SELF ASSESSMENT FRAMEWORK FOR DETECTING VULNERABILITY IN WEB APPLICATIONS

Nor Fatimah Awang, Azizah Abd Manaf
Advanced Informatics School (UTM AIS)
UTM International Campus
Kuala Lumpur, Malaysia
norfatimah@upnm.edu.my, azizah07@ic.utm.my

ABSTRACT

Security Assessment is widely used to audit the security protection of web applications. However, it is often performed by outside security experts or third party that has been appointed by the company. The problem appears when the assessment involves highly confidential areas that might impact company's privacy data which directly reveal the important information to the third party. Even though they might have signed an agreement of non-disclosure information, but as they have already had the information on the infrastructure and architecture regardless of the confidential data, it has to be considered as a high risk. It is important to keep the information within the project members to protect the confidential data used by the system. Therefore, due to confidentiality level of the system, we proposed Self-Assessment framework to conduct security assessment internally to ensure the safety of all the assets of the organization. The main objective of this paper is to discuss the activities and processes involve in conducting security assessment.

KEYWORD

Web application, vulnerability, security testing, security assessment, penetration testing

1 INTRODUCTION

Today, more than one billion people worldwide using the Internet as their daily routine for a variety of reasons, such as communicating with others, conduct research, shopping, banking and electronic commerce [1]. Due to the popularity of internet, some organizations make efforts to change their manual systems into dynamic web applications to make more profit. Several studies indicate that because of the popularity of internet and web application, these services attracted the attention of attackers. The Gartner Group estimates that over 70% of attacks today are focused at application layer. In fact, Symantec Group reported, attack against web has been increased in 2010 by 93% compared to 2009. Another report shows that as of May 2012, almost 150,000 new sites are registered per day on internet [2], which potential to introduce around two billion serious vulnerabilities [3]. In computer security, the term vulnerability is applied to a weakness in a system, which allows an attacker to violate the integrity of that system. Vulnerabilities may result from weak passwords, software bugs, software misconfigurations, a computer virus or other malware (malicious software), a script code injection, or a SQL injection and etc [4]. Web applications vulnerabilities can be exploited by the attackers to gain unauthorized access, obtain or modify sensitive data or even perform denial of service attacks. A lot of vulnerabilities appear in web applications make system or networks administrator more difficult to protect core assets such as personnel information, confidential data and customer credit card numbers. To cope with these threats, several techniques have been developed to prevent and detect the potential security loopholes. Such techniques can be used during the development phase and also during the testing phase. Detail of previous techniques will be discussed further in section 3.

Based on the above scenario, security is a big issue that should be seriously considered by the system administrator as well as top management [5], [6], [7] in order to protect the potential assets and target systems. This paper will discuss the Self-Assessment Framework (SA) where SA has become part of a techniques to detect vulnerabilities appear in the system. SA is a process to find all potential security loopholes or vulnerabilities in target systems. The importance of SA is to make sure all systems are safe and all vulnerabilities are discovered before deployed [8], [9].

The structure of this paper is as follows. Section 2 briefly described problem background and issues that discover in preventing and detecting vulnerabilities from web application. Section 3 discusses more related techniques that have been developed to prevent and detect the security flaws appeared in web application. Section 4 describes detail on the proposed framework and finally, Section 5 presents the conclusion.

2 PROBLEM BACKGROUND

According to [3] the most prevalent vulnerabilities appear in websites are cross site scripting and information leakage vulnerabilities with about 55% and 53% successful to be compromised at least once. Looking at these statistics, in Table 1, we see that most of the important sectors are underperforming in protecting their digital assets, despite those assets' sensitive nature. Even when vulnerabilities are addressed, the fixes might be incorrect or incomplete.

