Name: Kunal Goyal

Reg No.: 22BCY10053

Project Title: Who's Watching? (Simulating Man-in-the-Middle Attacks Over Public Wifi)

Tools Used: Ettercap, arpspoof, Wireshark, Python HTTP Server / Apache, Victim Machine (Windows/Linux) & Kali Linux.

Project Title

Who's Watching? Simulating Man-in-the-Middle Attacks Over Public Wi-Fi

Lab Environment Setup:

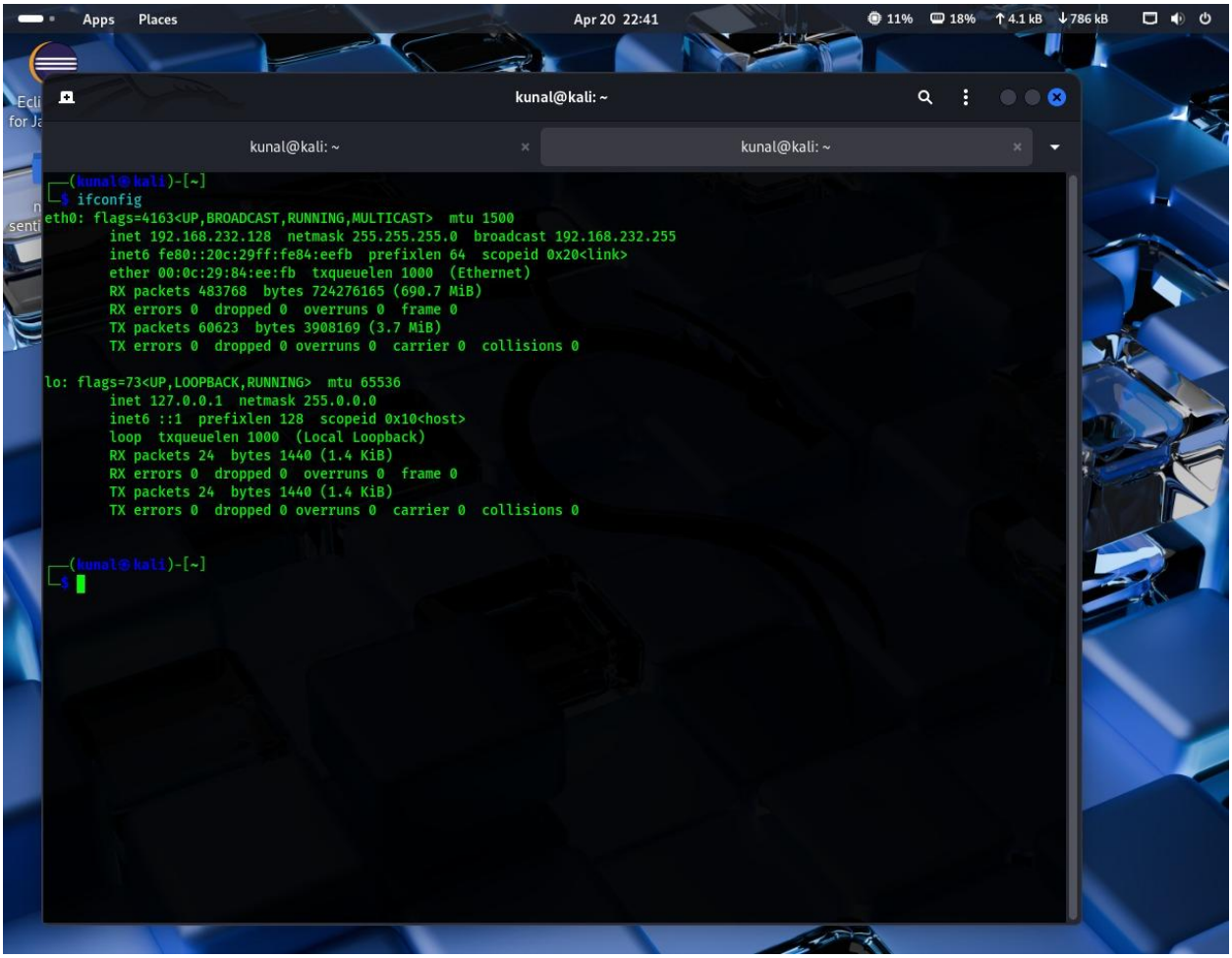
- **Host Machine: Windows 11**
- **VM1 (Kali Linux): Attacker machine**
- **VM2 (Windows 10): Victim machine**
- **Network Mode: Bridged Adapter - Host-only**
- **Attacker IP: 192.168.232.128**
- **Victim IP: 192.168.232.129**
- **Gateway IP: 192.168.232.2**
- **Key Tools Installed: Ettercap, arpspoof, Wireshark, Python3 HTTP Server**

Step-by-Step Implementation:

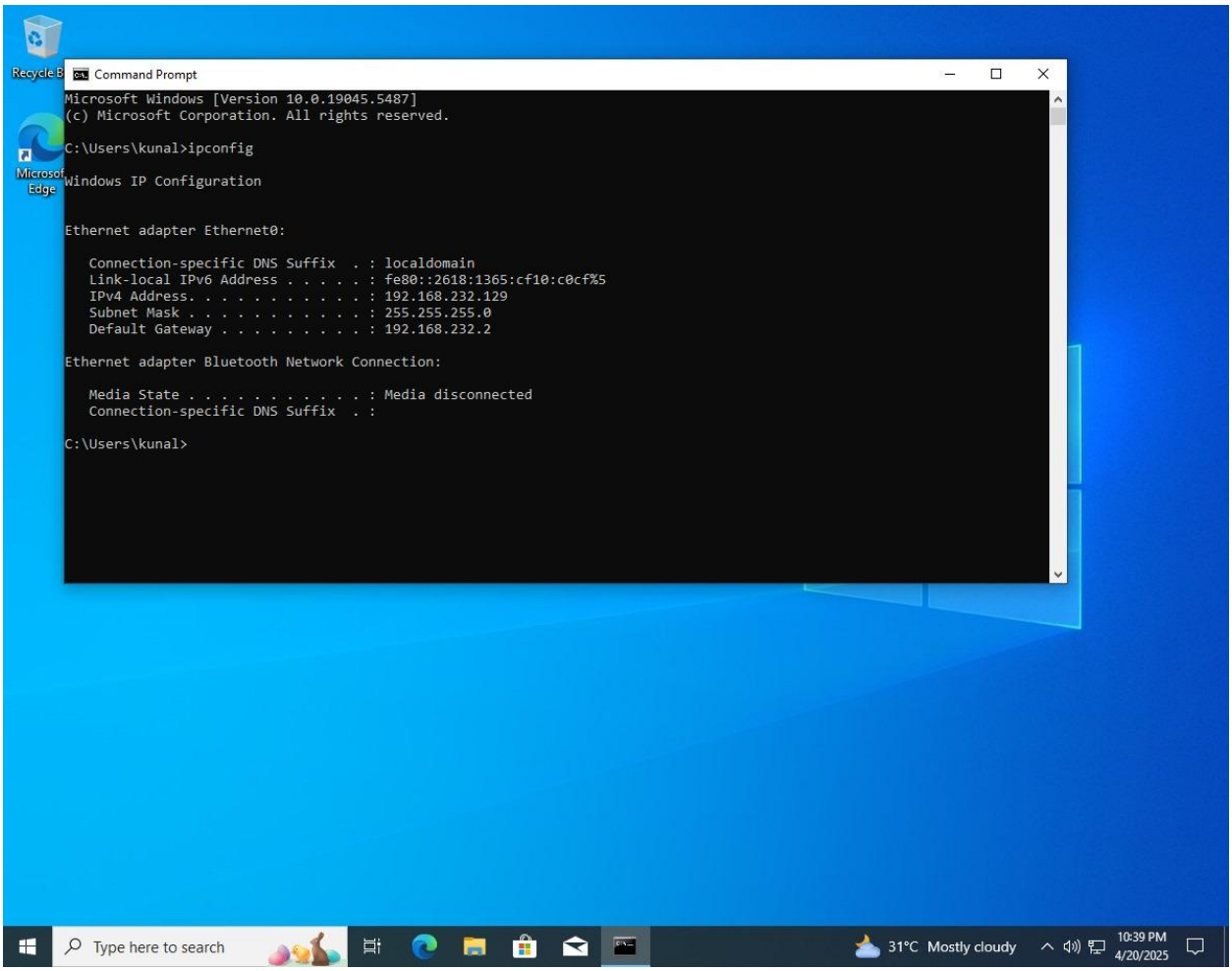
Step 1: Initial Setup:

1. Setting Up VM Machines:

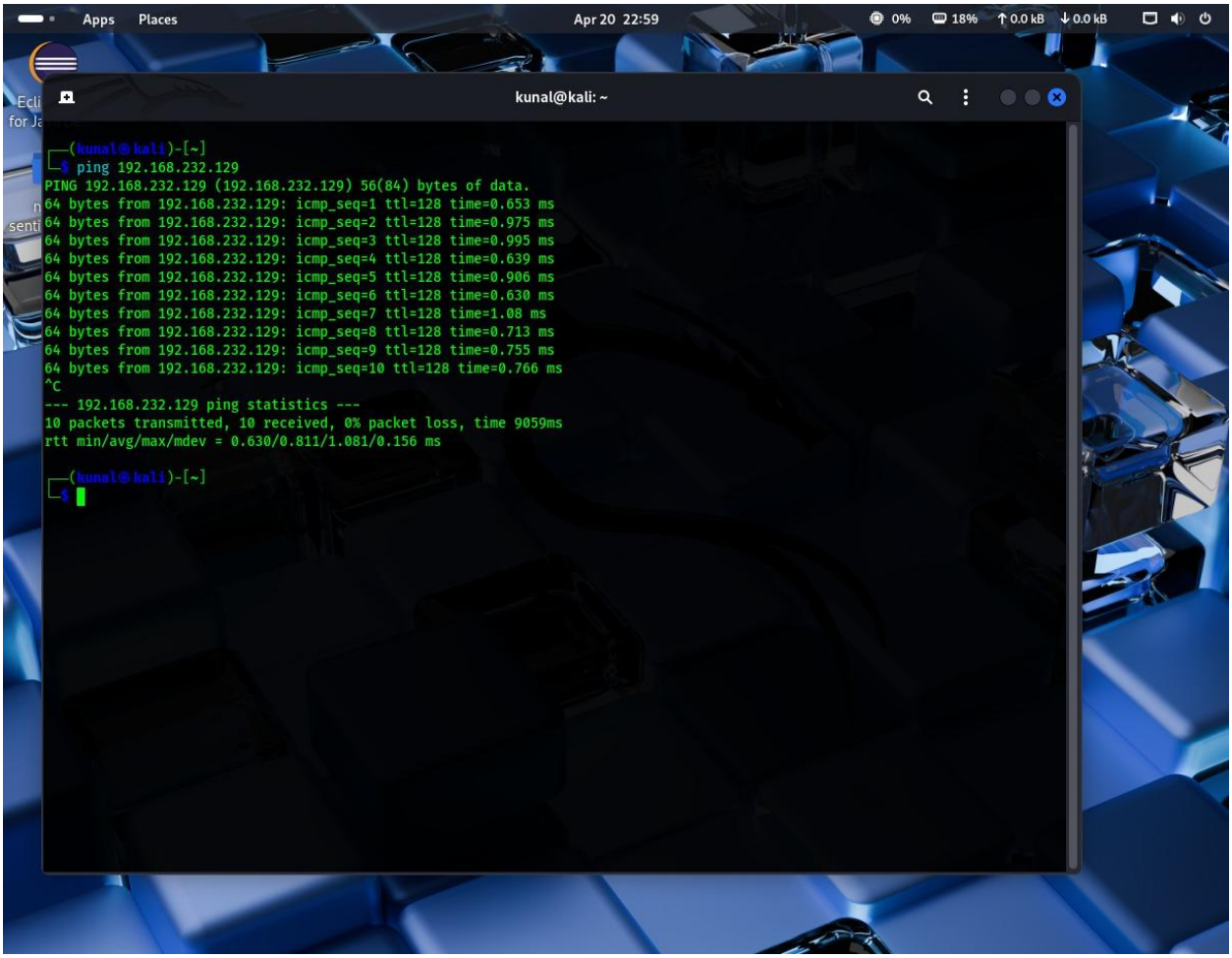
1.a. Attacker Machine(Kali Linux):



1.b. And Target Machine(Windows 10):



2. Check Connectivity:

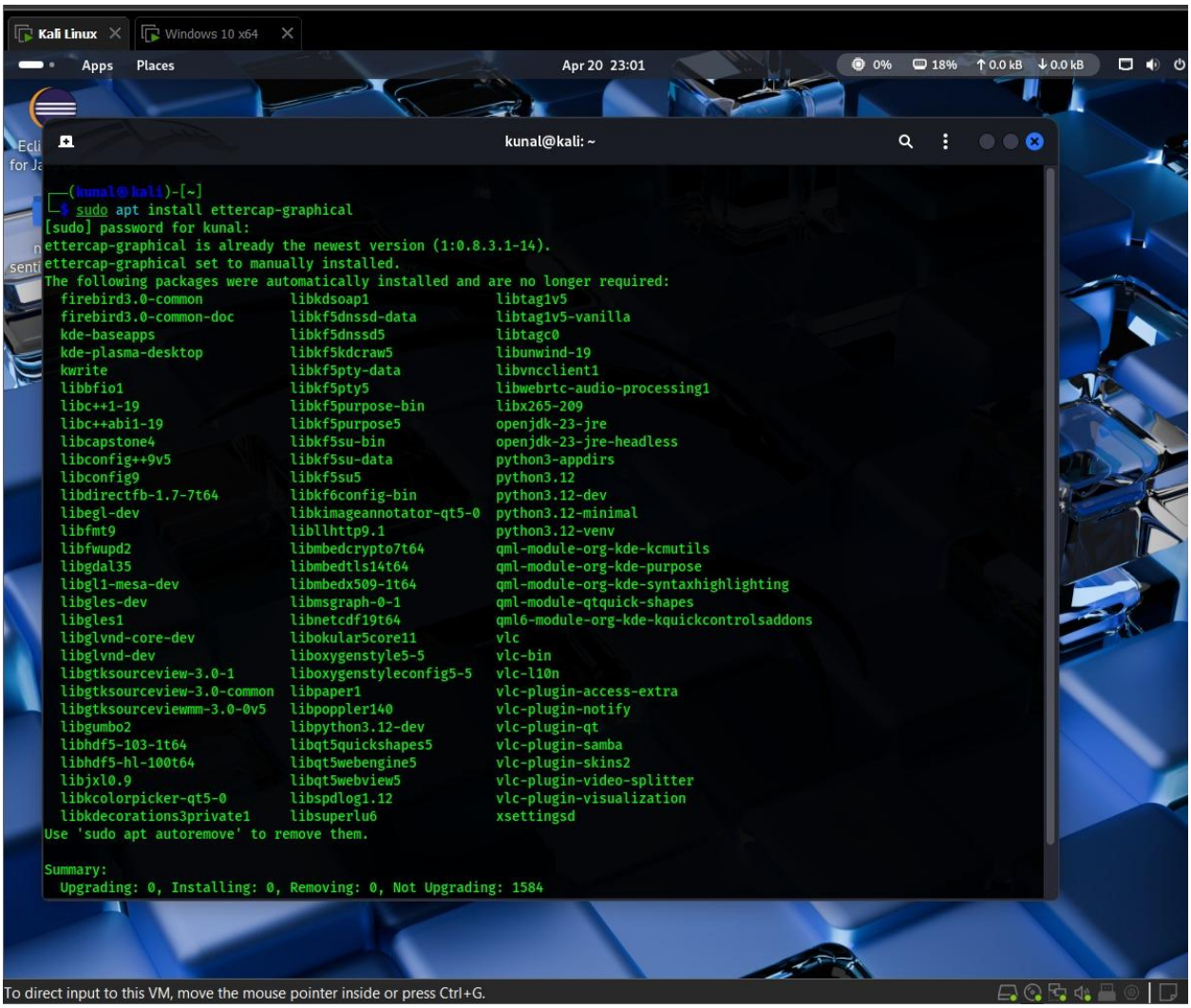


Step 2: Tool Configuration:

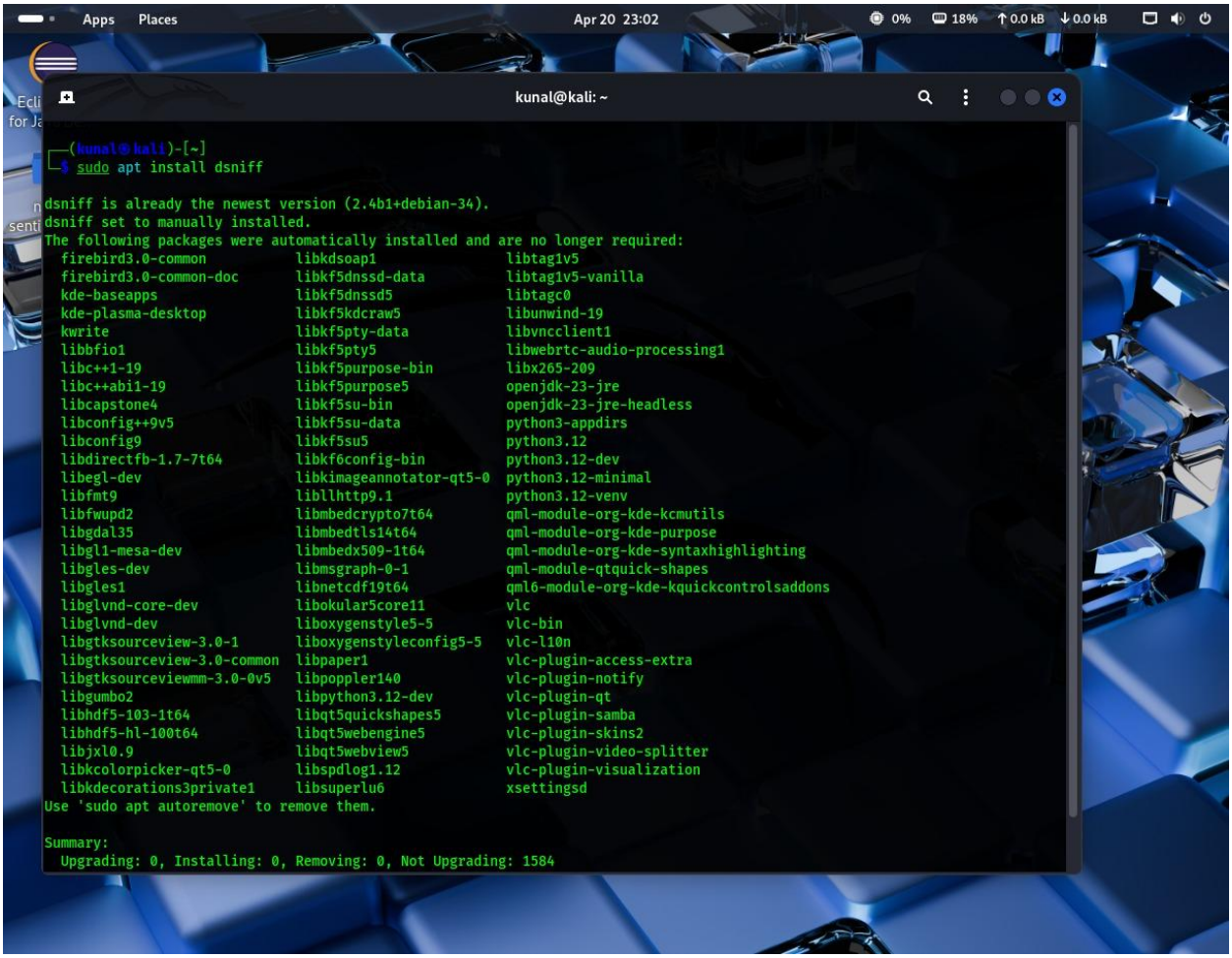
1. Update Your System

```
kunal@kali: ~
┌──(kunal㉿kali)-[~]
└─$ sudo apt update && sudo apt upgrade -y
[sudo] password for kunal:
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [21.0 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [51.6 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [121 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [328 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [204 kB]
Get:7 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [10.6 kB]
Fetched 73.3 MB in 1min 33s (784 kB/s)
1584 packages can be upgraded. Run 'apt list --upgradable' to see them.
The following packages were automatically installed and are no longer required:
  crackmapexec                libmbedtls14t64
  firebird3.0-common          libmbedx509-1t64
  firebird3.0-common-doc      libmsgraph-0-1
  fonts-inter-variable        libnetcdf19t64
  icu-devtools                libokular5core11
  kde-baseapps                libopencv-core406t64
  kde-plasma-desktop          libopencv-imgproc406t64
  kdoctools5                  liboxygenstyle5-5
  kwrite                      liboxygenstyleconfig5-5
  libbfio1                    libpaper1
  libc++1-19                  libpoppler140
  libc++abi1-19               libpoppler145
  libcapstone4                libpython3.12-dev
  libconfig++9v5              libpython3.12-minimal
  libconfig9                  libpython3.12-stdlib
  libdirectfb-1.7-7t64        libpython3.12t64
  libegl-dev                  libqt5quickshapes5
  libflac12t64                libqt5webkit5
  libfmt9                     libqt5webview5
  libfwupd2                   libspdlog1.12
  libgdal35                   libsuperlu6
  libgeos3.13.0               libtag1v5
  libgl1-mesa-dev             libtag1v5-vanilla
  libglapi-mesa               libtagc0
  libgles-dev                 libunwind-19
  libgles1                    libvncclient1
  libglvnd-core-dev           libwebrtc-audio-processing1
  libglvnd-dev                libx265-209
```

