THE LEAGUE OF WOMEN VOTERS OF SOUTH CAROLINA

PO Box 8453, Columbia, SC, 29202, (803) 251-2726, www.lwvsc.org

Unsafe for Any Ballot Count:
A Computer Scientist's Look at the ES&S iVotronic in Light of Reports from Ohio, California, and Florida

Prepared for the League of Women Voters of South Carolina by Duncan Buell
14 January 2008

Executive Summary

The states of Florida and Ohio have conducted analyses of the ES&S iVotronic voting machine and its accompanying system and software. California has conducted a similar analysis of other voting machines. California and Ohio have decertified the use of Direct Recording Electronic (DRE) voting machines; the machine no longer certified in Ohio is the same machine (except possibly for precise model and software revision numbers) as is used in South Carolina.

The main conclusion in the Ohio report is that "...the ES&S Unity EMS, iVotronic DRE and M100 optical scan systems lack the fundamental technical controls necessary to guarantee a trustworthy election under operational conditions."
I believe the analysis done in Ohio and the strong conclusions drawn in that state's report fully justify a recommendation that South Carolina stop using the iVotronic machines absolutely as soon as possible, preferably before the November 2008 elections.

The Ohio report concluded that "[t]he firmware and configuration of the ES&S precinct hardware [the iVotronic] can be easily tampered with in the field," that "[t]here are exploitable weaknesses in virtually every election device and software module," and that there are "practical attacks that can be mounted by almost any participant in an election."

The South Carolina State Elections Commission has insisted that their security procedures make trustworthy elections with the iVotronic possible in South Carolina. Quite to the contrary, the Ohio analysis teams specifically tried to "identify practical procedural safeguards that might substantially increase the security of the ES&S system in practice," but "ultimately failed to find any such procedures that we could recommend with any degree of confidence."

I believe the SCSEC position to continue using the iVotronic requires them first to reject in its entirety the Ohio analysis and report. This is, I believe, completely

unwarranted. Many of the flaws found in password protocols and in software design are truly fundamental and longstanding errors one finds in badly designed systems or in code written by naïve programmers. That these errors should appear in a commercial voting machine system speaks volumes negatively about the iVotronic and its designers. This system has not been designed with security as a basic requirement, and it should not be used for voting in South Carolina.

Background

Introduction

This is written in an attempt to explain for laypeople the meaning and import of the recent reports from Florida [YASINSAC] and Ohio [EVEREST] that analyzed the ES&S iVotronic Direct Recording Electronic (DRE) machine, and the California report [CALIFORNIA] that analyzed a different ES&S machine. I will write somewhat informally in an attempt to get the message across. My initial audience is the state board of the League of Women Voters of South Carolina and the county elections commissioners of South Carolina.

A Personal Statement

I am not a Luddite when it comes to election machinery. I happen to believe that the technology does exist, in terms of protocols and procedures that could be turned into machines, for conducting secure and reliable elections. I do not know of an extant commercially-marketed machine that implements such a protocol, however. The technology exists, but machines have not been built, and thus none of the existing "electronic" machines can be considered acceptable.

I am also not a conspiracy theorist. I believe that the South Carolina State Elections Commission wants to hold secure and reliable elections, but I believe they are reluctant to face the fact that they have gone down a path that is embarrassing to them and to the state and they now must reverse course to change the machinery on which we vote and reverse the adoption decision of a few years ago. The decision to purchase the ES&S machinery may in fact have been defensible when the machinery was bought. The decision to continue using this machinery, in light of the report from Ohio, is indefensible.

I happen to believe strongly in the "software independence" recommendation put forward by the National Institute of Standards and Technology (NIST) Technical Guidelines Development Committee [BURR, RIVEST]. We have all lived long enough now in a world managed by the ubiquity of computer-controlled systems to know that software is apparently very hard to write properly and that computers do in fact fail. The principle of software independence is that there should be some other mechanism besides yet more software for verifying the ballot counts stored in a voting machine.

