International Journal of Computer Architecture and Mobility
(ISSN 2319-9229) Volume 4-Issue 6, June 2016

SQL –Injection Attack Detection for infected SQL Queries

Priyanka Chauhan
chauhanpinku07@gmail.com
Research Scholar, Department of Computer Science and Engineering,
Oriental University, Indore

Abstract— Web application are invariably vulnerable to attacks, as several assault techniques are gift for internet application, banking ,reservation, education etc and numerous applications are utterly obsessed with web.SQL Injection attack, one in every of the main and open assault technique ,which destroys internet application by the analysis of information process and signatures. The SQL Injection attack happens thanks to modification of dynamically generated question. Here are the mapping techniques checks the believable of question with the assault points, if found in question, It if tested on totally different question data set and also the results are comparable.

I. INTRODUCTION

THIS Application security problems will increase apace further and internet applications are getting additional liable to worrisome vulnerabilities. during this chapter, we have a tendency to describe initially what internet applications [1] are, that structure they typically have and the way they move with admin, then we have a tendency to provide an summary of the common security issues self-addressed in internet applications. A Web application could be a client/server package application that interacts with admin or different systems victimization the machine-readable text Transfer Protocol (HTTP) [2].Companies don't have homepages anymore; they use dynamic internet applications for interacting with admin instead not like the standard static websites, that price quite substantial time and energy once content must be updated, internet applications generate web content [3] betting on user data and requests The great flexibility of dynamic internet applications makes the net additional interactive. However; it's answerable for a rise within the variety of security incidents. Attacks on internet applications are increasing dramatically. in keeping with the Symantec web Security Threat Report7 revealed recently, sixty nine of all vulnerabilities were related to internet applications within the last half of 2010; this represents a four-hundredth increase over in 2014 because it was accounted for forty ninth [4]. Internet applications are getting additional and additional vital. For a few businesses, the net application is that the single purpose between business and customers, so internet security becomes a good additional vital issue [5].Web applications would possibly touch upon sensitive data like financial data and medical data, it'll cause important injury, once attackers gain unauthorized access to the current data. Protection of networks and their services from unauthorized modification destruction, or revealing, and provision of assurance that the network performs its essential functions properly and there aren't any harmful side-effects[6].

SendsRequest:
The first tier may be a applications programmer like web soul, Mozilla Firefox and web browser, etc. The Presentation Layer generates the computer program, sometimes a dynamic electronic computer, within which case it's enforced on an internet server [8].

HandlesRequest:
The middle tier is Associate in Nursing engine, that generates pages dynamically mistreatment [8] technologies like PHP machine-readable text processor, Active Server Pages technology (ASP), and Java Server Pages technology (JSP). The Business Logic Layer contains the application's business rules and controls its flow. It's enforced on the appliance server and its elements are usually interfaced with net services.

GeneratesResponse:
The third tier may be info; it permits net applications to store information and alternative content parts. By mistreatment the structured source language (SQL), net applications will act with information bases to make tailor-made data for every user dynamically. The

information Layer manages the application's data. It is enforced by the direction system (DBMS) on the info server, and optionally contains the next level information Access Layer (DAL) like a persistence framework (e.g. Hibernate, Oracle prime Link) running on the appliance server. Fig 1.1 shows Static internet sites and Fig one.2 shows Dynamic net Applications structure. The responses are sent back to the online client: A shopper (Web browser) sends requests to the center tier that handles these requests, searches info needed by creating SQL queries against the info and generates response pages mistreatment this info, and shows them to the user within the browser [8].

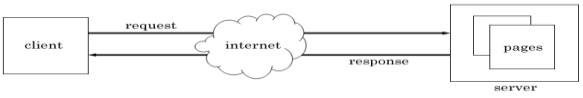


Fig 1.1 Static Web Sites [8]

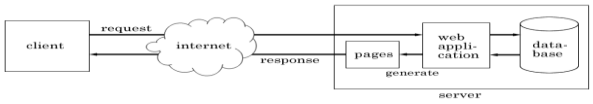


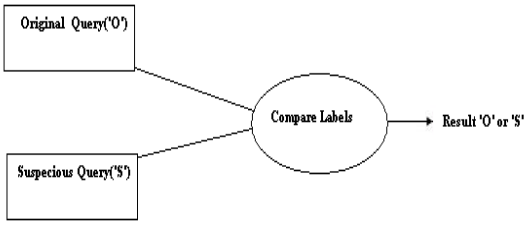
Fig 1.2 Dynamic Web Applications [8]

II. Related Work:

IDBC-Checker [4, 5] wasn't developed with the intent of police investigation and preventing general SQLI As, however will be wont to stop attacks that cash of sort mismatches in an exceedingly dynamically-generated question string. Xiang Fu and Kai Qian [7] projected the look of a static analysis framework, known as SAFELI for distinctive SQLIA vulnerabilities at compile time. CANDID [8, 9] modifies net applications written in Java through a program transformation. This tool dynamically mines the programmer-intended question structure on any input and detects attacks by scrutiny it against the structure of the particular question issued. CANDID's natural and simple approach seems to be terribly powerful for detection of SQL injection attacks. In SQL Guard [2] and SQL Check [3] queries square measure checked at runtime supported a model that is expressed as a descriptive linguistics that solely accepts legal queries. SQL Guard examines the structure of the question before and when the addition of user-input supported the model. In SQL Check, the model is nominal severally by the developer. Each approaches use a secret key to delimit user input throughout parsing by the runtime checker, therefore security of the approach relies on attackers not having the ability to get the key. In 2 approaches developer ought to switch code to use a special intermediate library or manually insert special markers into the code wherever user input is supplementary to a dynamically generated question.

