Universidad Francisco de Vitoria
Centro de Documentación Europea
UFV Madrid

Pentesting

Héctor Ramírez López

Índice

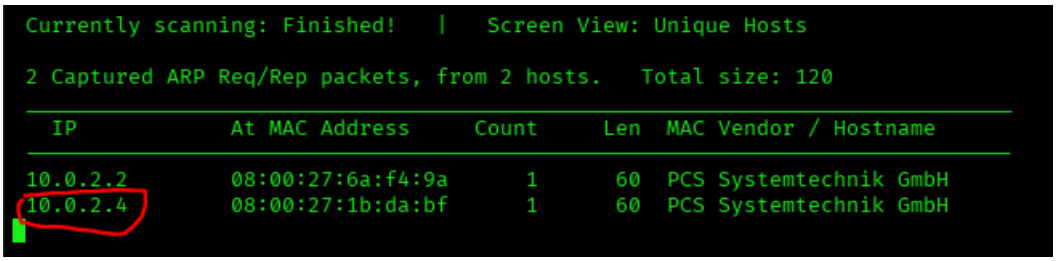
Informe de Auditoría Técnica: Máquina "Animetronic"

Fase 1: Reconocimiento y Enumeración de Servicios

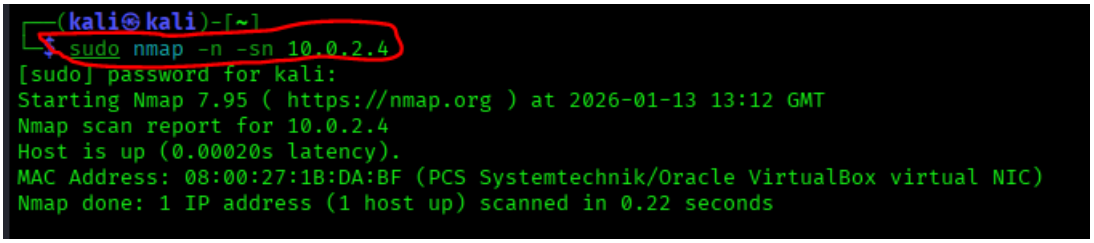
1.1. Descubrimiento de Activos (Host Discovery)

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).



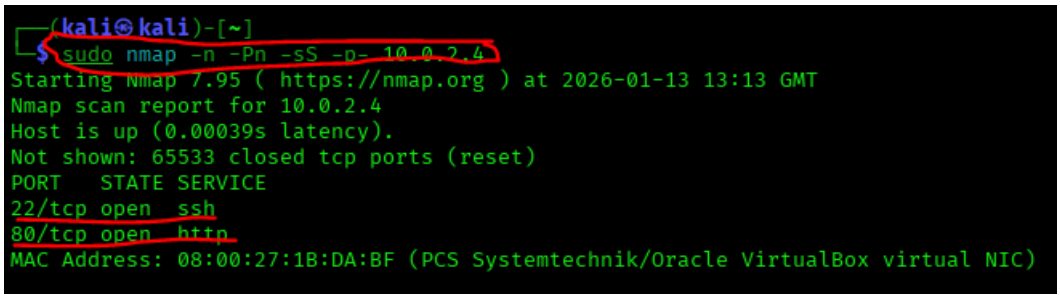
- **Confirmación de disponibilidad:** Para verificar que el host estaba operativo antes de realizar pruebas más intrusivas, se ejecutó un escaneo con Nmap.
    - **Comando:** sudo nmap -n -sn 10.0.2.4
    - **Resultado:** El host 10.0.2.4 se encuentra activo (Host is up) con una latencia de 0.00020s.



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.

