ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

New Web Doc Security Approach

Shubham Wankhede,Shubham Shekhar, Abhishek Purohit, Mukesh Kumar,

Department of Computer Engineering, SKNSITS, Lonavala, Pune, MH, India

ABSTRACT: In this project, we are going to provide security for the personal documents. By providing the secured website for the storage of the personal data. This approach not only provides the website security but also preserves the users privacy by maintaining the UID(Users identification).This UID can be used at the time of forgotten password, missing cell phone and provides every assistance required for getting the documents. In case of hacking of the account on the website, UID and the security questions plays the vital role. Hence, the authorized user can access their data securely.

I. INTRODUCTION

Web security testing tells us whether Web based applications requirements are met when they are subjected to malicious input data. Chess and West adopted the static analysis for identifying vulnerabilities was initially proposed as a way to support manual inspection . Initially called type-state analysis, taint analysis has been largely adopted to detect inadequate or missing input validation, resulting in cross-site scripting , SQL-injection and buffer overflow vulnerabilities Cross-Site Scripting (XSS) is a polymorphic category of attacks that may infect web applications as well as their clients, in many different direct and indirect ways. Many countermeasures can be deployed to face this threat: these security mechanisms are located in the internal parts of web applications (e.g. validation checks), on external security components (reverse proxies, web application firewalls (WAF) like Mod Security) or even on client-side web browsers. The technical contribution of this paper is a method to systematically test the impact of a large set of XSS vectors on web browsers, including mobile browsers (e.g. on Android).Our test driver, called XSS Test Driver executes a code within the web browser equivalent to the one ran by victims under XSS attacks.

Problem Statement:
Mostly, people doesn't carry their documents everywhere and even if carrying but it isn't secured. The account which is created on the website can get hacked. So to avoid these circumstances, we use UID at the time of registration and the security questions. This helps to overcome the problems such as missing of mobile phones. Hence, these provides good security for storage of personal documents.

Security Aspects : Security Testing is related to verify the application security services and to identify potential safety defects. complete WEB safety testing should cover deployment, infrastructure, input validation, authentication, authorization, configuration management, sensitive data, encryption, session management, operating parameters, exception management, auditing and logging, and several other aspects. Another one of the most prominent class of vulnerabilities for web application is Cross-Site Scripting, also called XSS. A XSS vulnerability consists in missing or inadequate validation of input data, so an attacker could provide data containing scripts or html fragments that get injected into web page displayed by the victim's browser

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

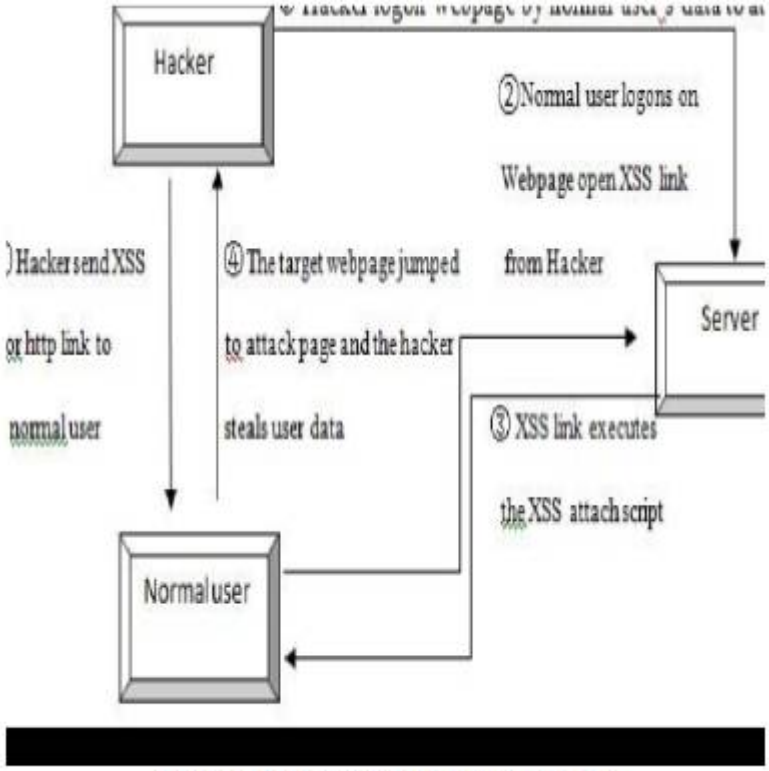


Figure 1. XSS attack mechanism

SQL injection attack testing:

Here we use a simple example to introduce SQL injection attack mechanism. Suppose the system has a users table containing two fields, which are username and password. Most Java code using SQL joint method to verify the user,just like "select id from users where username = '"+ username +"'and password = '" + password +" '", here username and password can be obtained from the users table. Next, we will introduce a simple SQL injection to let user logon system without inputting correct username and password. If we input username as 'or 1= 1—and password with random input, then the SQL becomes "select id from users where username = ''or 1=1-- and password = '123'" (here input password as 123). In this SQL statement, as 1=1 is always true and then password would not be validate and then hacker could logon system even inputting wrong username or password

II. METHODOLOGY

A. Method and suggestion to SQL injection:

Enterprises developers should use parameterized queries to build SQL queries, which could be distinguished from data and code. For most SQL queries, developers need to specify certain criteria; we need to use parameterized queries, in fact, passed parameters at run-time. Parameterized query in the SQL statement has one or more embedded parameter query. In this way, parameters will be embedded into the SQL statement and no easy to make mistakes, compared with the dynamically constructed SQL string. In database level, the database activities monitoring can also filter backstage attack. Database monitoring activities is a powerful tool again SQL injection. For now known injection attack, enterprises should be better to deploy the filter to warn database administrator when some not secure.

