Web sites that can take a punch

March 17 2010, by Larry Hardesty

The recent, well-publicized cyberattack on Google was just the latest skirmish in a long war. And like most long wars, this one features an arms race, as hackers seek out new security holes, and web site administrators try to close them.

Systems for detecting attacks against networked computers are commercially available, and academic and industrial researchers are constantly improving them. But when a web site is under attack, its only viable defense may be to take its servers offline, which, in the short term, can cost it money in lost revenue and productivity and, in the long term, could hurt its credibility. Indeed, knocking a site offline may be an attackers' sole intention.

MIT researchers have developed a system to keep web servers — or, for that matter, any Internet-connected computers — running even when they're under attack. The work was funded largely by the U.S. Defense Department's Defense Advanced Research Projects Agency (DARPA), and in a pair of tests whose thoroughness is unusual in academia, DARPA hired a group of computer security professionals outside MIT to try to bring down a test network protected by the new system. In both tests, says Martin Rinard, the professor of electrical engineering and computer science who led the research, the system exceeded all the performance criteria that DARPA set for it.

The MIT system was developed by a host of researchers, including not only Rinard but Jeff Perkins, a research scientist at MIT's Computer Science and Artificial Intelligence Lab, Postdoctoral Fellow Stelios Sidiroglou-Douskos and Professor Michael Ernst, who has since moved to the University of Washington. During normal operation, it monitors the programs running on an Internet-connected computer to determine their normal range of behavior, and during an attack, it simply refuses to let them wander outside that range.

To take a simple example, suppose that a program running on a web server routinely stores data in one of two memory locations — call them A and B. During an attack, malicious code tries to trick the program into storing data at location C instead. The MIT system won't let it: instead, it sends the data to either location A or location B.

Of course, the data may not be of a type that belongs at either of those locations. And the system will modify behaviors that could be even more disruptive than data storage. But in sites with large banks of servers, the MIT system gets several chances to find the best response to an attack. If storing at location A causes one server in the bank to crash, the MIT system will tell the other servers to store it at location B, instead.

“The idea is that you’ve got hundreds of machines out there,” Rinard says. “We’re saying, ‘Okay, fine, you can take out six or 10 of my 200 machines.’” But, he adds, “by observing what happens with the executions of those six or 10 machines, we’ll be able to deploy patches out to protect the rest of the machines.” The entire process of recognizing an attack, testing a number of countermeasures and deploying the most effective ones can take a matter of seconds.

Baptism by fire

In the first of DARPA’s two field tests, engineers at a computer security firm — the so-called red team — were given the code for the MIT defense system. (In the real world, a company that marketed such a system would make every effort to keep its code secret, but Rinard says that it’s standard practice in the security field to consider the worst-case scenario.) The red team had several months in which to devise attacks against a hypothetical network protected by the system. During the test itself, no malicious code was allowed to execute on the protected computers, and in 70 percent of cases, the MIT system kept the applications running on those computers from going down. DARPA also set performance goals for the system, such as the amount of extra processing power it required, and the extent to which it altered the applications’ normal operation. In all cases, the system was well within DARPA’s prescribed limits.

The first red-team exercise considered cases in which hackers tried to infect computers with malicious code, and the MIT researchers presented the results of the test at the Association for Computing Machinery’s Symposium on Operating Systems Principles last fall. A second red-team exercise, testing an updated version of the defense system that the MIT researchers developed together with defense contractor BAE Systems, concluded at the end of January. That test evaluated the system’s ability to handle a different kind of attack, which

seeks to circumvent security checks that web applications typically perform to ensure that users have permission to access protected information. Although the researchers are still sorting through the data from that test, Sidiroglou-Douskos says that the system's success rate in keeping applications up and running rose from 70 percent to 90 percent.

Angelos Keromytis, an associate professor of computer science at Columbia University, who works on related techniques for combating cyberattacks, says that the MIT approach is "very original," but cautions that Web developers may be reluctant to adopt it anytime soon. "They're wary of a system that changes another system automatically," Keromytis says. "When they manually make changes to their systems, they break them, so they think that automatically doing it is going to be worse." Keromytis points out, however, that while DARPA has run a number of red-team exercises evaluating new technologies in a range of areas, "This is probably one of the most successful exercises that I have seen." The mere fact that DARPA was willing to spend so much money testing the system, Keromytis says, indicates that "they think it's close enough to a rough prototype that works, which is more than one can say for most academic research."

Provided by Massachusetts Institute of Technology