International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

http://ijpsat.ijsht-journals.org
Vol. 7 No. 2 March 2018, pp. 175-182

Agile Web Software Development Security Testing with SQL Injection on Indonesia Higher Education Institution

IGN Mantra, Muhammad, Harya Damar Widiputra
Informatics Engineering, Information Technology Faculty
Perbanas Institute, Jakarta, Indonesia

Abstract **– Web-based software development is now mushrooming all over the world, every company will display information through internet web via published URL, as well as business on the Internet or ecommerce, then software development also keep increasing along with the existing start-up. Agile web software development is the latest web-based application development by combining several things such as: collaborative self-organizing cross-functional team and its customer / end user, involving adaptive planning, evolutionary development, early delivery and continuous improvement as well as encourage rapid and flexible response to change. When a web-based application is published then the application will be loaded by all end users in the world that is not known good or bad intentions, so need to protect the web application from a security interruption. The most nefarious disturbance by using SQL injection to get into the heart of the web server database, if this can be penetrated, and then the database can be misused by the hackers. Equally important is to conduct Agile Web-based application security testing with SQL injection attack against several higher education institutions in Indonesia. The world-renowned and proven OWASP web-based security test form. The outcome research concludes that not all websites have a weakness that may cause unusual information or confidential information being exposed of a result from SQL injection techniques. Furthermore, the research has also found that not all weaknesses that exist on a website can be compromised by the SQL injection techniques. Yet, there are weaknesses or vulnerabilities in each of the target website that is not related to attempt of information disclosure using the SQL injection technique that can still be further applied. Agile software development for security testing was important thing to help the web security services. Several outcomes from this security testing will be modeled for the preparation of security testing in higher education in Indonesia.**

Keywords **– Agile, Web; Computer; Security; Vulnerability; SQL injection**

I. INTRODUCTION

The website has been globally used by many world organizations and communities, it can be seen from the development of world information society that has no longer using paper to disseminate information such as newspapers, brochures, textbooks, and are shifting their dissemination process using electronic or digital media to provide information for public. In addition, web sites are also being utilized as a part of the communication activities and therefore they can also be classified according to their usefulness. In relation to this, the author has previously examined the website security of several universities in Indonesia, that are precisely located in Jakarta which utilized and applied the techniques in more details with SQL Injection Technique [1]. Universities in Indonesia and other countries have been using websites as medium for information publication that can be widely accessed by the world communities. Furthermore, many universities are also making use of their websites to provide technology and utilize them as teaching and learning media as well. For example, recently it has been known globally that some websites are experiencing hacking or commonly known by the term "hack". One of the largest companies in the entertainment field namely Sony Pictures Entertainment has experienced hacking up important data owned leak. Consequently, the author aims to perform a study to obtain more detailed uncommon information related to websites security by utilizing the science of information security. Additionally, the research would focus the evaluation only at private universities in Jakarta. As a result, the author would exploit security vulnerabilities that exist on each university's website and also suggest some improvements [2].

In this paper, we have introduced vulnerability and SQL Injection. Section 2 of this paper includes a brief background material on topics concerning vulnerability and

Corresponding Author: IGN Mantra

web application security risk. In Section 3, method of vulnerability assessment is outlined, whilst in Section 4, discussion and results are presented. Finally, the last section concludes the conducted study and analysis.

II. BACKGROUND

Vulnerability on each website is different and lead to different factors that may cause attack. There are so many kinds of vulnerability on the web, for example security weaknesses misconfiguration or sensitive data exposure. In relation to this actuality, the author has previously conducted a number of researches that utilized injection method on the top 10 web application security risk as identified by OWASP. These top 10 web application security risk include: 1) the method of injection; 2) cross site scripting (XSS); 3) broken authentication and session management; 4) insecure direct object references; 5) cross-site request forgery; 6) security misconfiguration; 7) insecure cryptographic storage; 8) failure to restrict URL access; 9) insufficient transport layer protection and 10) invalidated redirects and forward [2].

Injection method is included in part of the security dangers that can be exploited. With the injection method not only a vast amount of important information can be obtained, the approach can also be used as a staging process in gaining full-access to the system by utilizing the previously taken information [3].

