Analysis and Assessment of Web Application Security Testing Tools

Supriya Gupta¹ and Lalitsen Sharma²

¹²Dept. of Computer Sc. & IT, University of Jammu, Jammu-180 006, India
¹mangotwin22@gmail.com and ²lalitsen@yahoo.com

EXTENDED ABSTRACT

1. INTRODUCTION
The increasing number of web based attacks, which result in loss of data and unauthorized access to application, has drawn the attention of organizations to take various measures to secure their web applications. Web Application Vulnerability Assessment [1] is one such measure that involves identifying in applications’ logic, configurations, and coding, the vulnerabilities that compromise the availability, confidentiality and integrity of data. Various methods have been proposed for testing web applications for vulnerabilities like static analysis[2], automated testing tools [3], Taint analysis[4] etc. The most commonly employed mechanism for testing web applications for vulnerabilities is to use automated security testing tools like Web Application Scanners. A large number of open source as well as commercial web application scanning tools are available. These tools are commonly based on blackbox testing approach that crawls an application and locate application layer vulnerabilities, either by manipulating HTTP messages or by generating special input test cases and inspecting them for suspicious attributes. The effectiveness of the tool used for assessment is substantial for securing applications. In this paper we examined various web application vulnerability scanning tools and evaluate their usefulness in assessing web application security, in accordance with the OWASP Web Application Security Scanner Evaluation Criteria (WASSEC)¹. To conduct the study we developed a custom test application containing technical and logical vulnerabilities and the web application scanners were evaluated on their ability to effectively check the test application and identify potential vulnerabilities. The paper also discusses various issues regarding the use of vulnerability scanners as a protection against application layer attacks.

2. WEB APPLICATION VULNERABILITY SCANNERS
Web application security testing is checking the application to find out the vulnerabilities so as to improve the quality of these applications before deployment. There are vulnerabilities of different types, which can be used for evaluation of web application scanners. Web application vulnerabilities are grouped into two categories: 1) technical vulnerabilities [5] that are technical errors, such as improper input validation, or failure to properly filter database queries etc., and 2) logical vulnerabilities [5] that are the result of faulty application logic. Technical vulnerabilities include cross-site scripting (XSS), injection flaws and buffer overflows. Logical vulnerabilities are security gaps that can be exploited by understanding how an application works and circumventing the typical business flow. A web application scanner is an automated program that examines web applications for security vulnerabilities [6]. A test application development for evaluating scanners must differentiate web application scanners based on their vulnerability detection capabilities. An appropriate choice of vulnerability types is therefore very important. Vulnerabilities within one type differ significantly in terms of difficulty of exploiting them and types of attacks that are effective against them. A web application scanner may be able to find one SQL injection vulnerability, but fail to detect another. The test application that we developed contains both technical as well as logical vulnerabilities and some commercial and open source scanners like IBM Rational AppScan [9], WebInspect [8], Acunetix WVS [7] were tested to evaluate their usefulness in assessing web application security.

3. ANALYSIS
In the present study we have tested four web vulnerability scanners that are widely used. HP WebInspect [8] performs web application security testing and assessment for today's complex web applications, built on emerging Web 2.0 technologies. HP WebInspect delivers fast scanning capabilities, broad security assessment coverage and accurate web application security scanning results. This tool includes pioneering assessment technology, including simultaneous crawl and audit (SCA) and concurrent application scanning. It is a broad application that can be applied for penetration testing in web-based applications. WebInspect's scanning is based on a database of vulnerability signatures as well as some proprietary techniques labeled as "artificial intelligence". Because WebInspect is a closed source product we were unable to verify whether they actually use artificial intelligence techniques.

IBM Rational AppScan [9] is a leading suite of automated Web application security and compliance assessment tools that scan for common application vulnerabilities. This tool is suitable for users ranging from non-security experts to advanced users that can develop extensions for customized scanning environments. IBM AppScan can be used for security testing in web applications, including web services. Acunetix Web Vulnerability Scanner[7] is an automated web application security testing tool that audits web applications by checking for exploitable hacking vulnerabilities. Besides web...
services, Acunetix Web Vulnerability Scanner can be applied for security testing in web applications in general. WebScarab [10] is an enterprise level web application vulnerability scanner written in Java. The tool will be able to automatically spider a web site finding potentially vulnerable web applications and then dynamically build a set of security tests for problems based on potential scenarios it finds. Types of problems will include SQL Injection, Cross Site Scripting, Cookie Poisoning and Parameter Tampering. This tool will also have an interactive proxy for manual examination as well as using the VulnXML format for 1,000’s of static checks[10].

4. RESULTS AND DISCUSSION

Due to space constraints we do not present the detailed results. Our experiment showed that nearly all scanners identify technical vulnerabilities but they are not efficient at finding logical vulnerabilities. Logical vulnerabilities are specific to a particular web application, and might be domain specific. For example, the test application that we developed is an online store application that allows users to use coupons to obtain a discount on certain items. In principle, a coupon can be used only once, but an error in the implementation of the application allows an attacker to apply a coupon an arbitrary number of times, reducing the price to zero. Since business logic errors aren't strict syntactical slip-ups, they often require some creative thought to spot. Moreover these scanners are based on vulnerability specifications databases. None of these databases can ever contain all vulnerabilities. It adds to the problem if the databases are closed and it is impossible to be sure which vulnerabilities are included. New attacks will surface all the time and some attacks are specific to the application being tested. That’s why scanners aren’t highly effective at finding such problems, so these problems need to be identified by a knowledgeable expert performing a vulnerability assessment.

CONCLUSION

Based on the experiment and the facts discussed above it is concluded that the security of web applications with these tools is not enough. All of the tools assessed take a black-box approach to testing. They do not know the inside mechanisms employed by the application. A security expert doing a traditional audit can have inside information of the application and an attacker can have it as well. The current offering of tools clearly lacks a white-box web application security testing tool that would make use of inside knowledge of the application.

ACKNOWLEDGMENT

The authors are thankful to University Grants Commission (UGC), and Ministry of Human Resource Development (MHRD), Government of India for providing financial assistance to carry out this research work. The authors are also thankful to Prof. Devanand, Head, Department of Computer Science and IT, University of Jammu, for his kind support.

REFERENCES