5
Cyber-Boss Tweed
21st Century Ballot-Tampering Techniques

With old-style voting systems, for the most part, no special training was needed to realize something was amiss. Not so with rigging computers, but many public officials don't understand this.

"Subverting elections would be extremely unlikely and staggeringly difficult," said Georgia Secretary of State Cathy Cox when interviewed about Georgia's touch-screen voting system. "It would take a conspiracy beyond belief of all these different poll workers. … I don't see how this could happen in the real world." [1]

My premise, though, is this: An insider, someone with access, can plant malicious computer code without getting caught. Just as we know that banks will have robbers, that blackjack tables will have card-counters and that embezzlers will slip in amongst the bean-counters, so we should expect to find a few ethically challenged individuals among the honorable programmers and technicians who work with our voting machines.

Certainly, human nature did not change just because we entered the age of computers. Sooner or later, someone's going to try to steal votes on these things.

What kind of cheaters are we looking for?

Candidates may not be the most likely people to cheat. Few candidates are likely to possess the combination of motive and cash to rig their own election. I believe that vested interests behind the

candidate are more likely suspects, and the candidate need not even know.

Zealots are a bigger danger, especially if they happen to be connected to people with giant wallets. "True believers" may feel that the end justifies any means. Some are very wealthy, and some congregate in radical groups where they can pool their cash and push their agenda. Zealots of any kind may believe they are "helping" the rest of us by imposing their candidates on us. You do not need to hand a zealot a bribe, and the candidate they select never needs to know his election was rigged.

Gambling interests may not be squeamish about pulling strings. Gambling rights have turned into a brawl, with some tough players who are seeking riverboat gambling rights, the right to compete with Native American casinos and just plain liberalized and legalized gambling in communities all over the world.

Hackers, more accurately called "crackers," get their kicks by compromising legitimate software systems. These people may not need bribe money or a cause; like climbing a mountain, they just want to see if they can do it.

Profiteers can make billions by putting the right candidate into office. Electronic voting systems give a small number of people access to a great number of votes. If you control the counting software, ballot-tampering on a massive scale is possible. We should expect this to attract the all-star players.

In the old days, a city boss might want a particular candidate to win, perhaps throw a few construction contracts his way, take a kickback. But high-volume tampering provides a motive for a different clientele.

Defense contractors stand to make billions with the right candidate. Oil companies benefit from new pipelines all over the world, if they select candidates likely to vote for open exploration and geopolitically strategic development. Highway contractors garner hundreds of millions on freeway and bridge projects. Global financiers gain power and profit when international trade policies are set up to favor their interests. Pharmaceutical companies want legislative protection for pricing policies and product patenting and protection from international competition. Investment holding companies stand to gain control over privatized retirement and pension funds.

* * * * *

So much to spend, so few techies to corrupt. Where to begin?

Well, for starters, you could send your own compromised programmer into a voting machine company toting a resume. But suppose I am a political operative for a wealthy and powerful, but crooked, corporation, and I just want to buy off an employee. How would I find and contact an employee, and how would I know whom to approach?

I set out to answer that question. I figured that if a middle-aged woman like me who has never done a "covert op" in her life, working on the Internet, could find the people who program our voting machines, then certainly the bad guys must know who they are.

You can find software engineers who once worked for voting machine companies by looking at online resumes and job-search sites. The resumes often have home phone numbers. You can call them up, say you are writing an article and ask them how a machine can be rigged. And they will tell you. I know this because I did it.

You will find software engineers who currently work for voting machine companies by finding any company e-mail address. ES&S employees have e-mail addresses that end in "essvote.com." Enter "essvote" in a search engine, and you'll find people who submitted information to high-school reunion sites and programmers who post comments on forums, join listservs, create personal Web pages and post their wedding plans on the Internet. One guy even listed his hobbies and his favorite vacation spots.

I located eight dozen voting-company employees this way. I also found the home phone number for someone in human resources at ES&S, who in turn has access to contact information, including the home phone number, for every single employee. This took three hours.

How would you choose someone to approach?

