ISSN: 0975-766X
CODEN: IJPTFI
Available Online through *Research Article*
www.ijptonline.com

DETECTION AND PRECLUSION OF SQL INJECTION IN A DISTRIBUTED ENVIRONMENT USING CONTEMPORARY APPROACH

Rubidha Devi.D[1], R.Venkatesan*[2], Raghuraman.K[3]
[1,2,3,] Department of Computer Science and Engineering, Srinivasa Ramanujan Centre,
SASTRA University, Kumbakonam – 612001, Tamil Nadu, India.
Email: rubidhadevi@src.sastra.edu

Received on: 20.10.2016 *Accepted on: 25.11.2016*

Abstract

SQL injection is a code injection technique; it has been a top most security threat to web applications and it mainly targets the databases that are accessible through a web application. Any attacker can embed an injected string into the original query, which poses a serious threat to web application security. Attacker exploits the vulnerability by inserting crafted SQL keywords and values, effectively altering the semantics of dynamic queries, and causing them to return expected results. Besides many tools and research prototypes are available, that tool doesn't detect and prevent the injection attacks accurately in a web application. The existing system supports only single web application SQL injection attacks. In this experiment, a novel approach is followed by developing a hybrid model that prevents brute force attack at the initial stage. If exploited, an encrypted query parser tool will block the attacker from grabbing data from the database. This approach can also be ported to other web application development platforms without requiring major modifications. The proposed system can protect attackers attempting to intrude from multiple Internet Protocols interacting with a database server, which is an advantage over existing methods.

Keywords: SQL Injection, Tautological attack, Brute Force, SQLIPDT, Fencing tool, Query Parser.

1. Introduction

SQL Injection attack against web databases has become a severe threat. The significance of SQL injection can be understood from an article[1] stating, this attack stands first among top ten security threats. To add more fuel to this arena, According to Trustwave 2012 Global Security Report[2], SQL injection attack leads number one for the past four consecutive years. An online resource[4] indicates that SQL injection also has a special crown that breaches data around

97% across the world. The following are some of the famous areas where this attack has been used. [5] WHO Employee's personal details were stolen, Wall Street Journal's data has been pilfered and using this attack, invaders exploited the sites of US federal agencies too. [9]Reserve Bank of India's website has been tried to get hacked multiple times.

In this global village, e-banking sites, e-commerce and m-commerce has become part of our life. In all the above mentioned domains people have provision to share their private information like phone number, address and sometimes their credit card number, bank a/c number etc. Mischievous people, who are interested in ethical hacking finds their means to grab all the contents that are freely and easily available in these web portals. The websites which are vulnerable helps the attackers to excerpt the data from the data warehouse. Attackers use botnets[3] to discover the vulnerable web pages automatically.

All these scenarios grabbed our attention to this area to try a novel approach to strengthen the schema.

2. About SQL Injection

This attack has got its own style, because all the web applications deals with queries and attackers choose websites to penetrate into user's data warehouse.

The data security is breached out by injecting the queries through webpage URL. Therefore, the web application developers must focus on developing web pages using effective prevention mechanisms[10].

3. Causes and Effects

A study was conducted in which it says a list of prevention and detection mechanisms[10]. Despite of all these security measures, data loss due to SQL attack keeps continuing.

3.1 SQL Injection Types

Enormous methods have been found in this technique. Some basic methods were grouped by William G.J. Halfond et.al.[13]. Let us discuss few techniques in brief.

1. *Tautologies Based SQL attack*: This type of attack is very common to break the conditions by applying OR operator in WHERE clause.

Example 1: Select * from usertable where 1=1;

Example 2: Select * from userinfo where username = 'abc' and password = 'anything';

From the above examples we can conclude that the query is always true.

2. *Piggybacked Queries attack:* This attack is miserable. In a single query it leads a link to the next query.

Example: select * from usertable where userid = 'abc' and pwd ='''; drop table emp—'

3. *Union Query:* This attack will use UNION keyword for retrieving data from entire table.

Eg: select * from usertable where userid= "" union select * from emp--" and pwd='234'

Here the comment operator (--) will ignore the statements following it. That is the rest portion will never be executed.

