black box voting
CONSUMER PROTECTION FOR ELECTIONS
www.BlackBoxVoting.Org

330 SW 43rd St Suite K PMB 547 Renton WA 98055
425-793-1030 http://www.blackboxvoting.org

Diebold TSx Evaluation

SECURITY ALERT: May 11, 2006
Critical Security Issues with Diebold TSx

Note: Please refrain from speculation or public discussion
containing inappropriate technical details.

A Black Box Voting Project
Prepared by: Harri Hursti

On behalf of Black Box Voting, Inc.
A nonprofit, nonpartisan, 501c(3) consumer protection group for elections

Executive Summary

Due to the nature of this report it is distributed in two different versions. Details of the attack are only in the restricted distribution version considered to be confidential.

This document describes several security issues with the Diebold electronic voting terminals TSx and TS6. These touch-pad terminals are widely used in US and Canadian elections and are among the most widely used touch pad voting systems in North America. Several vulnerabilities are described in this report. One of them, however, seems to enable a malicious person to compromise the equipment even years before actually using the exploit, possibly leaving the voting terminal incurably compromised.

These architectural defects are not in the election-processing system itself. However, they compromise the underlying platform and therefore cast a serious question over the integrity of the vote. These exploits can be used to affect the trustworthiness of the system or to selectively disenfranchise groups of voters through denial of service.

Introduction

The Diebold AccuVote touch-screen (Direct-Recording Electronic, or "DRE") voting terminals TS6 and TSx are used in hundreds of jurisdictions and many different states and provinces in the US and Canada, respectively. They are among the most popular DRE voting machines.

Voting terminals TS6 and TSx employ custom made hardware running with an embedded Windows CE operating system. As is true for all Windows CE systems, they require a boot loader to prepare the hardware for the launch of operating system. Both the boot loader and the operating system are custom built specifically for the unique hardware of the terminals.

The TS6 and TSx do not share the same core level architecture. For example, they have different CPUs. Furthermore, they have been designed by different engineering companies.

As part of the typical engineering process, the hardware and Windows CE customization is interwoven and performed simultaneously. Due to the heavy customization required for embedded operating systems to meet the hardware requirements and the nature of their environment of use, it is difficult to support the argument that these systems are "Commercial Off The Shelf" (COTS) operating systems like their desktop counterparts. Instead, the operating system itself is custom-built for each and every platform separately by combining the Operating System (OS) core with platform specific modifications and drivers. In the case of Windows CE, the tool to build the operating system is Microsoft Platform Builder.

Based on the tape recorded public meeting in Emery County on March 27, 2006[1], the TSx comes with at least three different revisions. There is no documentation available to as the extent the hardware revisions differ from each other, nor to which extent modifications are needed in the boot loader and/or Operating System builds. An Emery County system that was inspected and sold as new in early 2006 appears to be revision A. This machine had a PXA250 CPU and a MediaQ display controller. At least the revision A architecture has end-of-the-life-cycle components which indicate the need for re-engineering and modification to low-level design and programming. At least one version requires a different boot loader or Operating System build due to hardware changes.

Three-layer architecture, 3 security problems
Each can stand alone or combine for 3-layer offense in depth

As an oversimplification, the systems in question have three major software layers: boot loader, operating system and application program. As appropriate for current designs, the first two layers should contain all hardware specific implementations and modifications, while the application layer should access the hardware – the touch pad, memory card, the network etc. – only via services and functions provided by the operating system and therefore be independent of the hardware design. Whether the architecture in question follows these basic guidelines is unknown.

Based on publicly available documentation, source code excerpts and testing performed with the system, there seem to be several backdoors to the system which are unacceptable from a security point of view. These backdoors exist in each of these three layers and they allow the system to be modified in extremely flexible ways without even basic levels of security involved.

Different files carry various subsets of the following features: Signature check, mode check and integrity check. None of these can be considered security features against tampering. For example, the integrity check is [redacted]. This check can be equated to a very crude spell-checker. It is effective against accidental typing errors but not deliberate attacks.

In the worst case scenario, the architectural weaknesses incorporated in these voting terminals allow a sophisticated attacker to develop an "offense in depth" approach in which each compromised layer will also become the guardian against clean-up efforts in the other layers. This kind of deep attack is extremely persistent and it is noteworthy that the layers can conceal the contamination very effectively should the attacker wish that. A quite natural strategy in these types of situations is to penetrate, modify and make everything look normal.