1. Client-Side Validation Code for Web Applications

Web application development practice currently treats client and server application components as two separate parts of a software device [8]. Each component is written independently, usually in the form of a programming language on a specific platform, a process known as susceptible sharing logic error in the client and server models. When the client and server are out of sync then the impedance mismatch will occur, which will lead to the occurrence of vulnerability in the software as shown in the parameter tampering.

This paper describes the basic approach in the development of new software called "Wave / Wave" in which the authors develop server-side applications and develop a device that automatically synthesizes client-side application logic. Wave uses program analysis techniques to extract the logical specifications of the server, where it also performs client code synthesis. Wave also synthesize interactive client interface that includes asynchronous callback (AJAX) that the performance and coverage manually written competitors while making sure no new security vulnerabilities. Wave effectiveness is shown and evaluated on 3 predefined web applications.

2. Web Application Vulnerability

Due to the limited time and resources [5], the web application engineers need support in identifying the vulnerability code. A practical approach to predicting vulnerability code will prioritize security audit efforts. In this paper, the authors propose the use of a set hybrid code attributes (static and dynamic) that characterize input validation and input patterns with sanitization code that is expected to be a significant indicator of the vulnerability of web applications. However, since the static and dynamic program of analysis will complement each other, both techniques are used to extract the proposed attributes in a scalable and accurate way.

Current vulnerability prediction techniques depend on the availability of labelled data and vulnerability information. For existing applications today, past vulnerability data is often not unavailable and incomplete. Therefore, to overcome the two situations in which past data labels are available or not, it is necessary to learn to monitor when constructing predictor vulnerability based on hybrid code. Given that semi-supervised learning has not been fully researched in this domain, the authors will explain how to use effective learning schemes for vulnerability prediction. The author also conducts empirical studies on 7 open source projects where built and evaluated supervisory models. When validated with full labelled data, the supervised model achieves an average of 77 percent recall and 5 percent false alarm probabilities to predict SQL injection cross site scripting, remote coding execution and applied vulnerability inclusion. With a low number of labelled data, when compared to both the supervised and semi-supervised models showing an average 24 percent higher recall rate and a 3 percent lower probability of false alarms, thus demonstrating semi-supervised learning to be a better solution for real world applications where Vulnerability has been lost.

3. Web Applications through XSS

Web applications have become a very popular means of software development [9]. This is because one of the many advantages of web applications is that they do not need any installation process on every computer client, data center, business cost reduction etc. However, as the trend of web applications is increasing, consequently they are also becoming more vulnerable to attacks. Cross Site Scripting (XSS) is a major threat to web applications and also the most basic attack of web applications. It provides interfaces

for other types of attacks such as cross site fraud requests, including hijacking sessions. There are 3 types of XSS attacks, which are the non-persistent XSS, persistent XSS and DOM-based vulnerability. Yet, there is one more type of the XSS attack which is more specific in comparison to the other three, which is the XSS induction. In this paper the authors aim to study and consolidate the understanding of XSS and its origin, manifestations, types of hazards and mitigation efforts for XSS. Different approaches proposed by the authors are presented and comprehensive analysis in regards to each approach is also provided.

4. Detecting Web Application Vulnerabilities

Currently, web applications have been widely used to deal with sensitive data and interfaces with back-end components, but they are often written by poorly experienced programmers with low security skills [6]. The majority of vulnerability affecting web applications may stem from the lack of proper validation of the user input, also before it is used as an argument of the output function. Some analysis of the proposed programming techniques automatically to see this vulnerability. One of the most effective is dynamic node analysis. This approach also introduces significant run time. In this paper, the authors present hybrid analytic tools that integrate together the forces of static and dynamic approaches to detect vulnerability in web applications such as: static analysis, performed only once, is used to reduce the overhead runtime of dynamic monitoring. The author also designed and implemented a tool called PHAN, which is able to analyze static PHP bytecode to search for malicious code reports, and next is to monitor the dynamic analysis phase.

III. METHODOLOGY

The methodology used in this study is a constructive research method that aims to construct a framework to test the security vulnerabilities of the college web server and then record the response of the test simultaneously by collecting identified web security vulnerabilities from the web server under examination. Accordingly, the study utilized and combined a number of test tools to conduct the web vulnerability assessment.

