Jurnal
Teknologi

Full Paper

Risk Analysis of Database Privelege Implementation in SQL Injection Case

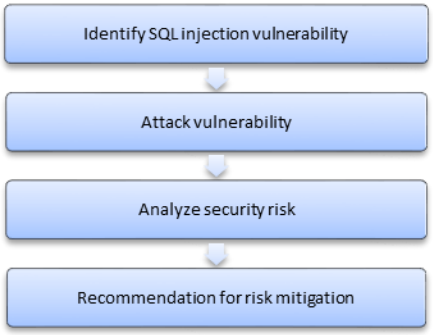
Prajna Deshanta Ibnugraha[a*], Lukito Edi Nugroho[b], Widyawan, Paulus Insap Santosa[b]

[a]Department of Electrical Engineering and Information Technology, Universitas Gadjah Mada, Yogyakarta, Indonesia
[b]School of Applied Science, Telkom University, Bandung, Indonesia

Article history
Received
01 June 2015
Received in revised form
15 October 2015
Accepted
02 January 2016

*Corresponding author
prajna.deshanta.i@mail.ugm.ac.id

Graphical abstract


Identify SQL injection vulnerability
Attack vulnerability
Analyze security risk
Recommendation for risk mitigation

Abstract

Software is important thing that needed by enterprises to support business. When developers build software, security must be concerned as important element. In bad condition, security incidents can make financial loss to organizaion so it need mitigation actions to minimize risk. Security testing and risk analysis become base process to choose good mitigation method. Implementation of database privilege become one of mitigation methods that can be used in SQL injection attack case. Based on DREAD analysis, it can decrease risk of SQL injection attack from high to medium ranking.

Keywords: Database privilege; risk analysis; DREAD

1.0 INTRODUCTION

Software is needed by organization to support business so security becomes important element that must be given attention. Failure in IT security may cause serious damage for organization like financial loss [1]. PWC survey in 2013 note that some American enterprises suffer financial loss about US$ 415.000 [2]. Organizations need to measure security software to prevent security incidents. In software life cycle, measuring software security can be included in testing phase and mitigating security vulnerability can be included in maintenance phase [3][4]. Security measuring is important for organization to identify vulnerability, threat and risk. The output from security measuring also can be used by decision-maker to take proper mitigation method [5].

Although measuring security is important, many organizations do not perform it. From study case, some university websites in Indonesia still have SQL injection vulnerability. SQL injection vulnerability is first ranked vulnerability in OWASP (*The Open Web Application Security Project*) Top 10 2013 [6]. Imperva survey declare that about 31% PHP website application and about 39% ASP website application are attacked throught SQL injection vulnerability [7]. SQL injection is attack against web application that has lack of input validation[8]. Direct insertion of malicious code into user input variables are basic way to execute SQL injection attack [9].

There are some methods that can be used to minimize risk of SQL injection attack. Common method is taken from perspective of application developer. Developers add mechanism to filter user input in application. This study tries to describe database privilege mechanism where it is taken from perspective of system administrator. Database privilege mechnism can be combined filtering mechnism to increase website security.

In this study, security testing is done to identify and explore impact of SQL injection vulnerability. Risk analysis is used to describe impact of vulnerability to organization and analyze effect of database privilege implementation as mitigation method.

2.0 RELATED WORKS

Some studies have discussed about testing and analyzing software security. Sushila et al [10] uses ADMIRE model to mitigate risk of SQL injection attack. Shusila et al [10] use DREAD to rank the threat. Ram et al [11] evaluate security risk in Geospatial Weather Information System (GWIS) with DREAD. Output of evaluation is recommendation documents. Sonia et al [12] uses fuzzy logic approach to build priority of risk based on DREAD model. Sushila et al [10], Ram et al [11] and Sonia et al [12] use similiar method to analyze risk.