I also believe strongly that, even if we have doubts about the security of the process, *we must vote*. It is only by voting that we gain the standing to challenge later the use of the

ES&S machines. If we choose now not to vote, we give up the right to complain about the process.

Finally, it is worth addressing the basic question of a statewide standard for a voting machine, which has both benefits and disadvantages. The clear disadvantage is that if an attacker wanted either to corrupt or to disrupt an election in South Carolina, he/she would need only corrupt one system. This means that extraordinary care must be taken in protecting the elections process and ensuring the security of the machines. On the other hand, elections are generally run with a very limited budget and with volunteers as poll workers conducting the elections, so there is a great benefit in a statewide system that will allow for more standardized training. There is no inherent reason not to put all the eggs into one basket provided one properly cares for that basket.

Caveat

There is always the possibility that the precise model of machine or the precise version of software used in South Carolina differs from what has been examined in Florida, California, and Ohio. Mention is made, for example, of the fact that a three-letter password for one version of software will be upgraded to a six-letter password in some pending version of the software.

On the one hand, this says that some of the conclusions drawn might not be entirely correct.

On the other hand, the problems and the security flaws described in the reports are so absolutely fundamental that they call seriously into question the judgment of the company and its system designers. Based on the very elementary nature of the security flaws, I believe it is entirely reasonable to conclude that ES&S as a company cannot be trusted to produce a secure and reliable voting machine product. If I as a homeowner call in a security company to burglar-proof my house, and then *I* have to tell the company not to install the security door with the hinges on the outside, the company gets no bonus points for correcting the improper installation, and I would have ample reason to question any of the other security judgments that company made. Some things are just that simple, and most of the errors in the reports are of this nature.

Assumptions and Background

Some basic assumptions and constraints must be pointed out that make elections and election machinery different from a lot of other similar things.

First, we have to point out the obvious. No one—not the candidates, nor the political parties, nor the electorate, nor the elections commission--ever knows the actual results of the elections. All that is known is what the official ballot count is declared to be. This may or may not be the actual ballot count, and those declared the winners may or may not be those for whom the most votes were cast.

This obvious constraint is due to the requirement for anonymity in balloting. There is no mechanism for verifying that the votes as they are counted are indeed the votes as they are cast. Since we do not know the answer to the voting question, we must take every step possible to make sure that the process by which we arrive at an answer is foolproof.

The second constraint is the fact that elections are one-time events, and thus there is no way to conduct a proper "scientific" experiment with a tested environment and a control environment. The same election held on two successive days would, except perhaps in very tiny towns with only a handful of voters, result in different ballot totals. We don't get "do-overs" in elections.

This second constraint imposes a second "system" requirement, and that is that the process of conducting the election must not interfere with the ability of voters to cast ballots. We could require a very secure and verifiable process for collecting ballots, but if it that process is so slow or so cumbersome as to discourage voting, then it cannot be used. Similarly, we must ensure that the process cannot be sabotaged on election day, because we don't get a second chance to run the same election. A system that is insufficiently reliable even without the threat of sabotage is also unacceptable; we must ensure an extremely high probability that things will run smoothly.

I am often asked why we can't make voting machines work when in fact we can make ATMs work. These two constraints highlight the difference between machinery for bank card or credit card transactions and machinery for elections. The first major difference with ATM-like transactions is that the user does not have anonymity, but rather has just the opposite. A user who makes a transfer of funds, for example, that shows up as a debit to one account without the proper credit to another account, can easily expose their identity to the bank in order to clear up the error. Further, in the case of such transactions, there are a great many laws protecting consumer rights.

No such protection exists for voting. The better analogy for voting is not with bank transactions but for slot machine transactions in a casino. When one puts money in the slot machine, one has no guarantee that the machine won't just take the money. One has no individual guarantee that one's transaction has been properly chosen randomly as a winner or loser.

And indeed this analogy has been made by a computer scientist from Nevada who has experience both with gaming industry regulation in Nevada and with electronic voting machines in Nevada [KUBILUS]. At the very least, the regulations governing slot machines in Nevada could be copied nationally for voting machines.