TABLE 1. The average number of serious vulnerabilities per website in 2011[3]

Sector	Avg. no. of vulnerabilities
Banking	17
Education	53
Financial Services	67
Healthcare	48
IT	85
Telecom	52
Retail	121

In order to secure the websites, one of the popular technique to detect vulnerability is a security assessment or well-known as vulnerability assessment. It is important to do assessment on the system to make sure that it will be safely release and not offer any illegitimate access that can affect availability, confidentiality and integrity of the system [10]. Some companies choose to use consultants or outsource to third party to perform vulnerability assessment. Outsource a vulnerability assessment is mandatory security audit for banking and online business industry, a software industry for the related business can just concentrate on developing their system and let a third party evaluating their product before releasing it to the market. However, according to study conducted by Corwill and Nassimbeni et al, shows that there are some security issues involved when we use external party to conduct an assessment [11], [12]. Therefore, SA is proposed to assess and find out what systems have flaws internally, detect vulnerabilities and take action in order to mitigate the risk before it goes to production. Using SA framework, many organizations have an opportunity to do vulnerability assessment by itself without outsources to third party or security expert.

3 TECHNIQUES TO DETECT VULNERABILITY

The first approach before conducting security assessment is to identify and analyse the different techniques that are currently used for web applications. There are many techniques in order to detect vulnerabilities during the process of software development life cycle such as:

- Static Analysis
- Dynamic Analysis
- Security assessment

3.1 Static Analysis

Static Analysis consists of the analysis of the source code of the application [13], [14], [15], [16]. It's performed on the source code without executing the application. This can be done manually or by using code analysis tools like FORTIFY, Ounce or Pixy [17]. After reviewing the source code, report will be generated and presented to the developer team. Generally it helps to catch implementation structural bugs early and it's important to know that static analysis can't solve all security problems. There are different tools available now for this kind of test but it's not easy to find mature tool to discover all the security defects in the application. The problem of this analysis may be difficult and cannot find all security flaws because of the complexity of the code [18].

3.2 Dynamic Analysis

Dynamic Analysis is also known as Dynamic Testing, is used to test a program by executing it in real-time [20]. Dynamic Analysis test will communicate with a web application through the web browser in order to identify potential security vulnerabilities and architectural weaknesses in the web application. The objective is to find security errors in a web application while it is running. This technique can be performed either manually or by using automated tools [19]. Automated tool provides an automatic way to search for vulnerabilities avoiding the repetitive and tedious task of doing hundreds or even thousands of tests by hand for each vulnerability type [21].

3.3 Security Assessment

Security assessment or also known as vulnerability assessment is a process of identifying and quantifying vulnerabilities in an application. In security assessment, there are some standards and testing guidelines that openly use and publicly available on the internet such as Open Source Security Test Methodology Manual (OSSTMM) [23], Open Web Application Security Project (OWASP) [22], Information Systems Security Assessment Framework (ISSAF) [24] and Payment Card Industry Data Security Standards (DSS) [25]. It is an in-depth evaluation of web applications or valuable systems, indicating weaknesses as well as providing the appropriate mitigation procedures required to either eliminate those weaknesses or reduce them to an acceptable level of risk.

4 THE PROPOSED FRAMEWORK

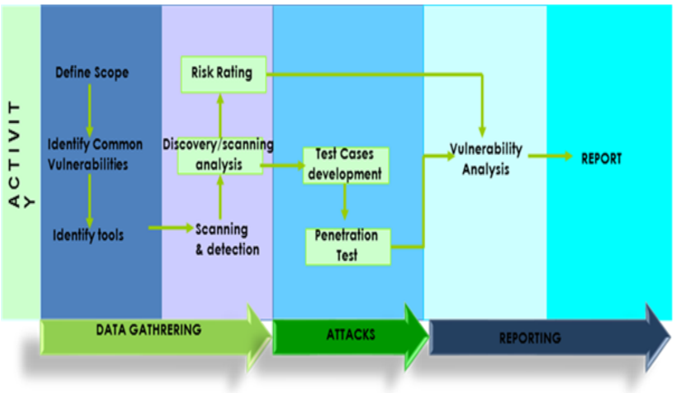

Figure 1. Framework for detecting vulnerability in web application

This section will describe the phases and activities of the proposed framework as shown in Figure 1. The three main phases are Data Gathering, Attacks, and Reporting. In each phase comprises of several major activities together with flow and stages:

- Data Gathering – Phase 1
  This is the first stage in the framework. There are six major activities involved in this stage. The first three stages are basically planning focused activities. In this phase, there are some issues that should be highlighted and prepared such as identify which target system that should be tested for detect vulnerability, what type of potential threat or vulnerability that commonly exists in web application (as shown in table 2), how long the testing will be carried out, which methodology that will be used and what restrictions or limitations need to be applied. The test plan should also outline the tools needed to conduct the tests, as well as any opportunities for automated testing. Find other criteria in Table 3.