Qian et al [13] propose security testing guidance for enterprise to improve testing efficiency. Qian et al [13] use SQL attack and XSS attack as case study. Database privilege become one of the suggested methods to limit access of attacker. In Qian et al [13] paper does not analyze risk so the effect of suggested methods are still not known clearly.

This study tries to add risk analysis process from Qian et al [13] study and add database privilege implementation as mitigation method of SQL injection vulnerability. DREAD is used as a method to analyze risk. Output from this research is recommendation about mitigation method for university website. This study uses two university websites in Indonesia as study case.

3.0 METHOD

Measuring software security can be done with penetration testing or static code analysis method [14]. Penetration testing is black-box method that used to identify vulnerabilities from attacker perspective. Static code analysis uses white-box approach to test the software security. Source code of software is traced to find security vulnerabilities. Static code analysis is often done in developer perspective. This study uses penetration testing to measure web application security. Steps that used to penetration testing consist of identifying of testing range, building of testing scenarios and executing of testing scenarios [15].

This study uses DREAD model to analyze security risk. It analyze output of penetration testing which DREAD consists of this metrics [16] :

- Damage potential
  How great is damage that caused by attack through vulnerability.
- Reproducibility
  How easy is attacker to reproduce attack
- Exploitability
  How easy is attacker to launch attack
- Affected users
  How many users are affected
- Discoverability
  How easy is attacker find vulnerability.

4.0 EXPERIMENT

Experiment is done in running application of university website and focus in SQL injection vulnerability. Testing scenarios can be shown in Figure 1.

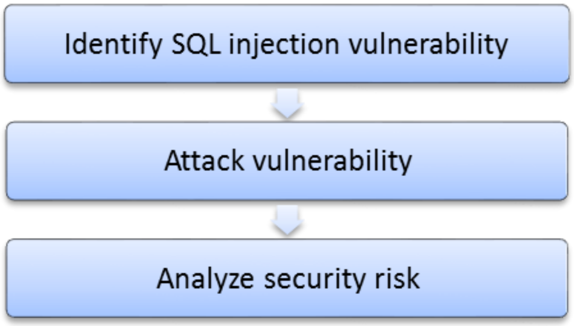


Figure 1 Scenarios of experiment

This study identifies SQL injection vulnerability in two university websites that have different characteristic in database implementation. The first website implements database privilege and the other website does not implement it. For attacking of SQL injection vulnerability, we use penetration testing method and the result will be evaluated with DREAD model.
Steps of experiment can be shown below :

1) Search target
   This study uses Google Search Engine to find some university websites that have SQL injection vulnerability. ***inurl*** parameter is used to filter the output of Google Search Engine.
2) Identify SQL injection vulnerability
   This study uses try and error method to identify vulnerability. one of the methods is by adding single quotation mark (‘) in the end of URL. It make website resulting error message. In this case, error message is shown “You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''44023'' ORDER BY tgl DESC' at line 1”
3) Execute attack
   This study uses havij tool to execute and explore databases. Havij is usually used to find and exploit SQL Injection vulnerabilities on a web page. It is develop by ItsecTeam [17].

5.0 RESULT AND ANALYSIS

Many information of databases can be explored like database version, database type, database name, tables, etc. This study take some information that has relation with database privilege implementation. Result is shown below :

Website A :

- Database type : MySQL version 5
- Database privilege implementation : Yes
- Name of database : [redacted]_web*
- Important tables : xadmin, anggota

Website B :

- Database type : MySQL version 5
- Database privilege implementation : No

Table 1 List of databases and important tables from penetration testing in website B

