UNIVERSITY OF RWANDA

SCHOOL OF ICT

DEPARTMENT: INFORMATIONSYSTEMS

COMPUTER AND NETWORK SECURITY

ASSIGNMENT #1

GROUP MEMBERS:

NAMES	REG NUMBER
1.NShimiyimana Hirwa Corso	221021228
2.Kemirembe Mellon	221008606
3.TUYISHIME Polycarpe	221023444
4.Umubyeyi Ange Melisse	221002720
5.Mukamisha Ange Sandrine	221021153
6.Uwihirwe Benitha	221005385
7.Ishimwe Tresor	221003345
8.BISANGWA Hamouran	220011301
9.MUTUYIMANA Divine	221010778
10.MUTUYIMANA Dina	221006828
11.Nshimiyimana Zachee	221012287
12.SHEMA Yussuf	221007610
13.NTWALI Billy Christian	220020020
14.UMUHIRE Christine	221025598
15.Furaha Ernestine	221009524
16.KYASIMIRE KAITESI JOAN	221010122
17.Iraguha Rodrigues	221011227
18.Mugisha Erme	221008395
19.Dushimiyimana Elisa	221023304
20.NKUNDUMUKIZA Albert	221008117

SQL injection (SQLi)

What is a SQL injection (SQLi)?

A **SQL injection** (SQLi) is a technique that attackers use to gain unauthorized access to a web application database by adding a string of malicious code to a database query.

A SQL injection **manipulates** Structured Query Language code to provide access to protected resources, such as sensitive data, or execute malicious SQL statements. When executed correctly, a SQL injection can expose intellectual property, customer data or the administrative credentials of a private business.

SQL injection attacks can be used to target any application that uses a SQL database, with websites being the most common prey. Common SQL databases include MySQL, Oracle and Microsoft SQL Server.

How does a SQL injection attack work?

The first step of a SQL injection attack is to study how the targeted database functions. This is done by submitting a variety of random values into the query to observe how the server responds.

Attackers then use what they've learned about the database to craft a query the server interprets and then executes as a SQL command. For example, a database may store information about customers who have made a purchase with customer ID numbers. Instead of searching for a specific customer ID, an attacker may insert "CustomerID = 1000 OR 1=1" into the input field. Since the statement "1=1" is always true, the SQL query would return all available customer IDs and any corresponding data. This enables the attacker to circumvent authentication and gain administrator-level access.

In addition to returning unauthorized information, SQL attacks can be written to delete an entire database, bypass the need for credentials, remove records or add unwanted data.

Types of SQL injection attacks.

In-band SQLi:

Also known as classic SQLi, in-band SQLi is when hackers use the same channel -- or band -- to launch database errors and to collect the results from an attack. In-band SQLi is most commonly achieved through two methods:

Error-based injection techniques force the database to produce error messages that reveal information about the structure of the database.

Union-based attacks use prepared statements that exploit the SQL union function, which combines the results of multiple queries into one result.

