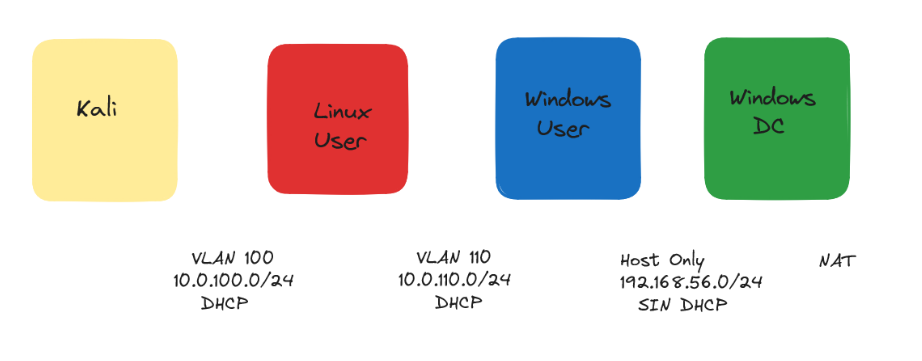


**SPRINGO 19**

**TEAM CHALLENGE**

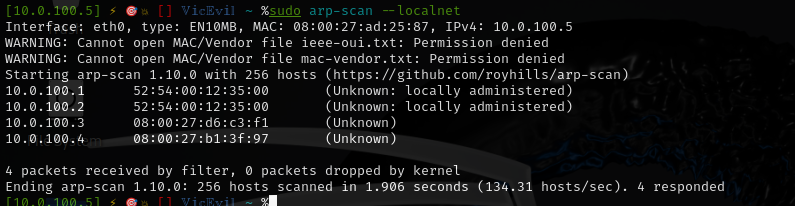
**RETO PIVOTING-AD**

El objetivo de este reto es conseguir pivotar entre las máquinas de la imagen de mas abajo, hasta llegar a obtener una conexión a la *“Windows* ***DC”***, desde tú máquina Kali aplicando las técnicas y tácticas para conseguir su explotación, siendo el objetivo final el acceso completo a esa máquina con privilegios elevados.

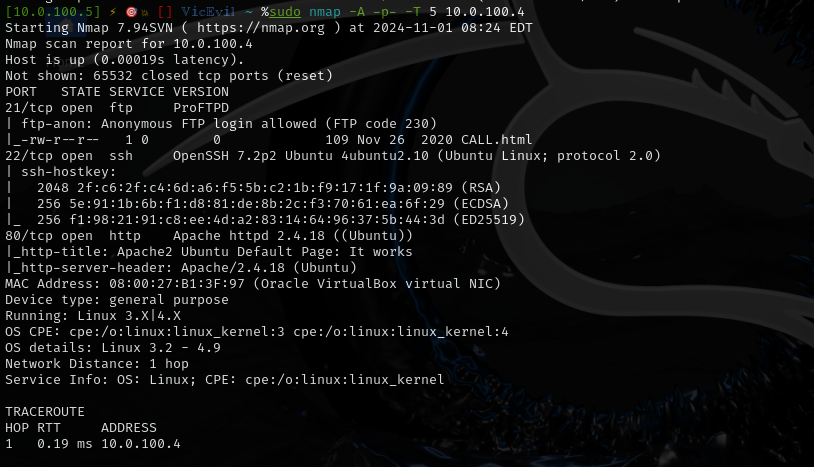


Para su realización se han llevado a cabo las siguientes gestiones:

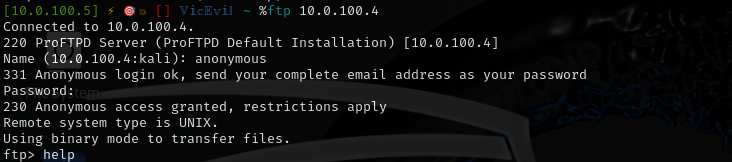
1. Una vez configurado el laboratorio, se realiza un escaneo de red, obteniendo la IP del primer adaptador de la máquina *“Linux User”* siendo la 10.0.100.4:

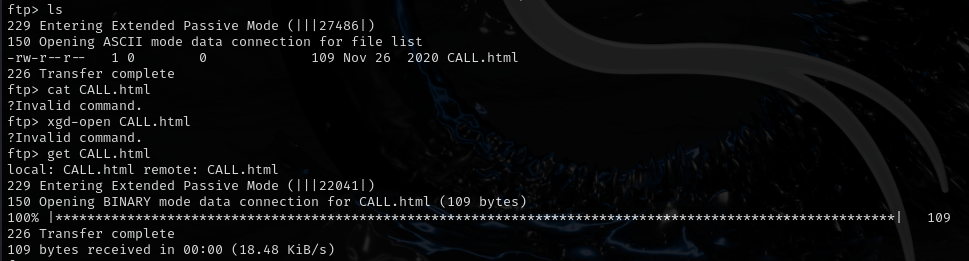


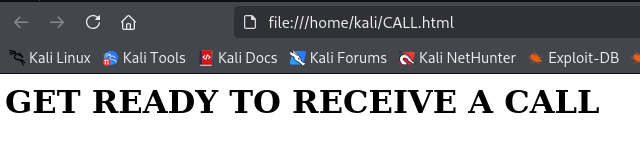
1. Se realiza un escaner profundo de la máquina objetivo, obteniendo como información relevante que esta permitida la conexión al servidor FTP con credenciales ***“anonymous:230”,*** no obstante, se verifica el resto de información:



1. Se procede a la conexión al servidor FTP, siendo positiva, encontrando un archivo ***“CALL.html”,*** que una vez transferido a la Kali, se procede a su apertura destacando que en el *head* del archivo html, tiene un titulo llamado ***“onion”*** y una frase en ingles ***“Get ready to receive a call”***.







