

Cross-Site Request Forgery (CSRF) Protection

Overview

In this lab, you will learn about Cross-Site Request Forgery (CSRF) attacks and how to implement protection mechanisms to safeguard web applications. You will explore how CSRF attacks exploit the trust that a web application has in the user's browser.

Prerequisites

- Basic knowledge of web application security concepts.
- Familiarity with session management and HTTP requests.
- Access to a web application environment (e.g., DVWA) for testing.

Lab Objectives

- Understand how CSRF attacks work and their impact on web applications.
- Implement CSRF protection using tokens.
- Test the effectiveness of CSRF defenses.

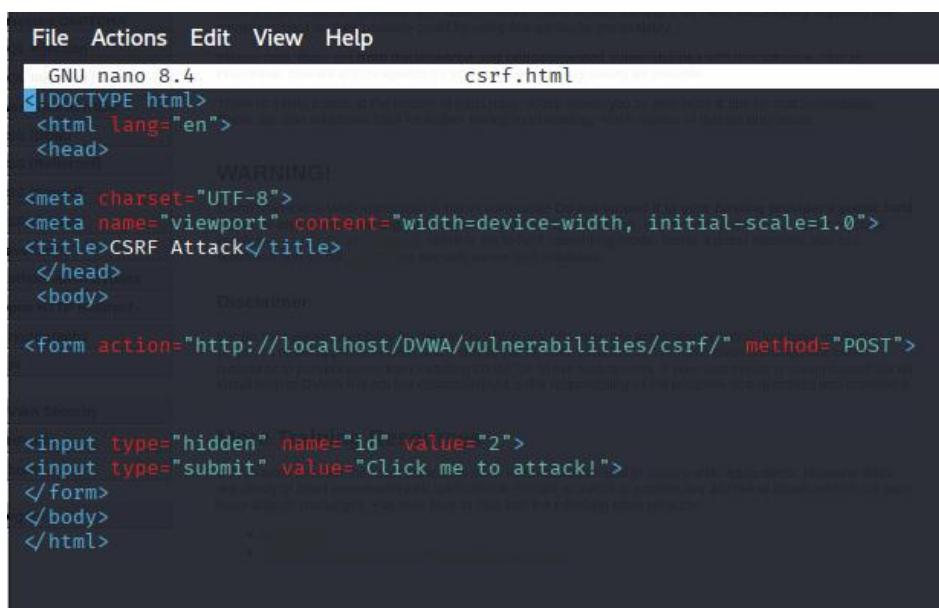
Exercise 1: Understanding CSRF Attacks

1. Explore DVWA:

- Open DVWA in your browser and log in with your credentials (admin/password).
- Navigate to the CSRF section of the application.

2. Simulate a CSRF Attack:

- Open a new browser tab or window and create a simple HTML form to simulate a CSRF attack.
- Replace `http://your-dvwa-url` with your actual DVWA URL.



The screenshot shows a terminal window with a black background and white text. At the top, there is a menu bar with options: File, Actions, Edit, View, Help. Below the menu, it says "GNU nano 8.4". The file name "csrf.html" is displayed. The content of the file is a simple HTML form designed to perform a CSRF attack. It includes meta tags for charset and viewport, a title, and a form with an action set to "http://localhost/DVWA/vulnerabilities/csrf/" and a method of "POST". The form contains two inputs: a hidden input with name "id" and value "2", and a submit button with the value "Click me to attack!".

```
File Actions Edit View Help
GNU nano 8.4                               csrf.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CSRF Attack</title>
</head>
<body>
    <form action="http://localhost/DVWA/vulnerabilities/csrf/" method="POST">
        <input type="hidden" name="id" value="2">
        <input type="submit" value="Click me to attack!">
    </form>
</body>
</html>
```

- Open this HTML file in your browser and click the submit button after logging into DVWA

The screenshot shows the DVWA application interface. On the left is a sidebar with various lab categories: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, **CSRF**, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript, Authorisation Bypass, Open HTTP Redirect, Cryptography, API, DVWA Security, PHP Info, About, and Logout. The 'Logout' button is highlighted.

The main content area has a title "Vulnerability: Cross Site Request Forgery (CSRF)". It contains a form titled "Change your admin password:" with fields for "Test Credentials", "New password:", and "Confirm new password:". A "Change" button is present, and a red message "Password Changed." is displayed below it. Below the form, a note states: "Note: Browsers are starting to default to setting the [SameSite cookie](#) flag to Lax, and in doing so are killing off some types of CSRF attacks. When they have completed their mission, this lab will not work as originally expected." Another note says: "As an alternative to the normal attack of hosting the malicious URLs or code on a separate host, you could try using other vulnerabilities in this app to store them, the Stored XSS lab would be a good place to start." A "More Information" section lists three links: <https://owasp.org/www-community/attacks/csrf>, <https://www.csafsecurity.com/csrf-faq.html>, and https://en.wikipedia.org/wiki/Cross-site_request_forgery.

