ISBN 978-93-84468-20-0
Proceedings of 2015 International Conference on Future Computational Technologies (ICFCT'2015)
Singapore, March 29-30, 2015, pp. 34-40

Analysis and Study of Vulnerability Scanning Result for Projects within Organization

Nor Izyani Daud[1], and Khairul Azmi Abu Bakar[2]

[1, 2]Information System Security Lab, MIMOS Berhad, 57000 Kuala Lumpur

Abstract: *This paper attempts to share about analysis and study of vulnerability scanning result for projects in our organization. For the case study, we collected vulnerability scanning result from projects in our organization for year 2013 and 2014. From the result, we did an analysis and study; and come out with a classification for the errors. A brief description about the classification for the errors is included in this paper. Referring to the result and classification, we elaborate the reason how the error may occur during our scanning based from our experience. A suggestion on how to solve the issue is discussed in detail.*

Keywords: *vulnerability analysis, vulnerability scanning, security scanning, security scanning result.*

1. Introduction

Referring to Wikipedia [1], vulnerability is a weakness that allows attacker to reduce a system's information assurance. According to a report from Microsoft, dated July 2011, titled 'Mitigating Software Vulnerabilities'[2], vulnerabilities are weaknesses in computer programs that provide a capable attacker with opportunities to compromise the integrity, availability or confidentiality of an affected user's computer or data. MSDN[3] defines vulnerability as a weakness in a product that could allow attacker to compromise the integrity, availability or confidentiality of that product. From these 3 references, we can summarize that vulnerability is a weakness of a system or an application that allows attacker to compromise the integrity, availability or confidentiality of a system or an application.

According to OWASP[4], the efficient method of finding security vulnerabilities on web application is through manual code review. However, it requires an expert in secured programming to analyze and review the source code. The alternative method to detect vulnerabilities is by doing security scanning and penetration test. In this method, user is required to use vulnerabilities scanning tool for example Nessus[12], Acunetix[6] and Cain & Abil[13].

Our organization has been doing security scanning for our internal project since 2010[5]. We have been using Nessus and Acunetix to perform security scanning [6].

The remaining of this paper is organized as follows. Section II describes about the case study in this paper. It includes the vulnerability scanning result that we have collected for the year 2013 and 2014. Then, we classify the vulnerability scanning result. We analyze the vulnerability scanning result to see the pattern of the error detected. Detail explanation about the cause of the vulnerability is explained in Section III. Section IV describes about the process improvement that could be done to overcome the issue. Finally, Section V draws the conclusion.

2. Case Study

2.1. Scanning result and classification

In this paper, we analyze vulnerability scanning result that we have collected for projects in our organization for year 2013 and 2014. All projects that we use in our case study are web based applications where there is at least one web server and one database for the application. As per mentioned in paper title 'A Case Study on Web Application Vulnerability Scanning Tools' [6], we have been using Nessus as our scanning tool.

We classify the scanning results into several categories as the following:

- Configuration
  It is related to the server or application configuration. Example of the error is when the port required for the server is not opened or there is unnecessary open port detected by the scanning machine.
- Programming
  Programming issue related to the source code of the application. This error is about non-secure coding; for example validation for input parameter. Examples of the error are SQL Injection, CGI and Cross Site Scripting.
- Security
  Security issue related to the application, software and certificate that are used. Examples of this type of error are certificate, default credential user for Apache Tomcat, MySQL and weak SSL cipher suite.
- Software
  It is related to operating system or any software installed at the server or in the application. Example of a software error is when the software installed for the application is not the latest version; MySQL, Apache Tomcat or PHP version.

Table I shows a summary for vulnerability scanning result and the classification for projects that we scanned in our organization for the year 2013. In 2013, we run vulnerability scanning on 10 projects. In all of those 10 projects, we detected 63 vulnerabilities or errors.