Security Through Obscurity

Much of the problem with all DREs is that the technology is proprietary rather than open to all to view. Both ES&S and SCSEC have advertised that the proprietary nature of their technology adds to security. This is called the "security through obscurity" argument, and it is *absolutely rejected* by any and all security experts. It is a strong negative statement about the technological maturity of the SCSEC that they would ever have advertised this as a benefit.

Kerckhoffs' principle, enunciated in 1883 by Auguste Kerckhoffs, is that a cryptographic system should be immune from weakness even if its entire structure is known. The security comes from the algorithm, not the possibility that the cryptographic system is kept secret from the opposition. Claude Shannon restated this as "the enemy knows the system," and Bruce Schneier (the author of the Blowfish algorithm used in the iVotronic) applies this to security systems in general: "every secret creates a potential failure point."

Another view of security through obscurity is this, and it is the principle applied by the U.S. government: We keep secret as much as we can, but we do that *on top of* and not instead of the inherent security in the system.

Finally, we present the skeptic's view of security. In 2004 there was something like a billion dollars spent in the combined presidential campaigns. If we are to rely in any way on security through secrecy and proprietary technology, then we must assume that nowhere in the manufacturing process, the software writing, the software updating, the printing of the use or repair manuals, the marketing, or anywhere in the chain from the state and county elections commissions anywhere in the country is there a conscience that can be bought. With a billion dollars going into the process, I think it's utterly foolish to assume that it would be impossible to find someone who could be bought for a million dollars or so skimmed off the bottom of a stack of cash. It just doesn't make sense.

The Rivest Committee Report to NIST

In the late fall of 2006 the Technical Guidelines Development Committee submitted a report on voting machines to the NIST Elections Assistance Commission responsible for voting standards. Prominent in the recommendations of that committee was the (justifiable in my opinion) rejection of any machine that did not have a software independent means for verifying the vote count [BURR, RIVEST].

What this means is the following. In a DRE such as the iVotronic, with no paper trail, the *only* record of the votes cast is the electronic record stored in the flash memory in the iVotronic itself. It is true that one can apply a number of fail-safe techniques to ensure that the electronic record is not corrupted and that the record is consistent. (In the iVotronic, for example, there are three identical records that are constantly checked against each other; if the records are not found to be the same, the machine is assumed to be malfunctioning and a warning is signaled that the machine should be taken out of use.)

However, there is in the iVotronic no ballot record that is independent of software. There is no paper ballot to be counted a second time. A second count in the iVotronic could only be done by reading the memory chips a second time. That read would be done by software, just as it was software that put the record in the chips. This does not make the "recount" a process that is independent of the process that did the first count.

The Analyses of the iVotronic Machines

The Florida Report

The first indication of problems in the iVotronic comes in the Florida report prepared with Alec Yasinsac of Florida State University as lead author [YASINSAC]. This report was prepared in response to a challenge in the 13th Congressional District in the 2006 election. In that election, there was a huge undervote reported for the congressional election, and the challenge to the election included a suggestion that the iVotronic machines had either failed or had been tampered with.

The Florida report had a very narrow scope, namely to determine whether the legitimacy of the election result in the 13th district could be called into question by problems with the iVotronic machines themselves. They did not, for example, concern themselves with general security issues, nor did they concern themselves with the ES&S Unity software running at county headquarters doing the aggregation of vote counts from the individual precincts.

The Florida report did not find a reason to believe that foul play had occurred. The conclusion with respect to the 13th district election was that a poorly designed ballot image had made it too easy to overlook the existence of that election contest on the same ballot screen as a much higher profile race.

However, Appendix D on passwords, found on pages 66-67 of the report and included here as Appendix A to this document, includes several troubling findings. The "modem password" and the "override password" can be set at the Unity server when the election is configured by the county. If the modem password is not set at the county level, it has a default value that is hard coded into the source code and that is the same for all machines across the United States.

All other passwords (there are five of them) are fixed, hard coded in the source code, and are the same for all machines anywhere in the United States. In one of the great understatements of the 21st century, Yasinsac writes, "This represents poor practice."

