Integrity Vulnerabilities in the Diebold TSX Voting Terminal

A. Kiayias L. Michel A. Russell A. A. Shvartsman

UConn VoTeR Center and
Department of Computer Science and Engineering,
University of Connecticut
{akiayias,ldm,acr,aas}@cse.uconn.edu

with the assistance of S. Davtyan, A. See, N. Shashidhar

July 16, 2007

Abstract

This report presents certain integrity vulnerabilities in the Diebold AV-TSx Voting Terminal[1]. We present two attacks based on these vulnerabilities: one attack swaps the votes of two candidates and another erases the name of one candidate from the slate. These attacks do not require the modification of the operating system of the voting terminal (as it was the case in a number of previous attacks). These attacks against the voting terminal can be launched in a matter of minutes and require only a computer with the capability to mount a PCMCIA card file system (a default capability in current operating systems).

The security problems are present in the system despite the fact that a cryptographic integrity check appears to be employed in the voting system's memory card. The attacks presented in this report were discovered through direct experimentation with the voting terminal and without access to any internal documentation or the source code from the manufacturer.

1 Introduction

Direct Recording Electronic (DRE) refers to voting terminals that use an interface to enable a voter to record his vote directly in digital format. The tallying is performed internally by the terminal that maintains counters for each candidate and race. DRE terminals have been criticized for lack of verifiability that they perform the tallying appropriately. As a result many DRE terminals today employ a VVPAT (Voter Verified Paper Audit Trail) system: the terminal is equipped with a printer that produces a record reflecting the choices of the voter; the voter is supposed to verify the VVPAT record. After the election it is possible to perform a manual count using the VVPAT records.

[1]Note: The AV-TSx voting terminals are quite different from the AccuVote Optical Scan terminals, and the vulnerabilities presented in this report do not apply to the Optical Scan terminals used by the State of Connecticut. The AV-TSx terminals are not used in Connecticut.

The Diebold TSx voting terminal was recently criticized in [5] and [6] due the following discovered security flaws: (i) it was possible to relatively easily circumvent the bootstrapping process and modify the operational environment of the system; this was identified in [5] where it was found that no cryptographic checks were performed during bootstrapping. (ii) the key management was by default using a fixed hard-coded key (that was leaked in the Internet); this was identified in [6] where the importance of choosing fresh signing keys was highlighted.

Fixing the above problems would require changes in the boot-loading process as well as adherence to an appropriate key management practice to be followed by election officials. In [6], it was reported that the AV-TSx has the advantage compared to other terminals such as the optical scan voting terminal (AV-OS) also from Diebold that it uses a cryptographic integrity check to make sure that the contents of the card have not been tampered.

Our Results. In our investigation we verify that there appears to be cryptographic integrity checking in the AV-TSx memory card. Nevertheless, we discover that the scope of the integrity checking is not as "wide" as it should have been. In particular, we find that in certain files that control the layout of the slate, the integrity checking is performed at the file level but not at the slate placement level. This flaw in the scope of the integrity check enabled us to modify the slate layout without triggering any alert from the terminal. Moreover, we found that when contents of slate components were invalidated the terminal did not issue an alert but instead it chose to simply suppress the corrupted file.

Based on the above vulnerabilities we design and test two attacks against the AV-TSx terminal. In the first, the attacker wishes to swap votes received by two candidates. The attacker can be successful in performing such attack assuming that the sizes of the two files that define the candidate representation in the digital slate are of the same size. We found that is not a rare occurrence and in fact our test election contained such pairs of candidates. The swapping was applied to the name definitions of the two candidates and included the integrity check. In the second attack, the attacker simply wishes to make one of the candidates disappear from the slate. By modifying the file that defines the layout of the name of the candidate this is achieved by the system.

The terminal we used in the above experiments is shown in Figure 1 and is manufactured by Diebold, Incorporated, Election Systems division. All our findings were based on reverse engineering and at no moment we had access to internal information about the terminal or access to source code.

Given the above findings, the employment of AV-TSx in an actual election becomes problematic. This is the case as the modification of a card can be done with merely a PC that is PCMCIA capable. If this terminal is used in an actual election it is extremely important to keep the memory card sealed in place. Moreover, it is very important to modify the operating system so that the integrity check is extended in all the card contents.

We also note that our terminal appear to lack the exact bootstrapping vulnerabilities that were reported in [5] (but lacking access to any internal information / system source code we are not able to vouch if the bootstrapping is any more secure now).