B)Database privilege limitation:

Companies need to better manage WEB applications associated with the account and back-end database interaction. However, this super-user accounts vulnerable to attacks, and will greatly increase SQL injection attacks and other Web

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

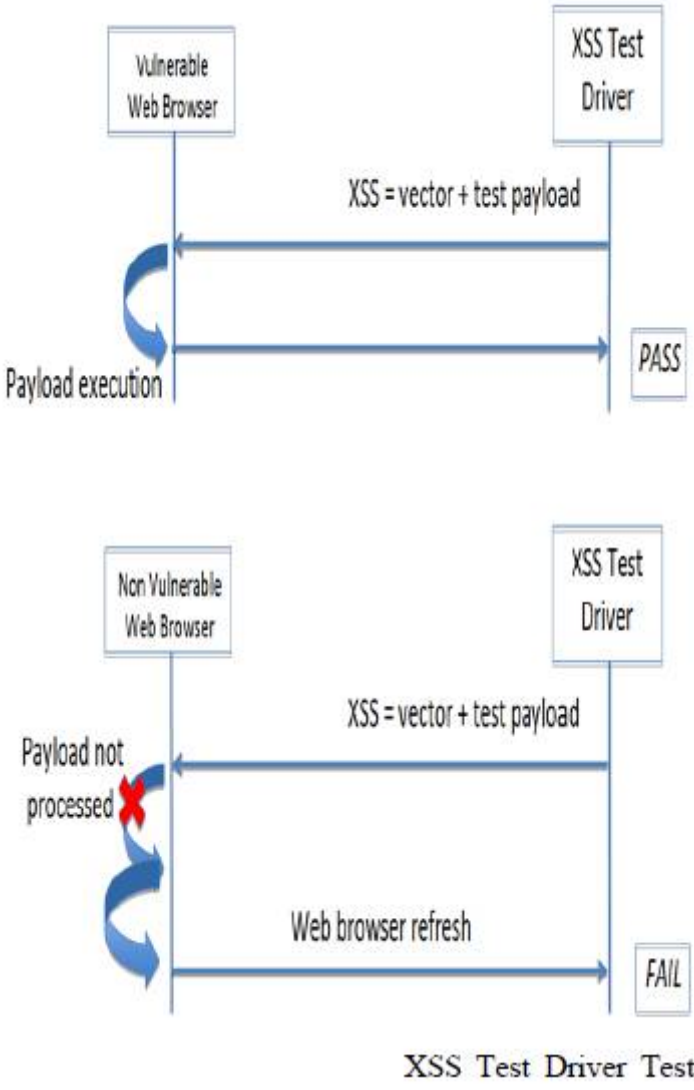
(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

attacks risks posed to the database. So make sure enterprise should be able to manage all accounts properly, offering them the lowest privilege to access background database, making sure these accounts will not have the right to change the database..

4)Method and suggestion to XSS attack:

XSS attack hacker can steal all kinds of user accounts, such as machine logon accounts, user's online back; the hacker can also steal enterprise data including reading, tampering, adding and deleting sensitive enterprise data. Since hacker steals user's bank account and password, then hacker can transfer user's money online illegally. Once user's machine is controlled by hacker through XSS attack, then hacker can control user's machine to attack other sites.



XSS Test Driver Testing

Logic

Firstly, verify reliable input for all users' submit content, including URL, query keywords, HTTP headers, POST data, etc., only accept a specified length range, appropriate format and the expected characters to submit, others are all filtered. Secondly, achieve Session tag/tokens, CAPTCHA system or HTTP reference header checks, to prevent

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

functions being performed by third-party websites. Lastly, acknowledge receipt of the content is properly normalized, only contains smallest and secure Tag no java script, no references to the remote content, especially style sheets and java script, using HTTP only cookie.

