways, such as allowing an attacker to remotely "script" DRE voting machines to cast arbi-

trary ballots as the attacker chooses. Furthermore, certain interfaces–in particular, those in the election tally software–were designed to be augmented at run time with additional software. Because these interfaces were apparently designed to allow previously unknown software to be arbitrarily introduced into the live system they represent a source of potential vulnerability, the magnitude of which is unknowable.

The Premier portion of the study is unique in its analysis of the Election Media Processor (EMP) and ExpressPoll (two new, previously unreviewed components), the Voter Card Encoder (VCE) (which received limited attention in previous reports), and Verdasys Digital Guardian (a third party tool used in Ohio to protect the GEMS server). Our findings are consistent with those of previous studies. When taken as a whole, this and previous studies highlight a central point of concern: there is a demonstrative lack of improvement in the security of Premier election equipment. Initial reviews of the Premier system were undertaken as early as 2001. After six years of reviews and many new software and hardware upgrades, reviewers not only continue to find the same and similar problems as reported earlier, but continue to uncover new serious issues. Thus, the only reasonable conclusion that one can draw is the engineering approaches undertaken by Premier to eliminate previous problems and avoid new ones are failing.

The flaws in the both the Hart InterCivic and Premier systems place the security of an election almost entirely on physical procedures. Our analysis suggests that when those practices are not uniformly followed, it will be difficult to know whether or not attacks occur. Even when the attacks are identified, it is unlikely that the resulting damage can be easily contained and the public's belief in the accuracy and fairness of the election restored.

The review team feels strongly that the continued issues of security and quality in both systems are the result of deep systemic flaws. Thus, we agree with previous analyses and observe that the safest avenue to trustworthy elections is to reengineer the Hart InterCivic and Premier systems to be secure by design.

2 Contribution

The EVEREST study significantly advanced public knowledge of both the Hart InterCivic and Premier electronic voting systems. This report abstracts many of the new vulnerabilities discovered and reflects upon our experiences from the review process. In the section, we enumerate our contributions.

With respect to the review process, in this paper:

- *We describe our methods of bootstrapping our knowledge of the Hart and Premier electronic voting systems.*
- *We provide recommendations for levels of documentation produced by future studies.*

While we received similar equipment and versions of the Hart InterCivic system as the California Top-to-Bottom Report (hereafter referred to as the CA TTBR) [4], the EVEREST study discovered substantial new vulnerabilities and attacks. Our main contributions are as follow:

- *We describe a vast amount of previously unknown functionality in the Hart system.*
- *We demonstrate that not only is it possible to replace Hart system firmware with malware, but also that concrete methods of exploitation are available.*
- *We describe how an attacker may subvert all back-end protections in the Hart Election Management System.*

The analyzed Premier systems were also of similar version as the CA TTBR; however, we were also provided the previously unreviewed components. In this paper:

- *We classify Premier issues into systemic classes of failure, showing how newly discovered vulnerabilities continue existing trends.*
- *We extend previously known attacks to the EMP, ExpressPoll, and VCE system components.*
- *We investigate the limitations of the Verdasys Digital Guardian security software used by the state of Ohio to defend the GEMS server.*

3 Methodology

Without an effective plan for evaluating these systems, conducting a truly comprehensive study is extremely difficult. Accordingly, while we believe the results of this study offer significant evidence of the systemic security problems with the Hart and Premier electronic voting systems, we expect future studies to follow. We therefore offer insight into our own methodology so that future researchers will be able to quickly and accurately evaluate such systems.

Key in evaluating any electronic voting solution is the understanding of that system's architecture. Simply understanding the components of a system is only the first step; more importantly, it is necessary to identify the relationships between components and understand both intended and unintended interactions. Previous evaluations of the same or similar equipment [22, 20, 21, 24, 3, 17, 1, 2, 13, 12] are an excellent means of bootstrapping this process. Our access to source code and equipment was

severely time-limited, while contract negotiations and legal issues took up an enormous quantity of time at the beginning of the study period. Future studies will likely run into similar issues; thus we recommend that preparing as much as possible by thoroughly examining these previous studies. Because the particular configuration of an election system can vary between states, vendor-provided training is also recommended. We participated in day-long, high-level sessions on each system run by Hart and Premier. Combined, these approaches allowed us to understand how these systems are configured and operated in Ohio before examining a single line of source code or performing any red teaming.

When source code and equipment became available, they were voluminous in scope and quantity. We received dozens of pieces of equipment and substantial codebases–over 360,000 lines of code in the Hart system and over 330,000 lines of code in the Premier system. The sheer magnitude of the code, documentation, and number of components, coupled with the limited amount of time to perform the study, necessitated understanding system internals as quickly as possible. In the first phase of the study, we sought to independently confirm previous vulnerabilities for two reasons. First, locating such vulnerabilities allowed us to develop a more intimate knowledge of the system and understand critical interfaces and functions in the code. Understanding the kinds of problems known to be in these systems and the context in which they exist helped to point us to similar problems in previously unevaluated components. Secondly, by offering an independent evaluation of previous work, we provide more evidence to the public that such problems do in fact exist.

The EVEREST study was a opportunity in that the each teams was given simultanous access to source code and hardware. Accordingly, we were able to locate weaknesses in the software and demonstrate them on the machines themselves. Helpful to this process were tools such as a complimentary copy of Fortify SCA [11] and Doxygen [25], a freeware automated functional graph creator. Buffer overflows, authentication circumvention and a wide variety of other attacks were then carried out against all of the evaluated systems. This procedure of validation shows the ease with which many attacks can be executed, therefore we recommend future studies include similar validation in their evaluation to help combat "lack of real-world conditions" claims.

It is notable that while a tool such as a source code analyzer (e.g., Fortify) is very useful in limited circumstances, the vast number of errors that it reported across the entire codebase provided us with too much output to feasibly examine every condition. A further limitation of such a tool is that its close focus on individual routines, while useful for finding specific errors, is not initially conducive to understanding broad designs of the system architecture. Understanding these details required a more comprehensive and analytic approach. To this end, we focused intently on inputs to components such as the user interface and closely examined cryptographic APIs and structures to understand how secure information was handled. This was critical for the second phase of our study, determining new vulnerabilities within each system and examining the new equipment supplied by Premier.

Finally, we recommend that future evaluators replicate our procedure of creating a detailed unredacted description for *every* vulnerability, independently confirmed by another member of the team. More precisely, our issue discovery and confirmation process proceeded as follows.

1. Identify a potential vulnerability or area of concern.
2. Perform a detailed source code analysis and/or exploit the vulnerability.
3. Write a detailed description of the vulnerability including enough information to replicate the experiment.
4. Acquire independent confirmation from a team member not involved in the discovery of the vulnerability.

Only after independent confirmation could a vulnerability be included in the report. The documentation played a key role in the process, because it allowed us to convince the rest of the team that a vulnerability exists. Possibly more important, it allows future analyses and third parties to recreate our work. As such, the descriptions should contain line numbers, code samples and file and function names. While the previous studies contained private portions for some vulnerabilities, lack of access to such information for all vulnerabilities required our team to spend significant time in the confirmation phase. Note that we were given limited access to the private appendices for some of the previous Premier studies; however, it occurred during the closing days of the study and therefore provided minimal value. No such information was provided for the Hart Systems. As a result, we spent many hours in some cases understanding esoteric and obscure code structures, often scattered amongst dozens of files, to track down what often turned out to be minor pieces of information that were nonetheless essential for confirming a vulnerability. Having access to the reports detailing how to find some of these vulnerabilities would have led to faster confirmations and led to faster understanding of the system, saving a large amount of time. To avoid these problems, future similarly sanctioned studies must be given unregulated access (under appropriate nondisclosure agreements) to private reports containing

specifics for each vulnerability at the *beginning* of their evaluation to ensure that the majority of the study can be spent on previously unevaluated components.

Our assessment methodology was particularly effective - in nine weeks, this study doubled the number of publicly known vulnerabilities in Premier systems and found over 25 new vulnerabilities in the Hart system. In fact, as the evaluation approached its end, the rate of vulnerability discovery continued to *increase*. Given more time, it is our firm belief that additional significant vulnerabilities would continue to be found. By structuring future studies in a similar manner, we believe that even more comprehensive evaluations can be carried out successfully.

4 Hart InterCivic Analysis

In this section, we describe the results of our evaluation of the Hart InterCivic system. We examined previous studies of the Hart system [2, 5, 24], drawing primarily from the Source Code Review prepared for the California Top to Bottom Report [17] (hereafter referred to as CA TTBR) to confirm, and often expand on, existing vulnerabilities. In addition, we discovered over 25 previously unreported vulnerabilities that may provide numerous opportunities to manipulate election outcomes or cast doubt on legitimate election activities. Such vulnerabilities are exploitable under election conditions, and often require minimal physical access to equipment or information. These vulnerabilities are a result of the following failures of the Hart system's design, implementation, and practices:

- *Failure to effectively protect election data integrity* - Virtually every ballot, vote, election result, and audit log is forgeable or otherwise manipulatable by an attacker with even brief access to the voting systems. These vulnerabilities place enormous burdens on physical procedures.

- *Failure to eliminate or document unsafe functionality* - There are a number of largely undocumented features in the system that are highly dangerous in a production election system. For example, existing features allow an attacker to remotely "script" DRE voting machines to cast votes as the attacker chooses, to allow a single (or photocopied) voter ballot to be counted many times, and to print prevoted ballots that will be accepted by voting equipment. Note that all of these activities are not attacks *per se*, but are the apparent intended use of existing Hart system features. These features are available during a live election.

- *Failure to protect election from malicious insiders* - The protections in the Hart system that are intended to prevent election officials, poll workers, and vendor representatives from using dangerous features or modifying election data are circumventable. Attackers with access to the system can quickly recover critical system passwords, extract cryptographic keys, and reproduce security hardware. These artifacts are the "keys to the kingdom" that can be used to forge election data and compromise nearly all of the Hart election equipment.

- *Failure to provide trustworthy auditing* - The auditing capabilities of the Hart system are limited. Those features that are provided are vulnerable to a broad range of attacks that can corrupt or erase logs of election activities. This severely limits the ability of election officials to detect and diagnose attacks. Moreover, because the auditing features are generally unreliable, recovery from an attack may in practice be enormously difficult or impossible.