For $80 you can run a background check. That will give you a person's Social Security number, which opens up more information. You can also run a credit check. Doing this, you find out if the programmer has a gambling problem, has gotten into credit-card debt, is over her head in student loans, has had run-ins with the law, likes fancy cars, is overcommitted on a mortgage. Additional searches reveal political affiliations and even lead you to people who are disgruntled or believe they will soon be fired.

How to compromise an Internet voting system

Some cities, like Manatowoc, Wisconsin, and Liverpool, England, are eager to vote by Internet. Among computer professionals, however, Internet voting advocates are difficult to find. Here's why:

Companies like VoteHere claim that encryption techniques are a key to Internet voting security, but encryption won't protect our vote from software programming errors.

Rigging an Internet election is as simple as "DoS"-ing a server. Denial of Service attacks can knock out servers in targeted areas, and no amount of encryption will help. (Let's take the technospeak out: Suppose you connect to the Internet using America Online, but on election day, for some reason, your AOL access numbers don't work. Can you vote on the Internet?)

A company that specializes in Internet voting, election.com, ran a January 2003 contest in Toronto, Canada, which was disrupted by a malicious attempt to shut down the computer system.

"Earl Hurd of election.com said he believes someone used a 'denial of service' program to disrupt the voting — paralysing the central computer by bombarding it with a stream of data," CBC News reported. "'We had one log-in attempt that corrupted the ability of everybody to get access to our servers,' he said ... When asked if a second ballot might be delayed by another act of computer vandalism, election.com conceded that the culprit might strike again.

"'Unless he died in the last few minutes because of the evil thoughts in my brain, he or she is still out there,' Hurd said." [2]

Even the most elaborate encryption can't solve a power outage. If some clown with a backhoe pulls the phone cables up out of the ground, how will you vote? If an ice storm takes out power in the city, will your modem work? If you forget to pay your cable bill and they turn it off on Election Day, what will you do?

If you can vote from the privacy of your home, you can sell that vote as well. Proof of how you voted would be as close as your printer.

And while we're talking about privacy, what if you neglect to put in the latest Microsoft patch? You know, the one that says "*A security issue has been identified that could allow an attacker to compromise a computer running Windows XP and gain control over it*."

Heck, if there is as much "spyware" out there as my spam claims,

Internet voting would mean big trouble. From what I can tell, a lot of people don't trust the privacy of their computer even when they are not doing something mission-critical, like casting a vote. Even if scientists make a safe system, how do we get everyone to trust it?

You might find other people voting for you. Read up on identity theft, which is getting worse every year. [3]

Dirty tricks will proliferate. Your elderly Aunt Martha may get convincing messages that send her to bogus voting sites which dispose of her vote. Come to think about it, beloved Aunt Martha is eighty-three years old. Learning to vote on the Internet might stress her out, and why should she have to?

Do you want to vote with your spouse looking over your shoulder? Many of us connect to the Internet at work: Do you really want to cast your vote next to your union leader or your boss?

And what about "technical difficulties?" You cast your vote and your computer screen turns blue and a message appears:

Iexplorer.exe has caused a general protection fault in vote.exe. Your system may be unstable. Save all your work, close all windows and reboot your system.

Ookay. Did your vote go through? How will you know?

If it didn't, will you be able to vote again? If you do and the same thing happens, then what? Where will we find enough people to staff the tech support desks on Election day? Will we farm the job out to a service company in Bombay? And if so, how secure is that?

People are out there pushing Internet voting, but this concept is flawed and cannot be repaired. Any money we would save closing down the polls would be lost trying to make the system secure and reliable, and new laws would have to be passed to deal with each problem that arises. People and agencies would have to be appointed to enforce those laws. Election law would come to resemble the tax code in complexity.

Bottom line? Voting for your favorite movie online may be cool, but it's no way to run the Republic.

How to compromise an optical-scan system

Optical-scan systems involve filling in an oval or drawing an arrow on a paper ballot, which then is fed into a scanner. People think

these systems can't be rigged because they have a paper ballot, but there are anecdotal reports of optical scan systems flipping elections as far back as 1980.

