Note: This testimony is based on the results of the Evaluation & Validation of Election-Related Equipment, Standards & Testing (EVEREST) report, commissioned by Ohio Secretary of State Jennifer Bruner to evaluate the security and reliability of electronic voting equipment. The full EVEREST report is available at http://micah.cis.upenn.edu/papers/everest-ohio.pdf

Testimony to West Virginia Joint Judiciary Subcommittee
by Micah Sherr, PhD, Postdoctoral Researcher at the University of Pennsylvania
msherr@cis.upenn.edu

Chairman Snyder, Chairwoman Brown, Members of the Subcommittee, it is an honor to join you here today. The election process is the cornerstone of our democracy. Ensuring reliable elections and increasing the public's confidence and trust in the election process are perhaps the two most important responsibilities of our federal, state, and local governments. As this committee considers how voters in West Virginia will cast their future ballots, it is my profound privilege to be here to testify today.

While completing my doctoral studies at the University of Pennsylvania, I participated in two statewide studies of electronic voting systems, the first on behalf of the State of California and the second on behalf of the State of Ohio.

CALIFORNIA

The California "Top-to-Bottom-Review" was initiated by California Secretary of State Debra Bowen in 2007 to evaluate the security, accuracy, reliability, and accessibility of the State's voting equipment. The Top-to-Bottom-Review was the first in-depth study in which the investigators had access to both election equipment and their source-code – the instructions and algorithms that determine how the machines operate.

The California study discovered significant security vulnerabilities in all tested touchscreen voting systems. (The ES&S equipment used in West Virginia was not evaluated during the course of the study.) The touchscreen voting systems that were analyzed were found to be susceptible to malicious software and viruses. A motivated individual with access to the voting equipment – for example, a poll-worker or other election official – could compromise ballot secrecy, resulting in inaccurate election results and audit information. Citing the results of the study, Secretary of State Bowen withdrew the approval of the tested touchscreen devices, permitting their use only if a 100% manual tally were conducted at the close of the election.

OHIO

Shortly after the California study, Ohio Secretary of State Jennifer Bruner commissioned the "Evaluation & Validation of Election-Related Equipment, Standards & Testing" (EVEREST) study. EVEREST's scope was similar to that of California's Top-to-Bottom-Review. The Ohio study investigated the security, reliability, and accuracy of the State's electronic voting machines, as configured for actual elections. EVEREST included researchers from academia as well as industry experts. A bipartisan committee of election board officials evaluated the final report and advised the Secretary of State in forming recommendations.

I participated in the academic team that evaluated voting equipment manufactured by ES&S. The software and hardware versions of the UNITY election management system and the iVotronic touchscreen DRE that were examined in the EVEREST study are the same versions that are currently in use in West Virginia. Although ES&S provides customized features for certain states and counties, all of the discovered security flaws were present in core components of the system.

It is therefore reasonable to infer that the vulnerabilities that we discovered in our study of Ohio's ES&S equipment are equally applicable to the ES&S hardware used here in West Virginia.

During the course of the study, we discovered serious security problems in all tested ES&S equipment, including the iVotronic touchscreen DRE, the M100 optical scanner, the M650 precinct batch scanner, and the UNITY backend election management software. *The results of our investigation suggest that ES&S equipment lacks the fundamental security mechanisms necessary to ensure a reliable and trustworthy election under operational conditions.*

We identified numerous exploitable vulnerabilities that allow a person with even limited access to the equipment (for example, a voter) to falsify election results, destroy audit records, and reprogram the devices to execute arbitrary instructions.

We also discovered undocumented functionality in the iVotronic touchscreen devices that enables a user with an easily obtainable device to bypass all password and security checks.

The ES&S equipment failed to protect election data, did not effectively control access to administrative election functions, incorrectly implemented security mechanisms, and did not follow standard software and security engineering practices.

Our final report classified the security failings of the ES&S voting systems as “severe and pervasive”. A separate study, conducted concurrently by a commercial laboratory, cited similar security flaws in ES&S systems and concluded that vulnerabilities “seem to stem from a lack of adoption of industry standard best practices.”

In response to the EVEREST reports, Ohio Secretary of State Bruner recommended that the use of DRE-based election equipment be eliminated at all polling locations.