Well documented viral attacks exist in similar systems deploying interception and falsification of hash-code calculations used to verify integrity in the higher application levels to avoid detection. The three-level attack is the worst possible attack. However, each layer can also be used to deploy a stand-alone attack. The TSx systems examined appear to offer opportunities for the three-level attack as well as the stand-alone attacks.

Unlike the desktop versions of Windows, the embedded versions of Windows CE 3.x and 4.x versions used in the Diebold system (which are both noncurrent versions) have very limited security features against a user with access below the application level. Because of the lesser security available in Windows CE, access to the standard Windows Explorer application grants users access to replace and modify files almost without restriction. This enables a hostile attacker to severely alter the system functionality and/or add new software (and hidden processes) to the system.

In addition to altering individual files, the TSx and TS6 systems also present opportunities to change the Operating System itself. This provides possibilities for hiding the attack and/or altering the application's behavior without any changes to the application itself. A major contributor to this is the ability to change the Operating System functions and libraries any application software relies on at a deep level.

It is important to understand that these attacks are permanent in nature, surviving through the election cycles. Therefore, the contamination can happen at any point of the device's life cycle and remain active and undetected from the point of contamination on through multiple election cycles and even software upgrade cycles.

Here is a rough analogy:

- The application can be imagined as written instructions on a paper. If it is possible to replace these instructions, as it indeed seems, then the attacker can do whatever he wishes as long as the instructions are used.

- The operating system is the man reading the instructions. If he can be brainwashed according to the wishes of the attacker, then even correct instructions on the paper solve nothing. The man can decide to selectively do something different than the instructions. New paper instructions come and go, and the attacker can decide which instructions to follow because the operating system itself is under his control.

- The boot loader is the supreme entity that creates the man, the world and everything in it. In addition to creating, the boot loader also defines what is allowed in the world and delegates part of that responsibility to the operating system. If the attacker can replace the boot loader, trying to change the paper instructions or the man reading them does not work. The supreme entity will always have the power to replace the man with his own favorite, or perhaps he just modifies the man’s eyes and ears: Every time the man sees yellow, the supreme being makes him think he is seeing brown. The supreme entity can give the man two heads and a secret magic word to trigger switching the heads.

In the world of the Diebold touch-screen voting terminals, all of these attacks look possible. The instructions (applications and files) can be changed. The man reading the files (Windows CE Operating System and the libraries) can be changed. Or the supreme entity (boot loader) can be changed, giving total control over the operating system and the files even if they are "clean software."

1) Boot loader reflashing

The prime responsibility of the boot loader is to set up the system hardware, ready it for launch of the operating system, and then launch it.

In the development phase of the system, additional features for debugging and flexible software testing cycles are often needed. It is the standard practice to remove these features from the release versions, even when security is not a concern at all.

Most importantly, the Diebold boot loader for both TS6 and TSx releases seems to contain the full capability to reflash itself and the operating system. (Reflashing refers to the capability to reprogram the flash memory which acts as the permanent storage media for the platform in question.) Additionally, there seem to be a number of other development features not as easily accessible, including boot monitor and diagnostics.

Furthermore, the boot loader seems to be network aware and supporting modified boot orders between permanent on-board memory, removable media and the network. This document and the testing done focus on use of removable storage, a standard PCMCIA (Personal Computer Memory Card International Association) memory card, as the delivery mechanism of the new boot loader, operating system and applications. An examination of the motherboard indicates that other delivery mechanisms also exist and these will be discussed briefly as well.

In the boot-up process after the primary hardware initialization phase is complete, the boot loader will, in the case of existence of standard memory card in the PCMCIA slot, mount it as a [redacted].

In the case of the TS6, the boot loader will look for the existence of any file with 8.3 filename:

[redacted]
[redacted]

and in the case of the TSx :

[redacted]
[redacted]

If these files can be found, those files will start to be processed based simply on the fact that the filename was right ("trusting the filename"). If the files pass rudimentary integrity and a file mode checks as they easily should, the boot loader will automatically read the file and write that to the machine as a new boot loader. This process is destructive for the pre-existing boot loader and there seem to be no fail-safe mechanisms. The reprogramming starts automatically without any interaction with the user. Due to the highly destructive nature of this attack, this process was not tested in Emery County. These observations are based on analysis of the documentation available.

