



WEB APPLICATION SECURITY: AUTOMATED SCANNING OR MANUAL PENETRATION TESTING?

DANNY ALLAN, STRATEGIC RESEARCH ANALYST

A whitepaper from Watchfire

TABLE OF CONTENTS

| | |
|--|----------|
| Introduction | 1 |
| History | 1 |
| Vulnerability Types | 1 |
| Technical Vulnerabilities | 2 |
| Logical Vulnerabilities | 3 |
| Statistics | 3 |
| Conclusion | 3 |
| About Watchfire | 4 |

Copyright © 2006. Watchfire Corporation. All Rights Reserved. Watchfire, WebXM, Bobby, AppScan, PowerTools, the Bobby Logo and the Flame Logo are trademarks or registered trademarks of Watchfire Corporation. All other products, company names and logos are trademarks or registered trademarks of their respective owners.

Except as expressly agreed by Watchfire in writing, Watchfire makes no representation about the suitability and/or accuracy of the information published in this whitepaper. In no event shall Watchfire be liable for any direct, indirect, incidental, special or consequential damages, or damages for loss of profits, revenue, data or use, incurred by you or any third party, arising from your access to, or use of, the information published in this whitepaper, for a particular purpose.

www.watchfire.com

INTRODUCTION

With more than 90 percent of web applications containing some type of security vulnerability,¹ and more than 75 percent of attacks occurring over the HTTP/S protocols, it is essential that organizations implement strong measures to secure their web applications. While the percentage of attacks occurring over ports 80 and 443 seems unusually large, consider the fact that these ports are the front door to the organization – usually exposed to the entire online community.

As web applications become increasingly complex, tremendous amounts of sensitive data – including personal, medical and financial information – are exchanged, and stored. The consumer not only expects, but demands, security for this information.

But securing a web application goes far beyond testing the application using manual processes, or by using automated systems and tools. It begins in the conceptual phase, by modeling the security risk introduced by the application as well as the countermeasures to be implemented. Security should be thought of as another quality vector of every application, analyzed and considered through every step of the application lifecycle. Discovering web application vulnerabilities can be performed in many ways:

- Automation
 - Scanning tools
 - Static Analysis
- Manual
 - Penetration testing
 - Code review

The purpose of this paper is to examine a few of these vulnerability detection methods – specifically comparing and contrasting manual penetration testing with automated scanning tools.

HISTORY

Manual security penetration testing is the oldest method for securing applications. Developers have tested their applications for flaws and problems during the development cycle for as long as software development has existed. Over time, as the frequency of attacks has grown and application complexity has increased, specialists whose sole purpose is to find and exploit such security problems have emerged. These individuals are known as “pen testers.”

The earliest recorded mention of automated web application testing was in 1999.² The web had graduated from its infancy to adolescence, and web browsers were only slowly becoming able to handle the complexities of dynamic applications. The goal of these tools was to automate the process of discovering a web application and injecting faults for the purposes of discovering vulnerabilities.

VULNERABILITY TYPES

Generally, most web application vulnerabilities can be grouped into one of two categories: technical and logical. Technical vulnerabilities include the following well-known tests: Cross-Site Scripting (XSS), Injection Flaws and Buffer Overflows. Logical vulnerabilities are much harder to explicitly categorize. These

¹ <http://www.imperva.com/company/news/2004-feb-02.html>

² <http://patft.uspto.gov/>

vulnerabilities manipulate the logic of the application to do things it was never intended to do. For example, in early 2002, a malicious individual used a logical vulnerability to bypass the required personal information validation in the Microsoft Hotmail application,³ allowing the user to reset passwords by guessing the answer to a single security question.

TECHNICAL VULNERABILITIES

Automated systems and tools are both methodical and comprehensive when it comes to testing for technical vulnerabilities. Consider for a moment the registration application for the Microsoft Hotmail application.⁴ This single form contains approximately thirty unique elements: some are hidden, while others are visually exposed. Each element of this form is potentially vulnerable to Cross-Site Scripting, Injection Flaws, Buffer Overflows or Improper Error Handling.

