Global Information Assurance Certification Paper

Electronic Voting:

An examination after the 2004 Presidential Election

Matthew M. Kinard
GIAC Security Essentials Certification (GSEC)

Version 1.4c
Option 1

11 January 2005

Abstract

The United States of America experienced what might be called a voting machine renaissance in the wake of the 2000 presidential election. As the 2004 presidential election approached and has since passed, the effort to improve the United States voting infrastructure was brought into the public's eye. The results of this effort are mixed. This paper will re-examine some of the important issues brought up since 2000, with emphasis on the inherent security problems that come along with electronic voting. It will begin with a brief lesson on the history and features of voting machines. We will follow that with a look at what is being done to combat security issues by thoroughly examining the software and implementation of a popular voting machine. Finally, there will be a report on the successes and failures of electronic voting in the 2004 presidential election.

Introduction

It has been said that the right to vote is something that many United States citizens take for granted. While this may be true, it would seem that in recent years something more fundamental than the right to vote has been taken for granted: the assurance that our elections are run fairly, accurately, and transparently. Since the 2000 presidential election, a great deal of time, money and effort has gone into the revamping of the United States voting infrastructure, much of which was urged on by the 2002 Help America Vote Act (Mercuri). While the ideas behind this act are noble, the rush to modernize the United States' voting has opened up the possibility for many unknown and undesirable effects.

We will examine a brief history of voting machines in the United States, and explore the ramifications of the new Direct Recording Electronic, or DRE, voting machines now being employed throughout the United States. We will spend some time examining some of the major security issues that are present in the DRE machines, and then look at several possible solutions to the problems at hand.

Important Features of Voting Machine Technology

It is important to begin any discussion of voting machines with a quick look at what qualities are important in a good election system. In his paper "Electronic Voting – Evaluating the Threat," Michael Ian Shamos proposes the following Six Commandments as a light-hearted, but thoughtful guideline for the examination of voting systems:

I. Thou shalt keep each voter's choices an inviolable secret.
II. Thou shalt allow each eligible voter to vote only once, and only for those offices for which she is authorized to cast a vote.
III. Thou shalt not permit tampering with thy voting system, nor the exchange of gold for votes.
IV. Thou shalt report all votes accurately.
V. Thy voting system shall remain operable throughout each election.
VI. Thou shalt keep an audit trail to detect sins against Commandments II-IV, but thy audit trail shall not violate Commandment I.

The order of the commandments is important. There are very few voters who would be willing to use a machine that violates any of the first three commandments (Shamos). We find though, that the last three are slightly more flexible. Many election officials are willing to accept small tabulation errors in an election, and most people who are familiar with technology are used to the inevitable failures that are associated with computers. We also find that people

routinely participate in elections where the audit trail is less than perfect (Kohono et al 17).

The six commandments can also be examined in terms of the three bedrock principals of computing security: Confidentiality, Integrity, and Availability, or CIA. Confidentiality covers the first commandment, and integrity the second, third, fourth and sixth. Finally, availability equates to the fifth commandment. It is clear that after the issues that arose during the 2000 presidential election, any new voting system will have to address and account for all of these principals in some form or fashion. Before we begin the exploration of these issues within the context of modern, direct recording electronic voting systems, we should briefly explore the history of voting methods and machines.

History of Voting Machines

Clearly, the earliest form of tallying the votes of a population was by the voter making a mark on a simple paper ballot, and then placing that ballot into a box. The first official, uniform paper ballots were employed in the Australian state of Victoria in 1856. After the adoption of this system throughout Australia, paper balloting systems simply became known as the "Australian ballot" (Bellis). While a uniform paper ballot is an inexpensive, effective and verifiable method of recording votes, it is flawed. It is a simple matter of "stuffing" the ballot box with fraudulent copies of the official ballots to invalidate the results of an election. Another major problem with this elementary method is the sheer amount of time required election officials to hand-tally the votes.

One of the first mechanical systems employed in elections was the mechanical lever machine. The first official lever machine was the "Myers Automatic Booth," originally used in Lockport, New York, in 1892. This machine had several decided advantages over the traditional paper balloting systems. The mechanical nature of the system allowed for officials to place some very rudimentary security checks on the election, and the tedious challenge of hand-counting thousands of ballots was a thing of the past (Bellis).