2 Security Vulnerabilities

This section discusses several security vulnerabilities in the AV-TSx. The attacks in section 3 focus mainly on the vulnerabilities with respect to the memory card (section 2.2), though the other issues mentioned here should be taken into consideration.

2.1 Basic Characteristics of the System

The system used in this study included the following components:

Figure 1: The AccuVote TSx voting terminal.

AccuVote TSx voting terminal:

- Firmware version 4.6.4
- Bootloader version BLR7-1.2.1
- Windows CE Operating system version WCER7-410.2.1

GEMS software version 1.18 install on a laptop.

Ethernet cable to connect the two above.

The GEMS software is used to manage the ballot information, load the election data onto the AV-TSx, and to receive the results after the election.

2.2 Memory Card

2.2.1 Description

The memory card is a standard PCMCIA flash card with a FAT file system. The card contents include the following file hierarchy

/ (root directory)
  Election Data/
    N.xtr
    N.edb
    M.adt
    K.brs
  Trashcan/

where N, M, and K are 32 character strings consisting of 0-9 and a-f (i.e., a 128 bit hex number). The .xtr file contains the election data information, the .edb file stores database information, the .adt file is the audit log, and the .brs file is the ballot box. The election data file is a bundle that contains many Rich Text Format (RTF) files for the displayed candidate names, wave files for use in audio voting, images displayed during the voting process and information about the precinct. All these files are packaged together in a single .xtr file along with 128 bit integrity checks for each of them. Votes are encrypted using 128 bit AES [8] and placed in the .brs file.

2.2.2 Vulnerabilities

Election Data and Database File Each candidate name (in an RTF file) is packaged with a 128 bit integrity check, however, these are not used correctly. A failed integrity check should render a voting machine unresponsive. However, in our terminal, a failed check of an RTF file simply makes that file not appear on the screen, effectively removing that candidate as an option.

The candidate names that are printed for the voter verified paper trail are based on the same RTF file that is displayed to the voter. However, the name printed for the final results is based on data from the .edb file. Because of this, voters could be unaware of any discrepancies between their cast votes and the internally recorded votes. Such a problem can only be detected by performing a manual count of the ballots from the VVPAT and comparing with the printed final counts.

There is also no global check to ensure the entire election data is correct. For example, the RTF files for candidates could be swapped (as we demonstrate below), along with their integrity check. A proper global integrity check should catch such manipulation.

Ballot Box There appears to be no global global cryptographic signature of the card contents to verify the contents were not tampered with outside the machine. Without this, it may be possible to stuff the ballot box by creating a custom ballot box file. This may depend on insider information to obtain the correct AES key and ballot format, but could be a threat nonetheless. Any changes to the memory card outside the voting terminal should result in an error.

Upgrade Files and Backdoors As documented in [4], previous versions of the TS machine were susceptible to attacks through back door files. If present on the memory card, the machine would give the user full access to the OS, for debugging purposes. For TSx machine it was documented in [5] that the back door files, with the different filenames, still exist. These files, if present on the memory card, will start to be processed based simply on their filenames; in this way an attacker can tamper with the boot loading process. We remark that the bootstrapping process in our TSx machine may still function as it is impossible to conclude positively that they are not working without having access to properly structured upgrade files. Still we tested the filenames that worked for previous versions and they no longer seem to function; moreover we have been unable to discover any similar backdoors as yet. However, without looking at the source code of the software being run on the machine it is impossible to say that there is no such back door still present in the system. A similar threat exists for the upgrading mechanism. In previous versions, only the name of the upgrade file was used to identify a valid software upgrade located on the memory card. Naturally, this represents a grave security vulnerability if no proper integrity checks are being used to authenticate the software upgrade. During our test we did not have examples of legitimate upgrade files, so we have been unable to test whether this vulnerability remains in the current version.

2.3 Internal Storage

The AV-TSx hardware includes internal flash memory in which it stores ballot information and voting results. This is used, for example, to accumulate results from several voting machines each with their own memory card. Each card is inserted in turn and an accumulator function in the ballot station software reads in the stored results (i.e., some accumulator values are maliciously preinitialized).

Vulnerabilities The accumulation functionality requires inserting each memory card into a AV-TSx terminal so that the results can be merged with those already stored on the internal flash memory. However, without source code, it is not clear precisely how the AV-TSx determines the data to be merged. In particular, it is unclear whether or not a AV-TSx terminal could ship with a set of election results already present which could be merged with valid results.

