SM

c/o 3548 Beechwood Boulevard
Pittsburgh, Pennsylvania 15217-2767
412-421-0178

VERIFIED PA VOTING
SM

c/o 1235 Malvern Ave.
Pittsburgh, Pennsylvania 15217
412-400-3773
PA.VerifiedVoting@gmail.com

Report on Allegheny County iVotronic Firmware Verification, December 22, 2008

Prepared by VoteAllegheny and PA Verified Voting

Revision 1.3 of March 22, 2009

Report authors:

David A. Eckhardt, Ph.D., representing PA Verified Voting

Kami Vaniea, representing VoteAllegheny

Background

On Monday, December 22, 2008, SysTest Labs, a software quality assurance and test engineering company, acting as a contractor for the Allegheny County Elections Division, conducted a procedure to investigate the authenticity of the software installed on the County's iVotronic voting terminals. Three election integrity advocates, representing the League of Women Voters of Greater Pittsburgh, VoteAllegheny, and PA Verified Voting, attended the event.

Motivation

Our iVotronic voting machines are "software dependent," which means that the results they report are right or wrong depending on what software they are running, because there is no software-independent way for voters to personally verify that their votes have been correctly and accurately recorded. This means that voters must trust the iVotronic program code to be correct.

Though the Secretary of the Commonwealth inspected the iVotronic software before certifying the election system for use in Pennsylvania, this does not ensure that each individual iVotronic terminal contains the software certified by the Secretary. In other states elections have used voting equipment running uncertified software, and, indeed, after the May 16, 2006 primary election in Allegheny County VoteAllegheny found that a PEB cartridge contained software not certified in Pennsylvania. In addition, a team of computer security researchers commissioned by the Secretary of State of Ohio has determined[1] that the security of the iVotronic system is weak enough that motivated attackers could replace legitimate iVotronic software with uncertified software, including software with malicious intent.

Voters are entitled to a system which credibly records and counts their votes as cast. Because computer scientists cannot presently prove correct software as large and complex as the iVotronic firmware, the "first line of defense" should be a software-independent cross-check on each vote. As long as this is not available and Allegheny County voters must trust their votes to a software-dependent system, we should have some reason to believe that our voting equipment is running the software (correct or incorrect) which it is designed to run: the inspection and certification process carried out by the Secretary of the Commonwealth is meaningless unless the County ensures the machines run exactly the same program the Secretary certifies.

Verification of a sample

A software verification process would ideally check the integrity of every machine[2] before each election; after the check, strong security measures (*e.g.*, 24-hour video surveillance) should be employed to ensure that tampering does not occur before custody of the voting machines is turned over to each polling place's election workers. A complete inventory of the software on all County voting machines is infeasible, however. Because the iVotronic terminals were not designed with verification in mind, checking a single machine requires substantial equipment and time. More importantly, opening an iVotronic and removing the modules containing its software voids both the warranty on the machine and the Secretary's certification of it. This limits the scope of a software-verification effort: in order for the County to retain enough operational machines to run elections, the integrity of only a small subset of the machines can be verified at any time.

1 "EVEREST: Evaluation and Validation of Election-Related Equipment, Standards, and Testing," December 7, 2007, sections 6.3.1 and 7.2.2.

2 "Report on Election Auditing," League of Women Voters of the United States, January 2009, guideline B(2), page 6.

Machine Selection

Gregory Pollich, Director of State Compliance for SysTest Labs, suggested that selection of the machines be made by the voting-integrity observers present. Kami Vaniea, representing VoteAllegheny, and David A. Eckhardt, representing PA Verified Voting, agreed to formulate a procedure.

Before arriving at the warehouse, the initial thinking had been to use flipped coins to do a binary search on a list of machines sorted by serial number, and to pull the machines thus indicated. But it turned out that the machines were scattered throughout the warehouse in a way that bore no relation to their serial numbers; in fact, there was no mapping available from serial numbers to locations.

