

Interpreting the Results of a Vulnerability Assessment: How to Focus on What's Important in Your Web Application Security Testing

By Kevin Beaver, CISSP, and Caleb Sima

As with many other business analysis issues, there are three sides to the story when looking at Web application security testing: yours, the findings of your vulnerability assessment, and the truth. Whether you're using a commercial or open source scanner, you're undoubtedly going to glean a lot of information and come across vulnerabilities. The problem is that many of these Web application security holes aren't as big a deal as they may seem. Regardless of any marketing fluff or pre-canned security policies and reports, your organization's network, business needs, and risk tolerance dictate what really matters when it comes to sorting through all the results you get when performing a vulnerability assessment.

Web application security testing tools are extremely savvy and are able to root out vulnerabilities in minutes that would take the best hacker in the world hours, months, or more to find. The issue is that you've got to take the tool results and determine what actually matters in your environment. We've seen inexperienced Web application security consultants, managed security service providers, and auditors run vulnerability assessment scans and then hand the results over to their clients purporting they have a bunch of problems that need to be fixed. Likewise, we've seen network administrators absolutely freak out when they see that their Web application security testing tool has found a dozen or more vulnerabilities. They believe the sky is falling and immediately run to management asking for more budget to buy more technology to fix the problems.

It doesn't have to be this way and shouldn't be this way if you want people to take your Web application security testing seriously - especially managers and developers. You'll have to step back and look at the big picture in order to really wrap your head around which results of a vulnerability assessment matter in your organization's specific situation. This may mean looking at the findings from a different angle (i.e. inside the firewall instead of from the outside only), while logged in (or logged in through various roles), or manually carrying out the exploit.

Bottom line, unless your Web application security testing tool has exploited the vulnerability and handed you the findings on a silver platter, it's going to require you digging into the results of the vulnerability assessment and verifying the problem is indeed a problem.

The following are some real-world examples of Web application security vulnerabilities discovered when testing various Web applications. Label them as false-positives, oversights, paranoia, or whatever - the bottom line is that they appeared serious on the surface but ended up not really mattering in the end.

- 1. Vulnerability Assessment Finding:** SQL injection discovered that can lead to database access
End result: Adequate user input validation was taking place behind the scenes and no data was actually extractable.
- 2. Vulnerability Assessment Finding:** SSL is not present on the login page allowing session IDs and login credentials to be sent in clear text that could lead to capturing, hijacking, and more
End result: The administrator had not yet loaded the digital certificate on the server.
- 3. Vulnerability Assessment Finding:** Buffer overflow vulnerability present in the Web server software that can be exploited to obtain a remote command prompt on the server.
End result: Trusting firewall and IDS rules were enabled allowing all traffic into the Web server.

- 4. Vulnerability Assessment Finding:** Microsoft FrontPage virtual directories, FTP directories, etc. were found that could lead to exploitation
End result: Proper directory permissions were present preventing actual access
- 5. Vulnerability Assessment Finding:** Backup files with a .old extension were found that could lead to source code leakage and exploitation
End result: The files were executable, documentation, and home page files that had little to no relationship to the application or bearing on its security.
- 6. Vulnerability Assessment Finding:** An outdated version of the Apache Web server was installed that has multiple known vulnerabilities that can be exploited and lead to unauthorized system access
End result: Apache was nowhere to be found on the system.
- 7. Vulnerability Assessment Finding:** Files, links, and email addresses found in the Google Hacking Database (GHDB) were discovered that leak sensitive information
End result: These files, links, and email addresses were necessary for proper operation of the Web application.
- 8. Vulnerability Assessment Finding:** Macromedia Dreamweaver remote database scripts were accessible that can lead to an attacker executing arbitrary SQL queries
End result: These files were only accessible when logged in as the manager/admin account for the Web application which was by design.

Some of these vulnerabilities may seem benign, but anyone taking them out of context can make a big deal out of nothing. These types of issues can be the difference between a relatively secure Web application that's safe to use and a failed Web application security audit that stirs up unnecessary controversy leading to time, effort, and money hastily spent on remediation.

When it comes to Web application security testing and remediation, focus on the urgent and important in your environment. That is, root out the vulnerabilities that can be exploited by an attacker in a typical working scenario in your specific situation. What needs to be addressed between now and next week? What can wait a month or so? What's not even worth the effort? Bottom line, focus on context. We are not saying ignore the other issues that come across in a vulnerability assessment. We're just saying that you've likely got much bigger fish to fry when thinking about Web application security than worry about random vulnerabilities that may never be found and if they are will have a very minor chance of being exploited leading to anything of value.

No matter how solid your Web application security is, someone somewhere will always find a way to attack the applications. That's why you have to have layered fail-safe controls such as firewalls, IPS, input validation, solid authentication requirements, minimum necessary access controls, hardened Web servers and underlying operating systems, system monitoring, and so on. When one control fails, you've got a half-dozen other things to protect the system.

Putting the results of a vulnerability assessment into perspective and not jumping at every issue your scanners come across will help take your Web application security testing to the next level. You'll not only show management that you understand the business side of balancing strong Web application security with reality, but, perhaps most importantly, you'll create less work for you, your team, and your developers so everyone can focus on things that really matter.

About Caleb Sima

Caleb Sima is the co-founder of SPI Dynamics, a [Web application security](#) products company. He currently serves as the CTO and director of SPI Labs, SPI Dynamics' R&D security team. Prior to co-founding SPI Dynamics, Caleb was a member of the elite X-Force R&D team at

Internet Security Systems, and worked as a security engineer for S1 Corporation. Caleb is a regular speaker and press resource on [Web application security testing methods](#) and has contributed to *(IN)Secure Magazine*, *Baseline Magazine* and been featured in the Associated Press.

About Kevin Beaver

Kevin Beaver is an independent information security consultant, speaker, and expert witness with Atlanta-based [Principle Logic, LLC](#). He has more than 19 years of experience in IT and specializes in performing information security assessments revolving around compliance and IT governance. Kevin has authored/co-authored six books on information security including *Hacking For Dummies* and *Hacking Wireless Networks For Dummies* (Wiley) as well as *The Practical Guide to HIPAA Privacy and Security Compliance* (Auerbach). He's also the creator of the *Security On Wheels* audiobook series. Kevin can be reached at kbeaver ~at~ principlelogic.com.