With all this said, it should be noted that the EVEREST study was not intended to be a comprehensive audit of Ohio's electronic voting machine equipment. Modern voting machine systems are very complex. For example, the software that controls the behavior of the ES&S system is comprised of nearly 670,000 lines of programming instructions – the equivalent of 10,150 printed pages. Given the voluminous nature of the software and the number and variety of potential voting system attacks, an exhaustive exploration of the system was infeasible during the ten-week study.

We therefore adopted a methodology that focused on particular voting machine components that we believed would be most attractive to an attacker. For instance, we closely examined the functionalities that recorded voter choices, protected ballot secrecy, managed audit data, and processed voter and poll-worker inputs.

While our analysis was admittedly not comprehensive, it was nonetheless alarming that *we discovered serious security vulnerabilities in every tested ES&S component*.

Since ES&S equipment is currently used to conduct elections, the EVEREST report disclosed only high-level descriptions of the discovered security problems. The technological details that precisely specified how a malicious individual could compromise the equipment were isolated to a confidential appendix and furnished only to the Office of the Secretary of State and the voting machine vendors.

HIGH LEVEL FINDINGS FROM EVEREST

I would like to take the next few minutes and highlight some of the high level findings of our EVEREST study. It should be reiterated that since West Virginia and Ohio use the same hardware and software versions of ES&S components, the security problems identified during the EVEREST study apply equally to ES&S equipment in West Virginia.

Voters, poll-workers, and election officials interface with the iVotronic DRE through a touchscreen interface. To authorize different features of the voting equipment – for example, the ability to cast a ballot in the case of a voter, or to configure the machine in the case of an election worker – the user inserts a small computing device called a Personal Electronic Ballot, or PEB, into the front of the iVotronic. The PEB is sometimes also called the Ballot Activator Cartridge.

There are different PEB types used to access different features of the iVotronic. For example, one type of PEB is used to authorize a voter to cast a ballot, while another PEB type may be used to access administrator functions. In this latter case, an election official may insert the election worker PEB into the iVotronic to configure the machine. In theory, the election official must also correctly enter secret passwords before she is able to administer the machine.

However, *we discovered an undocumented feature that allows a user with a special type of PEB, called a "Debug PEB", to place the iVotronic into configuration mode and bypass all password security checks*. Using a Debug PEB, an unauthorized individual can access all administrator functions on the iVotronic, allowing him to cast multiple ballots, remove votes and audit data, delete all logs, reset passwords, and install malicious software onto the voting machine that alters the device's behavior (for example, by miscasting ballots).

It should be noted that this Debug PEB does not appear to be documented in any of the manuals or training materials that were provided to us by ES&S during the EVEREST study. In security parlance, we call such a hardwired bypass of security checks a "backdoor" into the system. Because the PEB interface slot is on the voter-facing side of the iVotronic and is not protected by any doors or security seals, **any person who possesses this special type of PEB, including a voter, effectively has complete control over the voting machine**. Bypassing security checks requires a few seconds. Clearing the terminal and removing all vote and audit data takes less than five minutes, well within the amount of time a voter may reasonably spend to complete the voting process.

Given that ES&S does not advertise the existence of Debug PEBs, one may assume that such an attack could only be conducted by a rogue ES&S employee. However, we found that different PEB types use the same underlying hardware and software. Any PEB can be reprogrammed to behave like a Debug PEB. In other words, anyone who has access to any PEB, in any county, state, or country where they are sold, may transform that PEB into a Debug PEB that bypasses the password checks that exist in West Virginia.

As we discovered during the EVEREST evaluation, access to a PEB is not required to bypass password checks in iVotronic equipment. PEBs communicate with the iVotronic using a magnet and an infrared

device – the same technology used in TV remote controls. Using a computer with an infrared interface, it is possible to emulate a Debug PEB. Our team constructed such a fake Debug PEB using a $100 Palm Pilot electronic organizer and a $2 magnet. By holding our electronic organizer near the iVotronic's PEB slot, we were able to bypass all password checks, cast multiple ballots, permanently wipe electronic election counts and audit information, and install modified software on the iVotronic.

