Ethical Hacking Lab
Module 6: SQL Injection and Session Hijacking

Assignment 6: Use software tools and commands to perform SQL injection and session hijacking, and generate an analysis report.

1. SQL injection for website hacking.

Aim: To exploit vulnerabilities in web applications by injecting malicious SQL code into input fields

Theory:

SQL injection (SQLi) is a type of security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. It occurs when user input is improperly sanitized, enabling the attacker to execute arbitrary SQL code.

How It Works:

SQL injection takes advantage of the way SQL queries are constructed. If an application concatenates user input directly into a SQL statement without proper validation, an attacker can manipulate the query. For instance, a typical login form might use a query like:

```
SELECT * FROM users WHERE username = 'user_input' AND password = 'pass_input';
```

If an attacker inputs something like ' OR '1'='1, the query becomes:

```
SELECT * FROM users WHERE username = '' OR '1'='1' AND password = 'pass_input';
```

This query will always return true, allowing the attacker to bypass authentication.

Ways to Overcome SQL Injection:

- The best way to prevent SQL injection is to use safe programming practices that make SQL injection impossible, such as parameterized queries (prepared statements) and stored procedures.
- Implement CAPTCHA to distinguish between human users and automated scripts.
- Use One-Time Passwords (OTP) for additional security during authentication.

SQL Query:

```
# mysql -u root
MariaDB [(none)]> CREATE DATABASE stud_users;
MariaDB [(none)]> SHOW DATABASES;
```

```
MariaDB [(none)]> USE stud_users;

MariaDB [stud_users]> CREATE TABLE login_users (
        id INT AUTO_INCREMENT PRIMARY KEY,
        name VARCHAR(100),
        user_name VARCHAR(50) UNIQUE NOT NULL,
        password VARCHAR(255) NOT NULL
);

MariaDB [stud_users]> INSERT INTO login_users (name, user_name, password) VALUES
('Alice Smith', 'alice', 'alicepassword'),
('Bob Johnson', 'bob', 'bobpassword'),
('Charlie Brown', 'charlie', 'charliepassword');
```

Program Code:

> register.php

```
<?php
session_start();
$conn = mysqli_connect('localhost', 'root', '', 'stud_users');

if ($_SERVER["REQUEST_METHOD"] == "POST") {
   $name = $_POST['name'];
   $user_name = $_POST['username'];
   $password = $_POST['password']; // Store password as plain text (vulnerable)

   // Vulnerable SQL query (for educational purposes)
      $query = "INSERT INTO login_users (name, user_name, password) VALUES ('$name',
'$user_name', '$password')";
   mysqli_query($conn, $query);

   echo "<p>Registration successful! You can now <a href='login.php'>login</a>.</p>";
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Register</title>
   <style>
      body { font-family: Arial, sans-serif; background-color: #f4f4f4; margin: 0; padding: 0; }
         .container { max-width: 400px; margin: 50px auto; background: white; padding: 20px;
border-radius: 5px; box-shadow: 0 0 10px rgba(0,0,0,0.1); }
      h2 { text-align: center; }
       input[type="text"], input[type="password"] { width: 100%; padding: 12px; margin: 10px 0;
border: 1px solid #ccc; border-radius: 5px; box-sizing: border-box; } /* Increased padding */
```

```
        input[type="submit"], input[type="reset"] { background-color: #28a745; color: white; border: none; padding: 10px; border-radius: 5px; cursor: pointer; margin: 5px 0; }
        input[type="submit"]:hover, input[type="reset"]:hover { background-color: #218838; }
        .link { text-align: center; margin-top: 10px; }
    </style>
</head>
<body>

<div class="container">
    <h2>Register</h2>
    <form method="POST" action="">
        Name: <input type="text" name="name" required><br>
        Username: <input type="text" name="username" required><br>
        Password: <input type="password" name="password" required><br>
        <input type="submit" value="Register">
        <input type="reset" value="Reset">
    </form>
    <div class="link">
        Already have an account? <a href="login.php">Login here</a>
    </div>
</div>

</body>
</html>
```

> login.php

```
<?php
session_start();
$conn = mysqli_connect('localhost', 'root', '', 'stud_users');

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST['username'];
    $password = $_POST['password']; // Accepting password as plain text (vulnerable)

    // Modify the SQL query to handle cases where username may be empty
        $query = "SELECT * FROM login_users WHERE ('$username' = '' OR user_name = '$username') AND ('$password' = '' OR password = '$password')";

    // Debugging SQL query
    echo "SQL Query: " . $query . "<br>";

    $result = mysqli_query($conn, $query);
```

```
    // Check for query execution errors
    if (!$result) {
        die("Query failed: " . mysqli_error($conn));
    }

    if (mysqli_num_rows($result) > 0) {
        $_SESSION['user'] = $username; // Set session
        header("Location: welcome.php");
        exit();
    } else {
        echo "<p>Invalid login</p>";
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <style>
        body { font-family: Arial, sans-serif; background-color: #f4f4f4; margin: 0; padding: 0; }
            .container { max-width: 400px; margin: 50px auto; background: white; padding: 20px;
border-radius: 5px; box-shadow: 0 0 10px rgba(0,0,0,0.1); }
        h2 { text-align: center; }
         input[type="text"], input[type="password"] { width: 100%; padding: 12px; margin: 10px 0;
border: 1px solid #ccc; border-radius: 5px; box-sizing: border-box; } /* Increased padding */
              input[type="submit"], input[type="reset"] { background-color: #007bff; color: white;
border: none; padding: 10px; border-radius: 5px; cursor: pointer; margin: 5px 0; }
        input[type="submit"]:hover, input[type="reset"]:hover { background-color: #0056b3; }
        .link { text-align: center; margin-top: 10px; }
    </style>
</head>
<body>

<div class="container">
    <h2>Login</h2>
    <form method="POST" action="">
        Username: <input type="text" name="username"><br>
        Password: <input type="password" name="password"><br>
        <input type="submit" value="Login">
        <input type="reset" value="Reset">
    </form>
    <div class="link">
```

```
    Don't have an account? <a href="register.php">Register here</a>
  </div>
</div>

</body>
</html>
```

> welcome.php

```
<?php
session_start();
if (!isset($_SESSION['user'])) {
  header("Location: login.php");
  exit();
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Welcome</title>
  <style>
    body { font-family: Arial, sans-serif; background-color: #f4f4f4; margin: 0; padding: 0; }
      .container { max-width: 400px; margin: 50px auto; background: white; padding: 20px;
border-radius: 5px; box-shadow: 0 0 10px rgba(0,0,0,0.1); text-align: center; }
    h2 { margin: 0; }
       a { display: inline-block; margin-top: 20px; background-color: #007bff; color: white;
padding: 10px 15px; text-decoration: none; border-radius: 5px; }
    a:hover { background-color: #0056b3; }
  </style>
</head>
<body>

<div class="container">
  <h2>Welcome, <?php echo htmlspecialchars($_SESSION['user']); ?>!</h2>
  <p>You are logged in.</p>
  <a href="logout.php">Logout</a>
</div>

</body>
</html>
```

> logout.php

```
<?php
session_start();
session_unset();
```

```
session_destroy();
header("Location: login.php");
exit();
?>
```

Steps to be followed:

1. Create Database:
   - Create a database named stud_users.
2. Create Table:
   - Create a table named login_users with columns for id, name, user_name, and password.
3. Insert Data:
   - Insert sample user data into the login_users table.
4. Create Login Page:
   - Create a login.php page for user authentication.
5. Create Dashboard:
   - Create an index.php page to serve as the dashboard after successful login.
6. Create Logout Page:
   - Create a logout.php page to handle user logout.
7. Test SQL Injection:
   - On the login page, test the following input:
     - Username: ' OR '1'='1
     - Password: blank (or any valid password from the database).

