CTF – Challenge Pack Documentation (with Solutions)

This repository contains two intentionally vulnerable web challenges designed for training and Capture-the-Flag exercises. The uploaded PHP files (`index.php` `loginSql.php` , `config.php` , `YellowStorm1701.php` , `EndOfLine.php`) are the authoritative implementation; this document describes the intended design and how to solve each challenge.

Challenge 1 — Login Bypass (SQL Injection)

Category: Injection / SQLi

Files

- `loginSql.php`
- `config.php`

Description

This challenge presents a simple login page with username and password fields. Credentials are validated against a database using an intentionally unsafe SQL statement constructed via direct concatenation:

```
$sql = "SELECT id FROM users
        WHERE username = '$username'
          AND password = '$password'
        LIMIT 1";
```

On successful authentication, `loginSql.php` sets a session flag and displays the first flag defined in `config.php` :

```
$FLAG = 'FLAG{YellowStorm1701.php}';
```

Objective

Obtain the first flag by bypassing authentication **without** knowing valid credentials.

How to Solve

1. Open `loginSql.php`.
2. Enter any username.
3. Use the following payload as password:

```
' OR ''='
```

4. Submit the form. Authentication is bypassed.
5. The page displays:

```
FLAG{YellowStorm1701.php}
```

6. The value is also the file name of Challenge 2.

Learning Points

- Unsafe SQL concatenation
- Boolean-based injection
- Importance of prepared statements

Challenge 2 — Role Escalation via Cookie Tampering

Category: Broken Access Control

Files

- `YellowStorm1701.php`

Description

`YellowStorm1701.php` reads the access level purely from cookies:

```php
$username = $_COOKIE['username'] ?? null;
$role     = $_COOKIE['role'] ?? null;
```

Authorization is then based on this value alone.

The second flag is defined as:

```
$FLAG2 = 'FLAG{EndOfLine.php}';
```

Objective

Escalate your role to administrator and retrieve the second flag.

How to Solve

1. Open browser DevTools.
2. Go to cookies for the challenge domain.
3. Set:

   ```
   username = anything
   role     = admin
   ```

4. Reload `YellowStorm1701.php`.
5. The page reveals:

   ```
   FLAG{EndOfLine.php}
   ```

6. Visit `EndOfLine.php` to finish the challenge.

Learning Points

- Never trust client-controlled state
- Authorization must be server-side
- Why signed tokens exist

End Page

`EndOfLine.php` confirms challenge completion with a static success message.