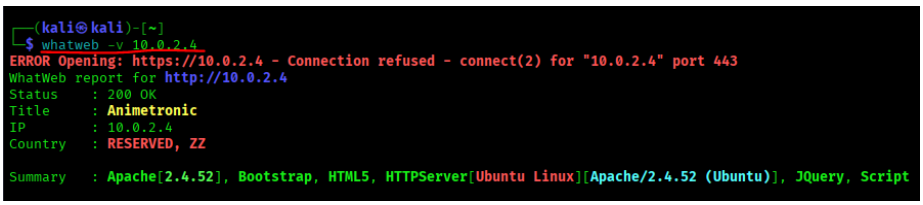
- **Comando ejecutado:** sudo nmap -n -Pn -sS -p- 10.0.2.4
    - -n: Sin resolución DNS.
    - -Pn: Omitir descubrimiento de host (asumir que está online).
    - -sS: TCP SYN Scan (sigiloso).
    - -p-: Escaneo de todos los puertos.
- **Resultados del escaneo:** Se detectaron dos puertos abiertos:
 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



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".



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:

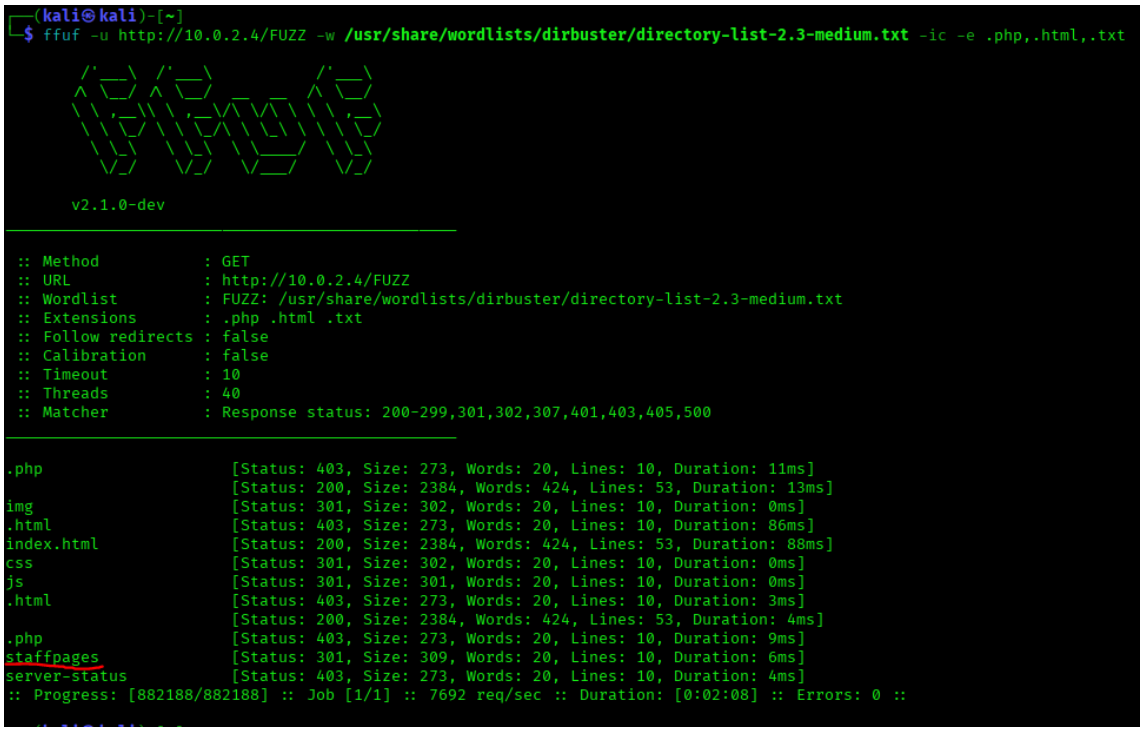
Bash

```
