Connecting Work on Threat Analysis to the Real World

Douglas W. Jones*
The University of Iowa

presented at
Threat Analyses for Voting System Categories
A Workshop on Rating Voting Methods
VSRW 06,
George Washington University,

8-9 June 2006

As a prefatory remark, I should note that I initially hesitated to present this paper because it names vendors and describes long-standing problems with some of their products. But there is no way around this. If we cannot speak about real systems and real threats here, where can we speak about them? In any event, while products made by Diebold feature prominently here, they are not the only vendor I mention, and my intent is to focus on the larger system that allowed these problems to persist years after they were first uncovered, not to focus on the particular vendors that were responsible for these problems in the first place.

A Voting System Security Problem

Almost a decade ago, on November 6, 1997, I had the opportunity to examine a new voting system developed by I-Mark Systems and recently acquired by Global Election Systems. At the time, I was serving on the Iowa Board of Examiners for Voting Machines and Electronic Voting Systems, and the vendor, Global, had brought their AccuTouch system to us for examination. It did not take me long to find a significant security vulnerability. I pointed this out to the vendor's representatives who were present at the examination, and I assumed that this was a sufficient action on my part.

It seemed unnecessary to raise a public alarm or to call in the press. The meeting where I pointed this problem out was a public meeting, although I don't recall more than one or two county election officials being present as members of the public. The minutes of the meeting are a public record, available to anyone. They described the security vulnerability as follows: "Dr. Jones also expressed concern about data encryption standards used to guarantee the integrity of the data on the machine. DES requires the use of electronic keys to lock and unlock all critical data. Currently all machines use the same key."

As I recall the discussion that was reported so tersely in the minutes, I told Bob Urosevich and Barry Herron, the Global representatives at the meeting, that embedding the encryption keys in the source code might be acceptable in a prototype proof-of-concept system, but that they needed to do better key management before the system went into widespread public use. I told them that, so long as the encryption keys were in the source code, they needed to guarantee that the source code would be tightly guarded, and they needed to guarantee that no voting systems would ever be sent to the landfill or to surplus sales outlets without first deleting all object code from them.

* The author's participation in this workshop was partially supported by NIST and by NSF grant CNS-052431 (ACCURATE);

What alarmed as much as the security flaw I'd found was the fact that Wyle Labs, the Independent Testing Authority that had examined the I-Mark system, had not seen the problem. Their September 10, 1996 report on *Qualification Testing of the I-Mark Electronic Ballot Station* contained enough information to reveal the presence of the key-management problem I had identified, but the source code examiners had not noticed any problem, and in fact, the report indicated that the security of this system was particularly impressive because of its use of DES.

In the years that followed, Global was acquired by Diebold, and the AccuTouch was repackaged and re-branded as the AccuVote TS. Bob Urosevitch rose to President of Diebold Election Systems, and the AccuVote TS was widely adopted for use in many states. I assumed that the security vulnerability that I had identified had been fixed, although Global and Diebold did not bring that machine back to Iowa for many years.

On On July 24, 2003, Tadayoshi Kohno, Adam Stubblefield, Aviel D. Rubin and Dan S. Wallach released a report entitled *Analysis of an Electronic Voting System* [1]. This report has since come to be known as the Hopkins Report. As it turned out, Global had left their source code on a public FTP site, and neither Diebold nor Global had seen fit to make any repairs to the security problem I had identified. The DES key was still a constant exposed in the source code.

My response was immediate. I was still on the Board of Examiners for Iowa. I E-mailed our state election director and asked that the Diebold AccuVote be immediately decertified. This was an empty action, since the original certification was provisional and had expired, but I assumed that our state election director was in regular contact with others about such matters.

One of the followups to the Hopkins Report was an examination of the Diebold AccuVote TS by RABA Technologies commissioned by the State of Maryland. The RABA *Trusted Agent Report* was issued on January 20, 2004 [2]. In addition to confirming the presence of the flaws reported in the Hopkins Report, the RABA report also documented a new security vulnerability.

On page 19 of the *Trusted Agent Report*, item 3 begins: "Load a PCMCIA card with an update file. The PCMCIA card can be used to update the software on the AccuVote-TS terminal. This can be done by placing a PCMCIA card with an update file into the terminal and rebooting the terminal. The update file allows an attacker to overwrite any file on the system."

General Recommendation 10, on page 24 of the report, directly addressed this vulnerability: "Do not allow software updates without authentication."

The State of Maryland assured the public that the recommendations in the RABA report were being taken seriously in an official *Response* to the *Trusted Agent Report*, issued on January 29, 2004 [3]. This response itemized the state's response to the Immediate Recommendations of the Trusted Agent Report, but it was largely silent about the general recommendations. Nonetheless, it was reasonable to expect the vendor to implement the general recommendations, particularly those that were relatively straightforward, and it was reasonable to expect the State of Maryland to demand that Diebold do this.

