New Software Continuously Scrambles Code to Foil Cyber Attacks

Technique Sets a Deadline on Hackers to Severely Limit Chances of Success

As long as humans are writing software, there will be coding mistakes for malicious hackers to exploit. A single bug can open the door to attackers deleting files, copying credit card numbers or carrying out political mischief.

Code-shuffling software developed at Columbia effectively eliminates opportunities for hackers to reuse code to take control of a machine.

A new program called Shuffler (http://www.cs.columbia.edu/~junfeng/papers/shuffler-osdi16.pdf) tries to preempt such attacks by allowing programs to continuously scramble their code as they run, effectively closing the window of opportunity for an attack. The technique is described in a study (http://www.cs.columbia.edu/~junfeng/papers/shuffler-osdi16.pdf) presented this month at the USENIX Symposium on Operating Systems and Design (https://www.usenix.org/conference/osdi16) (OSDI) in Savannah, Ga.

"Shuffler makes it nearly impossible to turn a bug into a functioning attack, defending software developers from their mistakes," said the study's lead author, David Williams-King, a graduate student at Columbia Engineering (http://engineering.columbia.edu/). "Attackers are unable to figure out the program's layout if the code keeps changing."

Even after repeated debugging, software typically contains up to 50 errors (http://www.mayerdan.com/ruby/2012/11/11/bugs-per-line-of-code-ratio) per 1,000 lines of code, each a potential avenue for attack. Though security defenses are constantly evolving, attackers are quick to find new ways in.

In the early 2000s, computer operating systems adopted a security feature called address space layout randomization, or ASLR. This technique rearranges memory when a program launches, making it harder for hackers to find and reuse existing code to take over the machine. But hackers soon discovered they could exploit memory disclosure bugs to grab code fragments once the program was already running.

Shuffler was developed to deflect this latter style of code-reuse attack. It takes ASLR's code-scrambling approach to the extreme by randomizing small blocks of code every 20 to 50 milliseconds, imposing a severe deadline on would-be attackers. Until now, shifting around running code as a security measure was thought to be technically impractical because existing solutions require specialized hardware or software.

In the above demo, "#"s represent code in memory as a typical web server runs. When the server shifts to running with Shuffler, the '#'s move every 50 milliseconds. The

"By the time the server returns the information the attacker needs, it is already invalid—Shuffler has already relocated the respective code snippets to different memory locations," said study coauthor Vasileios Kemerlis, a computer science professor at Brown University.

Designed to be user-friendly, Shuffler runs alongside the code it defends, without modifications to program compilers or the computer's operating system. It even randomizes itself to defend against possible bugs in its own code.

The researchers say Shuffler runs faster and requires fewer system changes than similar continuous-randomization software such TASR (http://web.mit.edu/ha22286/www/papers/CCS15_2.pdf) and Remix (http://www.cs.fsu.edu/~whalley/papers/codaspy16.pdf), developed at MIT Lincoln Labs and Florida State University respectively.

shuffled web server serves the web page seen at the end of the demo.

As an invitation to other researchers to try and break Shuffler, Williams-King is currently running the software on his personal website (http://shuffled.elfery.net:8000/). (He can check that the code is shuffling and whether anyone has attacked the site by reviewing the program's logs).

On computation-heavy workloads, Shuffler slows programs by 15 percent on average, but at larger scales—a webserver running on 12 CPU cores, for example—the drop in performance is negligible, the researchers say.

This versatility means that software distributors as well as security-conscious individuals could be potential end users. "It's the first system that is trying to be a serious defense that people can use, right now," said Williams-King.

Shuffler needs a few last improvements before it is made public. The researchers say they want to make it easier to use on software they haven't yet tested. They also want to improve Shuffler's ability to defend against exploits that take advantage of server-crashes.

"Billions of lines of vulnerable code are out there," said the study's senior author, Junfeng Yang (http://datascience.columbia.edu/junfeng-yang), a computer science professor at Columbia Engineering (http://engineering.columbia.edu/) and member of the Data Science Institute (http://datascience.columbia.edu/). "Rather than finding every bug or rewriting all billions of lines of code in safer languages, Shuffler instantly lets us build a stronger defense."

The study is titled "Shuffler: Fast and Deployable Continuous Code Re-Randomization." The other authors are Graham Gobieski, James Blake, Xinhao Yuan and Michelle Zheng, of Columbia; and Kent Williams-King, Patrick Colp and William Aiello, of the University of British Columbia.

— *Kim Martineau*

Posted: Nov 17 2016