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 ::

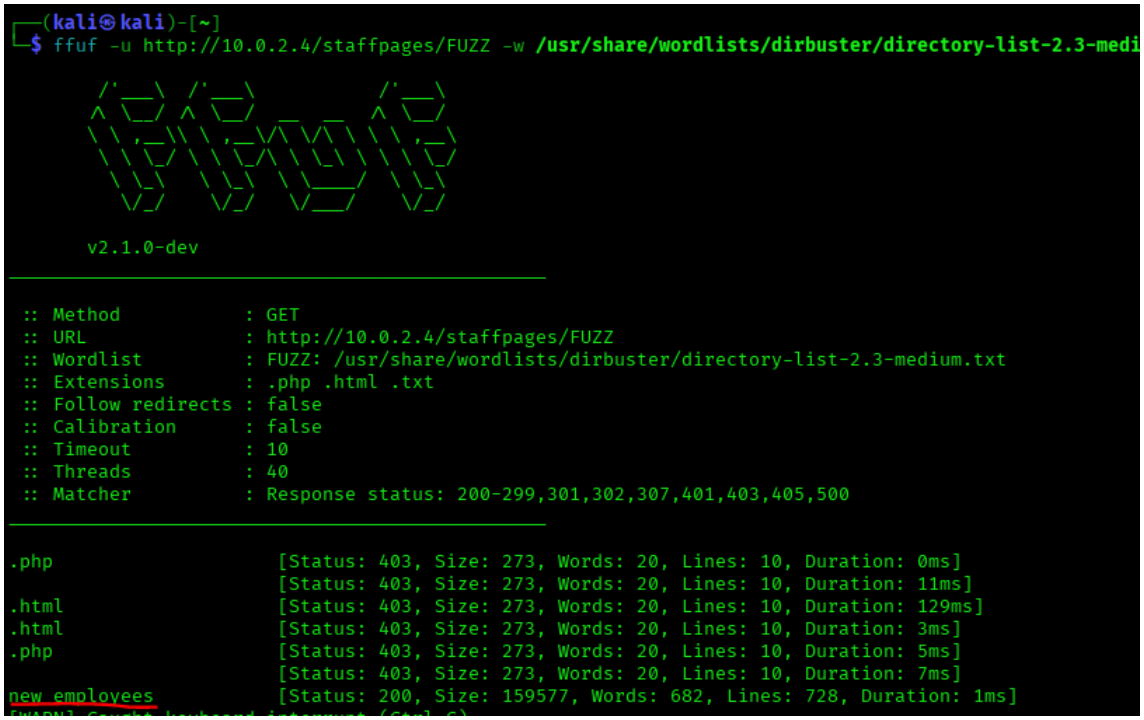
2.2. Enumeración de Subdirectorios

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.

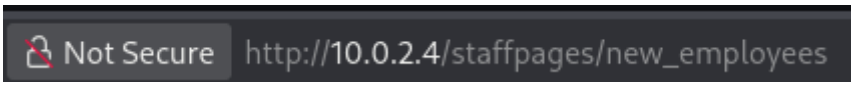


Fase 3: Adquisición de Evidencias Digitales

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.

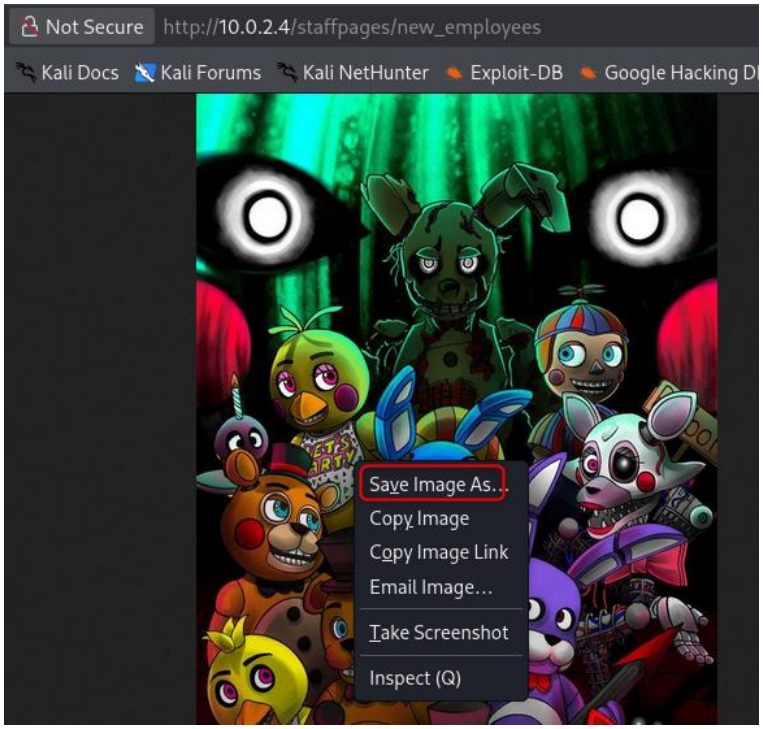
- **URL Objetivo:** http://10.0.2.4/staffpages/new_employees
- **Observación:** La URL no cargaba una página HTML estándar, sino que renderizaba directamente un archivo de imagen (JPEG/PNG).