A bit more than 2 years later, on May 11 of this year, Harri Hursti, in conjunction with Black Box Voting, issued a report entitled *Diebold TSx Evaluation* [4]. This report clearly documents the fact that the security flaw documented in the RABA report is still present, and that, in fact, it is a much bigger problem than the RABA report had implied. There are actually three security flaws, technically differing from each other, but each allowing a devastating attack on the system from the PCMCIA card.

I want to emphasize that this story is not just about Diebold, it is about our voting system standards and about the demands that states make on the vendors. Other vendors have included similar security flaws in their systems. Consider, for example, the PBC 100 mark-sense scanner formerly made by American Information Systems and now made by Election Systems and Software.

When upgrades to this system came before Iowa in 2001 I found this item in the software change log for firmware version 4.04: "Added support for reading PCMCIA modem communication utilities from the election definition PCMCIA SRAM card." As is the case with the Diebold AccuVote product line, the PBC 100 is configured for each election by loading an election definition onto a PCMCIA card which is then inserted into the machine. Here was evidence that object code, part of the voting system firmware, was being dynamically loaded from this removable card into the voting system, exactly the same problem that RABA Technologies and Harri Hursti would later identify in the Diebold TSx.

In answer to my followup questions about this find, Steve Bolton of ES&S reported that this "was a temporary fix due to memory allocation and has subsequently been eliminated. This placed the drivers of the modem on the PCMCIA card to save room for the compiled source code of the firmware on the system. It did not raise any concern at Wyle as it was only placing 'off the shelf' modem drivers on the card that were read only during the modem transmission."

Between my original discovery of the problem with key management, Harri Hursti's discovery of the PCMCIA card attacks on the Diebold TSx, and my discovery of the problem with the ES&S Model 100, we have evidence of three distinct cases where Wyle Labs or their subcontractors failed in their evaluation of voting system firmware. It is fair to ask why! Why is it difficult to evaluate security? Why is it that neither states, the independent testing authorities, nor the vendors appear interested in taking public disclosures of security problems seriously.

The Root Problem

Some measures of voting system performance can be stated objectively and are easily compared. Purchase price, size and weight are obviously in this category. Other measures can be stated objectively but are difficult to measure. Among these, lifetime operating cost and accessibility are quite interesting. Finally some performance factors are either extremely difficult to quantify, or are simply not quantifiable. Unfortunately, voting system security appears to fall into this class.

As a general rule, when people make evaluations based on a mix of objective and subjective measures, it seems that they tend to discount the subjective. Where some measures are easily quantified while others are difficult, to quantify, it seems that people generally discount those that are difficult.

In the area of voting system price, it is quite clear that the amortized price per vote over the lifetime of a voting system is far more important than the purchase price. Given this, one would hope that state and county governments would spend a significant effort attempting to document these costs for existing systems and predict these costs for new systems. After a century of voting machines and two centuries of democracy, it would seem quite reasonable to expect a fairly detailed literature to have developed on this subject. Sadly, this is not the case. Even after a century of experience dealing with voting system vendors, most counties are basing purchase decisions on up-front costs, with only the vaguest attention to the other costs involved.

In the area of voting system accessibility, there is one obvious objective criterion we could use: For any given voting system, we can measure the fraction of the population able to cast a vote as intended without the need for assistance. Unfortunately, this blindingly simple criterion is difficult to measure, so we typically fall back on a distinctly inferior criterion, simply enumerating the particular disabilities that our voting systems are expected to accommodate. This risks creating a system where those with disabilities that are covered by the enumeration have voting rights superior to those with un-enumerated disabilities.

When it comes to security, it can be difficult to find objective criteria. Certainly, it is not sufficient to merely say that the system shall be secure. As we have seen above, the security requirements of Section 6 of the 2002 voting system standards have proven to be ineffective [5]. Hursti's attack demonstrates very clearly that the controls required by Section 6.4.1 are not present. That section says that "the system bootstrap, ... may be resident permanently as firmware, provided that this firmware has been shown to be inaccessible to activation or control by any means other than [those] authorized ..."

Security Metrics That Would Have Helped

The fundamental problem is that our security standards ask that a negative be demonstrated, that is, that it be shown that something is not possible. Demanding a demonstration of impossibility is sufficiently difficult that we should avoid it. What we need are security metrics that are positive.

One interesting metric is provided by *fuzz testing* [6]. We can require that voting systems pass a fuzz test on every input to the system. Fuzz testing one voting system input involves repeatedly presenting random values at that input. As a pass-fail test, we say that a product fails the test if it ever halts, throws an unhandled exception, goes into an infinite loop, or suffers any other damage under these tests.