At the bottom of the main content area, there are "View Source" and "View Help" buttons. The footer displays user information: Username: admin, Security Level: low, Locale: en, and SQLI DB: mysql.

3. Evaluate the Attack:

- Yes it successfully change the user settings or execute a sensitive action?
 - Yes, it did work because it displayed “Password Changed”
 - Also, when i log out i was able to log in with the new password i change to

Discuss how this demonstrates the importance of CSRF protection.

- CSRF exploit the trust between a website and its user
- It shows DVWA doesn't check for csrf validation because it accept the forged request
- It poses a real-world danger

Exercise 2: Implementing CSRF Protection

1. Modify the Application Code:

- In your DVWA environment, navigate to the CSRF protection implementation area. If you don't have one, you can add a CSRF token to forms manually.
- PHP Code Example for generating and validating CSRF tokens:

```
<?php
define( 'DVWA_WEB_PAGE_TO_ROOT', '..../..');
require_once DVWA_WEB_PAGE_TO_ROOT . 'dvwa/includes/dvwaPage.inc.php';

dvwaPageStartup( array( 'authenticated' ) );

// Start session (DVWA does this, but ensure active)
if (session_status() === PHP_SESSION_NONE) {
    session_start();
}

// Generate CSRF token
if (empty($_SESSION['csrf_token'])) {
    $_SESSION['csrf_token'] = bin2hex(random_bytes(32));
}

$page = dvwaPageNewGrab();
$page[ 'title' ] = 'Vulnerability: Cross Site Request Forgery (CSRF)' . $page[ 'title_separator' ].$page[ 'title' ];
$page[ 'page_id' ] = 'csrf';
$page[ 'help_button' ] = 'csrf';
$page[ 'source_button' ] = 'csrf';

dvwaDatabaseConnect();

$vulnerabilityFile = '';
switch( dvwaSecurityLevelGet() ) {
```

2. Validate CSRF Tokens:

- Add validation logic to check the CSRF token upon form submission:

```
session_start(); // Validate the CSRF token
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    if (!hash_equals($_SESSION['csrf_token'], $_POST['csrf_token'])) {
```

3. Reflection:

- Discuss how CSRF tokens work to prevent unauthorized actions. What other security measures could complement this?

How CSRF Tokens Work:

CSRF (Cross-Site Request Forgery) tokens are unique, unpredictable values generated by the server and tied to a user's session. Each time a form or state-changing request is made, the token is embedded in the request (usually as a hidden form field or in a header). When the server receives the request, it validates the submitted token against the one stored in the user's session.

- If the token is valid the request will be accepted.
- If the token is missing, invalid, or reused the request will be rejected.

This prevents attackers from performing unauthorized actions on behalf of a logged-in user because they cannot predict or obtain the valid token.

Session Management and Web Security

Challenge Title: Testing for Session Hijacking on AIVTIC Portal

Objective

The objective of this CTF challenge is to enhance your practical skills in identifying and exploiting session hijacking vulnerabilities. You will use your login credentials to attempt to detect and exploit potential session hijacking issues on the AIVTIC portal.

Challenge Overview

1. Understanding Session Hijacking

- Research: Start by reviewing concepts of session hijacking, focusing on how attackers exploit weaknesses in web session management. Understand the tools and methods commonly used to detect session-related vulnerabilities.

2. Test Setup

- Access: Log in to the AIVTIC portal using your assigned credentials.
- Observation: Use browser Developer Tools, Burp Suite, or OWASP ZAP to observe session cookies and their management in your browser.

The screenshot shows the "Storage" tab of the browser developer tools for the "Student Personalized Portal". The left sidebar lists storage types: Cache Storage, Cookies, Indexed DB, Local Storage, and Session Storage. Under Cookies, there are two entries for the domain `https://portal.icdfa.edu.ng`: `_ga_L2B...` and `_ga`. The `PHPSESSID` cookie is selected. The right panel displays detailed information about the `PHPSESSID` cookie, including its name, value, domain, path, and expiration date (Wed, 01 Sep 2025). It also shows the parsed value, which is an array containing the session ID.

| Name | Value | Domain | Path | Expires / Max-Age |
|-------------------------|-------------------------------|------------------------------|----------------|-------------------------------|
| <code>_ga_L2B...</code> | <code>GS2.1.s175669...</code> | <code>.icdfa.edu.ng</code> | <code>/</code> | <code>Wed, 01 Sep 2025</code> |
| <code>_ga</code> | <code>GA1.1.1377041...</code> | <code>.icdfa.edu.ng</code> | <code>/</code> | <code>Wed, 01 Sep 2025</code> |
| <code>PHPSESSID</code> | <code>O1J4v%2CTaA...</code> | <code>portal.icdfa...</code> | <code>/</code> | <code>Session</code> |

