Research Journal of Applied Sciences, Engineering and Technology 8(5): 658-663, 2014
DOI:10.19026/rjaset.8.1019
ISSN: 2040-7459; e-ISSN: 2040-7467
© 2014 Maxwell Scientific Publication Corp.
Submitted: May 09, 2014 Accepted: June 16, 2014 Published: August 05, 2014

**Research Article**

# Improving Web Application Security Using Penetration Testing

[1]D. SriNithi, [1]G. Elavarasi, [1]T.F. Michael Raj and [2]P. Sivaprakasam
[1]Department of CSE, SRC, SASTRA University, Thanjavur, Tamil Nadu, India
[2]Department of CS, Sri Vasavi College, Erode, India

**Abstract:** The main issues of current web application is easily hacking the user information by unauthorized person. The development of entire web application depends on scripting languages that easily displays the user authentication code to the web browser. All code must be transferred through query string parameter (URL) of the web application. This kind of application security fails when verifying it by penetration testing which is based on XSS languages. This study overcomes the security issues by developing a web application based on cross site scripting technique which the user codes are encrypted using RSA algorithm and cookies, cross domain verification based on encrypted use code. XSS vulnerabilities come in different forms and may be categorized into two varieties: reflected and stored. Reflected is on type of attack which can be performed against applications that employ a dynamic page error message to users. Stored XSS vulnerability appears when data submitted by one user is store in the application or in the back-end database. The user cookies of the web browser store only the encrypted key values. These techniques applied in Enterprise web application it support multiple organization for processing product purchase order, sales order and invoice details.

**Keywords:** Penetration testing, security issues, threats in web applications, web application testing

## INTRODUCTION

The growth of the new technologies like SOA, cloud computing and distributed computing influences the Enterprise applications in such a way to incorporate the new inventions so as to improve the productivity of the systems. These systems facing the many problems related to security, while their system is sharing the data with other organizations. The increase in number of nodes obviously increases the number of requests and responses and may contain vulnerabilities (McGraw, 2006). Organizations which are implementing the real time systems has to share the data among themselves to increase the growth of the productivity.

In order to provide the high degree of security for the enterprise applications, systems are required to protect the data over the communications. The data of an organization which are in a sharable mode might face risks and has to adopt new structured approach or an integrated model to protect the data*.

In this study we employed a technique called penetration testing also known as "Pen Testing" to help the enterprise applications to adopt new technologies in their business applications.

Before the adoption of the new technology the system has to undergo for a series of process which comprises set of actions to identify and exploit security vulnerabilities. Aileen *et al*. (2011) discussed various penetration testing tools. Various phases involved in improving the application security can be classified as follows (Avramescu *et al*., 2013):

- Hiring an ethical hacking company
- Train the employees to apply the penetration testing
- Identify the vulnerabilities
- Develop a model to protect the system

The main objective of this study is to identify the vulnerabilities over the business applications by applying the penetration testing and also providing possible solutions for the security issues. The following block diagram, Fig. 1 depicts the environment used for the testing.

## METHODOLOGY

**Survey on security issues and pen testing:** There are several intrusion detection and protection models were discussed and implemented (Su and Wassermann, 2006; Pietraszek and Berghe, 2005; Halfond and Orso, 2005). But the reason to choose the penetration testing is that it is an integrated approach provides the complete trusted computing to test and evaluate the security of business applications. It's been widely accepted and accredited by OWASP[1] and

**Corresponding Author:** D. SriNithi, Department of CSE, SRC, SASTRA University, Thanjavur, Tamil Nadu, India
This work is licensed under a Creative Commons Attribution 4.0 International License (URL: http://creativecommons.org/licenses/by/4.0/).

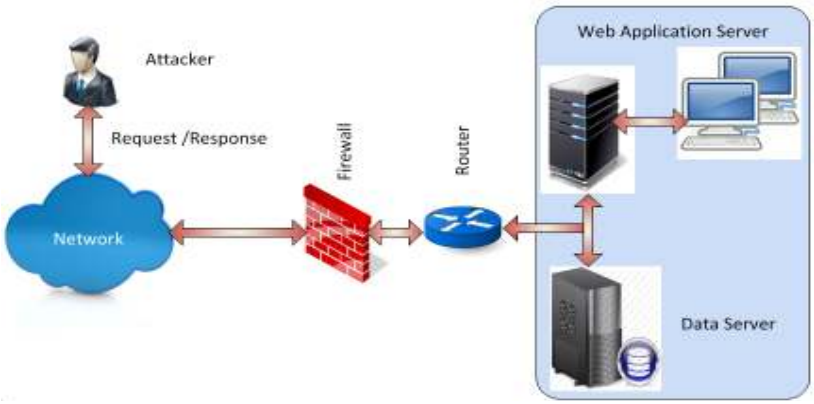


Fig. 1: Penetration testing environment

OSSTMM[2]. Penetration testing can be done in many ways some of them are as black box testing where user may not aware of what is happening inside the applications or in a module and as white box, where user may aware of the inside operations in an applications. In our example user knows the architecture of the applications and its operations.