2.4 Limited Auditing Ability

The auditing features of the AV-TSx are limited to election results and system modifications, including turning on/off, loading data, and changing settings. However, there is no (documented) way in which to examine the software currently installed on the machine. Ideally one could (with proper access) dump the contents of the internal storage containing the operating system and voting software (or a hash of these contents) in order to verify the machine was not tampered with. In light of the possible lack of a secure method of loading new software, as mentioned above, this auditing ability maybe useful.

3 The Attacks

The attacks we present were developed with precisely the same information and access to the system that is normally available to, for example, election administrators (supervisors, poll workers and other town officials). Note that to carry out the attack, one only needs physical access to the voting machine, without the privileges of an election administrator. An attacker only needs a few minutes with the card and a hex editor to perform the attack. In addition, an attacker may need to open the lock which covers the removable card. Furthermore, the attacker needs no knowledge of the particulars of the election he is to undermine (such as exact candidates' names, ballot layout, precinct names, or any kind of passwords). What the attacker needs is to find two .rtf strings which have the same size (first 4 bytes of the .rtf string contain the .rtf file size) within the .xtr file. The whole process can be completed in a matter of a few minutes. In the following we perform a step-by-step demonstration of the attack.

3.1 Gaining physical access

Prior to the election the terminal is presumably locked within the ballot box. The first thing an attacker must do is to gain access to the memory card of the AV-TSx machine that is concealed by the ballot-box. If the box is unlocked or the attacker has the keys this is straightforward. The fact that DIEBOLD appears to be using the same keys across machines makes it easier to unlock the ballot-box (we had two terminals and they both shared the same keys). From what we have found online, in contrast to the keys used by DIEBOLD for OS machines, the keys for AV-TSx machines are very difficult to copy because they do not use a standard size. However, a copy of this key is sent to every precinct as part of the supply kit. Additionally the keys assigned to each location are not individually numbered, nor is there any record of which key is assigned to each precinct [11].

3.2 Reading the memory card contents

Once the PCMCIA card is accessible the attacker can have an immediate access to its contents through the PCMCIA card reader. Note, that a lot of laptops nowadays are equipped with a PCMCIA card reader.

3.3 The details

In order to understand the specifics of the attack, the following gives an overview of the election setup of the AV-TSx system. The removable memory card with the election loaded on it contains four different types of files: BRS, ADT, EDB and XTR. The .xtr file contains the ballot data. The .edb file, which has the same name as the .xtr file, contains the encrypted data from the Election Data Base. The .brs file stores the votes. The .adt file stores the Machine Log. For the attacks we present we are particularly interested in the .xtr file contents. This file bundles .rtf, .wav, and .bmp files. The .rtf files include the candidate names and voting instructions. The .wav files include the candidate names spoken. There are two .bmp files that illustrate the insertion of the voting ID into the machine. Each file is stored by the following way:

4 bytes - filesize(N); N bytes - data; 16 bytes - checksum.

What the attacker needs to do is to find two candidates for which the .rtf file sizes are the same and swap these two .rtf files with their corresponding checksums. Note, if the checksums are not swapped, thus the data do not correspond to the checksum, the voting software will simply not display this file. There are two possible scenarios for carrying out the attacks. First is just to nullify the candidate name, and the second is to swap two candidates.

Nullifying Attack: As it was discussed earlier each .rtf file consists of 4 bytes file size, following the N bytes data, and 16 bytes checksum. If the checksum is not consistent with the data it will result in nullifying the candidate name. Thus, if a single bit is changed in the data part of the .rtf file, without altering the length of the file, then the corresponding .rtf file will simply not be displayed on the screen without causing any error while loading the election. For example, we attempted to alter a candidate's .rtf file by replacing a 'C' with a 'D' The corresponding cell is left blank, but the checkbox remains. Voting proceeds as usual. When printing the ballot, if there were no votes for the (now blank) candidate, then it is printed with no name for that candidate. For example, if we would originally have

....

[X] THOMAS C. XXXXXX

....

We now have

....

[X]

....

An example of the original untampered slate is given in the Figure 2.

The exact same slate after a candidate has been nullified is given in Figure 3.

