OPINION

Musings

RIK FARROW

Rik is the editor of *;login:*.
rik@usenix.org

Once upon a time, Anonymous meant just that. Today, at least for the security conscious, Anonymous means part of the anti-security underground. In particular, Anonymous has been a group of politically aware hackers who have been very good at embarrassing large organizations and corporations by exposing their data.

I mentioned Anonymous (and LulzSec) in my August 2011 column, but a couple of things got me thinking about them again. The first was a short article [1] written by an author I have worked with, Adam Penenberg, about some research on plagiarism published by another friend.

It seems incredible to me that people would dare to plagiarize online materials and call it their own work when all it takes to unmask them is the time and focus necessary to do a bit of searching. Jericho, one of the founders of Attrition.org, began investigating the published work of security charlatans and quickly discovered entire books that had been plagiarized [2].

Adam's article lists other security books where most of the content was copy-and-pasted from online sources. But Adam's article also got me thinking about security. He starts his article by writing about how common it is to copy other people's code, with implicit permission, and add it to your own code. Copying of working code has a long history, and it certainly makes it possible to build your own software faster. The problem with this quick approach is that you may be copying flaws that result in exploitation of your software, or you may quickly build Internet-facing Web applications that you don't understand very well and have failed to secure properly.

And that gets me to the Anonymous connection. A private source had pointed out that the publicized, that is, successful, attacks by Anonymous and LulzSec were using well-known penetration testing tools: Web Application Attack and Audit Framework (W3AF) [3]; sqlmap (for SQL injection of Web apps or databases) [4]; and Nikto, a Web security scanner with roots in the '90s [5]. All of these tools are covered by Snort signatures, implying that noticing that your Internet-facing sites are being probed is easy—if you are paying attention. It also suggests that you could be doing the probing yourself: the tools are free, although you do need to learn how to use them.

None of this is rocket science. Far from it. Just as guns make murder easier, tools like these make hacking much easier. If you don't scan your own mass of Internet-facing code, sooner or later someone will do it for you. I strongly encourage you to do your own scanning, as well as to follow good programming practices [6] and

watch for attacks on your servers using IDS or IPS (which could at least block the scanners' IP addresses).

The Lineup

I am cutting my own column short this month, because this issue has the reports from USENIX Federated Conferences Week, and those alone consume a huge portion of the printed version of *;login:*. I hope you appreciate the reports as much as I do, as a good report can tell you a lot more about a paper than the abstract can (or will). By reading reports, you can catch up on current research, or decide to go to the USENIX Web site for the event to watch or listen to a recorded invited talk or panel.

We have several excellent articles in this issue, starting with Greg Burd explaining NoSQL. I am excited about Greg's article, not just because he explains NoSQL clearly, but because he explains why NoSQL became necessary. Greg compares relational databases with NoSQL, as well as contrasting different types of NoSQL databases with each other. As I was reading, it became really obvious to me why Oracle needed to buy Sun, but that's just an aside. Margo Seltzer suggested Greg as an author and also read the near-final draft, which helped in polishing this article.

I contacted Paul Vixie when something he had posted suggested that there are other potential uses for DNSSEC. In the August 2011 *;login:*, Peter Gutmann had pointed out how easy it is to spoof SSH fingerprints, something I wasn't aware of. Paul's post suggested a simple fix for this if you are using DNSSEC. Paul's article describes other real and potential uses for DNSSEC, with the ability to securely use self-signed certificates high on the list.

Don Revelle's article answers questions about virtualization technology that I have been asking for a long time. I have helped publish a handful of virtualization articles in *;login:*, but none that covers the breadth of the technologies, explaining how they differ, the way Don's does. The point of this article is not to get you to change your choice for virtualization, but to understand how the technologies differ.

The final article in this issue explains the thinking behind changing the whois protocol. Andy Newton (ARIN), Dave Piscitello (ICANN), Benedetto Fiorelli (RIPE), and Steve Sheng (ICANN) have written about how the whois protocol, and client software, performs poorly for an Internet that is both international in scope and embraces several regional authorities that have differed in how they present whois data. Whois is supposed to work uniformly today, but it does not. They present a new interface for whois servers that supports both Web browsers and new clients, as well as internationalization.

David Blank-Edelman has focused his steely eyes (actually, they are quite a bit more friendly than that) on performance. David shows how to use two different tools that can help you pinpoint performance issues in your Perl scripts. Fixing them is another issue, but David makes it easy to find where the problem may be.

Dave Josephsen tries out a new approach to his column: he interviews a provider of a hosted monitoring service. Dave asks Theo Schlossnagle about his company's (OmniTI) Circonus service and about the future of monitoring systems in general.

Peter Galvin has both a column and a book review in this issue. In his column, Peter continues on the cloud theme from the August 2011 *;login:*, explaining issues for enterprises considering using some form of cloud.

Robert Ferrell also decided to take a hard look at virtualization for this issue, coming up with a parable about how virtualization has disturbed the firmament.

Elizabeth Zwicky has four book reviews this time, along with two from Sam Stover, one from Trey Darley, and one from Peter Galvin.

Tools and frameworks have it easier than ever to write code—but to secure it. In the best of all possible worlds, tools would produce only secure code. In the real world, you must either audit and assess your own code or take responsibility for your organization facing embarrassment and possibly lawsuits.

References

[1] Adam Penenberg, "When Hacks Attack: The Computer Security Textbook Plagiarism Epidemic," *Fast Company*, July 27, 2011: http://www.fastcompany.com/1769244/plagiarism-professionals.

[2] Example of Jericho's work in uncovering plagiarism: http://attrition.org/errata/charlatan/gregory_evans/spyware_reference_study_guide.html.

[3] W3AF (Web Application Attack and Audit Framework): http://w3af.sourceforge.net.

[4] sqlmap (automatic SQL injection and database takeover tool): http://sqlmap.sourceforge.net/.

[5] Nikto, Web server scanner in Perl: http://cirt.net/nikto2.

[6] A good place to get started is the Open Web Application Security Project: https://www.owasp.org/index.php/Getting_Started.