Due to the fact that the boot loader is the primary mechanism for its own reprogramming, if the boot loader is suspected to be compromised with a deep attack, using the boot loader itself to install a known clean version of a boot loader is no longer a viable option as a recovery path to clean the system.

At the time of this writing, existence of a safe recovery path for the TS6 from any suspected boot loader compromise is unknown.

Based on an examination of the motherboard, the TSx appears to have a hardware-level interface which can facilitate initial programming of the boot loader. This interface is accessible with a [redacted], enabling the auxiliary system to take over [redacted] of the motherboard.

Fig. 1 [Redacted]

At the time of this writing it seems reasonable to assume that this mechanism will enable the auxiliary system to prevent starting untrustworthy code execution onboard, allowing reprogramming of the flash memory to provide a safe recovery path.

It is unknown if this mechanism can be used to retrieve data for forensic studies from a system suspected of contamination, because the reprogramming operation is destructive and prevents any other forensic studies.

Unfortunately, the same mechanism used to provide a safe recovery path can be employed just as easily for a delivery mechanism for malicious code to the system. Due to the very nature of the process, no software-based security mechanism can provide a remedy against this type of attack.

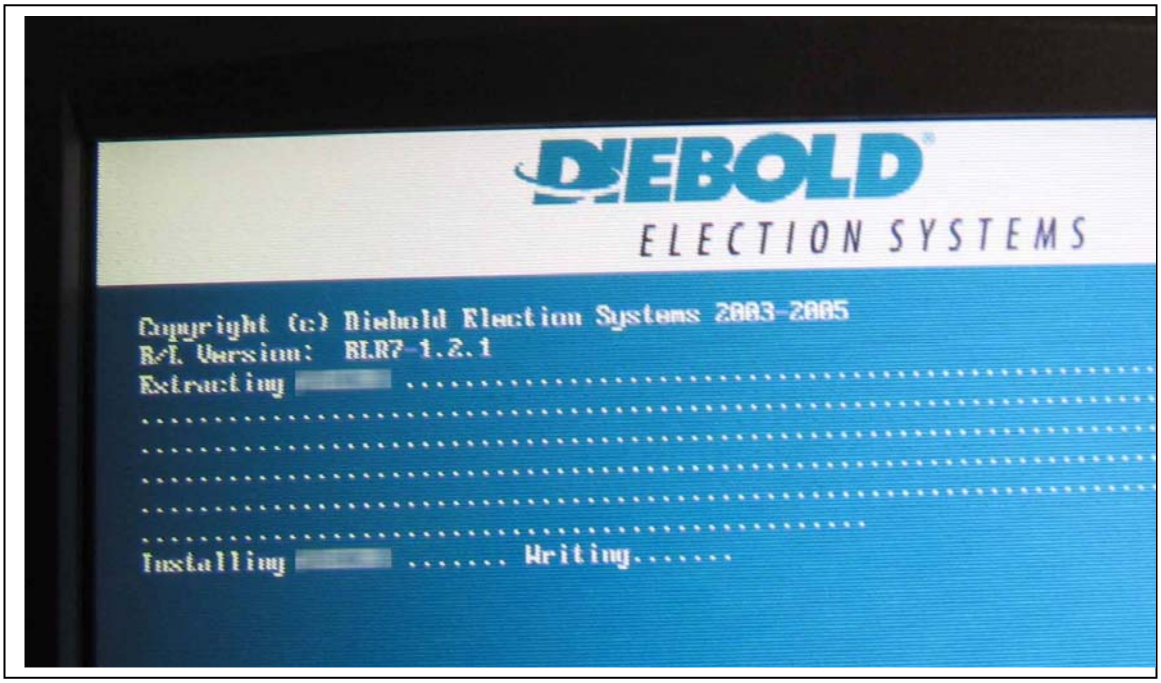
2) Operating system reflashing

After processing of the new boot loader files, the boot loader will continue without any additional soft boot. This means that the programming code being executed can be temporarily different than the code in storage media, until the next reboot. At the reboot all traces of this temporary discrepancy will be erased.

