Security Analysis of the Diebold AccuBasic Interpreter

David Wagner David Jefferson Matt Bishop
Voting Systems Technology Assessment Advisory Board (VSTAAB)

with the assistance of:

Chris Karlof Naveen Sastry
University of California, Berkeley

February 14, 2006

1 Summary

This report summarizes the results of our review of some of the source code for the Diebold AV-OS optical scan (version 1.96.6) and the Diebold AV-TSx touchscreen (version 4.6.4) voting machines. The study was prompted by two issues: (1) the fact that AccuBasic scripts associated with the AV-OS and AV-TSx had not been subjected to thorough testing and review by the Independent Testing Authorities when they reviewed the rest of the code for those systems, and (2) concern over vulnerabilities demonstrated in the AV-OS optical scan system by Finnish investigator Harri Hursti in Leon County, FL. Mr. Hursti showed that it is possible for someone with access to a removable memory card used with the AV-OS system to modify scripts (small programs written in Diebold's proprietary AccuBasic language) that are stored on the card, and also to modify the vote counts stored on the card, in such a way that the tampering would affect the outcome of the election and not be detected by the subsequent canvass procedures.

The questions we addressed are these:

- What kinds of damage can a malicious person do to undermine an election if he can arbitrarily modify the contents of a memory card?
- How can the possibility of such attacks be neutralized or ameliorated?

The scope of our investigation was basically limited to the above questions. We did not do a comprehensive code review of the whole codebase, nor look at a very broad range of potential security issues. Instead, we concentrated attention to the AccuBasic scripting language, its compiler, its interpreter, and other code related to potential security vulnerabilities associated with the memory cards.

We found a number of security vulnerabilities, detailed below. Although the vulnerabilities are serious, they are all easily fixable. Moreover, until the bugs are fixed, the risks can be mitigated through appropriate use procedures. Therefore, we believe the problems as a whole are manageable.

Our findings regarding the scope of possible attacks on the AV-OS optical scan and AV-TSx touchscreen systems can be summarized as follows:

- *AccuBasic is a limited language:* The AccuBasic language itself is not a powerful programming language, but a very restricted one, narrowly tailored to one task: calculating and printing reports before and after an election. From a security point of view this is very desirable; minimal functionality generally means fewer opportunities for error or security vulnerability. In particular, *when its interpreter is properly implemented* (see below) an AccuBasic program cannot modify votes or ballot images; it can read vote counters (AV-OS) or ballot images (AV-TSx), but it cannot modify them.

- *The AccuBasic interpreter is well-structured:* The code in the AccuBasic interpreters for both machines is clean, well-structured, and internally documented. We were able to understand it with little difficulty despite the lack of external documentation.

- *Memory card attacks are a real threat:* We determined that anyone who has access to a memory card of the AV-OS, and can tamper it (i.e. modify its contents), and can have the modified cards used in a voting machine during election, can indeed modify the election results from that machine in a number of ways. The fact that the the results are incorrect cannot be detected except by a recount of the original paper ballots.

- *Harri Hursti's attack does work:* Mr. Hursti's attack on the AV-OS is definitely real. He was indeed able to change the election results by doing nothing more than modifying the contents of a memory card. He needed no passwords, no cryptographic keys, and no access to any other part of the voting system, including the GEMS election management server.

- *Interpreter bugs lead to another, more dangerous family of vulnerabilities:* However, there is another category of more serious vulnerabilities we discovered that go well beyond what Mr. Hursti demonstrated, and yet require no more access to the voting system than he had. These vulnerabilities are consequences of bugs—16 in all—in the implementation of the AccuBasic interpreter for the AV-OS. These bugs would have no effect at all in the absence of deliberate tampering, and would not be discovered by any amount of functionality testing; but they could allow an attacker to completely control the behavior of the AV-OS. An attacker could change vote totals, modify reports, change the names of candidates, change the races being voted on, or insert his own code into the running firmware of the machine.

- *Successful attacks can only be detected by examining the paper ballots:* There would be no way to know that any of these attacks occurred; the canvass procedure would not detect any anomalies, and would just produce incorrect results. The only way to detect and correct the problem would be by recount of the original paper ballots, e.g. during the 1 percent manual recount.

- *The bugs are classic, and can only be found by source code review:* Finding these bugs was only possible through close study of the source code. All of them are classic security flaws, including buffer overruns, array bounds violations, double-free errors, format string vulnerabilities, and several others. There may, of course, be additional bugs, or kinds of bugs, that we did not find.

- *AV-TSx has potential cryptographic protection against memory card attacks:* A majority of the bugs in the AV-OS AccuBasic interpreter are also present in the interpreter for the AV-TSx touchscreen system. However, the AV-TSx touchscreen has an important protection that the AV-OS optical scan does not: the key contents of its removable memory card, including the AccuBasic scripts, are digitally signed. Hence, if the cryptographic keys are managed properly (see next bullet), any tampering would be quickly detected and the attack would be unsuccessful. All of the attacks we describe, and Hursti's attack as well, would be foiled, because the memory card by itself would in effect be cryptographically tamperproof.