Output:

```
Administrator: XAMPP for Windows - mysql  -u root

Setting environment for using XAMPP for Windows.
itsak@DESKTOP-J910NPG c:\xampp
# mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.4.28-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

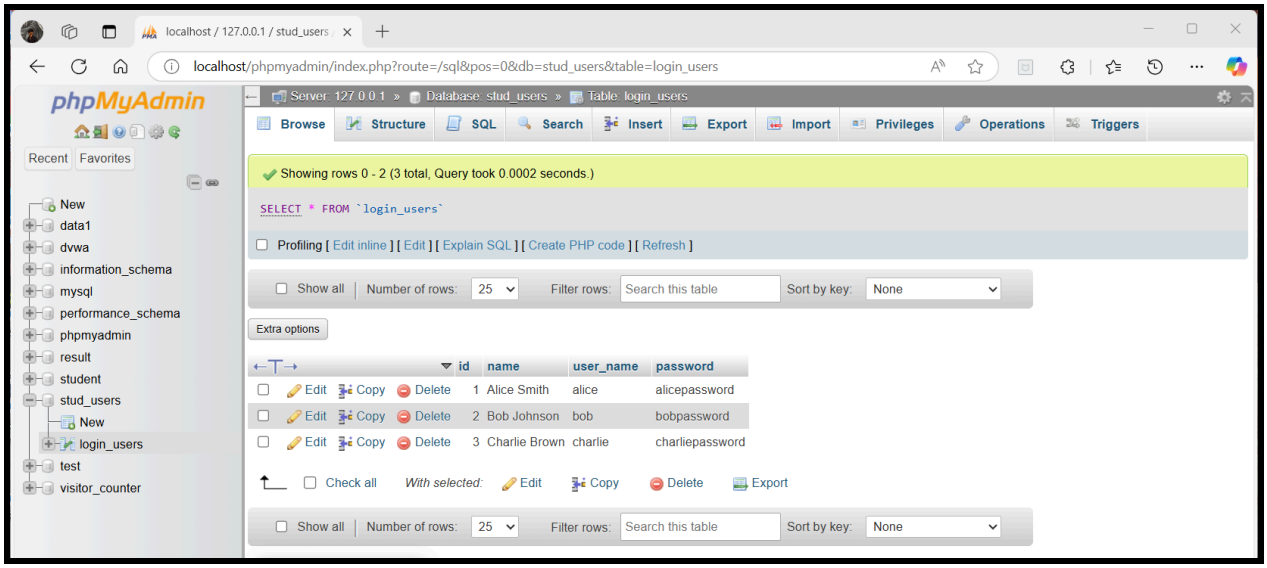
```
MariaDB [(none)]> CREATE DATABASE stud_users;
Query OK, 1 row affected (0.002 sec)
```

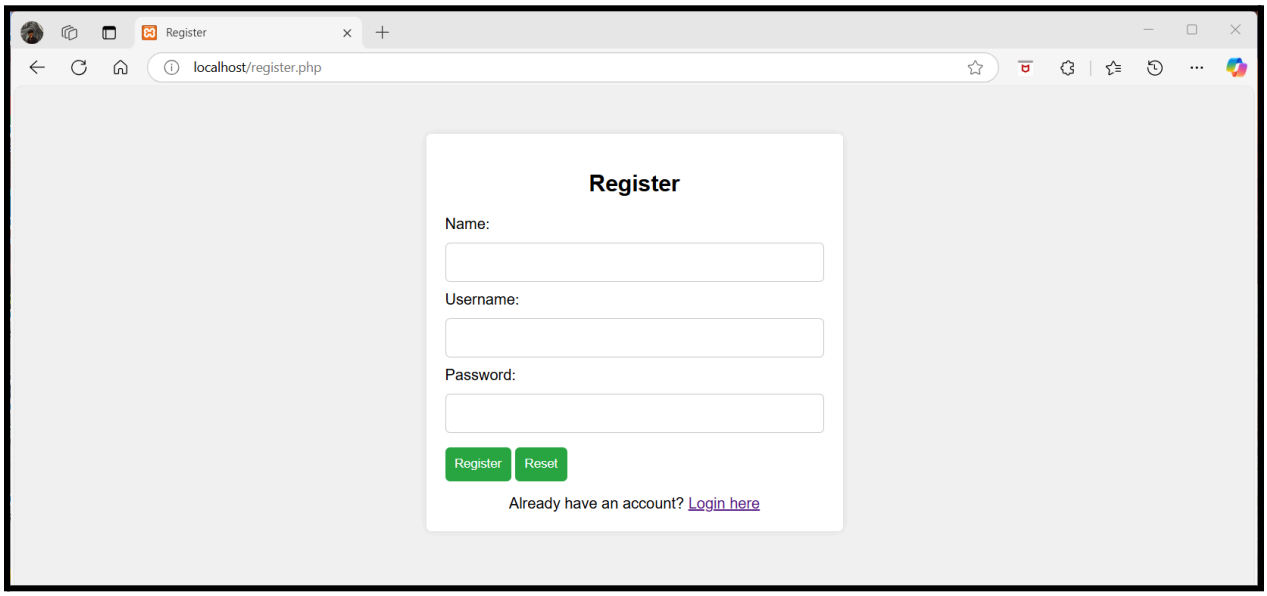
```
MariaDB [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| data1              |
| dvwa               |
| information_schema |
| mysql              |
| performance_schema |
| phpmyadmin         |
| result             |
| stud_users         |
| student            |
| test               |
| visitor_counter    |
+--------------------+
11 rows in set (0.006 sec)
```

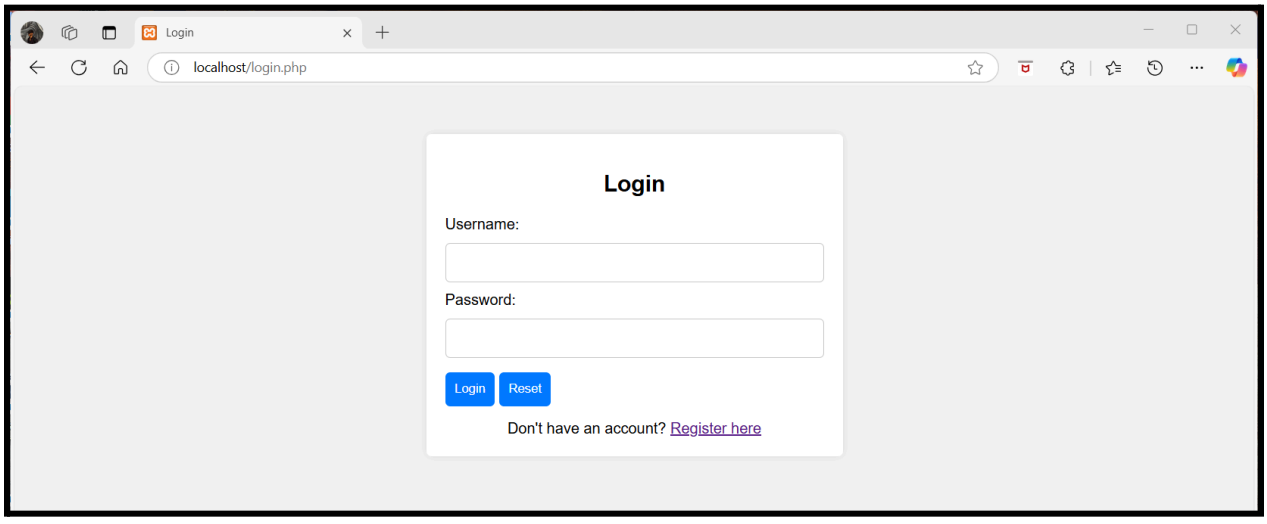
```
MariaDB [(none)]> USE stud_users;
Database changed
```

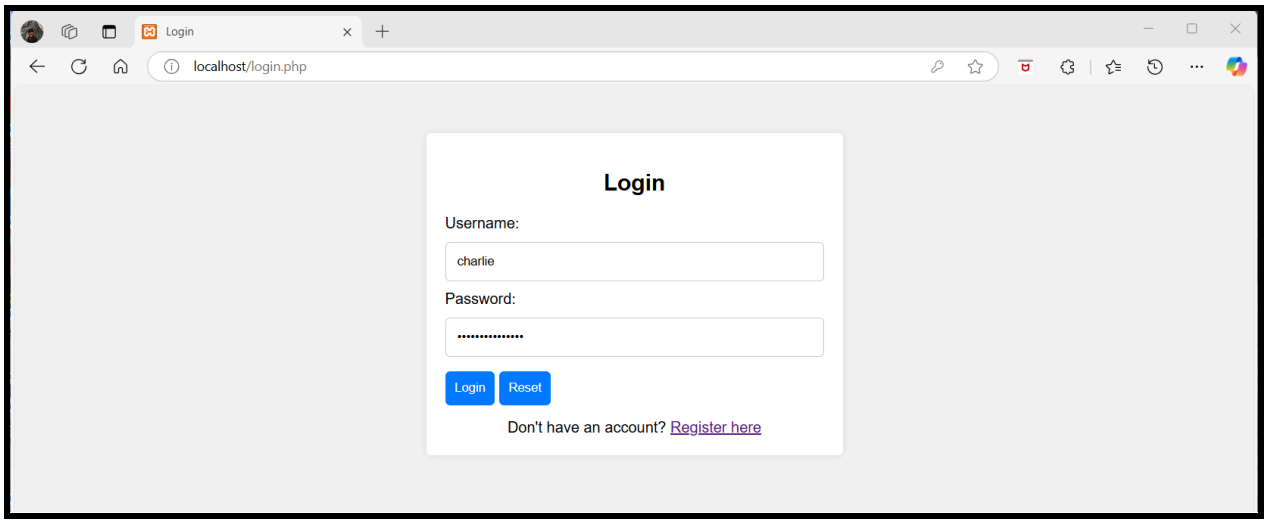
```
MariaDB [stud_users]> CREATE TABLE login_users (
    -> id INT AUTO_INCREMENT PRIMARY KEY,
    -> name VARCHAR(100),
    -> user_name VARCHAR(50) UNIQUE NOT NULL,
    -> password VARCHAR(255) NOT NULL
    -> );
Query OK, 0 rows affected (0.015 sec)
```

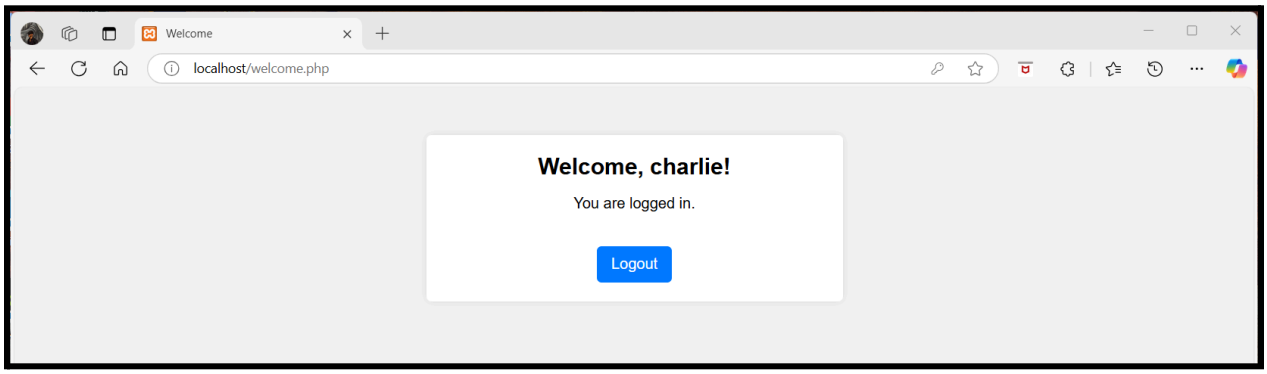
```
MariaDB [stud_users]> INSERT INTO login_users (name, user_name, password) VALUES
    -> ('Alice Smith', 'alice', 'alicepassword'),
    -> ('Bob Johnson', 'bob', 'bobpassword'),
    -> ('Charlie Brown', 'charlie', 'charliepassword');
Query OK, 3 rows affected (0.007 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

