Computer scientists take over electronic voting machine with new programming technique (w/ Video)

10 August 2009

UC San Diego computer science Ph.D. student Stephen Checkoway clutches a print out demonstrating that his vote-stealing exploit that relied on return-oriented programming successfully took control of the reverse engineered voting machine. Credit: UC San Diego / Daniel Kane

(PhysOrg.com) -- Computer scientists demonstrated that criminals could hack an electronic voting machine and steal votes using a malicious programming approach that had not been invented when the voting machine was designed. The team of scientists from University of California, San Diego, the University of Michigan, and Princeton University employed “return-oriented programming” to force a Sequoia AVC Advantage electronic voting machine to turn against itself and steal votes.

“Voting machines must remain secure throughout their entire service lifetime, and this study demonstrates how a relatively new programming technique can be used to take control of a voting machine that was designed to resist takeover, but that did not anticipate this new kind of malicious programming,” said Hovav Shacham, a professor of computer science at UC San Diego’s Jacobs School of Engineering and an author on the new study presented on August 10, 2009 at the 2009 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections (EVT/WOTE 2009), the premier academic forum for voting security research.

In 2007, Shacham first described return-oriented programming, which is a powerful systems security exploit that generates malicious behavior by combining short snippets of benign code already present in the system.

The new study demonstrates that return-oriented programming can be used to execute vote-stealing computations by taking control of a voting machine designed to prevent code injection. Shacham and UC San Diego computer science Ph.D. student Stephen Checkoway collaborated with researchers from Princeton University and the University of Michigan on this project.

“With this work, we hope to encourage further public dialog regarding what voting technologies can best ensure secure elections and what stop gap measures should be adopted if less than optimal systems are still in use,” said J. Alex Halderman, an electrical engineering and computer science professor at the University of Michigan.

The computer scientists had no access to the machine’s source code—or any other proprietary information—when designing the demonstration attack. By using just the information that would be available to anyone who bought or stole a voting machine, the researchers addressed a common criticism made against voting security researchers: that they enjoy unrealistic access to the systems they study.

“Based on our understanding of security and computer technology, it looks like paper-based elections are the way to go. Probably the best approach would involve fast optical scanners reading paper ballots. These kinds of paper-based systems are amenable to statistical audits, which is something the election security research community is shifting to,” said Shacham.

“You can actually run a modern and efficient election on paper that does not look like the Florida 2000 Presidential election,” said Shacham. “If you are using electronic voting machines, you need to have a separate paper record at the very least.”

Last year, Shacham, Halderman and others authored a paper entitled “You Go to Elections with the Voting System You have: Stop-Gap Mitigations for Deployed Voting Systems” that was presented at the 2008 Electronic Voting Technology Workshop.”

“This research shows that voting machines must be secure even against attacks that were not yet invented when the machines were designed and sold. Preventing not-yet-discovered attacks requires an extraordinary level of security engineering, or the use of safeguards such as voter-verified paper ballots,” said Edward Felten, an author on the new study; Director of the Center for Information Technology Policy; and Professor of Computer Science and Public Affairs at Princeton University.

Return-Oriented Programming Demonstrates Voting Machine Vulnerabilities

To take over the voting machine, the computer scientists found a flaw in its software that could be exploited with return-oriented programming. But before they could find a flaw in the software, they had to reverse engineer the machine’s software and its hardware—without the benefit of source code.

Princeton University computer scientists affiliated with the Center for Information Technology Policy began by reverse engineering the hardware of a decommissioned Sequoia AVC Advantage electronic voting machine, purchased legally through a government auction. J. Alex Halderman—an electrical engineering and computer science professor at the University of Michigan (who recently finished his Ph.D. in computer science at Princeton) and Ariel Feldman—a Princeton University computer science Ph.D. student, reverse-engineered the hardware and documented its behavior.

It soon became clear to the researchers that the voting machine had been designed to reject any injected code that might be used to take over the machine. When they learned of Shacham’s return-oriented programming approach, the UC San Diego computer scientists were invited to take over the project. Stephen Checkoway, the computer science Ph.D. student at UC San Diego, did the bulk of the reverse engineering of the voting machine’s software. He deciphered the software by reading the machine’s read-only memory.

Simultaneously, Checkoway extended return-oriented programming to the voting machine’s processor architecture, the Z80. Once Checkoway and Shacham found the flaw in the voting machine’s software—a search which took some time—they were ready to use return-oriented programming to expose the machine’s vulnerabilities and steal votes.

The computer scientists crafted a demonstration attack using return-oriented programming that successfully took control of the reverse engineered software and hardware and changed vote totals. Next, Shacham and Checkoway flew to Princeton and proved that their demonstration attack worked on the actual voting machine, and not just the simulated version that the computer scientists built.

The computer scientists showed that an attacker would need just a few minutes of access to the machine the night before the election in order to take it over and steal votes the following day. The attacker introduces the demonstration attack into the machine through a cartridge with maliciously constructed contents that is inserted into an unused port in the machine. The attacker navigates the machine’s menus to trigger the vulnerability the researchers found. Now, the malicious software controls the machine. The attacker can, at this

point, remove the cartridge, turn the machine's power switch to the "off" position, and leave. Everything appears normal, but the attacker's software is silently at work.

When poll workers enter in the morning, they normally turn this type of voting machine on. At this point, the exploit would make the machine appear to turn back on, even though it was never actually turned off.

"We overwrote the computer's memory and state so it does what we want it to do, but if you shut off the machine and reboot from ROM, the exploit is gone and the machine returns to its original behavior," explained Checkoway.

The computer scientists tested a machine that is very similar to machines that are used today in New Jersey and Louisiana. These New Jersey and Louisiana machines may have corrected the specific vulnerabilities the computer scientists exploited, but they have the same architectural limitations. The researchers highlight the possibility that current voting machines will be vulnerable to return-oriented programming attacks similar to the attack demonstrated in this study.

"This work shows how difficult it is to design voting machines that will remain secure over time. It's impossible to anticipate what new kinds of attacks will be discovered in the future," said Halderman.

More information:

Related publications:

J.A. Halderman, E. Rescorla, H. Shacham, and D. Wagner. "You Go to Elections with the Voting System You Have: Stop-Gap Mitigations for Deployed Voting Systems." In D. Dill and T. Kohno, eds., Proceedings of EVT 2008. USENIX/ACCURATE, July 2008. cseweb.ucsd.edu/~hovav/papers/hrsw08.html

R. Roemer, E. Buchanan, H. Shacham, and S. Savage. "Return-Oriented Programming: Systems, Languages, and Applications." 2009. In review. cseweb.ucsd.edu/~hovav/papers/rbss09.html

E. Buchanan, R. Roemer, H. Shacham, and S. Savage. "When Good Instructions Go Bad: Generalizing Return-Oriented Programming to RISC." In P. Syverson and S. Jha, eds., Proceedings of CCS 2008, pages 27-38. ACM Press, Oct. 2008. cseweb.ucsd.edu/~hovav/papers/brss08.html

Source: University of California - San Diego (news : web)