TABLE 2. Potential Vulnerability based on OWASP Top Ten 2010 [22]

Ranking	Vulnerability
A1	Injection
A2	Cross site scripting (XSS)
A3	Broken Authentication /

	Session Management
A4	Insecure Direct Object References
A5	Cross Site Request Forgery
A6	Security Misconfiguration Sensitive Data Exposure
A7	Insecure Cryptographic Storage
A8	Failure to Restrict URL Access
A9	Insufficient Transport Layer Protection
A10	Unvalidated Redirects and Forwards

TABLE 3. Test Planning Criteria

Criteria	Planning Detail
No. of Security Tester and Qualification	To get the number of certified security tester and the tester unified qualification
Type of Tools	To see if they use open source tools available on the net or use commercial tools
Number of Server	How many servers will be involved in this assessment
Test Time Frame	How long the duration for this assessment

Another three activities as discussed below, will be more hands-on and mostly based on the first three activities data gathering and findings.

Scanning - This phase is more on mapping of the potential vulnerabilities detected by scanners with main systems' components. This activity uses the vulnerability scanner to scan the services to identify potential vulnerabilities in web application.

Discovery Scanning Analysis - In this activity, results produced by different tools will be compiled for further analysis purposes.

Risk Rating - In this activity, discovery analysis findings will be used as main source and subject in risk rating. The risk rating outcomes or results are more specific to the assessed system.

- Attacks – Phase 2
  This is the second stage in the framework. As the name suggests, it is responsible for performing the attacks on the system. The attacks are performed on the vulnerabilities that have been discovered through the data gathering phase. The attack phase is completed in a cascaded manner where every successful attack leads to obtaining more privileges and system information. There are two major activities involved in this phase, which are test cases development and penetration testing.

Test Cases - Structured test cases will be developed based on the weaknesses that have been discovered through scanning phase and added with new test cases created based on the OWASP testing guidelines as shown in Table 4.

TABLE 4. OWASP Test Cases [21]

Test Criteria	Number of Test Cases
Information Gathering	6
Configuration Management Testing	8
Authentication Testing	10
Session Management	5
Authorization Testing	3
Business logic testing	1
Denial of Service Testing	8
Data Validation Testing	16
Web Services Testing	7
Ajax Testing	2

Penetration Testing - This stage performs manual penetration testing to confirm vulnerabilities that have been detected through scanning phase (to check false positive of the vulnerabilities). Some test cases that have been created in test cases development section are also executed to discover new vulnerabilities.

- Reporting – Phase 3
  This final stage in the framework will conclude the assessment from the combination of two main phases – data gathering and attacks. Vulnerability analysis result will be based on

two activity's results from two different phases. Mapping the risk rating conducted in phase 1 and validation of penetration testing in phase 2 is the major source in vulnerability analysis. Once complete, the report will be provided with identifying all of the security vulnerabilities found. Each finding will be assigned a risk rating based on the following criteria, along with remediation recommendations to resolve the vulnerability.

5 CONCLUSION

This paper aims to provide web security assessment frameworks for in house self-assessment exercise in which will help to identify the weaknesses and potential vulnerabilities to web application. Using OWASP Top Ten vulnerabilities classification as main reference or guidelines to seek security holes in web applications and simulate hackers' actions via specific test cases to validate the real existence of vulnerabilities. The overall methodology is relatively straightforward; it covers initial and full system scanning, discovery analysis, risk rating, test cases development, penetration test and reporting based on analysed vulnerabilities.