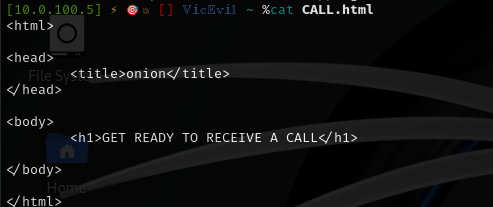


Imagen 1.- Detalle de la conexión al servidor FTP, la transferencia y apertura del archivo “CALL”

1. Se comprueba el servidor web con la herramienta gobuster, dirsearch y otras, así como con distintos diccionarios, encontrando únicamente un directorio interesantes: **/files**

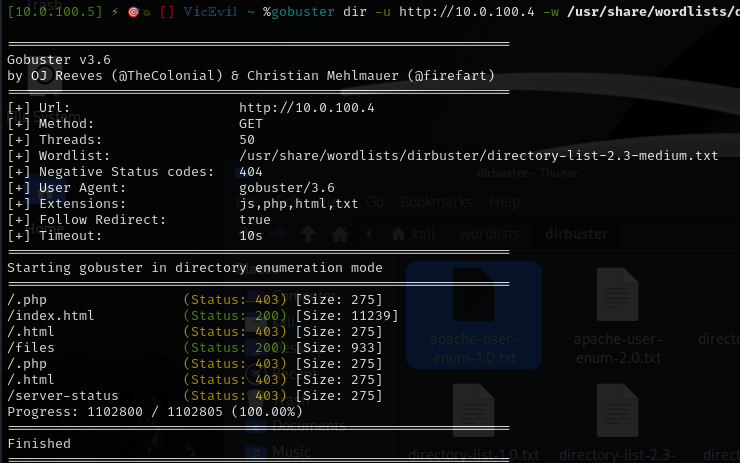
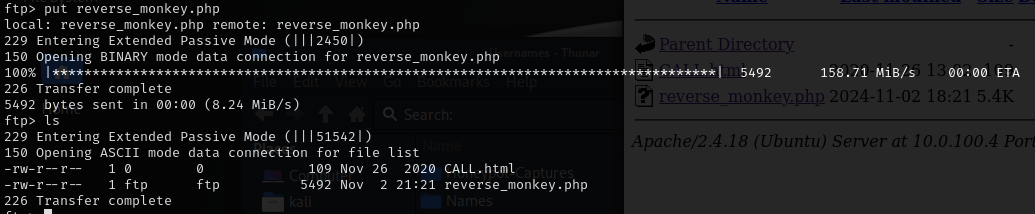
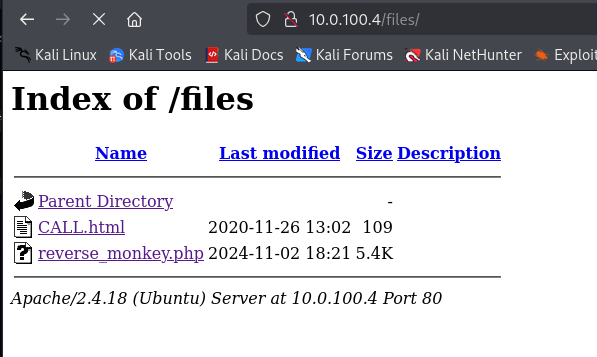
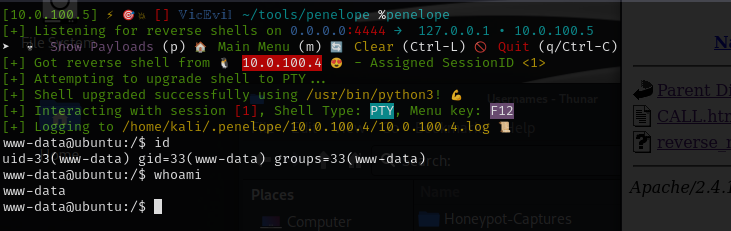


Imagen 2.- Resultado de la búsqueda de directorios con una de las aplicaciones usadas.

1. Se comprueba que al subir archivos al servidor FTP, estos se muestran en el directorio web /Files, por lo que procedo a subir una ***reverse shell*** , siendo ejecutada desde el navegador con resultado positivo, consiguiendo una shell con escasos privilegios:



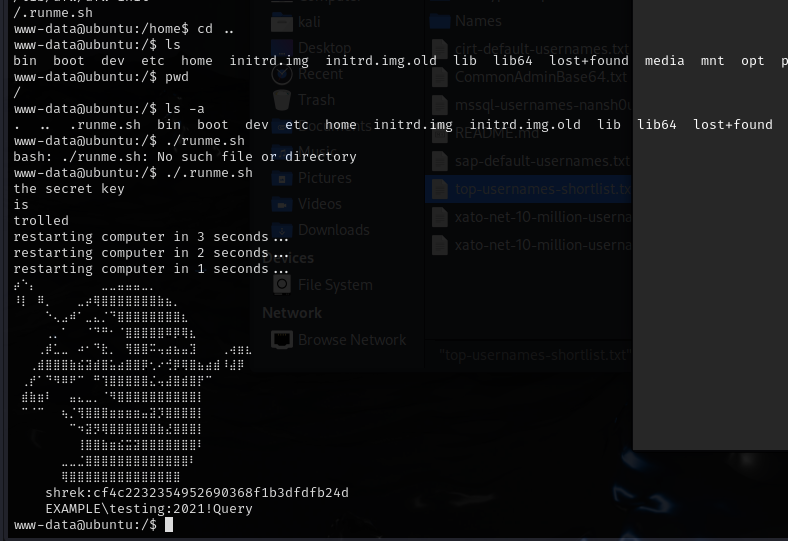




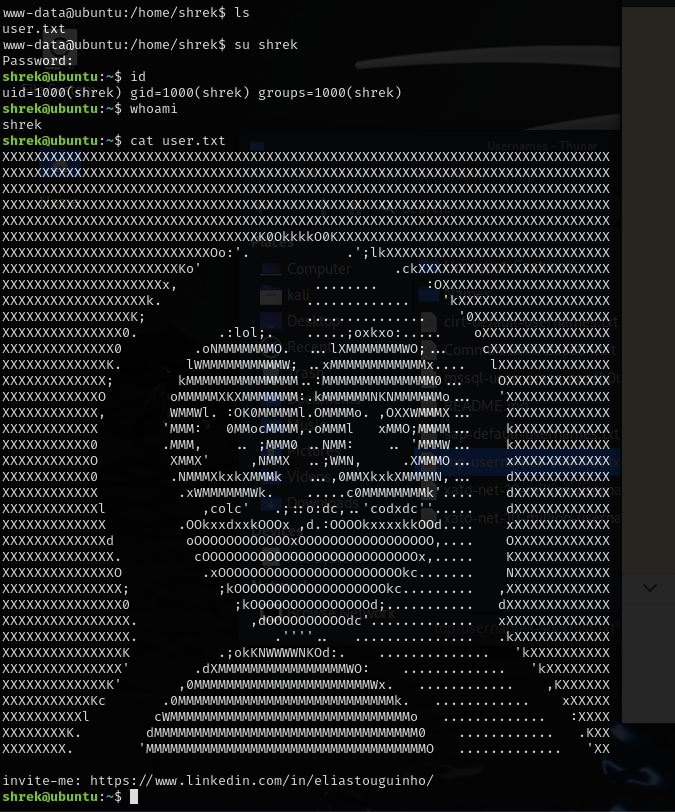
1. Se procede a investigar por diferentes directorios de la máquina “Linux-User”, encontrando información sobre un archivo ***“runme.sh”***, por lo que se realiza la búsqueda del archivo usando el comando “find”, siendo encontrado al final de una larga lista, el cual esta oculto. Una vez en el directorio donde se halla el archivo, se ejecuta consiguiendo información importante:

- ***shrek:cf4c2232354952690368f1b3dfdfb24d***, siendo este ultimo un hash MD5, procediendo a su descifrado correspondiendo a ***“onion”*** *(titulo archivo HTML)*

***- EXAMPLE\testing:2021!Query***.- Parece corresponder a un dominio, usuario y una contraseña de Windows, por la estructura y la barra “\”.

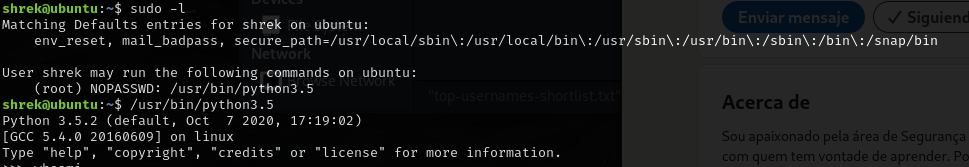
  


1. Con las credenciales ***“shrek:onion”*** se realiza conexión en el sistema, con resultado positivo, consiguiendo un usuario con privilegios mas altos que el anterior. Una vez logueados, nos dirigimos a uno de los directorios investigados: ***/home/shrek***, procediendo a aperturar el archivo **user.txt** (antes no me dejaba),el cual, contiene una imagen de un pinguino , que te remite a un enlace de linkedin:

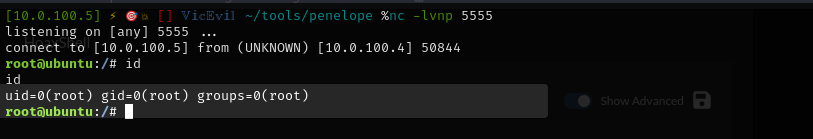


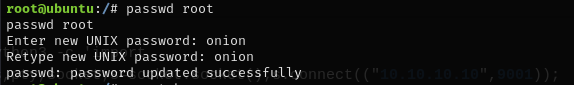


1. Se procede a ejecutar el comando “sudo -l”, donde se observa que podemos ejecutar la aplicación python3.5 en modo **root sin contraseña**, procediendo a su ejecución junto un shell de python, consiguiendo acceso root, procediendo al cambio de contraseña:

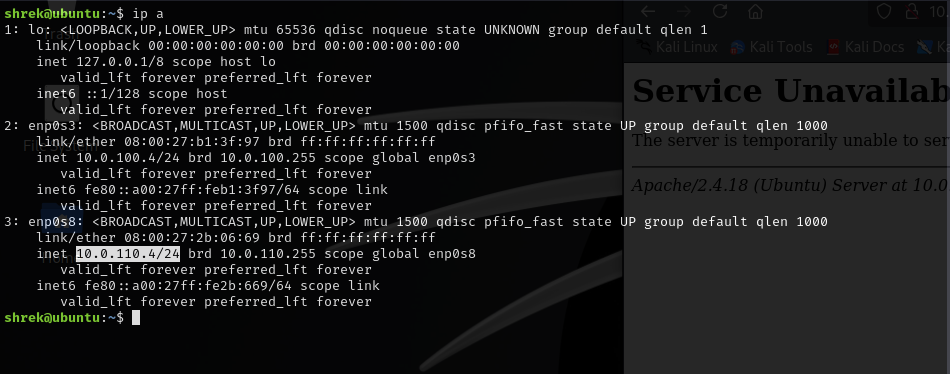




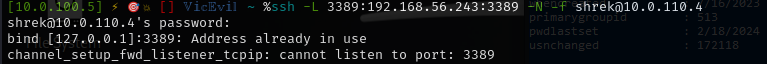


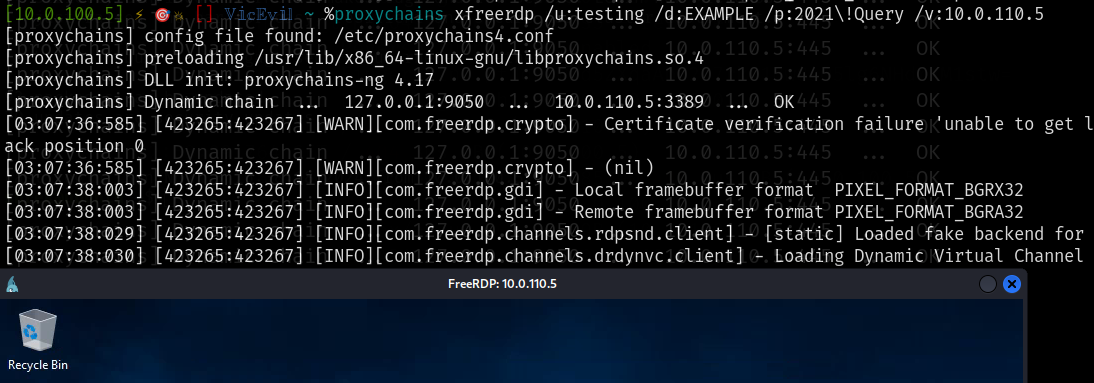
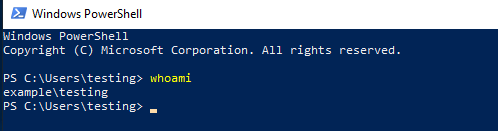


1. Se realiza una conexión mediante un **túnel dinámico** a través el usuario shrek, consiguiendo la IP del segundo adaptador de la maquina Linux User: **10.110.0..4**

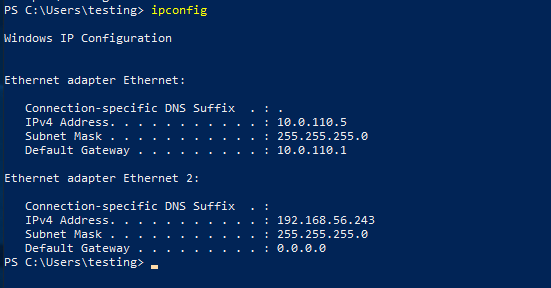


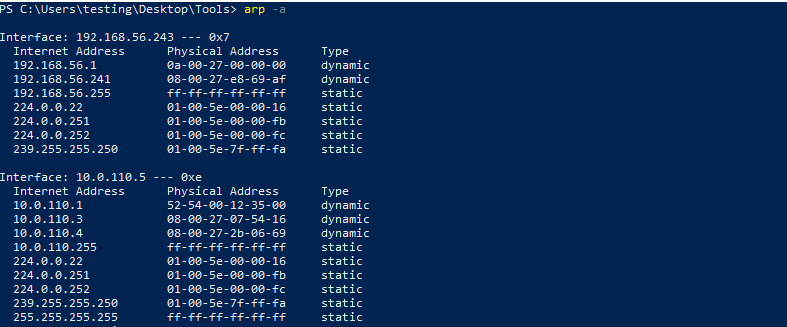
1. Con la contraseña extraída anteriormente de Windows, y a traves de un túnel local establecido a través de la maquina “Linux User” por el puerto 3389, accedemos a la maquina Windows User:



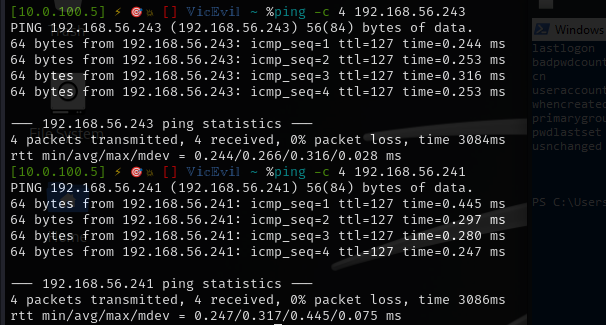
  
  


1. Una vez en la maquina **“Windows User”,** se consigue la IP del segundo adaptador y mediante el comando “*arp -a*”, la IP del adaptador de la máquina “Windows DC”:





1. Se realiza *“ping”* **desde la Kali** para comprobar si tenemos **conexión con las máquinas Windows** con resultado positivo:



1. Una vez ejecutada la herramienta **PowerView** en “***window user***”,se procede a enumerar la información necesaria para realizar el ataque:

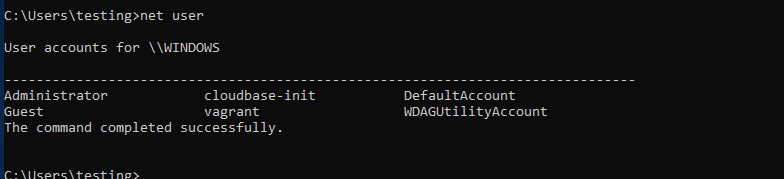


Imagen 3.- Usuarios del sistema

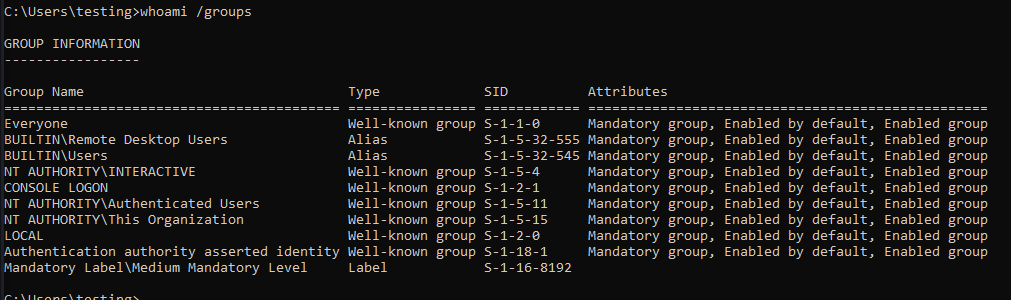


Imagen 4.- Grupos del sistema

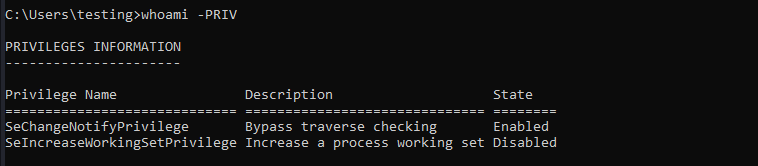


Imagen 5.- Privilegios de nuestro usuario en el sistema[[1]](#footnote-0)

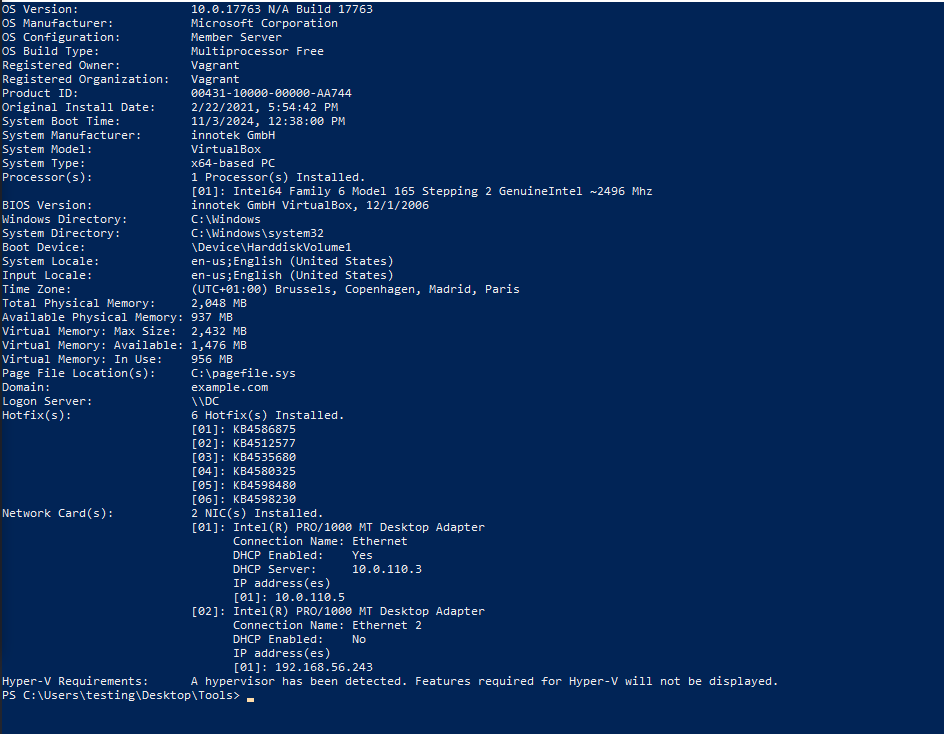


Imagen 6.- ejecución de **systeminfo** aportando información completa del sistema a traves de PS

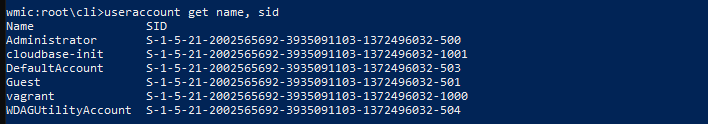
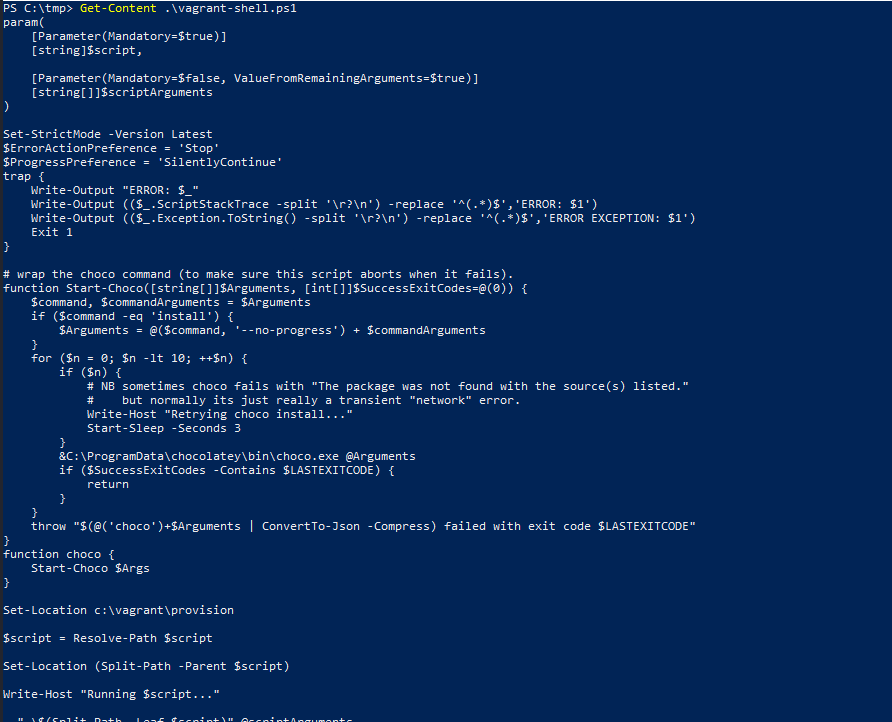