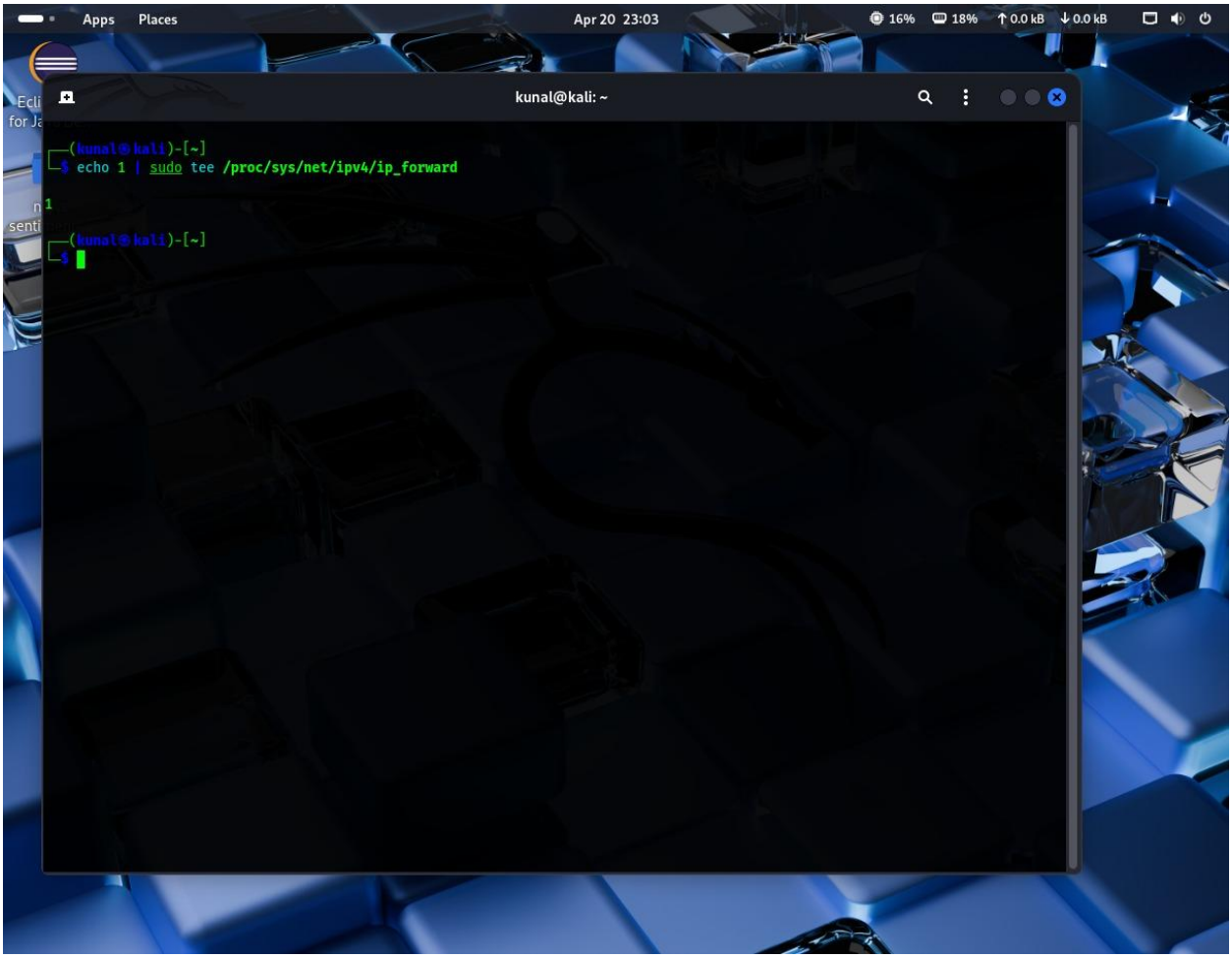
2. Install Ettercap:



3. Install arpspoof:

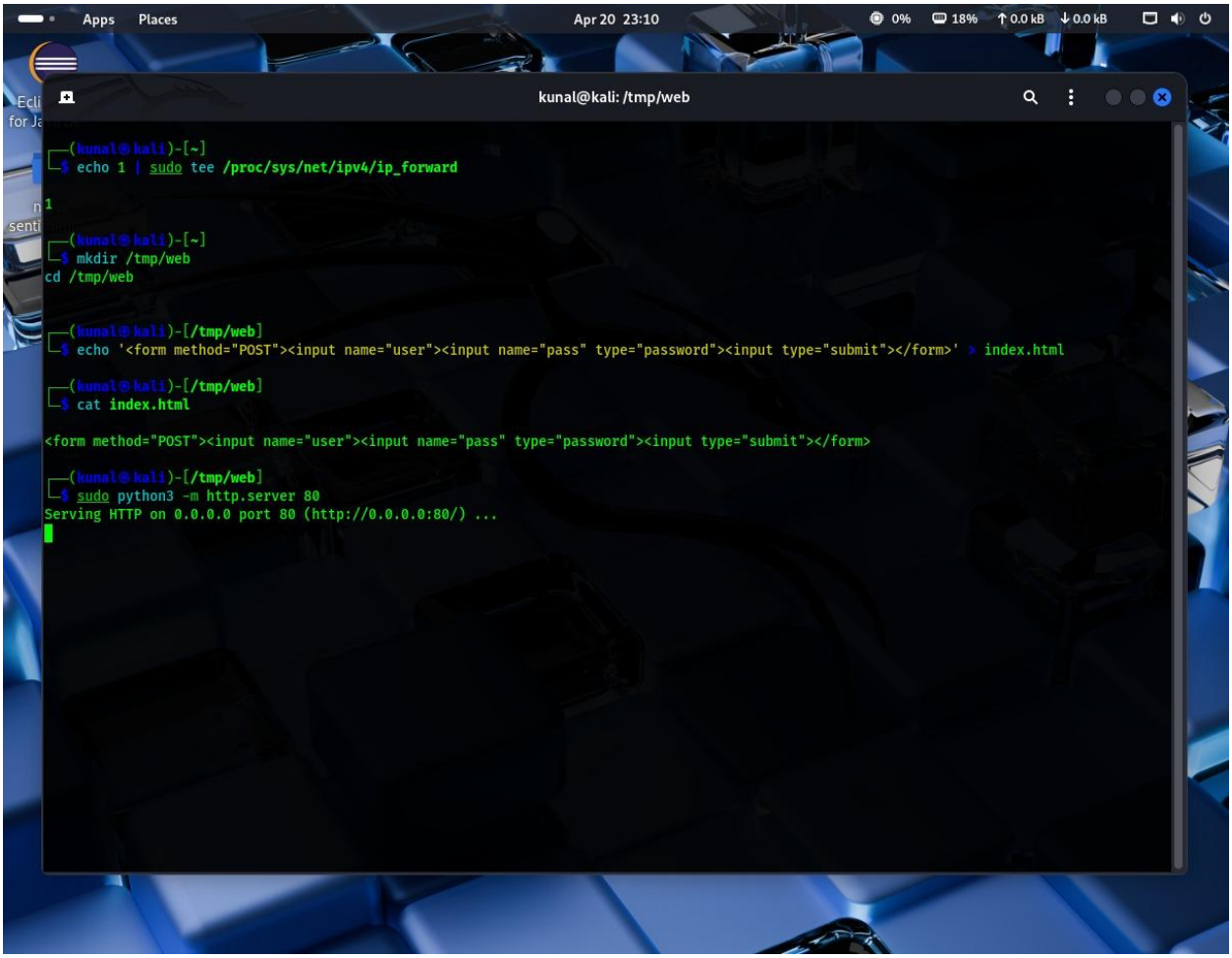


4. Enable IP Forwarding (Required for MITM):

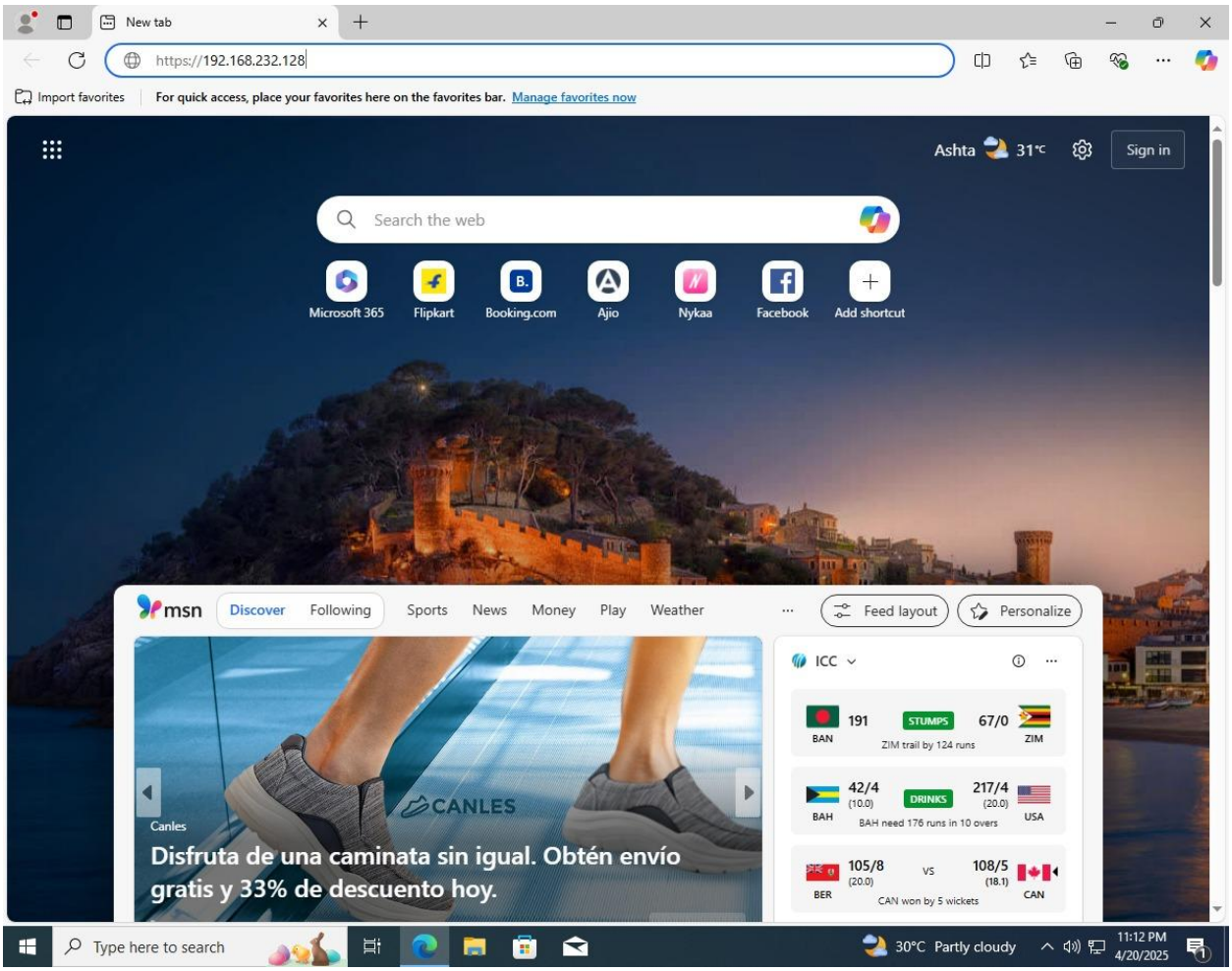


Step 3: Simulate HTTP Login Page (Victim Machine):