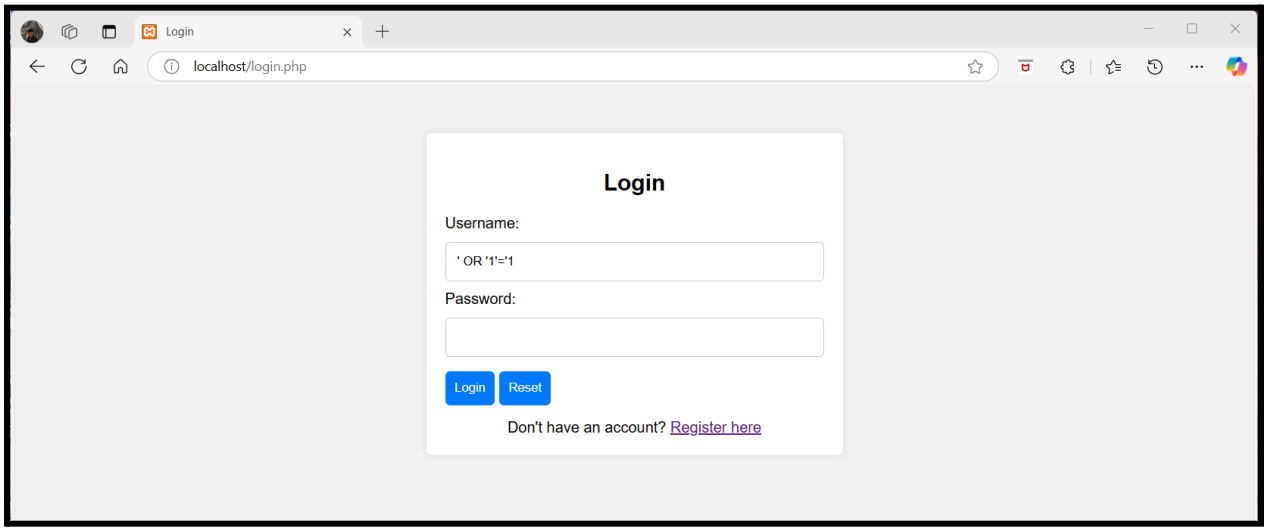

localhost / 127.0.0.1 / stud_users
localhost/phpmyadmin/index.php?route=/sql&pos=0&db=stud_users&table=login_users
phpMyAdmin
Recent Favorites
New
data1
dvwa
information_schema
mysql
performance_schema
phpmyadmin
result
student
stud_users
New
login_users
test
visitor_counter
Server: 127.0.0.1 » Database: stud_users » Table: login_users
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Showing rows 0 - 2 (3 total, Query took 0.0002 seconds.)
SELECT * FROM `login_users`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
id name user_name password
Edit Copy Delete 1 Alice Smith alice alicepassword
Edit Copy Delete 2 Bob Johnson bob bobpassword
Edit Copy Delete 3 Charlie Brown charlie charliepassword
Check all With selected: Edit Copy Delete Export
Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

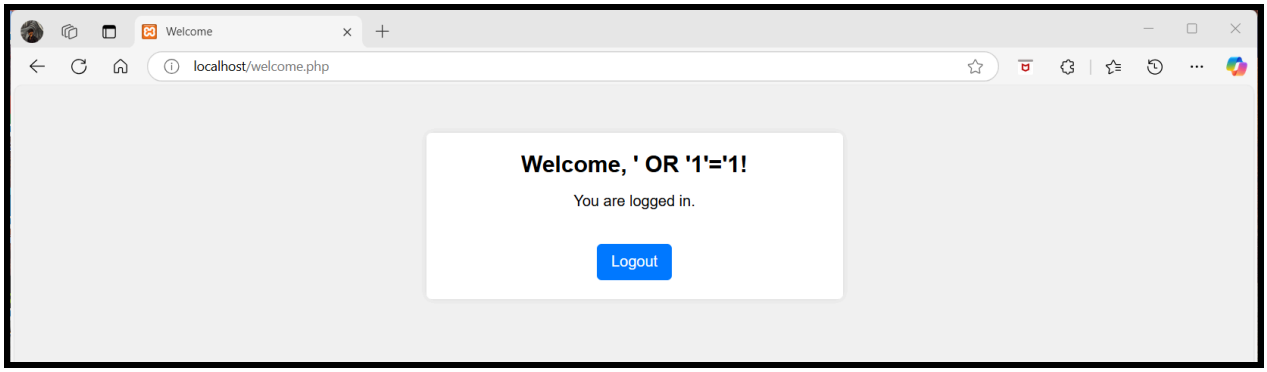

Register
localhost/register.php
Register
Name:
Username:
Password:
Register Reset
Already have an account? Login here


Login
localhost/login.php
Login
Username:
Password:
Login Reset
Don't have an account? Register here


Login
localhost/login.php
Login
Username:
charlie
Password:
Login
Reset
Don't have an account? Register here


Welcome
localhost/welcome.php
Welcome, charlie!
You are logged in.
Logout


Login
localhost/login.php
Login
Username:
' OR '1'='1
Password:
Login
Reset
Don't have an account? Register here


Welcome
localhost/welcome.php
Welcome, ' OR '1'='1!
You are logged in.
Logout

2. Session Hijacking

Aim: To exploit the web session control mechanism by stealing or predicting a valid computer session to gain unauthorized access to information or services in a computer system

Theory:

Session hijacking occurs when an attacker takes over a user's active session. This can happen through various methods, such as:

1. Session ID Theft: The attacker captures the session ID through methods like:
    - Packet Sniffing: Monitoring network traffic to capture session tokens.
    - Cross-Site Scripting (XSS): Injecting malicious scripts into web pages that can send session IDs to the attacker.