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.



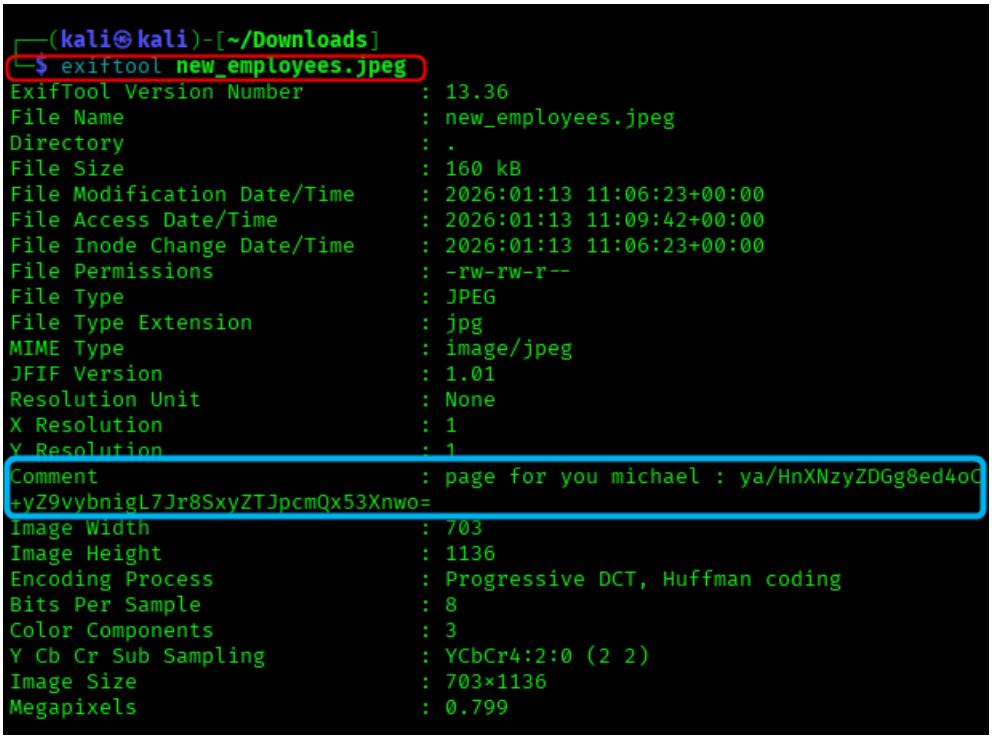
Fase 4: Análisis Forense y Esteganografía

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="



4.2. Decodificación de Datos

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

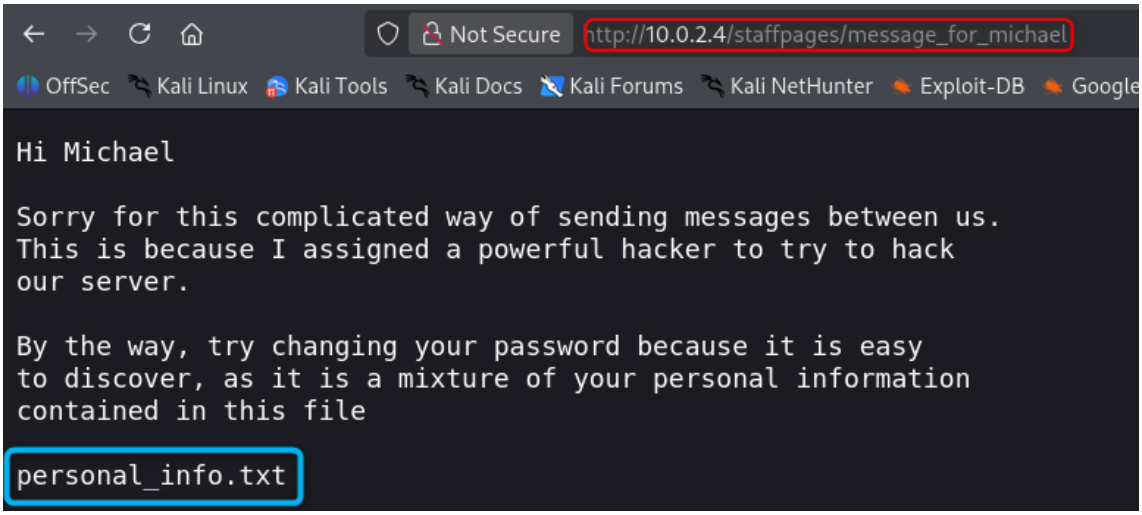
- **Comando:** echo "..." | base64 -d.
- **Resultado de la decodificación:** Se obtuvo la cadena: leahcim_rof_egassem.