When the election is finalized, the results are printed using the candidate's original name. This means that the name is in fact stored in two places:

1. A label in a database record
2. Within the formated .rtf file

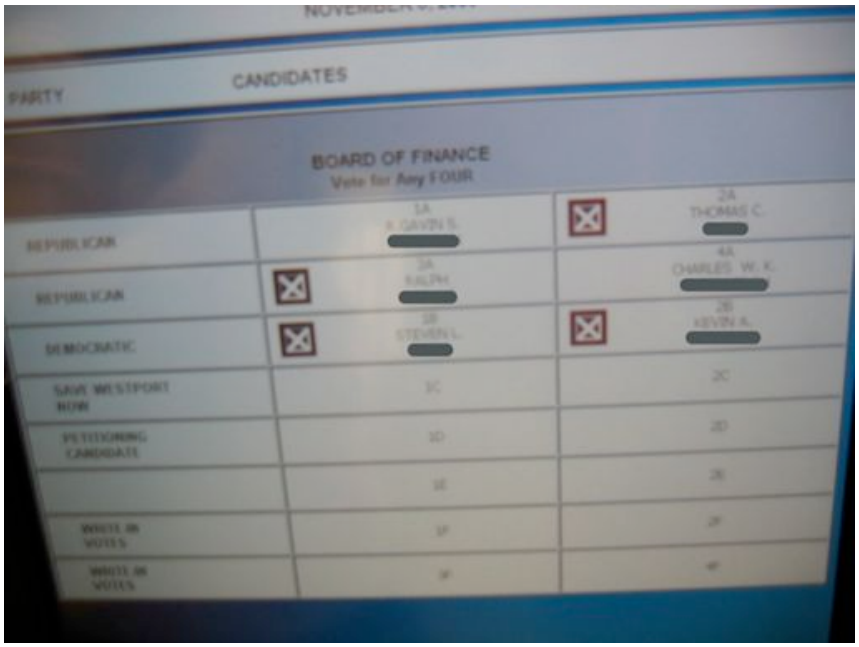

Figure 2: The original, not untampered, slate. Some choices have been made by the voter.

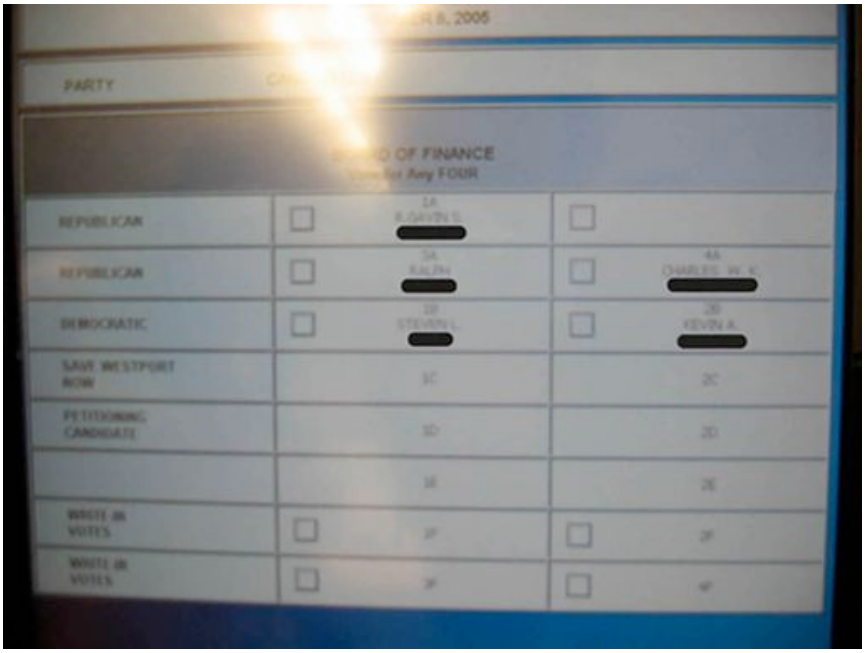

Figure 3: The slate with the nullified candidate name.