4. *Illegal & Logically incorrect queries:* The queries are injected to retrieve the table name or column name for further hacking using some incorrect queries.

5. *Inference:* In this attack, the hacker camouflages himself as a regular user or web portal viewer even if the condition is false. The hacker tries to attack blindly along with the time based attack (performs delay if condition is false).

6. *Stored Procedures:* Normally, batch query processing will be programmed using stored procedure. If the user finds the table name and column name using Illegal/ logical incorrect queries, the injection will be done inside the stored procedures which affects the entire database.

Example: select * from usertable where userid='' and pwd='''; SHUTDOWN;--'

These queries will shut down the entire database.

To overcome all the above mentioned methods, we list some of the prevention and detection mechanisms that has been widely accepted[10].

3.1.1 Materials and Methods

[6]Code injection and detection tool is proposed to deal with SQL injection attack. Algorithms like Query detector, script detector and modified data cleansing algorithm have been used to trace Maximum vulnerabilities.

The false alarm rate is very high in case of parse tree validation and if code convention method is used, obviously it will increase the volume of the database vigorously. In order to overcome this problem, cookies/server variables can be used[11].

SQL injection is detected and prevented using hidden web crawling with the combination of parse tree and digital signature[10]. Access Authorization Data Table (AADT) is used to maintain all authorized rights to access information. Divya Jain[10] has achieved 50% of prevention. If Parse Tree mechanism is being used, chances are high that hackers can masquerade themselves as a normal user and can excerpt the data from database.

[12]Another Detection and prevention is done using Fuzzy parameters with two main rules. These two rules help to knock out the difference between authorized and unauthorized users. Unauthorized users are automatically blacklisted with 3 false attempts.

The inputs given by the user and the execution time are saved in database for further testing. The author extends further by using Meta heuristic technique to make prevention better.

[16]Query transformation and document similarity measures are presented for detecting SQL injection attacks. Multiple web applications hosted on web server is protected by the system which acts as a database firewall. They have used basic semantics for their schema, which is considered as weaker region. In future weighting schema are expected which reduces false negative/positive rates.

4. Proposed System

From the above discussions we came to know that even so many mechanisms are identified, attackers still have opportunities to exploit the database. We followed a novel approach for prevention and detection[12, 16] mechanisms.

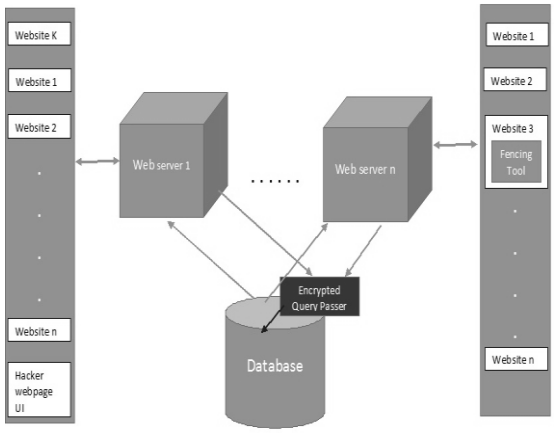

Figure-1 SQLI Prevention and Detection too.

Explanation:

Figure 1 shows overall system architecture of SQLIPDT (SQL Injection Prevention and Detection Tool) that has the following core components:

1. Webserver
2. Database
3. Encrypted query parser

4. Vulnerable Websites
5. Hacker's webpages.
6. Fencing tool

4.1 Web Server

It performs the routine methodology of working and communicates with the database for data manipulation. SQL injection must be prevented here.

4.2 Database

It is the crux that contains all the confidential data of end users. Only requests made by authorized webpages and authenticated queries should be sent to it, in order to avoid unnecessary query injections.

4.3 Encrypted Query Parser

Before explaining the need for this query parser, first let us understand the way how tautological attack is done through SQL injection.

Step 1: The attacker discern the list of vulnerable web pages using some predefined tools like Botnets.

Step 2: Endeavor to inject the vulnerable web page using tautological attack.