AMNESIA combines static analysis and runtime watching [9]. In static section, it builds models of the various styles of queries that AN application will wrongfully generate at every purpose of access to the info. Queries square measure intercepted before they're sent to the info and are checked against the statically designed models, in dynamic section. Queries that violate the model square measure prevented from accessing to the info. The first limitation of this tool is that it success relies on the accuracy of its static analysis for building question models. Webs dress use static analysis to see taint flows against preconditions for sensitive functions. It works supported alter input that has responded to a predefined set of filters. The limitation of approach is adequate preconditions for sensitive functions can't be accurately expressed therefore some filters is also omitted [10].

III. Proposed Technique:



Here unique technique is used for distinguishing original query and malicious query,

Original Query: select * from admin where uid=1;

Malicious Query: select * from admin where uid= ""OR 1=1;

Below table shows mapping approach.

Table-1(Original query mapping table)

Tokens	Mapping Function values(1 to 1 and 1 to many)	
Select	*	0 mapping
*	Select	From
From	*	Admin
Admin	From	Where
Where	Admin	Uid
Uid	Where	=
=	Uid	1
1	=	0 mapping

Here in original queries, different tokens are given and mapped with their sequential next or previous tokens, If both tokens occurs that is, in next or in previous order 1 to many mapping occurs, and if Either next or previous occurs 1 to 1 mapping occurs.

Here, mapping function verifies the occurrence or connectivity order of tokens existing in the SQL-Query .If the order ,sequence or mapping breaks ,that means there is SQL-Injection attack, means extra tokens are inserted or the query is been modified, if order or mapping remains same there is no SQL-Injection, query is free for further execution.

Table-2(Malicious query mapping table)

Tokens	Mapping Function values(1 to 1 and 1 to many)	
Select	*	0 mapping
*	Select	From
From	*	Admin
Admin	From	Where
Where	Admin	Uid

When both the tables are compared original and malicious ,it is found the length of tokens are different and the mapping function values are irregular and are not satisfying the sequential order, that means there is SQL-Injetion, But if all the conditions are satisfied ,that means the query is free from sql-injection.

IV. Result:

The system is tested on varieties of SQL-queries and parameters, Different SQL query schema has taken for finding the vulnerability points and mapping analysis functions The accuracy of proposed system is found to be 93%, and which is comparable, The system detect all types of Injection techniques .The architecture of mapping function provides best results as it compares and creates true mapping values. The technique could be easily implemented on existing system with negligible overhead.

References

[1] Ntagw Abira Lambert and Kang Song Lin ," Use of Query Tokenization to detect and prevent SQL Injection Attacks", IEEE,2010

[2] Shaukat Ali, Azhar Rauf and Huma Javed," An authentication Mechanism Against SQL Injection", In European Journal of Scientific research Vol.4(2009),pp 604-611

[3] C. Gould, Z. Su, and P. Devanbu," Static Checking of Dynamically Generated Queries in Database Applications". In Proceedings of the27th International Conference onSoftwareEngineering (lCSE 06), 2006.

[4] Xiang Fu , Kai Qian," SAFELI-SQL Injection Scanner Using Symbolic Execution". Proceedings of

the 2008 workshop on Testing, analysis, and verification of web services and applications. (2008).pages 34-39: ACM.

[5] Peter Scherer, Martin Vicher, Jan Martinovic,"using svm and clustering algorithms in ids system", pp. 108-119, ISBN 978-80-248-2391-1, 2011.

[6] Ankit Anchlia, Sheela Jain," A novel Injection Aware Approach for the Testing of Database Applications", IEEE 2010.

[7] Kang song lin, Ntagw abira lamber," Use of Query Tokenization to detect and prevent SQL Injection Attacks",IEEE 2010,

[8] Mehdi Kiani, Andrew Clark and George Mohay," Evaluation of Anomaly Based Character Distribution Models in the Detection of SQL Injection Attacks", The Third International Conference on Availability, Reliability and Security, IEEE 2008.

[9] Abdul Razzaq, Ali Hur, Nasir Haider and Farooq Ahmad," Multi-Layered Defense against Web Application Attacks", Sixth International Conference on Information Technology: New Generations 2009.

[10] Melanie R. Rieback, Bruno Crispo and Andrew S. Tanenbaum," Is Your Cat Infected with a Computer Virus", Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'06), IEEE 2006.