Fuzz testing would probably have detected the problems discovered by California Voting Systems Technology Assessment Advisory Board this spring in their *Security Analysis of the Diebold AccuBasic Interpreter* [7]. Those problems involved a variety of buffer overflow vulnerabilities and unchecked pointer uses, errors that are typical of those that fuzz testing can detect. The Diebold TS and TSx patch file mechanism would also be likely to fail fuzz testing, so long as the patch files were properly identified as inputs into the voting system.

Fuzz tests are, without question, stupid. Random inputs make no use of any knowledge about the application other than the mere fact that it has an input that may be manipulated. It is fair to ask for more robust tests [8].

Syntax testing is one example of a more sophisticated test. This uses syntactically correct input data seeded with random errors. Syntax testing is practical wherever you have a file of valid input that you can experimentally corrupt. Again, as with fuzz testing, it is not generally sufficient to do only one test. Repeated tries with different random alterations to various correct files are necessary. As with fuzz testing, syntax testing is a pass-fail test.

Of course the original problem I found in the Global AccuTouch was of a different sort entirely. The lack of proper key management in that product cannot be detected by testing. There, it is clear, we need to offer, to the election industry, a set of acceptable models for key management and similarly difficult aspects of secure software design. The models should come with this simple message: Either implement one of these models or demonstrate that you can do better [9].

Another thing we can do is construct a library of security exploits and demand that each new incarnation of each voting system be tested against the exploits that have previously been identified for that system. This requires that security evaluators take a dangerous step! Normally, we have considered it sufficient to merely identify a vulnerability, without actually constructing the necessary framework to exploit that vulnerability. This has protected security researchers from the possible charge that they are providing tools that might be used by criminals, but it also means that those intent on ignoring security problems have been able to dismiss the security problems we have uncovered as mere alleged and theoretical vulnerabilities.

A Secondary Problem

While subjective security measures are difficult to compare with objective measures of such variables as cost and accessibility, the security problems with today's voting systems have hardly been obscure. Various of the problems discussed here have been identified in many other reports. In addition to the RABA report, Maryland commissioned an extensive study of the Diebold system by SAIC [10], and Ohio commissioned a report from Compuware that found problems in a number of voting systems [11].

These reports state firm conclusions. For example "Compuware identified several significant security issues, which left unmitigated would provide an opportunity for an attacker to disrupt the election process or throw the election results into question." The strongest claim they make about the mitigation measures they suggested is that they "will limit the likelihood of a successful attack or inadvertent disruption to the election process" [11, page 26].

This is hardly reassuring, yet most states have consistently implemented some subset of the mitigating measures proposed by one or another of these reports, and then made firm public pronouncements that their voting systems are now secure.

Limiting the likelihood of a successful attack is not the same as eliminating a security vulnerability, and in many cases, the vulnerabilities identified in one or another of the reports cited here could have been eliminated without great difficulty. Why is it, then, that the states have not demanded much from the vendors?

I can think of three distinct reasons, each of which is worthy of discussion:

First, to admit that a security vulnerability is significant could reduce public confidence in the system of elections. Therefore, election officials conscientiously avoid public discussion of shortcomings of the systems they have in place. It is clearly true that public confidence matters. Even if our elections are entirely honest, the legitimacy of our government is threatened if the public does not have confidence in this honesty.

Of course, we want public confidence to be earned and justified. Public confidence maintained by propaganda that misrepresents or manipulates the truth does not become a democratic system, yet we have seen exactly this kind of material put out by some state election offices [12].

The second reason is as folows: To admit that a security vulnerability is significant in a system purchased with public funds suggests a violation of public trust. Once the public money has been spent, it is therefore natural for those who spent it to defend their purchase decision vigorously.

I do not suggest that such defense is cynical, but rather, that the psychology of spending large sums of money naturally predisposes people to overlook flaws in whatever it is that they purchased. So long as the product is basically functional, whether it be a car or a house or a voting system, purchasers hesitate to be too critical, even when it was their own money.

Finally, to admit that a security vulnerability is significant may threaten the working partnership between the state or county and the vendor. Most states and counties need a strong working partnership with the vendor because they rely on the vendor for technical expertise in the management of the voting system. Partnerships rest on trust, and to raise questions that threaten that trust could be dangerous.

Unlike automobiles or houses, where independent technicians are widely available, technical support for voting systems is typically available only from the vendor. Therefore, unless the state or county decides to discard the voting system entirely, they are a captive customer of that vendor and must carefully protect their partnership with the vendor.

I believe that each of the above reasons, in isolation, are sufficient to explain the way that many states and counties have reacted to the news of security flaws in their voting systems. It

is quite natural to reject allegations of security flaws as mere hypotheticals. When taken together, it seems unlikely that election jurisdictions will take concerns about security seriously unless each security problem is clearly and convincingly demonstrated.

