Voting Systems Security

G.D. Ritter and Nick Tiller

February 8, 2012

On February 8, 2012, Getty and Nick presented on the security of electronic voting systems. The content was based on a 2008 paper by Butler et. al. entitled Systemic Issues in the Hart InterCivic and Premier Voting Systems: Reflections Following Project EVEREST.

Background and Architecture of Hart - Getty

The study in question was slowed down by the fact that there were hundreds of thousands of lines of code per system. Getty gives a detailed description of how the current system works in practice, and some of the glaring problems that exist in this system.

Existing Weaknesses - Getty

The system consists of disparate pieces that use third-party software, some of which has known vulnerabilities, such as the cryptographic USB devices. the eScan configuration file can be pushed via ethernet, with simple changes to the configuration allowing multiple ballots to be cast - this is scary.

Susceptibility to soft key presses, with an inability to distinguish between these and actual keypresses. Dependence on windows reveals vulnerability through the windows registry and existing debug or testing modes that still exist in the source. Many oversights that make the system vulnerable to insiders.

VVPAT is a paper audit trail that is meant to reflect votes that occurred. Not only is it easily tampered with, the port by which it connects reflects a huge vulnerability in that it allows any information in the system to be printed out on paper.

The main take-away: There is a laundry list of vulnerabilities here, almost too many to list, and there are almost certainly more than were detailed in the publication.
Background and Architecture of Premier - Nick

There is a supervisor card to start and stop elections, and then there are cards given to voters that are meant to allow a single vote. The claim is that this is not the case due to poor attention to security in the software.

After a previous study exposed vulnerabilities, some measures were added to the Diebold/Premier system. We focus on the Verdasys Digital Guardian. How can we get around it?

Existing Weaknesses - Nick

A misconfiguration causes the GEMSUser to be in the Windows Administrator group. Also, apparently, the Nero CD burning software can rename GEMS database files.

Software update vulnerabilities also exist. Inserting a memory card causes the system to look for certain files and use them to perform an update, but there is not a system to prevent this from happening in a malicious manner.

The Digital Guardian software keeps a whitelist of allowed software to be run, but this checks simply the name of the software. So by renaming malicious software with a name that exists on the whitelist, this simple block is circumvented.

The logging software used by ExpressPoll is a plain, unsecured XML file. Digital Guardian warns that an attempt at malicious access will be logged, similar to trying a 'sudo' command on a UNIX machine. The difference is that Digital Guardian doesn’t actually do anything, despite the warning.

The take-away here is similar - full of crazy security holes. This is apparently a theme with voting machines.

Web-Based Voting in Washington D.C. - Getty

A web-based voting system was introduced in D.C. in 2010. The system was open-source, and the system was initially praised for its transparency. A public-private key system was used to encrypt ballots.

The exploit that turned up here was due to the implementation of the system using Ruby on Rails. A system was devised to save arbitrary shell scripts to the remote machine with malicious intent. Also gave the attackers the same permissions as a web server, which led to many other exploits.
They were also able to successfully attack the Network infrastructure using more or less standard network tools and password crackers. They wreaked all kinds of havoc in this way, including watching the server administrators freak out using their own cameras. Hilarious.

Even though this system seemed to be much better designed, it was still pretty easy to tear apart.

**Questions**

**Ben:** You have mentioned that the logs were able to be altered. Were these logs ever actually used to confirm the votes in reality?

**Getty:** Yes, there are instances in which the logs were used to figure out what actually happened.

**Nick:** When glaring problems would arise, such as 50,000 votes coming from a county of 5,000 people, the logs would be used.

**Paul:** It seems like when making a system like this it makes sense to build from the ground up with security in mind.

**Getty:** Some of the holes were a result of the development process, like debug functions that were still hanging around.

**Ben:** So what is the moral of this story?

**Getty:** (obscure reference to Greek history, or something, and the fleeting nature of victory). The degree to which such a sensitive application has to be bulletproof is basically insanity. There are almost always exploits, especially when you’re talking about physical access.