We begin by overviewing the Hart InterCivic voting system architecture as used in Ohio and then visit each failure in turn.

4.1 Hart InterCivic Architecture

We briefly overview the Hart InterCivic Voting System by walking through a sample election procedure (as typical in Ohio); a more detailed description can be found in the EVEREST report [23]. Refer to Figure 1 for component orientation and interaction; all county headquarters components run on the Windows 2000 Server operating system.

Before the election begins, the eSlate Cryptographic Module Manager, or *eCM Manger* (5), is used to generate a cryptographic master key, which is stored on every eCM token (simply a Spyrus Rosetta USB cryptographic token) used in the election (i.e., there is one master key for a county). The Ballot Origination Software System, or *BOSS* (1) creates an election database, including precinct and race definitions and the corresponding ballots for every county precinct. BOSS then writes the data to PCMCIA storage cards called Mobile Ballot Boxes, or *MBBs* (7); one MBB is written for each Judge's Booth Controller, or *JBC* (8), and *eScan* (9) used in the county, along with one additional MBB to be used by *Ballot Now* (2) for recording absentee ballots. Meanwhile, in the warehouse, the System for Election Records and Verification of Operations, or *SERVO* (6), software is used to reset the memory of all JBCs and *eScans* (10) and to reset their vote count to zero. SERVO is also used to transfer the shared key from an eCM onto the JBCs and eScans.

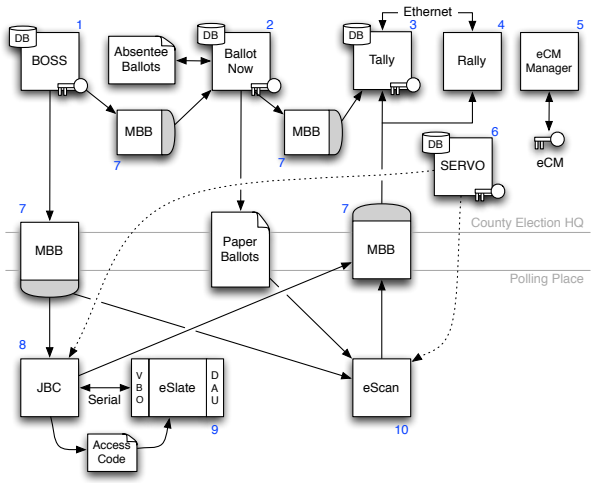

Figure 1: The major portions of the Hart system architecture and some of the connected components. Unless indicated otherwise, solid lines indicate a physical relationship between components, i.e., components are either physically connected or must be physically transported to interact with each other. Dashed light lines represent connections that take place between components outside the course of an operational election in progress.

On election day, voters using the eScan fill out paper ballots and enter them in the machine, which tallies the results and writes them to an MBB. Voters using the eSlate DRE first go to a poll worker, who provides them with a 4-digit access code generated by the JBC. The voter enters the code into the eSlate, which provides on-screen instructions for casting a ballot. The voter can verify their vote by checking the paper trail printed on the Verified Ballot Option (VBO) attached to the eSlate. The confirmed vote is recorded to the JBC's MBB and internal memory, and the internal memory of the eSlate. Absentee ballots are processed by Ballot Now, which records results to an MBB.

After the election, MBBs are retrieved and taken to election headquarters. In Ohio, these are either loaded directly to a machine running *Tally* (3), or onto a machine running *Rally* (4) that is on an internal private network with Tally. Tally tabulates results from each of the MBBs and produces an election result database along with a variety of reports. After election night, the audit logs and vote records from the JBC, eSlate, and eScan machines are backed up by SERVO and the firmware is verified.

4.2 Election Data Integrity

The Hart system displays serious shortcomings that prevent it from protecting the integrity of election data. A single shared key is distributed throughout a county, which is relied upon to provide the majority of cryptographic protections (Issue 22, CA TTBR). This key may be used to forge arbitrary election data (Issue 24, CA TTBR) and is trivial to retrieve by an adversary with a modicum of physical access to precinct or back-end equipment. For example, the key may be downloaded to a file from eCM Manager (Issue 23, CA TTBR), or from an eScan or JBC over an Ethernet or parallel connection, respectively (Issue 24, CA TTBR). In this section, we expand on the new vulnerabilities we discovered relating to the integrity of election data. Namely, we discuss how data on the MBB, the primary method of retrieving vote totals, may silently drop votes for a time period that the attacker may determine. We also discuss an adversary's abilities to bypass protections in the EMS applications to perform arbitrary operations and, in the case of Tally, generate arbitrary vote count totals.

4.2.1 MBB Images

Hart systems use a mobile ballot box, or MBB, as the main conduit for recording and tabulating votes. The data on an MBB, also known as its "image", is stored on a PCMCIA card. At a low level, it is a persistent storage device that can have data removed from it by simple copying (e.g., using the UNIX `cpio` utility). An attacker can use this feature to manipulate an election. If the attacker can access the MBB and copy data from it, then come back later and write the copied data back to the MBB, all of the votes cast in the period after the MBB was initially replaced will be effectively erased from it (EVEREST, 20.1.1). From the MBB's standpoint, there is no indication that any votes occurred during this time period, with only the internal logs of the precinct equipment (the eScan or the JBC and eSlate, depending on whether optical-scan or DRE voting is used) maintaining this record. Because the MBBs are used for the final vote tally, unless a recount is performed, the chances of the missing votes being caught are small. This attack was briefly alluded to in the California red team report [2] but was non-specific in terms of the threat or the attack.

We validated this attack by developing a program that would retrieve the MBBs image. This was written in approximately 100 lines of C code. We then implemented the attack by demonstrating the removal of votes from the MBB. As an example, we ran a number of ballots through the eScan and determined the number that should have been processed. We then removed the MBB, wrote the image to file, and replaced the MBB. After running more ballots through the eScan, we removed the MBB and restored the image; when the eScan was tallied, ballots cast after the image was made were missing.

4.2.2 Bypassing of Passwords

The EMS applications BOSS, Ballot Now, SERVO, and Tally, require a username and password to log in. These credentials are stored in a security database associated with each application. We were able to connect to the database though an attack described in Issue 15 of the CA TTBR, where the database passwords are kept in configuration files that are easily read. At this point, we can delete the usernames found in the database. Once this has been done, the applications may be opened and a new administrator account created. The application can then be logged into with administrative access (EVEREST 20.1.3).

With supervisor access to these applications, it is possible to modify the processes of ballot definition and creation, tallying of votes, and maintenance of the voting equipment. Ballots may be arbitrarily printed in Ballot Now, and the audit logs for voting equipment may be cleared (see Section 4.5.3 for more details).

The Ballot Now application contains an additional password-based vulnerability. Ballot Now connects to a back-end Sybase database, which runs a stored procedure when a user logs in, taking a hash of the username and password as input to be validated. By replacing the stored procedure definition, found in the security database, with a single line of code, we were able to allow any user to log into Ballot Now with any username and password, or with none at all (EVEREST 20.7.1).

4.2.3 Potential Third-Party Software Vulnerabilities

The vulnerabilities listed above point to a general design issue with the Hart system: reliance on third-party functionality for a large number of sensitive operations. For example, the eCM tokens in use are Spyrus Rosetta USB devices with seemingly no validation of the cryptographic operations done by Hart. A Cryptoki API is exported that provides signing and encryption operations that are necessarily opaque; however, all of the trust in these tokens is reliant on the correct implementation of cryptographic functionality within these tokens, something that is difficult to validate when dealing with COTS hardware.

A potentially greater issue along these lines is the Hart system's extensive use of functionality from the underlying Windows operating system. In particular, the generation of eCM signing keys relies extensively on the `CryptGenRandom` function called by the Windows 2000 random number generator. Recent work by Dorrendorf et al. has shown that this generator contains vulnerabilities [9]. It is possible to find all previous states of the generator in about 19 seconds on a Pentium IV computer, and future keys may be predicted due to a lack of both forward and backward security in the PRNG. This vulnerability is outside of the scope of Hart's software to fix, meaning that there is a reliance on the willingness of outside vendors to solve these sorts of potential vulnerabilities. The issue of potentially insecure back-end Windows systems has been discussed in previous reports on the Hart system, notably Issue 20 of the CA TTBR.

4.3 Unsafe Functionality

During our source code analysis of the Hart system, we identified features that were undocumented and largely, unsafe. The majority of these features are likely used for testing purposes, and have been left in the production versions of the software. Instead of being isolated to test interfaces, these features are sprinkled throughout legitimate interfaces used for ballot generation, cryptographic key management and voting machine maintenance, making them difficult to remove. Our analysis of these features shows that they are unfit for inclusion in production level software, and that no equipment incorporating them should ever be deployed in the field. We now review unsafe and undocumented functionality in each Hart component, both at the polling place and county headquarters.

4.3.1 eScan

One example of unsafe functionality being seamlessly added to a necessary interface is the eScan's configuration file. This file can be retrieved and uploaded via the eScan's Ethernet port, as describe in Issue 3 of the CA TTBR. The protocol used to communicate over this port is simple and has no facilities for authentication between the eScan and any host to which it is connected. The configuration file is obtained by issuing a single numerical command to the eScan, and uploaded by issuing a similar command and sending the file. We wrote programs to do both using standard sockets APIs.

The default configuration file contains an option to "allow duplicate ballots", which is commented out. We uncommented this option and uploaded the file. We then carried out an election using photocopies of a single filled in paper ballot. With the option enabled, the ballots were accepted by the scanner and the vote totals stored to the MBB (EVEREST 20.3.6). These votes were counted and reported on the eScan's paper printout and were tallied by Tally. Note that without enabling the duplicate ballots option, any copy of a paper ballot is rejected by the scanner after the first instance is scanned. Along with the photocopied ballots, we were also able to attach a piece of tape to a single ballot and retrieve it from the eScan after scanning, allowing us to vote multiple times with a single ballot, albeit in a more conspicuous manner

than with photocopied ballots (EVEREST 20.3.9).