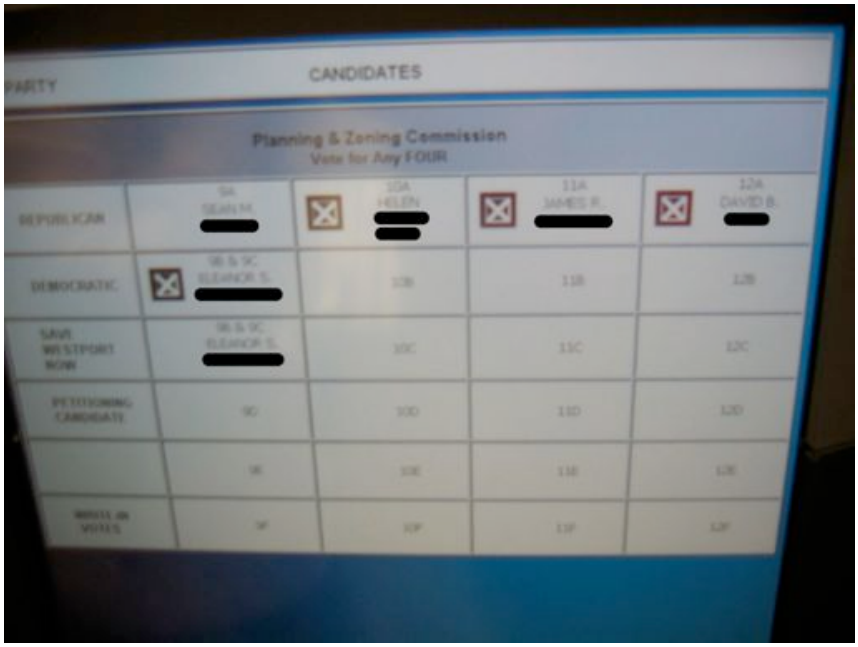

Figure 4: The ballot with unaltered candidates' names (before swapping)

Both of these appear in the GEMS database. Only the .rtf file is visible in the clear within the card contents though. The database label must be either encrypted or compressed with other data. The database label is used on the zero report and the final report, while the .rtf file is displayed on the screen and printed on the printed ballot.

Furthermore, we ran an election with two machines, each with a memory card, one of which was tampered in the aforementioned way (not displayed). After finishing the election, the results can be combined on the AV-TSx with no errors. That is, there is no check of whether the .xtr files match. Any votes for the blank spot would be assigned to the candidate that originally should have appeared there. The results can then be uploaded to GEMS with no errors.

Swapping Candidates: With the above observations, we then tried to swap two candidates with .rtf files of equal length along with their checksums. We again held a two machine election, swapping the .rtf files for one machine only. The slate presented by the untampered machine is given in Figure 4.

The tampered machine ran correctly, with the two candidates swapped on the screen compared to the untampered machine. Candidates 'DAVID B. XXXXX' and 'SEAN M XXXXX' are swapped (Figure 5).

We then voted twice for one of these candidates, candidate 'DAVID B. XXXXX', on each machine (with the original and tampered elections loaded correspondingly). The votes on the screen agreed with that on the printed VVPAT records (two for 'DAVID B. XXXXX') in both cases (see the scans of the records in Figure 6, Figure 7). Thus it appears that the election ran correctly and a voter can verify that the printed record indeed corresponds to the choices made on the screen. However, the final results on the tampered machine showed two votes for candidate 'SEAN M XXXXX' and zero for candidate 'DAVID B. XXXXX' (Figure 8). On untampered machine printed ballots and the results correpond to each other (Figure 7 and Figure 9 correpondingly). In this case we also were able to combine the results and send the tally to GEMS, with no errors. The results show two votes for each candidate 'DAVID B. XXXXX' and 'SEAN M XXXXX' correspondingly (Figure 10), even though during the election no votes were given to the candidate 'SEAN M