The different phases of the penetration testing has been determined by the systematic approach and includes the following (Saindane, 2011):

- **Planning:** Defines the strategy
- **Discovery:** Testing strategies
- **Attack:** Exploits the vulnerabilities
- **Reporting:** List out the stages

Data exchange is the important and frequent operations between the organizations. It is estimated that normally web application layer is attacked by 75% of the vulnerabilities. The traditional approach is that the web crawler can be used to identify the information in the URL and this provides the incomplete data for the testing (Halfond *et al*., 2009). Static analysis technique determines the stale results for the selected application domain (Halfond and Orso, 2007). Dynamic analysis can improve the reporting phase of the penetration testing (Halfond *et al*., 2008). The above two techniques can be applied on the server side that is if the source code of the web application is available (Cova *et al*., 2007).

SQL injection or Cross-Site Scripting (XSS) are the common vulnerabilities found on the web application and are considered as top level application attacks. There are several intrusion detection and protection models were discussed and developed. Model based technology requires more effort to design, prepare test cases and identifying the user interfaces (Lebeau *et al*., 2013).

**SQL injection:** Normally web applications are suffered by the SQL injection. It is technique can be used to identify the vulnerabilities. Example: To retrieve the product information from the PRODUCT table the following query has been used:

```
Select * from PRODUCT
```

If the query is passed with explicit argument then the query look like the following:

```
Select * from PRODUCT
where, PRODUCTID = '101'
```

The un-authorized user or attacker tries to retrieve the other PRODUCTID details from the PRODUCT table by substituting the values over the SQL queries. The common attacks are identified to retrieve the following information are:

- Name of the DB and the TABLE
- COLOUMN names on the TABLE

Some techniques used to prevent SQL injections are:

- Usage of parameter objects
- Stored procedures
- Implementation of cryptographic functions
- Usage of regular expressions to define the user inputs
- Avoiding detailed error messages

**Cross-Site Scripting (XSS):** Cross-site scripting is a technique is a kind of attacks can be done by injections of malicious data or script on the trusted web applications. Different types of XSS are classified as follows[3]:

- Persistent
- Non persistent
- DOM based

## ARCHITECTURE AND IMPLEMENTATION

The main issues of current web application is easily hacking the user information by unauthorized person. The development of entire web application depends on scripting languages that easily displays the user authentication code to the web browser. All code must be transferred through query string parameter (URL) of web application. This kind of application security fails when verifying it by penetration testing which is based on XSS languages. To overcome this security attack to develop a web application based on Cross Site Scripting (XSS) technique which the user codes are encrypted using RSA algorithm and cookies, cross domain verification based on encrypted user code.

Figure 2 shows the architecture of the secured system. It includes three phases, Phase-I deals with developing an enterprise application, Phase-II used for identifying the threats by using the penetration testing and Phase-III implements RSA algorithm and protects the system from the vulnerabilities.

The existing system conduct a penetration test in a real-case scenario of multiple attacks against the network, the web application and the SQL database. Web application penetration testing refers to a set of tests used to detect various security vulnerabilities with web applications, including:

- Vulnerabilities such as SQL injection, XSS, buffer overflow, configuration of web server, etc.
- **Business logic errors:** Modification of pricelist, unauthorized logins or funds transfer, etc.
- XSS is a type of security vulnerability typically found in web applications; it allows attackers to insert client-side script into web pages viewed by other users.

The main objective is to prove the importance of secure coding through penetration testing a exploiting some of discovered vulnerabilities:

- It overcomes the hacker attack in web application while using Cross Site Scripting (XSS).
- URL of web application can be encrypted using RSA algorithm.
- Proposed web application can provide more security its passing penetration testing.
- The encrypted code only stored in cookies, token, cross domain verification.

**Design and implementation constraints:**

**Input design:** Input design is the process of converting user originated inputs to a computer based format. Input design is one of the most expensive phases of the operation of computerized system and it is major problem of a system.

**Output design:** Output design generally produces the data which satisfies the user requirements or the objective of the systems. It can also be used to evaluate the usefulness of the application. The output is designed in such a way that is attractive, convenient and informative. As the outputs are the most important sources of information to the users, better design should improve the system's relationship with us and also will help decision making, from the design elaborates the way output is presented.

**Functional requirements:**

**Application development and maintenance:** The system is develop and maintains multiple enterprise system, product details, sales order and purchase order with account ledger.