The next iteration in voting technology is particularly infamous. Punchcards were first employed in Fulton and De Kalb counties in Gerogia in 1964. When using a punchcard ballot, a voter would remove a small, pre-cut, paper dot that lined up with the desired candidate or issue. The completed ballot then ran through a computer, tallying the votes. While this system was a marked improvement over previous systems, it was still hopelessly flawed. In the 2000 United States presidential election, the issue of the "hanging chad" came into the national forefront. A "hanging chad" is a term coined to describe a mark on the punchcard ballot where the voter did not completely remove the pre cut paper circle. This lead to a great deal of controversy regarding what qualified as "voter intent" (Jackson). Another issue that came to light was ballot complexity and design. In Palm Beach County, Florida, a two-paged, "butterfly" ballot was designed to accommodate the larger font size needed by many of the elderly residents. This design lead to unneeded complexity in the ballot and confusion on Election Day (Bisantz).

The many issues with punchcard ballots in the 2000 United States presidential election prompted the U.S. government to pass the 3.8 billon dollar Help America Vote Act. This act provided that " … a substantial portion of these funds allocated to US states and territories for the purpose of replacing their punch card and lever voting machines and making voting systems accessible to the disabled. " (Mercuri). The goal is to eliminate those machines by 2006 (Sieberg). These old machines were often replaced with Direct Recording Electronic or DRE voting machines.

DRE Machines

Given the bad reputation punchcard voting systems garnered in the 2000 United States presidential election, it was a matter of time before a new standard began to emerge. The DRE systems that have sprung up across the United States are generally completely paperless systems. The voter goes to their polling place, usually presents some form of identification, and then receives some sort of authentication mechanism (a smartcard, a PIN, etc…). The voter then proceeds to a terminal where they authenticate themselves using their PIN or token, and record their vote onto the computer. After the voter makes their choice, the machine usually offers up one final opportunity to make changes to the selections, and then the ballot is officially "cast" (Kohno et al. 3).

While these machines seem to be an excellent solution to the problems posed by traditional voting machines, the DRE machines bring about a whole new set of problems and issues. While many of the problems that plague a traditional voting system are present with DRE systems, the use of complex, proprietary software within the voting terminal arises as the greatest possible area for problems. The issue of software in DRE machines is so fundamental, that Kohno, Stubblefield, Rubin, and Wallach say, "the entire election hinges on the correctness, robustness, and security of the software within the voting terminal" (Kohno et al. 3).

An examination of a DRE system

Since the DRE software is fundamental to the success of an election where it is employed, it is prudent to critically examine some of this software before it is deployed in elections. However, most of the DRE machines used in prior elections have not had their functionality critically examined. It is interesting to note that a similar lack of a critical, public evaluation has also been a chief complaint about mechanical lever voting machines for ages (Jones). Some manufacturers of DRE software have stood on the ground of "security through obscurity" even though the security community has always held that obscurity is never adequate in providing meaningful security (Kohno et al. 4). There has been one case where a DRE system was critically examined by a group of knowledgeable individuals, and the results were less than flattering.

Tadayoshi Kohno, Adam Stubblefield, Aviel Rubin, and Dan Wallach took the source code for a popular DRE system, Diebold's AccuVote-TS, and put it to

the test. The examination covered all aspects of the machine's functionality. They examined how the system is set up, how it is operated, and how the votes are recorded. They took a long look at the engineering of the software, and spent significant time examining the AccuVote's use of cryptography (Kohno et al. 1-21).

In order to understand the issues that Kohno, Stubblefield, Rubin, and Wallach encountered, it is important to understand how the system is deployed. Before the election takes place, the election officials must first create the ballot for the election by creating a standard file called a "ballot definition" (Kohno et al. 7). Once this file is in place, and the machines are physically set up, the election can begin. To vote, the voter approaches the terminal with his or her "voter card" (i.e. smartcard) given to him or her by election officials. The voter then inserts the card into the terminal, interacts with the terminal, and casts his/her ballot. After the ballot is cast, the card is "cancelled" as to prevent the voter from casting multiple ballots. The voter removes his/her smartcard, and the machine is ready for the next voter (Kohno et al. 7).

At the end of the Election Day, the officials insert a special smartcard (called an "administrator" or "ender" card) to terminate the voting. The insertion of this card prompts the official for a PIN, and, upon the entry of this PIN, the election can be terminated. The results of the election are then collected at a central back-end server by either a flash memory card, or they can even be transmitted electronically over a network (Kohno et al. 7).