Next, image files for new operating systems are searched. Unlike a standard desktop computer operating system, in embedded systems it is customary to deliver the whole runtime-ready operating system as [retracted]. The boot loader will look for the following files:

[redacted]
[redacted]
[redacted]

As in the previous case, the file processing is launched based on the correct file name alone. The [redacted] file is assumed to contain the [redacted]. The file will be processed from the memory card without user interaction, overwriting the previous content and therefore destructive for future forensic studies.



```
Copyright (c) Diebold Election Systems 2003-2005
B/L Version:  BLR7-1.2.1
Extracting [redacted] ..............................................................
................................................................................
................................................................................
................................................................................
................................................................................
....................................................................
Installing [redacted] ....... Writing.......
```

Fig. 2 – Photo taken during replacement of the Windows CE operating system on TSx machine

There are no cryptographic signatures or other security related measures involved. Replacement of the Windows CE operating system file is performed without even the most basic level of source, authentication or compatibility testing, allowing even code that is impossible to execute to be installed.

[1 redacted sentence]

Other mechanisms exist in addition to the PCMCIA card for replacement of the Windows CE operating system. Network awareness also enables a configuration where the Windows CE image will be downloaded from a remote network device. While not the focus of this document, the protocol seems to be standard and lacking implementation of security features.

3) Selective file replacement

The Diebold touch-screen voting application is called "Ballot Station."

After the boot loader has launched the Windows CE operating system, the start-up phase of the Ballot Station application begins with a custom-made start-up program. Before starting the existing Ballot Station application, the memory card is searched for existence of any files with an [redacted] extension. All files matching this criteria will be processed sequentially. The [redacted] file is a Diebold proprietary format batch overwrite install file, which can encapsulate multiple files to be replaced in the system.

This install procedure does check [redacted] as a measure to pre-qualify the file for processing. Unlike in previous phases, the file also contains a description field and a public version number to be displayed to a user in dialog for acceptance of the batch. Whereas the operating system will be replaced automatically with no questions asked of the user, the [redacted] file will request user approval before installing the files.

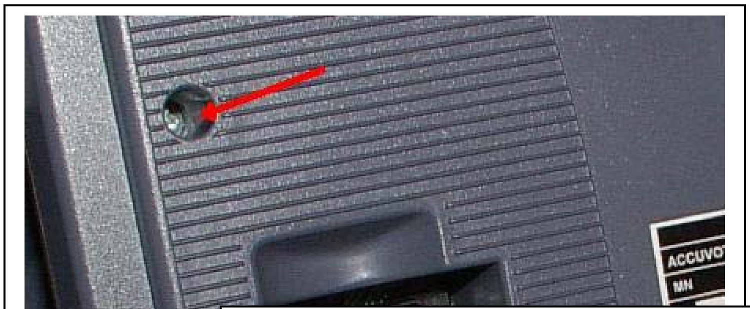
No system log entries will be produced when [redacted] files are processed, not even when rejected or invalid files are getting processed.

Additional concerns

4) Removable non-secured casing

All of the above attacks are persistent in nature. The attacks can be deployed any time during the life cycle of the machine. It is safe to assume that a sophisticated attacker can install an election-independent core of the attack engine into the machine years ahead, delivering election-specific instructions to the engine by various easy delivery mechanisms available to, for example, any voter.

Operational procedures to lock and seal the machines before sending the machines to the homes of poll workers (as is customary in many jurisdictions) are in most cases not adequate. The TSx casing is affixed with simple standard phillips screws. When unscrewed, the back end of the casing comes off with the locked bay doors configured such that seals remain intact. When the casing is open, access to PCMCIA slots is unrestricted.