TABLE I. SUMMARY OF VULNERABILITY RESULT FOR PROJECTS - 2013

Vulnerability Detected/Error	Number	Classification
Apache 2.2 < 2.2.16 Multiple Vulnerabilities, Apache 2.2 < 2.2.17 Multiple Vulnerabilities, Apache 2.2 < 2.2.22 Multiple Vulnerabilities, Apache 2.2 < 2.2.23 Multiple Vulnerabilities, Apache 2.2 < 2.2.25 Multiple Vulnerabilities	5	Software
Apache 2.2 < 2.2.18 APR apr_fnmatch DoS	2	Software
Apache 2.2 < 2.2.21 mod_proxy_ajp DoS	1	Software
Apache 2.2 < 2.2.24 Multiple Cross-Site Scripting Vulnerabilities	1	Software
Apache Shiro URI Path Security Traversal Information Disclosure	1	Software
Apache Tomcat 6.0.x < 6.0.36 Multiple Vulnerabilities, Apache Tomcat 6.0.x < 6.0.37 Multiple Vulnerabilities	2	Software
Apache Tomcat servlet/JSP container default files	2	Software
CGI Generic Command Execution (time-based)	2	Programming
CGI Generic SQL Injection (blind, time based)	2	Programming
Extra Port	7	Configuration
HTTP TRACE / TRACK Methods Allowed	3	Security
Jenkins < 1.514 / 1.509.1 and Jenkins Enterprise 1.466.x / 1.480.x < 1.466.14.1 / 1.480.4.1 Multiple Vulnerabilities	1	Software
MySQL Protocol Remote User Enumeration	1	Security
nginx < 1.0.10 ngx_resolver_copy Function DNS Response Parsing Buffer Overflow	1	Software
nginx < 1.0.14 / 1.1.17 HTTP Header Response Memory Disclosure	1	Software
nginx on Windows Directory Aliases Access Restriction Bypass	1	Software
OpenSSH < 5.7 Multiple Vulnerabilities	5	Software
OpenSSH 6.2 and 6.3 AES-GCM Cipher Memory Corruption	4	Software
OpenSSH LoginGraceTime / MaxStartups DoS	5	Software
PHP expose_php Information Disclosure	1	Software
phpMyAdmin 3.4.x < 3.4.10.1 Cross-Site Scripting (PMASA-2012-1)	1	Software

phpMyAdmin 3.4.x < 3.4.8 Cross-Site Scripting (PMASA-2011-18)	1	Software
phpMyAdmin 3.4.x < 3.4.9 Cross-Site Scripting (PMASA-2011-19 and PMASA-2011-20)	1	Software
phpMyAdmin Replication Setup js/replication.js Database Name XSS	1	Software
phpMyAdmin simplexml_load_string() Function Information Disclosure (PMASA-2011-17)	1	Software
Port Not Detected	3	Configuration
PostgreSQL Default Unpassworded Account	1	Security
SSL Certificate Cannot Be Trusted	2	Security
SSL Medium Strength Cipher Suites Supported	1	Security
SSL Self-Signed Certificate	1	Security
Trojan Horse Detection	1	Security
Unsupported Unix Operating System	1	Software

Table II shows a summary for vulnerability scanning result and the classification for projects that we scanned in our organization for the year 2014. In 2014, we run vulnerabilities scanning on 9 projects. In all those 9 projects, we detected 83 vulnerabilities or errors.