Algorithms:

1.Aho-corasick Algorithm: Aho corasicks static approach for preventing sql injection: Static Pattern Matching Algorithm

1: Procedure SPMA(Query, SPL[ ]) INPUT: Query Use Generated Query SPL[ ]Stat I Pattern List with m Anomaly Pattern
2: For j = 1 to m do
3: If (AC (Query, String. Length (Query), SPL[j][0]) = =)then
4: *StringLeng th SPL j Anomaly Matching value Query SPL j score*
5: If ( ) *Score Value Anomaly* ≥ *Threshold*
6: then
7: Return Alarm Administrator
8: Else
9: Return Query Accepted
10: End If
11: Else
12: Return Query Rejected
13: End If
14: End For
End Procedure

Aho – Corasick Multiple Keyword Matching Algorithm

1: Procedure AC(y,n,q0)
INPUT: y array of m bytes representing the text input (SQL Query Statement) n integer representing the text length (SQL Query Length) q0initi a state (first character in pattern)
2: State q0
3: For i = 1 to n do
4: While g (State, y[i] = = fail) do
5: *State* ←*f* (*State*)
6: End While
7: *State* ←*g*(*State*,.*y*[*i*])
8: If o(State) then
9: Output i
10: Else
11: Output
12: End If
13: End for
14: End Procedure

Binary algorithm:

Data: Anonymize records DATA from providers P, an EG monotonic C, a fitness scoring function score F , and the n.
Result: if DATA is private secure C then True, else false
1. Sites = sort sites(P, increasing order, score)
2. Apply slicing
3. While verify data-privacy (DATA, n, C) = 0 do
4. Super = next instance size (n− 1)&& (size_of_tupples (Σ)) // identification of column
5. If privacy breached by (Psuper, C) = 0 then
6. prune_all_sub-instances_downwards (Psuper)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

7. Continue
8. Psub = next_sub-instance_of (Psuper)
9. If privacy_is_breached_by (Psub, C) = 1 then
10. Return 0 // early stop
11. While instance between (Psub, Psuper) do
12.I = next instance between(Psub, Psuper)
13. If privacy breached by (P,C) = 1 then
14. Psuper = P
15. Else
16. Psub = P
17. prune_all_sub-instances_downwards (Psub)
18. prune_all_super-instances_upwards (Psuper)
19. Return 1

3.Advanced Encryption system Analysis:

The Existing system provide 2 level security whereas our system provides 3 level protection. The Existing system uses the cloud for document storage, which is unsecured. No Internet No Access to the cloud, whereas our system provides offline access to the database.

III. IMPLEMENTATION & CODING

Access to the database for retrieval of the private data both at client and server side. It provides 3 level security. Along with that it can provide security under the circumstances like mobile stolen, password forgotten.

IV. CONCLUSION

In this paper, we list the main targets and content of Web security testing, and also introduce two important tools .Most importantly, this paper illustrates XSS and SQL injection attacks and also gives some methods and suggestion to defense these attacks. It is estimated that companies lose between 0.5 and 2.5% of their revenue because of security-related losses and downtime. Web security testing is really important for enterprises; not only code developers but also QA should take corresponding responsibilities for Web security testing. In the future, we will improve our methods to cope with more and more complex security attack and develop an automatically testing tool to solve XSS and SQL injection attacks.

REFERENCES

[1] 2010-2011 CSI Computer Crime and Security Survey by the CSI Computer Security Institute, http://goCSI.com
[2] 2010 Data Breach Investigation Report, available a thttp://www.verizonbusiness.com/resources/ex ecutives ummaries/es_2010-data-breachreport_en_xg.pdf
[3] 2010 Report the Identity Theft Resource Center (ITRC), http://www.idtheftcenter.org/artman2/ publish/lib_survey/ITRC_2009_Data_Breaches.shtml
[4] Computer Science Curriculum 2013, Ironman Draft (Version 0.8), November 2012,http://ai.stanford.edu/users/sahami/CS2013/ironman-draft/cs2013-ironmanv0.8. pdf
[5] Y.-W. Huang, F. Yu, C. Hang, C.-H. Tsai, D.-T. Lee, andS.-Y. Kuo,"Securing web application code by static analysis and runtime protection," in WWW '04: Proceedings of the13th international conference on World Wide Web. New York, NY, USA: ACM,2004,pp. 40–52.
[6] U. Shankar, K. Talwar, J. S. Foster, "Detecting format string vulner abilities with type qualifiers," inSSYM'01: Proceedings ofthe10th conference on USENIX Security Symposium. Berkeley, CA, USA: USENIX Association, 2001, pp. 16.
[7] J. Krinke, "Information flow control and taint analysis with dependence graphs," in 3rd International Workshop on Code Based Security Assessments. (CoBaSSA 2007), 2007, pp.6–9.
[8] V. B. Livshits and M. S. Lam, "Finding security vulnerabilities in java applications with static analysis," in SSYM'05: Proceedings of the 14th conference on USENIX Security Symposium. Berkeley, CA, USA: USENIX Association, 2005, pp.271–286.
[9] L. Wang, Q. Zhang, and P. Zhao, "Automated detection of code vulnerabilities based on program analysis and model checking," in Source Code Analysis and Manipulation, 2008 Eighth IEEE International Working Conference on, Sept. 2008, pp.165–173.
[10] W. Chang, B. Streiff, and C. Lin, "Efficient and extensible security enforcement using dynamic data flow analysis," in CCS'08: Proceedings of the 15th ACM conference on Computer and communications security. New York, USA: ACM, 2008, pp.39–50.