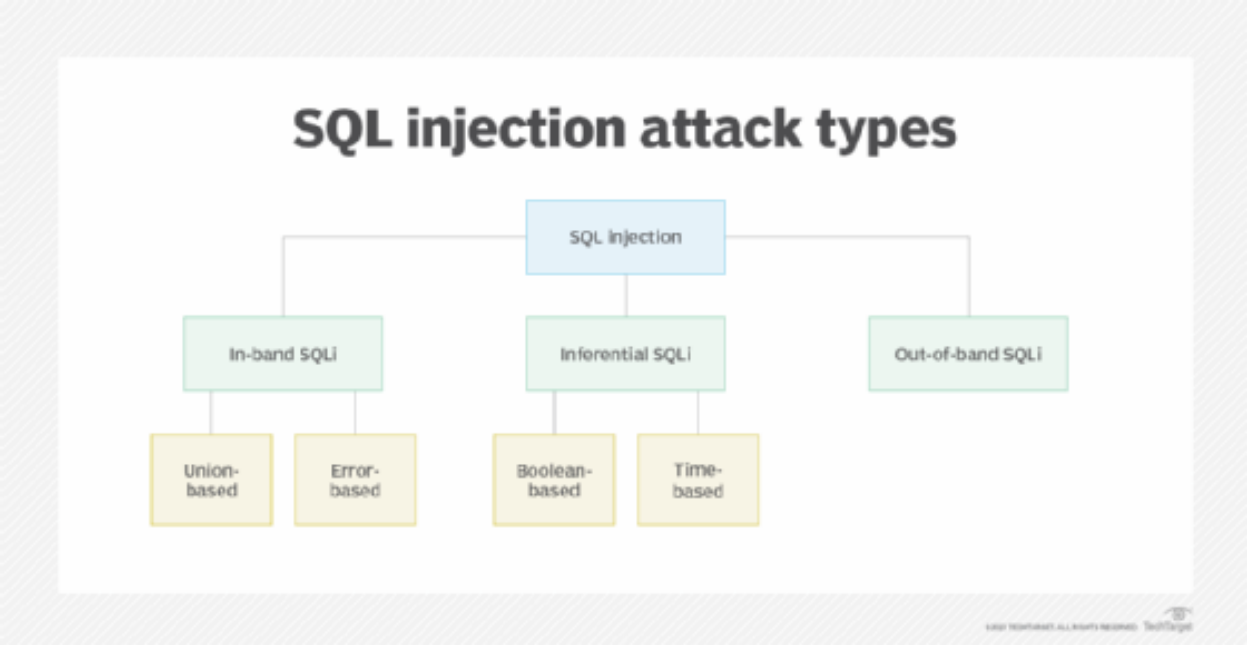
Inferential SQLi:

Also known as blind SQLi, inferential SQLi is when hackers send data payloads to a database server to observe its response and behaviour without being able to see what is occurring within the database. The server's response provides attackers with clues that they can use to adjust their attack strategy.

Inferential SQLi can be either **Boolean-based** or **time-based**. Boolean SQLi uses true or false statements to solicit a response, while time-based SQLi sets a designated response period.

Out-of-band SQLi:

Out-of-band SQLi is when hackers take advantage of domain name system or Hypertext Transfer Protocol requests to retrieve data. Out-of-band SQLi is usually only performed when a web server is too slow or when in-band SQLi is not possible to execute.



Examples of SQL injection attacks

There are numerous SQL injection vulnerabilities, attacks and strategies that can occur in a variety of settings.

The following are some common SQL injection examples:

- **Changing SQL query.** To retrieve hidden data, a SQL query can be changed by cybercriminals to reveal additional information.
- **Login bypass.** To get around authentication and access the program or website, a hacker can inject a SQL command into a login form.
- **Undermining application logic.** This is where a cybercriminal alters a query to obstruct the logic of the application.
- **Union attacks.** These attacks enable the retrieval of data from many database tables.
- **Database analysis.** This is where the cybercriminals analyze the database to glean information about its version and structure.
- **Blind SQL injection.** During a blind SQL injection, no error messages are received from the database.
- **Distributed denial of service (DDoS) attacks.** During this attack, an attacker injects a SQL statement to generate a DoS or DDos attack, overwhelming a system.

What if I told you that we can automate everything!

Using **Jsql Injection** tool we can automate testing if a webpage is vulnerable and launch an SQL injection attack too.

What is jsql injection?

jSQL Injection is a lightweight application used to find database information from a server.

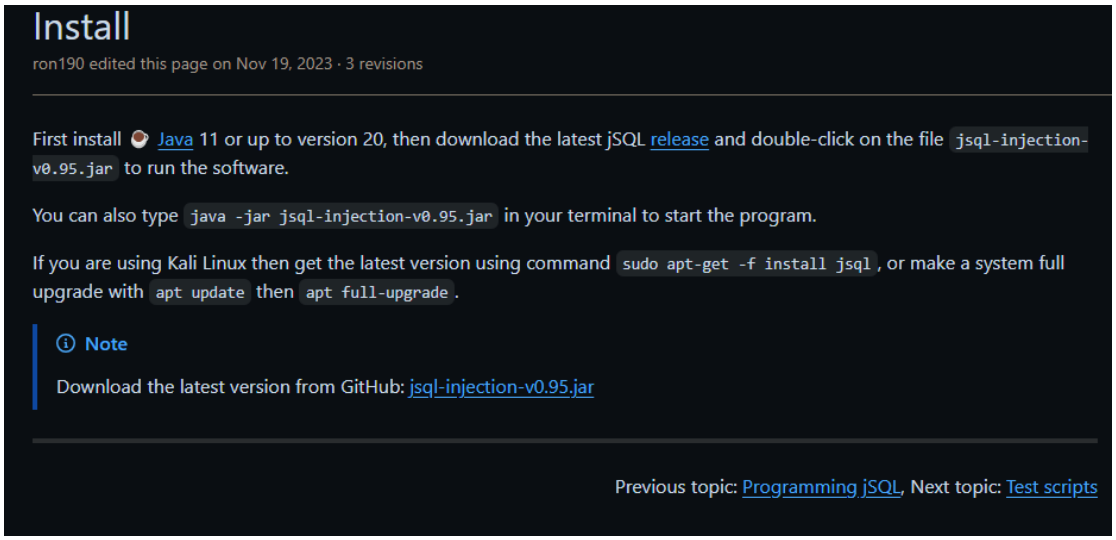
It's **free**, **open source** and **cross-platform** for Windows, Linux and Mac and it works with Java from version 11 to 20.