PHPSESSID: "O1J4v%2CTaAdNFJ...o9AElyx%2C36D3S"
Created: "Mon, 01 Sep 2025 01:56:09 GMT"
Domain: "portal.icdfa.edu.ng"
Expires / Max-Age: "Session"
HostOnly: true
HttpOnly: true
Last Accessed: "Mon, 01 Sep 2025 02:08:21 GMT"
Path: "/"
SameSite: "Strict"
Secure: true
Size: 143

Parsed Value:

```
PHPSESSID:Array
0:"O1J4v"
1:"TaAdNFJ2tMEbWtEhuC4..UiIqpx2dzgUgVBxCR"
2:"dVlrmGzjwISE57lwC-K...YnCx78mwQPBo9AElyx"
3:"36D3S"
length:4
__proto__:Array
```

| Time | Type | Direction | Method | URL |
|------------|-------|-----------|--------|--|
| 03:22:0... | HT... | → Request | POST | https://portal.icdfa.edu.ng/portal/index.php |

Request

Pretty Raw Hex

```

1 POST /portal/index.php HTTP/1.1
2 Host: portal.icdfa.edu.ng
3 Cookie: _ga_L2BZ8LBRST=GS2.1.s1756691738$ol$gl$t1756691763$j35$l0$h0; _ga=GAI.1.1377041483.1756691738; PHPSESSID=a52e4386fd1aa5a02cb8dcf3534f118b
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 151
10 Origin: https://portal.icdfa.edu.ng
11 Referer: https://portal.icdfa.edu.ng/portal/index.php
12 Upgrade-Insecure-Requests: 1
13 Sec-Fetch-Dest: document
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-User: ?1
17 Priority: u=0, i
18 Te: trailers
19 Connection: keep-alive
20
21 csrf_token=3d0f9431ce2e3db31e96c3f40b16b28115231db5e6e4d921f4f1f0392b90ead0&
txtEmail=int258475%40interns.icdfa.org.ng&txtpass=muritalA%401508&btnlogin=

```

Inspector

- Request attributes
- Request query parameters
- Request body parameters
- Request cookies
- Request headers

CDSA Student Login + Private browsing

→ ⌛ ⌂ ⌃ https://portal.icdfa.edu.ng/portal/index.php 80% ⭐

Sec 🐞 Kali Linux 📲 Kali Tools 📖 Kali Docs 🚩 Kali Forums 🚧 Kali Nethunter 🚨 Exploit-DB 🚨 Google Hacking DB

| Storage | Name | Value | Domain | Path | Expires / Max-Age |
|-----------------|-----------|----------------------------------|---------------------|------|-------------------|
| Session Storage | PHPSESSID | a52e4386fd1aa5a02cb8dcf3534f118b | portal.icdfa.edu.ng | / | Session |

PHPSESSID: "a52e4386fd1aa5a02cb8dcf3534f118b"
 Created: "Mon, 01 Sep 2025 02:28:01 GMT"
 Domain: "portal.icdfa.edu.ng"
 Expires / Max-Age: "Session"
 HostOnly: true
 HttpOnly: true
 Last Accessed: "Mon, 01 Sep 2025 02:29:12 GMT"
 Path: "/"
 SameSite: "None"
 Secure: true
 Size: 41