What We Must Do

The situation we face with the evalution of voting system security is not unlike the situation faced by a home buyer evaluating the safety of a house. Price is easy to evaluate. Architecture is subjective, but people tend to know what they like. Safety, on the other hand, is extremely difficult to assess. Most homeowners have never suffered from a serious safety defect in the design of a house. Major safety defects can be invisible, hiding in structural details that only an engineer can evaluate.

A century and a half ago, home safety was dealt with rather poorly. The free market, operating on its own, certainly did little to ensure that our homes were safe. This situation probably would not have changed very much if the question of safety were entirely between the homebuilder and homeowner.

What changed the picture in the housing market was the development of the insurance industry, with strong government involvement. The primary driving force came from fire insurance policy underwriters. It was the insurance underwriters who saw the big picture. They undertook detailed studies of what it was that caused house fires and what technical measures could be taken to reduce that risk. From this work, we have inherited two great institutions, the Underwriters Laboratory, which sets safety standards for a large number of products that go into peoples homes, and the various building codes that govern the way we build our houses.

Unfortunately, in the world of elections there does not seem to be any institution that serves the role that the insurance industry. Jurisdictions certainly cannot take out election fraud insurance policies. If we could, I am certain that the insurers would impose strict financial penalties on any jurisdiction that purchased insecure voting systems.

This leaves us little choice but to lean hard on the voting system vendors, on the federal government, and on the states, demanding effective voting system regulation. The Election Assistance Commission or their successor agency must have enforcement powers in the area of voting system security comparable to those of the Consumer Product Safety Commission with regard to product safety. This demand will almost certainly be more effective if we follow through on allegations of security vulnerabilities by actually demonstrating them.

There is a risk here. Currently, there are very few contexts where it is legal for a security researcher to demonstrate a security vulnerability on current voting systems. Where occasional rare election officials such as Ion Sancho in Leon County Florida or Bruce Funk of Emery County Utah have allowed security experts to examine systems, they have been subject to threats of lawsuits and they have faced significant disciplinary actions. We must create a legal way for security experts to demonstrate voting system vulnerabilities without fear of retribution.

References

[1] Tadayoshi Ohno, Adam Stubblefield, Aviel Rubin, Dan Wallach, Analysis of an Electronic Voting System, *IEEE Symposium on Security and Privacy 2004*. IEEE Computer Society Press, May 2004. http://avirubin.com/vote.pdf

[2] *Trusted Agent Report Diebold AccuVote-TS Voting System*, RABA Technologies LLC, January 20, 2004.
http://www.raba.com/press/TA_Report_AccuVote.pdf

[3] *Response to: Department of Legislative Services Trusted Agent Report on Diebold AccuVote-TS Voting System*, Maryland State Board of Elections, January 29, 2004.
http://mlis.state.md.us/Other/voting_system/sbe_response.pdf

[4] Harri Hursti, *Diebold TSx Evalutaion*, Black Box Voting, May 22, 2006.
http://www.blackboxvoting.org/BBVtsxstudy.pdf

[5] *2002 Voting System Standards*, Federal Election Commission.
http://www.eac.gov/election_resources/vss.html

[6] Barton Miller, Lars Fredriksen and Bryan So, An Empirical Study of the Reliability of Unix Utilities, *Communications of the ACM, 33*, 12 (December 1990) pages 32-44.
ftp://ftp.cs.wisc.edu/paradyn/technical_papers/fuzz.pdf

[7] David Wagner, David Jefferson. Matt Bishop, *Security Analysis of the Diebold AccuBasic Interpreter*, California Voting Systems Technology Assessment Advisory Board, February 14, 2006,
http://www.ss.ca.gov/elections/voting_systems/security_analysis_of_the_diebold_accubasic_interpreter.pdf

[8] C. C. Michael and Will Radosevich, *Black Box Security Testing Tools*, Cigital, Inc, December 28, 2005.
https://buildsecurityin.us-cert.gov/daisy/bsi/articles/tools/black-box/261.html

[9] Douglas Jones, *Voting System Transparency and Security: The need for standard models*, Testimony before the Election Assistance Commission Technical Guidelines Development Committee, September 20, 2004.
http://www.cs.uiowa.edu/~jones/voting/nist2004.shtml

[10] *Risk Assessment Report: Diebold Accuvote-TS Voting System and Processes*, SAIC-6099-2003-261, September 2, 2003.
http://truevotemd.org/content/view/194/61/

[11] *Direct Recording Electronic (DRE) Technical Security Assessment Report*, Compuware Corporation, November 21, 2003.
http://www.sos.state.oh.us/sos/hava/compuware112103.pdf

[12] Douglas Jones, *Confusion of Myth and Fact in Maryland*, July 19, 2004; this is an annotated copy of the pamphlet *Maryland's Better Way to Vote*, put out by the Maryland State Board of Elections in 2004.
http://www.cs.uiowa.edu/~jones/voting/myth-fact-md.html