TABLE II. SUMMARY OF VULNERABILITY RESULT FOR PROJECTS - 2014

Vulnerability Detected/Error	Number	Classification
Apache 2.2 < 2.2.16 Multiple Vulnerabilities, Apache 2.2 < 2.2.17 Multiple Vulnerabilities, Apache 2.2 < 2.2.22 Multiple Vulnerabilities, Apache 2.2 < 2.2.23 Multiple Vulnerabilities	8	Software
Apache 2.2 < 2.2.18 APR apr_fnmatch DoS	2	Software
Apache 2.2 < 2.2.21 mod_proxy_ajp DoS	2	Software
Apache 2.2 < 2.2.24 Multiple Cross-Site Scripting Vulnerabilities	1	Software
Apache 2.2 < 2.2.24 Multiple XSS Vulnerabilities	1	Software
Apache 2.2 < 2.2.25 Multiple Vulnerabilities, Apache 2.2 < 2.2.27 Multiple Vulnerabilities, Apache 2.2 < 2.2.28 Multiple Vulnerabilities, Apache 2.4 < 2.4.8 Multiple Vulnerabilities	6	Software
Apache Shiro URI Path Security Traversal Information Disclosure	1	Software
Apache Tomcat 7.0.x < 7.0.30 Multiple Vulnerabilities, Apache Tomcat 7.0.x < 7.0.50 Multiple Vulnerabilities, Apache Tomcat 7.0.x < 7.0.53 Multiple Vulnerabilities	7	Software
Apache Tomcat 7.0.x < 7.0.32 CSRF Filter Bypass	1	Software
Apache Tomcat 7.0.x < 7.0.33 Session Fixation	1	Software
Apache Tomcat 7.0.x < 7.0.52 Content-Type DoS	2	Software
Apache Tomcat 7.0.x < 7.0.55 Multiple OpenSSL Vulnerabilities	1	Software
Apache Tomcat Manager Common Administrative Credentials	1	Security
Apache Tomcat servlet/JSP container default files	1	Software
ASP.NET DEBUG Method Enabled	1	Software
Backup Files Disclosure	1	Software
CGI Generic Cross-Site Scripting (persistent, 3rd Pass)	2	Programming
CGI Generic Cross-Site Scripting (Parameters Names)	1	Programming
CGI Generic SQL Injection (blind)	1	Programming
CGI Generic Unseen Parameters Discovery	1	Programming
Extra port detected	1	Configuration
HTTP TRACE / TRACK Methods Allowed	1	Programming
OpenSSH < 4.4 Multiple Vulnerabilities	1	Software
OpenSSH < 4.5 Multiple Vulnerabilities	1	Software
OpenSSH < 4.7 Trusted X11 Cookie Connection Policy Bypass	1	Software
OpenSSH < 4.9 'ForceCommand' Directive Bypass	1	Software
OpenSSH < 5.2 CBC Plaintext Disclosure	1	Software
OpenSSH < 5.7 Multiple Vulnerabilities	3	Software
OpenSSH < 6.6 Multiple Vulnerabilities	2	Software
OpenSSH 6.2 and 6.3 AES-GCM Cipher Memory Corruption	2	Software
OpenSSH LoginGraceTime / MaxStartups DoS	2	Software
OpenSSH X11 Forwarding Session Hijacking	1	Software
Oracle TNS Listener Remote Poisoning	1	Software
PHP 5.4.x < 5.4.12 Multiple Vulnerabilities, PHP 5.4.x < 5.4.16 Multiple Vulnerabilities, PHP 5.4.x < 5.4.18 Multiple Vulnerabilities, PHP 5.4.x < 5.4.24 Multiple Vulnerabilities	4	Software
PHP 5.4.x < 5.4.13 Information Disclosure	1	Software
PHP 5.4.x < 5.4.17 Buffer Overflow	1	Software

PHP 5.4.x < 5.4.23 OpenSSL openssl_x509_parse() Memory Corruption	1	Software
Port not detected	2	Configuration
SQL Dump Files Disclosed via Web Server	1	Security
SSL Certificate Cannot Be Trusted	4	Security
SSL Medium Strength Cipher Suites Supported	3	Security
SSL Self-Signed Certificate	1	Security
SSL Version 2 (v2) Protocol Detection	1	Security
SSL Weak Cipher Suites Supported	2	Security
SSL/TLS Protocol Initialization Vector Implementation Information Disclosure Vulnerability	1	Software
Unsupported Unix Operating System	1	Software

2.2. Analysis

Table I and Table II illustrate the vulnerability scanning results that have been matched with the defined classification. We then summarized the result in the form of a graph to see the pattern for type of error for the year 2013 and 2014. The graph as shown in Figure 1 shows the result from the classification.

