How to Prevent SQL Injection Attack Based on Web Applications

Zheng Haiyan*, Wu Weituan, Zhang Ruili
Qingdao Branch of Naval Aeronautical Engineering Institute, Qingdao 266041, China
** Corresponding Author*

ABSTRACT: With the advent of automated injection attacks, SQL injection attacks in web applications is more common, Related technologies has been refurbished. SQL injection is modify Modification the pages of the website database, it can add a user with administrator privileges directly in the database, and ultimately gain administrator privileges, the paper analyzes the principles of SQL injection attacks, procedures, and types. We propose the effective method of in preventing SQL injection attacks.

KEYWORDS: SQL injection attacks; SQL injection attack prevention; Network Security

1 INTRODUCTION

With the rapidly growing popularity of the Internet industry and the rapid development of network technology, more and more applications move to the network. These models of network applications are generally B / S, mostly using the Web scripting languages (such as ASP, PHP, etc.) with the database (such as My SQL, Oracle, etc.) development, which gives hackers provides a convenient to attack the network application layer, because no matter how strong the firewall rule or patching vulnerabilities mechanism, if network applications programmer does not follow the safety code development, the attacker access system through TCP 80 port, SQL injection attacks (SQL injection At tack) is one of the attack techniques are widely used.

2 PRINCIPLE OF SQL INJECTION ATTACKS

SQL injection attack is an attacker deliberately submit specific SQL statements mixed with input from the client, to make these SQL statements can be mixed into a normal SQL statement to be executed in the system, in order to obtain sensitive information, destroy database, and even control the server attacker.

SQL injection is usually modify database through a web site. It can add a user with administrator privileges in the database directly, so that acquiet the system administrator privileges ultimately. Hackers can use it to obtain any access documents on the website or a variety of Trojans and malicious programs add on the web, theses has tremendous harm to the site's users.

Because SQL injection attacks is a legitimate SQL syntax, this attack can not be detected by the firewall, with a difficult recognition; Theoretically, any standard database based on SQL language are applicable,suchasMSSQLServer,Oracle,DB2,Sybase, and so on.

3 TYPES OF SQL INJECTION ATTACK

3.1 *Attack of that is not properly filtered triggered character*

When the user’s input is not filtered for escape characters, such injection attacks is occur, it will be passed to a SQL statement, which would lead to the application end-user manipulate database statement. The following code will demonstrate this vulnerability:

statement: = "SELECT * FROM users WHERE name = '" + userName + "';"

The code is designed to remove a specific user from the user table, but if the user name is a malicious user in a specific way forgery, operation performed by this statement may be more than expect of the code author. For example, the user name variable (ie username) is set to: a 'or' t '=' t, this time the original statement will change:

SELECT * FROM users WHERE name = 'a' OR 't' = 't';

If this code is used in an authentication process, it can force the user to select a legitimate name, because the assignment 't' = 't always right.

On some SQL Server, such as SQL Server, any SQL commands can be injected through this method, including the multiple statements implement. Value of following statement username will result in delete "users" table, they can select all the data from the "data" table (in fact, revealed each user's information).

a '; DROP TABLE users; SELECT * FROM data WHERE name LIKE'%

eventually, a SQL statement is:

SELECT * FROM users WHERE name = 'a'; DROP TABLE users;

SELECT * FROM DATA WHERE name LIKE '%';

Other SQL execution will not perform the same query multiple commands as a security measure. This will prevent an attacker from injecting entirely separate queries, but it does not prevent an attacker to modify the query [1].

3.2 *Attacks trigger by Incorrect type handling*

If the field is not a user-provided a strongly typed, or no implementation type coercion, this form of attack will occur. When using a numeric field in a SQL statement, such an attack will occur if the programmer does not check the validity of user input (whether numeric type).

For example:

statement: = "SELECT * FROM data WHERE id =" + a_variable + ";"

From the statement can be seen, the author hopes a_variable is a figures relate to"id" field. However, if the end user select a string, bypassing the need for escape characters. For example, a_variable set to: 1; DROP TABLE users, it will remove "users" table from the database, SQL statement is:

SELECT * FROM DATA WHERE id = 1; DROP TABLE users;

3.3 *Attacks triggered by Database server vulnerability*

Sometimes, the database server software have loopholes, such as MYSQL server has mysql_real_escape_string()function.This vulnerability allows an attacker implement a successful SQL injection attacks according to Unicode based on incorrect.