It is still possible however, to detect that multiple duplicate ballots have been scanned. The eScan's audit log contains the serial number of every ballot scanned, allowing a vigilant auditor to uncover the duplicate ballots. This could be avoided with the assistance of a malicious poll worker erasing the eScan's audit logs at the polling place as described in Section 4.5.3. Even if the audit logs are deleted, the duplicate ballots can be discovered by examining the bar codes on each paper ballot in the ballot box. This too is undetectable if the above approach of retrieving a scanned ballot is used.

We also discovered an undocumented telnet server running on the eScan (EVEREST 20.3.2). The server is the Microsoft Windows CE Telnet service. Most likely, the server started by default, suggesting a lack of proper configuration of the underlying OS. While we were not able to login to the telnet server, vulnerabilities have been discovered in other Microsoft telnet servers [6, 7], indicating that it may be possible to gain control of the eScan by exploiting the server. While disabling the server may easily mitigate this issue, the extent of the misconfiguration of the OS underlying the eScan software remains unknown.

4.3.2 JBC and eSlate

The eSlate and JBC also have a significant amount of unsafe and undocumented features integrated into their standard functionality. The most outstanding of these is the ability of the JBC to receive and issue "soft" button presses (EVEREST 20.4.3).

These are button presses not created by the actual buttons on the JBC or eSlate, but encoded in a communication protocol. The JBC receives these soft button presses via its parallel port and can forward them an attached eSlate via its serial port. Upon receiving a soft button press, the JBC will decide whether to process it or relay it to an attached eSlate.

When a device receives a soft button press, it first makes a call to the underlying OS to insert the button press as a regular keyboard interrupt. The OS then delivers the keycode to the application for processing. This method of delivery makes it impossible for the keyboard input processing components of the JBC and eSlate to determine whether a button press is from the keyboard or an external device.

Using the soft button press functionality, we carried out a "Ghost Voting" attack on the JBC and eSlate. This attack allowed us to connect a laptop to the JBC's parallel port and automatically vote for selected candidates an arbitrary number of times. The laptop was running a program we wrote that works as follows:

1. Obtain a voter code from the JBC's parallel port.
2. Enter the voter code into the JBC by sending soft wheel turns over the serial cable connecting the JBC to the eSlate.
3. Send the appropriate soft button presses and wheel turns to the eSlate to vote for the desired candidates.
4. Complete voting and approve the VVPAT
5. Repeat

This program contained approximately 200 lines of new code, and required slightly over two hours to complete. With it, we were able to enter a registration code, vote and approve the VVPAT once every 20-30 seconds. Note that no authentication was required to send the soft button presses. Each vote was recorded on the eSlate's VVPAT, the JBC's unofficial printout and the cast vote records stored on the JBC's MBB. These vote records were tallied by Tally and there was no evidence in the audit logs suggesting that malicious behavior had occurred. Along with the soft button presses, step 1 of our program also relied on the ability to generate voter codes via the JBC's parallel port as described in Section 4.4.1.

4.3.3 EMS

Another example of undocumented and unsafe functionality is the ability of the Hart Election Management System (EMS) applications (BOSS, Ballot Now, Tally, SERVO and eCM Manager) to silently write all or part of the eCM key to a debug file in cleartext (EVEREST 20.2.1). By silently, we mean without any notification through the user interface that the key will be stored.

This functionality is not a part of the EMS applications proper, but of the Spyrus library they use to read and write the eCM tokens, which are Spyrus Rosetta USB tokens. When any EMS application reads the key from the token, the Spyrus library checks a specific entry in the Windows registry for a path to a debug file. If this entry is found, 16 out of 40 bytes the key are saved to the debug file in plaintext. When the eCM manager writes the key to the token, the Library writes the entire 40-byte plaintext key to the debug file. An attacker with very brief access to an EMS system could enable the Spyrus registry entry and later check the contents of the debug file to obtain the county wide key.

4.3.4 Ballot Now

A final example of unsafe features intentionally added to the Hart systems is the Ballot Now's "Autovote" feature (EVEREST 20.7.2). Autovote allows for the creation of pre-filled-in paper ballots. Once again, this feature is enabled through Windows registry entries. Once these entries are enabled, Ballot Now displays the Autovote menu option when started.

The Autovote menu allows the Ballot Now user to choose the number of pre-filled-in ballots to print. The user has no control over the selected filled in entry for each contest, however, the selected entries are uniformly distributed. This allows an arbitrary number of ballots with the desired results to be printed with the overhead of some ballots with undesired results that may simply be discarded.

Paper ballots generated by Autovote initially say "Autovote" on the front and back, making them conspicuous and easy to detect in an audit or recount. We were able to overcome this by installing a PNG printer driver on the Ballot Now machine. This driver allows ballots to be printed to PNG files as opposed to paper. We could then open the files in an image editor, remove the Autovote label and print them. Aside from the label, Autovote ballots are identical to regular ballots. We conducted a normal election and an election with Autovote ballots, and could not identify any differences in the eScan unofficial printout, the audit logs, or the cast vote records on the eScan's MBB.

Autovote could be used in tandem with the eScan's duplicate ballot feature to perform a ballot stuffing attack. Using Autovote ballots is advantageous over using photocopies, as each Autovote ballot has a unique serial number, and thus cannot be differentiated from legitimate votes in an audit.

4.4 Malicious Insiders

The Hart system fails to provide adequate protection against malicious insiders. While some protections have been put in place, they are easily bypassed. As a result of this the majority of the security in the Hart system is dependent upon insiders correctly following procedures.

Election insiders often have equal or greater political motivation and ties than voters, thus we must assume that insiders will attempt to compromise or cast doubt on election results, interfere with the election process and coerce voters to vote a certain way. For our purposes, insiders include election officials, normally located at election headquarters and poll workers, normally located at the polling place.

4.4.1 Polling Place

Poll workers may collude with voters to influence election results or monitor them to determine vote choice. They may also attempt to interfere with the voting process or take measures that would cast doubt on the results. We now review several vulnerabilities in the Hart system leaving it open to attack by poll workers.

eScan We were able to exploit a number of vulnerabilities in the eScan that could give election insiders the ability to compromise election results and voter privacy. Some of these were a result of a lack of physical security. We were able to replace the eScan's internal flash memory card containing the eScan executable and configuration file with only a screwdriver in about 2 minutes. After replacing the card, we were able to boot the eScan into the Linux operating system as shown in figure 2. This simple attack gives a single poll worker with a few minutes of unobserved access to the eScan to undermine all votes cast at a precinct (EVEREST 20.3.1).

While opening the eScan to replace the memory card, we broke three tamper evident seals. While such seals may prove that a machine was opened, a preventative measure is preferable. A poll worker may intentionally break these seals in order to cast doubt on election results. It has also been shown that tamper evident seals do not always correctly show that tampering occurred [19].

Insiders may also wish to use their access to ballots to determine voter choice. This can be done with the eScan due to the design of its ballot box (EVEREST 20.3.4). The eScan's scanner sits on top of its ballot box, which is essentially a plastic tub. When a ballot is scanned it is then dropped into the box. No measures are taken to disturb the order in which ballots are scanned, allowing a malicious poll worker to note the position in which certain votes are cast and then relay these positions to an election official with access to the ballots. We observed ten numbered ballots as they were cast with the eScan, and verified that the vote order was preserved.

JBC Normally, the voter access codes needed to vote using an eSlate are generated by the JBC and printed. It was previously shown that these voter codes could be rapidly generated from the JBC's serial port during the early voting phase of an election (Issue 4, CA TTBR). This was accomplished by disabling the JBC's printer through the menus. Rapid generation of voter codes allows a poll worker to collude with voters to vote multiple times. In our investigation of this vulnerability, we found that contrary to initial findings that the maximum number of outstanding access codes was 150, we were able to generate over 10,000 access codes within an expiration period (set to 30 minutes by default, but configurable to as high as 16 hours) [18] This ballot stuffing attack is limited however, in that a large number of votes during early voting would likely be conspicuous and easily identified as fraudulent. For this reason, we investigated ways to rapidly generate voter codes during the normal voting period.

It is not possible during the regular voting period to disable the JBC's printer through its menus. We discovered that requesting an "Access Code Report," over the

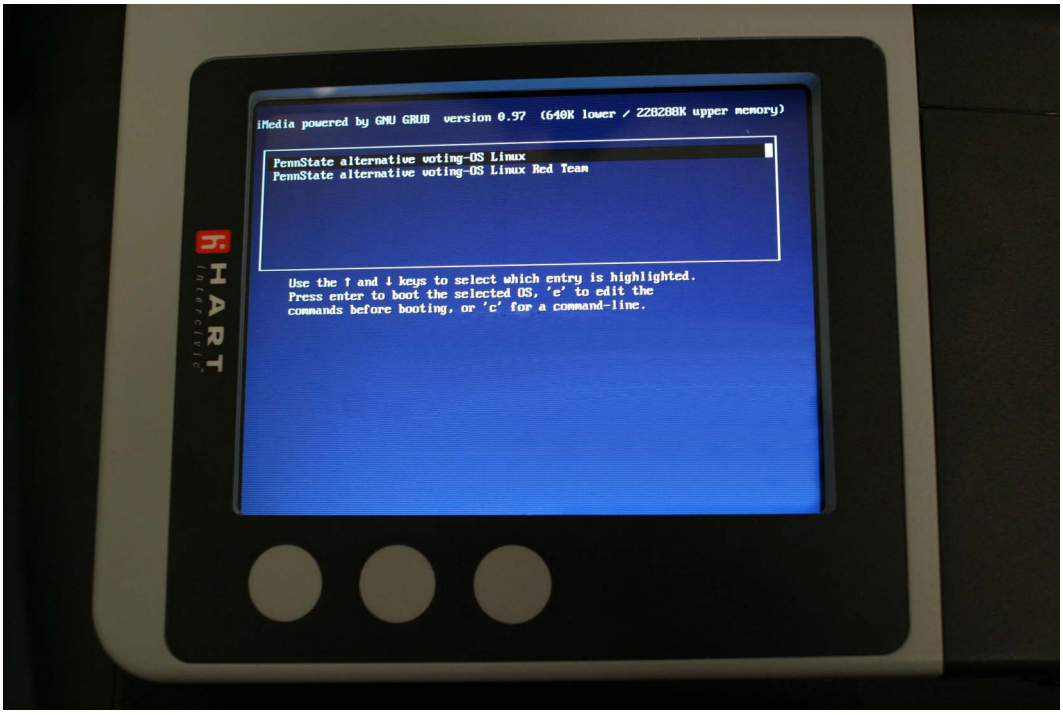

Figure 2: The eScan booting an alternative operating system

serial interface while there was no paper in the printer re-enabled the menu option to disable the printer. Once this option is available, the printer can be disabled and voter codes can once again be generated rapidly (EVEREST 20.4.1). This is an example of bad exception handling, which is seen elsewhere in the Hart system, such as in the case when a user database is empty allowing the creation of administrator accounts as described in 4.2.2.