REFERENCES

1. Vermaat. S., : Discovering Computers 2009 – Complete, Cengage Learning Course Technology, (2009).
2. Dell, J. O., : How Big is the Web & How Fast Is it growing?, http://mashable.com/2011/06/19/how-many-websites/#17193Who-Registers-the-Webs-Domains
3. Jeremiah, G., : The State of Website Security, Security & Privacy, IEEE , vol.10, no.4, pp.91-93, (2012).
4. Su, Z., Wassermann, G., : The Essence of Command Injection Attacks in Web Applications , In Conference Record of the 33rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, pp. 372-382 (2006)
5. Christey, S., Martin, R. A., : Vulnerability type distributions in CVE, V1. 0, vol. 10, pp. 04, (2006)
6. Zanero, S., Carettoni, L., Zanchetta, M., : Automatic Detection of Web Application Security Flaws, Black Hat Forum, (2005).
7. Trustewave, The Trustwave 2012 Global Security 2012, https://www.trustwave.com/spiderlabs, (2012).
8. Ahmad, A., Ahmad, S.R., Awang, N.F., Ali, Z.M., : Web Vulnerability Assessment: Outsource dilemmas, Electrical Engineering and Informatics (ICEEI), 2011 International Conference , vol., no., pp.1,6, (2011).
9. Xiong, P., Peyton, L., : A model-driven penetration test framework for Web applications," Privacy Security and Trust (PST), 2010 Eighth Annual International Conference on , vol., no., pp.173-180, (2010).
10. Meier, J.D., Mackman, A., Dunner, M., Vasireddy, S., Escamilla, R., Murukan, A.,: Improving Web Application Security: Threats and Countermeasures, Microsoft Corporation,http://msdn.microsoft.com/en-us/library/aa302420.aspx, (2003).
11. Colwill C., Gray A., : Creating an effective security risk model for outsourcing decisions; BT Technology Journal, Vol. 25, No. 1, pp. 79-87, (2007).
12. Nassimbeni, G., Sartor, M., Daiana, D., : Security risks in service offshoring/outsourcing: an assessment model based on the Failure Mode and Effect Analysis. POMS 21st Annual Conference, Vancouver, Canada, (2010)
13. Jovanovic, N., Kruegel, C., Kirda, E., : Static analysis for detecting taint-style vulnerabilities in web applications, Journal of Computer Security, pp. 861-907, (2010).
14. Xie, Y., Aiken, A., : Static detection of vulnerabilities in scripting languages, Proc. 15th USENIX Security Symposium, pp. 179-192, (2006)
15. Nuno, A, Marco, V., : Comparing of Effectiveness of Penetration Testing and Static Code Analysis on the Detection of SQL Injection Vulnerabilities in Web Services. 15th IEEE Pacific Rim International Symposium on Dependable Computing, (2009).
16. Y. Huang, F. Yu, C. Hang, C. H. Tsai, D. T. Lee, and S. Y. Kuo. : Securing Web Application Code by Static Analysis and Runtime Protection. In Proceedings of the 12th International World Wide Web Conference (WWW 04), (May 2004).
17. Ayewah, N., Hovemeyer, D., Morgenthaler, J.D., Penix, J., Pugh, W., : Using Static Analysis to Find Bugs, IEEE Software, pp 22-29. (2008).
18. Vieira, M., Antunes, N., Madeira, H., "Using Web Security Scanners to Detect Vulnerabilities in Web Services", IEEE/IFIP Intl Conf. on Dependable Systems and Networks, DSN (2009).
19. Curphey, M., Araujo, R., : Web Application Security Assessment Tools, IEEE Security & Privacy, Published By The IEEE Computer Society, (2006).
20. Basaval, R.S., : Web application vulnerability detection using dynamic analysis with penetration testing, International Journal of Enterprise Computing and BusinessSystems, Vol 2, (2012).
21. OWASP Testing Guideline, https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents.
22. The Open Web Application Security Project: The Ten Most Critical Web Application Security Vulnerabilities. https://www.owasp.org/index.php/Main_Page:OWASP_Top_ Ten_Project
23. Open-Source Security Testing Methodology Manual, http://isecom.securenetltd.com/osstmm.en.2.1.pdf. March 2013
24. http://www.oissg.org/
25. Payment Card Industry Data Security Standards (PCI DSS),https://www.pcisecuritystandards.org/security_standards/pci_dss.shtml.