Further, four of these passwords are three letters long and are case insensitive. (By case insensitive is meant that "the", "thE", "tHe", "The", "tHE", "ThE", "THe", and "THE" would all be the same password.) In the words of the report, "Each one is chosen to be mnemonic and easy to remember. The problem is that they are also likely to be fairly easy to guess. They follow a memorable pattern. Someone who knows one of these passwords can probably guess what the other ones are without too much difficulty. These passwords provide very little security." (In a discussion of this with a colleague in my department we both came to the conclusion that this description means that the passwords are probably something like "ABC", "DEF", and "GHI".)

Two of the four passwords mentioned above allow one to replace the existing software on the iVotronic, perhaps inserting malicious code, changing votes in any direction, or infecting the machine with a virus.

In the words of the report, "Our judgment is that the password mechanisms on the iVotronic are poorly conceived and poorly implemented. The consequence is that the passwords by themselves do not do a good job of preventing unauthorized individuals from accessing critical system functions."

Finally, there is a special kind of Personalized Electronic Ballot (PEB), called the Factory Test PEB (later called the Factory Quality Assurance PEB in the Ohio report). This is distinguished from all other PEBs by only a single character that is sent to the iVotronic when the PEB is inserted in the slot. The special value is hard coded into the software (and thus probably the same value for all machines produced across the country). Anyone who knows this one-character value, has access to another PEB and can program that PEB, can reprogram the PEB to send the Factory Test character instead of the regular PEB character. By doing this, all password codes are bypassed and the user has full access to the machine for uploading/changing the software, loading or changing votes cast, etc. This is referred to in the Florida report as an "undocumented backdoor". When such a PEB is used, the log of activity is written as if the correct passwords had been provided by an ordinary PEB.

An Aside: What Does This Mean?

Let me try to explain what is meant by some of these security flaws. First, what does it mean to say that the passwords are hard-coded into the software, that the passwords are stored in the clear, or that there is a back door into the machine?

When a computer scientist says that something is hard coded, it means that the source code of the program explicitly contains the information. The source code might look something like this:

```
prompt ← "Please enter the password:"
display the prompt
password ← get password from the console
if( password = "abc") then
  allow access
else
  prevent access
```

In this case the password, namely the string "abc," literally and specifically appears in the near-English text of the computer program. If as an attacker I can get access to the source code, then I can discover the password. It is usually the case that even if I get access only to the executable version of the program (the "dot exe" file on Microsoft

Windows machines), the string “abc” will appear in the clear as “abc” in the dot exe file and can be read as plain English. This is a very insecure way of doing business. And in order to change the password (from “abc” to “def”, for example), it is necessary to load a completely new version of the software. Since reloading software would have to be done on all machines in a jurisdiction, this is a painful process that would be done only very infrequently or else there would be complications of conflicting versions of the software.

A somewhat more secure, but still not truly secure, approach is to store the password in a file on the computer. The source code might now look like the following.

```
prompt ← "Please enter the password:"
display the prompt
userinput ← get password from the console
storedpassword ← get actual password from the file
if( userinput = storedpassword) then
  allow access
else
  prevent access
```

At least now the actual password is not stored as a string of characters in the program. The password can be changed by changing the stored file, but the program need not change. This is somewhat more secure, but if the password in the file is stored as plain text, then an attacker can discover the password by looking in the file. (I actually did this once as part of a computer security consulting job I had. I had been called in with a colleague by a company to investigate whether there was malfeasance on the part of the computer systems staff. In the middle of the night, with the chief of physical security next to us, we uncovered the main system password in the clear in the appropriate file and proceeded to dump the system logs.)