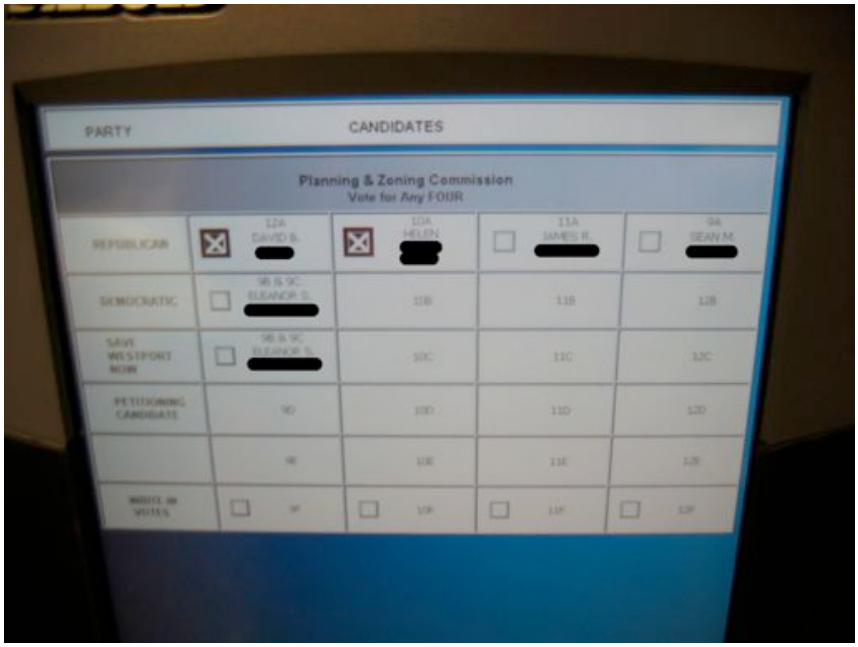

Figure 5: The ballot with swapped candidates' names

XXXXX'.

Conclusion: if an attacker has access to the memory card and two candidates have names of the same length, then the attacker can swap their votes on that machine. Note, the names themselves do not have to be the same size, but the .rtf files (these files contain formatting, such as spaces, newlines, and font information thus names of equal number of characters may still result in differently sized rtf files).

3.4 Completing the attack

Once all the changes have been made with the corresponding .xtr file the memory card is ready to be placed back. After this step, the AV-TSx terminal will be found by poll-workers in its expected pre-election state. The terminal will appear to be functioning normally for all operations during the election. The total time required to compromise the card is only a few minutes, depending on the dexterity of the attacker in picking the lock of the ballot box.

4 Conclusions

In this report, we performed a security analysis of the AV-TSx system and demonstrated two serious attacks against the integrity of the election process. It is important to point out the fact that we did not possess the source code for the voting terminal or GEMS, nor did we need any specialised equipment. Compromising a terminal takes a few minutes with the card and a hex editor. Most laptops these days are equipped with PCMCIA card readers, obviating the need for a special card reader. In the light of these attacks, it suggests that great caution is warranted before employing AV-TSx in an actual election. Some recommendations follow.

Figure 6: The votes represented on the printed ballot (altered case)

4.1 Improving the security of the AV-TSx

There have been several studies (e.g., [2, 9, 10]) that have specifically addressed the issue of designing e-voting systems and offering recommendations for improvement. Here, we point out the particular shortcomings of the AV-TSx terminal and identify aspects that need to be dealt with to obtain a secure and robust system.

Global Integrity Check. The memory card of the AV-TSx, a standard PCMCIA card, as discussed before, holds the election data, ballot box and the audit information. The major shortcoming in this regard is the lack of a global integrity check computed on the entire contents of the card. Our attacks were possible because of the lack of such a global check.

Modified Election Data Files and Integrity Checks. The .xtr file contains the names of the candidates in RTF format. Each .xtr file does have a 16 byte integrity check. A failed integrity check should under reasonable assumptions put the machine in an "insecure" state and have an alert presented. However, in the AV-TSx, a failed check of an RTF file simply makes that file not appear on the screen. A cryptographic check would not be effective if a failed check is not handled appropriately by the system.

Inconsistent File Usage. The candidate names that are printed for the voter verified paper trail are based on the same RTF file that is displayed to the voter, while the name printed for the final results is based on data from the .edb file. Because of this, a modified .xtr file may go undetected by initial testing by poll workers. The slate options displayed to voters should correspond exactly to the choices displayed on the final results.

Backdoor Files. Previous versions of the TS machine were susceptible to attacks through back door files [4]. Such files, if present on the memory card, gives the user unrestricted access to the terminal for debugging purposes. It is unclear whether there exist any backdoor files in use for the current AV-TSx terminal; further investigation would be necessary to make sure that such backdoors do not exist.