3. Session Hijacking Attempt

- Simulate an Attack: Utilize the knowledge gained to simulate a session hijacking attack on your own session. Use tools like Wireshark to monitor session traffic, intercept cookies, or test session expiration scenarios.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|--------------|-----------------|-----------------|----------|--------|-------------------------|
| 13 | 11.693758735 | 192.168.35.61 | 131.153.147.186 | TCP | 74 | 33530 - 443 [SYN] Seq: |
| 14 | 11.694096325 | 192.168.35.61 | 131.153.147.186 | TCP | 74 | 33538 - 443 [SYN] Seq: |
| 15 | 11.694285594 | 192.168.35.61 | 131.153.147.186 | TCP | 74 | 33554 - 443 [SYN] Seq: |
| 18 | 11.905625934 | 131.153.147.186 | 192.168.35.61 | TCP | 74 | 443 - 33554 [SYN, ACK] |
| 19 | 11.905672231 | 192.168.35.61 | 131.153.147.186 | TCP | 66 | 33554 - 443 [ACK] Seq: |
| 20 | 11.906822203 | 192.168.35.61 | 131.153.147.186 | TLSv1.3 | 733 | Client Hello (SNI=port) |
| 21 | 11.906847454 | 131.153.147.186 | 192.168.35.61 | TCP | 74 | 443 - 33538 [SYN, ACK] |
| 22 | 11.908678798 | 192.168.35.61 | 131.153.147.186 | TCP | 66 | 33538 - 443 [ACK] Seq: |
| 23 | 11.908798654 | 131.153.147.186 | 192.168.35.61 | TCP | 74 | 443 - 33530 [SYN, ACK] |
| 24 | 11.908809182 | 192.168.35.61 | 131.153.147.186 | TCP | 66 | 33530 - 443 [ACK] Seq: |
| 25 | 11.909596238 | 192.168.35.61 | 131.153.147.186 | TLSv1.3 | 733 | Client Hello (SNI=port) |
| 26 | 11.910274457 | 192.168.35.61 | 131.153.147.186 | TLSv1.3 | 733 | Client Hello (SNI=port) |
| 27 | 12.121794962 | 131.153.147.186 | 192.168.35.61 | TCP | 66 | 443 - 33554 [ACK] Seq: |
| 28 | 12.121812872 | 131.153.147.186 | 192.168.35.61 | TCP | 66 | 443 - 33538 [ACK] Seq: |
| 29 | 12.125989133 | 131.153.147.186 | 192.168.35.61 | TLSv1.3 | 1434 | Server Hello, Change |
| 30 | 12.126634028 | 192.168.35.61 | 131.153.147.186 | TCP | 66 | 33554 - 443 [ACK] Seq: |
| 31 | 12.126168507 | 131.153.147.186 | 192.168.35.61 | TCP | 1434 | 443 - 33554 [ACK] Seq: |
| 32 | 12.126179823 | 192.168.35.61 | 131.153.147.186 | TCP | 66 | 33554 - 443 [ACK] Seq: |
| 33 | 12.126235991 | 131.153.147.186 | 192.168.35.61 | TLSv1.3 | 546 | Application Data, App |
| 34 | 12.12624332 | 192.168.35.61 | 131.153.147.186 | TCP | 66 | 33554 - 443 [ACK] Seq: |
| 35 | 12.130578441 | 192.168.35.61 | 131.153.147.186 | TLSv1.3 | 130 | Change Cipher Spec, A |
| 36 | 12.130577924 | 131.153.147.186 | 192.168.35.61 | TLSv1.3 | 1434 | Server Hello, Change |
| 37 | 12.130618599 | 131.153.147.186 | 192.168.35.61 | TCP | 1434 | 443 - 33538 [ACK] Seq: |