- **Análisis del resultado:** La cadena es un texto invertido . Al leerla de derecha a izquierda, se revela una ruta válida: message_for_michael.

Fase 5: Recolección de Información (OSINT Interno)

5.1. Acceso a Comunicaciones Internas

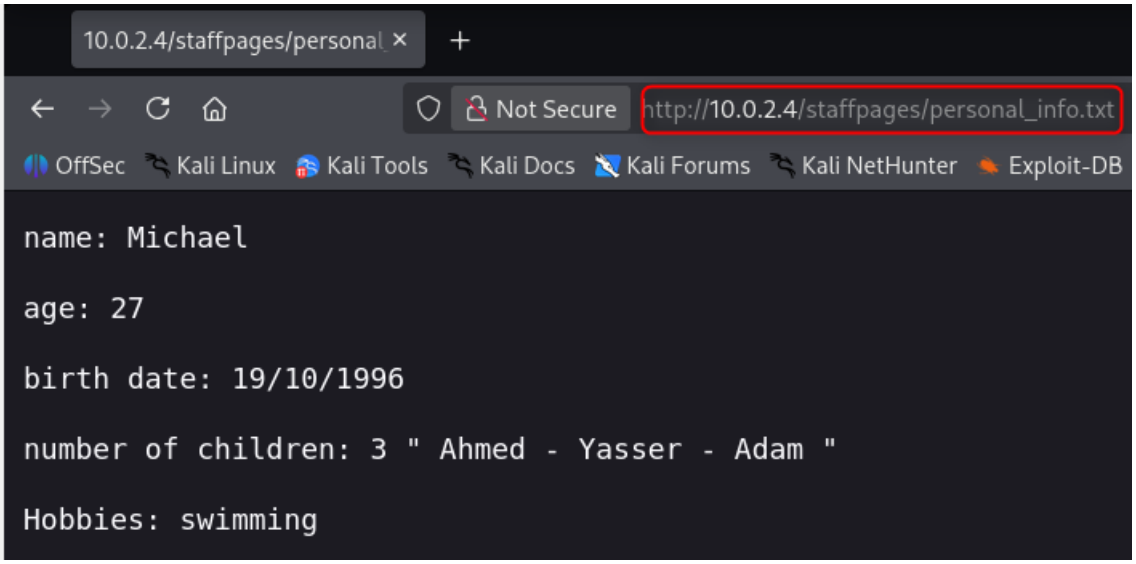
Utilizando la ruta descifrada, se navegó a la URL http://10.0.2.4/staffpages/message_for_michael.

- **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.