3.4 *Blind SQL injection attacks*

When a Web application vulnerable to attack while the results could not seen by attackers, it will happen blind SQL injection attacks. There are not display the data in the loopholes page, but display different content according to the result of a logical statement injected into the legitimate statement. This attack is quite time-consuming, because it must carefully construct a new statement for each obtained byte. However, the location and the target information location of vulnerability once is established, a tool called Absinthe can make this attack automation[2].

These are only a rough classification to SQL attack. But Technically, SQL injection attacks who identify vulnerabilities in websites has become more intelligent and more comprehensive. There have been some new SQL attacks. Hackers can use various tools to accelerate the use of loopholes in the process. For example, the Asprox Trojan horse, it is mainly released through a botnet to spread the message, the entire work process can be described: First, it installs Trojan on the computer through spammers of the controlled host, then, infected PC will download some binary code, when it starts, it will search a website which are loopholes create by Microsoft's ASP technology using a search engine, Search results has become a target list of SQL injection attacks. Next, the Trojan will start SQL injection attacks to these sites, so some sites are controlled and destroyed. Users access these sites will be deceived, download some malicious JavaScript code from another site. Finally, the code directs the user to a third site, there are more malicious software, such as Trojans steal passwords.

4 DETECTION AND PREVENTION OF SQL INJECTION ATTACK

4.1 *Detection both the client and server side*

For SQL injection attacks, the client and server-side should have two-stage testing, as long as any level detection does not pass, the information submitted will not enter the query statement, it can not constitute an attack. The main effect of client detects is to reduce network traffic, reduce server load, separate the general misuse, low-level attacks and high-grade aggressive behavior. From a technical perspective, the experienced attacker may bypass the detection of the client, at this time, the data presented will be sent directly to the server, so the server side setting two detection is necessary. Because normal data submitted to the server-side has been detected on the client, the server detects abnormal submit behavior can be identified as malicious attacks, at this time, we should suspend the processing to submission information, record the attack, the client should have an error message, as Figure 1 shown.

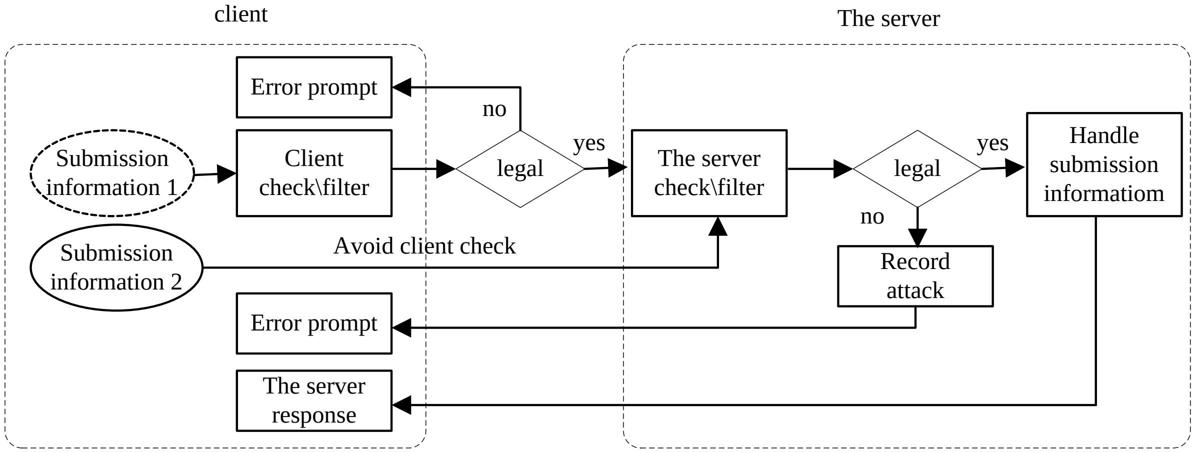

Figure 1 Defense flowchart of SQL injection attack

4.2 *Security configuration of Web server*

(1) Modify the initial server configuration: it is promptly remove permissions account or modify permissions of the login account name after the installation is complete. Close all service port, then open the port to be used.

(2) Timely installed server security patches: the server keep the latest service pack, run the stable version.