1. Create a HTTP login page in the attacker machine for the target to put details into and start the server on port 80:

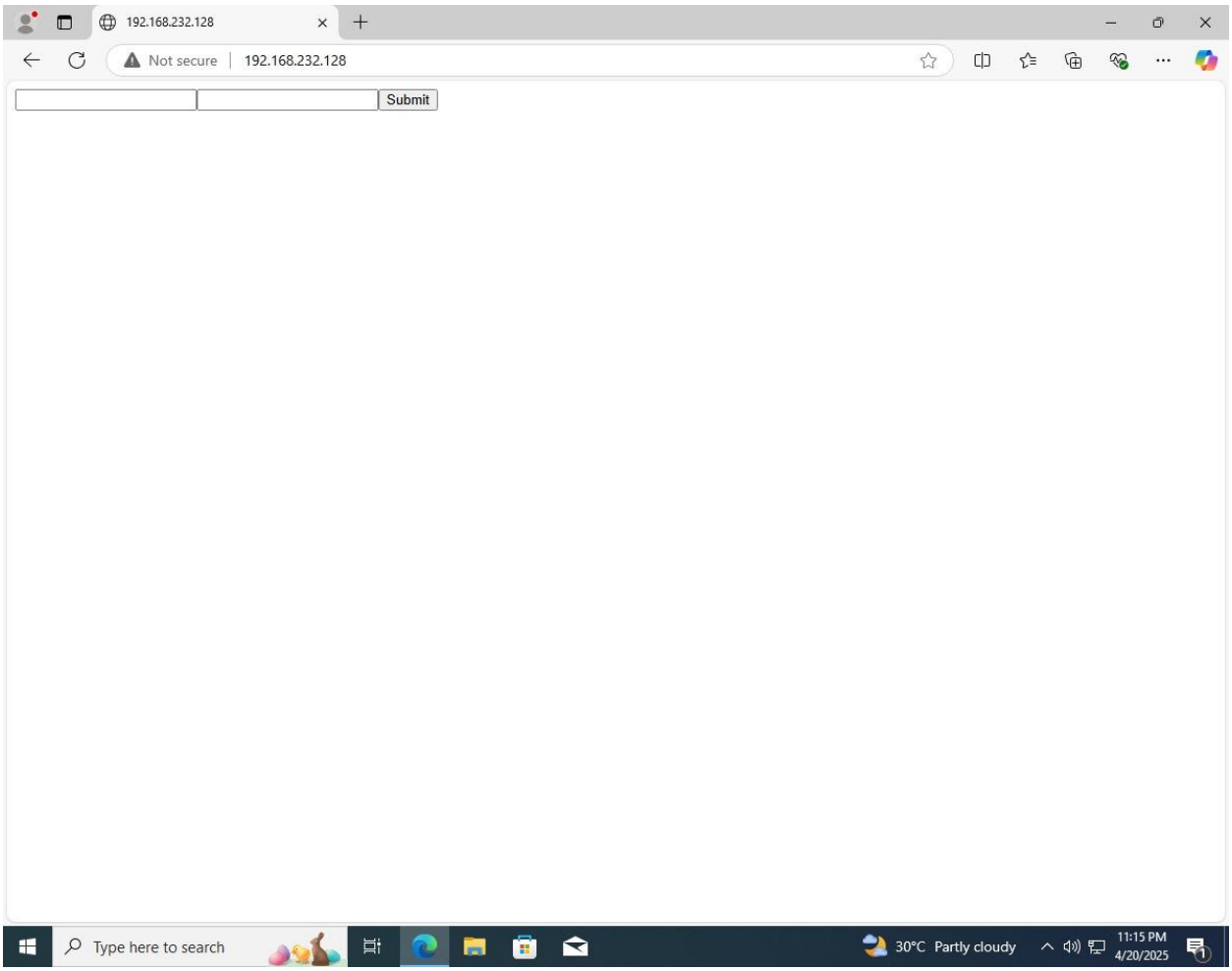


2. On Victim Machine open browser and visit:



3. This will open a webpage asking for userId and Password for login.