4.4.2 Election Headquarters

The Hart system places nearly complete trust in the physical security and the procedures at election headquarters. The lack of security in the Hart components located at election headquarters is in direct conflict with the total power that election officials have. One of the most crucial components of the back end system is Tally, the vote tallying software. Improper use of Tally can lead to partial or total corruption or loss of election results.

Tally maintains a database containing the state of all MBBs used in an election. If an MBB is marked as tallied in this database, Tally will refuse to count the results on that MBB. Thus deliberate or accidental tallying of an MBB by a poll worker can lead to the results on the MBB not being counted. Note that because the state of the MBB is stored in the database and not the MBB itself, a malicious election official could mark MBBs as tallied by manipulating the database (EVEREST 20.6.1).

A unique feature of Tally among the EMS components is that its user interface is completely configurable through the Windows registry. Each registry entry specifies the DLL used to implement the behavior of a certain UI component. Modifying these behaviors in the registry can lead to subtle errors that are hard to detect (EVEREST 20.6.2). For example the import MBB and export MBB dialog boxes are exactly the same with the exception of one word. Unless the EMS systems are reinstalled and reconfigured between elections, which is highly unlikely, an election official could introduce such errors to Tally that would affect future elections. Such actions are nearly impossible to trace.

4.5 Auditing

A fundamental and critical requirement for a complex system such as election management is the ability to audit every element of it. Audit logs serve a vital purpose, as they can alert an auditor of suspicious or uncommon events that occurred, which could indicate the presence of malicious intent against the system. It is therefore critical that audit logs are complete and accurate.

In this section, we show that the audit logs for every component in the Hart InterCivic system are subject to manipulation and deletion. Taken in isolation, each of these attacks may seriously affect the auditability and ultimately, confidence in the election process. With just a

small number of well-placed insiders, however, or a combination of insiders and malicious outsiders, it is possible to compromise logs at the polling places and at election headquarters, resulting in a catastrophic loss of verification and accountability for the county. As every piece of the system is vulnerable to attacks against audit logs, there are insufficient protections within the Hart voting equipment and software to prevent a motivated adversary from compromising an entire election.

4.5.1 EMS Audit Logs

Many EMS applications (BOSS, Ballot Now, SERVO and Tally) all maintain audit logs of the functions they have performed. These logs are stored in databases, with every entry including a date and time when an action was performed, the name of the user performing the logged action, a numeric identifier for the action (the pairing of this identifier and its verbal description are located in another database table) and data pertaining to the log entry (e.g., an adjusted vote total).

The database storing the audit log may be accessed by an unprivileged attacker and the logs modified such that any evidence of tampering in the voting system is covered (EVEREST 20.1.4). This can be done by first extracting database passwords from application configuration files, as detailed in Issue 15 of the CA TTBR. We used a freeware software utility that allowed us to communicate to the database through an ODBC interface and issue SQL commands directly. We were able to perform arbitrary operations on the databases in this manner. For example, an operation in Tally allows for the manual changing of vote totals; we were able to remove the audit log entry for this operation, or modify it to reflect an innocuous operation instead by changing the numeric identifier for the action.

4.5.2 Compromising the VVPAT record

In Ohio, eSlate DRE machines are used in conjunction with VBO printers that produce a verified-voter paper audit trail (VVPAT), with the resulting generated paper record acting as the legal ballot. The eSlate controls the VBO through a 1/8-inch port that is accessible by removing the VBO from its housing. The eSlate housing has a large black release button above the VBO, allowing it to be removed. The accessible port is the interface through which a variety of operations to the VBO are performed, including sending messages to be printed, checking whether the printer is low on paper, setting the VBO's serial number, printing debug information, and checking for general printer error conditions. There is no authentication of commands that arrive over this interface. As a result, an adversary who can control the interface to the printer can print arbitrary data to it, as described in issue 34 of the CA TTBR. Notably, other interfaces may lead to the sending of privileged commands to the VBO. In particular, the serial number may be changed through the parallel port of the JBC and the eSlate's serial port in addition to using the 1/8" VBO port; we successfully changed the VBO's serial number using the JBC's parallel port by writing a short C program on a laptop and attaching it to the JBC. A modified serial number could call into validity the votes recorded to the VVPAT (EVEREST 20.5.5).

The VBO printer is easy to disable. The VBO connects into a power cable and a data cable. If either of these is severed, particularly if it is done skillfully, then the connected eSlate will show a communication error that is hard to diagnose. Since the VBO is not field-serviceable, a new one would need to be brought in and determining the core problem may be difficult. The eSlate can hence be knocked out of service for a significant amount of time, perhaps the duration of the election, potentially causing voter disenfranchisement. The eSlate takes approximately 15 seconds to report an alarm to the JBC, leaving ample time for an attacker to leave the polling place before malfeasance is suspected (EVEREST 20.5.2).

The VBO may potentially be handled by the voter, as a large black button on the eSlate's housing allows the unit to be removed, though it is not meant to be handled in a polling place. The back of the VBO has a pair of screws that may be turned by hand to access the interior of the unit. The paper may then be removed from the spools and either replaced or the reattached after removing the portion of the roll on the take-up spool (EVEREST 20.5.4). We found it was possible to perform this in as little as one minute, with the movements obscured by the privacy shield attached to the eSlate housing. However, the JBC's LED for the eSlate may flash when the data cable is detached from the VBO, although it is possible with care to perform the operation without causing the JBC to flash.

Even if the VBO is not itself compromised, there is little assurance that the generated VVPAT is trustworthy. When the VBO prints the accepted vote, a two-dimensional barcode is printed in the standard PDF-417 format, making it easy to generate. The rest of the ballot is generated in plain text, as alluded to in the CA TTBR. Nowhere is any authenticating information (such as an HMAC) embedded into the barcode or printed anywhere else on the ballot. As long as an adversary knows the serial number of the VBO, an entire roll can be forged and either replaced in the VBO (an operation that can take about a minute in a precinct) or when the tape from the VBO is removed (EVEREST 20.5.5). It is not clear whether the bar code is used to tabulate results from the

paper roll or whether it is examined at all.

4.5.3 Open Interfaces on Voting Equipment

Both the JBC and eScan have open interfaces that allow for the erasure of votes and audit log records. As detailed in Issue 3 of the CA TTBR, the eScan is managed through an accessible Ethernet port that listens for connections on TCP port 4600. This port is normally used for sending and receiving commands from SERVO, such as file transmission and reading images of the eScan's memory. No cryptographic tokens are required for these operations to occur.

We discovered that with a handheld device such as a Palm computer, an attacker with an Ethernet cable can mimic the actions of SERVO to the eScan during a live election, and cause the vote records and audit logs to be erased from both the eScan's internal memory and the MBB inserted into it (EVEREST 20.3.7). Any voting that had occurred on the eScan to that point would be erased, necessitating a manual recount.

The JBC is similarly vulnerable to attack (EVEREST 20.4.2). SERVO connects to the JBC over a parallel port interface. If a Palm handheld with a parallel port interface is connected to the JBC, it may be used to clear the vote records and audit logs from the JBC's internal memory and the MBB attached to it. Since the JBC controls the eSlates as well, it is also possible to clear their vote records and audit logs from the JBC's parallel interface. We wrote a program and that allowed us to reset the JBC and eSlate from a laptop, and found that all evidence of voting on that machine had been cleared.

5 Premier Analysis

This section focuses on the systemic vulnerabilities found in Premier Elections Systems, and uses examples from new or previously unevaluated systems to ground these observations. In particular, we show that vulnerabilities in the Election Media Processor (EMP) server, Voter Card Encoder (VCE), Digital Guardian and ExpressPoll units not only exhibit many of the same kinds of vulnerabilities discovered in the past, but in some cases contain line-for-line copies of the same vulnerable code.

From our analysis, we demonstrate that these vulnerabilities are the result of the following larger failures of the system's design or implementation. We discuss each issue, in detail and order, throughout this section:

- *Failure to effectively protect vote integrity and privacy* - Numerous vulnerabilities allow an attacker to modify or replace ballot definitions, to change, miscount, or discard completed votes, or to corrupt the tally processes. Further issues expose voter choices and can lead to voter coercion and vote selling.

- *Failure to protect election from malicious insiders* - The Premier system does not provide adequate protections to ensure election officials, poll workers, or vendor representatives do not manipulate the system or its data. These attacks are often invisible after the fact, and therefore misuse is difficult or impossible to uncover later.

- *Failure to validate and protect software* - The Premier system makes only limited and often ineffective attempts to validate the software running within system. Thus, an attacker may exploit software and replace it with their own with little fear of detection. Further, the recommended means of installing and upgrading software is frequently highly dangerous.

- *Failure to provide trustworthy auditing* - The auditing capabilities of the Premier system are limited. Those features that are provided are vulnerable to a broad range of attacks that can corrupt or erase logs of election activities. This severely limits the ability of election officials to detect and diagnose attacks. Moreover, because the auditing features are generally unreliable, recovery from an attack may in practice be enormously difficult or impossible.

- *Failure to follow standard software and security engineering practices* - A root cause of the security and reliability issues present in the system is the visible lack of sound software and security engineering practices. Examples of poor or unsafe coding practices, unclear or undefined security goals, technology misuse, and poor maintenance are pervasive. This general lack of quality leads to a buggy, unstable, and exploitable system.

  We found the Premier software to be unstable. Frequent crashes, system lock-ups, and unexplained errors were commonplace in our experiments. Stability problems were acute in the GEMS server, where failures occurred during normal use and under limited loads.