The Right and Proper Way to maintain password security is not to maintain any passwords as plain text in the system. On my Linux server at work, the **`/etc/passwd`** file might have an entry

```
     buell:7nksfN2%w23
```

that was the password entry for my computer account. But my password is not the string **`7nksfN2%w23.`** Rather, that string is an *encrypted* version of my password. When I change my password, the new password is encrypted and it is the encrypted version that is stored in the **`/etc/passwd`** file[1]. Then when next I try to login (or when anyone tries to log in to my account), I provide my password to the password checking program. My input is then encrypted, and the encrypted version is compared against the encrypted string **`7nksfN2%w23`**. If there is a match, I am allowed to log in. If not, I must try again.

[1] Actually, these days it is usually the **`/etc/shadow`** file but it is often still referred to as **`/etc/passwd`**.

This is in fact an excellent example of the use of Kerckhoffs' principle. Nothing in the process or the system is hidden from anyone. The only secret is my password. Anyone in the world dealing with a Unix/Linux system knows to go look in **`/etc/passwd`** for account names and passwords. Anyone in the world knows that Unix/Linux systems have handled passwords with this mechanism since at least the late 1970s. The encryption algorithm is even published[2]. All the security is centered on the power of the encryption algorithm and on my ability to keep my password a secret.

And this kind of password mechanism is not rocket science; it is high school material. I can almost see in my mind's eye a ninth-grader at a science fair looking up at me[3] and sheepishly admitting that he or she had stored passwords hard coded in the clear in the program. "Yes," I would be told, "I know it's a dumb thing to do, but I wanted to concentrate on other things and I ran out of time." For 36 million dollars, we in South Carolina deserve better than something a ninth grader would admit was a dumb mistake.

So much for password systems. What about back doors? It is not at all uncommon that a computer or software vendor will build a back door into a system during the development process. In geekspeak, this would be referred to as a means by which a systems person could declare to the software "I am god. Let me in." There are good reasons for doing something like this. If a software developer has to go through a lengthy (but secure) login process to get to the testing phase for his or her work, then the entire development process can be slowed unnecessarily. If one is developing the actual password checking program, then it might be advisable to have a way to circumvent password checking just in case something disastrous happened. (For example, if the **`/etc/passwd`** file gets corrupted on disk, then perhaps no one could ever get in to the system.) The purpose of the back door is to speed the development process and to allow for the possibility of a showstopping error while in development.

Back doors, however, should always be carefully documented. If left undocumented, they represent a serious failure point in the security system. I remember a lawsuit from the 1980s. A customer had bought a software package with a back door of which they were unaware. When someone broke in through the back door and caused financial mayhem to the customer's business, the customer sued the vendor on the basis that they should have been told about the potential security hole so they could take steps to disable that "feature".

The 2007 Ohio Report

Now we come to the December 2007 report from the Secretary of State for Ohio [EVEREST, MICROSOLVED1, MICROSOLVED2, MICROSOLVED3]. This report

[2] From the **`man`** page on my server: "The UNIX System encryption method is based on the NBS DES algorithm and is very secure. The size of the key space depends upon the randomness of the password which is selected."

[3] Literally, not necessarily figuratively; I am taller than most ninth graders.

covers the iVotronic as well as several other electronic devices. I will address only that part of the report that refers to systems using the iVotronic; this is Part II, pages 27-99. In what follows here, page numbers will refer to the EVEREST report [EVEREST]. This was the result of source code analysis and penetration testing done by a team at the University of Pennsylvania and by WebWise Security, Inc., in Santa Barbara, California. The systems analyzed (both the iVotronic and the M100 optical scan system) consisted of nearly 670,000 lines of code in twelve programming languages and running on five hardware platforms.

For the first time, we have in the Ohio report an analysis of the *complete* election and ballot-counting process. The Florida report specifically notes that its scope was limited. The Ohio report includes not just the iVotronic hardware but the Unity software system used at the county (or equivalent) level for collecting ballot counts.