Each website that can be accessed by the world community has a variety of uses. Information related to security that sometimes goes unnoticed by users, developers and Web site developers. Therefore, in this study the authors conducted a study and provide knowledge associated with testing to get deeper information on several websites tested. The technique used to test the web server by using SQL injection attacks.

Further analysis, testing of the danger of mistakes in the design of publicly accessible information, then will get deeper and important information such as database systems are used, the database name, username and password or even using SQL injection techniques [2].

Further above problems are solved by performing library research, field study and analysis of the results obtained from testing directly to any website which has the potential to be applied SQL injection technique. There are steps being taken on the part of the methodology applied to get a good result from the test.

IV. DISCUSSION AND RESULTS

After collecting sufficient amount information about the website being tested, the following step is to conduct an analysis on the application of an attack or technique to get information using SQL injection. At this stage information related to a weakness in the application that relates to the SQL injection techniques can be acquired.

Accordingly, more weaknesses will be analyzed manually by utilizing the assessment tools to perform searches of weakness on the website of specifically that has any relation to the part of data storage using databases. The process of manual analysis can be performed directly on each website URL. Manual analysis is usually done on the following URL: http://sub.univ1.ac.id/informasi.php?id=37'

If an error occurs when the manual analysis is performed, the possible solution is to apply some techniques to get information about the SQL injection, but when they are being further analyzed, the analysis can be done manually as mentioned earlier in the application of the SQL injection techniques. In order to accelerate the process to ensure that the URL can be attacked using the next SQL injection, then one can utilized the tools that are being used to search, analyze, and verify that the URL has a weakness or can be applied with a technique to get deeper information with SQL injection [5].

A. *Step by step SQL Injection*

SQL injection is a code injection technique that exploits a security vulnerability that took place in the database layer of an application. The vulnerability is present when user input is either incorrectly filtered for string literal escape characters embedded in SGL statements or user input is not strongly typed and thereby unexpectedly executed. It is an instance of a more general class of vulnerabilities that can occur whenever one programming or scripting language is embedded inside another. SQL injection attacks are also known as SQL insertion attacks. The step by step process of

an SQL injection attack is described as follow:

Step -1 Find a website that is vulnerable to the attack. Once get through this, the rest step is actually a cake-walk.

Step-2 Once fa vulnerable site is found, a process to enumerate the number of columns and those columns that are accepting the queries need to be conducted. This can be done by appending an 'order by' statement to the URL,e.g. https: www.univ1.ac.id

Step -3 Enumerating the SQL version
It is required to reveal the version of used SQL as we would like to inject the command in one of the open columns.

Step -4 Exploit
This step is actually the final phase which is to get the list of databases and afterwards the result will be displayed as a list of databases available in the back-end of the website.

There are several commands to search, detect, and remove the contents of the storage data from the web being tested. Some commands that can be embedded to the URLs to perform the injection attacks are as follows:

This is Crawl Command
Sqlmap –u http://univ15.ac.id/index.php --crawl=3

In addition to the command, if it is to ensure it can be also with the following command:

This is to check for Injectable Command
sqlmap –u http://sub.univ13.ac.id/informasi.php?id=37 --random-agent

Complete command of the tools can be found in the tools directory or if you use Linux command can write in the terminal:

"sqlmap --help" **OR** ***"man sqlmap"***

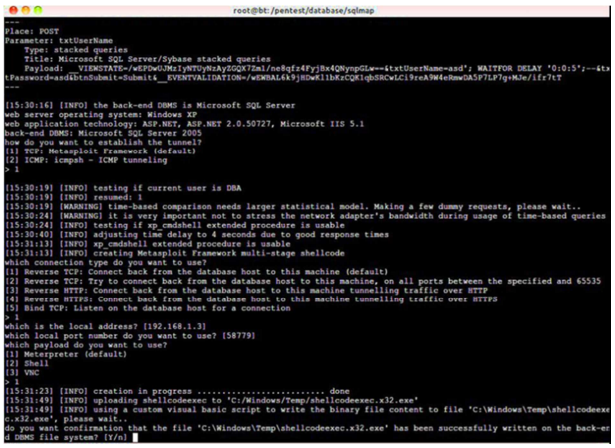

Fig. 1. Crawl Commands

At the stage of finding a weakness, further assisted analysis tool burp suite to obtain more detailed information related vulnerability being tested website use, after doing an analysis on the target and crawl on the website, then use a scanner or burp scanner selection tool. The following selection burp scanner is shown on Figure 3.