We begin by overviewing the Premier voting system architecture as used in Ohio and then visit each failure in turn.

5.1 Premier Architecture

We briefly overview the Premier Voting System by walking through a sample election procedure (as typical in Ohio); a more detailed description can be found in the EVEREST report [23]. Refer to Figure 3 for component

orientation and interaction. Note that our study is unique in its access to the EMP or ExpressPoll, as well as Verdasys Digital Guardian, a third party tool used to secure the GEMS server in Ohio counties.

Using the Global Election Management System server, or *GEMS* server (1), an administrator begins an election by defining a ballot. This includes determining the races, candidates and issues that will appear. When the ballot is approved, the GEMS server communicates over a local area network with either the *Central Office AV-TSX* (2) or the *Election Media Processor* (3), which encode 128 MB PCMCIA memory cards (7) used at the polling place AV-TSX. The Election Media Processor, or EMP, is a PC running either Windows 2000 or XP connected an external drive bay containing multiple memory card readers and is incorporated for efficiency reasons. GEMS also communicates with a *Central Office AV-OS Precinct Count* (4) in order to encode 128 KB EPSON 40-pin memory cards used by the polling place AV-OS. Memory cards are then sent to the polling station either independently or pre-inserted into voting machines, depending on policy. Also configured by GEMS at the county election headquarters is the *AV-OS Central Count* (5) (used for absentee ballots) connected via a *Digi PortServer II* (6), which multiplexes serial connections into Ethernet.

For counties using Premier touchscreen voting systems, a precinct administrator opens an election by inserting a *Supervisor Card* (a smart card) into the *AV-TSX* (8). After voters receive a *Voter Card* (9) from a poll worker with either the *Voter Card Encoder*, VCE (10) for short, or *ExpressPoll* (11) (an electronic replacement for the traditional voter log book, which runs Windows CE), they approach an AV-TSX and insert it into the machine. After casting their vote, the voter returns their used Voter Card and leaves the polling station. When the poll closes, the precinct administrator then reinserts the Supervisor Card and closes the election. Elections using optical scan units instead begin by having a precinct administrator place the device into election mode. Voters in these precincts fill out paper ballots and then feed them to the *AV-OS PC* (12), which scans their results. In both systems, memory cards are shipped back to the county elections headquarters at the close of elections for centralized tabulation.

Upon arriving at the county's election headquarters, memory cards are then inserted into the appropriate devices, which communicate the results of the election to the GEMS server over the local area network. The GEMS server then prints an official election results summary, which is used as the official outcome of the election.

5.2 Vote Integrity and Privacy

Many previous studies have scrutinized the integrity of the Premier voting system, proposing and confirming various attacks that influence the number of tallied votes or expose voter choices. For example, Hursti [14] originally described techniques to "pre-stuff" the AV-OS PC's counters while feigning the per-election "zero report." Such attacks frequently exploit combinations of implementation flaws existing throughout the system components, e.g., buffer overflows and integer overruns. However, in this section, we discuss two vulnerabilities resulting from unsafe functionality designed into system components. We conclude with a discussion of recycled vulnerable code in the EMP server.

5.2.1 Casting an Unlimited Number of Ballots

Premier Election Systems use smart cards to ensure that each voter is only able to cast a single ballot per election. After casting their ballot on an AV-TSX, the card reader marks the card as "Cast". If this card is reinserted into an AV-TSX before it is re-enabled by a poll worker (using either the ExpressPoll or Voter Card Encoder), the voting machine ejects the card and alerts the user that it has already been used. Implemented correctly, this mechanism should prevent a single user from casting more than their allotted single ballot.

Using multiple vulnerabilities discovered during the EVEREST evaluation, it is possible to enable a voter to bypass these mechanisms and cast an unlimited number of votes. Moreover, the evidence that such an attack has been launched can also be erased. Worse still, this attack requires no special tools or private knowledge of the system.

We assume that our attacker approaches the voting booth during an election under normal circumstances. The attacker brings with them a stack of smart cards containing the default Smart Card Key (published on the Internet). After approaching the AV-TSX, the attacker begins by covering his/her tracks. Because the AV-TSX notes in its audit logs when cards have been encoded, the attacker accesses the Central Administrator mode by exploiting EVEREST Issue 14.8.7. Here, the attacker can delete the contents of both the memory card and the AV-TSX, thereby erasing most evidence of the attack. To hide the card creation operations, the attacker then simply changes the time and date of the AV-TSX to a period before the election. This portion of the attack can be accomplished in just over one minute. Moreover, deleting the contents of the memory card and changing the time/-date are not logged. Should the attacker also worry about the log information encoded on the VVPAT, weaknesses in the enclosure allow the paper record to be rendered unreadable (EVEREST, Issue 14.8.3).

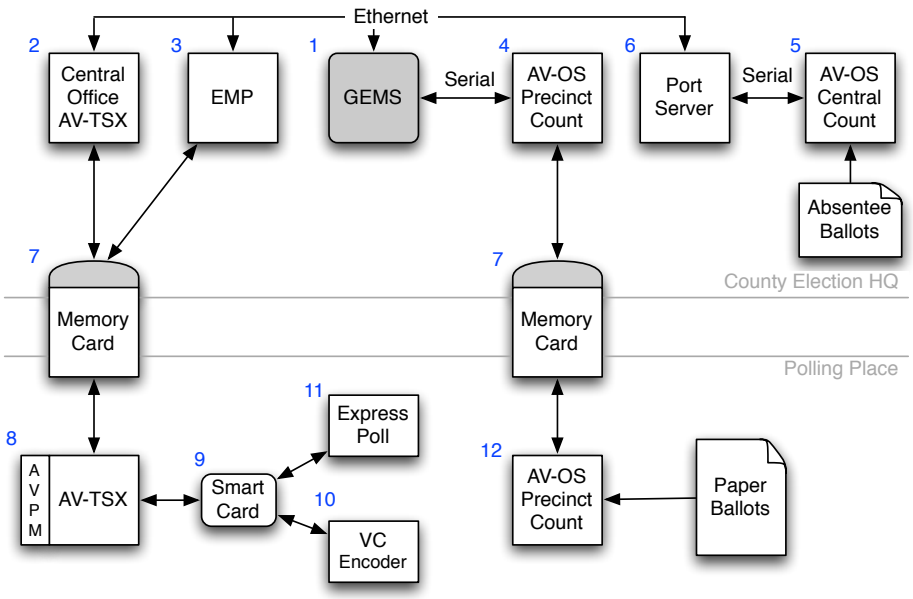

Figure 3: The major components and their relation to each other for counties using Premier Electronic Voting Systems. Unless otherwise stated, arrows depict physical transport of cards or ballots.

To encode voter cards, the attacker gains access to the Supervisor Menu by exploiting the vulnerability described in EVEREST Issue 14.8.8. Access to this menu can be achieved consistently in under one minute. The attacker then encodes the stack of smart cards smuggled into the voting precinct as valid Voter Cards, each of which takes a few seconds. There is no limit to the number of cards that can be programmed.

The attacker can then walk away from the machine and give the cards to colluding adversaries in the parking lot. These adversaries can use all of the cards to cast extra votes.

5.2.2 Exposing Voter Choices

Premier Election Systems recently introduced the ExpressPoll to replace traditional paper voter log books at the polling place. While the ExpressPoll contains various vulnerabilities allowing files and software to be manipulated (see Section 14.6 of the EVEREST report), a non-trivial privacy concern results from the technique used to audit the device. Note that the ExpressPoll does not directly participate in vote tallying; however, it encodes Voter Cards, and therefore the audit log should indicate if a voter's status was reset to allow multiple votes.

When voters enter the poll place, they are authenticated via information present in the ExpressPoll. Once authenticated, the voter is given a Voter Access card, and the voter information database is updated to indicate the voter has already entered the polling place. Along with this update, the ExpressPoll appends the activity audit log indicating the `voterId`. The `voterId` field recorded in the audit log matches a similar field in the voter information database. As the audit log is appended, the order voters enter the polling place is captured with a sequence number, and while a timestamp is not recorded for these entries, other entries, e.g., power-on, include a timestamp (EVEREST, Issue 14.6.7). Hence, an attacker can derive approximate times for voter entries. The voter order can also be correlated with VVPAT records for the AV-TSX to determine with some probability each voter's choice. Such information enables vote coercion and places significant tension on the efficacy of the election process.

5.2.3 Failure to Address Previous Vulnerabilities

In the early phases of our study, the EMP was the focus of much of our research planning. As a previously unevaluated device, the EMP represented an opportunity to determine whether past vulnerabilities were being repeated or fixed as Premier systems evolved. Much to our dismay, large portions of the EMP source code were copied line-for-line from the AV-TSX. According, vulnerabilities found in the AV-TSX, such as Issue 14.1.1, exactly mirror previously reported weaknesses. This particular discovery, in addition to the consistent misapplication of security mechanisms and practices in the other newly evaluated components, led us to conclude that the security of Premier systems is not only not improving,

but in fact repeating many of the same mistakes brought to light in previous studies.

5.3 Malicious Insiders

Due to the results of previous studies of the Diebold/Premier elections equipment, the state of Ohio required that Premier include additional third party security software to harden the GEMS server. Specifically, the GEMS server setup in Ohio includes: *Verdasys Digital Guardian*, *Sygate Security Agent* network firewall, and *McAfee VirusScan*. The latter two security tools provide standard system protection and warrant little discussion. However, Digital Guardian is presented as a remedy to a number of significant GEMS vulnerabilities, such as the ability for an attacker to perform arbitrary modification of an election database simply by having access to the GEMS server file system (CA TTBR, Issue 5.3.2; EVEREST, Issue 13.1.2). Note that because Digital Guardian is considered COTS software, Premier was not required to provide any source code, nor were we provided any technical documentation describing how the system works. However, we were provided the current policy specifications and some notes from a Premier technician, which greatly aided our understanding of how Digital Guardian protects a system.

5.3.1 Protecting GEMS with Digital Guardian

Digital Guardian was designed to protect a system running Windows 2000 or XP. It allows an administrator external from the local system to specify policies that control how all local users are allowed to execute programs and access files. In Ohio's setup, a state employee possesses a special laptop called the Digital Guardian console. Each GEMS server contains the Digital Guardian Agent that enforces the policy specified by the console. The only way the Digital Guardian Agent can be disabled is if a state employee directly connects the Digital Guardian console to the GEMS server and specifies that the agent should be disabled.