jSQL Injection is also part of the official penetration testing distribution Kali Linux and is included in various other distributions like Blackbuntu, Pentest Box, Parrot Security OS, ArchStrike and BlackArch Linux.

How to install:

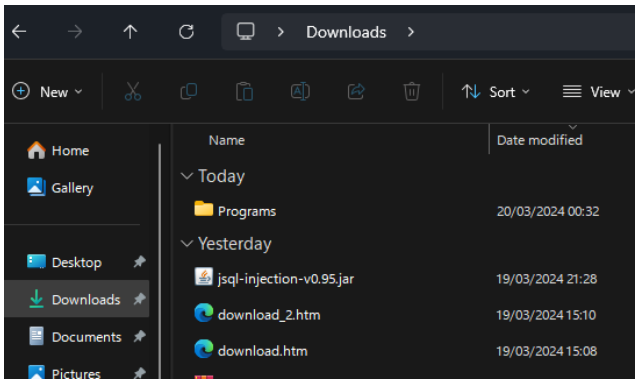
--Download and install **Java**: Ensure that Java is installed on your computer. JSQL Injection requires at least Java 11 to run. You can download Java from the official Oracle website.

--Download JSQL Injection: Download the latest release of JSQL Injection from the official GitHub repository at https://github.com/ron190/jsql-injection/wiki/Install and follow instruction on the page.



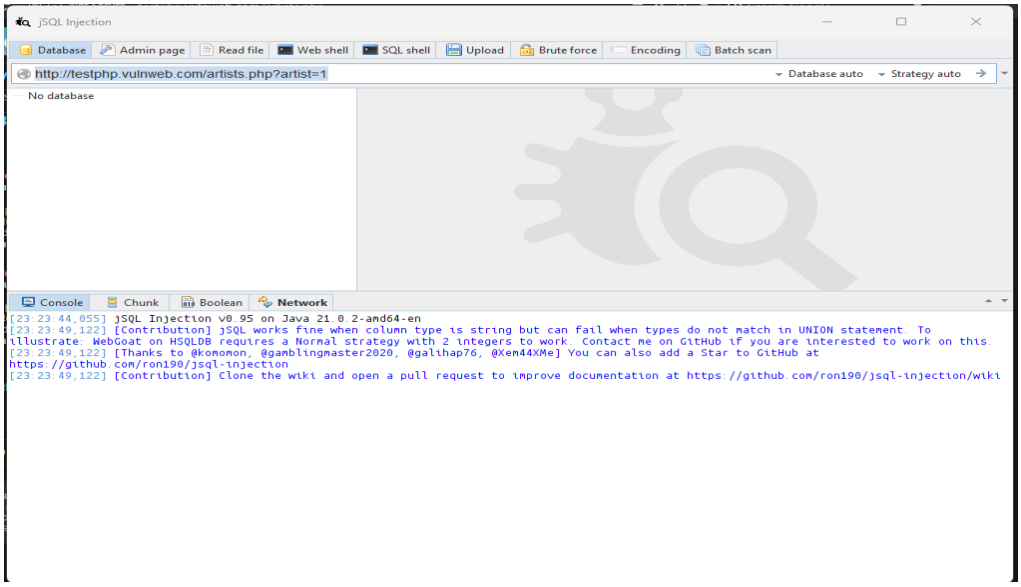
How to use:

-- go to where u saved the tool in your directory and open it.