Imagen 7.- SID[[2]](#footnote-1) de los usuarios del sistema

1. En el directorio C:\tmp de la máquina “Windows User”, se localiza un archivo bastante interesante, siendo un script, el cual,utilizando la función *“param”* acepta un parámetro *“$script”*, que se resuelve y luego se ejecuta. También utiliza una función llamada *“choco”*, que parece estar relacionada con el gestor de paquetes Chocolatey (choco.exe). Sin embargo, lo más **relevante** es que el **script finalmente ejecuta cualquier script que se le pase como argumento,** llegando a ejecutar una reverse shell, comentando la linea *“#set location C:\vagrant\provision”,* consiguiendo únicamente una shell del mismo usuario (testing), por lo que se deja esta vía.



1. Se procede a buscar archivos que tengan el ejecutable en lugares pocos habituales, o que estuvieran mal configurados para comprobar si alguno es posible su explotación, usando los distintos métodos existentes, no teniendo permisos plenos (F) sobre ningún archivo y además todas las rutas estas escapadas, por lo que también se abandona esta vía.

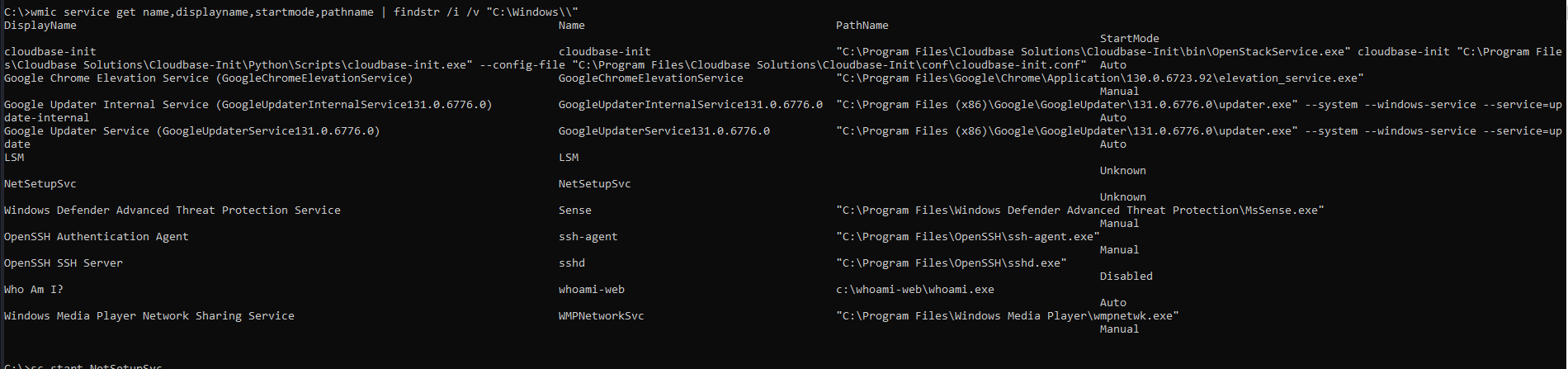
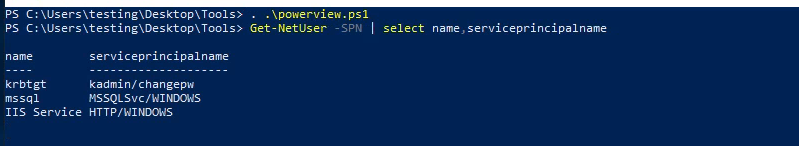


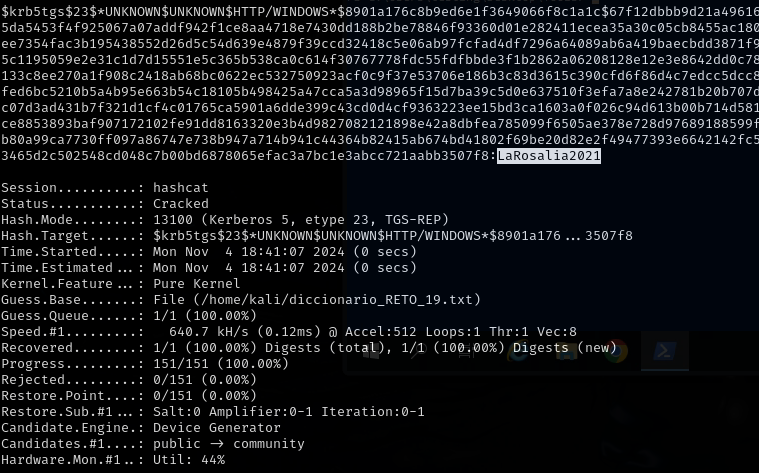
Imagen 8.- Ejecución del comando con la herramienta wmic[[3]](#footnote-2)

1. Mediante la herramienta “PowerView[[4]](#footnote-3)” nuevamente, consultando los usuarios del dominio y su nombre principal en el entorno AD, con la finalidad de ejecutar un ticket TGS al KDC y poder autenticarnos en alguno de los servicios, descifrando primeramente su hash NTLM:



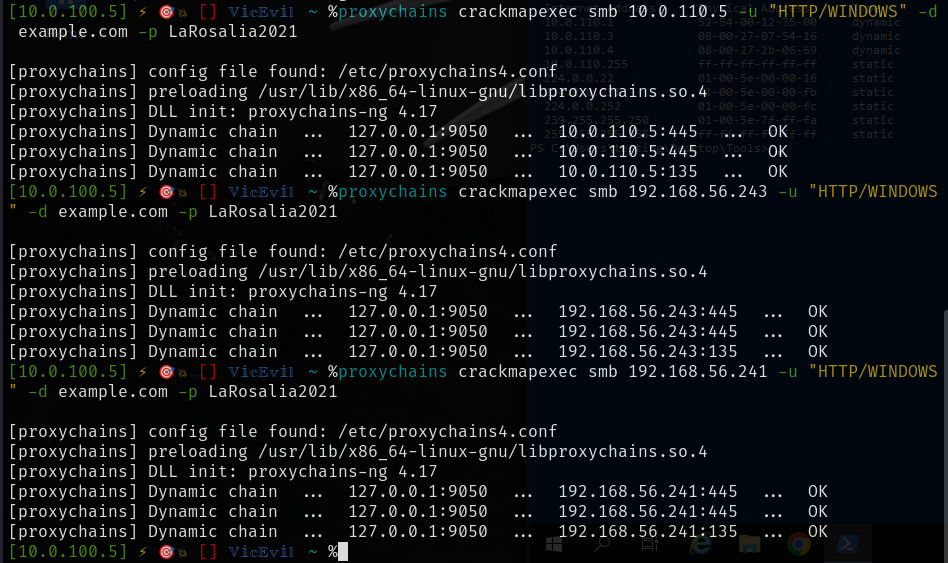
1. Se comprueban los tres servicios del punto 16, extrayendo su hash Kerberos para intentar descifrarlo mediante la herramienta hashcat, obteniendo resultados positivos únicamente en el servicio **“iis\_service”** (HTTP/WINDOWS), consiguiendo una la contraseña : **LaRosalia2021**



**Imagen 9.- Hash Kerberos del servicio “HTTP\WINDOWS” usando PShell

*Imagen 10.- resultado del descifrado con la herramienta Hashcat*

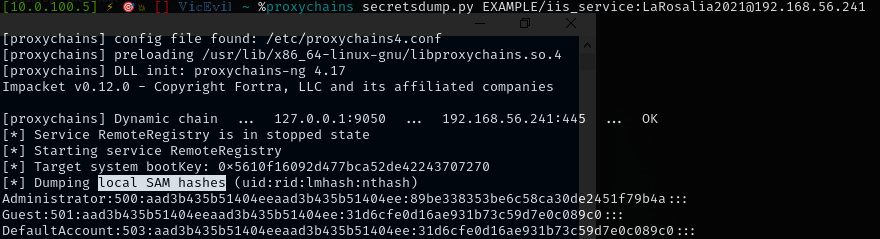
1. Mediante el uso de la herramienta **crackmapexec***[[5]](#footnote-4)*, usando el protocolo SMB, con el usuario del servicio, el dominio y la contraseña conseguida, se intenta conseguir hashes de usuarios del sistema que nos permita conseguir mayores privilegios en el sistema AD, tanto en Windows User y en Windows DC, con resultado infructuoso, conecta el proxy correctamente, pero **no devuelve resultado**:



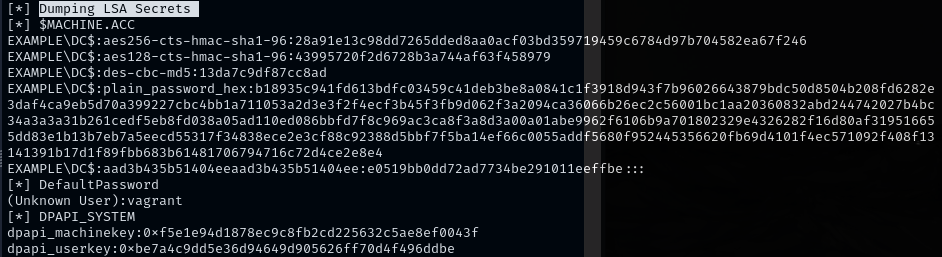
1. Por lo expuesto en el punto anterior, se busca por Internet alternativas para conseguir el resultado de este ejercicio, encontrando una colección de herramientas de Python llamada **“Impacket”,** la cual, permite realizar diversas tareas pentesting en redes Windows, pudiendo interactuar con protocolos de red como SMB, RDP, LDAP, Kerberos, entre otros, facilitando acciones como la extracción de hashes, ejecución remota de comandos, y ataques basados en autenticación, estando entre las mas conocidas: **secretsdump.py[[6]](#footnote-5)** y **wmiexec.py[[7]](#footnote-6).**
2. Se procede a la búsqueda de hashes de contraseñas y secretos del sistema Windows, ejecutando para ello la herramienta secretsdump.py, la cual aporta mucha información que se puede clasificar en:

- **Hashes locales - SAM.-** Permiten la autenticación y pueden ser utilizados para *“ataques de pass-the-hash(PTH)”,* si el hash corresponde a un usuario con privilegios.

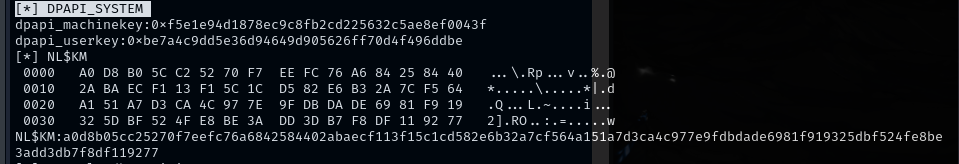
- La ***“bootKey”*** es una clave de encriptación esencial del sistema Windows, utilizada para proteger datos en el registro, incluyendo los hashes de usuario en SAM y LSA.