In Figure 1, we may see that software is the highest classification of error that almost all of the projects had in year 2013 and 2014. This type of error has been increased 56% from 39 to 61 errors for those two years. Security is the second highest type of error that we detected in the vulnerabilities security scanning in year 2013 and 2014. This type of error also has been increased from 10 in year 2013 to 13 in year 2014. However, for configuration error, the number has been decreased 70% from 10 in year 2013 to 3 in year 2014. For programming, the number of error has been increased from 4 in year 2013 to 6 in year 2014.

Figure 1 below illustrates the type of error based on classification for projects in our organization for the year 2013 to 2014.

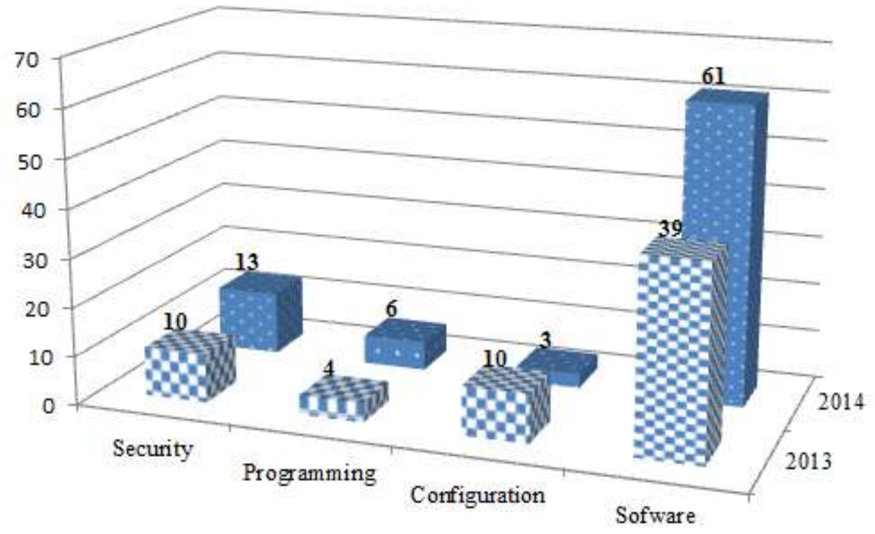

Fig. 1: Type of error based from classification.

3. What goes wrong?

Based from the analysis in Figure 1, we made a study and did an interview with the project and support team about errors that occurred during vulnerability scanning. Issues that have been highlighted are:

- Default configuration for port

  In our organization, most of the projects are developed based on Virtual Machine (VM). Each time a project team requests a new VM from Central Engineering (CE) department for their project, CE will clone the VM that they have and provide it to the project team.

CE department has their own set of VMs as their master copy. All of the VMs use the default or standard configuration of the opening port. CE does not specify or configure any required open port for any project.

Once project team receives the VM, they will start to build their application. Almost all of them did not bother to check or configure their port setting of their VM. Based on our process flow policy [5], all projects that need to be a public accessed domain server, requires a security team to perform vulnerability scanning. At this stage only, security team can detect either if there is any unnecessary port opened or port that is required for the application is not opened yet.

- Software version

  The project team will start to build their application once they receive the VM from CE team. Some of the projects may take some time to complete. At the time when project is ready for the vulnerability scanning, some of the software that had been installed in their application or operating system may not be the latest version. Based from data that we collected, every project that we scanned has at least one of the software versions issue. This is one reason why error in software always the highest from the others.

- Invalid certificate

  Project team uses (Secure Socket Layer) SSL protocol to secure connection to the server from their client site. To have SSL connection, system needs to have a valid digital certificate. The common problem about the digital certificate is they usually use invalid certificate; for example self-signed certificate and wildcard certificate.

  Self-signed certificate [7] is an identity certificate that is signed by the same entity whose identity it certifies; meaning is one signed with its own private key. Wildcard certificate is a public key certificate where you can use this certificate at multiple subdomains of a domain [8].