This photograph illustrates the ease at which a voter could use an electronic organizer to bypass the security checks on the iVotronic. Since the PEB interface slot is directly in front of the voter, a malicious voter could easily position himself or herself to obscure the electronic organizer from view. This photograph was taken as we were using our fake Debug PEB to modify all of the passwords on the iVotronic, in this case, to the word "EVEREST". Although it is difficult to discern from the photo, the iVotronic touchscreen reads:

```
  Following the Initialization Process the iVotronic Terminal
will contain the
  following Passwords:
  [...]
  Election Central Menu Password: EVEREST
  Clear and Test Password: EVEREST
  Override Password: EVEREST
  Lock Unlock Password: EVEREST
  Upload Firmware Password: EVEREST
```

As an alternative means to compromising the security features of an iVotronic, an individual who has prolonged access to a PEB – for example, a poll-worker or other election official – can insert a program on the PEB that, when processed by an iVotronic, will install unauthorized software onto the iVotronic. Such software could destroy or corrupt electronic audit trails, cause legitimate votes to be discarded

and non-existent votes to be recorded, or cause the machine to cease functioning entirely. These vulnerabilities are due to the system's failure to verify that a PEB has not been altered by unauthorized parties. Flaws in the software running on the iVotronic assume that data on the PEB is trustworthy; when that assumption is violated, programming errors allow a program stored on the modified PEB to be executed on the iVotronic.

As with the previous described vulnerabilities, a malicious individual can surreptitiously compromise the voting machine during the election by inserting an altered PEB into the unprotected PEB interface in the front of the iVotronic.

iVotronics, both in Ohio and West Virginia, include a printer attachment to produce *Voter Verifiable Paper Audit Trails* or VVPATs. VVPAT printers are intended to record voters' selections on paper in human-understandable form for potential future manual audits or recounts. Unfortunately, iVotronic touchscreen DREs do not adequately protect the physical interface to the VVPAT printer, allowing anybody with access to an iVotronic to print arbitrary information – for example, nonexistent ballots – to the paper audit trail. The VVPAT printer attaches to the DRE through a printer port at the top of the iVotronic. The port is not protected by a seal, allowing anyone with access to the machine to easily disconnect the printer from the VVPAT and connect it instead to a small computing device – for example, a small netbook computer.

This photograph shows how a voter can detach the printer cable from the iVotronic without breaking any protective seals. The blue seal to the left of the printer port protects only an internal memory storage device.

By connecting our laptops to the unprotected VVPAT printer cable and running a free and easily obtainable software program, we were able to print text of our choosing to the VVPAT. Since the VVPAT printer uses a standard protocol, no special or proprietary printer drivers are required.

Many of the security mechanisms put in place by ES&S can be easily bypassed due to serious design and engineering flaws. For example, critical election data – that is, election configuration files and precinct vote tallies – are stored on PEBs and other electronic media. To protect such critical information, the data stored on PEBs are encrypted using a robust and well-studied cryptographic algorithm. The use of encryption prevents programs from reading the private contents of the PEB unless they have knowledge of a secret piece of information called a *key*.

An often-used analogy to encryption is a combination safe that stores confidential information. Without knowledge of the combination (that is, the "key"), the safe cannot be opened and the data cannot be accessed. Although the cryptographic algorithm is correctly implemented, the iVotronic, as well as the UNITY election management system that prepares the PEBs, stores the key alongside the encrypted data. This is exactly equivalent to shipping confidential information in an impenetrable safe, but enclosing the combination to the safe in the same package. If an election worker has access to the PEB, he can read the PEB to discover the key, and use that key to access (and, if he desires, modify) the critical information stored on the PEB.

In addition to the problems that I have just described, we discovered numerous other serious security flaws in the ES&S equipment during the EVEREST study. Unfortunately, there is insufficient time for me to enumerate all of the discovered vulnerabilities. I urge any interested party to refer to the full EVEREST report for more details.

It is worth emphasizing that even if ES&S corrects the software bugs that were identified in the EVEREST study, it is very likely that the iVotronic will still have several significant security weaknesses. The iVotronic lacks basic security features that should be present in any secure system. In particular, the iVotronic fails to guard encryption keys and does not authenticate data that are transferred through exposed interfaces. Revising the source code to correct such fundamental problems would be a significant undertaking, requiring a substantial re-engineering effort of both the software and hardware components that comprise the ES&S systems.