Security design flaws in a DRE system

One of the system's supposed security features, the use of smartcards, generated the most serious security flaw that Kohno, Stubblefield, Rubin, and Wallach found. The key to the issue is that smartcard does not provide any sort of secure authentication to the voting terminal (Kohno et al. 10). There is nothing in place to prevent a malicious user from creating his/her own homemade smartcard and wreaking havoc on the system. Once someone has reverse engineered and appropriately programmed a homemade smartcard, he/she could proceed with several methods of attack. The homemade smartcard could even be programmed to allow the user to cast multiple votes (Kohno et al. 10).

An adversary could also home-brew his/her own version of one of the administrator or ender cards for a system. This scenario allows the user complete access to all of the administrative functions of the system. Given a willing adversary, the entire election could be shut down (Kohno et al. 10). One additional layer of security exists with the administrator's card. Recall, upon insertion of an administrator card, the user is prompted for a PIN. This is an important layer of security, but the PIN is transmitted in clear-text between the smartcard and the voting terminal. This greatly reduces the efficacy of the PIN by allowing any adversary with the ability to reverse-engineer the protocol between the smartcard and the terminal an avenue to easily snoop the PIN (Kohno et al. 11).

It is also important that the terminals provide integrity and confidentiality in the storage of critical election data (i.e. votes, configurations, etc…). It is

conceivable that an election insider could tamper with this data via physical access to the voting system, or a knowledgeable outsider could mount a man-in-the-middle style attack if the terminals are networked (Kohno et al. 11). This problem is compounded by the fact that all of the configuration information is stored in the clear, in a Windows registry, without any form of integrity protection (Kohno et al. 12). An adversary must merely modify this unprotected system registry to change the machine's configuration.

The records of the actual votes and the audit logs for the system are both encrypted and checksummed, but the techniques used to secure these files do not use established security practices (Kohno et al. 14). The manufacturers of the AccuVote-TS system eschewed the significantly stronger pseudo-random, two key triple DES (RSA 3.2.6). Instead, Diebold elected to employ a single, hard-coded single DES key (Kohno et al. 14). In fact, Kohno, Stubblefield, Rubin, and Wallach examined the history of the AccueVote-TS source code and deduced that the hard-coded key had been used without change since December 1998, and they feel that it is safe to assume that the key was in use well before that date (Kohno et al. 14-15).

Another flaw found in the system is a clear violation of Shamos's first commandment (or the confidentiality portion of the three CIA principals). Every time a voter casts a vote on the AccuVote-TS system, the vote is written sequentially to a file on the machine that records the vote (Kohno et al. 16). The sequential nature of file storage employed by the TS system provides an adversary a simple mechanism for linking voters with the vote that they cast based on the order in which the votes are cast. The voter's identity is not stored along with his/her votes on the machine, but a generated serial number is assigned to each voting record. This would be an effective method of preventing an adversary from associating a given vote with a voter if a cryptographically secure pseudo-random number generator had been chosen. However, the AccuVote-TS uses the following hard-coded linear congruential generator to assign the "random" number to each vote (Kohno et al. 16):

```
//LCG–Linear Congruential Generator–used to generate ballot
serial numbers
// A psuedo-random-sequence generator
// (per Applied Cryptography, by Bruce Schneier, Wiley, 1996)
#define LCG_MULTIPLIER 1366
#define LCG_INCREMENTOR 150889
#define LCG_PERIOD 714025

static inline int lcgGenerator(int lastSN)
{
return ::mod(((lastSN*LCG_MULTIPLIER)+LCG_INCREMENTOR),
LCG_PERIOD);
}
```

While none of these problems alone are enough to bring down an entire election, it is the sheer number of major problems that affect the AccuVote-TS and call its overall security into question. While it is clear that the AccuVote is

inherently flawed from a design perspective (poor use of cryptography, over-reliance on the security of smartcards), we will now examine the system from the perspective of a software engineer.

Software engineering in the Diebold System

Kohno, Stubblefield, Rubin, and Wallach were fortunate as they were able to examine the code from the AccuVote-TS from a broad span of time. This allowed them to see the software development and configuration management process "in action." One of the first things they noticed was that, while the software was written in C++, there was careful attention paid to avoiding common, destructive errors like buffer overflows (Kohno et al. 18). Kohno, Stubblefield, Rubin, and Wallach make numerous remarks regarding the coding style, lack of comments, and lack of clear change control. In response to these comments, Diebold, "developed, documented, and implemented a change control process" and published this fact in Risk Assessment Report: Diebold AccuVote-TS Voting System and Processes.