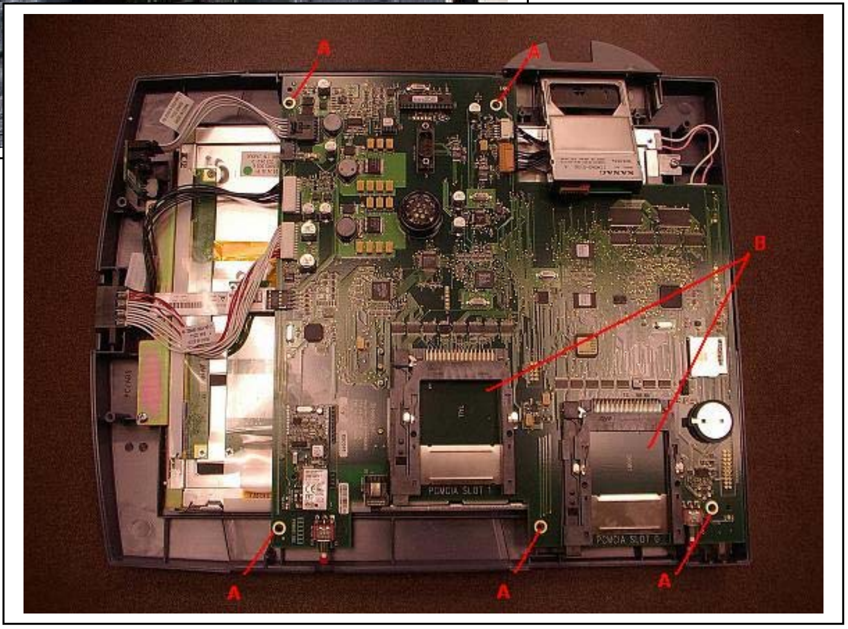


Fig. 3 – Casing is removed quickly and easily with a Phillips-head screw driver. At the time of this writing, removing and replacing the casing leaves no telltale signs.

Fig. 4 - Diebold TSx motherboard
A = Phillips head screw holes
B = PCMCIA card slot

5) Hidden SD slot

The TSx unit also comes with a hidden MMC/SD – a standard slot (Multimedia Card / Secure Data). These types of removable memory cards are standard components in many home consumer electronic devices, and the standard card has grown from its original purpose to a flexible general purpose interface, which can hold vast amounts of data in the gigabyte range, and also facilitate other types of peripheral functionalities, like networking. The slot is designed with enough room to facilitate other types of SD cards besides simple memory cards. The SD slot is always active and once the casing is open it is accessible. The presence of the SD card is undetectable when the case is closed.

It is unknown what support drivers are installed with Diebold-provided operating systems, but since additional support features can be added, a sophisticated attacker can, for example, introduce wireless capabilities to facilitate attack even if the system was not originally configured for wireless communication.

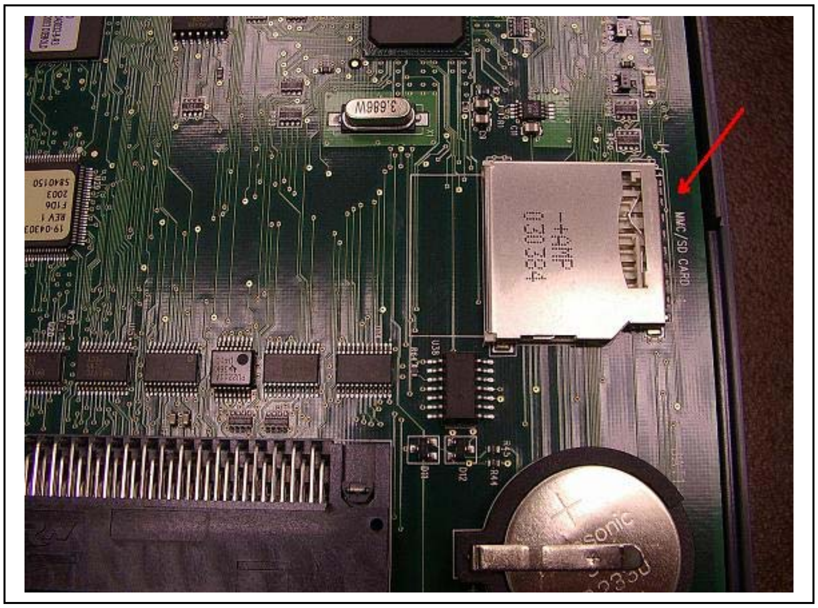


Fig. 5 – SD card slot on Diebold TSx motherboard

6) Jumper-enabled additional features

The motherboard also has jumpers to enable otherwise disabled software features. Based on the documentation, the Diebold standard implementation has debug features built in but disabled in the absence of having the jumper connected. Again, these features enable various simple attack vectors against the system.