Did you know that more than 70 different techniques can be used to exploit Cross-Site Scripting – a technical vulnerability?⁵ This means that the single registration form could require more than 2000 tests (30 elements x 70 XSS techniques) to exhaustively test for this one exploit on the form! It is little wonder that more than 80 percent of all applications are vulnerable to this one issue.⁶

Automated systems and tools which crawl, analyze and test the web application are much better equipped to test for technical vulnerabilities than manual penetration tests. While automated scanning and testing tools may not currently address 100 percent of all technical vulnerabilities, there is no reason to believe that this will not happen in the near to short-term. Initial hurdles existed with application scanning tools having troubles in certain areas:

- Client-side generated URLs
- Required JavaScript functions
- Application logout
- Transaction-based systems requiring specific user paths
- Automated form submission
- One time passwords
- “Infinite” web sites with random URL-based session IDs

As automated web application security tools have matured, these hurdles have all been met and solved.

Over time, automated assessment will continue to both reduce any uncertainty of determination (false positives) and the potential to miss some issues (false negatives). Conversely, time will cause the feasibility of manual testing for technical vulnerabilities to increase from difficult to impossible as application size and scope increases. In many enterprise organizations, it simply will not be possible to dedicate the time, effort and money required to assess the thousands of web applications that exist. Secondly, relying on human efforts to test for thousands to millions of technical vulnerabilities is subject to error and simply cannot be trusted. Opinions from analyst research firm IDC have concluded that “The issue is scale and cost. Doing a manual review is time consuming and costly. If you get really good people, then it is very secure, but they

³ <http://www.computeruser.com/news/02/02/13/news2.html>

⁴ <https://accountservices.passport.net/reg.srf?roid=2&sl=1&vv=310&lc=1033>

⁵ <http://hackers.org/xss.html>

⁶ <http://www.imperva.com/company/news/2004-feb-02.html>

can only look at so many lines of code a day. With the software scanner, you can get the work done faster, cheaper and cover much more territory.”

LOGICAL VULNERABILITIES

Logical vulnerabilities are those that can be exploited by understanding how an application works and by circumventing the business flow. While both an automated scanning tool and skilled penetration tester can navigate through a web application, only the latter is able to understand “how” the application works and the logic behind the workflow. Understanding the logic and flow of an application allows the manual penetration tester to subvert the business logic and expose a security vulnerability. For example, an application might direct the user from point A to point B to point C, with point B being a security validation check. A manual review of the application might show that it is possible to go directly from point A to point C, bypassing the security validation at point B entirely.

STATISTICS

Based on a recent analysis of 100 websites,⁷ the following statistics were uncovered:

- In 36 percent of websites, manual testing revealed no further vulnerabilities than automated scanners.
- In 17 percent of websites, manual testing revealed all vulnerabilities while the automated scanner found none.
- In 46 percent of websites, the findings of the manual tester and the automated scanning tool were complementary.

Even statistics can be misleading, and the 80 - 20 rule does not necessarily apply. Finding 80 percent of the vulnerabilities is not sufficient if one significant vulnerability is missed - leading to complete server/application compromise.

CONCLUSION

It was noted in the introduction that there are various methods used to discover web application security vulnerabilities. None of these methods are exhaustive in isolation, and each method has its own inherent strengths and weaknesses.

Both manual penetration testing and automated tools can be used to discover critical security vulnerabilities in web applications. Automated tools were never intended to, and should never entirely replace, the manual penetration test. However, if used correctly, automated tools can be used by organizations to find a broad range of technical security vulnerabilities in web applications, saving time and money, with manual penetration testing being used to augment the results for logical vulnerabilities.

Sophisticated organizations will determine the correct mix of automated scanning versus manual penetration testing to provide the best web application security coverage possible.

⁷ <http://www.webappsec.org/lists/websecurity/archive/2005-06/msg00014.html>

ABOUT WATCHFIRE

Watchfire provides Online Risk Management software and services to help ensure the security and compliance of websites. More than 500 enterprises and government agencies, including AXA Financial, SunTrust, HSBC, Vodafone, Veterans Affairs and Dell rely on Watchfire to audit and report on issues impacting their online business. Watchfire has been the recipient of several industry honors including the HP/IAPP Privacy Innovation Award, *InfoSecurity Product Guide's* Hot Security Company 2006, *Computerworld's* Innovative Technology Award, "Recommended" rating by *Computer Reseller News*, finalist in *SC Magazine Awards* 2006. Watchfire was named by IDC as the worldwide market share leader in web application vulnerability assessment software. Watchfire's partners include IBM Global Services, PricewaterhouseCoopers, TRUSTe, Microsoft, Interwoven, EMC Documentum and Mercury. Watchfire is headquartered in Waltham, MA. For more information, please visit www.watchfire.com.