General Remarks regarding the AccuVote-TS.

Throughout the short history of the computing age software vendors of all sorts have attempted to convince the public that keeping their software in a closed environment makes it more secure. Kohno, Stubblefield, Rubin, and Wallach are convinced that the closed nature of the AccuVote-TS is exactly what causes it to be full of problems. All one needs to do is look at the success of the development of the Advanced Encryption Standard to see the success of open, public software development (Kohno et al. 21). This is not to say that using open source software is the ultimate solution to the problems that plague DRE machines, but making the software that helps maintain a democratic system open to knowledgeable, public scrutiny seems fair and reasonable.

Documented Problems with DRE systems

Clearly bolstered by the 2002 Help America Vote Act, an unprecedented number of paperless ballots were cast in the 2004 United States presidential election. Verifiedvoting.org contends that thirty percent of voters used paperless systems in 2004, up from twelve percent in 2000. While the United States did not have a repeat of the near-meltdown of 2000, verifiedvoting.org's Nationwide Election Incident Reporting Center has reported 38,097 separate election incidents as of December 2004. While not all of these are related to the use of DRE systems, at least nine hundred incidents have been reported to verfiedvoting.org.

One of the more widely published errors that appeared in the 2004 United States presidential election occurred in the ever-important "swing" state of Ohio. On November 5th CNN reported an error in the electronic voting system in Franklin County, Ohio gave President George W. Bush an additional 3,893 votes in that county. In fact, Bush only received 365 of the county's 638 cast votes. The election officials in Franklin County could not pinpoint the exact problem

with the machine in question.

While this error certainly was not serious enough to affect the outcome of the 2004 election (Bush won Ohio's electoral votes by more than 136,000 votes), it is a clear example of what sorts of problems have been encountered.

Proposed Solutions

Increasing voting machine reliability and security has been a prominent topic since the 2000 presidential election. A number of people have weighed in, offering their opinions as to what could be done to ensure that the principles of confidentiality, integrity, and availability are all covered. Ideas that seem to have broad-based support are the addition of a voter-verifiable paper trail to DRE machines and opening the DRE software up to public scrutiny.

Voter-verifiable paper trail

Renowned computer security expert Bruce Schneier proposed the following method in an article that he penned for his weblog and posted on November 10, 2004;

> DRE machines must have a voter-verifiable paper audit trail (sometimes called a voter-verified paper ballot). This is a paper ballot printed out by the voting machine, which the voter is allowed to look at and verify. He doesn't take it home with him. Either he looks at it on the machine behind a glass screen, or he takes the paper and puts it into a ballot box. The point of this is twofold. One, it allows the voter to confirm that his vote was recorded in the manner he intended. And two, it provides the mechanism for a recount if there are problems with the machine.

Now that the voter can see that his/her ballot is cast properly, the need for blind faith in the DRE machine is eliminated, and the integrity of the ballot, and thereby the integrity of the will of the voter, is assured.

One thing that is important to note about Schneier's proposed method is that the voter is not able to remove the paper ballot from the voting machine. This is extremely important. If a voter was able to take a "receipt" home from the polling place, this would not only jeopardize the principle of confidentiality, it would also provide a mechanism by which votes could be bought. A candidate could ask a voter to cast his/her vote for the candidate in exchange for money, and then using his/her "receipt," the voter could prove to the candidate that he/she did indeed cast the desired vote.

DRE software

Another idea gaining support amongst the security industry is allowing open, complete, public scrutiny of DRE software. As mentioned before, the software companies who develop DRE software have hidden behind the fundamentally flawed idea of "security through obscurity." This has proven to be problematic. We have seen clear, documented problems with DRE machines, and it only seems fair that the public who relies on the security of these

machines should have the opportunity to examine them in depth. In his November 10, 2004 weblog entry Bruce Schneier points out that allowing the public to examine DRE software accomplishes two important things. First and most obviously, it will allow knowledgeable people to point out bugs and flaws in the code, which can then be corrected. The computer security industry is rapidly evolving, with new threats appearing on a daily basis. The only way that DRE software can ever come close to being secure is to allow the entire security community access to the code.