  By using invalid certificates, more security issues can be exploited. Man in the middle attack is one of the attacks that can be exploited when the user uses untrusted certificate or self-signed certificate [9]. For wildcard certificate, attacker can hack multiple subdomains using the certificate once the public key for wildcard certificate is compromised.

- Use default username and password for software

  People always use a username and password that is easy for them to remember. Sometimes, they also use default username and password for software. One of the errors that we detected during vulnerability scanning is the users still using a default username and password for Apache Tomcat administrative. Based on one of our vulnerability scanning result, they use ‘admin’ as their username, and ‘password’ for the password. By using default username and password for Apache Tomcat administrative, attacker can make an easy guess and might be able to install any application or remove any application from the web server.

- Include example, info or default file container

  Another common mistake from our security analysis is the users include the example, file or default file container for PHP, Apache Tomcat servlet or JSP container in their application. One of the errors that we detected is the users include Web Server info.php file into their application. The file contain a function call, “phpinfo()”. This function will show system information; such as the IP address of the host, version of the operating system, web server version and root directory of the web server. With this information, it will be easy for an attacker to find out and dig more information related to the application.

- Use weak security configuration

  In some cases, the project team, did not put much effort to look at the security configuration that they are using. For them, as long as the application is working, it is good enough for them. However, when we perform vulnerability scanning, Nessus can detect error if the application still using weak security configuration. An example of this issue is SSL weak cipher that the application uses. By having a SSL weak cipher suite, the attackers can easily exploits the encryption if they are on the same physical network.

- Programming
  Source code is the heart of an application. One of the errors that we detected is about un-secure code that the developer provided. This error usually happens if the developer is not doing a proper checking for lines of their code; for example input validation and SQL statement.

4. Solving issues

Several actions have been taken by our organization to avoid or reduce vulnerabilities that we could detect during security scanning. The actions are as the following:

- Always check for latest or updated information about software
  As a project team, you need to know the latest update information about the software or product that you are using for your current project. Usually, new version is released when there is bug or vulnerability fixed. New version also may contain new features that are supported by particular software. One of the easiest ways to get the latest updated information about the software is by subscribing to the product information. For example, Apache Tomcat user can subscribe latest information about Apache Tomcat by sending email to users-subscribe@tomcat.apache.org. It is advisable for the project team to subscribe to the products that they are using in their project to make sure that they will always have the latest update or information.
- Alert with updated security trend
  New security trend is one of the item that project team must be aware of. Latest security trend will cater the latest technology or finding in security area. One example of security trend is people are start moving from SSL (Secure Socket Layer) to TLS (Transport Layer Secure). The purpose of SSL is to establish a secure connection between a web server and a browser. There are several versions of SSL; SSLv1, SSLv2 and latest SSLv3. However, in September 2014, Google discover that SSLv3 has vulnerable to a POODLE attack [10]. This is the reason why users should start moving to TLSv1, TLSv2 and TLSv3.

  Project team must be aware and keep themselves updated with the latest security trend issues. One of the portals that we can use as security reference is OWASP[14]. OWASP is a free and open software security community.
- Not using default username and password for the software
  It is a compulsory for the project team not to use default username and password for the software that they use, especially for open source software. Project team must create personalized username and password combination and not using the default one.
- Remove default or example file.
  Before deploying the application to production, project team should remove any example or default file inside their application. For example info.php file. Example files, such as info.php file, normally contain an executable function such as phpinfo() function that can extract system information like IP address of the machine.
- Buy a proper certificate.
  It is advisable for the project team to purchase a valid digital certificate for each of domain name. Avoid using self-signed certificate. Each of the SSL connection requires a proper digital certificate. It is not secured to use SSL with self-signed.
- Secure programming.
  Secure programming is one of a skill that all programmer should have. A programmer with that skill can produce a better source code. There are many way that we can do to guide the programmer to produce a secure code. The author of an article title ‘Teaching Secure Programming’ [11] suggested using checklist while doing programming. Checklist can be used as an examples and guidance while preparing source code. In our organization, we have implemented checklist for our source code. Currently our organization

is using C-Code Checklist and Java Code Checklist for application that we build using C and Java Language.