Based on a walk-through of the warehouse, it appeared that the machines were organized into roughly five rectangular blocks, e.g., a block of machines 39 wide and 26 deep, one 35 wide and 18 deep, etc. The width and depth of each block were multiplied to obtain its size, and these products were added to get the total number of machines in the warehouse. Eighteen random numbers in the range from zero to one less than the number of machines, inclusive, were drawn using the random.org online random-number service. Each random number was assigned to the appropriate block based on the list of block sizes. Next, the base number of the block (the number of the first machine in that block) was subtracted from the random number to yield the block-relative machine number; this was then divided by the depth of the block to compute the row, and the remainder was used to indicate which machine in the row was selected.

For each block an arbitrary machine was chosen to be (row=0, column=0); when there was a center aisle between two blocks of machines, counting proceeded outward from the aisle (i.e., toward the left for the left block and toward the right for the right block).

This procedure was imperfect because several of the blocks didn't contain exactly the number of machines that that had been calculated based on the quick walk-through of the warehouse. Some blocks had an extra partial row; machines there were overlooked by the procedure. Due to the placement of roof support columns, some rows were missing a small number of machines; in order to obtain, for example, machine number 45 from a row which turned out to contain 40 machines, machine number 5 was used.

Another bias was introduced at the request of Elections Director Mark Wolosik: the number of "ADA" machines (machines with limited support for use by blind voters) was artificially restricted to two, in order to maintain a sufficiently large population of those machines.

Finally, because the procedure was executed as fast as it could be designed, using a manual calculator, calculation errors were possible. For example, two extra machines were selected from one of the blocks and thus two machines too few from another.

Overall, however, the selection was made without human bias; "clumping" (selection of physically close machines) was observed which seemed consistent with genuine randomness.

The table below lists the machines which were selected, in terms of where they had been deployed in the November, 2008 general election. Note that in some communities polling places are designated by ward and district, but in other communities the only specification is a district.

Machine location	Serial number
Bethel Park W3 D3	5185607
Bethel Park W5 D1	v5179741
Bethel Park W7 D1	v5178557

Machine location	Serial number
Castle Shannon D7	v5183261
Clairton W2 D3	v5180415
Duquesne W2 D1	v5184193
Hampton D11	v5180468
Homestead W2 D1	v5180608
Indiana D3	v5178091
North Braddock W1 D3	v5186813
O'Hara W2 D2	v5186636
Penn Hills W1 D1	v5172221 (“ADA”)
Pittsburgh W4 D7	v5181797
Penn Hills W8 D5	v5183252
Pittsburgh W14 D30	v5182277
Pittsburgh W29 D5	v5180802
Plum D17	v5178307
South Park D9	v5185578
Verona D3	v5182539
Wilkinsburg W1 D6	v5173239 (“ADA”)

Firmware Verification Procedure

The firmware verification relied on three pieces of equipment: a County-maintained Windows desktop machine, normally used by Voting Machine Custodian John O'Brien, a Logical Devices ChipMaster 6000XPu EEPROM programmer, and an adapter jig manufactured by SysTest which enables iVotronic memory modules to be plugged into the EEPROM programmer.

The following steps were taken for each iVotronic terminal:

1. A PEB was inserted to activate the unit and verify that it displayed the expected firmware revision number, 9.1.4.1.
2. The iVotronic was powered off and the PEB was removed.
3. The iVotronic's battery was disconnected and removed.
4. Screws were removed (breaking the factory seal), and the rear cover of the iVotronic was removed.

5. Memory module "U1" was removed from the motherboard by slicing through an epoxy sealant and prying the module loose. The manufacturer, lot number, and serial number of the memory module were recorded.
6. The U1 module was inserted into the EEPROM reader, which was then powered on.
7. Software provided with the EEPROM reader was used to load a specific part of the module (starting from a specified base address and continuing for a specified number of bytes) and store the result in a file on the PC.
8. A SHA-1 hash utility was used to compute the hash of the firmware image stored on the PC; this hash fingerprint was compared to the hashes of all previous firmware images. The hash fingerprint of the image extracted from the first iVotronic was compared to a reference value read verbally by Mr. Pollich of SysTest.
9. The EEPROM reader was powered off and the memory module was removed.
10. The U1 memory module was reinserted into its socket on the iVotronic motherboard.
11. The rear cover and battery of the iVotronic were reinstalled.
12. Once again a PEB was used to activate the iVotronic to check that it was in working order and reported the expected firmware revision.