The second objective that public scrutiny accomplishes is that it will undoubtedly increase the public's confidence in DRE technology. DRE machines are one major, public meltdown away from going the way of the dinosaur. Public confidence in these machines is imperative if officials are going to continue to employ them. Schneier makes the point clear by pointing out "If the software is public, no one can insinuate that the voting system has unfairness built into the code."

The companies that manufacture the DRE code claim that making the code openly available would allow threats to be more easily developed. The bottom line is: threats will be developed with or without the source code being publicly available, so why not let the public help with the massive job of keeping the voting infrastructure secure?

Defense in Depth

While both of these ideas would make DRE machines much more secure than they are now, the careful application of the defense-in-depth principles to DRE machines is the ultimate solution to the problems that plague these computers.

All new computer security professionals have the concept of defense-in-depth pounded into their heads. They are told to believe that each single security measure they will employ will fail. Layers of redundant security are essential. It is clear that the manufacturers of the DRE machines have neglected to embrace this fundamental concept. Douglas W. Jones states that currently "…there is little evidence of effective defense in depth in current voting systems." Thoughtful application of simple, effective, redundant security measures would greatly increase the likelihood that DRE machines will function properly and fairly, thereby ensuring the security and accuracy of the election process.

Conclusion

The struggle with voting machine security is not a new problem. The United States has been dealing with these issues since the days of the old mechanical lever machines. However, it is of the utmost importance that we take these issues seriously. It might sound overly dramatic, but the very essence of democracy is in question here. If the voting public cannot be one hundred percent confident their votes are being counted faithfully and accurately

on Election Day, the very heart of the United States' electoral system is in jeopardy.

There are several things that can be done to keep the electoral system free from problems, and none of these ideas involve throwing DRE machines out the window. We begin by allowing the voter to physically validate the vote that was cast on the DRE machine. Then we open up the software to public scrutiny. Finally, we make sure that time tested, effective security measures are applied to each and every machine used to cast a ballot. These methods might not create the perfect voting machine, but the application of these ideas would certainly be a great leap forward from where we are now.

Works Cited

Bellis, Mary. "The History of Voting Machines." November 1998. About.com. 11 November 2004. <http://inventors.about.com/library/weekly/aa111300b.htm >.

Bisantz, Ann M. "Election 2000: A Case Study in Human Factors and Design." National Center for Case Study Teaching in Science. 10 December 2004. <http://www.sciencecases.org/election/election.asp>.

Doherty, Will. "Statement of The Election Verification Program." VerifiedVoting.org. 18 November 2004. Verified Voting Foundation, Inc. 29 November 2004. <http://www.verifiedvoting.org/article.php?id=5302>.

"Glitch gave Bush extra votes in Ohio." CNN.com. 5 November 2005. Cable News Network. 15 November 2004. <http://www.wanttoknow.info/041105cnn.orig>.

Jackson, Brooks. "'Hanging chads' often viewed by courts as sign of voter intent." CNN.com. 16 November 2000. Cable News Network. 30 November 2004. <http://archives.cnn.com/2000/ALLPOLITICS/stories/11/16/recount.chads/index.html>.

Jones, Douglas W. "Voting System Transparency and Security: The need for standard models." 20 September 2004. 30 November 2004. <http://www.cs.uiowa.edu/~jones/voting/nist2004.shtml>.

Kohno, Tadayoshi, et al. "Analysis of Electronic Voting Systems." IEEE Symposium on Security and Privacy. May 2004. IEEE Computer Society Press. 1 November 2004. <http://avirubin.com/vote.pdf>.

Mercuri, Rebecca. "Electronic Voting." 2004. 15 November 2004. <http://www.notablesoftware.com/evote.html>.

Schneier, Bruce. "The Problem with Electronic Voting Machines." 10 November 2004. 30 November 2004. <http://www.schneier.com/blog/archives/2004/11/the_problem_wit.html>.

Shamos, Michael Ian. "CFP '93 – Electronic Voting – Evaluating The Threat." 1993. 10 November 2004. <http://euro.ecom.cmu.edu/people/faculty/mshamos/CFP93.htm>.

Sieberg, Daniel. "Jury's still out on E-voting." CNN.com. 5 November 2004. Cable News Network. 11 November 2004. <http://www.cnn.com/2004/ALLPOLITICS/11/05/evoting.evaluation/index.html>.

"What is Triple-DES?" RSA Security.com. 2004. RSA Security. 5 December 2004. <http://www.rsasecurity.com/rsalabs/node.asp?id=2231>.