Step 3: Seize the data from the database.

To prevent this, we introduce the so called "encrypted query parser" which slabs the injected query from the database hit. It is a tool specially designed with encrypted semantics for all the keywords used in the SQL Query.

Table 1. Encrypted semantics.

Characters	Semantics
@	IBTI
#	J
$	EMMS
%	QSDOU
^	YPS
&	CJUBOE
|	CJUPS

Table 1 shows the sample encrypted semantics that the query parser tool uses to authenticate the incoming queries to the database.

4.4 Vulnerable websites

These are found easily by the attackers. There are number of tools available over the internet for finding the vulnerable pages at a fraction of a second. These pages are the loopholes for the hackers to inject the query.

In Figure 1, the webpage 'k' has been identified as the vulnerable web page by the attacker. Once it has been found, chances are high that the attacker can wander into the database and exploits it.

4.5 Fencing Tool

Fencing tool is especially developed for the brute force attack to breach the security. It is quite common that all the websites provide 3 attempts for the users to type their credentials correctly. There is a catch in it. Let us assume we have 3 valid attempts to login into user account. If it is negative the user account gets locked temporarily. During the first attempt, 1000 and 1000 of user accounts is snatched from the database.

Brute force attack is a method developed for password identification by trying all the possibilities including the hash combination. Sometimes, even encrypted passwords can also be hacked using this strategy. Actually this attack is easy to understand but difficult to prevent [14]. This also attracted to work on this area.

A hacker uses XML-RPC for injecting the code in to the website and tries more combinations within first attempt. Our Fencing tool added to the web page will prevent from Brute Force attack. Figure 2 shows how this attack works to get more possibilities.

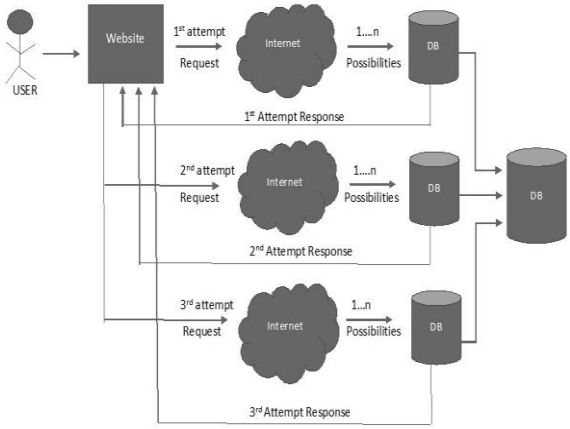

Figure 2. Brute force prevention and detection tool.

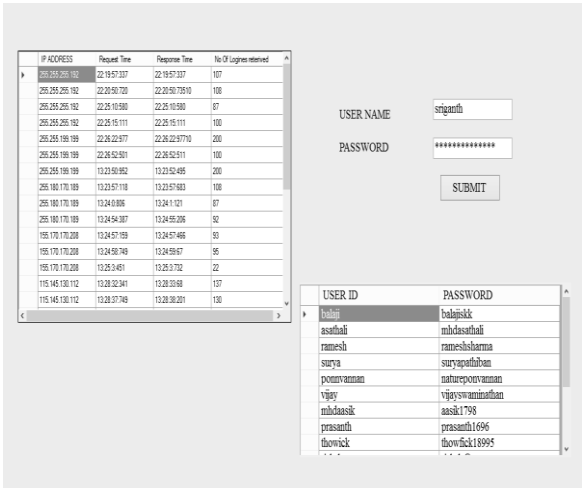

Figure 3. Brute force attacking tool.

We designed a Brute force attacking tool for snatching login credentials from the database. Figure 3 illustrates how Brute force attempts to fetch number of data from the database at its initial attempt.