The Digital Guardian Agent running on all GEMS servers enforces two high level policies (keep in mind that in Ohio, all county GEMS servers are administered by Premier employees). First, the election databases should only be accessed by the GEMS program. Second, the vendor employee should not have access to any GEMS data. The remainder of the policy installed in each Digital Guardian Agent exists purely to retain the system integrity and keep an attacker from circumventing Digital Guardian.

In order to provide separation between users, three Windows users have been created: *Administrator*, *GEMSAdmin*, and *GEMSUser*. The *Administrator* account performs basic administration and maintenance of the GEMS server, but operations that involve GEMS data are forbidden. The *GEMSAdmin* account is not a system administrator, rather, it is the only user allowed to perform file manipulation operations, e.g., copy, move, delete, on the election database files. Finally, the *GEMSUser* account may only modify election database files using the GEMS program, and it should not be able to delete, copy, or paste the files. Additionally, *GEMSUser* is allowed to burn backups of the election database, as this is a necessity on election day.

5.3.2 Circumventing Digital Guardian

We performed penetration testing to investigate how a malicious insider can circumvent Digital Guardian to exploit exiting GEMS vulnerabilities. Our analysis of Digital Guardian focused on its ability to enforce the high level protection policies. Due to time constraints, we only studied the GEMS server and not the Digital Guardian console laptop or the network communication. Vulnerabilities fit into three categories: configuration flaws, means of disabling Digital Guardian, and flaws in the Digital Guardian software itself.

The Digital Guardian configuration contains a number of addressable flaws. One of the more significant enablers for circumventing Digital Guardian is the configuration of Microsoft Windows. Specifically, the *GEMSUser* user account is in the Windows *Administrators* group (EVEREST, Issue 14.7.2). Many of the deeper vulnerabilities we discovered rely on administrative access, which is easy to assume given this configuration. The Digital Guardian policy itself also contained simple misconfigurations. For example, the Nero CD burning application can rename GEMS database files (EVEREST, Issue 14.7.8) thereby allowing an attacker to modify its contents before replacing the original. In both cases, configuration fixes could mitigate the vulnerabilities.

Deeper configuration errors stemmed from limitations of the general approach for policy specification. That is, the policy "blacklists" specific potentially dangerous applications (EVEREST, Issue 14.7.6), e.g., the registry editor. Blacklisting has a fundamental limitation: it cannot practically identify all current and future applications. For example, we were able to use a command line task scheduler to launch a shell as the "SYSTEM" user, which bypasses all Digital Guardian protections (EVEREST, Issue 14.7.5). Furthermore, one blacklist identification technique relies upon the cryptographic hash (MD5) of the application, thereby allowing an attacker to circumvent protections by simply modifying one bit in the binary.

The limitations of the blacklist policy more fully manifest in techniques to disable Digital Guardian all together. While BIOS passwords help prevent an attacker from booting from external media to disable Digital Guardian (EVEREST, Issue 14.7.3), the policy failed to blacklist access to `C:\ntldr`, which defines the location of the boot loader configuration (`C:\boot.ini`), a file specifically blacklisted by the policy. Hence, an attacker can modify `C:\ntldr` to use a different file, e.g., `C:\b00t.ini`, that is under the attacker's control (EVEREST, Issue 14.7.1). By modifying the boot loader configuration, Grub4DOS can be used to boot from a CD-ROM and disable Digital Guardian. Additionally, the policy did not blacklist "Device Manager," which we found can be used (only once) to disable the device drivers implementing the Digital Guardian enforcement mechanism (EVEREST, Issue 14.7.4).

A final category of discovered vulnerabilities were unrelated to the configuration. Rather, we believe them to be flaws in the Digital Guardian implementation. While we were not provided technical documentation from Verdasys, experience with similar tools brought us to the conclusion that when the policy identifies an application by a cryptographic hash (e.g., for blacklisting to deny execution) the enforcement mechanism should calculate the application's hash on demand (e.g., if an application is blacklisted from executing, every time an application executes, the hash should be calculated and compared against those in the blacklist). However, this was not the case, as we were able to copy black listed applications to a new location and execute them (EVEREST, Issue 14.7.7). While we were unable verify the exact enforcement technique, our best speculation indicated that Digital Guardian caches a table mapping file paths to hash values, and the file path is used to identify applications. This design leaves the system susceptible to a TOCTTOU attack.

Our study of the Digital Guardian protection of the GEMS server showed it to be insufficient. Securing an operating system is a nontrivial task. However, providing operating system level protection against the intended user of an insecure application is even more daunting. Even with a correct configuration and absence of implementation flaws in Digital Guardian, attacks such as the GUI "un-graying" (CA TTBR, Issue 5.3.3; EVEREST, Issue 13.1.3) remain possible. Hardening the operating system goes a long way towards securing GEMS against malicious insiders; however, it is no substitute for fixing the vulnerabilities within the application itself.

5.4 Software Update Authentication Vulnerabilities

The lack of software update authentication mechanisms in Diebold/Premier systems has long been known. Reports including Hursti's analysis of the AV-TSX [15, 16] have previously demonstrated the ability of an adversary to replace the operating system, bootloader and application software simply by including files with the correct name (EBOOT.NB0 and NK.BIN) or suffix (.ins) on a memory card. Despite being widely criticized as insecure, such vulnerabilities appeared repeatedly in our study of new and previously unevaluated equipment. In this section, we discuss the lack of robust software update authentication mechanisms in the ExpressPoll, VCE and Digital Guardian.

5.4.1 ExpressPoll

In order to allow updates to the bootloader and operating system, the ExpressPoll scans all inserted memory cards (both PCMCIA and CF) on boot. If the bootloader finds a file purporting to be a new bootloader (EBOOT.BIN) or Windows CE (NK.BIN), it erases the previous version of the software and loads the new version from the above file(s). Like the vulnerabilities previously discovered by Hursti, at no time is the source of these files authenticated; rather, a file on the memory card with either of these names will automatically be loaded and executed. Accordingly, anyone that can power cycle an ExpressPoll and insert a new memory card (i.e., any poll worker) can exploit this vulnerability. This vulnerability exactly mirrors Hursti's report on the AV-TSX, except that it has been re-implemented in a new system.

We note that there is a chance that placing the Windows CE file (NK.BIN) will not replace the current operating system, but rather only boot from it. Due to the potentially destructive nature of the testing and the fact that we were not given builds or most of the source code for the ExpressPoll, we verified that the files are accepted, but did not allow the process to be completed. Regardless, either set of functionality, booting as a runtime image or direct flashing, offer the same potential. Once booted, the runtime image can flash itself to permanent memory.

These vulnerabilities are one of a number of ways by which an adversary can gain access to the database of eligible voters. Accordingly, voters could be arbitrarily added or removed from such a list, thereby potentially compromising the integrity of the election and or disenfranchising voters.

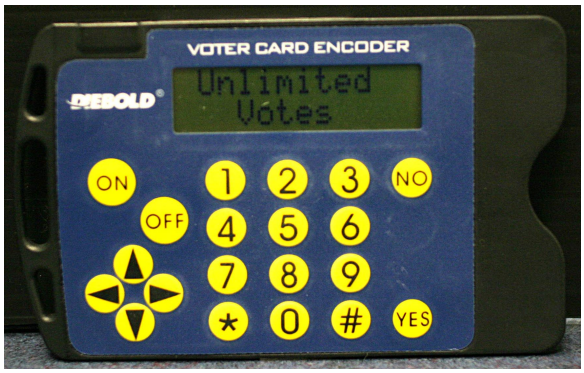

Figure 4: A VCE running arbitrary software.

5.4.2 VCE

In order to allow for software updates, the VCE can be reprogrammed using a 9-pin serial cable attached to a PC. To load new software onto the VCE, a user simply turns the device off. When the user presses the off button again, the Voter Card Encoder prompts the user to press the "Yes" button if they would like new software to be loaded.

The problem with this update mechanism is that it lacks any authentication of the new software loaded onto the VCE. As a demonstration of this issue, we created and loaded new software. Where the software provided by the manufacturer requires a user to activate the VCE with a Supervisor Card, we allowed any card (even unrecognizable formats) to enable the device. An adversary could therefore steal a VCE[1], load their own software and then create valid Voter Cards. Figure 4 shows an example of our modified software created in Issue 14.5.3 of the EVEREST report. Alternatively, we could have used this vulnerability to encode any type of card we wished. For instance, an attacker could easily create Central Administrator, Security or Supervisor Cards by further modifying the software running on the VCE.

By providing no real barrier to replacing software, a compromised VCE represents a significant threat to the integrity of an election.

5.4.3 Digital Guardian

In the state of Ohio, Premier provides Verdasys Digital Guardian on the GEMS server to harden the server and protect against many known vulnerabilities. As mentioned earlier, the Digital Guardian protection policy was designed to enforce two high level goals: only allow the GEMS application to access the election database, and the vendor technicians should never gain access to the election database. In doing so, the GEMS server is configured to ensure the integrity of the GEMS application. Specifically, Digital Guardian policy includes the MD5 hash corresponding to the correct release of GEMS. The system uses this value to ensure that only an application matching that hash can access election databases.

Section 5.3.2 described how flaws in Digital Guardian allow an adversary to execute blacklisted applications by copying the binary to a new file system location. However, executing blacklisted applications only indirectly gain an adversary access to election data. Due to the identification flaw, instead of executing blacklisted applications by coping them to a new location, the adversary can replace any *whitelisted* application to gain its privileges. This vulnerability poses a significantly different threat to the GEMS system, as it allows the adversary to overwrite the GEMS application without immediate detection (EVEREST, Issue 14.7.11). Depending on attacker motivations, this replacement may be temporary (e.g., to gain unfettered access to election data) or long term (e.g., to run an election with a malicious version of GEMS). Due to the disconnected nature of the Premier architecture, the latter replacement may go undetected for long periods of time and is only exacerbated by the lack of auditing present in the Digital Guardian configuration, as discussed in the next section.