| No. | Time | Source | Destination | Protocol | Length | Info |
|------|---------------|-----------------|-----------------|----------|--------|---|
| 1584 | 442, 08733399 | 192.168.35.61 | 131.153.147.186 | TCP | 74 | 36620 - 443 [Seq=9] Win=64248 Len=0 MSS=1466 SACK_FWMR Tsvl=1951431075 Tscr=1951431075 |
| 1585 | 442, 08733399 | 192.168.35.61 | 192.168.35.61 | TCP | 68 | 36620 - 443 [ACK] Seq=1 Win=32456 Len=0 MSS=1399 SACK_FWMR Tsvl=3853109703 Tscr=3853109703 |
| 1586 | 442, 08733399 | 192.168.35.61 | 131.153.147.186 | TLSV1.2 | 1384 | Client Hello (SNI=portal.icdfds.edu.ng) |
| 1587 | 442, 08733399 | 192.168.35.61 | 192.168.35.61 | TCP | 66 | 36620 - 443 [ACK] Seq=1 Ack=1 Win=32456 Len=0 MSS=1399 SACK_FWMR Tsvl=3853109703 Tscr=3853109703 |
| 1588 | 442, 08733399 | 192.168.35.61 | 131.153.147.186 | TLSV1.2 | 1384 | Client Hello (SNI=portal.icdfds.edu.ng) |
| 1589 | 442, 08733399 | 192.168.35.61 | 192.168.35.61 | TCP | 66 | 3643 - 36028 [ACK] Seq=1 Ack=1 Win=32456 Len=0 MSS=1399 SACK_FWMR Tsvl=3853109703 Tscr=3853109703 |
| 1590 | 442, 08733399 | 192.168.35.61 | 131.153.147.186 | TCP | 103 | 36028 - 3643 [ACK] Seq=1 Ack=245 Win=64128 Len=0 Tsvl=1901431955 Tscr=3853109919 |
| 1591 | 442, 08733399 | 192.168.35.61 | 131.153.147.186 | TLSV1.3 | 138 | Change Cipher Spec, Application Data, Application Data |
| 1592 | 442, 08733399 | 192.168.35.61 | 131.153.147.186 | TCP | 66 | 3643 - 443 [ACK] Seq=1999 Ack=245 Win=64128 Len=0 Tsvl=1901431955 Tscr=3853109919 |
| 1593 | 442, 08733399 | 192.168.35.61 | 192.168.35.61 | TLSV1.3 | 76 | Application Data |
| 1594 | 442, 08733399 | 192.168.35.61 | 131.153.147.186 | TCP | 66 | 36620 - 443 [ACK] Seq=1999 Ack=532 Win=64128 Len=0 Tsvl=1951431825 Tscr=3853101135 |
| 1595 | 442, 08733399 | 192.168.35.61 | 192.168.35.61 | TCP | 134 | 443 - 36628 [ACK] Seq=532 Ack=1999 Win=33292 Len=1368 Tsvl=3853101173 Tscr=1951431825 [TCP] |
| 1596 | 442, 08733399 | 192.168.35.61 | 131.153.147.186 | TCP | 66 | 36620 - 443 [ACK] Seq=1999 Ack=532 Win=33292 Len=1368 Tsvl=1951432273 Tscr=1951431825 |
| 1597 | 442, 08733399 | 192.168.35.61 | 131.153.147.186 | TCP | 134 | 443 - 36628 [ACK] Seq=1999 Ack=532 Win=33292 Len=1368 Tsvl=1951432273 Tscr=3853101732 |
| 1598 | 442, 08733399 | 192.168.35.61 | 192.168.35.61 | TCP | 66 | 36620 - 443 [ACK] Seq=1999 Ack=532 Win=33292 Len=1368 Tsvl=1951432273 Tscr=3853101732 |
| 1599 | 442, 08733399 | 192.168.35.61 | 131.153.147.186 | TLSV1.3 | 466 | Application Data |
| 1600 | 442, 08733399 | 192.168.35.61 | 192.168.35.61 | TCP | 66 | 36620 - 443 [ACK] Seq=1999 Ack=532 Win=62128 Len=0 Tsvl=1951432273 Tscr=3853101732 |
| 1601 | 442, 08733399 | 192.168.35.61 | 131.153.147.186 | TLSV1.3 | 466 | Application Data |
| 1602 | 442, 08733399 | 192.168.35.61 | 192.168.35.61 | TCP | 66 | 36620 - 443 [ACK] Seq=1999 Ack=532 Win=64128 Len=0 Tsvl=1951431825 Tscr=3853101135 |
| 1603 | 442, 08733399 | 192.168.35.61 | 131.153.147.186 | TCP | 134 | 443 - 36628 [ACK] Seq=532 Ack=1999 Win=33292 Len=1368 Tsvl=3853101173 Tscr=1951431825 [TCP] |
| 1604 | 442, 08733399 | 192.168.35.61 | 192.168.35.61 | TCP | 66 | 36620 - 443 [ACK] Seq=1999 Ack=532 Win=33292 Len=1368 Tsvl=1951432273 Tscr=1951431825 |
| 1605 | 442, 08733399 | 192.168.35.61 | 131.153.147.186 | TCP | 134 | 443 - 36628 [ACK] Seq=1999 Ack=532 Win=33292 Len=1368 Tsvl=1951432273 Tscr=3853101732 |
| 1606 | 442, 08733399 | 192.168.35.61 | 131.153.147.186 | TCP | 66 | 36620 - 443 [ACK] Seq=1999 Ack=532 Win=70616 Len=0 Tsvl=1951432273 Tscr=3853101732 |
| 1607 | 442, 08733399 | 192.168.35.61 | 131.153.147.186 | TCP | 134 | 443 - 36628 [ACK] Seq=1999 Ack=532 Win=70616 Len=0 Tsvl=1951432273 Tscr=3853101732 |
| 1608 | 442, 08733399 | 192.168.35.61 | 192.168.35.61 | TCP | 66 | 36620 - 443 [ACK] Seq=1999 Ack=532 Win=70616 Len=0 Tsvl=1951432273 Tscr=3853101732 |
| 1609 | 442, 08733399 | 131.153.147.186 | 192.168.35.61 | TLSV1.3 | 466 | Application Data |
| 1610 | 442, 08733399 | 192.168.35.61 | 131.153.147.186 | TCP | 66 | 36620 - 443 [ACK] Seq=1999 Ack=532 Win=75648 Len=0 Tsvl=1951432273 Tscr=3853101732 |
| 1611 | 442, 08733399 | 192.168.35.61 | 192.168.35.61 | TLSV1.3 | 466 | Application Data |
| 1612 | 442, 08733399 | 192.168.35.61 | 131.153.147.186 | TCP | 66 | 36620 - 443 [FIN, ACK] Seq=2623 Ack=5309 Win=75648 Len=0 Tsvl=19514344324 Tscr=3853101732 |
| 1613 | 442, 08733399 | 131.153.147.186 | 192.168.35.61 | TLSV1.3 | 9 | Application Data |
| 1614 | 442, 08733399 | 192.168.35.61 | 131.153.147.186 | TCP | 66 | 36620 - 443 [FIN, ACK] Seq=2623 Ack=5309 Win=75648 Len=0 Tsvl=19514344324 Tscr=3853101732 |
| 1615 | 442, 08733399 | 192.168.35.61 | 192.168.35.61 | TCP | 66 | 36620 - 443 [FIN, ACK] Seq=1999 Win=64128 Len=0 |
| 1616 | 442, 08733399 | 192.168.35.61 | 131.153.147.186 | TCP | 64 | 36620 - 443 [FIN, ACK] Seq=1999 Win=64128 Len=0 |

- Key Focus Areas: Session ID Predictability: Investigate if the session ID can be easily guessed or regenerated.
 - The session ID can not be easily guessed so its not predictible
 - The session ID generated after login appeared random and non-sequential.
 - No pattern or predictable structure was identified.
 - reduces the risk of attackers guessing or brute-forcing valid session IDs.