Port Based Vulnerability Scanner

To get an early idea, some traditional penetration tests usually conduct an automated port scan process. This step is used to obtain an initial picture about any information that can be displayed on a targeted website.

The port scanner will perform a series of tests to determine whether a particular port on the remote host is willing to accept the connection or not. In general, the port scanning process is using some IP-based protocols such as TCP, UDP, and ICMP.

However, the detected port is depending on the environment itself with two possible outcomes, which are:

Open - the port is able to receive data
Closed - the port is not able to receive data

The scanner tool will also display another form such as "filtered", because it cannot accurately stated whether the port is open or closed. When the scanner displays that the port is open, then a tentative assumption that the opened port is vulnerable can be made. For example, based on the fact that the port scanner is connecting to TCP port 23 and that the port is listening, then the scanner is reporting that the telnet service is being provided on the remote host and the flag being clear text authentication protocol enabled.

Fig. 2. Checking for injectable commands

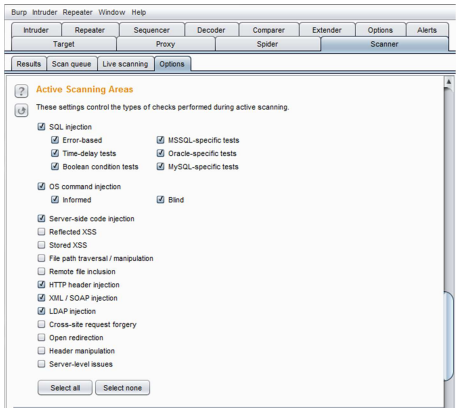

Fig. 3. Scanner Options

The scanner tool can perform passive and active scanning. As it can be seen in the picture with several options ticked, the options contained in the active scanning areas that can be selected, or focused on a few options. This study focuses not only on the weaknesses in which the SQL injection technique can be applied, but also would like to explore the possibility of gaining knowledge or information if there are other drawbacks when checklist of all the options that exist in Burp Suite tool [3].

The following are the descriptions of vulnerability found at the college when an evaluation is conducted on http://univ2.ac.id:

Furthermore, this research does not only focus on the results of the scan tools that are used, yet it also aims to test directly whether there was indeed a weakness as the existing results or it did not. It is associated with the test results there is false positive and false negative that have been submitted previously.

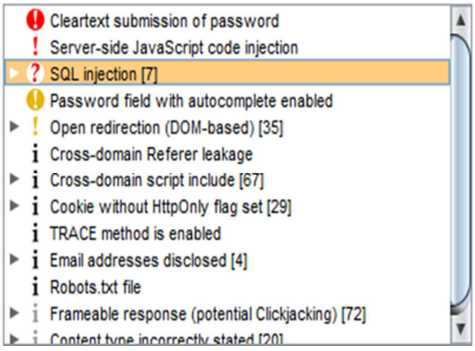

Fig. 4. Vulnerability Scanner

B. Test Reporting

Results of evaluation based on testing stages performed on several website of some best universities in Indonesia that run courses on informatics study program using the SQL injection techniques are outlined in Table 2. The attempt of attacks were managed to extract some information from data storage using the SQL injection techniques. The study also found some weaknesses that can be used as reports, these drawbacks are only additional and do not apply to the other techniques in order to get more information even though it potentially can be done.

Weakness in website and application that are related to attacks in the form of SQL injection techniques is disguised, as pointed out earlier by the alias www.univ1.ac.id up to www.univ15.ac.id. Websites that have weaknesses and can be compromised by the SQL injection techniques can be seen in the following table: --TABLE-NYA?--Whereas, in Table 4.2 it can be seen that there is a weakness on the website that can be compromised by the SQL injection techniques. However, not the entire attempt managed to extract valuable information from the data storage. In the statement on the table there are 4 information given and 4 other information are intended for further research. Yet, weaknesses that are associated with the implementation of SQL injection or other weaknesses that have the potential to obtain more detailed information from websites tested may also be found [4].