Databases [d_k]	Important Tables [T_k]
db_absenjur	tbl_absensi
db[redacted]*	tblbiodata, login, tbluser, tblnilai
dbreg_online	tbl_pin, tbl_usr, tbl_infologin, tblbiodata
Dbwebsite	tblduk
lib_analis	tbluser, vwuser
lib_direktorat	-
lib_keperawat	tbluser, vwuser
lib_p[redacted]*	tbluser, vwuser
lib_[redacted]*	tbluser, vwuser
m[redacted]*	member
Mysql	user
phpmyadmin	-
reg_ol_2013_1	-
reg_ol_2013_2	-
reg_ol_2014_1	-
reg_ol_2014_2	-
reg_ol_2014_3	-
reg_ol_2015_1	-
reg_ol_2015_2	-
reg_ol_2015_3	tbl_listujian, tblbiodata
riyan_perpus	tbluser, vwuser
riyan_perpus_gigi	tbluser, vwuser
riyan_perpus_[redacted]*	tbluser, vwuser
riyan_perpus_keperawatan	tbluser, vwuser

*) some table or database names must be censored

Categories of important table in this study are based on Personally Identifiable Information (PII) criteria. The following list contains information criteria that considered PII [18]:

- account information (username and password)
- address information : email, home address
- financial information, bank account
- educational information : student grades
- telephone number
- individual information : date of birth, place of birth, etc

Number of databases [ND] and number of important tables [IT] can be represented in equation 1, equation 2 and Figure 1.

$$ND = \sum_{k=1}^{n} d_k \quad (1)$$

$$IT = \sum_{k=1}^{d} T_k \quad (2)$$

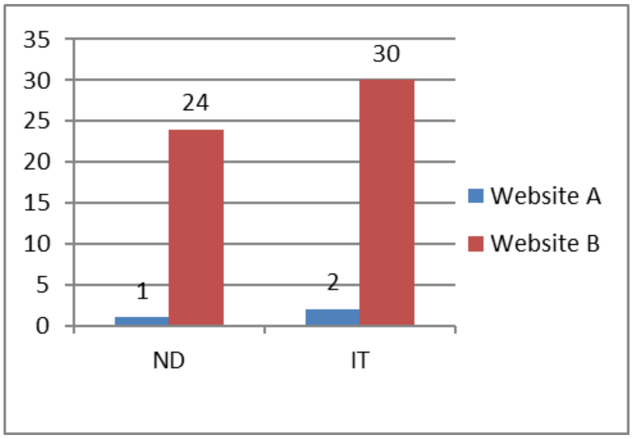


Figure 2 Output penetration testing in number

Result of experiment shows that website B has a greater damage than website A because more information can be explored by attacker. The condition affect Damage Potential and Affected Users metric in DREAD analysis. Website characteristics are shown in Table 2 [16]:

Table 2 The characteristics of attack website based on DREAD metrics

DREAD Metrics	Website A	Website B
Damage Potential	Attacker can view sensitive information in one database that used by its website	Attacker can view sensitive information and change content from databases in the server (24 databases)
Reproducibility	The attack can be reproduced every time	The attack can be reproduced every time
Exploitability	Skilled attacker can attack website and repeat the attack	Skilled attacker can attack website and repeat the attack
Affected users	Some user	All user consist of administrator, lecturer, staff, students.
Discoverability	Attacker must take some thinking to find vulnerability	Attacker must take some thinking to find vulnerability

Based on website characteristic, risk calculation can be shown in Table 3 :

Table 3 Score of attack website in DREAD metrics

DREAD Metrics	Website A	Website B
Damage potential	2	3
Reproducibility	3	3
Exploitability	2	2
Affected users	2	3
Discoverability	2	2
Total	11	13
Risk Rating	Medium	High

Risk rating definition refers to Table 4 [16]:

Table 4 DREAD ranking definition

Total of DREAD metrics	Ranking Definition
12–15	High
8–11	Medium
5 –7	Low

In this study, database privilege is implemented in database users. It making database users only have limited access in appropriate databases. It can minimize risk of SQL injection attack. It is proven by result of DREAD analysis that implementation of database privilege can decrease ranking of risk from high to medium.