| Name | Value | Domain | Path | Expires / Max-Age | Data |
|------------|-------------------------------------|-----------------|------|-------------------|---|
| _ga_L2B... | GS2.1.s175669... | .icdfa.edu.ng | / | Wed, 01 Sep 20... | PHPSESSID: "UVeVfZkc6KAVVgFY...pfGLb9XwXVTmqj1" Created: "Wed, 03 Sep 2025 01:09:05 GMT" Domain: "portal.icdfa.edu.ng" Expires / Max-Age: "Session" HostOnly: true HttpOnly: true Last Accessed: "Wed, 03 Sep 2025 01:30:15 GMT" Path: "/" SameSite: "Strict" Secure: true Size: 143 |
| _ga | GA1.1.1377041... | .icdfa.edu.ng | / | Wed, 01 Sep 20... | |
| PHPSESSID | UVeVfZkc6KAVVgFY... portal.icdfa... | portal.icdfa... | / | Session | |

Parsed Value

```

PHPSESSID:Array
0:"UVeVfZkc6KAVVgFYz7yO2x%2CQIhjtC0fyuhTYnq7npY66GHkay4kogxZGuug7
1:rJJuQ%2CzVW%2CsfZehrvN34e4hUo2oTWbYUtOGFDjLct6KEepKpQpfGLb9X
wXVTmqj1"
1:"OlhjtC0fyuhTYnq7npY6...ay4kogxZGuug7fr6JluQ"
2:"zW"
3:"sfZehrvN34e4hUo2oTW...pQpfGLb9XwXVTmqj1"
length:4
__proto__:Array
  
```

Below are the two cookies i get after logging out and they did not look predictable

- UVeVfZkc6KAVVgFYz7yO2x%2CQIhjtC0fyuhTYnq7npY66GHkay4kogxZGuug71rJJuQ%2CzVW%2CsfZehrvN34e4hUo2oTWbYUtOGFDjLct6KEepKpQpfGLb9XwXVTmqj11
- uuPZC0mOrsdZLqB4jt-3qNT36zHL6CfBoyuUHOaF1CnuQcgvAYIIlcz7MOAM9jzDbGv1WNMaM7ZzFNoEh9Z7qgYSF0dbFGpIej29bEFZOnesneUiZX1O%2ChWzGeIAjEcT

| Name | Value | Domain | Path | Expires / Max-Age | Data |
|------------|--|-----------------|------|-------------------|---|
| _ga_L2B... | GS2.1.s175669... | .icdfa.edu.ng | / | Wed, 01 Sep 20... | PHPSESSID: "uuPZC0mOrsdZLqB...%2ChWzGeIAjEcT" Created: "Wed, 03 Sep 2025 01:09:05 GMT" Domain: "portal.icdfa.edu.ng" Expires / Max-Age: "Session" HostOnly: true HttpOnly: true Last Accessed: "Wed, 03 Sep 2025 01:36:43 GMT" Path: "/" SameSite: "Strict" Secure: true Size: 139 |
| _ga | GA1.1.1377041... | .icdfa.edu.ng | / | Wed, 01 Sep 20... | |
| PHPSESSID | uuPZC0mOrsdZLqB4jt-3qNT36zHL6CfBoyuUHOaF1... | portal.icdfa... | / | Session | |

Parsed Value

```

PHPSESSID:Object
uuPZC0mOrsdZLqB4jt-3qNT36zHL6CfBoyuUHOaF1...
__proto__:Object
  
```

Session Fixation: Test whether you can set a user's session ID before login and hijack the session.

- No you cant set a users session id before log in so you cant hijack the session
- The system returned “CSRF token validation failed”, preventing fixation attempts.
- The CSRF token mechanism is functioning as an additional security layer