- *But the implementation of cryptographic protection is flawed:* There is a serious flaw in the key management of the crypto code that otherwise should protect the AV-TSx from memory card attacks. Unless election officials avail themselves of the option to create new cryptographic keys, the AV-TSx uses a default key. This key is hard-coded into the source code for the AV-TSx, which is poor security practice because, among other things, it means the same key is used in every such machine in the U.S. Worse, the particular default key in question was openly published two and a half years ago in a famous research paper, and is now known by anyone who follows election security, and can be found through Google. The result is that in any jurisdiction that uses the default keys rather than creating new ones, the digital signatures provide no protection at all.

- *All the bugs are easy to fix:* In spite of the fact that the bugs we have identified are very serious, all of them are very local and very easy to fix. In each case only a couple of lines of code need to be changed. It should take only a few hours to do the whole job for both the AV-OS and AV-TSx.

- *No use of high assurance development methods:* The AccuBasic interpreter does not appear to have been written using high-assurance development methodologies. It seems to have been written according to ordinary commercial practices. In the long run, if the interpreter remains part of the codebase, it and the rest of the codebase should be revised according to a more rigorous methodology that would, among other things, likely have prevented the bugs we found.

- *Interpreted code is contrary to standards:* Interpreted code in general is prohibited by the 2002 FEC Voluntary Voting System Standards, and also by the successor standard, the EAC's Voluntary Voting System Guidelines due to take effect in two years. In order for the Diebold software architecture to be in compliance, it would appear that either the AccuBasic language and interpreter have to be removed, or the standard will have to be changed.

- *Bugs detailed in confidential companion report:* In a companion report we have listed in great detail all of the bugs we identified, the lines at which they occur, and the threats they pose. Because that report contains Diebold proprietary information, and because it details exactly how to exploit the vulnerabilities we discovered, that report must be confidential.

Clearly there are serious security flaws in current state of the AV-OS and AV-TSx software. However, despite these serious vulnerabilities, we believe that the security issues are manageable by a reasonably careful combination of short- and long-term approaches. Here are our recommendations with regard to mitigation strategies.

In the short term, especially for local elections, the security problems related to AccuBasic and the memory cards might be managed according to guidelines such as these:

- *Strong control over access to memory cards for the AV-OS:* The AV-OS optical scan is vulnerable to both the Hursti attack and attacks based on the AccuBasic interpreter bugs we found. It would be safest if it is not widely used until these bugs are fixed, and until a modification is made to ensure that the Hursti attack is eliminated. But if the AV-OS is used, strong procedural safeguards should be implemented that prevent anyone from gaining unsupervised or undocumented access to a memory card, and these procedures should be maintained for the life of all cards. Such controls might include a dual-person rule (i.e. no one can be alone with a memory card); permanent serial numbers on memory cards along with chain-of custody documentation, so there is a paper trail to record who has access to which cards; numbered, tamper evident seals protecting access to the cards whenever they are out of control of county staff; and training of all personnel, including poll workers, regarding proper treatment of cards, and how to check for problems with the seals and record a problem. Any breach of control over a card should require that its contents be zeroed (in the presence of two people) before it is used again.

- *Require generation of new crypto keys for the AV-TSx:* The AV-TSx is not vulnerable to any of these memory card attacks *provided* that the default cryptographic key used for signing the contents of the memory card is changed to a new, unguessable key and kept secure. If the key is changed then these threats are all eliminated, at least for the short term. If this is not done, however, then the AV-TSx is no more secure than the AV-OS.

- *Control access to GEMS:* Access to GEMS should be tightly controlled. This is a good idea for many reasons, since a malicious person with access to GEMS can undermine the integrity of an election in many ways. In addition, in a TSx system, GEMS holds a copy of the cryptographic key used for signing the contents of the memory cards, and in both systems the GEMS server may hold master copies of the AccuBasic scripts loaded onto the memory cards.

In the longer term, one would want to consider a number of additional measures:

- *Fix bugs:* Certainly the bugs in the source code of the interpreters for both the AV-OS and AV-TSx should be corrected with all deliberate speed, the Hursti vulnerability should be fixed, and the code re-examined by independent experts to verify that it was properly done.

- *Defensive and high assurance programming methodology:* The source code of the interpreters should be revised to introduce systematic defensive programming practices and high assurance development methods. In particular, eliminate in the firmware, insofar as possible, any trust of the contents of the memory card.

- *Protect AccuBasic code from tampering:* The AccuBasic object code could be protected from tampering and modification, either by (a) storing AccuBasic object code on non-removable storage and treating it like firmware, or by (b) protecting AccuBasic object code from modification through the use of strong cryptography (particularly public- key signatures).

- *Don't store code on memory cards:* The architecture of the AV-OS and the AV-TSx could be changed so they do not store code on removable memory cards.

- *Remove interpreters and interpreted code:* The architecture of the AV-OS and the AV-TSx could be changed so they do not contain any interpreter or use any kind of interpreted code, in order to bring the codebase into compliance with standards.

This is a 5 page SUMMARY of a 38 page report which can be found on VTVOTERS.ORG under key documents. The memory cards Vermont uses with its Diebold AV-OS (optical scanning system) has been proven insecure by many studies. Why are we using this system?

Submitted by
Cynthia Johnson
456-1233
cynthiajohnson@
myfairpoint.net
Calais, Vt 05650