U.S. DEPARTMENT OF ENERGY'S
CYBERFORCE® PROGRAM

CyberForce® 101

SQL Injection

October 2023

cyberforcecompetition@anl.gov

SQL Injections 101

Databases

To understand SQL Injections, it is important to know how databases work. They are a centralized location to store all types of information. This includes names, dates of birth, prices, phone numbers and any other types of data.

The standard format of a database is a series of tables with multiple rows and columns that make it easy to read/understand the data. These tables may also have relationships to each other.

Databases are usually managed by a management system such as MySQL. What system is used can vary by how the developer wants to access, store, and manipulate the data as well as budgetary restraints.

SQL

Structured Query Language (SQL) is used to manipulate and access databases. It is compatible with almost all databases. SQL usually follows a simple structure of:

```
SELECT blank FROM blank WHERE blank;
```

SELECT is the data that you want to be returned.

FROM is the location of the data.

WHERE is a condition that must be met for a data point to be selected.

Like other languages, SQL needs a semicolon at the end of the command to signify that it is over.

Example:

```
SELECT Name FROM table WHERE Name = “John”;
```

SQL Injection

A SQL injection is when SQL quires are strategically placed into a database, or a website that uses a database, to view/manipulate data that should not be available. Certain designs and management systems allow the end user to click on a command line, such as a search bar, and run SQL quires if properly formatted. If developers do not take proper precautions, anyone may be able to view and/or change the contents of the database.

It is important to understand that what is typed in a search bar is part of a SQL quire the website will run. For example, if a user is on a website and they search for a product, such as a shirt, then the website may run the command:

```
SELECT * FROM Products WHERE Product = “Shirt”;
```

As a result, if a user enters their own SQL quire, the management system may run it (this all depends on the system type and what vulnerabilities it has). So, if they know that Names is a popular column name and Employees is a popular database table, they may be able to get the names of all the employees by entering the quire:

```
Shirt”; SELECT Names FROM Employees
```

We need two quotations so the system does not think that the rest of the quire is part of what it should be searching for. The semicolon

signifies that the previous quire (the one the developers made) is over and the system should run the SQL injection.

This is the quire that the system will run:

```
SELECT * FROM Products WHERE Product = “Shirt”; SELECT Names FROM Employees;
```

The way to change the contents of the database is to use the UPDATE quire. For example, to change the name of a product in the database from "Shirt" to "Pants", a user could enter this quire into the search line:

```
Shirt"; UPDATE Products SET product = "Pants" WHERE Product = "Shirt"
```

The system will then run the quire:

```
SELECT * FROM Products WHERE Product = “Shirt”; UPDATE Products SET Product =
"Pants" WHERE Product = "Shirt";
```

One of the most popular SQL Injection attacks is a 1=1 attack. These have mostly been patched by now, but they make a good starting point for beginners.

The point of the attack is to enter in a statement that is always true to bypass the need for a password. This is done by entering a variegation of the command ‘OR 1=1’. The system reads this as:

```
SELECT * FROM Employees WHERE Password =”” OR 1=1 ‘’;
```

Since 1=1 is always true, the system will give the user access to the information protected by the password. However, these attacks rarely work anymore.

Another common SQL Injection is to DROP a table or database. This is the equivalent of deleting a table or database. The basic command is DROP table; or DROP DATABASE databaseName;

The quire in the search line to drop the table titled “Products” would be:

```
Shirt”; DROP Products
```

The website would then run the quire:

```
SELECT * FROM Products WHERE Product = “Shirt”; DROP Products;
```

When performing a SQL injection, it takes a lot of trial and error to figure out the name of columns and tables and where the information you want is stored. Some of this information can be found if you click on the website’s URL, but this is not always the case. It is important to remember that SQL is case sensitive, so it may be necessary to capitalize some characters.

Sources

- What is SQL Injection? Tutorial & Examples | Web Security Academy (portswigger.net)
- MySQL WHERE Clause (w3schools.com)
- https://www.hacksplaining.com/exercises/sql-injection