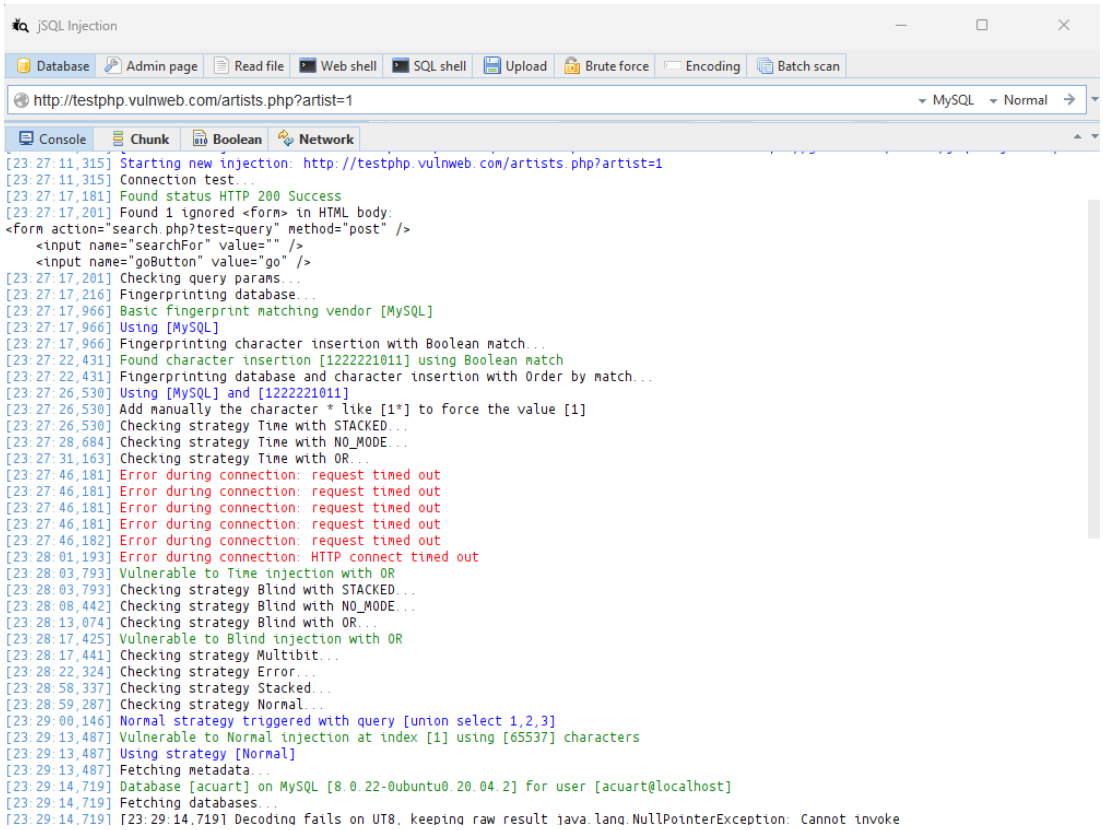
-- we now need to choose which website to attack. We will be using http://testphp.vulnweb.com/artists.php?artist=1 from this site http://www.vulnweb.com/ that offers free sites to hack that were created with vulnerabilities. We can't just hack any site without permission cause it's unethical and illegal.

-On our app interface we will copy our website link and click the -> arrow on the far right side

