It's Always Risky Business with Web Applications

By Don Spencer, President, IONA Software Solutions
May 30, 2014

There is a sign on the front door of our office building. It's been there all winter. It says something like this: "Would the last person leaving the building please make sure the doors are locked? They sometimes stick". Sometimes it seems like web application security is handled the same way. I can almost picture a project manager standing up in a team meeting and saying, to nobody in particular, something like this: "We know we may have a security problem with our site, could somebody please check it before we go live?"

> "damage to the reputation of the business"

Losing your customers' data is not only embarrassing, it can also be expensive. Both in terms of direct financial penalties and damage to the reputation of the business.

What Are Web Application Vulnerabilities Anyway?

A vulnerability is a weakness that allows an attacker to compromise the availability, confidentiality or integrity of a computer system. Vulnerabilities may be the result of a programming error or a flaw in the application design that will affect security.

Attackers will typically target every spot in an application that accepts user input and systematically attempt to discover and exploit these weaknesses. It is important to note that there is a distinct difference between the discovery of a web application vulnerability and the subsequent exploit. Often, a combination of different, seemingly harmless, vulnerabilities will be chained together by a creative attacker resulting in a major security breach. Therefore, security testing experts usually focus on identifying and demonstrating the individual vulnerabilities, rather than attempting a specific exploit. For example, a security analyst may identify that an application has both cross site scripting (XSS) and cross site request forgery (CSRF) vulnerabilities rather than demonstrating that these can be combined to steal a user's confidential information.

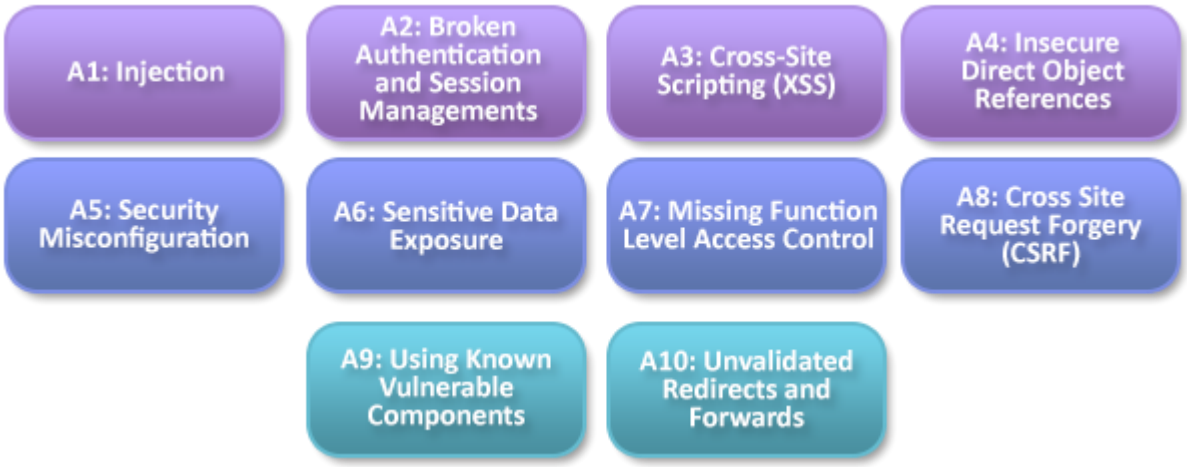
> "Attackers will typically target every spot in an application that accepts user input"

The bad news is that the majority of web applications deployed today have these types of vulnerabilities. Statistics published by web security companies and organizations suggest that up to 96% of custom web applications have a least one serious vulnerability. My personal

experience is that 100% of the web applications I have analyzed have multiple serious security vulnerabilities. The good news is that organizations like the Open Web Application Security Project (OWASP) are working to raise awareness and provide resources to make the web a safer place.

"OWASP is an open community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted"[1]

The OWASP Application Security Verification Standard (ASVS) provides a comprehensive basis for testing web applications. It provides developers with a tool they can use to measure the degree of trust that can be placed in their applications. The ASVS can also be used by the security test team to develop a test plan that addresses the security requirements of a web application. OWASP also periodically publishes a list of the top ten web vulnerabilities.



Source: https://www.owasp.org/index.php/Top_10_2013-Top_10

To one degree or another everyone associated with a web application development project should have a solid understanding of the OWASP Top Ten.

Whose Job Is It To Find These Vulnerabilities?

Developers?

A lot of developers tend to follow "best practices" and rely on the "baked-in security features" in their framework of choice. The problem is that sometimes business requirements or other circumstances can push the developer outside of these "best practices".

When that happens, all bets are off. A developer that is not aware of the implications of the vulnerabilities in OWASP Top Ten can, and probably will, introduce security flaws into the

application. It also goes without saying that if your developers are not aware or trained in the issues then they certainly are not going to think to test for them.

The QA Team?

Traditional functional testing or performance testing is typically focused on verifying that the application under test meets the functional and non-functional requirements defined by business. A quick glance at the OWASP Top Ten shows that these are not the types of requirements you will see on a typical test plan. QA usually focuses on making sure the application works as intended and all the test cases pass. Security testing is little different. You are trying to make sure that specific attacks do not succeed and all the test cases fail.

> "Security is best approached as a combination of people, process and technology."

End Users?

That just leaves your users (and really – who leaves testing up to their users?). In the normal usage of an application, you will not be hearing from your users that you have security problems. The only people that will uncover these types of issues are those that have the training and motivation to find them. These people fall into two camps, white hats and black hats. The white hats will not try to break into your web application without explicit permission. You will not hear from the black hats until it is too late…

Where Do I Start?

Security is best approached as a combination of people, process and technology. Everybody involved in the development and testing of a web application needs to learn more and a great place to start is the OWASP web site. Developers can use the resources on the web site to develop more secure code by:

- Following the best practices in OWASP's Guide to Building Secure Web Applications: https://www.owasp.org/index.php/Guide
- Using the cheat sheets: https://www.owasp.org/index.php/Cheat_Sheets
- Reviewing the application using the OWASP Code Review Guide: https://www.owasp.org/index.php/Code_Review_Guide

There is content on the OWASP web site for the QA team as well. They can:

- Use OWASP's Application Security Verification Standard as a guide to what an application needs to be secure: https://www.owasp.org/index.php/ASVS
- Review the application following the OWASP Testing Guide: https://www.owasp.org/index.php/Testing_Guide

Last and not least, consider hiring a security expert to verify your application is secure before it goes live. A security expert should have a

strong development background, know how developers think and be able to think like an attacker. Having an expert examine your application is your last chance to ensure that the "doors are locked" before the bad guys start looking. By the way, I noticed today that the sign on the door is gone. The landlords hired a professional to fix them.

1https://www.owasp.org/index.php/About_The_Open_Web_Application_Security_Project

About The Author

Don Spencer is the founder and president of IONA Software Solutions located in Fredericton, NB. As an IT professional with over 25 years of experience in software application design and development, Don has in-depth knowledge of all aspects of the software development and product lifecycles. A strong proponent of Agile Development Methodologies, Don has a proven track record developing and motivating highly productive teams capable of creating high quality commercial software products and solutions. His areas of specialization include Agile Development Methodologies, team leadership and web application security.