5.5 Trustworthy Auditing

The key to any successful election is the ability to determine whether or not the outcome correctly reflects voter intent. Reliable auditing mechanisms provide a means of independently evaluating the correctness of such results after an election. Previous studies of Diebold/Premier voting machines have demonstrated a lack of reliable audit trails in polling station equipment, especially the AV-OS PC to the AV-TSX [3]. In this section, we show that the lack of reliable audit trails continues to be a problem in new and previously unevaluated equipment. Failure to properly record the events occurring at any one of these devices may allow an adversary to negatively impact an election without threat of detection.

5.5.1 ExpressPoll

The ExpressPoll logs all user activities, including login attempts and modification of voter information, using an unprotected DB3 database file. System exceptions are also logged, however, these events are recorded in a separate .xml file. As we note in Issue 14.6.6 of the EVEREST report, neither files or their contents are adequately protected against an adversary. In the absence of cryptographic controls, these logs can be modified by anyone in possession of the ExpressPoll device. Alternatively, entire logs can be deleted or replaced as the operating system (Windows CE) allows the user full administrative control. Upon deletion of either file type, the ExpressPoll

simply creates a new audit file without indicating any error to the user.

An attacker with access to the log files could remove all traces of malicious activity. For example, if a malicious poll worker changes a voter's status back to "unvoted," the corresponding log entry can be removed or changed to an otherwise benign event. Accordingly, should a post-election audit occur, the log information from the ExpressPoll can not be used as a reliable account of events.

5.5.2 Digital Guardian

As mentioned in Section 5.3, Digital Guardian is installed on the GEMS server in Ohio as an attempt to provide additional protection and auditing. When a user performs an action that is forbidden by the Digital Guardian policy, that action is denied and a dialog box informs the user that the action has been blocked and recorded. However, installation guides provided for setting up Digital Guardian on GEMS servers explicitly indicates that the "Enable Activity Detail Logging" option should be unchecked. This guide corroborates discussions with state employees indicating Digital Guardian logging is disabled due to storage concerns. In particular, the possibility of generating and having to process voluminous security logs discouraged any event logging in these systems. As such, Digital Guardian does not record attempts to circumvent it, despite the displayed message indicating an infraction has been logged. This weakness is noted in Issue 14.7.10 in the EVEREST report.

Accordingly, the configuration of Digital Guardian used in Ohio provides no useful forensic evidence for use in a post-election audit. Election administrators would simply have no indication that any malicious activity was attempted in such systems.

5.5.3 EMP

The EMP server, which is responsible for the parallel reading and writing of memory cards used in the AV-TSX, keeps logs of many of the operations it performs. For instance, when a blank memory card is inserted and a new ballot definition downloaded, the EMP server creates a log entry. Logging also occurs when cast ballots are uploaded to the GEMS server or when an error (e.g., connection timeout) occurs. These logs provide evidence with which an auditor can reconstruct the events on an election.

Like the ExpressPoll, the integrity of the EMP logs is not protected. During the course of our investigation, we were also able to alter entries from outside of the application, and then properly view them in the EMP's log screen. By escalating our privileges in the operating system, we were able to simply erase such files without raising any alarms. Instead, like the ExpressPoll, the EMP simply created new log files when old ones were deleted. This vulnerability was reported in Issue 14.1.5 of the EVEREST report.

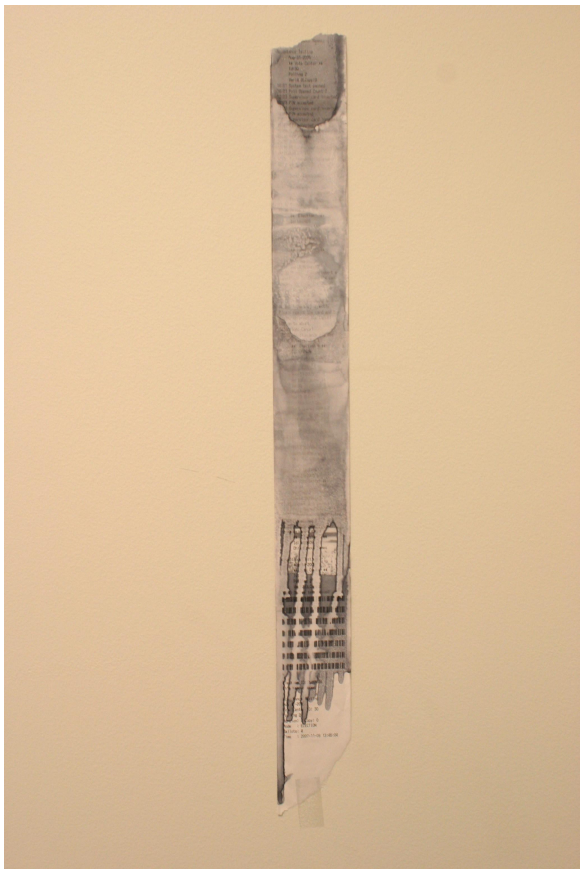

Figure 5: A printed system log destroyed by injecting a household chemical into the AV-TSX. No tamper-evident seals were broken or disturbed in the attack.

The EMP is, in most configurations, the gateway between back-end processing and all touchscreen voting machines throughout the county. Accordingly, the events that take place on this platform are extremely valuable to recreating the events of an election. For instance, if a virus were to spread from a precinct to the central headquarters, as was suggested in a number of previous works [3, 10], logs at the EMP would be a valuable tool in identifying the source of such an attack. However, such mechanisms are of limited value to any post-election audit as their integrity simply can not be trusted.

5.5.4 AV-TSX VVPAT

The paper audit trail generated by the AV-TSX machines operated in Ohio is cited by many as a failsafe means of recording a voter's intent. Before a ballot is cast, each

voter is afforded the opportunity to evaluate a printout of their selections and, should the electronic count be disputed, election administrators can rely on these receipts as the official legal record of the ballot. Unfortunately, the VVPAT system used by these machines is poorly constructed and subject to a number of attacks that negate their perceived value.

Chief among these problems is the construction of the VVPAT system itself. Protected by a thin and flexible plastic enclosure, the physical security of the printer is a significant risk in the Premier system. For instance, as discussed in Issue 14.8.1 of the EVEREST report, the wires connecting the printer to the AV-TSX can easily be exposed by pushing the edge of plastic covering. An adversary wishing to disable the printer on a AV-TSX could simply cut these wires without breaking any tamper-evident seals on the device. Alternatively, the plastic housing itself can simply be removed from the AV-TSX with minimal physical effort. Issue 14.8.2 of the EVEREST report notes that this enclosure is attached to the AV-TSX by a 1/8 inch plastic latch. By applying the appropriate pressure, an adversary can gain access to all previously cast votes without raising significant attention. Should the results of an election be disputed, the absence of a paper trail from these attacks would prevent a complete recount from occurring.

An attack less likely to catch the attention of poll workers until the close of an election is possible because of the inadequate sealing of the printer enclosure. As discussed in Issue 14.8.3 of the EVEREST report, an attacker can exploit this weakness by using a syringe to inject a common household substance known to degrade/destroy information written to thermal printer paper. Such a compound could be inserted in multiple ways such that all previous paper ballots stored in a machine would become unreadable. Alternatively, all of the unused paper in the machine could be attacked, preventing all future votes from being tallied. An example of the first attack is shown in Figure 5. Note that the results of the audit log are unreadable. A similar vulnerability was discussed in the California Red Team report (Issue 4.f) [1]; however, the details of this vulnerability were not listed in the public report.

A number of factors of the AV-TSX VVPAT system combine to make such attack possible. The use of an inexpensive and pliable plastic enables the first two attacks. Thermal printers, of which the use for creating long-lived records is recommended against due to fading problems, enable the latter. Because of these weaknesses in implementation, the VVPAT results generated by an AV-TSX can not be relied upon as the only auditing mechanism for Premier systems. Unlike more traditional systems in which ballots are kept in a central, guarded ballot box, VVPATs simply do not provide the same protection of a voter's intent.

5.6 Security Engineering Practices

The value of strong security techniques can be instantly voided if such mechanisms are improperly used. Such is the case in all of the Premier systems investigated in the EVEREST report. We examine two particular areas, the failure to correctly apply security mechanisms and miscalculations of trust to demonstrate systemic security problems in Premier systems.

5.6.1 Ineffective Application of Security Techniques

Key management problems are well known in Premier systems. As was demonstrated in the CA TTBR (Issue 5.2.5), keys are insufficiently protected in units such as the AV-TSX. However, our analysis of the EMP uncovered additional problems in the key management of such systems. As conjectured in the CA TTBR and confirmed in Issue 14.1.2 of the EVEREST report, the Data Key used to protect the results of an election is the same in every machine in a county. Key management in the EMP is substantially more dangerous. As discussed in Issue 14.1.7, the System Key used to encrypt the Data Key is derived from the system's serial number. Like the System Key in the AV-TSX, the System Key for the EMP server is created in a predictable manner. The machine's serial number is fed as input to the MD5 hash algorithm, the deterministic result of which becomes the System Key. On each AV-TSX, this serial number (and therefore the resulting System Key) is unique. However, the serial number used is a fixed value on all machines: 0. Accordingly, every EMP server created uses the same System Key. Such key management strategies fail to provide containment against compromise and therefore allow a successful attack on a single machine to potentially compromise elections on a large scale.

Settings provided by the operating systems of a number of Premier devices also fail to prevent an adversary from causing damage to an election. The ExpressPoll, for instance, fails to provide any protection of the database containing voter names (Issue 14.6.3) or resource files (Issue 14.6.4). Accordingly, anyone possessing this device can easily modify or replace voter lists or given themselves extended capabilities including the use of Windows Explorer.

5.6.2 Systemic Trust Assumptions

Assuming that interactions between two entities in a system can inherently be trusted often leads to the exploitation of vulnerabilities. Such problems are often embodied as a lack of input checking on memory cards or format filtering on a user interface; however, misplaced trust

can also lead to vulnerabilities that exploit falsely placed trust in users of systems components. As shown in the EVEREST report, such problems are rampant in Premier systems. We use multiple examples from the EMP server to illustrate such issues.