Figure 7: The votes represented on the printed ballot (unaltered case)

Figure 8: The results of the election held on a tampered machine

Figure 9: The results of the election held on unaltered machine

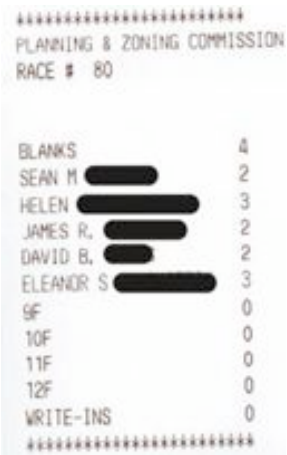

Figure 10: The results accumulated from both machines

Limited Software Accountability and Auditability. There is no (documented) way in which to examine the software (Operating System) currently installed on the machine.
Internal Flash Accumulator. The AV-TSx has the ability to accumulate voting results from several voting machines, each with their own memory card. This potentially may entail the possibility to "stuff" the ballot box prior to the machine being used. Further testing would be required to make sure that the terminal is not susceptible to this attack.

4.2 Safe Use Recommendations

In 2002, in the United States, the Help America Vote Act mandated that one handicapped accessible voting system be provided per polling place, which most jurisdictions have chosen to satisfy with the use of DRE voting machines, some switching entirely over to DRE. Thus, it is imperative to follow safe-use guidelines before deploying these terminals. Here, we outline a set of safe-use recommendations for the AV-TSx and extend to electronic voting machines in general.
Tamper Evidence. Any access point or storage media should be sealed in a tamper evident fashion. On the AV-TSx in particular, this includes the memory card PCMCIA slots, but on other systems could include network ports, PCMCIA slots, USB ports, serial ports, phone jacks etc. Indeed, the presence of a USB port on a voting machine with an operating system that has sufficient drivers could allow the connection of keyboards, flash memory devices, or other common devices in order to take control of the system.
Chain of Custody. Assuming that the machine is initially prepared by trustworthy individuals, the problem then is to ensure that the machine remains physically secure at all times prior to election day. As mentioned, the simple lock on the ballot box used for the AV-TSx machine was insufficient since it can be easily picked or a key obtained without much difficulty. The machines must also be secured after the election in case auditing is deemed necessary, especially in the absence of a paper trail to audit.
Removable Media. Voting terminals should perform a cryptographic integrity check on their removable devices. Without this capability, any removable device (e.g., a memory card) must be sealed and any removed device should be considered compromised.
Random Audits. Post-election random audits of the voting machines coupled with manual recounts would help identify faulty or compromised voting machines. In [3], a simple method for sampling precincts in an observable way is presented. There remains a concern, however, that the physical ballots that will be manually recounted are machine generated (as opposed to actually produced by the voters). This could open the door for biasing the election results as described in [1].

References

[1] Brennen Center Task Force on Voting System Security. *The machinery of democracy: Protecting elections in an electronic world, 2005*. Lawrence Norden, Chair. Brennen Center for Justice, NYU School of Law. www.brennancenter.org

[2] David Chaum, Peter Y. A. Ryan, Steve A. Schneider, A Practical Voter-Verifiable Election Scheme. ESORICS 2005, pp. 118-139.

[3] Arel Cordero, David Wagner, and David Dill. The Role of Dice in Election Audits – Extended Abstract, IAVoSS Workshop On Trustworthy Elections (WOTE 2006), June 29, 2006

[4] Ariel J. Feldman, J. Alex Halderman, and Edward W. Felten
Security Analysis of the Diebold AccuVote-TS Voting Machine
http://itpolicy.princeton.edu/voting/

[5] Harri Hursti, Diebold TSx Evaluation, Black Box Voting Project, May 11, 2006
http://www.blackboxvoting.org/BBVtsxstudy.pdf
http://www.bbvdocs.org/reports/BBVreportIIunredacted.pdf

[6] David Wagner, David Jefferson and Matt Bishop, Security Analysis of the Diebold AccuBasic Interpreter, Voting Systems Technology Assessment Advisory Board, University of California, Berkeley, February 14, 2006.

[7] Election Data Services Inc. 2006 Voting Equipment Study, February 6, 2006.
http://www.electiondataservices.com/EDSInc_VEStudy2006.pdf

[8] Diebold Election Systems FAQ http://www.diebold.com/dieboldes/faq.asp

[9] Rebecca Mercuri, A Better Ballot Box?, IEEE Spectrum, Volume 39, Number 10, October 2002.

[10] D. Molnar, T. Kohno, N. Sastry, and D. Wagner, Tamper-Evident, History-Independent, Subliminal-Free Data Structures on PROM Storage -or- How to Store Ballots on a Voting Machine. Extended abstract, in IEEE Security and Privacy, 2006.

[11] http://www.law.csuohio.edu/CERP/documents/CERP_A-E.pdf/