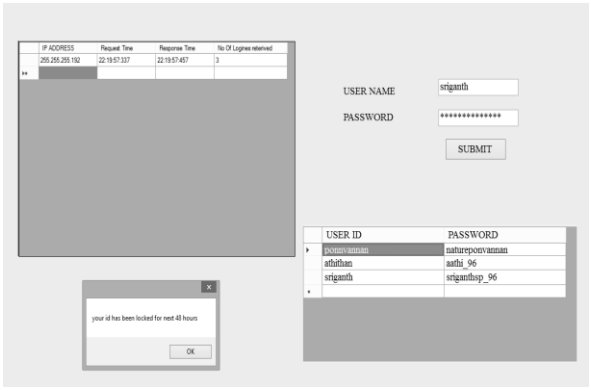

Figure 4. Brute Force Prevention and Detection Tool.

Figure 4 shows how fencing tool will prevent this Brute force attack in its 4th attempt done in first cycle. IP address of the attacker is traced and blocked for the next 48 hours for security reasons.

5. Experimental Evaluation

We developed a hybrid model of technique as a tool that traces the hackers in many ways. Unauthorized users are prevented from access. The Primary focus is to stop brute force attack. For example bank sites, social sites, mail & storage drive may contain sensitive data.WordPress[15] has protected this brute force attack using wordfence tool where the solution is kept as secret. We tried to use this approach in our paper for powerful security shown using relevant screen shots. Now our fencing tool will detect the multiple attempts in database at the third instance of the first attempt. For security reason the IP address of the attacker is captured and blocked for further access. In case, if the attacker

bypasses this stage, then we need to provide the next level of security in our database using "Encrypted Query Parser" under which we have three modes of securities.

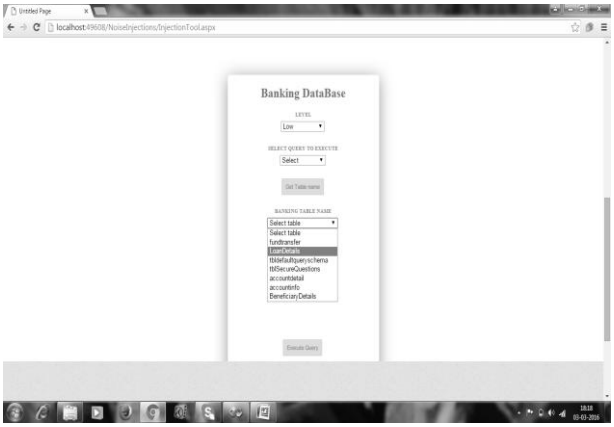

Figure 5. Low level security.

Figure 5 shows the First level as a preliminary mode where there is no security provided as, the attackers can enjoy all types of privileges. Following are some of the web portals that has this type of security (Tutorial point, Game zone) etc., may not create interest among attackers to hack it.

Figure 6. Medium level security.

The medium level security is displayed in Figure 6, where the attackers are restricted from manipulating the database. Only the attacker can only fetch the data. A bit of hard work can make the attackers to break this level of security and gain access to the core database. For example Shopping zones may have the midlevel security because it never allows anyone to add new item/product without DBA knowledge. It gives access to only view the products.

Figure 7 shows the high level security. In this level all privileges are blocked for unauthorized users. We do this using "encrypted query parser" where the keywords used in queries are encrypted thereby preventing tautological attack.

Even though, if the query tries to infringe, error messages thrown out will show only the message typed by the web developer and not the exact error generated by the server.

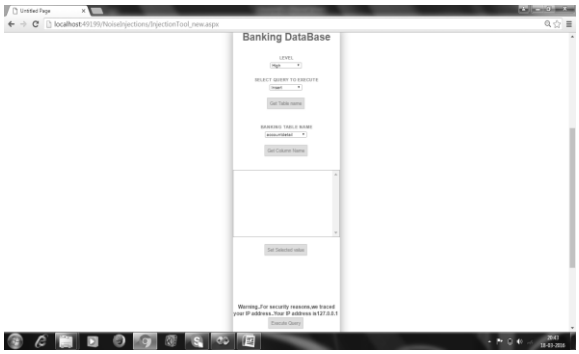

Figure 7. High Level security.

To check the efficiency of the tool we developed, it has been installed and checked in and around 300 machines with the database server and tried to attack the websites using brute force which is blocked. After a long duration if the attackers bypass this stage we tried to inject the webpage using tautological attack.

