International Journal of Computer Architecture and Mobility
(ISSN 2319-9229) Volume 1-Issue 4,February 2013

Categorization of web application scanners and testing tools

Rajeshwari Uikey

rajeshwariuikey1987@gmail.com

***Abstract*:** Web application and tools are free to take any input, designers can't specified pattern of input provided by user but still various scanners and tools are available to check ,to test code, O.S module checking, language specification or syntax checking etc . Here, in the propose work various categorization of tools has been given to check, or scan web application.

I. INTRODUCTION

Web application suffers from various attack categories,out which SQl-Injection has discussed here**.** The Structural Query Language Injection (SQLI) attack occurs when an attacker changes the logic, semantics or syntax of a SQL query by inserting new SQL keywords or operators. SQL databases are attractive targets [4, 5, 7] because they often contain valuable information such as user names, passwords, e-mail addresses, personal data, and financial records. SQL Injection Attack is a class of code injection attacks that happens when there is no input validation. In fact, attackers can shape their illegitimate input[3,4,] as parts of final query sting which operate by databases. Banking web applications or secret information systems could be the victims of this vulnerability because attackers by abusing this vulnerability can threat their authority, integrity and confidentiality [4]. So, developers addressed some defensive and secured coding practices to eliminate this vulnerability but they are not sufficient. SQLIAs can also escape traditional tools such as firewalls and Intrusion Detection Systems because they performed through ports used for regular web traffic. SQL injection attacks can be carried out easily using only a web browser going through port 80 which is frequently left open by firewalls.

II. RELATED WORK

A real-time intrusion detection mechanism based on the profile of user roles has been present by Bertino et al. (2005). This approach is based on mining2, 3] SQL queries stored in database audit log files. The result of the mining process is used to form profiles that are capable to model normal database access behaviour and identify intruders.

Rietta (2006) proposed an application layer intrusion detection system, which should take the form of a proxy server and employ an anomaly detection model based on specific characteristics of SQL and the transaction history of a particular user and application.

Another approach, payload based anomaly detector (PAYL) extends the work in [5] by utilizing a different statistical model, a full byte distribution and the use of clustering [6] which they utilize to detect worms. Though they state that their approach could detect other types of attacks, it was designed specifically to detect worms. They tested their approach with other web attacks using the 1999 DARPA dataset[1,2]. They also used datasets from their university web server to detect Code Red and a buffer overflow attack. In terms of their character distribution these two attacks are vastly different to normal requests. In this paper we focus on the detection of more subtle attacks.

Anomaly detection [8, 9,] is an approach to intrusion detection that is complementary to the use of signatures. The anomaly-based detection techniques are successful for detecting new attacks. A thread base approach is used in [7] for anomaly detection but solution is static in nature and updating the system is tedious job. Control flow graph and program slicing is explained in [3] for the validation of user input but the focus is not most critical attacks like XSS and SQL Injection at application level.

Different types of learning-based anomaly detection techniques have been proposed to analyse different data streams. Valeur et al. proposed an anomaly-based system [5] that learns the profiles of the normal database access performed by web-based applications using a number of different models. Estevez. et al. proposed an approach [8] which used Markov model to detect the web application attacks. Their approach is based on the monitoring of incoming HTTP requests to detect attacks. Naiman uses the statistic methods [6] to analyse the data which are collected from web servers However, all of these researches just considered the validation of user input. They did not consider the relationship between the page transition with the page attributes and unreasonable user visiting behaviour.

III. PROPOSE WORK

In the proposed paper various web scanning techniques and methods has been discussed, as shown below

1) Static code checkers(It deals with malicious Programs):

   a) Splint: It is used to scan C source code for security vulnerabilities and programming design flaws.

b) Flaw finder: Performs security checks in source code and generates possible security weaknesses in code.

c) RATS: scans PHP, C/C++, Python & Perl languages source code files for security weakness and flaw present in programming.

d) Compaq Esc: Use to check Java code for finding errors that are even not detected at runtime.

e) UNO: Use to scan software defects like uninitialized variables, NULL pointer reference, Out of bound array indexing.

2) Runtime code checker (Deal with program code)

a) Propolice: it protects from stack smashing attacks.

b) Immune tool: It protects from buffer overflow coding flaws.

c) Purify plus: It detects memory leaks and problems present in software.