This process took quite some time for the first few machines, but eventually settled down to requiring approximately 10 minutes per unit. Initially the procedure was carried out by SysTest personnel, but eventually the responsibility shifted to County employees.

Analysis

The SHA-1 hash fingerprints of the memory modules for the 20 randomly-selected machines were observed to match each other and the fingerprint provided by Mr. Pollich. This suggests to some degree that the population of all Allegheny County iVotronics was running certified firmware during the November general election. This section of the document will consider various factors which influence the degree of assurance provided by the verification procedure observed on December 22. These factors are broken down into two categories. One category is issues affecting the degree to which all relevant software was examined; the other consists of issues potentially affecting the accuracy of the results.

Degree of software coverage

Probably the salient weakness of the current process is that does not check the integrity of all software which influences the critical path of vote selection and storage.

1. The current process does not examine the software running on the PEB cartridges. Based on the EVEREST report, we know that a malicious PEB (or, indeed, any device capable of emulating the PEB protocol, such as a properly programmed cellular phone) can compromise the integrity of iVotronics. The current process also does not verify the integrity of the firmware on the M650 bulk ballot scanners used to scan absentee, provisional, and emergency ballots, or the integrity of the software on the machines which together form the Unity tabulation system.
2. The current process does not examine the integrity of all software running on the iVotronic terminals. In particular, each iVotronic contains at least one other memory module containing

executable code, U2, described as containing the iVotronic "proprietary operating system." It is very likely that malicious code located on U2 could compromise the integrity of vote recording.

3. Not all of U1 is examined. To an extent this makes sense because some parts of the memory module may differ from machine to machine (e.g., machine serial number, touch-screen calibration data). However, the current procedure is also designed to skip "unused" areas of the memory, which is unfortunate: malicious software could find un-monitored non-volatile storage useful, so unused memory should be initialized to a known pattern, which should be checked.[3]
4. The sample size used (20 machines) is smaller than required to provide a statistically significant assurance.[4] Even combining the December 22 sample with the sample of machines examined in private by the County before the election probably results in a number of machines which is too small by a factor of three or more.
5. Because malicious software could erase itself after an election, sampling before an election is probably more likely to detect tampering than sampling afterward. That said, both is probably better than either one alone.

Noteworthy Dependencies

The verification process, as carried out on December 22, assumed that a variety of technological tools were what they appeared to be. A sufficiently motivated attacker might be able to violate one of these assumptions, interfere with the measurement process, and thus conceal the presence of uncertified code on iVotronics. While some of these attacks may seem far-fetched, they are mentioned both to illustrate the difficulty of conclusive demonstrations of software integrity and because, if carried out, they would undermine extensive effort invested by the Elections Division.

6. The verification process depended on a "stack" or "tool chain" starting with the EEPROM programmer and continuing up through the USB device driver software on the PC, the entire Windows software base (potentially compromised by undetected illicit software), the software application which drove the EEPROM programmer, and the SHA-1 hash program provided by SysTest. Historically, authors of malicious software have been able to conceal code of substantial complexity in environments of this size.
7. The EEPROM programmer was ostensibly used to read data from the iVotronic memory modules. However, because the device is also capable of *writing* to the modules, it would be possible to craft malicious software on the PC which could conceal the presence of incorrect software on the iVotronic memory module by overwriting it with legitimate software before reading it back.
8. The process depended on the correct operation of the adapter jig, since the pin spacing on the iVotronic memory modules is not compatible with the socket on the EEPROM reader. The jig is physically large enough to contain enough logic to conceal illicit software.

3 "SWATT: SoftWare-based ATTestation for Embedded Devices," Seshadri et al., 2004 (see "Empty memory regions").
4 "iVotronic Software Verification Protocol: Allegheny County Proposals," VoteAllegheny, September 28, 2008.

Recommendations

1. The most urgent issue is that the current procedure checks the integrity of only a fraction of the executable code used during voting. The two verification sessions to date have not yet examined all executable code in even one iVotronic terminal. Scanning must be expanded to cover all of the iVotronic code base and the PEB code base as well.