This report is utterly devastating in its conclusions. I will begin by quoting from the executive summary (pages 29-30, included as Appendix B):

> "Our analysis suggests that the ES&S Unity EMS, iVotronic DRE and M100 optical scan systems lack the fundamental technical controls necessary to guarantee a trustworthy election under operational conditions.
>
> "The firmware and configuration of the ES&S precinct hardware can be easily tampered with in the field.
>
> "Access to administrative and voter functions are [sic] protected with ineffective security mechanisms.
>
> "Many of the most serious vulnerabilities in the ES&S system arise from the incorrect use of security technologies such as cryptography.
>
> "...taken as a whole, the security failures in the ES&S system are of a magnitude and depth that, absent a substantial re-engineering of the software itself, renders [sic] procedural changes alone unlikely to meaningfully improve security. Nevertheless, we attempted to identify practical procedural safeguards that might substantially increase the security of the ES&S system in practice. We regret that we ultimately failed to find any such procedures that we could recommend with any degree of confidence.
>
> "The security failings of the ES&S system are severe and pervasive. There are exploitable weaknesses in virtually every election device and software module, and we found practical attacks that can be mounted by almost any participant in an election."

For reference, we mention here that the Unity system that is referred to is the Windows-based election management software suite. One feature of this report from Ohio is that it is the first time that the entire Unity suite has been analyzed and reported upon.

I believe that much of this document is self explanatory, so I will simply quote extensively.

> (page 50) "Access to the iVotronic DRE configuration is protected by several hardware and password mechanisms, all of which can be defeated through apparently routine poll worker (and in some cases voter) access."
>
> "In spite of the proprietary nature of the "official" PEB, we found it to be relatively simple to emulate a PEB to an iVotronic or to read or alter the contents of a PEB using only inexpensive and commercially available [infrared]-based computing devices (such as Palm Pilot PDAs and various mobile telephones)."
>
> (page 51)
> "Many of the more sensitive iVotronic administrative functions (closing the polls, clearing the terminal, etc.) require the entry of passwords in addition to the insertion of a a supervisor PEB. However, there is a special Quality Assurance (QA) PEB type recognized by the iVotronic firmware that behaves essentially as a supervisor PEB but that, when used, does not require the entry of any passwords. This PEB does not appear to have been described or documented in any of the ES&S manuals or training materials provided to our review."
>
> "Because PEBs themselves enforce no passwords or access control features, physical contact with a PEB (or sufficient proximity to activate its magnetic switch and [infrared] window) is sufficient to allow reading or writing of its memory."
>
> "An attacker who has access to a precinct's main PEB when the polls are being closed can alter the precinct's reported vote tallies, and, as noted in Section 6.3, can inject code that takes control over the county-wide back-end system (and that thus affects the results reported for all of a county's precincts)."

As another indication of the quality of the hardware as designed, the report states that the mechanical locks "were uniformly of very low-security designs that can easily be picked or otherwise bypassed." On the other hand, this turned out to be a positive thing for the analysis, as was stated in a footnote: "For the first weeks of the project, we did not have the correct keys for much of the equipment; we frequently had to pick the locks in order to conduct our analysis."

The Unity software has buffer overflow errors (this is a very common software error, to which the iVotronic is also subject) that allow, for example, a single PEB returned from a single precinct to compromise an entire county's election: "Note that because these vulnerabilities affect the central counting system, a corrupted media attack conducted from *any single* precinct [italics in the original] can corrupt results for the entire county. We have successfully implemented PEB-based attacks against Unity (at the University of Pennsylvania and at WebWise) and have confirmed that such attacks represent a readily-exploitable threat…" (page 53)

It is worth going into a slight digression about buffer overflows, which apparently are absolutely all over the place in the ES&S system. These are perhaps the most common, most well-known, and easiest to fix of all computer software bugs that negatively affect security. When the Robert Morris worm brought down the internet in November of 1988, the public network systems at my previous job went down along with the rest. By mid-afternoon we had word from Pittsburgh (by the predecessor of the CERT organization that is now the center of internet attack analysis and prevention) that it was a buffer overflow in the **`sendmail`** program. One of my colleagues immediately responded, "That's cheating. We knew all about that bug." It says a great deal about ES&S's software people, and all of it is negative, that a software error that has been a no-no for more than twenty years, and which is detected by many programs that will analyze code for security flaws, seems to be prevalent in all software modules and on all the hardware platforms in the voting system.

