Computer System- B Security

Introduction to Web Security P4

SQL injection

Sanjay Rawat

SQL Injection Attack

SQL Injection Attack

- Many web applications take user input from a form

SQL Injection Attack

- Many web applications take user input from a form
- Often this user input is used literally in the construction of a SQL query submitted to a database. For example:

SQL Injection Attack

- Many web applications take user input from a form
- Often this user input is used literally in the construction of a SQL query submitted to a database. For example:
  - `SELECT user FROM table WHERE name = ‘user_input’;`

SQL Injection Attack

- Many web applications take user input from a form
- Often this user input is used literally in the construction of a SQL query submitted to a database. For example:
  - `SELECT user FROM table WHERE name = ‘user_input’;`
- An SQL injection attack involves placing SQL statements in the user input (again, data-code confusion!)

SQL: Standard Query Language

SQL lets you access and manage (Query) databases

A database is a large collection of data organized in tables for rapid search and retrieval, with row and columns

SQL: Standard Query Language

SQL lets you access and manage (Query) databases

A database is a large collection of data organized in tables for rapid search and retrieval, with row and columns

Table: CS166

First_Name	Last_Name	Code_ID
Bernardo	Palazzi	345
Roberto	Tamassia	122
Alex	Heitzman	543
.....	....	....

SQL: Standard Query Language

SQL lets you access and manage (Query) databases

A database is a large collection of data organized in tables for rapid search and retrieval, with row and columns

Table: CS166

A field or Column

First_Name	Last_Name	Code_ID
Bernardo	Palazzi	345
Roberto	Tamassia	122
Alex	Heitzman	543
....	....	....

SQL: Standard Query Language

SQL lets you access and manage (Query) databases

A database is a large collection of data organized in tables for rapid search and retrieval, with row and columns

Table: CS166

A field or Column

A Record or Row

First_Name	Last_Name	Code_ID
Bernardo	Palazzi	345
Roberto	Tamassia	122
Alex	Heitzman	543
.....	....	....

SQL Syntax

```
SELECT  column_name(s) or *
FROM table_name
WHERE column_name operator value
```

- SELECT statement is used to select data FROM one or more tables in a database
- WHERE clause is used to filter records

Result-set is stored in a result table

- ; is statement terminator and -- is remark beginning

Login Authentication Query

- Standard query to authenticate users:
  - select * from users where user='$usern' AND pwd='$password'
- Classic SQL injection attacks
  - Server side code sets variables $username and $passwd from user input to web form
  - Variables passed to SQL query
    - select * from users where user='$username' AND pwd='$passwd'

Login Authentication Query

- Standard query to authenticate users:
  - select * from users where user='$usern' AND pwd='$password'
- Classic SQL injection attacks
  - Server side code sets variables $username and $passwd from user input to web form
  - Variables passed to SQL query
    - select * from users where user='$username' AND pwd='$passwd'
- Special strings can be entered by attacker
    - select * from users where user='M' OR '1=1' AND pwd='M' OR '1=1'
- Result: access obtained without password

Some improvements …

Some improvements …

- Query modify:
- `select user, pwd from users where user='$usern‘`

Some improvements …

- Query modify:
- `select user, pwd from users where user='$usern‘`
- **$usern**=“M' OR '1=1”;

Some improvements …

- Query modify:
- `select user, pwd from users where user='$usern‘`
- **$usern**=“M' OR '1=1”;
- Result: the entire table
- We can check:
  - only one tuple result
  - formal correctness of the result

Some improvements …

- Query modify:
- `select user, pwd from users where user='$usern‘`
- **$usern**=“M' OR '1=1”;
- Result: the entire table
- We can check:
  - only one tuple result
  - formal correctness of the result
- **$usern**=“M' ; drop table user;”?

Correct Solution

Correct Solution

- We can use an Escape method, where all “malicious” characters will be changed:

Correct Solution

- We can use an Escape method, where all “malicious” characters will be changed:
- Escape(“t ' c”) gives as a result “t \' c”
- `select user, pwd from users where user='$usern'`
- `$usern=escape(“M' ;drop table user;”)`

Correct Solution

- We can use an Escape method, where all “malicious” characters will be changed:
- Escape(“t ' c”) gives as a result “t \' c”
  - `select user, pwd from users where user='$usern'`
  - `$usern=escape(“M' ;drop table user;”)`
- The result is the safe query:
  - `select user,pwd from users where user='M\' drop table user;\''`