**RSA algorithm implementation:** RSA is a cryptosystem, which is known as one of the first practicable public key cryptosystems and is widely used for secure data transmission.

**Hackers attack:** Hackers list is the instruction detection method, which helps the user to find out the other users entering into the network. It contains tracker information, date of attack and time of hacking. So the user can identify who is the other user intruding into the network. So that user can identify the hackers easily through their information.

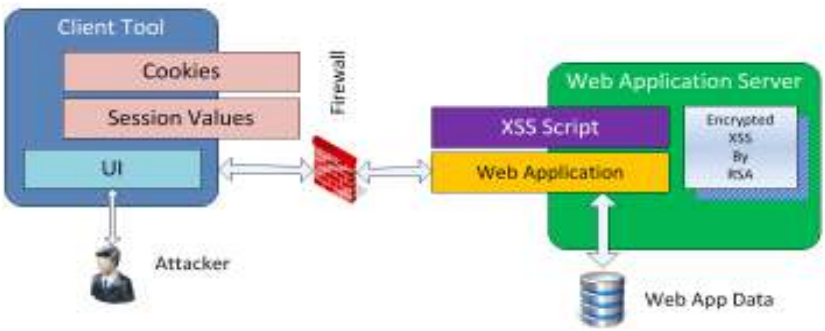


Fig. 2: Architecture of the secured system

**Evaluation of application maintenance:** To evaluate number of vulnerabilities occurs on this web application and how it can be overcome the hacker attack.

**Non-functional requirements:**

**Performance requirements:** This system is easily manageable through this project interface. The system and interface have information about product, sales, purchase order and account ledger for every enterprise system stored in database.

**Safety and security requirements:** Organization details, sales and purchase order are stored in the database in safety manner:

- Software quality attributes
- Maintainability
- Security
- Usability

**Implemented algorithm:**

**Input:** Enterprise application, PenTest cases

**Output:** Hackers_List, Hackable URLs

- Develop a sample application to implement the Pen Test
- Intentionally hack the system to identify the URL
- Prepare the hackers report
- Use RSA to protect the system
- Verify the security level of the system

## RESULTS AND ANALYSIS

To improve the application security we have developed an enterprise application and applied the penetration testing. The Enterprise application maintains the information for multiple organizations, these includes the details of product and sales and purchase orders and are stored in the application dataset. The valuable information of the organizations can be accessed by the various authorized clients of the organizations. Figure 3 shows the various processes of the applications.

The Table 1 shows the session level attacks done by the individual company hackers, which can be measured in-terms of numbers.

To strengthen the application in-terms of providing data security we have applied penetration testing over the various page URLs of the applications and identified company id and product id be the vulnerability content in the URL. This information provides attackers versus access ratio for the session and is depicted in the Fig. 4.

This ratio gives the protected level of the system access and the data access. It shows that how much the system contents are worthy to the attackers. This feature also used to disable the client accounts to prevent the system from the unauthorized access and make it as a highly secured system.

Here we have used RSA algorithm to protect the system from the vulnerability attacks, because this algorithm is easy to implement and provides the

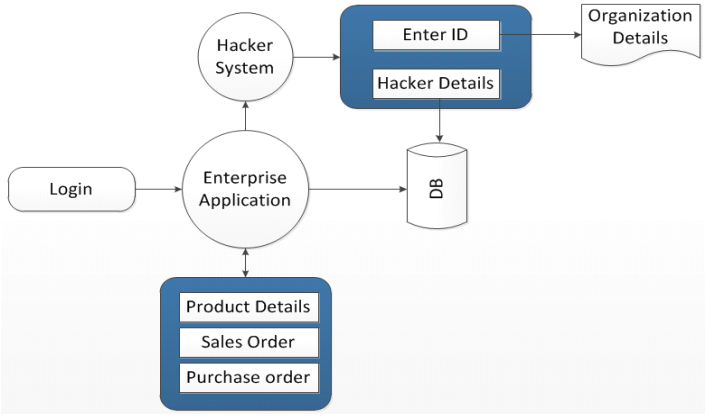


Fig. 3: Processes of the system

Table 1: Session level attacks details

| Company code | Name | Session starts at | No. of attacks |
|---|---|---|---|
| 24 | AS enterprise | 4/13/2014 6:43:27 PM | 1 |
| 23 | SA solutions | 4/13/2014 6:43:57 PM | 1 |
| 18 | Ben tech | 4/13/2014 6:45:02 PM | 1 |
| 26 | STS solutions | 4/13/2014 7:51:14 PM | 2 |
| 21 | DD enterprise | 4/13/2014 7:51:41 PM | 1 |
| 17 | Info enterprise | 4/13/2014 7:52:06 PM | 5 |