Given the ease at which we identified software and hardware bugs and the pervasiveness of the security failures, *it is my opinion that there are likely many additional security flaws in the ES&S equipment and software that were not discovered during the EVEREST study*. This belief is supported by the following findings in the EVEREST evaluation of the ES&S system.

First, the ES&S system is extraordinarily complex, consisting of nearly 670,000 lines of source code written in twelve programming languages for five different hardware platforms. Given the size and breadth of the source code, it is unlikely that any internal Quality Assurance process could conduct a comprehensive analysis of the entire ES&S system. During the ten week EVEREST study, we examined only a small fraction of the source code. A programming error in any segment of source code could lead to system-wide security vulnerabilities that may be exploited by an election worker or voter.

Second, the type of discovered security bugs strongly suggests that ES&S did not perform an adequate level of code analysis. It is customary for complex software systems – particularly those that operate in security-critical domains – to undergo rigorous internal quality assurance testing. Commercially available software called "static analysis tools" automates some of this process by scanning the source code for bugs and potential security vulnerabilities. Such software is imperfect and should not replace

careful human inspection of program code. However, these tools are very useful in helping software manufactures catch potential weaknesses before the product is released or sold.

Unfortunately, it appears that such tools were not adequately used by ES&S. Using Microsoft's C Compiler, a tool used to convert source code to machine instructions, we discovered several serious programming errors. Of particular concern, the source code was inundated with what are known as "unsafe" programming features. The use of these unsafe practices enables serious security vulnerabilities that allow a malicious user to gain complete control over the voting equipment. The use of these unsafe programming techniques is strongly discouraged in undergraduate computer science curricula.

Using another tool, hundreds of potentially exploitable software bugs were immediately exposed. Unfortunately, we did not have adequate time during the EVEREST study to investigate the degree to which the identified programming errors led to exploitable security vulnerabilities. However, the discovery of so many programming mistakes implies that serious undiscovered security and reliability problems likely do exist, and arguably equally disconcerting, indicate that ES&S did not sufficiently validate their code.

ATTACK SCENARIOS

Given the vulnerabilities that I have previously described, I would like to briefly present plausible attack scenarios that may be carried out against the iVotronic touchscreen DRE. It should be emphasized that these are not merely academic attacks that may or may not be practically achieved. We were able to successfully carry out each of these attacks during the EVEREST study.

Scenario 1: Changing an Voter's Vote

In this scenario, a malicious poll-worker exploits programming flaws in the iVotronic to load unauthorized software onto the touchscreen DRE. As previously discussed, such software can be loaded by modifying the contents of a PEB.

Our modified software took the following actions. The compromised iVotronic allows the voter to select her choices, with her selections being correctly displayed on the touchscreen and printer. Additionally, the confirmation screen displays the voter's chosen selections and the correct choices are printed on the VVPAT (the paper audit trail).

However, after the voter presses the "Cast Ballot" button on the confirmation screen, the malicious software modifies the electronic ballot – the ballot stored internally in the iVotronic's memory – and negates the voter's intent. The VVPAT printer then quickly prints a message to the paper audit trail, canceling the voter's selection and instead forging the selection of a different choice. This printing process takes less than 4 seconds and can occur with the paper rapidly scrolling up and down, making it very difficult for even a perceptive voter to read what is printed on the paper audit trail. The audit log is then immediately scrolled out of view – the standard procedure carried out on iVotronics to protect the privacy of the selections from the subsequent voter.

As a result of this attack, both the electronic record and the paper audit trail show selections chosen not by the voter, but rather by a compromised iVotronic.

Scenario 2: Disabling iVotronic Equipment

In the second scenario, a malicious poll-worker wishes to disrupt the voting process by causing repeated failures of the iVotronic equipment. Instead of modifying data on the PEB, in this attack the poll-worker instead corrupts a clipart file stored on a CompactFlash card that is inserted into a slot on the top of the iVotronic. When the iVotronic loads the clipart file – for example, to show the logo for a particular political party – a flaw in the system's software causes the iVotronic to crash. Although the voting process can be restarted by rebooting the iVotronic, the machine will again crash whenever it attempts to display the corrupted graphics file.

