Introduction to SQL Injection

Paul Murley
UT Dallas CSG – 10/26/2016
HTTP/1.0 (Browser) → GET / HTTP/1.0 → Web server → HTTP/1.1 200 OK → Database server → index.php
<table>
<thead>
<tr>
<th>pnumber</th>
<th>pname</th>
<th>plocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Project X</td>
<td>Dallas</td>
</tr>
<tr>
<td>2</td>
<td>Project Y</td>
<td>Dallas</td>
</tr>
<tr>
<td>3</td>
<td>Project Z</td>
<td>Houston</td>
</tr>
<tr>
<td>4</td>
<td>Middleware</td>
<td>Austin</td>
</tr>
<tr>
<td>5</td>
<td>Lazer Printers</td>
<td>Dallas</td>
</tr>
</tbody>
</table>
SQL (Structured Query Language)

• Language used to query databases to do things like:

  • Read data using a specific filter – “SELECT”
  • Modify existing data – “UPDATE”
  • Add or delete rows or columns – “INSERT”
  • Create or delete entire tables – “CREATE”, “DROP”
SQL (Structured Query Language)

• Language used to query databases to do things like:

  • Read data using a specific filter – “SELECT”

  • Modify existing data – “UPDATE”

  • Add or delete rows or columns – “INSERT”

  • Create or delete entire tables – “CREATE”, “DROP”
SELECT * FROM Project WHERE pname = 'Project X';

<table>
<thead>
<tr>
<th>pnumber</th>
<th>pname</th>
<th>plocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Project X</td>
<td>Dallas</td>
</tr>
<tr>
<td>2</td>
<td>Project Y</td>
<td>Dallas</td>
</tr>
<tr>
<td>3</td>
<td>Project Z</td>
<td>Houston</td>
</tr>
<tr>
<td>4</td>
<td>Middleware</td>
<td>Austin</td>
</tr>
<tr>
<td>5</td>
<td>Lazer Printers</td>
<td>Dallas</td>
</tr>
</tbody>
</table>
### SQL Query

```sql
SELECT * FROM Users WHERE username = 'admin' AND password = '7E7274BAC45E467C5AB832170F12E418';
```

### Table

<table>
<thead>
<tr>
<th>id</th>
<th>username</th>
<th>password</th>
<th>passwordHint</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>admin</td>
<td>7E7274BAC45E467C5AB832170F12E418</td>
<td>k3wl dud</td>
</tr>
<tr>
<td>2</td>
<td>pumpkin22</td>
<td>5377DBF76D995CC213ED76924A31CB13</td>
<td>my favorite holiday</td>
</tr>
<tr>
<td>3</td>
<td>johndoe</td>
<td>512239D9AE0C3B5567DE188739F689F2</td>
<td>Freddie Mercury's band</td>
</tr>
<tr>
<td>4</td>
<td>alexa45</td>
<td>2FE5421E49061F8225C2FB7CB81980FD</td>
<td>password</td>
</tr>
<tr>
<td>5</td>
<td>guy</td>
<td>ABE35E2827DDA834C9612FE9E9C92CE0</td>
<td>NULL</td>
</tr>
<tr>
<td>6</td>
<td>maryjane</td>
<td>198670893B2781C83F3DA5D45150123D</td>
<td>I'm one!</td>
</tr>
<tr>
<td>7</td>
<td>dudson123</td>
<td>59E2113217E65B9885F9DA73FDC5697B</td>
<td>scary movie!</td>
</tr>
</tbody>
</table>
So What’s the Problem?
# Define POST variables
uname = request.POST['username']
passwd = request.POST['password']

# SQL query vulnerable to SQLi
sql = "SELECT id FROM users WHERE username='" + uname + "' AND password='" + passwd + "';"

# Execute the SQL statement
database.execute(sql)
# Define POST variables
uname = request.POST['username']
passwd = request.POST['password']

# SQL query vulnerable to SQLi
sql = "SELECT id FROM users WHERE username='" + uname + "' AND password='" + passwd + "';"

# Execute the SQL statement
database.execute(sql)
# Define POST variables
uname = request.POST['username']
passwd = request.POST['password']

# SQL query vulnerable to SQLi
sql = "SELECT id FROM users WHERE username='" + uname + "' AND password='" + passwd + "';"

# Execute the SQL statement
database.execute(sql)
# Define POST variables
uname = request.POST['username']
passwd = request.POST['password']

# SQL query vulnerable to SQLi
sql = "SELECT id FROM users WHERE username='" + uname + "' AND password='" + passwd + "';"

# Execute the SQL statement
database.execute(sql)
# Define POST variables
uname = request.POST['username']
passwd = request.POST['password']

# SQL query vulnerable to SQLi
sql = "SELECT id FROM users WHERE username='" + uname + "' AND password='" + passwd + "';"

# Execute the SQL statement
database.execute(sql)
# Define POST variables
uname = "admin"
passwd = request.POST["password"]

# SQL query vulnerable to SQLi
sql = "SELECT id FROM users WHERE username=" + uname + "' AND password='" + passwd + ";"

# Execute the SQL statement
database.execute(sql)
# Define POST variables
uname = "admin"
passwd = "pass123"

# SQL query vulnerable to SQLi
sql = "SELECT id FROM users WHERE username='" + uname + "' AND password='" + passwd + ";"

# Execute the SQL statement
database.execute(sql)
# Define POST variables
uname = "' OR 1=1; --"
passwd = "pass123"

# SQL query vulnerable to SQLi
sql = "SELECT id FROM users WHERE username='' + uname + '' AND password='' + passwd + '';

# Execute the SQL statement
database.execute(sql)
# Define POST variables
uname = "' OR 1=1; --"
passwd = "pass123"

# SQL query vulnerable to SQLi
sql = "SELECT id FROM users WHERE username='" + "' OR 1=1; --" + "' AND password='" + passwd + "'";

# Execute the SQL statement
database.execute(sql)
# Define POST variables
uname = "' OR 1=1; --"
passwd = "pass123"

# SQL query vulnerable to SQLi
sql = "SELECT id FROM users WHERE username='" + 
"' OR 1=1; --" + "' AND password='" + "pass123" + 
"'";

# Execute the SQL statement
database.execute(sql)
# Define POST variables
uname = "' OR 1=1; --"
passwd = "pass123"

# SQL query vulnerable to SQLi
sql = "SELECT id FROM users WHERE username=' OR 1=1; -- ' AND password='pass123';"

# Execute the SQL statement
database.execute(sql)
# Define POST variables
uname = "' OR 1=1; --"
passwd = "pass123"

# SQL query vulnerable to SQLi
sql = "SELECT id FROM users WHERE username=' OR 1=1;"

# Execute the SQL statement
database.execute(sql)
HI, THIS IS YOUR SON’S SCHOOL. WE’RE HAVING SOME COMPUTER TROUBLE.

OH, DEAR - DID HE BREAK SOMETHING? IN A WAY -

DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?

OH, YES. LITTLE BOBBY TABLES, WE CALL HIM.

WELL, WE’VE LOST THIS YEAR’S STUDENT RECORDS. I HOPE YOU’RE HAPPY.

AND I HOPE YOU’VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

http://www.codebashing.com/sql-demo