SQL INJE CTION

**Contents:**

1.Introduction

2.Attack Description

3.Flow of attack

4.Snort

**Introduction:**

SQL INJECTION is a code injection technique used to attack data driven applications, in which nefarious SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker). SQL injection must exploit a security vulnerability in an application's software, for example, when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly and unexpectedly executed. SQL injection is mostly known as an attack vector for websites but can be used to attack any type of SQL database.

SQL injection attacks allow attackers to spoof identity, tamper with existing data, cause repudiation issues such as voiding transactions or changing balances, allow the complete disclosure of all data on the system, destroy the data or make it otherwise unavailable, and become administrators of the database server.

In a 2012 study, it was observed that the average web application received 4 attack campaigns per month, and retailers received twice as many attacks as other industries.

**Attack description:**

Modern websites include login pages, search pages, support and product request forms, shopping carts, feedback forms and so on.

These website features are all vulnerable to SQL injection attacks due to the availability of user-input fields. An attacker can easily execute arbitrary SQL statements if these websites are prone to SQL injection. This may compromise the databases’ integrity and can expose sensitive data.

Based on the back-end database used, SQL injection vulnerabilities can result in varying levels of injection attacks. Attackers may manipulate existing queries, use sub selects, or add additional queries. In some instances, it may be even possible to read in or write out to files. Also, the attackers may execute shell commands on the root operating system (OS).

Some SQL Servers like Microsoft SQL Server incorporate stored and extended procedures. If an SQL injection attacker obtains access to these procedures, it can lead to highly undesirable outcomes. Improperly coded websites and web apps are always prone to this kind of attack.

The ideal way to avoid injection attacks is by detecting the vulnerabilities of websites and web apps before going live. There are automated SQL injection scanners which help the penetration testers verify the vulnerability of websites and web apps for potential SQL injection attacks.

This helps the web admin to instantly fix the vulnerable code and protect the websites from any potential SQL injection attacks.

**Flow of attack:**

1. Deploy SQLinjection Application in Apache Tomcat server and run it.
2. Run the given Java code
3. Compile: javac Sqlinjection.java
4. Run: java Sqlinjection

**Snort:**

I have filtered out the incoming packets intended for SQL injection using content of the packets if they contain any SQL like statements or any other statements which can result in tautological expressions to obtain all the information.

1.alert tcp $EXTERNAL\_NET any -> $HTTP\_SERVERS $HTTP\_PORTS (msg:"SQL Injection - Paranoid"; flow:to\_server,established;uricontent:".pl";pcre:"/(\%27)|(\')|(\-\-)|(%23)|(#)/i"; classtype:Web-application-attack; sid:9099; rev:5;)

2. reject tcp any any -> $HOME\_NET 8080 (msg: "SQL Injection Attempt - and 1=1"; content:"and"; pcre: "/[0-9]{1,10}=[0-9]{1,10}/iU"; classtype:web-application-attack; sid:1000006; rev:1;)

**Messages:**

The following example is an overly verbose error message

com.mysql.jdbc.exceptions.**MySQL**SyntaxErrorException: Table '**sqlInjectionTest**.test' doesn't exist

at com.mysql.jdbc.SQLError.createSQLException(SQLError.**java**:936)

at com.mysql.jdbc.MysqlIO.checkErrorPacket(MysqlIO.java:2985)

at com.mysql.jdbc.MysqlIO.sendCommand(MysqlIO.java:1631)

at com.mysql.jdbc.MysqlIO.sqlQueryDirect(MysqlIO.java:1723)

at com.mysql.jdbc.Connection.execSQL(Connection.java:3277)

at com.mysql.jdbc.Connection.execSQL(Connection.java:3206)

at com.mysql.jdbc.Statement.executeQuery(Statement.java:1232)

at sqlInjectionBefore.main(before.java:28)

This error message discloses that the application is using the Java programming language and the MySQL database platform and that the queried database is named "sqlInjectionTest." Each piece of information could assist attackers in crafting their application input, which increases the odds that they will be successful

**References:**

1. [www.techopedia.com](http://www.techopedia.com)
2. https://en.wikipedia.org/wiki/SQL\_injection