3) Profiling tools (Deals with O.S.):

a) Papillion: It screens and prevent malicious event by system users.

b) Valgrind: It profiles and debugs Linux executables.

c) Janus: It protects system calling from unauthorized access.

4) Penetration testing tools (Deals with complete networks):

a) Nmap: network port scanner for finding the open port.

b) Nessus: It finds Vulnerability present in a network.

c) ISS internet scanner: it scans host for vulnerability and protects from unwanted activities.

5) Application scanning tools (Deals with databases)

a) Appscan: checks malicious inputs injected by attacker in a web application.

b) Whisker: checks for CGI flaws.

c) ISS database scanner: Scans database server applications(SQL server , Oracle etc)

d) SQLler: checks injection flaws in URL.

e) SQL Injection Brute force: Performs logic for SQL Operations to detect and exploit SQL Injection Vulnerability.

f) SQLBrute: It is a tool for brute forcing data out of databases.

Other SQL-Injection tools:

BOB Cat, SQL map, Absinthe, SQL bf tools, SQLID, SQL power Injector, FJ-injection framework, sqlNinja, Automatic SQL injection.

5) Patterns based system tools

a) Field security validator and page security validator.

They are used to detect SQL-Injection script injection attack, here, strong algorithm and engines to analyse the input text for the possibility of attacks are used ,after detecting attack they generates log for future research based on log information's and blocks them.

Here, the levels of security are provided for the analysis of attacking patterns. If any attack if found by the application, it safely handles them without informing user .Thus provides good sense of security frameworks.

b) Slow down manager & log and respond engine.

Here, the systems monitor attacker's behavior, who hits a page several times for protecting against brute force attack.

If any malicious activity is found, the system blocks the page, and diverts the link of attacker to other page (page could be used by security validator).dummy page.

TABLE I

COMPARATIVE

Features / **Tools**	**Support ability**	**Up gradation**	**Techniques used**	**Failure cases**
Static code checkers	Languages	Less overhead	syntax	Depends on analysis
Runtime code checker	Code design	Design overhead	Symantic	Depends on design cases

Profiling tools	System process	System routine overhead	Network algorithms	Depends on network designs
Application scanning tools	Web design	Web access overhead	Web security techniques	Depends on security frameworks
Patterns based system tools	Security techniques	Pattern analysis overhead	Attack analysis and fingerprinting	Depends on detection methods

IV. RESULT

The web scanning tools and applications are shown or compared by different parameters, every tool and scanner has their own specific area and domain. Some tools scans only user application and codes abut some are have capability to scan system processes and module and provide transparency.

V. CONCLUSIONS

The application are always susceptible to attack, the only way to protect the system is s to create strong analysis and designed with best or updated parameters, here different tool are shown, they are best at their levels, as attacker are using advanced techniques, most secure algorithms are needed to provide safe environment for web or software applications.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their detailed, valuable comments and constructive suggestions.

REFERENCES

[1]Aziah Asmawi, Zailani Mohamed Sidek,Shukor Abd Razak,"System Architecture for SQL Injection and Insider Misuse Detection System for DBMS",IEEE 2008.

[2]Abdul Razzaq, Ali Hur, Nasir Haider, Farooq Ahmad,"Multi-Layered Defense against Web Application Attacks",IEEE 2009.

[3]Yu-Chin Cheng ,Chi-Sung Laih ,Gu-Hsin Lai ,Chia-Mei Chen ,Tsuhan Chen,"Defending On-Line Web Application Security with User-Behavior Surveillance",IEEE 2008.

[4] D. Stuttard and M. Pinto, "The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws", Wiley Publishing Inc., 2007

[5] Symantec. "Symantec Report on the Underground Economy", 2008.

[6] F. Valeur, D. Mutz, and G. Vigna. A Leaing-Based Approach to the Detection of SQL Attacks. In Proceedings of the Conference on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA), Vienna, Austria, July 2005.

[7] M. Howard and D. LeBlanc, "Writing secure code", Microsoft Press, 2003

[8] R. McClure and I. Kruger, "Sql dom: Compile time checking of dynamic sql statements," in Proceedings of the 27th International Conference on Software Engineering (ICSE 05), St. Louis, Missouri, USA, May 2005.

[9] S. W. Boyd and A. D. Keromytis, "Sqlrand: Preventing sql injection attacks," in Proceedings of the 2nd Applied Cryptography and Network Security (ACNS) Conference, June 2004, pp. 292–302.