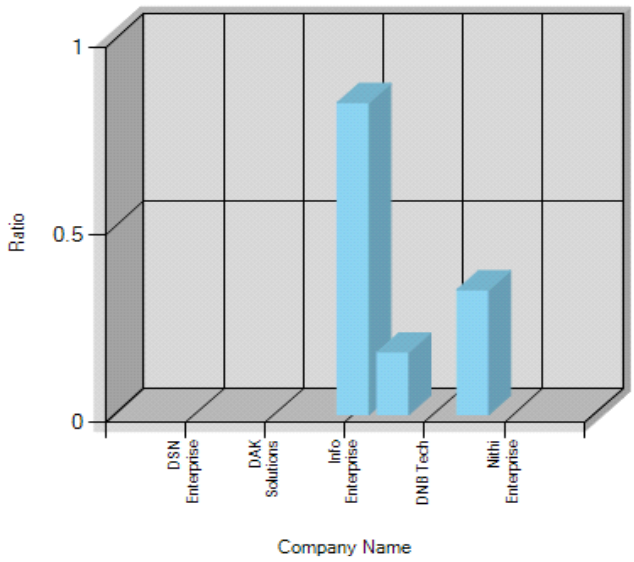


Fig. 4: Attackers vs. access

efficient results when compared with other cryptosystems (Pateriya *et al*., 2009; Fu and Zhi-Liang, 2008).

## CONCLUSION

The developed web application is tested with penetration testing and identified the vulnerability contents from the application. To make the application more secure and reliable a cryptographic method, RSA algorithm is used. This study proves that the importance of developer to understand the coding through penetration testing and discover a vulnerabilities. To implement the proposed penetration test cases without using XSS scripting language and include the features of session tracking mechanism for displaying information without using query string parameter instead of that information can be passed over http cookies. We have tested the system with various test cases and the result shows that there is an improvement in the results over the previous approaches. It ensures that the system will enable the security of web application to provide an integrated and reliable product to the user of the application.

## REFERENCES

Aileen, G.B., Y. Xiaohong, B.C. Bei-Tseng and J. Monique, 2011. An overview of penetration testing. Int. J. Netw. Secur. Appl., 3(6).

Avramescu, G., M. Bucicoiu, D. Rosner and N. Tăpuş, 2013. Guidelines for discovering and improving application security. Proceeding of the 19th International Conference on Control Systems and Computer Science (CSCS, 2013), pp: 560-565.

Cova, M., V. Felmetsger and G. Vigna, 2007. Vulnerability Analysis of Web Applications. In: Baresi, L. and E. Dinitto (Eds.), Testing and Analysis of Web Services. Springer, Heidelberg.

Fu, C. and Z. Zhi-Liang, 2008. An efficient implementation of RSA digital signature algorithm. Proceeding of the 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'08), pp: 1-4.

Halfond, W.G.J. and A. Orso, 2005. AMNESIA: Analysis and monitoring for neutralizing SQL-injection attacks. Proceeding of the 20th IEEE/ACM International Conference on Automated Software Engineering, pp: 174-183.

Halfond, W.G.J. and A. Orso, 2007. Improving test case generation for web applications using automated interface discovery. Proceedings of the the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, pp: 145-154.

Halfond, W., A. Orso and P. Manolios, 2008. WASP: Protecting web applications using positive tainting and syntax-aware evaluation. IEEE T. Software Eng., 34(1): 65-81.

Halfond, W.G.J., S.R. Choudhary and A. Orso, 2009. Penetration testing with improved input vector identification. Proceeding of the International Conference on Software Testing Verification and Validation (ICST'09), pp: 346-355.

Lebeau, F., L. Bruno, P. Fabien and V. Alexandre, 2013. Model-based vulnerability testing for web applications. Proceeding of the IEEE 6th International Conference on Software Testing, Verification and Validation Workshops (ICSTW, 2013), pp: 445-452.

McGraw, G., 2006. Software Security: Building Security in. Addison Wesley, Upper Saddle River, NJ.

Pateriya, R.K., J.L. Rana, S.C. Shrivastava and P. Jaideep, 2009. A proposed algorithm to improve security & efficiency of SSL-TLS servers using batch RSA decryption. Int. J. Comput. Sci. Inform. Secur., 3(1).

Pietraszek, T. and C.V. Berghe, 2005. Defending against injection attacks through context-sensitive string evaluation. Proceeding of the 8th International Symposium on Recent Advances in Intrusion Detection, pp: 124-145.

Saindane, M., 2011. Penetration Testing: A Systematic Approach. Retrieved from: http://www.infosecwriters.com/text_resources/pdf/PenTest_MSaindane.pdf (Accessed on: November 23, 2011).

Su, Z. and G. Wassermann, 2006. The essence of command injection attacks in web applications. ACM SIGPLAN Notices, 41(1): 372-382.