Unlike all previously described attacks, this scenario requires the malicious poll-worker to break a tamper-evident security seal on the iVotronic. Since the EVEREST study did not have access to the seals used in West Virginia, I cannot conclusively comment on the security of the protective seals used in this state. However, other studies have shown that all but the most sophisticated commercially available seals are often surprisingly easy to defeat.

It should also be noted that the use of such seals do not *prevent* tampering; at best, they *detect* tampering. Even perfect tamper-evident seals may be unsatisfying in an election. If a seal is discovered to be broken once polling has started, it is unclear what should be done. If the compromised equipment is used, the election results may be fraudulent. If the equipment is not used, previously cast legitimate votes may be lost, in which case breaking a seal is a simple way for any voter to destroy votes.

Scenario 3: Compromising the Entire Election Process with a Virus

The ES&S system consists of various components: the backend UNITY election management system, iVotronic touchscreen DREs, and optical scanners. By design, these components are heavily interconnected: UNITY is used both to load ballot and election information onto the touchscreen DRE and optical scanning equipment, and also to load data from these devices at the close of elections in order to tabulate election results. To carry data back and forth between election equipment, ES&S uses memory devices such as PEBs and CompactFlash cards.

Due to software bugs in each component of the ES&S system and the manner in which data is exchanged between devices, we discovered that it is possible to propagate an attack from one infected piece of election equipment to another.

During the EVEREST study, we conducted a viral attack in which a modified PEB was able to corrupt the iVotronic terminals in a polling location and spread the infection to the UNITY software in the election office. Once UNITY had been compromised, we were able to propagate the attack further, infecting any PEBs that were subsequently programmed using UNITY as well as election definitions prepared by UNITY for the optical scanners. In other words, by modifying a single PEB, we were able to compromise the iVotronic terminals, the UNITY backend system, and the optical scanners.

The attack works as follows. A malicious voter inserts a PEB (or an emulated PEB using an electronic organizer) into an iVotronic terminal. As described earlier, bugs in the iVotronic software allow a specially crafted PEB to install unauthorized software on the iVotronic. This iVotronic becomes infected and can spread the virus to any device that is connected to it. In particular, when a well-intentioned poll-worker inserts his PEB into the iVotronic, the infected iVotronic will copy the virus onto the inserted PEB. Since at the close of the election the poll-worker will use the same tainted PEB

to collect votes from the other iVotronics in the polling location, he will inadvertently spread the virus to multiple DREs.

The normal post-election procedure is to load the PEBs from the polling locations into UNITY so that the backend software may tabulate the results of the election. As with the iVotronic, UNITY contains numerous programming errors that permit corrupted PEBs to surreptitiously install and execute unauthorized programs. Hence, the infected PEBs used to convey voting data will take control of the UNITY backend system. At this point, the iVotronics in the polling location, the PEBs used to access those iVotronics, and the UNITY backend system are all infected with a virus that is able to execute malicious software.

Unfortunately, this viral attack gets worse. A corrupted UNITY backend system is able to further propagate the attack. Since UNITY prepares the memory cards that are used by the optical scanners, a malicious program running alongside UNITY can install malicious software onto these memory cards. The optical scanners contain no security features to validate the memory cards, and will themselves become infected.

Finally, if the same infected computer that runs UNITY is used to prepare iVotronics and optical scanners for subsequent elections, the compromised election management system will infect iVotronics and optical scanners in later elections.

To clear the infection, the computer running UNITY along with the iVotronics, the optical scanners, and the devices used to convey data (that is, the PEBs and memory cards) would all have to be erased. Software would need to be reinstalled on all equipment and devices. The omission of a single infected PEB would cause the virus to again replicate and spread during the course of the next election to the other ES&S components.

CRITIQUES OF THE E-VOTING REPORTS

Those who promote e-voting systems sometimes criticize the Top-to-Bottom-Review and the EVEREST report by suggesting that our methodology was flawed. In particular, they point out that the studies' participants had access to the source code or program code that controls the operation of the voting machines. They reason that it is unlikely that individuals intent on disrupting elections would have such unfettered access. While it is true that knowledge of the systems' source code increased the speed at which we were able to discover security vulnerabilities, it is certainly not the case that security can be achieved by hiding program code from potential attackers.