5. Conclusion

There are many vulnerability errors exist when we did vulnerability scanning for projects in our organization. Based from the classification, software is the highest security issue that occurred in every project in our organization. Most of the issues are resulting from the human factor; for example project team itself. To minimize the vulnerabilities scanning issue, we have to tackle the human factor. The project team needs to have adequate expertise, knowledge and skill on how to handle and fix vulnerabilities security issues. This requires some effort from them to learn and develop the experience that related to the problems.

The vulnerability issues are not limited to the items that are discussed in this paper. It is highly advisable for the project team to become up to date with the current security issues so that the issues can be fixed accordingly.

6. References

[1] http://en.wikipedia.org/wiki/Vulnerability_(computing), retrieved December 2014.

[2] “Mitigating Software Vulnerabilities”, http://download.microsoft.com/download/5/0/5/505646ED-5EDF-4E23-8E84-6119E4BF82E0/Mitigating_Software_Vulnerabilities.pdf, retrieved December 2014.

[3] http://msdn.microsoft.com/en-us/library/cc751383.aspx, retrieved December 2014.

[4] Curphey, M., Wiesman, A., Van der Stock, A., Stirbei R.:”A Guide to Building Secure Web Applications and Web Services”. OWASP(2015).

[5] Haron, G.R.; Ng Kang Siong; "Extrapolating security requirements to an established software process: Version 1.0," *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for* , vol., no., pp.752-757, 11-14 Dec. 2011.

[6] Daud, N.I.; Abu Bakar, K.A.; Md Hasan, M.S., "A case study on web application vulnerability scanning tools," *Science and Information Conference (SAI), 2014* , vol., no., pp.595,600, 27-29 Aug. 2014 doi: 10.1109/SAI.2014.6918247.

http://dx.doi.org/10.1109/SAI.2014.6918247

[7] Kissell, Joe. "Chapter 27 - Working with SSL Certificates". *Mac Security Bible*. John Wiley & Sons. © 2010. Books24x7. <*http://common.books24x7.com/toc.aspx?bookid=40737*> (accessedDecember 31, 2014)

[8] http://en.wikipedia.org/wiki/Wildcard_certificate, retrieved December 2014.

[9] Callegati, F.; Cerroni, W.; Ramilli, M., "Man-in-the-Middle Attack to the HTTPS Protocol," *Security & Privacy, IEEE* , vol.7, no.1, pp.78,81, Jan.-Feb. 2009, doi: 10.1109/MSP.2009.12

http://dx.doi.org/10.1109/MSP.2009.12

[10] Bodo Möller, Thai Duong, Krzysztof Kotowicz, "This POODLE Bites: Exploiting The SSL 3.0 Fallback", https://www.openssl.org/~bodo/ssl-poodle.pdf, retrieved December 2014.

[11] Bishop, M.; Frincke, D.A., "Teaching secure programming," *Security & Privacy, IEEE* , vol.3, no.5, pp.54,56, Sept.-Oct. 2005, doi: 10.1109/MSP.2005.133

http://dx.doi.org/10.1109/MSP.2005.133

[12] (ed), Russ Rogers. "Chapter 2 – Introducing Nessus". *Nessus Network Auditing, Second edition*. Syngress Publishing. © 2008. Books24x7. <*http://common.books24x7.com/toc.aspx?bookid=32366*> (accessed December 31, 2014)

[13] Beaver, Kevin. "Chapter 8 - Network Infrastructure". *Hacking For Dummies, 3rd Edition*. John Wiley & Sons. © 2010.Books24x7. <*http://common.books24x7.com/toc.aspx?bookid=34819*> (accessed December 31, 2014)

[14] https://www.owasp.org/index.php/Main_Page, retrieved December 2014.