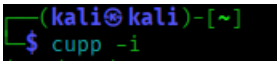
Datos extraídos del objetivo (Michael):

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

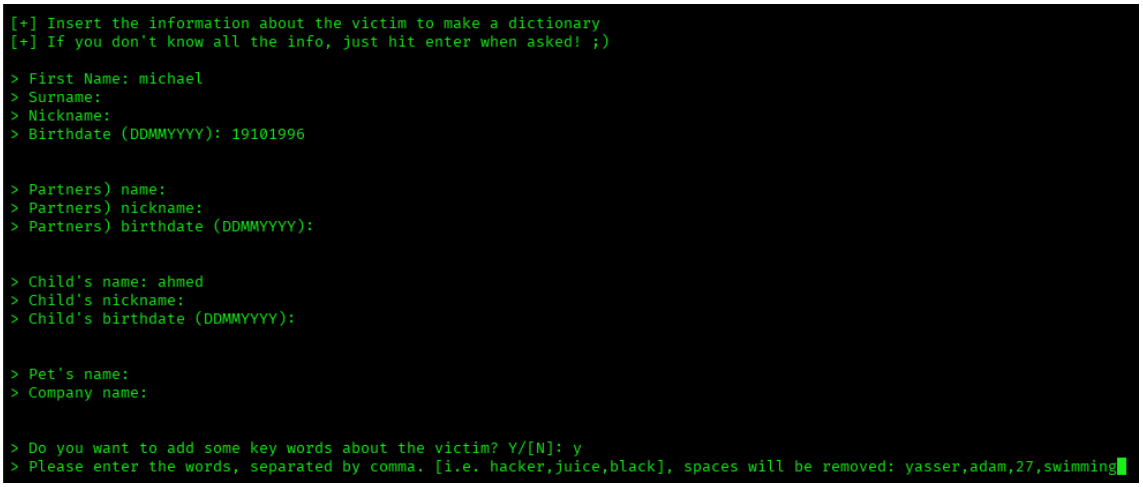
Fase 6: Generación de Diccionario Personalizado

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).



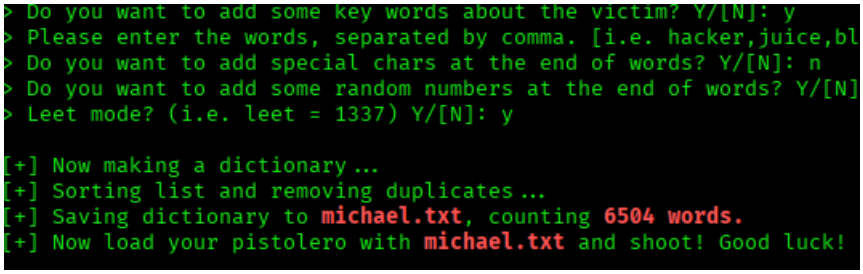
- **Herramienta:** cupp -i (Modo interactivo).
- **Datos introducidos (Input):** Se alimentó al algoritmo con los datos personales extraídos del servidor:
    - **Nombre:** michael
    - **Fecha de nacimiento:** 19101996
    - **Nombre de hijo:** ahmed
    - **Palabras clave adicionales:** yasser, adam,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



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 conteniendo un total de **6504 posibles contraseñas** únicas basadas en la vida personal del objetivo.