To develop voting machine software, a programmer writes a computer program using a programming language. The text of this program is called the source code. To get the source code to run on the voting machine, the source code must first be transformed into what is called *machine instructions*. The machine instructions are loaded onto the voting machine and control the machine's entire operation.

An individual who has prolonged access to any iVotronic may probe the machine's memory to gain access to the machine instructions. Although it is difficult to convert these machine instructions back into source code, *it is not difficult to discover security flaws using only the machine instructions*. In fact, there are a number of commercially available software programs that automate this process. Such techniques are regularly used to find security bugs in software for which the source code is not

available. Vulnerabilities in "closed-source" applications such as Microsoft Windows, Microsoft Office, and Adobe Acrobat are regularly discovered using such approaches. Simply put, hiding source code does not represent a serious barrier to parties that are motivated to find security flaws.

Similarly, it may be pointed out by proponents of touchscreen voting systems that the investigators in the two academic studies had prolonged access to the voting machine hardware, therefore reflecting an unrealistic advantage that would not be present for actual attackers.

However, access to the hardware was not required to develop many of the discovered attacks. In fact, during the California Top-to-Bottom-Review, our team did not have direct access to the voting equipment and discovered nearly all of the vulnerabilities by examining only the programming code.

It should also be pointed out that malicious individuals could easily gain access to the voting equipment. For instance, many states (although not West Virginia) permit "sleepovers" in which a poll-worker picks up the voting equipment from a central election office before the election and stores the equipment at their home until election day. A rogue poll-worker has prolonged access to the voting equipment, and can use this opportunity to download the machine instructions stored on the iVotronic to learn exactly how the software operates. The poll-worker could then disseminate the machine instructions to a team of attackers in various locations throughout the world who could then refine their attack. Again, the compromise of any iVotronic in any county, state, or country that runs the same software as in West Virginia could be used to develop attacks that target this state's election equipment.

One of the more common responses to our reports from touchscreen voting advocates is that the studies are merely academic and describe theoretical but impractical vulnerabilities. However, this simply is not the case. We were able to design, develop, and carry out attacks that compromised all tested ES&S systems as configured for actual elections.

Moreover, the types of security flaws in ES&S equipment were not particularly novel. In most instances, the discovered vulnerabilities were due to well-understood attacks. Software bugs similar to those uncovered during EVEREST are routinely found by industry experts when examining more traditional computer applications. What was surprising, however, was the degree to which these problems persisted in such a security-critical system.

IVOTRONIC "SECURITY FEATURES"

The iVotronic is advertised as having a number of security mechanisms that make the system "accurate, reliable, and secure". Unfortunately, these mechanisms fail to deliver an acceptable level of security.

For example, the iVotronic product literature promotes the fact that the device uses "three independent but redundant memory paths". Using redundant memory is a good design decision, and is one that is implemented in most DRE systems with which I am familiar. The redundant memory ensures that if one of three memory storage areas in the DRE malfunctions, votes are not lost since the remaining two functional memories can be used as backups. In essence, the redundant storage is a reliability feature.

It is **not**, however, a security feature. Reliability and security are related, but they are not synonymous. An iVotronic that has been corrupted using any of the many methods that I have previously described will happily store incorrect data to all three of the memory storage areas. This misinformation will be

consistent among all three memory storage devices. In other words, *the wrong results will be stored with high reliability.*

iVotronic marketing literature implies that the DREs are more secure because they use proprietary Personal Electronic Ballots, or PEBs. The argument that the proprietary nature of PEBs makes them somehow difficult to misuse is unfortunately false. A malicious individual with access to a PEB in any county, state, or country can modify his or her PEB and configure it to exploit iVotronic software vulnerabilities in West Virginia. Moreover, we have demonstrated in the EVEREST study that it is fairly simple to trick an iVotronic into believing it is interfacing with a PEB when it is in fact communicating with an electronic organizer.