An election official I spoke with from California reported that in her county, Jimmy Carter soundly defeated Ronald Reagan during the 1980 presidential election. However, the computer tally from the optical scanner reversed the results, giving Carter's votes to Reagan and vice versa. By doing a hand audit using the paper ballots, they were able to straighten out the results, but when she requested that the state of California do more audits to see how widespread the problem was, she was ignored.

Most people believe that optical-scan machines are tamper-proof because they provide a paper ballot. But election officials generally don't use the ballots to check the machine count, and in some states it's against the law to do so. If you don't audit properly, optical-scan machines are no safer than paperless touch screens.

Some people think that all we need to do is vote absentee and the touch-screen problem is solved. Unfortunately it will not be solved until we actually look at those ballots. When you vote absentee, your ballot is usually run through an optical-scan machine. Hack either the scanner or the main accumulation and you take the election away, while ballots sit forlorn in a box that no one is allowed to open.

The official results come from the county, not the polling place, so if you adjust the optical scan data before it gets into the county accumulator, you've just rigged the election. No one's going to look at those paper ballots, but if they do a spot check, see below. I'll show you how a crooked programmer can create a safety net for spot checks.

The greatest danger is during the transfer of the vote from the polling place to a central counting facility. Optical-scan votes are vulnerable when transferred by modem or, by *cell phone*, as happened in Marin County, California, during the recall election on Oct. 7, 2003. [4]

Another way to compromise an optical-scan system is to attack the program that accumulates the votes from the polling place.

One way to do this would be to enable a double set of books. If the software keeps a duplicate set of records and uses the first set for the totals, and the second set for the real numbers, you can rig

the totals but keep the detail intact in case of spot checks.

With our current lack of auditing controls, anyone with access to the central count machine can hack an election, and this access may be available through telephone lines or Internet connections, allowing complete strangers to tamper. One way to deter this tampering, or detect it, is to audit the paper ballots against the totals.

More ways to compromise an electronic voting system

Hiding functions in software programs is called putting in "back doors." Visit any computer forum on the Internet, and you'll find that programmers can think up back doors faster than anyone can figure out how to test for them. I spoke with sources who had worked for voting-machine companies and who came up with one method after the next. Here are some of their ideas:

Create a program that checks the computer's date and time function, activating when the election is scheduled to begin, doing its work, and then self-destructing when the election is over. It is possible to write hit-and-run code that changes the *original votes*, then destroys itself. It can pass testing because it activates only on election day.

Create a dummy ballot using a special configuration of "votes" that launches a program when put through the machine. Quite diabolical, actually: You rig the election by casting a vote! You could extend this to all machines using the same software by embedding the program in the "ender card," which is run through some systems to close the election.

Create a replacement set of votes, embed them on a chip, and arrange for someone with access to substitute the chip after the election. Chip replacement took place in the 2002 general election in Scurry County, Texas. Another chip replacement was done in 2002, also by ES&S, in South Dakota, where technicians discovered a machine double-counting Republican votes.

Overwrite the approved program with new commands by installing upgrades or "patches" that have not been examined. I asked Paul Miller, an official from the Washington State Secretary of State's election division, about procedures for updates. He told me that tracking and examining program updates is "not an issue." *But any time a*

program is changed, it can change things you don't see.

Include a layer of software that is insulated from certification testing. Diebold voting machines use Microsoft Windows, but when examining the code, no one looked at the Windows files. By embedding malicious programs in the Microsoft operating system instead of the voting software, a hacker can skip right through certification. Some Diebold machines run old versions of Microsoft operating systems, such as Windows 95 and Windows 98, which are not recommended, even by Microsoft, for use in security-sensitive applications.

Work with an unscrupulous vendor for your components. Manufacturers are not required to disclose who their vendors are. Some companies reportedly use components from Russia or the Philippines. Others share components from vendors in the USA who are not scrutinized by independent testing authorities.

Find a video-game programmer to tamper with the video driver. Because so many people create video games, the source codes are fairly readily available. A good game programmer can make the screen do one thing while the innards do something else.

Exchange files with support techs by putting them on a server. Anyone who gains access to the server can replace one with another — for example, replacing the central counting program with a file of the same name that contains a variation of the program.