2. Session Fixation: The attacker sets a user's session ID to a known value, then tricks the user into logging in with that session ID. Once the user is authenticated, the attacker can use that session ID to access the user’s account.

3. Man-in-the-Middle (MitM) Attacks: The attacker intercepts communication between the user and the server, allowing them to capture session IDs and other sensitive information.

4. Browser Vulnerabilities: Exploiting vulnerabilities in the user's web browser can also lead to session hijacking.

Ways to Detect if Your Data Has Been Hacked by Packet Sniffers:

1. Use HTTPS: The secure version of HTTP, HTTPS, prevents packet sniffers from seeing the traffic on the websites you are visiting.
2. Check for HTTPS: To ensure you are using HTTPS, look for a lock icon in the upper left corner of your browser.
3. Use a Virtual Private Network (VPN): A VPN encrypts the traffic being sent between your computer and the destination. This includes information used on websites, services, and applications. A packet sniffer will only see encrypted data being sent through your VPN service provider.

To protect against session hijacking, several best practices can be employed:

- Secure Cookie Attributes: Use the HttpOnly and Secure flags on cookies to prevent access via JavaScript and ensure they are only sent over HTTPS.
- Use HTTPS: Encrypt data in transit to prevent interception.
- Implement Session Timeouts: Automatically log out users after a period of inactivity.

- Regenerate Session IDs: Change session IDs after login and periodically during a session to reduce the risk of fixation.
- User Education: Inform users about the dangers of phishing attacks and the importance of secure browsing practices.

Steps to be followed:

Step 1: Start Wireshark.

- Select the adapter for loopback traffic capture.
- Double-click on the selected adapter to begin capturing.

Step 2: Apply an HTTP filter in Wireshark.

Step 3: Launch the Chrome browser.

Step 4: Navigate to your login page.

Step 5: Log in using your credentials.

Step 6: Go back to Wireshark.

- Look for packets containing welcome.php.

Step 7: Identify the cookie in the captured packets.

- Copy the cookie value and note it down in a text editor or notepad.
- Alternatively, right-click on the relevant packet (highlighted in red) and select "Copy" > "Value."

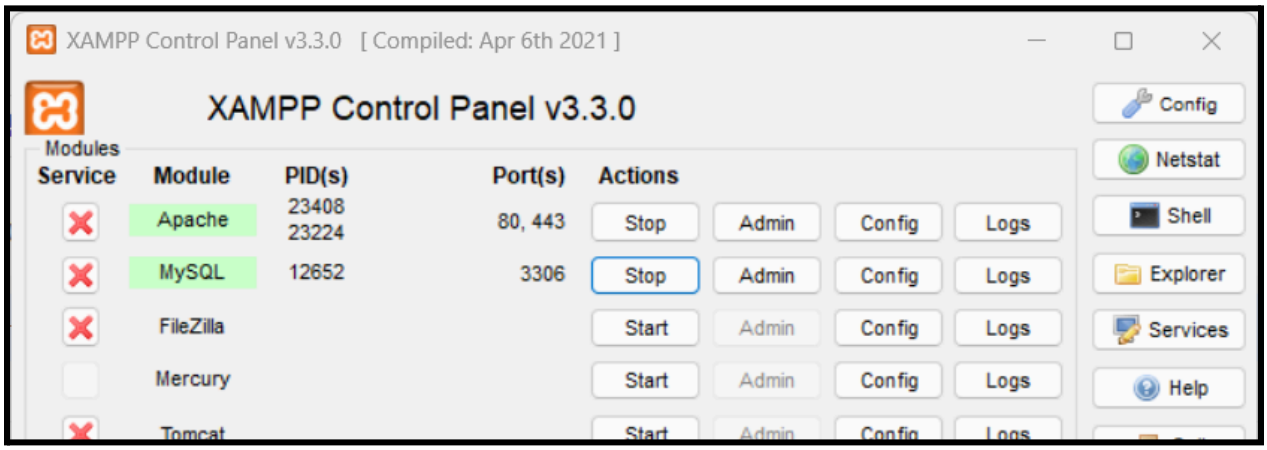
Step 8: Launch Edge(to use a different browser).

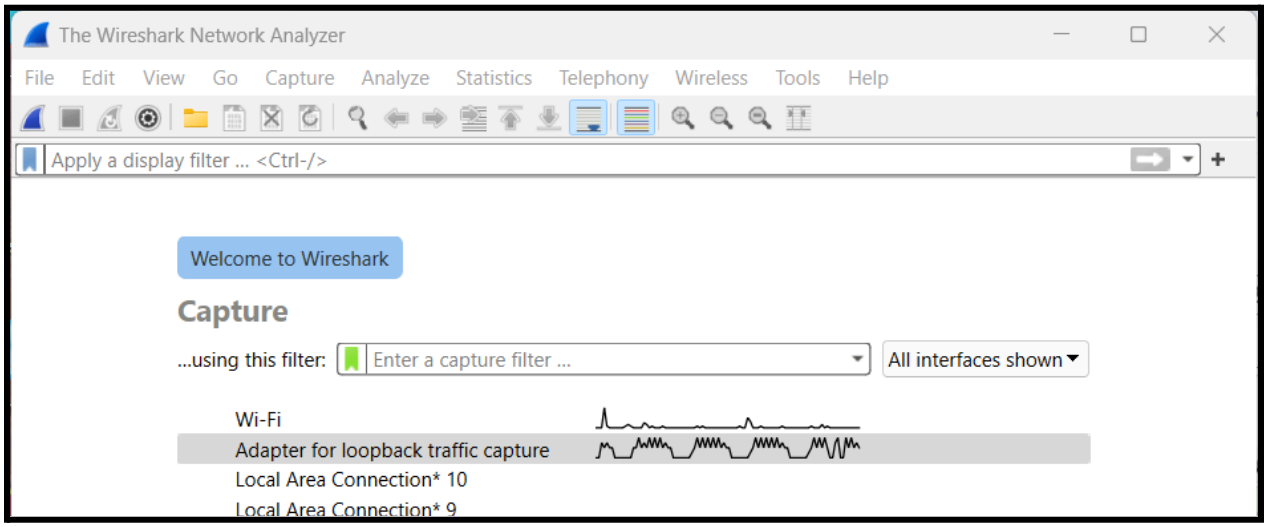
Step 9: Log in with a new user ID and password.

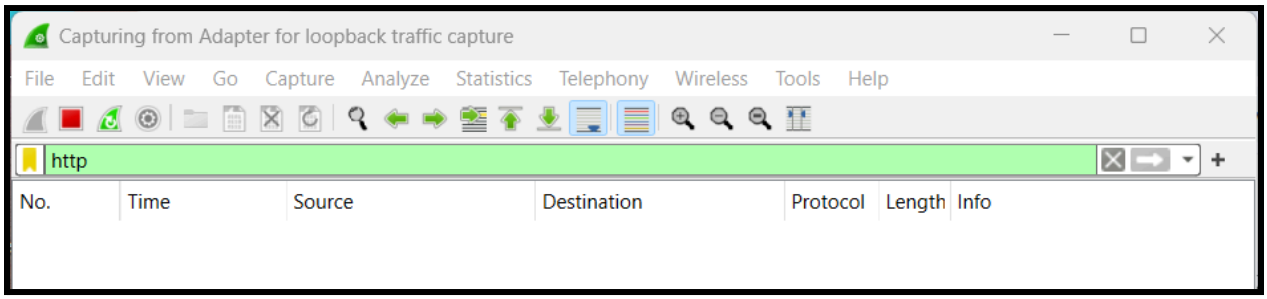
Step 10: Paste the cookie value obtained from the first user into the Cookie value field for the second user.