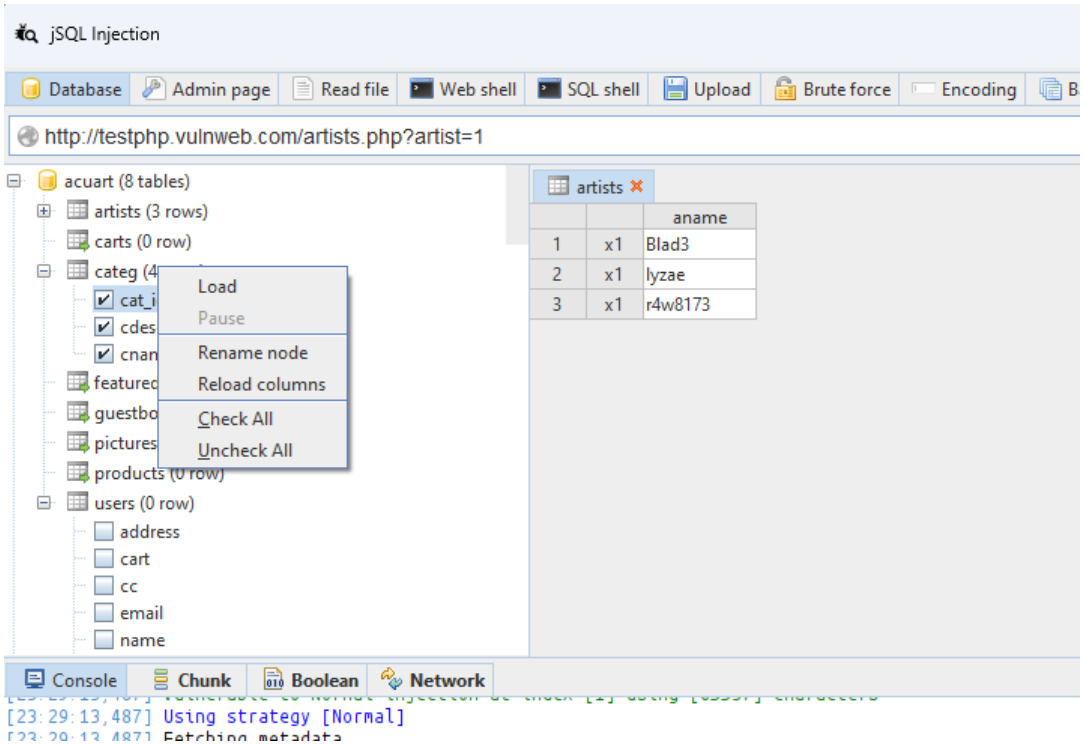


-wait for it to end and after this it what you should see

Picture 1:



Picture 2:



Break down of the results:

In picture 1:

-the tool does fingerprinting of the site and identifies that it's running MySQL engine in the backend

-then it proceeds to carry on sql injections like blind injections, error based and a few others. It also signals if the site is indeed vulnerable to those attacks.

In picture 2:

-it fetches all the databases with all the information in them.

How can a SQL injection attack be detected and prevented?

If a SQL injection attack is successfully carried out, it could cause extensive damage by exposing sensitive data and damaging customer trust. That's why it is important to detect this type of attack in a timely manner.

Web application firewalls (WAFs) are the most common tool used to filter out SQLi attacks. WAFs are based on a library of updated attack signatures and can be configured to flag malicious SQL queries in web applications.

To prevent a SQL injection attack from occurring, businesses can follow these practices:

Train employees on prevention methods. It's important that IT teams -- including DevOps, system administrators and software development -- receive proper security training to understand how SQLi attacks happen and how they can be prevented in web applications.

Don't trust user input. Any user input provided in a SQL query increases the likelihood for a successful SQL injection. The best way to mitigate this type of risk is to put security measures around user input.

Use an allowlist instead of a blocklist. Validating and filtering user input via an allowlist, as opposed to a blocklist, is recommended because cybercriminals can usually bypass a blocklist. This is because a blocklist includes a list of all the applications or executables that might pose a threat to the network. Therefore, everything on the network can operate besides the items on the blocklist. Unfortunately, thousands of new malware and virus samples are created every day, and it's impossible for administrators to keep the blocklists updated with newer attack variants and zero-day vulnerabilities, so a security breach is entirely possible before the list is updated.

Perform routing updates, and use the newest version of applications. One of the most common SQL injection vulnerabilities is outdated software. Not only is older technology unlikely to have built-in SQLi protection, but unpatched software is also often easier to manipulate. This includes programming languages, too. Older languages and syntax are more vulnerable. For example, use PHP Data Objects as a substitute for older MySQL.

Use validated prevention methods. Query strings written from scratch offer insufficient protection against SQLi. The best way to protect web applications is through input validation, prepared statements and parameterized queries.

Perform regular security scans. Regularly scanning web applications catches and remedies potential vulnerabilities before they do serious damage.

Some database administrators believe that a stored procedure statement can often aid in the prevention of SQL injection attacks by restricting the types of statements that can be supplied to its parameters. However, this doesn't prevent all exploits, as there are numerous workarounds and intriguing statements that can still be provided to stored procedures.