6.0 CONCLUSION

This study focus in security testing and mitigation technique to minimize risk of security incidents that it can be included as part of testing phase and maintenance phase in software life cycle. Penetration testing is choosen as method to explore database in university website that has SQL injection vulnerability. DREAD model is used to analyze output of penetration testing process. Based on analysis, implemetation of database privilege can be used as one of mitigation technique to minimize risk of SQL injection attack. It is proven by DREAD ranking. Implementation of database privilege can decrease risk ranking from high to medium. It is affected by reduction of damage potential value and number of affected user.

References

[1] Albakri, S.H., Shanmugam, B., Samy, G.N., Idris, N.B., Ahmed, A. 2015. Traditional Security Risk Assessment Methods in Cloud Computing Environment: Usability Analysis. *Jurnal Teknologi (Sciences & Engineering)* 73(2): 85–89.

[2] Mickelberg, K., Pollard, N., Schive, L. 2014. US cybercrime: Rising risks, reduced readiness Key findings from the 2014 US State of Cybercrime Survey. PWC.

[3] Howard, M., LeBlanc, D. 2003. *Writing Secure Code*. Microsoft Press.

[4] Ibnugraha, P.D., Ferdiana, R., Suharyanto, Santosa, P.I. 2015. Evaluation of Security in Software Architecture Using Combination of ATAM and STRIDE. *Journal of Theoretical and Applied Information Technology*. *75. 112-119*.

[5] Dewi, L.P., Gunawan, I., Winoto, C. 2014. Risk Assessment in Securing Radio Frequency Identification (RFID) Systems: A Case Study on Petra Christian University Library. *Jurnal Teknologi (Sciences & Engineering)*. 68(3): 89–95.

[6] OWASP. 2013. Top 10 2013 – Top 10. [Online]. From: https://www.owasp.org/index.php/Top_10_2013-Top_10. [Accesed on 01 October 2015].

[7] Imperva. 2014. Web Application Attack Report #5. Imperva.

[8] Djuric, Z. 2013. A Black-box Testing Tool for Detecting SQL Injection Vulnerabilities. *Second Interntional Conference on Informatics and Applications (ICIA)*. 216-221.

[9] Dukes, L., Yuan, X., Akowuah, F. 2013. A Case Study on Web Application Security Testing with Tools and Manual Testing. *Proceedings of IEEE Southeastcon*, April 2013.

[10] Sushila, M., Supriya, M. 2010. Bulwark Against SQL Injection Attack– An Unified Approach. *International Journal of Computer Science and Network Security (IJCSNS)*. 10(5): 305-313.

[11] Rao, K. R. M., Pant, D. 2010. A threat risk modeling framework for Geospatial Weather Information System (GWIS): a DREAD based study. *International Journal of Advanced Computer Science and Applications (IJACSA)*. 1(3): 20-28.

[12] Sonia, Singhal, A., Banati, H. 2011. Fuzzy Logic Approach for Threat Prioritization in Agile Security Framework using DREAD Model. *International Journal of Computer Science Issues*. 8(4): 182-190.

[13] Qian, L., Wan, J., Chen, L., Chen, X. 2013. Complete Web Security Testing Methods and Recommendations. International Conference on Computer Sciences and Applications. IEEE Computer Society. 86-89

[14] Antunes, N., Vieira, M. 2009. Comparing the 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*.

[15] Shahriar, H., Zulkernine, M. 2009. Automatic Testing of Program Security Vulnerabilities. *33rd Annual IEEE International Computer Software and Applications Conference*.

[16] Meier, J.D., Mackman, A., Dunner, M., Vasireddy, S., Escamilla, R., Murukan. A. 2003. *Improving Web Application Security Threats and Countermeasures*. Microsoft Corporation. 5(6): 63-65.

[17] ITsecTeam. 2012. Havij Advanced SQL Injection. [Online]. From: http://itsecteam.com/products/havij-advanced-sql-injection/. [Accesed on 01 October 2015].

[18] McCalliste, E., Grance, T., Scarfone, K. 2010. Guide to Protecting the Confidentiality of Personally Identifiable Information (PII). *NIST Special Publication* 800-122.