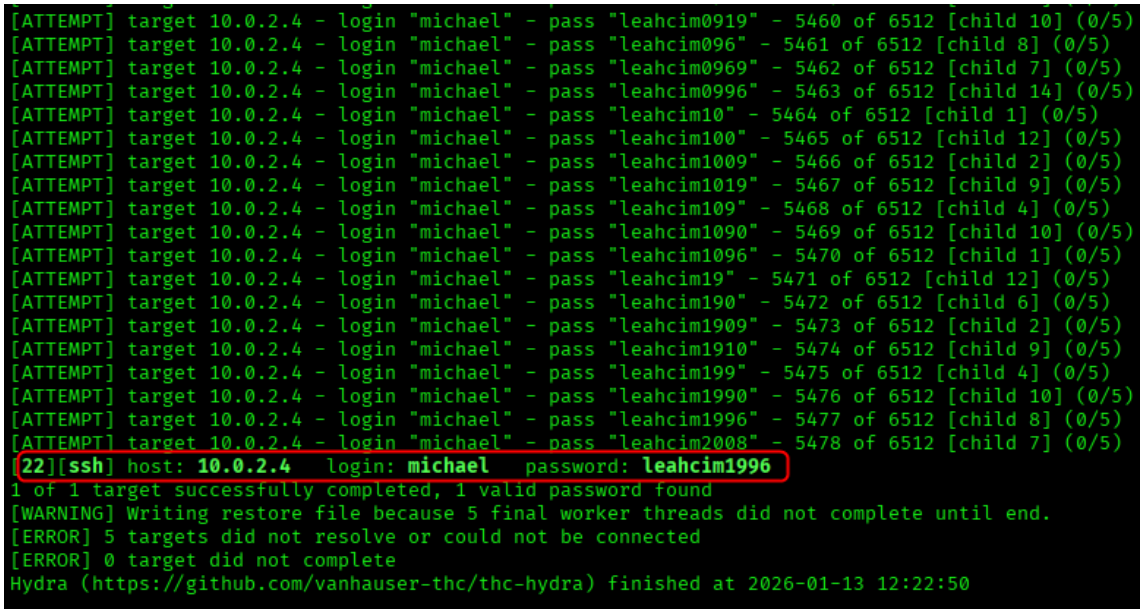
Fase 7: Ataque de Fuerza Bruta

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.txt -V -T 15 -l -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.

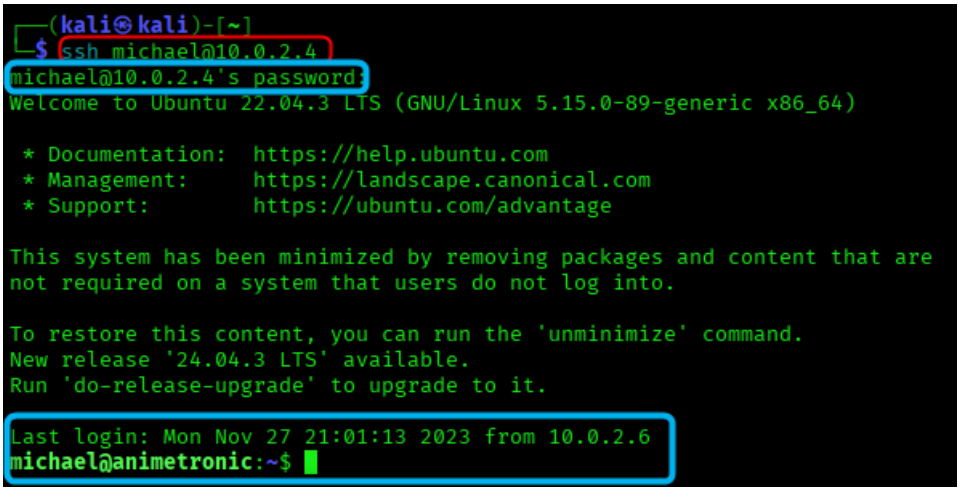


Fase 8: Acceso Inicial

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)**.



Fase 9: Enumeración Post-Explotación

9.1. Verificación de Identidad y Privilegios

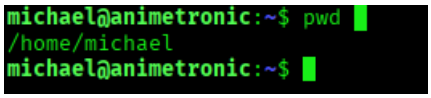
Una vez dentro del sistema, el primer paso es verificar el contexto del usuario comprometido para entender el nivel de privilegios actual.

- **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.

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.

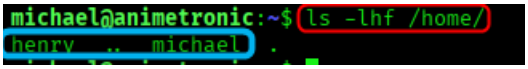
- **Comando:** pwd
- **Resultado:** /home/Michael



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.