The four (4) statements as mentioned above are described as follows:

1. Not tested

This statement indicates that the website cannot be compromised by the SQL injection technique. Indications are visible from the look of the website and the target is to have knowledge about computer security website better than the other targets. In addition, there are targets that have been compromised to such websites does not perform as it should. Another reason of having the not tested statement at the university is related with the limitation of research time, the focus of this research is a website that could potentially be applied SQL injection technique.

2. Not found

This information is associated with the discovery of weaknesses that lead to the adoption SQL injection technique where it can be applied.

3. There is a sub-domain

This description is given for targets that are during testing and analysis process can extend coverage not only on the analysis of weaknesses that get the address of the Kopertis III and Google.

4. Not in the domain

The description is absent in this domain associated with the testing and analysis of the weaknesses that allow if further analysis of the sub-domains are weaknesses that can be applied to SQL injection techniques.

Aside from the SQL injection technique, previous studies have also found some weaknesses in the other universities that if being exploited by people who are not responsible can lead to a catastrophic condition. However, not each of the identified weaknesses has been validated since the focus of the study is only on those techniques to extract valuable information through the application of SQL injection. Additionally, weakness among others is presented in Table 2.

Another drawback is attached to the writing test website security vulnerabilities Colleges Jakarta as additional information to Table 2. This extra information are expected to be of a help for the readers who keen to conduct further analysis, testing and research, as material information, information, and references.

Table. 1. Test Reporting

Domain / Web Address	Description
univ1.ac.id	Not in test
univ2.ac.id	Not found
univ3.ac.id	Not in test
univ4.ac.id	Not found
univ5.ac.id	Not found
univ6.ac.id	Not in test
univ7.ac.id	Not found
univ8.ac.id	Not found
univ9.ac.id	Contained in a sub-domain - other host
univ10.ac.id	Not found
univ11.ac.id	Contained in domain
univ12.ac.id	Not found
univ13.ac.id	Contained in a sub-domain - other host
univ14.ac.id	Contained in a sub-domain - other host
univ15.ac.id	Not found in domain

Table. 2. Weakness Website

No	Weakness
1	Email address disclosed
2	Server side injection
3	Ip internal disclosure

Table. 3. Random IP for Testing with SQL Injection Technique using TOR Network.

NO	Date	Time	IP/DNS	Descriptions
1	7-Dec-16	22:13:04	103.23.22.XXX	Search and Target Mapping
2	10-Dec-16	21:18:23	139.255.45.XXX	Try to Access and Get Information (Information Gathering)
3	15-Dec-16	00:15:02	103.247.8.XXX	Manual Test for Some Target from Information Gathering Data
4	17-Dec-16	23:17:26	104.31.86.XXX	Manual Test for Some Target from Information Gathering Data
5	23-Dec-16	22:34:09	117.102.97.XXX	Manual Test for Some Target from Information Gathering Data
6	10-Jan-17	21:12:19	202.58.182.XXX	Information Gathering (2)
7	13-Jan-17	22:34:06	118.98.73.XXX	Information Gathering (2)
8	20-Jan-17	22:19:08	122.200.2.XXX	Mapping/Count Dynamic and Static Pages 1
9	24-Jan-17	22:10:45	202.152.198.XXX	Mapping/Count Dynamic and Static Pages 2
10	28-Jan-17	21:33:08	202.51.119.XXX	Mapping/Count Dynamic and Static Pages 3
11	2-Feb-17	22:23:25	202.43.180.XXX	Crawling and Testing Some Target with Sql Injection Technique 1