(3) Shut down the server error message: Set yourself a wrong message, all errors are only returned an error message, so an attacker can not obtain valid information.

(4) Configuration directory permissions: the directory where the Web application can be set as read-only files, which is uploaded by the client is stored separately and is set to no executable permissions, System files which is not allow Web access directories to store secret.

(5) Remove the dangerous server components

(6) Tracking and timely analysis of system logs: trace log is useful for the diagnosis of the attack, the past six months of SQL injection attacks are occurring by malicious HTTP requests. The log server program stored in a secure directory, the log files are analyzed on a regular basis, so that the first time to find invasion. On the server URI query string check can help confirm the attacks, and can become the starting point for future investigations[3].

4.3 *Security Configuration Database*

(1) Modify the database initial configuration: After installation is complete, it should promptly remove the default permissions of the account or modify the default login name.

(2)Update the database promptly: a database module must update timely, especially patches which is help improve database system security by office, it can solve a known vulnerabilities of database.

(3) Minimum rights law: Web Applications connect to the database account only have the necessary permissions, which protect the entire system invasion as little as possible. with a different user account query, insert, or delete operation, you can prevent to insert malicious perform INSERT, UPDATE, or DELETE statement while executing SELECT.

4.4 *Settings script parser Security*

For the PHP programming language, you can configure some security settings in PHP inifile, these can increase the difficulty of SQL injection, reduce the risk of SQL injection.

(1) Set "magic_ quotes_ gpc" to "on"

This option can enter some special characters escaped automatically.

(2) Set "register_ globals" to "off"

"register_ globals" set the enable / disable PHP to create a global variable for user input, set to "off" is: If the user submits the form variable "a", PHP will not create "& a", but will only create _GET ['a'] or _POST VARS ['a'].

(3) Set "safe_ mode" to "on"

Enabling this option will increase several limitations: specified command can be executed, the specified function can be used, based on a ownership of script file and target file access restrictions, prohibit file upload.

(4) Set "open_ basedir ' to" off "

It can prohibit operations the file of outside the specified file directory, effective solution attacks by include () function .

(5) Set "display_ errors" to "off"

At this point, the error message is prohibit display on the page, as these statements may return some application information related to the variable name, database user name, the table structure and other. A malicious user may inject attacks using obtain information. You can also set this option to "on", but we must modify error message return by the script, there are only display a message when an error occurs[4].

4.5 *Using parameterized filtration statements*

User input must filter or use parameterized statements to defense SQL injection. Parameterized statement uses parameters instead of user input is embedded in statement. Here is an example using Java and JDBC API:

Prepared Statement prep = conn.prepareStatement ("SELECT * FROM USERS WHERE PASSWORD = ")?; prep.setString (1, pwd);

4.6 *Avoid using an interpreter*

Interpreter is a means whereby Hackers execute an illegal command.

4.7 *Using professional vulnerability scanning tools*

A perfect vulnerability scanner can specifically find SQL injection vulnerabilities on the site, the latest vulnerability scanner can find newly vulnerabilities. Army should invest in some professional vulnerability scanning tools, such as the famous Acunetix Web Vulnerability Scanner procedures.

4.8 *Implemented code Security checks*

At all stages of Web application development process, there must implement code security check. First, we must implement safety testing before deploying Web applications, the significance of such measures is greater than ever, more far-reaching. Then it should be tested on the site after deployment with vulnerability scanning tools and site monitoring tools. If you use a third-party software, make sure it follows the best way, focus on program test, developing strength and software upgrade capability.

5 CONCLUSION

SQL injection attacks is use of development process of the application, it is not tight, to prevent such attacks, one website builder have sufficient knowledge for SQL injection attacks and improve programming; on the other hand, ordinary users can also use automated detection tool for SQL injection attacks, take the right measures, protect common data and network security.

REFERENCES

[1] Andrews M., Whittaker JA, Wang Qingqing, “translation, security testing and countermeasure of Web intrusion”, Beijing: Tsinghua University Press, 2006.

[2] Zhang Bo, “SQL injection attack and detection technology research”. Information Security and Communications Privacy .2010, (5).

[3] Gao Ming, “injection attack and defense strategy analysis”. Computer Knowledge and Technology .2013, (2)

[4] MiaoLong, YeMao, Wang Guanhua.” research and practice of SQL injection attacks and Web security technology”. Computer Applications. 2009, (1).