Another advertised feature of the iVotronic DREs is that they are "self-contained election systems" in which the "malfunction of one machine does not affect others". However, the ES&S system is actually composed of highly integrated voting equipment. iVotronics interface with other iVotronics in the polling location by communicating data using a PEB. After poll closing, PEBs and CompactFlash cards stored in the iVotronic are collected and transferred to the UNITY backend system. These interactions between systems provide opportunities for viruses to spread between voting equipment. As I previously described, it is possible for a single modified PEB to cause unauthorized software to be installed on multiple DREs, the UNITY backend system, and M100 optical scanners.

Finally, it has been pointed out by ES&S and other proponents of touchscreen DRE systems that major security vulnerabilities were not discovered when these systems were certified for use in elections. In particular, they often point to evaluations by Independent Testing Authorities (or ITAs) that did not uncover significant security weaknesses.

However, ITA testing procedures are not necessarily designed to evaluate the security of a system. Rather, they examine voting hardware and software to determine whether the voting equipment meets a fixed set of standards. For example, ITA reports describe whether a particular piece of source code has the proper formatting and uses consistent programming styles. The reports indicate whether the tested voting system conformed to a checklist of requirements. My remarks are not intended to diminish this process in any way. Requiring voting machine manufacturers to abide by fixed guidelines likely decreases the prevalence of certain types of programming errors.

In contrast to the ITA reports, the California and Ohio studies took a radically different approach. Rather than follow a particular methodology, we instead adopted an approach in which we focused our attention on components of the hardware and software that we believed would be most valuable to a person intent on disrupting an election. That is, we purposefully sought out vulnerabilities in the software.

By not following specific guidelines or checking off whether the *syntax* of the programming code met certain government standards, we were able to concentrate our attention on evaluating the *semantics* of the code – that is, what does the program actually do, what assumptions is it making, and what happens if those assumptions are broken?

Using this targeted and somewhat *ad hoc* approach, we were able to quickly identify security flaws in all tested ES&S equipment.

CONCLUDING REMARKS

To conclude, by exploiting software bugs and poorly designed hardware interfaces, we were able to install unauthorized software on the iVotronic touchscreen DREs, the optical scanners, and the UNITY backend election management system. Many of our discovered attacks could be carried out by a voter during election day in a matter of minutes. We identified an undisclosed backdoor into the system that allowed anybody with a modified PEB or an inexpensive electronic organizer to gain access to all administrative functions on the iVotronic. Multiple security flaws enabled malicious election workers and voters to insert or remove votes, modify tabulation results, write arbitrary information to the paper audit trails, erase electronically stored audit information, access the contents of confidential election information, and install unauthorized software onto the voting equipment. These attacks are not theoretical. They have all been successfully implemented and tested during the EVEREST study on the same equipment that is in use in West Virginia.

Developing secure software is an exceptionally difficult art. A single typo in a million line program could enable an attacker to gain complete control over a system. Although new techniques, tools, and best-practices are continually developed, refined, and enhanced to help programmers write more secure code, the human element in software production guarantees that mistakes will always be made. And incentives to exploit these errors ensure that software will always be vulnerable to some degree of manipulation.

West Virginia is in the process of debating whether touchscreen DREs should be used in elections. *It is my strong opinion that the pervasive security flaws present in ES&S systems, coupled with the unfortunate truism that computer science has not adequately advanced to a point at which we can write secure software, warrant the exclusion of touchscreen DREs in the election process.*

I do not believe, however, that optical scanners are immune from manipulation. They are, like touchscreen DREs, merely computers dressed in specially tailored packages. They are susceptible to the same flaws and limitations of other computing devices.

However, unlike touchscreen devices, optical scanners have the inherent advantage that they leave an accurate trail of voter intent. Voters, not machines, mark their choices on paper ballots. This is the principle difference between DREs and optical scanners: DREs interpret human intent and store their rendering of voter choices on electronic media that may be modified and printer transcripts that may become compromised. In contrast, optical scanners are ballot boxes with wires. If you remove the electronics from optical scanners, intent can always be reconstructed by examining the same media in which the voter marked his or her intentions. Regardless of malfunction or compromise, paper ballots can always be processed by human examiners.

I would like to thank the committee again for the opportunity to testify here today. Securing and protecting the election process is an important and difficult responsibility, and one that is critical to our democracy. Thank you.