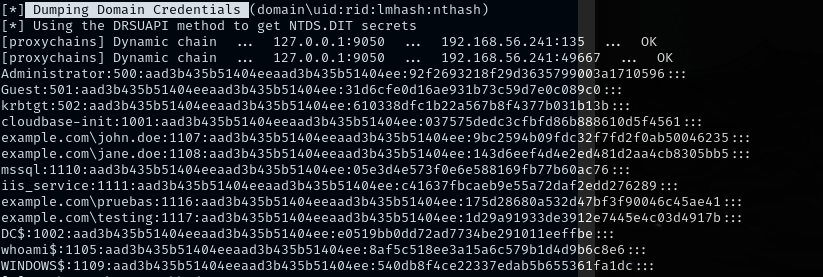
- **Dumping LSA Secrets.-** Se revelan datos de autenticación en la cuenta de seguridad local (Local Security Authority), como claves y contraseñas de servicios (Información de cuentas del sistema).



- **DPAPI\_SYSTEM.**- Contiene la **clave maestra de cifrado del sistema,** permitiendo acceder a los datos cifrados a nivel del sistema Windows, pudiendo d**esencriptar datos co**mo: contraseñas de servicio, secretos de red y otros datos que usan DPAPI para su protección, que si no están bien protegidos, se lograría acceder a credenciales de usuarios DPAPI y/o movimientos laterales dentro de la red.



- **Dumping Domain Credentials.-** Aquí se muestran los hashes de NTLM y claves de Kerberos para cada usuario del dominio.



1. Con la información obtenida en el punto anterior, se va a proceder a realziar un ataque pass-the-hash (PTH)[[8]](#footnote-7) , usando para ello la parte del hash NT del hash NTLM del administrador: “Administrator:500:aad3b435b51404eeaad3b435b51404ee:**92f2693218f29d3635799003a1710596**:::”, el cual, mediante la aplicación de Impacket, llamada **wmiexec.py** , que permite autenticarse usando Pass-the-Hash, ofreciendo una conexión remota al sistema objetivo, consiguiendo una shell de la maquina “Windows DC” con máximos privilegios (example\administrator), consiguiendo la resolución de este Team Challenge.

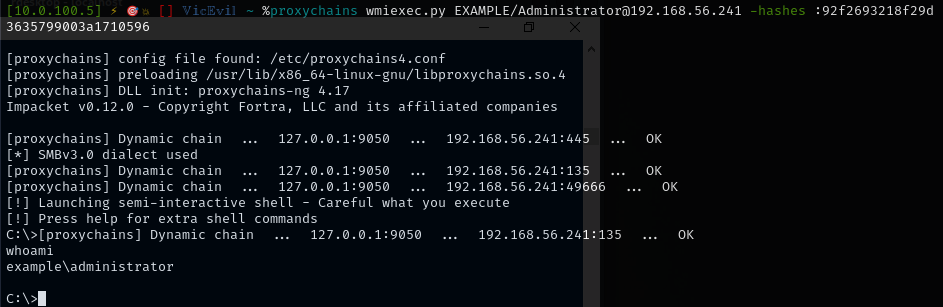


Imagen 11.- Shell remota de la máquina Windows User con privilegios de Administrador

1. · **SeChangeNotifyPrivilege (Bypass traverse checking).-** Este privilegio está habilitado, lo que permite al usuario eludir las verificaciones de acceso para atravesar directorios en el sistema de archivos, siendo un privilegio común, que no proporciona privilegios administrativos. [↑](#footnote-ref-0)
2. **Security Identifier o Identificador de Seguridad.-** Identificador único asignado a cada objeto de seguridad en el sistema operativo, como usuarios, grupos y dispositivos, permitiendo que Windows administre permisos y seguridad de forma precisa, ya que cada objeto tiene su propio SID que permanece constante incluso si el nombre del objeto cambia. [↑](#footnote-ref-1)
3. **Windows Management Instrumentation Command-line**, es una herramienta de línea de comandos de Windows que permite interactuar con la WMI (Windows Management Instrumentation) para consultar y gestionar información del sistema. [↑](#footnote-ref-2)
4. herramienta cuya principal función es **enumerar y mapear entornos de Active Directory (AD)**, proporcionando información detallada sobre usuarios, grupos, permisos, políticas y configuraciones del dominio. [↑](#footnote-ref-3)
5. **CrackMapExec** (CME) es una herramienta de post-explotación utilizada para realizar auditarías en redes y sistemas Windows, siendo su principal objetivo facilitar la interacción con redes basadas en Active Directory, utilizando diferentes protocolos como SMB, RDP, WinRM, y LDAP. [↑](#footnote-ref-4)
6. **secretsdump.py**.- Extrae hashes de contraseñas del sistema desde SAM, NTDS.dit y LSA, siendo ideal para obtener credenciales de usuarios y hashes de NTLM. [↑](#footnote-ref-5)
7. **wmiexec.py.**- Permite ejecutar comandos en sistemas remotos a través de WMI usando credenciales o hashes, siendo útil para ejecutar comandos en máquinas con privilegios administrativos. [↑](#footnote-ref-6)
8. **Pass-the-Hash (PtH)** es una técnica de ataque que permite a un atacante autenticarse en un sistema Windows remoto, usando el hash NTLM de la contraseña en lugar de la contraseña en texto claro. [↑](#footnote-ref-7)