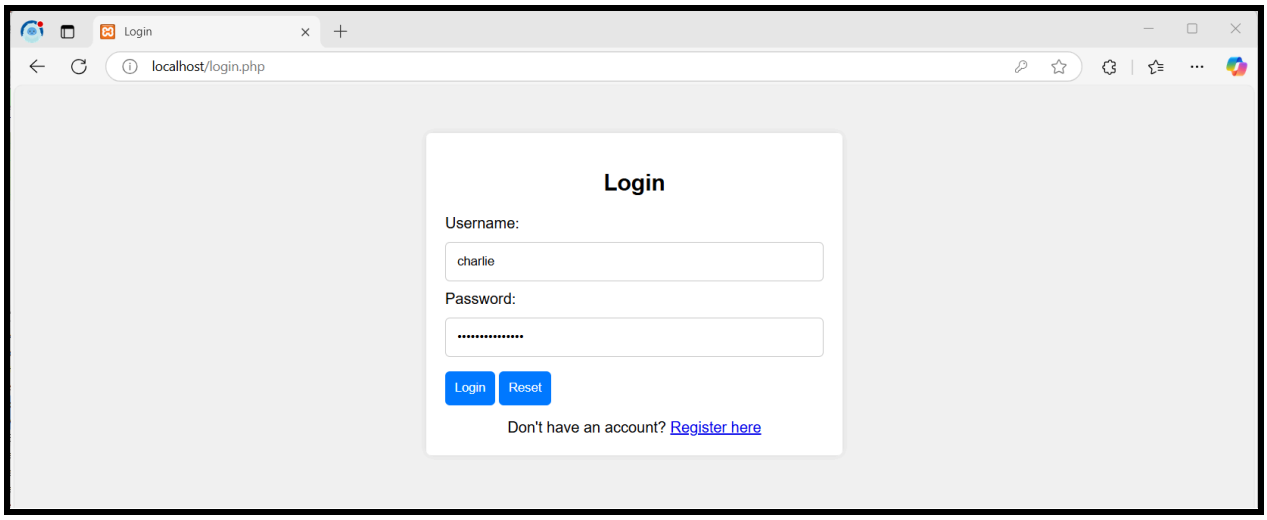
- Use the Cookie Editor extension:
    - Click on "Allow" to launch the extension.
    - Replace the existing cookie value with the copied value.
    - Click "Save" to apply the changes.

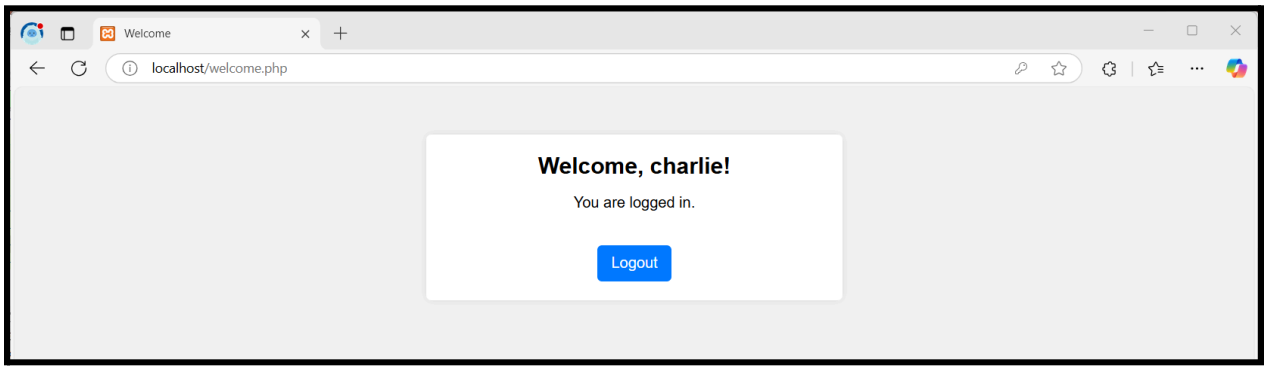
Output:

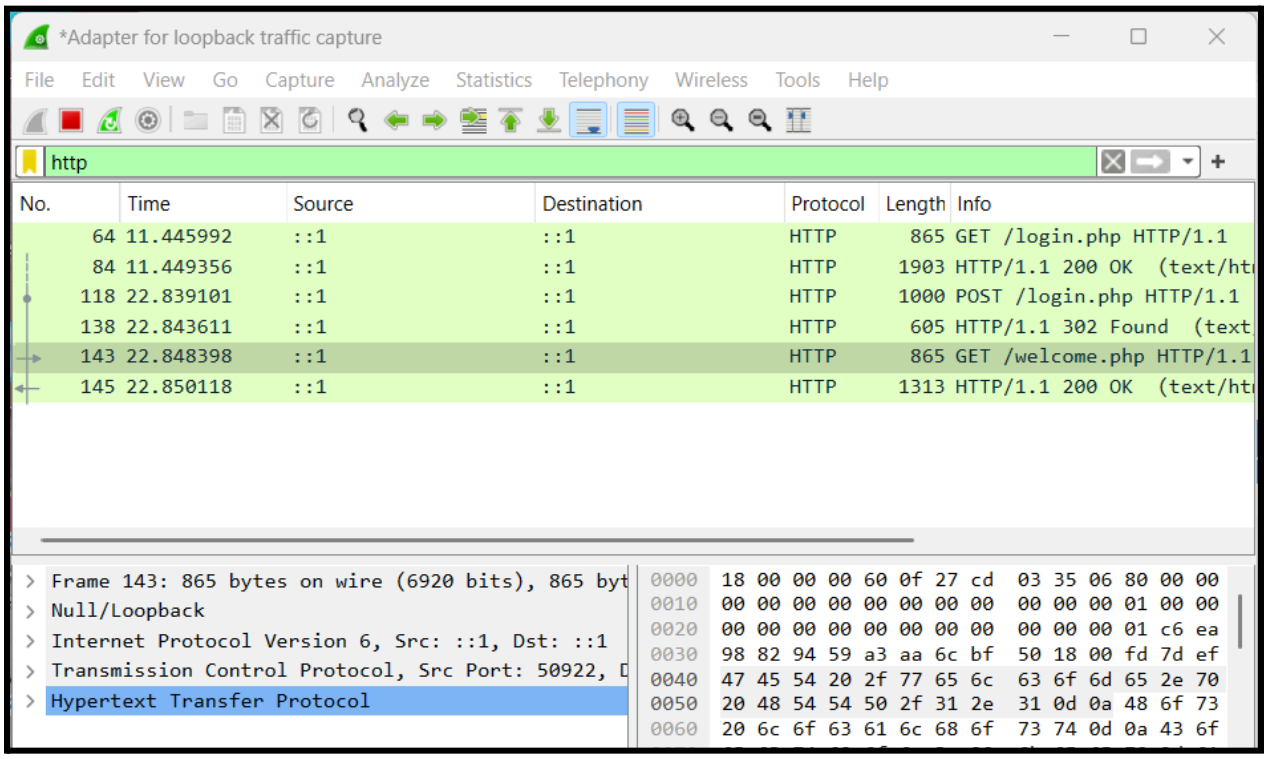

XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Modules
Service
Module
PID(s)
Port(s)
Actions
Apache
23408
23224
80, 443
Stop
Admin
Config
Logs
MySQL
12652
3306
Stop
Admin
Config
Logs
FileZilla
Start
Admin
Config
Logs
Mercury
Start
Admin
Config
Logs
Tomcat
Start
Admin
Config
Logs
Config
Netstat
Shell
Explorer
Services
Help

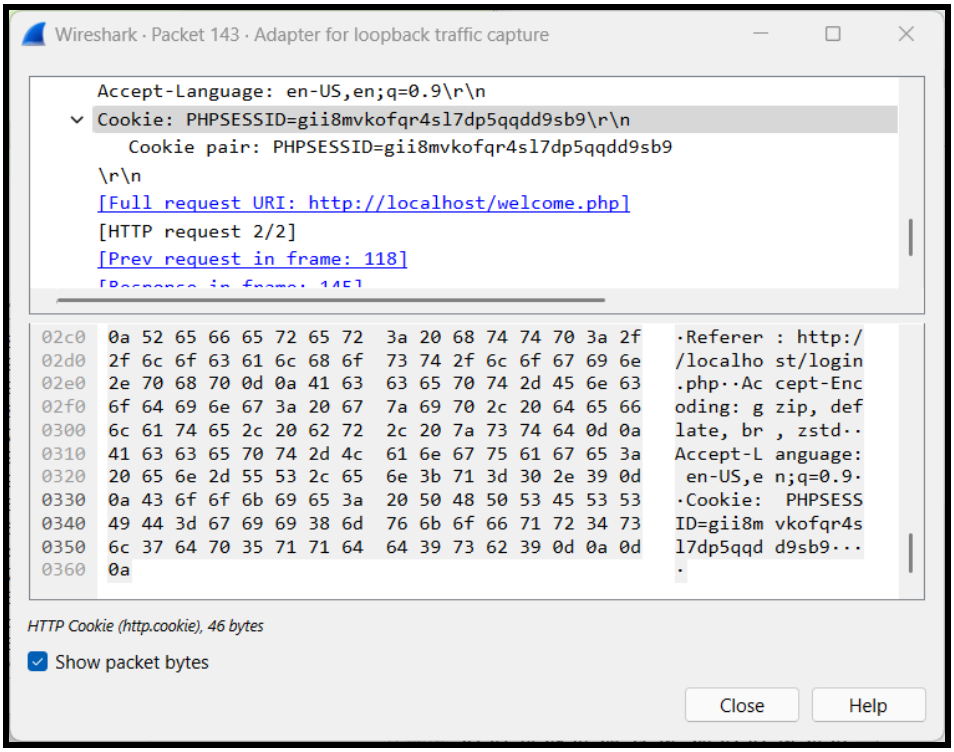

The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark
Capture
...using this filter:
Enter a capture filter ...
All interfaces shown
Wi-Fi
Adapter for loopback traffic capture
Local Area Connection* 10
Local Area Connection* 9