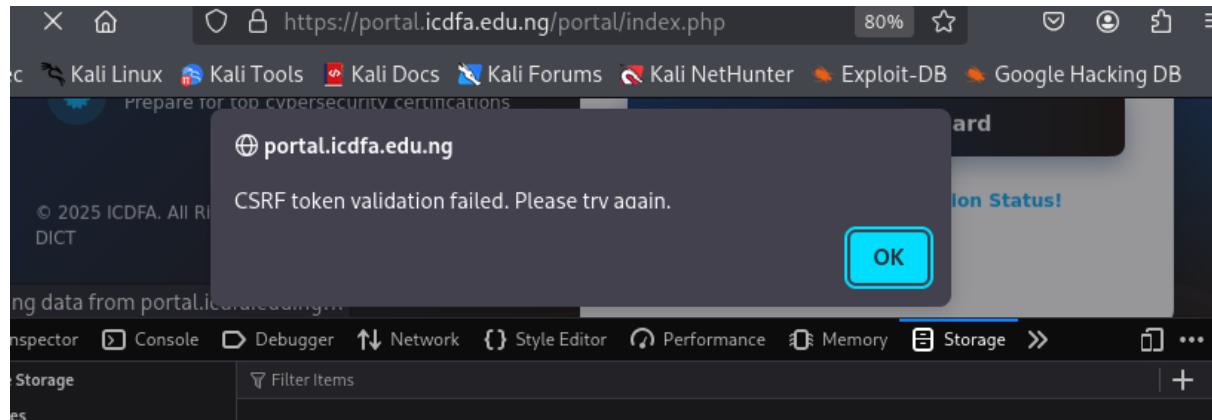
The screenshot shows the Network tab of a browser developer tools interface. The left sidebar lists storage types: Cache Storage, Cookies, Indexed DB, Local Storage, and Session Storage. The Cookies section is expanded, showing a table of cookies for the domain https://portal.icdfa.edu.ng. One cookie, PHPSESSID, is selected and expanded to show its details: Created: "Wed, 03 Sep 2025 01:51:03 GMT", Domain: "portal.icdfa.edu.ng", Expires / Max-Age: "Session", HostOnly: true, HttpOnly: true, Last Accessed: "Wed, 03 Sep 2025 01:55:33 GMT", Path: "/", SameSite: "None", Secure: true, Size: 16.

| Name | Value | Domain | Path | Expires / Max-Age | Data |
|------------|------------------|-----------------|------|-------------------|--|
| _ga_L2B... | GS2.1.s175669... | .icdfa.edu.ng | / | Wed, 01 Sep 20... | PHPSESSID:"fake123" |
| _ga | GA1.1377041... | .icdfa.edu.ng | / | Wed, 01 Sep 20... | Created:"Wed, 03 Sep 2025 01:51:03 GMT" Domain:"portal.icdfa.edu.ng" Expires / Max-Age:"Session" HostOnly:true HttpOnly:true Last Accessed:"Wed, 03 Sep 2025 01:55:33 GMT" Path:"/" SameSite:"None" Secure:true Size:16 |
| PHPSESS... | fake123 | portal.icdfa... | / | Session | |

The screenshot shows a browser window with the ICDFA website. The main content area displays a "Student Login" form with fields for Email and Password, and a "Forgot Password?" link. A modal dialog box is overlaid, showing the error message: "CSRF token validation failed. Please try again." There is an "OK" button on the dialog. At the bottom of the page, there is a footer with copyright information: "© 2025 ICDFA. All Rights Reserved. | Powered By ICDFA DICT".

Session Expiration: Confirm if the session terminates correctly after logout or inactivity.

- Yes the session terminates correctly after logout or inactivity
- Session reuse attempts also triggered “CSRF token validation failed.”
- The portal correctly invalidates old sessions, preventing hijacking via expired cookies.



Transport Security: Evaluate if session tokens are transmitted securely (via HTTPS).

- Using Wireshark, login and session-related traffic appeared as TLS 1.2/1.3 “Application Data” packets, indicating encryption.
- Credentials and tokens are not exposed over the network, preventing packet-sniffing attacks