One of the better examples of how trust is systemically misappropriated is discussed in Issue 14.1.9 of the EVEREST report. When memory cards are read by the EMP server after the election, the EMP decrypts the results on each card using the Data Key. Because the same Data Key is known by the EMP and all AV-TSX devices and the serial number of the AV-TSX associated with each vote is included in the header of the results file (see CA TTBR Issue 5.2.5), the EMP server can therefore perform all of the operations associated with any AV-TSX and use the correct cryptographic keys to "validate" the results. There would be no means of distinguishing where a vote was written. There is therefore no reason for the GEMS server to believe the accuracy of the results reported from the EMP server. By providing the EMP with such functionality, the attack surface of the system is significantly expanded.

A number of vulnerabilities make malicious control of an EMP possible. Trust that the contents of memory cards are benign specifically endangers the system. Whether an adversary controls the GEMS server (Issue 14.1.10) or can compromise a single AV-TSX in any precinct (Issue 14.1.11), the EMP is susceptible to multiple format string vulnerabilities. Because both of the above vulnerabilities are exploited immediately on the insertion of a memory card, an attacker need not compromise an AV-TSX or have knowledge of the cryptographic keys used in the system in order to successfully launch an exploit; rather, simply ensuring that a card with malformed election header information reaches the EMP server is sufficient.

One final example comes from the user interface of the EMP (Issue 14.1.4). As part of setup, the EMP software requires that a user enter the IP address or host name of the GEMS server. This allows the EMP server to connect to GEMS when performing its uploading and downloading duties. In theory, if the EMP user accidentally enters a malformed IP address or hostname, they should be able to use the Communications Setup menu to correct their error. However, the user may never be given such an opportunity. On startup, the EMP server immediately becomes unresponsive and fails to correctly render the user interface. Because none of the menus have yet been rendered on the screen, the EMP user is never given the opportunity to change this setting back to a correct value.

The value for the GEMS server's address is stored as an entry in the registry. Because the EMP user is given minimal rights (which is a good security practice), they can not edit the registry to fix this problem. Moreover, unless the administrator understands that the host string is stored in the registry, it is unlikely that they will be able to fix the problem. Because the uninstall program included with the EMP software fails to remove these entries from the registry, this problem persists across reinstallations. Note that no error checking is present on this interface; rather, the EMP always trusts that the user correctly entered the data.

6 Conclusion

Project EVEREST was a unique opportunity to evaluate the security and integrity of elections run using equipment and software created by Hart InterCivic and Premier Elections Solutions. Whereas researchers in previous studies were often limited in their ability to access to both hardware and source code, members of the EVEREST team were able to use our unfettered access to identify and in many cases more fully characterize vulnerabilities throughout the systems. The results of the study were significant - in less than nine weeks of study, our team discovered 27 new issues in the Hart system and doubled the number of publicly known weaknesses in Premier systems; given the increasing discovery rate at the close of the study, we expect many more issues remain. In particular, with more time, a deeper understanding of much of the full functionality of the Hart system, much of which is currently unknown, could serve to present a greatly increased attack surface.

Our findings in the Hart study showed that while some action could be taken to patch software and remove obvious points of vulnerability, such as what appears to be test harness code in production systems (e.g., the Autovote function), many other issues remain that will only be solved with a thorough re-architecting and redesign of the Hart InterCivic system with security as a top priority for every design point. A system whose technical security failings leave the system with only procedural protections in place is not adequate for the diverse and substantial needs of states.

In the study of Premier's systems, we demonstrated that such problems are systemic - previously known vulnerabilities not only still exist in Ohio's current voting systems, but the newly evaluated Premier components contain many of the same problems. In a number of cases, vulnerable code has been copied line-for-line from old (AV-TSX) to new (EMP) systems. This discovery demonstrates that not only are old problems not being addressed, but they are in fact being repeated in newer systems.

Our analysis will certainly not be that last evaluation of electronic voting equipment. If and when the next study occurs, we hope that other researchers will find our methodology helpful. In particular, by forcing our-

selves to begin with the confirmation of known vulnerabilities, we were able to quickly learn about the innerworkings of the Hart and Premier systems. This process not only added value to the community by providing independent validation of previously known problems, but also served to help us quickly identify new vulnerabilities in both previously evaluated and new components of the system. We recommend that future studies follow a similar model not only to create further confidence in the results of previous reports, but also to allow researchers in such studies to understand these systems as quickly as possible so as to allow them to identify additional serious weaknesses.

Notes

[1]A large number of VCEs have been unaccounted for after past elections. The 2006 evaluation of elections in Cuyahoga County, Ohio noted some 215 VCEs missing after the election occurred [8].

References

[1] ABBOT, R., DAVIS, M., EDMONDS, J., FLORER, L., PROEBSTEL, E., PORTER, B., SHENOI, S., AND STAUFFER, J. UC Red Team Report: Diebold Elections Systems, Inc. University of California, Berkeley under contract to the California Secretary of State, July 2007.

[2] ABBOT, R., DAVIS, M., EDMONDS, J., FLORER, L., PROEBSTEL, E., PORTER, B., SHENOI, S., AND STAUFFER, J. UC Red Team Report: Hart InterCivic. University of California, Berkeley under contract to the California Secretary of State, July 2007.

[3] CALANDRINO, J., FELDMAN, A., HALDERMAN, A., WAGNER, D., YU, H., AND ZELLER, W. Source code review of the Diebold voting system. University of California, Berkeley under contract to the California Secretary of State, July 20, 2007.

[4] CALIFORNIA SECRETARY OF STATE. Top-To-Bottom Review, July 2007.

[5] COMPUWARE CORPORATION. Hart InterCivic Direct Recording Electronic (DRE) & Voter Verifiable Paper Audit Trail (VVPAT) Technical Security Assessment Report. For Ohio Secretary of State, December 13, 2005.

[6] CORPORATION, M. Microsoft Security Bulletin (MS00-0500). http://www.microsoft.com/technet/security/Bulletin/MS00-050.mspx, July 2000.

[7] CORPORATION, M. Microsoft Security Bulletin (MS02-004). http://www.microsoft.com/technet/security/Bulletin/MS02-004.mspx, February 2004.

[8] CUYAHOGA ELECTION REVIEW PANEL. Final Report. http://www.cuyahogavoting.org/CERP_Final_Report_20060720.pdf, 2006.

[9] DORRENDORF, L., GUTTERMAN, Z., AND PINKAS, B. Cryptanalysis of the Random Number Generator of the Windows 2000 Operating System. In *Proceedings of the ACM Conference on Computer and Communications Security* (Alexandria, VA, Nov. 2007).

[10] FELDMAN, A., HALDERMAN, J. A., AND FELTEN, E. Security Analysis of the Diebold AccuVote-TS Voting Machine. In *USENIX/ACCURATE Electronic Voting Technology Workshop (EVT)* (2007).

[11] Fortify Source Code Analysis (SCA). http://www.fortifysoftware.com/products/sca/.

[12] GAINEY, D., GERKE, M., AND YASINSAC, A. Software Review and Security Analysis of the Diebold Voting Machine Software: Supplemental Report. Security and Assurance in Information Technology (SAIT) Laboratory, Florida State University, For the Florida Department of State, August 10, 2007.

[13] GARDNER, R., YASINSAC, A., BISHOP, M., KOHNO, T., HARTLEY, Z., KERSKI, J., GAINEY, D., WALEGA, R., HOLLANDER, E., AND GERKE, M. Software Review and Security Analysis of the Diebold Voting Machine Software. Security and Assurance in Information Technology (SAIT) Laboratory, Florida State University, For the Florida Department of State, July 27, 2007.

[14] HURSTI, H. The Black Box Report: Critical Security Issues with Diebold Optical Scan Design. Black Box Voting, July 4, 2005.

[15] HURSTI, H. Diebold TSx Evaluation: Critical Security Issues with Diebold TSx. Black Box Voting, Unredacted release July 2, 2006, May 11, 2006.

[16] HURSTI, H. Supplemental report, additional observations. Black Box Voting, Unredacted release July 2, 2006, May 22, 2006. http://www.blackboxvoting.org/BBVtsxstudy-supp.pdf.

[17] INGUVA, S., RESCORLA, E., SHACHAM, H., AND WALLACH, D. Source code review of the Hart InterCivic voting system. University of California, Berkeley under contract to the California Secretary of State, July 20, 2007.

[18] INTERCIVIC, H. BOSS Operations Manual 6100-019. Rev. 43-62A.

[19] JOHNSTON, R. G. Tamper-indicating seals. *American Scientist 94* (November-Decemeber 2006), 515–523.

[20] KIAYIAS, A., MICHEL, L., RUSSELL, A., AND SHVARTSMAN, A. A. Security Assessment of the Diebold Optical Scan Voting Terminal. UConn Voting Technology Research (VoTeR) Center, October 30, 2006.

[21] KIAYIAS, A., MICHEL, L., RUSSELL, A., AND SHVARTSMAN, A. A. Integrity Vulnerabilities in the Diebold TSX Voting Terminal. UConn Voting Technology Research (VoTeR) Center, July 16, 2007.

[22] KOHNO, T., STUBBLEFIELD, A., RUBIN, A., AND WALLACH, D. Analysis of an Electronic Voting System. In *Proceedings of the IEEE Symposium on Security and Privacy* (May 2004).

[23] MCDANIEL, P., BUTLER, K., ENCK, W., HURSTI, H., MCLAUGHLIN, S., TRAYNOR, P., BLAZE, M., AVIV, A., CERNY, P., CLARK, S., CRONIN, E., SHAH, G., SHERR, M., VIGNA, G., KEMMERER, R., BALZAROTTI, D., BANKS, G., COVA, M., FELMETSGER, V., ROBERTSON, W., VALEUR, F., HALL, J. L., AND QUILTER, L. EVEREST: Evaluation and Validation of Election-Related Equipment, Standards and Testing. http://www.sos.state.oh.us/, 2007.

[24] PROEBSTEL, E., RIDDLE, S., HSU, F., CUMMINS, J., OAKLEY, F., STANIONIS, T., AND BISHOP, M. An Analysis of the Hart Intercivic DAU eSlate. In *Proceedings of the USENIX/ACCURATE Electronic Voting Technology Workshop* (Aug. 2007).

[25] VAN HEESCH, D. Doxygen. http://www.doxygen.org.