A further comment about Unity is this: "… there are many potential vulnerabilities that can be mitigated only through careful, expert system management. Unfortunately, the precise requirements for using Unity in a networked Windows environment are not specified by ES&S, and appear to be left to individual counties to manage without specific guidance." We note that the rules in South Carolina for security for elections are not subject to a FOIA request, so we have no idea in South Carolina what general computer security policies are mandated nor whether they are practiced. This appears as a theme in the California reports as well and is a problem not just for ES&S but for other vendors—the focus is on the first step in the ballot-counting process, namely the voting machines themselves, and then the vendors are silent when it comes to advising election authorities on secure configurations of the computers used downstream to aggregate the vote counts at the county level.

Finally, we mention the ineffective use of cryptography: "The iVotronic DRE uses cryptography to protect data stored on the PEB and the [Compact Flash] card. … Unfortunately, the manner in which the encrypted data is stored on the PEBs effectively neutralizes the cryptographic protection. The PEB contains [an Election Qualification Code (EQC)], encoded using an unkeyed (non-cryptographic) algorithm. The EQC is used to encrypt the [cryptographic] key, which is used to encrypt the rest of the data on the PEB. That is, although much of the data on the PEB is encrypted, there is unencrypted information stored along with it that allows an attacker to easily discover the key."

This again is a common problem for systems that are not designed from the very beginning to be highly secure. The best cryptographic algorithm in the world is only as secure as is the protocol for managing the cryptographic keys, and it is often the case that the key management system is the weak link. A comment on the Ohio report says that the ES&S key management is like locking a secret in an unbreakable safe and then painting the combination on the outside.

Conclusions

What can we conclude from all this?

Two conclusions are, I believe, entirely warranted. First of all, we should accept the fact that the Ohio analyses have been done by experts and that they describe realistic exploits that could be mounted against the iVotronic and the downstream system that aggregates votes from iVotronic machines. Many, perhaps most, of the vulnerabilities are not just "reports and studies produced in the sterile environment of the laboratory" [WHITMIRE]. In fact, the Ohio analysis considered procedural changes (such as have been alleged to be sufficient in South Carolina) and specifically says that procedural changes alone are "unlikely to meaningfully improve security."

Second, I believe the password flaws, the buffer overflow errors, the flimsy physical locks, and similar complaints, allow us to be justified in asserting that ES&S is not competent to produce voting machines that should be used in elections. These are simple, standard, naïve errors. They are exactly the kinds of errors that students are taught how to avoid in an undergraduate university computer science course, and the ES&S machine, as analyzed by Ohio, would not get a passing grade. The machine as produced and sold should be a great embarrassment to ES&S. That they continue to defend what they have built indicates to me that they are unwilling to admit that they have not followed standard security practices and are unwilling to change so as to follow them in the future.

The machines should be decommissioned by South Carolina and replaced absolutely as soon as possible.

A Positive Recommendation

I am often asked what I am *for*, given that I am against the iVotronic and similar software-only DREs. Here is my response.

At present, we have a number of companies who have manufactured hardware for electronic voting machines. Nearly all of these (or indeed perhaps all of them) have proprietary technology that can be understood only in a limited way if at all, and all the machines are rather different from one another.

This is a horrible way to run something as important as elections. We get a bunch of very different machines, so in order to verify their security and reliability we have to examine a number of different kinds of problems, and yet we only get to see the details under a Non-Disclosure Agreement or other controlled circumstances. This makes it hard to test for security and hard to compare one product against another, and it makes every security test a completely new exercise.

A better approach would be this: The U.S. National Institute for Standards and Technology should be mandated (by Congress) to promulgate a standard. Vendors would then have to build to that standard. In this way, we would have a benchmark against which to test. Any vendor could make a machine, but the basic rules of operation and the basic security requirements would be the same. This recommendation would make full use of NIST in its standards-making capacity and would require vendors then to concentrate on how best they could manufacture to the standard. The governments and the states would get the added benefit of institutional memory when it comes to testing, since the errors and flaws would be held up against a common standard that would carry forward in time from one machine and from one test to the next.