2. The sample-size issue should be addressed. A productive step would be to engage the services of a statistician to clearly express the implications of particular sample sizes on the degree of assurance provided by the process. This analysis should take place and be published before the sample size is selected for the next verification session.

3. If a machine-selection procedure similar to the one used on December 22 is used going forward it would help if the iVotronics were arranged in perfectly rectangular blocks. In any case, as with sample size, the selection procedure would probably benefit from being formalized in advance by a statistician.

4. It would be prudent for scanning of memory modules to be performed by a device incapable of writing to the memory. It would also be desirable for the scanning platform to be small enough to be itself scanned for integrity.

5. A fundamental problem is that the iVotronic equipment was not designed for verification of its integrity, nor was a verification plan developed before the equipment was deployed. Verification would be easier if each memory device were either wholly constant across units (*e.g.*, boot loader, OS, application software) or wholly unit-dependent (serial number, log information). Verification planning would be eased if a memory map were published for each device. If opening a device to verify the integrity of its software forces removal from service of that device, then re-certification of opened devices should be a standard service offered by the equipment vendor, with pricing quoted and figured into operation costs. In the long term, it may make more sense to replace the iVotronics with equipment designed from the beginning to be verified.

Observations

If examining a machine to verify the integrity of its software requires removing it from service pending re-certification, this provides an opportunity for the examination of some machines to be carried out by citizen observers. The use of an independent procedure based on tools developed by The Election Transparency Project in Humboldt County, California, recently discovered that a voting system vendor's tabulation system systematically skipped certain votes.[5] In general, opening up the examination of a system to multiple parties increases the likelihood that errors in the system will be detected.

At least in the case of machines for which verification was not a design goal, there is a fundamental tension between the desire to increase trust in the system by verifying the integrity of more machines and the need to preserve a large fraction of the equipment pool in a certified condition. This tension does not exist for all types of voting system. In particular, it is possible to verify the end-to-end integrity of a software-independent system based on paper ballots by re-counting the ballots – because such a system does not depend so utterly on the integrity of its software, it is possible to check the *operation* of the software "from the outside," without such a strong need to open machines to check the *contents* of the software.

5 "Report to the Election Assistance Commission Concerning Errors and Deficiencies in Diebold/Premier GEMS Version 1.18.19," Secretary of State of California, March 2, 2009 (see "How the Deletion of Votes Was Detected").

Moving Forward

Since 2006, Allegheny County election integrity activists have called for verification of the software which forms the foundation of our election system.[6,7] The efforts put forth by the Allegheny County Elections Division as part of the November 2008 general election were substantial and ground-breaking. However, to date a *complete* verification of the software in even one iVotronic has not taken place; other software on the critical path has not been examined at all; and the sampling process is not yet based on statistical significance. Furthermore, the work done here is equally urgent for all counties in the Commonwealth which will be deploying software-dependent voting equipment in future elections. The Secretary of the Commonwealth should develop a verification plan for each voting system as part of the certification process, so that the software in voting systems can be verified as counties purchase and deploy them, instead of after the fact.

Allegheny County should prepare a document explaining election procedures, threats, and defenses. This document should include treatment of software verification, parallel testing, and post-election audits. Public comment should be sought and incorporated into the plan. Such a document could serve as the model for similar planning across the Commonwealth.

Conclusion

In a healthy democracy the legitimacy of the government depends on the consent of the governed. This consent requires not only the pomp and circumstance of an election process but also the widespread and strong confidence that the process is accessible and accurate. When the lack of a voter-verifiable paper ballot means that voting-system integrity hangs by the thread of software correctness, voter confidence demands the most comprehensive possible software certification process coupled with overwhelming certainty that the software deployed on voting systems exactly matches the software which was certified.

VoteAllegheny and PA Verified Voting commend Allegheny County officials and the Secretary of the Commonwealth for the recent improvement in election system integrity represented by the December 22, 2008 partial examination of iVotronic firmware, and hope that this will be a stepping-stone toward voter verifiability and increased election system integrity.

6 "VotePA-Allegheny Report on Irregularities in the May 16th Primary Election," VotePA, June 5, 2006.
7 "VoteAllegheny Report on ES&S M650 Logic & Accuracy Testing," VoteAllegheny, October 31, 2006.