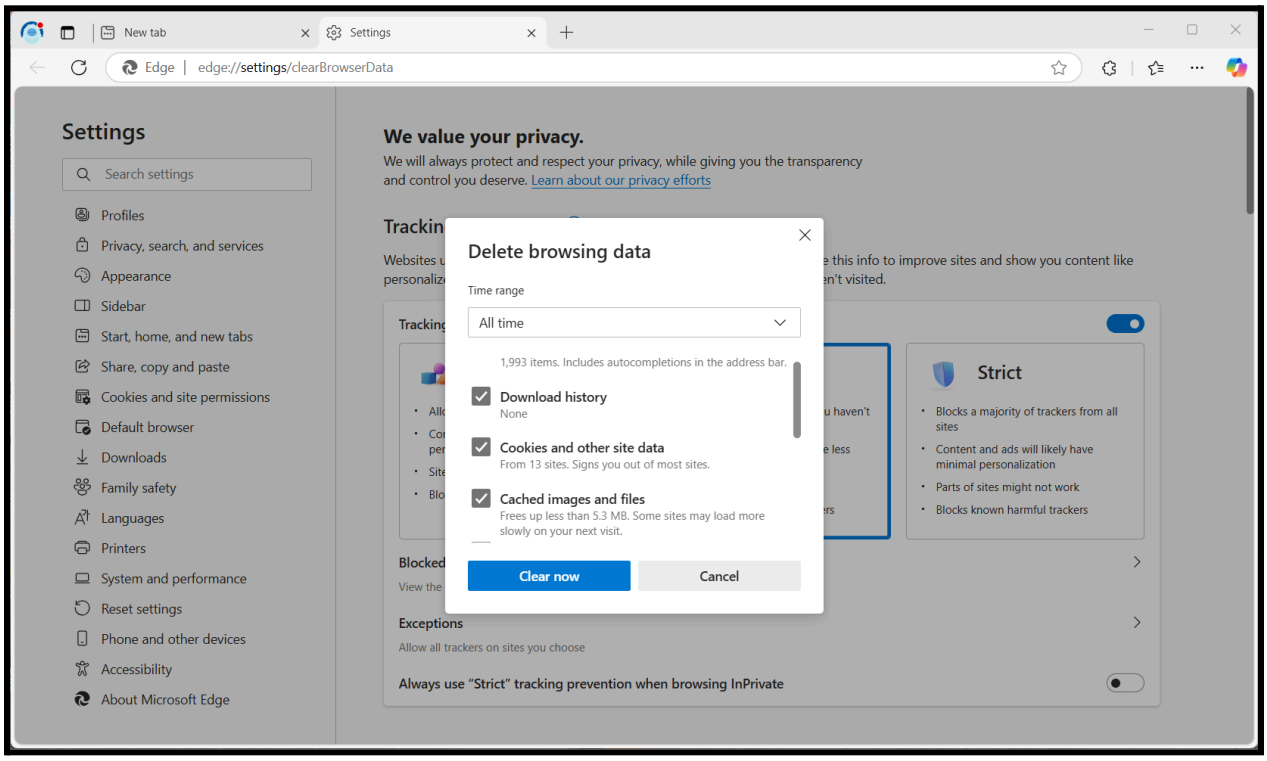

Capturing from Adapter for loopback traffic capture
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
No.
Time
Source
Destination
Protocol
Length
Info

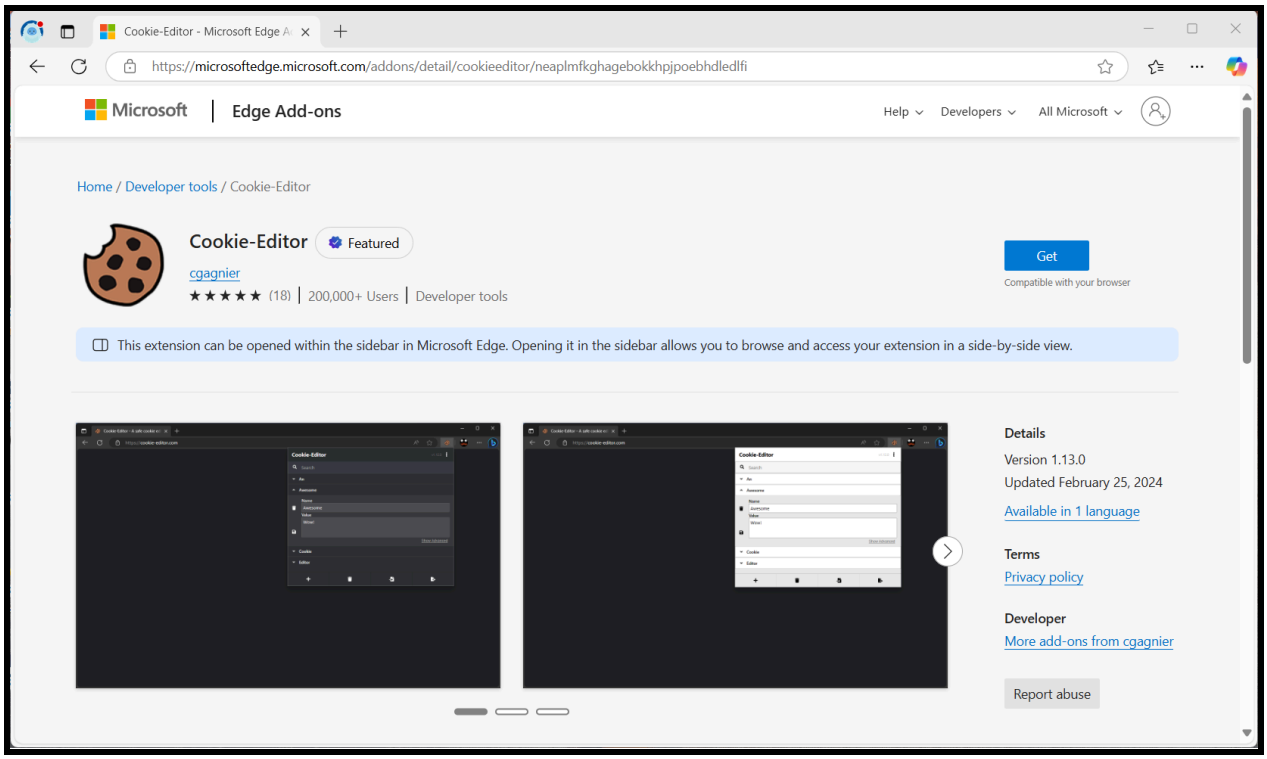

Login
localhost/login.php
Login
Username:
charlie
Password:
Login
Reset
Don't have an account? Register here