| No. | Time | Source | Destination | Protocol | Length | Info |
|------|----------------|-----------------|-----------------|----------|--------|-------------------------|
| 1632 | 502.413756579 | 192.168.35.61 | 131.153.147.186 | TLSv1.3 | 1384 | Client Hello (SNI=port) |
| 1634 | 502.642896665 | 131.153.147.186 | 192.168.35.61 | TLSv1.3 | 310 | Server Hello, Change C |
| 1636 | 502.645115430 | 192.168.35.61 | 131.153.147.186 | TLSv1.3 | 130 | Change Cipher Spec, Ap |
| 1637 | 502.647848567 | 192.168.35.61 | 131.153.147.186 | TLSv1.3 | 762 | Application Data |
| 1638 | 502.946363875 | 131.153.147.186 | 192.168.35.61 | TLSv1.3 | 369 | Application Data |
| 1647 | 503.491757637 | 131.153.147.186 | 192.168.35.61 | TLSv1.3 | 460 | Application Data |
| 1649 | 505.516584048 | 131.153.147.186 | 192.168.35.61 | TLSv1.3 | 90 | Application Data |
| 1652 | 505.5222733997 | 192.168.35.61 | 131.153.147.186 | TLSv1.3 | 90 | Application Data |
| 1659 | 545.307359252 | 192.168.35.61 | 34.107.243.93 | TLSv1.3 | 105 | Application Data |
| 1660 | 545.307564884 | 192.168.35.61 | 151.101.65.91 | TLSv1.2 | 112 | Application Data |
| 1661 | 545.307884656 | 192.168.35.61 | 34.107.243.93 | TLSv1.3 | 90 | Application Data |
| 1663 | 545.308336081 | 192.168.35.61 | 151.101.65.91 | TLSv1.2 | 97 | Encrypted Alert |
| 1681 | 551.593221902 | 151.101.65.91 | 192.168.35.61 | TLSv1.2 | 97 | Encrypted Alert |
| 1693 | 559.682366106 | 192.168.35.61 | 34.107.243.93 | TLSv1.3 | 1354 | Client Hello (SNI=push) |
| 1695 | 559.856292470 | 34.107.243.93 | 192.168.35.61 | TLSv1.3 | 278 | Server Hello, Change C |
| 1697 | 559.857109938 | 192.168.35.61 | 34.107.243.93 | TLSv1.3 | 130 | Change Cipher Spec, Ap |
| 1698 | 559.857497371 | 192.168.35.61 | 34.107.243.93 | TLSv1.3 | 158 | Application Data |
| 1699 | 559.996556450 | 34.107.243.93 | 192.168.35.61 | TLSv1.3 | 684 | Application Data, Appl |
| 1700 | 559.996596535 | 34.107.243.93 | 192.168.35.61 | TLSv1.3 | 97 | Application Data |
| 1702 | 559.999481020 | 192.168.35.61 | 34.107.243.93 | TLSv1.3 | 97 | Application Data |
| 1706 | 560.142898383 | 192.168.35.61 | 34.107.243.93 | TLSv1.3 | 1312 | Client Hello (SNI=push) |
| 1709 | 560.365781728 | 34.107.243.93 | 192.168.35.61 | TLSv1.3 | 284 | Server Hello, Change C |
| 1711 | 560.377413028 | 192.168.35.61 | 34.107.243.93 | TLSv1.3 | 130 | Change Cipher Spec, Ap |
| 1712 | 560.378002749 | 192.168.35.61 | 34.107.243.93 | TLSv1.3 | 688 | Application Data |
| 1714 | 560.655430999 | 34.107.243.93 | 192.168.35.61 | TLSv1.3 | 898 | Application Data, Appl |
| 1715 | 560.673822774 | 192.168.35.61 | 34.107.243.93 | TLSv1.3 | 205 | Application Data |
| 1717 | 560.952016966 | 34.107.243.93 | 192.168.35.61 | TLSv1.3 | 258 | Application Data |

a52e4386fd1aa5a02cb8dcf3534f118b

The screenshot shows a browser developer tools interface with the Network tab selected. A single cookie entry is highlighted for the domain `https://portal.icdfa.edu.ng`. The cookie name is `PHPSESSID` and its value is `a52e4386fd1aa5a02cb8dcf3534f118b`. The cookie is set to expire at `Mon, 01 Sep 2025 02:28:01 GMT` and is marked as a session cookie (`Session`). The cookie has the following properties:

- Created: "Mon, 01 Sep 2025 02:28:01 GMT"
- Domain: "portal.icdfa.edu.ng"
- Expires / Max-Age: "Session"
- HostOnly: true
- HttpOnly: true
- Last Accessed: "Mon, 01 Sep 2025 02:50:23 GMT"
- Path: "/"
- SameSite: "None"
- Secure: true
- Size: 41

The screenshot shows a browser developer tools interface with the Network tab selected. A single cookie entry is highlighted for the domain `https://portal.icdfa.edu.ng`. The cookie name is `PHPSESSID` and its value is `92900b3882833ec0e3cf00fe041033a5`. The cookie is set to expire at `Mon, 01 Sep 2025 02:28:01 GMT` and is marked as a session cookie (`Session`). The cookie has the following properties:

- Created: "Mon, 01 Sep 2025 02:28:01 GMT"
- Domain: "portal.icdfa.edu.ng"
- Expires / Max-Age: "Session"
- HostOnly: true
- HttpOnly: true
- Last Accessed: "Mon, 01 Sep 2025 02:51:22 GMT"
- Path: "/"
- SameSite: "None"
- Secure: true
- Size: 41

4. Defense Mechanisms

- Identify Vulnerabilities: After testing, propose practical countermeasures to prevent session hijacking attacks. This may include enhancing session management, adjusting cookie attributes, or implementing encryption.

Identified Vulnerabilities

- Burp Suite interception showed username and password in plaintext inside the HTTPS request body.
- No evidence that cookies are set with HttpOnly, Secure, or SameSite.
- Lack of these attributes increases risk of theft via XSS or CSRF.
- While expired sessions returned “**CSRF token validation failed,**” there was no strict idle timeout confirmation.
- Long-lived sessions may remain active if a user forgets to log out.
- The portal does not appear to monitor for unusual session behavior (e.g., IP changes, concurrent logins).
- Attackers could exploit valid tokens without being detected.

Countermeasures to prevent session hijacking attacks.

- Enforce Strong TLS Security
- Enable HTTP Strict Transport Security (HSTS).
- Protect Login Credentials
- Implement Multi-Factor Authentication (MFA).
- Improve Session Timeout Policies
- Reauthentication required for sensitive actions.
- Monitor Active Sessions
- Detect anomalies (IP changes, multiple logins).
- Allow users to view/revoke active sessions.
- Alert or block suspicious concurrent logins.