- **Comando ejecutado:** ls -lhf /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/

- **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

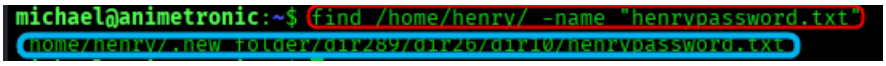
Siguiendo las pistas encontradas en el directorio del usuario henry derivadas del análisis de Note.txt, se identificó una cadena de texto en formato Base64 que ocultaba información relevante.

- **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.

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.



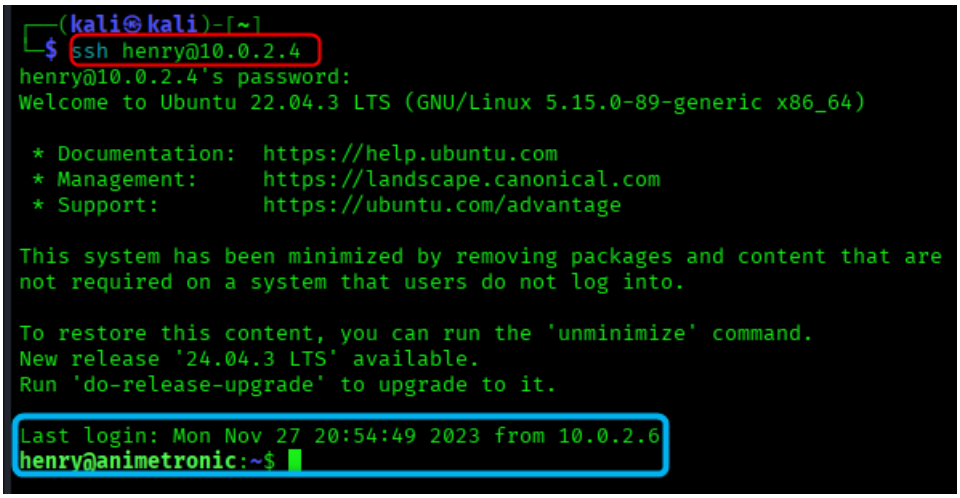
Cuando entramos a esa ruta encontramos la contraseña del usuario Henry.

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.



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.

Fase 13: Escalada de Privilegios (Root)

13.1. Análisis de Permisos Sudo

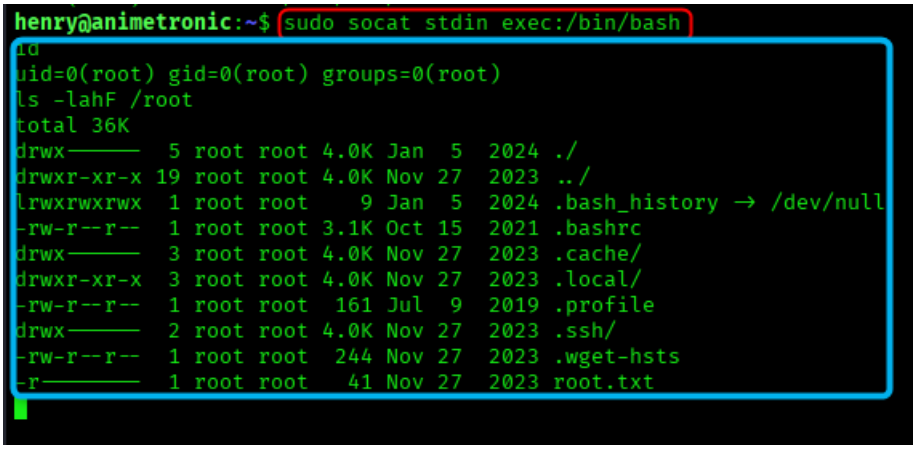
Se ejecutó el comando para listar qué binarios específicos puede ejecutar Henry como superusuario.

- **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.

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



- **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

RAMIREZ LOPEZ HECTOR - 53858461M

Firmado digitalmente por RAMIREZ LOPEZ HECTOR - 53858461M
Fecha: 2026.01.13 16:02:26 +01'00'