12	7-Feb-17	21:20:12	104.28.19.XXX	Crawl and Test Some Target with Sql Injection Technique 2
13	8-Feb-17	22:24:27	202.146.254.XXX	Crawl and Test Some Target with Sql Injection Technique 3
14	9-Feb-17	23:37:15	202.43.183.XXX	Crawling and Testing Some Target with Sql Injection Technique 4
15	10-Feb-17	21:45:07	203.128.81.XXX	Crawling and Testing Some Target with Sql Injection Technique 5
16	11-Feb-17	21:34:18	103.23.22.XXX	Crawling and Testing Some Target with Sql Injection Technique 6
17	13-Feb-17	21:35:17	139.255.45.XXX	Crawling and Testing Some Target with Sql Injection Technique 7
18	15-Feb-17	22:20:23	103.247.8.XXX	Crawling and Testing Some Target with Sql Injection Technique 8
19	20-Feb-17	23:30:12	104.31.86.XXX	Crawling and Testing Some Target with Sql Injection Technique 9
20	20-Feb-17	21:38:21	117.102.97.XXX	Crawling and Testing Some Target with Sql Injection Technique 10
21	8-Mar-17	22:39:36	202.58.182.XXX	Access and Get Data from Target with SQL Injection Technique 1
22	12-Mar-17	23:28:38	118.98.73.XXX	Access and Get Data from Target with SQL Injection Technique 2
23	15-Mar-17	23:33:45	122.200.2.XXX	Access and Get Data from Target with SQL Injection Technique 3
24	20-Mar-17	22:29:05	202.152.198.XXX	Access and Get Data from Target with SQL Injection Technique 4
25	25-Mar-17	21:55:21	202.51.119.XXX	Access and Get Data from Target with SQL Injection Technique 5
26	1-Apr-17	23:47:38	202.43.180.XXX	Retest, Manual and Crawl Target 1
27	5-Apr-17	22:32:21	104.28.19.XXX	Retest, Manual and Crawl Target 2
28	8-Apr-17	23:49:37	202.146.254.XXX	Retest, Manual and Crawl Target 3
29	13-Apr-17	22:20:21	202.43.183.XXX	Retest, Manual and Crawl Target 4
30	18-Apr-17	23:49:49	203.128.81.XXX	Retest, Manual and Crawl Target 5

Fig. 5. Agile Web Software Development with Security Testing Model Framework

V. CONCLUSIONS

The study concludes that not all websites have a weakness that may cause unusual information or confidential information being exposed of a result from SQL injection techniques. Furthermore, the study has also found that not all weaknesses that exist on a website can be compromised by the SQL injection techniques. Yet, there are weaknesses or vulnerabilities in each of the target website that is not related to attempt of information disclosure using the SQL injection technique that can still be further applied. In addition, engineering information with SQL injection remains as one of the techniques that have been known to the developers. Attackers commonly performed the SQL injection and it turns out that there are still weaknesses in some coding to be used for the deeper penetration test. The targeted website of analysis that the initial of information related to the specified target can be obtained by a technique commonly performed to search for information, either using search engines or tools available

for gathering information or intelligence gathering.

REFERENCES

[1] Halfond, W. G., Viegas, J., & Orso, A. (2006, March). A classification of SQL-injection attacks and countermeasures. In *Proceedings of the IEEE International Symposium on Secure Software Engineering* (Vol. 1, pp. 13-15). IEEE.

[2] Kieyzun, A., Guo, P. J., Jayaraman, K., & Ernst, M. D. (2009, May). Automatic creation of SQL injection and cross-site scripting attacks. In *Proceedings of the 31st International Conference on Software Engineering* (pp. 199-209). IEEE Computer Society.

[3] Khoury, N., Zavarsky, P., Lindskog, D., & Ruhl, R. (2011, October). An analysis of black-box web application security scanners against stored SQL injection. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on* (pp. 1095-1101). IEEE.

[4] Khoury, N., Zavarsky, P., Lindskog, D., & Ruhl, R. (2011, June). Testing and assessing web vulnerability scanners for persistent SQL injection attacks. In Proceedings of the First International Workshop on Security and Privacy Preserving in e-Societies (pp. 12-18). ACM.

[5] Shar, L. K., Briand, L. C., & Tan, H. B. K. (2015). Web application vulnerability prediction using hybrid program analysis and machine learning. IEEE Transactions on Dependable and Secure Computing, 12(6), 688-707.

[6] Monga, M., Paleari, R., & Passerini, E. (2009, May). A hybrid analysis framework for detecting web application vulnerabilities. In Proceedings of the 2009 ICSE Workshop on Software Engineering for Secure Systems (pp. 25-32). IEEE Computer Society.