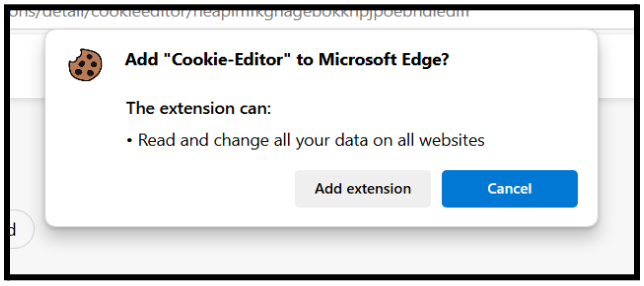

Welcome
localhost/welcome.php
Welcome, charlie!
You are logged in.
Logout

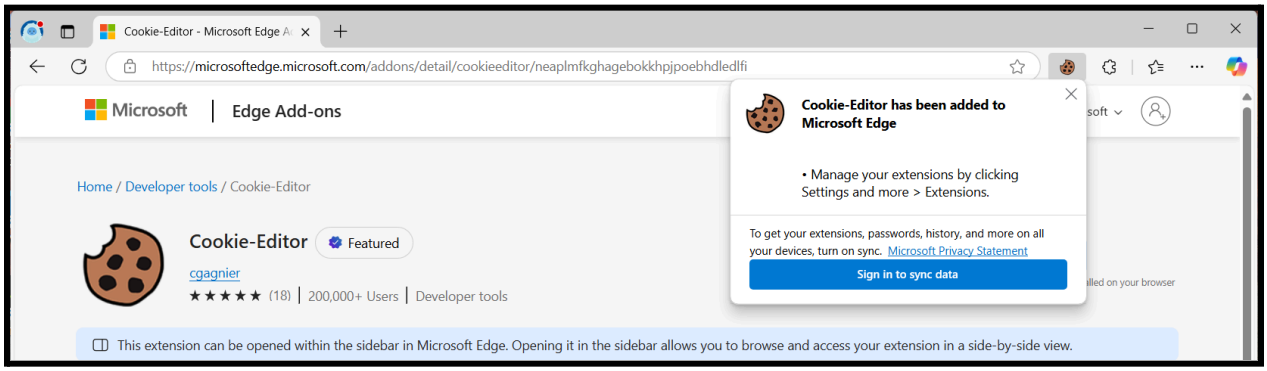

*Adapter for loopback traffic capture
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
No. Time Source Destination Protocol Length Info
64 11.445992 ::1 ::1 HTTP 865 GET /login.php HTTP/1.1
84 11.449356 ::1 ::1 HTTP 1903 HTTP/1.1 200 OK (text/ht
118 22.839101 ::1 ::1 HTTP 1000 POST /login.php HTTP/1.1
138 22.843611 ::1 ::1 HTTP 605 HTTP/1.1 302 Found (text
143 22.848398 ::1 ::1 HTTP 865 GET /welcome.php HTTP/1.1
145 22.850118 ::1 ::1 HTTP 1313 HTTP/1.1 200 OK (text/ht
Frame 143: 865 bytes on wire (6920 bits), 865 byt
Null/Loopback
Internet Protocol Version 6, Src: ::1, Dst: ::1
Transmission Control Protocol, Src Port: 50922, D
Hypertext Transfer Protocol
0000 18 00 00 00 60 0f 27 cd 03 35 06 80 00 00
0010 00 00 00 00 00 00 00 00 00 00 00 01 00 00
0020 00 00 00 00 00 00 00 00 00 00 00 01 c6 ea
0030 98 82 94 59 a3 aa 6c bf 50 18 00 fd 7d ef
0040 47 45 54 20 2f 77 65 6c 63 6f 6d 65 2e 70
0050 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73
0060 20 6c 6f 63 61 6c 68 6f 73 74 0d 0a 43 6f

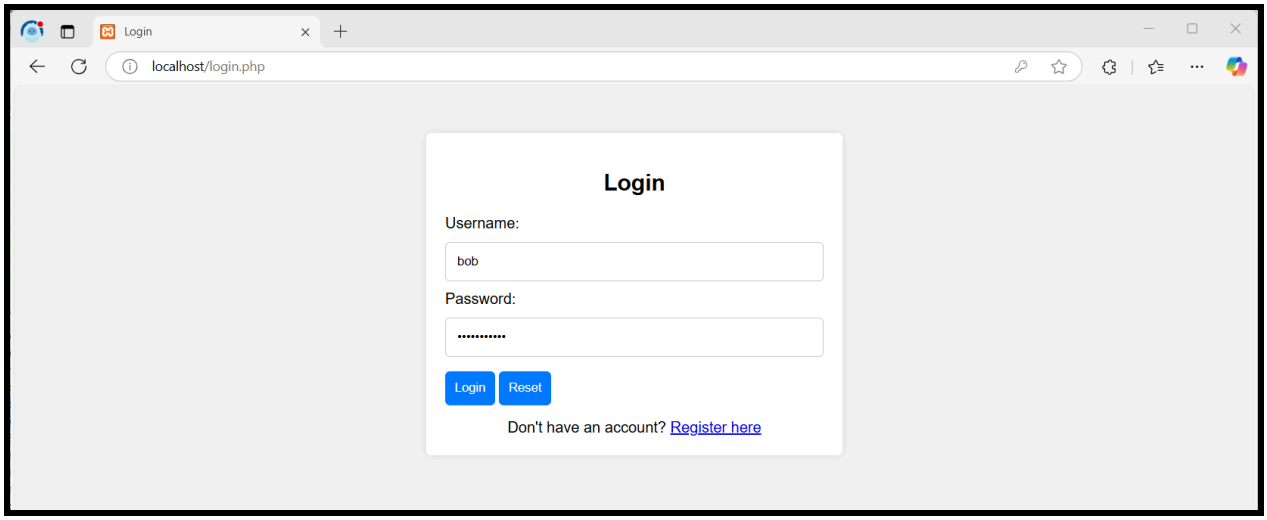

Wireshark · Packet 143 · Adapter for loopback traffic capture
Accept-Language: en-US,en;q=0.9\r\n
Cookie: PHPSESSID=gii8mvkofqr4sl7dp5qqdd9sb9\r\n
Cookie pair: PHPSESSID=gii8mvkofqr4sl7dp5qqdd9sb9
\r\n
[Full request URI: http://localhost/welcome.php]
[HTTP request 2/2]
[Prev request in frame: 118]
02c0 0a 52 65 66 65 72 65 72 3a 20 68 74 74 70 3a 2f ·Referer : http:/
02d0 2f 6c 6f 63 61 6c 68 6f 73 74 2f 6c 6f 67 69 6e /localho st/login
02e0 2e 70 68 70 0d 0a 41 63 63 65 70 74 2d 45 6e 63 .php··Ac cept-Enc
02f0 6f 64 69 6e 67 3a 20 67 7a 69 70 2c 20 64 65 66 oding: g zip, def
0300 6c 61 74 65 2c 20 62 72 2c 20 7a 73 74 64 0d 0a late, br , zstd··
0310 41 63 63 65 70 74 2d 4c 61 6e 67 75 61 67 65 3a Accept-L anguage:
0320 20 65 6e 2d 55 53 2c 65 6e 3b 71 3d 30 2e 39 0d en-US,e n;q=0.9·
0330 0a 43 6f 6f 6b 69 65 3a 20 50 48 50 53 45 53 53 ·Cookie: PHPSESS
0340 49 44 3d 67 69 69 38 6d 76 6b 6f 66 71 72 34 73 ID=gii8m vkofqr4s
0350 6c 37 64 70 35 71 71 64 64 39 73 62 39 0d 0a 0d l7dp5qqd d9sb9···
0360 0a ·
HTTP Cookie (http.cookie), 46 bytes
Show packet bytes
Close
Help

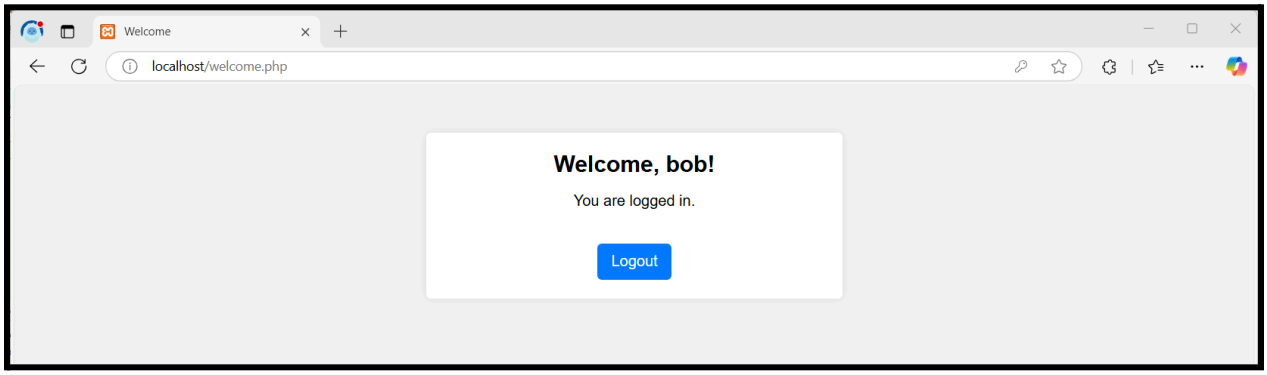

New tab
Settings
Edge | edge://settings/clearBrowserData
Settings
Search settings
Profiles
Privacy, search, and services
Appearance
Sidebar
Start, home, and new tabs
Share, copy and paste
Cookies and site permissions
Default browser
Downloads
Family safety
Languages
Printers
System and performance
Reset settings
Phone and other devices
Accessibility
About Microsoft Edge
We value your privacy.
We will always protect and respect your privacy, while giving you the transparency and control you deserve. Learn about our privacy efforts
Delete browsing data
Time range
All time
1,993 items. Includes autocompletions in the address bar.
Download history
None
Cookies and other site data
From 13 sites. Signs you out of most sites.
Cached images and files
Frees up less than 5.3 MB. Some sites may load more slowly on your next visit.
Clear now
Cancel
Strict
Blocks a majority of trackers from all sites
Content and ads will likely have minimal personalization
Parts of sites might not work
Blocks known harmful trackers
Exceptions
Allow all trackers on sites you choose
Always use "Strict" tracking prevention when browsing InPrivate