Fig. 6 – [redacted]

7) Software controlled voter accessible hidden button

The TSx also has an unmarked button hidden in the casing. On the circuit board, this switch is labeled "battery test". The switch is physically similar to many reset buttons, necessitating application of substantial force to press the button, requiring it to be depressed by about 1/5 - 1/6 inch in order to activate the switch. This switch is also software accessible. It is completely accessible for all voters in the standard voting booth configuration. The logic behind the button is unknown, but for an attacker it presents yet another way to interact with the machine, and an exceptionally convenient button switch for an attack designed to be triggered by a voter.

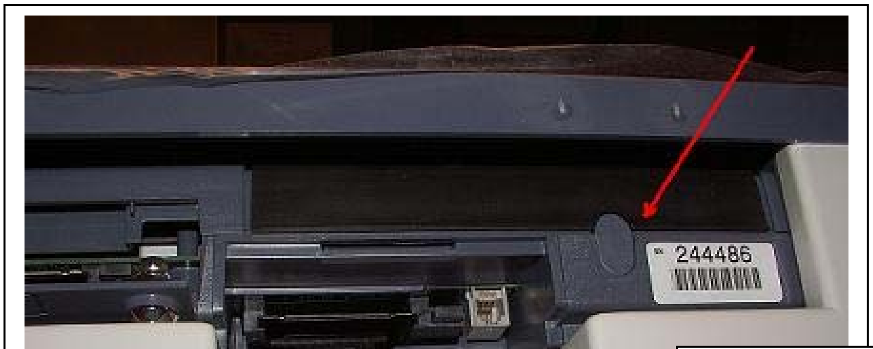


Fig. 6 – Voter-accessible hidden button – Exterior view

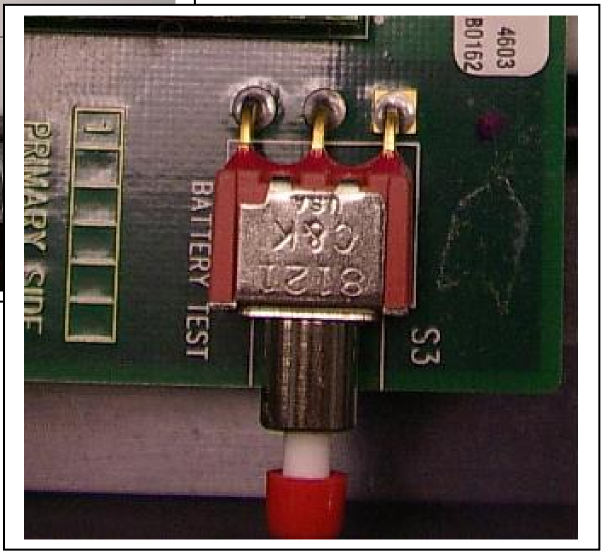


Fig. 7 – Voter-accessible hidden button – Interior view

Conclusions and Recommendations

Because there is no way of having chain of custody or audit trail for machines, the machines need to be reflashed with a known good version (assessing the risks potentially inherited). Ideally this should be done by the proper governmental authorities rather than being outsourced.

After that, extensive chain of custody management has to be established to make sure that machines do not potentially get recontaminated. Less than five minutes is required for contamination.

The bootloader needs to be re-engineered.

The cases need to be properly and permanently sealed.

Further study is warranted around these issues and others in the May 15, 2006 Supplemental Report for the Emery County TSx study.

While these flaws in design are not in the vote-processing system itself, they potentially seriously compromise election security. It would be helpful to learn how existing oversight processes have failed to identify this threat.

FOOTNOTES

[1] Tape recorded Emery County meeting with state elections director, county commissioners and Diebold attorneys, March 27, 2006

[2] Files found by Bev Harris on Diebold FTP site Jan. 23, 2003.

ACKNOWLEDGEMENTS

The citizenry owes an immense debt of gratitude to Bruce Funk, the Emery County Clerk for Emery County, Utah who, upon noticing anomalies in the Diebold TSx machines delivered to his county, requested an independent evaluation of this voting system.

Appreciation is expressed to Kalle Kaukonen for providing his perspective on this report.