6. Future Enhancements

We have tried to give security at the initial level using fencing tool, which blocks the attackers to penetrate inside the database from the same system for the next 48 hrs. Even if the attacker traverse the initial stage, we transformed the way of querying into a semantic based querying process using encrypted query parser where SQL injection can be prevented at a marginal level. In future, our team will focus on strengthening our schema and the query using fuzzy sets also.

7. References

1. Article title: https://www.owasp.org/index.php /Top_10_2013-A1-Injection. Date accessed: 10/08/2016.
2. Article title: Trustwave: Executive Summary: Trustwave 2012 global- security -report. https://www.trustwave.com/global-security- report (2012) Accessed: 2013-06-24.
3. Maciejak, D, Lovet,G "Botnet-Powered Sql Injection Attacks: A Deeper Look Within." In: Virus Bulletin Conference, pp. 286–288 (September 2009) https://www.virusbulletin.com/conference/vb2009/abstracts/botnet-powered-sql-injection-attacks-deeper-look-within/
4. Aticle title: http://www.techworld.com/news/security/barclays-97-percent-of-data-breaches-still-due-sql-injection-3331283/ Date Accessed: 13/08/2016
5. Article title: http://motherboard.vice.com/read/the-history-of-sql-injection-the-hack-that-will-never-go-away Date last accessed : 13/08/2016

6. Digambar Patil, Abhishek Bhardwaj, Laxman Teli, Dattatray Lugade, , Vijay Girange, "Defeating SQL injection using Data cleansing algorithm". Asian Journal of Engineering and Technology Innovation 03 (06); 2015; 07-09.
7. Pratik H Sailor, Prof. Jaydeep Gheewala. "Detection and Prevention of SQL Injection Attacks", International Journal of Engineering Development and Research (IJEDR), ISSN:2321-9939, Vol.2, Issue 2, pp.2660-2666, June 2014, Available :http://www.ijedr.org/papers/IJEDR1402215.pdf.
8. Article title: http://timesofindia.indiatimes.com/business/india-business/Hacking-attempt-on-RBI-website/articleshow/13491399.cms.
9. Tejinderdeep Singh Kalsi, Navjot Kaur. "Detection and Prevention of sql injection attacks using novel method in web applications", International Journal of Advanced Engineering Technology (IJAET), ISSN: , Vol. VI, Issue IV, pp.11-15, October-December 2015.
10. Divya Jain, Naveen Choudhary, "An Automatic Detection System for SQL Injection" International Journal of Computer Applications (0975–8887)Volume 126 –No.11, September 2015. Available: http://www.ijcaonline.org/ research/ volume126 /number11/jain-2015-ijca-906218.pdf
11. Ashish John, Ajay Agarwal, Manish Bhardwaj, **"**An Adaptive Algorithm to Prevent SQL Injection", American Journal of Networks and Communications. Special Issue:Ad Hoc Networks. Vol. 4, No. 3-1, 2015, pp. 12-15. doi:10.11648/j.ajnc.s.2015040301.13. Available:http://www.sciencepublishinggroup.com/specialissue/paperinfo.aspx?journalid=132&specialissueid=132009&doi=10.11648/j.ajnc.s.2015040301.13.
12. Er. Kanika, Er. Prabhjot Kaur. "A Dynamic Approach to Detect & Prevent SQL Injection Attack to Overcome Website Vulnerability", International Journal of Innovative Research in Science, Engineering and Technology, Vol. 4, Issue 12, December 2015. Available: http://www.ijirset.com/upload/2015/december/152_A_Dynamic.pdf.
13. William G.J. Halfond, Jeremy Viegas, and Alessandro Orso, "A Classification of SQL Injection Attacks and Countermeasures" IEEE-2006. Available:http://www.cc.gatech.edu/~orso/papers/halfond.viegas.orso.ISSSE06.pdf.
14. Article title: https://www.owasp.org/index.php/Blocking_Brute_Force_Attacks Last Accessed : 13/08/2016
15. Article title: https://wordpress.org/plugins/wordfence/,Last Accessed: 13/08/2016.