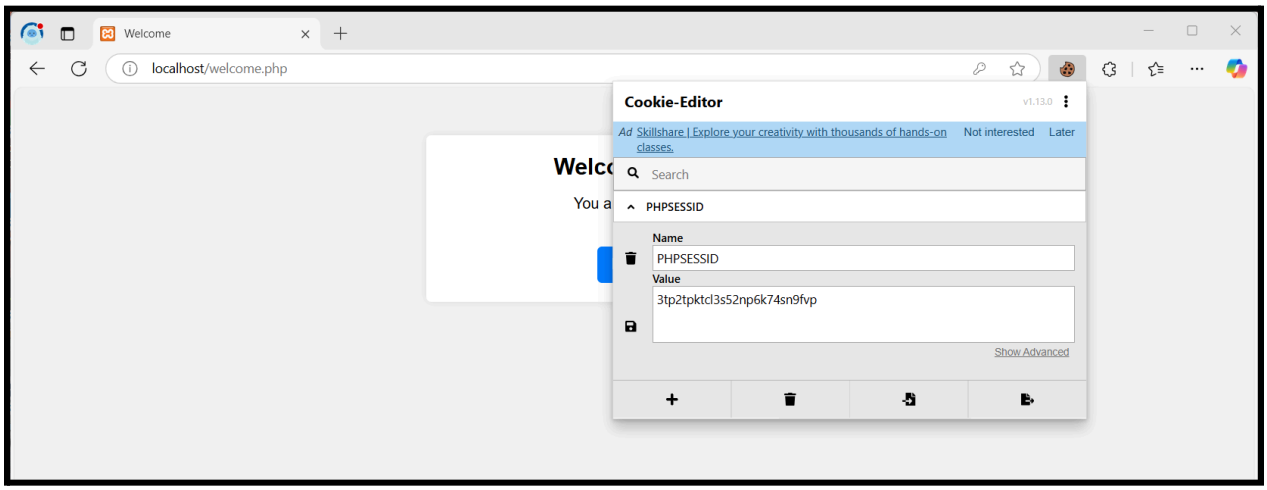

Cookie-Editor - Microsoft Edge
https://microsoftedge.microsoft.com/addons/detail/cookieeditor/neaplmfkghagebokkhpjpoebhdledlfi
Microsoft | Edge Add-ons
Help
Developers
All Microsoft
Home / Developer tools / Cookie-Editor
Cookie-Editor
Featured
cgagnier
(18) | 200,000+ Users | Developer tools
Get
Compatible with your browser
This extension can be opened within the sidebar in Microsoft Edge. Opening it in the sidebar allows you to browse and access your extension in a side-by-side view.
Details
Version 1.13.0
Updated February 25, 2024
Available in 1 language
Terms
Privacy policy
Developer
More add-ons from cgagnier
Report abuse

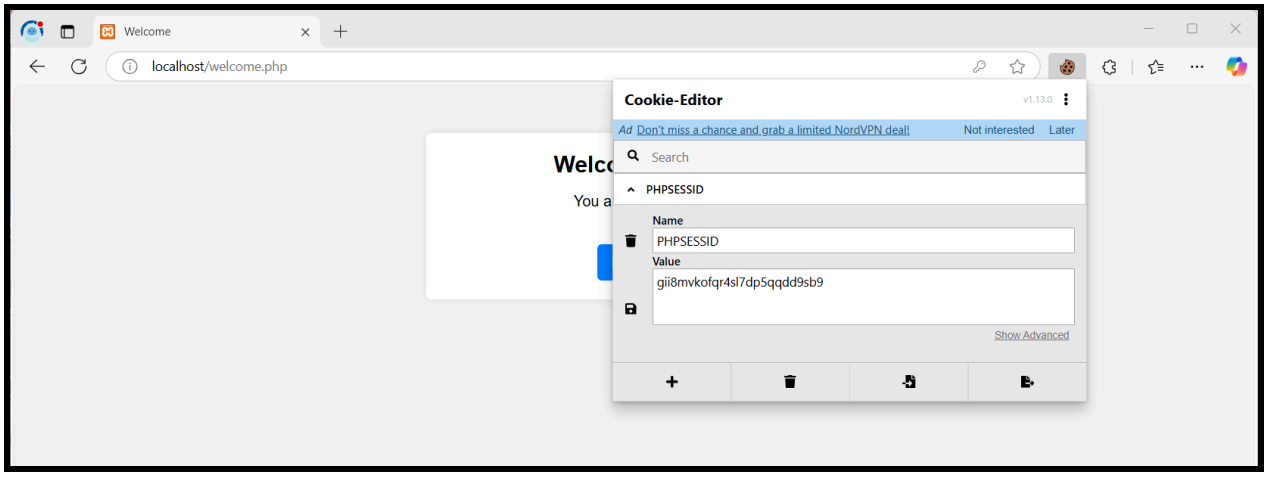

Add "Cookie-Editor" to Microsoft Edge?
The extension can:
• Read and change all your data on all websites
Add extension
Cancel

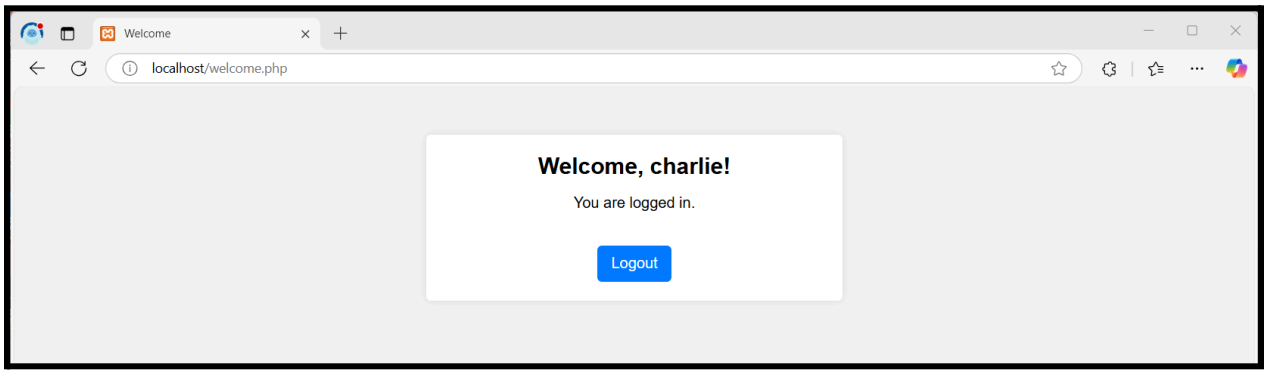

Cookie-Editor - Microsoft Edge
https://microsoftedge.microsoft.com/addons/detail/cookieeditor/neaplmfkghagebokkhpjpoebhdledlfi
Microsoft | Edge Add-ons
Cookie-Editor has been added to Microsoft Edge
• Manage your extensions by clicking Settings and more > Extensions.
To get your extensions, passwords, history, and more on all your devices, turn on sync. Microsoft Privacy Statement
Sign in to sync data
Home / Developer tools / Cookie-Editor
Cookie-Editor
Featured
cgagnier
(18) | 200,000+ Users | Developer tools
This extension can be opened within the sidebar in Microsoft Edge. Opening it in the sidebar allows you to browse and access your extension in a side-by-side view.


Login
localhost/login.php
Login
Username:
bob
Password:
Login
Reset
Don't have an account? Register here


Welcome
localhost/welcome.php
Welcome, bob!
You are logged in.
Logout


Welcome
localhost/welcome.php
Cookie-Editor
Ad Skillshare | Explore your creativity with thousands of hands-on classes.
Not interested
Later
Search
PHPSESSID
Name
PHPSESSID
Value
3tp2tpktcl3s52np6k74sn9fvp
Show Advanced


Welcome
localhost/welcome.php
Cookie-Editor
Ad Don't miss a chance and grab a limited NordVPN deal! Not interested Later
Search
PHPSESSID
Name
PHPSESSID
Value
gii8mvkofqr4sl7dp5qqdd9sb9
Show Advanced


Welcome
localhost/welcome.php
Welcome, charlie!
You are logged in.
Logout