Add a field into the program that attaches a multiplier to each vote, based on party affiliation, rounding one party slightly up and the other slightly down, using a decimal so that when votes are printed one by one (which is almost never done), they round off and print correctly, but when tallied, the total is shaved. For example: "Affiliation = Democrat; multiplier = 0.95 ... Affiliation = Republican; multiplier = 1.05." This will create totals that correlate with demographics.

Buy a tech and plant him as a poll worker in a key precinct where your competitor's machines are used. Have him go through the training and then have him flub the election by preventing machines from booting up, or causing them to crash and then blaming it on the manufacturer. If things really get messed up, have him call the press and grant interviews.

Using wireless technology embedded in the voting machine, monitor the election results on a remote basis as the contest proceeds and send your adjustment in when the election nears its end.

Put a back door into the compiler used for the source code (a compiler is used to "compile" software code from a high-level programming language into faster machine language). The source code can be clean, but no one looks at the compiler, and with this method, the digital signature (a method for detecting changes in software after certification) will remain intact.

Switch the card used to start up the machine. For some models, this overwrites the voting program with a new one. In Palm Beach County, Florida, in a March 2003 election, some precincts reported problems with electronic cards used to activate touch-screen machines, but according to the news reports, "backup cards worked." [5]

Compromise the binary code, below the level of the source code, which will not be detectable even with a line-by-line examination of the source code and won't be solved by using a digital signature.

By the way, people who have worked around touch screens know that rubbing them can screw them up big time. And almost everyone who works on computers knows that strong magnets and magnetic storage don't mix.

Accidentally put a few bugs in the software. Software engineering is like writing music or creating a painting. It is inspired, sometimes in the middle of the night, and in the wee hours things slip past the best of them. Sometimes engineers just don't catch bugs in the code. Or perhaps, a programmer plays with bugs for a hobby…

Bugs in the Code

Voting-machine source code apparently has turned into the digital equivalent of "The Blob," with such massive code, around a million lines long, that no one really catches all the bugs.

With such bulbous source code, who would notice a few *malicious* lines that can be explained away? Just a bug. A glitch. Remember, it's easy and fun to vote on these machines.

Following are examples of actual voting-machine software bugs.

Found on Internet voting source code, called votation

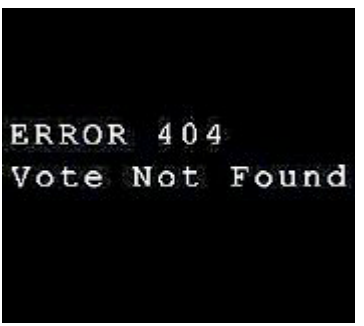
// really no idea on how to resolve rollback failure. :(perhaps praying :) //

Found these comments in Diebold source code files:

- Fix bug in VIBS causing Straight Party races not to work properly.

Diebold bugs, continued:

- Fix problem with race stats results not being sent correctly.
- Fixed bug in BallotDLG when ballot with the votes appears after touching Start button or anywhere else on the screen couple of times.
- Revert improvement in detection of invalid smart cards
- Fixed minor bug when internal keyboard did not work properly.
- Fix problem with transfer sending wrong precinct id
- Fix problem with not closing election after setting for election.
- Fixed problem that caused an error when view ballot results.
- Fixed problem in FileUtil that did not correctly determine if path was empty.
- Fixed problem in PollBook for Closed Primary Elections.
- Work around problem reporting zero totals when runing [sic] on Win95 units and Win98 units upgraded from Win95
- Fix bug with starting PollBook when main and def. Directories do not match.
- Fix bug uploading candidate totals
- Fixed problem in Poll Book where it fails to clear totals.
- Fixed bug that did not accumulate write-in votes.
- Handle failure of some files during upload.
- Fix bug in validating ResultFile
- Ballot station remembers opened election (again)
- Truly fixed the bug in LanSelView
- Enter a start condition. This macro really ought to take a parameter, but we do it the disgusting crufty way forced on us by the ()-less definition of BEGIN.



Do the bugs ever make it into the software used in elections? Absolutely. That's why "patches" (after-the-fact program modifications) are put on the machines.