



HOME >> SPECIAL REPORTS

Think like a hacker

By [Jeff Feinman](#)

February 15, 2009 — In the eyes of Mike Weider, the correct way of doing software security testing requires getting into the

The director of security products for IBM Rational said it takes a special breed of software professional to step into the driver's seat. While quality assurance professionals can do security testing and smoke out some vulnerabilities, they usually have the customer's perspective. The hacker, on the other hand, is the customer.

"There is a need for this specialized security testing professional to anticipate how hackers think and use this slightly different

From a technology standpoint, there are two main approaches for testing software for security, and they are well known to developers. The first is the outside-in approach: testing to see how the application responds to a simulated attack. The second is the inside-out approach: testing coding patterns that would highlight vulnerabilities in the code.

But security testing can be fundamentally a different ballgame than traditional testing because the emphasis is put on what attackers do. First, users don't usually try to search out software bugs, while malicious attackers intentionally seek out vulnerabilities. Vulnerabilities arise for other users instead of just a developer or group of developers.

Additionally, developers usually learn to avoid poor programming practices for their own projects, but it is difficult for security professionals to grow every year. This makes it more difficult to ensure that secure programming practices are followed.

So what is the best way to carry out proper security testing? Vic DeMarines, vice president of products at security tool maker Fortify, thinks like the attackers themselves and to look for the easiest way to initiate a threat. Applications will have different levels of

"If we're talking about a financial application that's running in a hosted environment, and it's a Web application, the tester will look for logic," DeMarines said. "If we're talking about a piracy threat to a publisher, someone who is issuing software as part of their business, they take a look at how those attacks are carried out, and then look at ways to bolster your defense against that. The first step is to understand the application and the methods they're going to use to go after the vulnerabilities."

Depending on what the threat is, organizations can either use their own resources or outsource for analysis to carry out this function. The use of an outside resource to test for threats depends on the amount of money an organization has.

Jacob West, manager of Fortify Software's security research group, said the most important thing in assessing the security of an application is security testing activity because it puts the focus on things the software is not supposed to do instead of things it's supposed to do.

"The challenges for security testers are going to be very similar to the challenges that developers went through as software development evolved," West said. "Early on, we had developers saying security isn't their problem, that's what the security team does, and slowly we've moved security into the development process."

"The same evolution is going to happen in the security testing space. People who have thought of themselves as doing traditional development as part of their approach. As such, they'll start to build up some knowledge of security and the kinds of things that can happen."

Gwyn Fisher, CTO of Klocwork, another security tool maker, said the whole issue with security is that a tester should not interfere with the development process. Instead, security testing must involve everyone; the architect should test his or her design assumptions for vulnerabilities as they are being produced, and so forth.

"You get a secure product by following a secure development life cycle. You don't get it by testing," Fisher said. "It's all about outside in; it comes from the inside out."

Adding security to the cycle

There is a great deal of talk in the software industry about making sure security is instilled throughout the software development process. However, Weider said IBM Rational has seen "an evolutionary approach," where the security team looks at the beginning of the development process.

"That works well to begin with, to have sort of a gatekeeper, but that also has the potential to become a bottleneck in that the number of developers," Weider said. "With the movement to agile approaches [that have] many smaller iterations, having as much sense as having it embedded throughout the process."

An agile process caters well to the idea of implementing security throughout the development life cycle because the concept of security in the past, security testing has been something done last minute, and it was typical for testers to find a glaring vulnerability that then be left with the sticky situation of either delaying the release to fix the vulnerability, or looking the other way and letting that, Weider said, there has been a push to integrate security testing into earlier phases of development.

One of the things IBM Rational is trying to do is to integrate security within development tools in such a way that it becomes a part of the development process rather than something that's bolted on after the software is developed.

Vi Labs' DeMarines agreed that security needs to be kept in mind throughout the development process. In terms of thinking about security during construction, a tester has the chance to figure out how to make it more resistant to tampering or piracy during the design phase. A tester can analyze or reverse-engineer an application. This can give the tester a better sense of how sturdy the application is.

"You don't want to be starting to think about testing security as you're coming into a release candidate," DeMarines said. "You want to make sure the functionality has been implemented in a way that you can test it, and then figure out how to make it resistant to the kinds of attacks."

While there are many products on the market that allow software providers to scan source code for vulnerabilities, and it is important to understand what the threat is and putting that feedback into the design, DeMarines added.

West said that mentality is pretty well ingrained in most developers today, and the enterprise software industry has realized that "There's still a wide range of maturity levels in terms of how close companies are getting to obtaining that utopia of software that companies are doing whatever they're able to now in order to make that a reality," he said.

Positive vs. negative

When attempting to implement security throughout the development process, defining the proper security requirements is not redundant or difficult to comprehend for other professionals on a project. Some security requirements do come in the same form as positive security features, such as a particular encryption algorithm the software should use, or making sure user accounts are protected.

In security requirements, there is a much greater emphasis on negative requirements, where testing revolves around statements like "The contents of the Web page," or, "Unauthorized users should not be able to access data." This will have a big effect on the development process because requirements involves creating conditions that will verify that the requirement is satisfied by the software. On the other hand, negative requirements never occur, which would involve creating every possible set of conditions to determine that it won't occur, which is close to impossible.

Most relevant requirements aren't going to be positive, said West. Some will be redundant because they will tell the tester to "The attacker should never be able to take control of an application," is the type of requirement that should be moved away from being a requirement. Other requirements can give clues to testers downstream about what might constitute unintended behavior.

"You could say, for example, the application will never render un-encoded HTML or Javascript," said Fortify's West. "That's something that your knowledge could go into a specific test case and verify. They can submit HTML characters to that page, and when it is displayed, it's going to be a challenge because there will always be a risk of turning a vulnerability around into a negative requirement that is not verifiable in some way."

West added that testers need to go to a lower level of granularity and turn vague requirements into smaller, more actionable requirements. For example, instead of stating that there is a possibility of a command injection vulnerability because shell scripts are running on the system, one might say, "No user input should be allowed to appear in a shell command that's executed."

IBM Rational's Weider said that articulating security requirements can be difficult, and rarely are there people with the right skill set. He emphasized the importance of security in the design of applications, and said it is important to integrate threat modeling and ensure that security is captured in the design and architecture.

Additionally, it's not always easy to have all requirements accepted because there can be many more requirements than can be implemented "as far as what requirements make it into the process. Because of this, testers need to prioritize security improvements. Security professionals prioritize quality and functional improvements.

“That is the challenge, but I think every year it’s getting a little better as security is becoming more accepted within development was well understood, but now as we’ve seen application security becoming one of the main vulnerabilities on the Internet as development requirements has been getting steadily better every year.”

Related Search Term(s): [professional development](#), [security](#), [testing](#), [IBM](#), [Fortify](#), [Klocwork](#), [Vi Labs](#)

Share this link: <http://www.sdtimes.com/link/33274>

Comments

On a related note and for similar content, see our book "The Art of Software Security Testing" published in 2006 http://www.identified.com/Identifying/dp/0321304861/ref=ntt_at_ep_dpt_2, for example chapters 1, pg 11 "Think like an Attacker" and chapter 3 "Th

Elfriede Dustin

[HOME](#)

[SITE MAP](#)

[CURRENT ISSUE](#)

[BACK ISSUES](#)

[SUBSCRIBER SERVICES](#)

[ABOUT US](#)

Copyright © 1999-2010 BZ Media LLC, all rights reserved. Legal and Privacy Policy
Phone: +1 (631) 421-4158 • E-mail: info@bzmedia.com