References

BURR: W. Burr, J. Kelsey, R. Peralta, and J. Wack, "Requiring software independence in VVSG 2007: STS recommendations for the TGDC", Draft report to NIST, November 2006. Available at http://vote.nist.gov/DraftWhitePaperOnSIinVVSG2007-20061120.pdf, last accessed 9 January 2008.

CALIFORNIA: website for the Secretary of State Debra Bowen http://www.sos.ca.gov/elections/elections_vsr.htm, last accessed 9 January 2008.

COMPUWARE2003: Compuware Corporation, "Direct Recording Electronic (DRE) Technical Security Assessment Report," prepared for the Ohio Secretary of State Kenneth Blackwell, 21 November 2003.

COMPUWARE2005: Compuware Corporation, "ES&S Direct Recording Electronic (DRE) and Voter Verified Paper Audit Trail (VVPAT) Technical Security Assessment Report," prepared for the Ohio Secretary of State Kenneth Blackwell, 4 November 2005.

EVEREST: "EVEREST: Evaluation and Validation of Election-Related Equipment, Standards and Testing," prepared for the Ohio Secretary of State Jennifer Brunner, 7 December 2007. Available at http://www.sos.state.oh.us/sos/info/everest.aspx, last accessed 9 January 2008.

HAUG: Nola M. Haug, "Maryland/Ohio Security Assessments Gap Analysis," prepared for the Ohio Secretary of State Kenneth Blackwell, 26 February 2004. Available at the EVEREST website.

KUBILUS: Norbert J. Kubilus, Letter to the editor, *ComputerWorld*, 30 October 2006.

MICROSOLVED1: Microsolved, Inc., "ES&S System: Executive Summary Report," prepared for the Ohio Secretary of State Jennifer Brunner, 7 December 2007. Available at the EVEREST website.

MICROSOLVED2: Microsolved, Inc., "ES&S System: Technical Manager's Report," prepared for the Ohio Secretary of State Jennifer Brunner, 7 December 2007. Available at the EVEREST website.

MICROSOLVED3: Microsolved, Inc., "ES&S System: Technical Details Report," prepared for the Ohio Secretary of State Jennifer Brunner, 7 December 2007. Available at the EVEREST website.

RABA: RABA Technologies, "Trusted Agent Report: Diebold AccuVote-TS Voting System," prepared for the Maryland Department of Legislative Services, 20 January 2004. Available at http://www.raba.com/press/TA_Report_AccuVote.pdf, last accessed 9 January 2008.

RIVEST: Ronald L. Rivest and John P. Wack, "On the notion of 'software independence' in voting systems," draft report to NIST, 28 July 2006. Available at http://vote.nist.gov/SI-in-voting.pdf, last accessed 9 January 2008.

WHITMIRE: Chris Whitmire, speaking for the South Carolina State Elections Commission, as quoted in the Greenville (South Carolina) *News* 5 January 2008.

YASINSAC: A. Yasinsac, D. Wagner, M. Bishop, T. Baker, B. de Medeiros, G. Tyson, M. Shamos, and M. Burmester, "Software review and security analysis of the ES&S iVotronic 8.0.1.2 voting machine firmware," Florida State University, February 23, 2007. Web copy at http://election.dos.state.fl.us/pdf/FinalAudRepSAIT.pdf, last accessed 7 January 2008.

Biography

Duncan Buell earned a Ph.D. in mathematics in 1976 from the University of Illinois at Chicago and is presently a professor in and chair of the Department of Computer Science and Engineering at the University of South Carolina. Prior to his move to USC in 2000, he spent fifteen years with the Institute for Defense Analyses in Bowie, Maryland, a research laboratory conducting mathematics and computing research for the National Security Agency.

The League of Women Voters, a nonpartisan political organization, encourages informed and active participation in government, works to increase understanding of major public policy issues, and influences public policy through education and advocacy. Membership in the League is open to men and women of all ages.