I created a simple site but to fool the target we can add many details so that it looks like a real website.



Step 4: Launch the MITM Attack:

Here we have 3 options by which we can sniff user details:

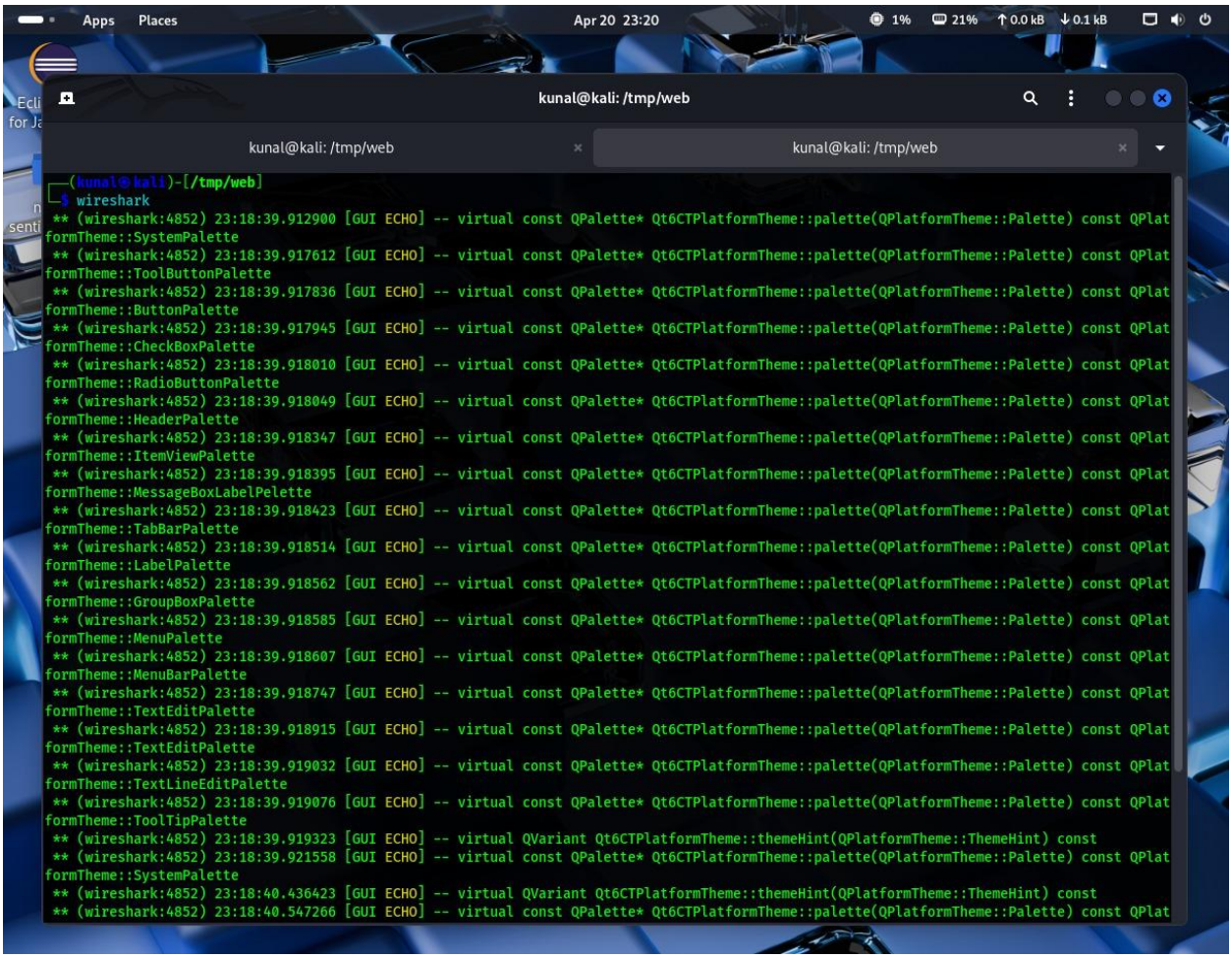
Those are :

- **Wireshark**
- **Arpspoof**
- **Ettercap**

I will be using Wireshark.

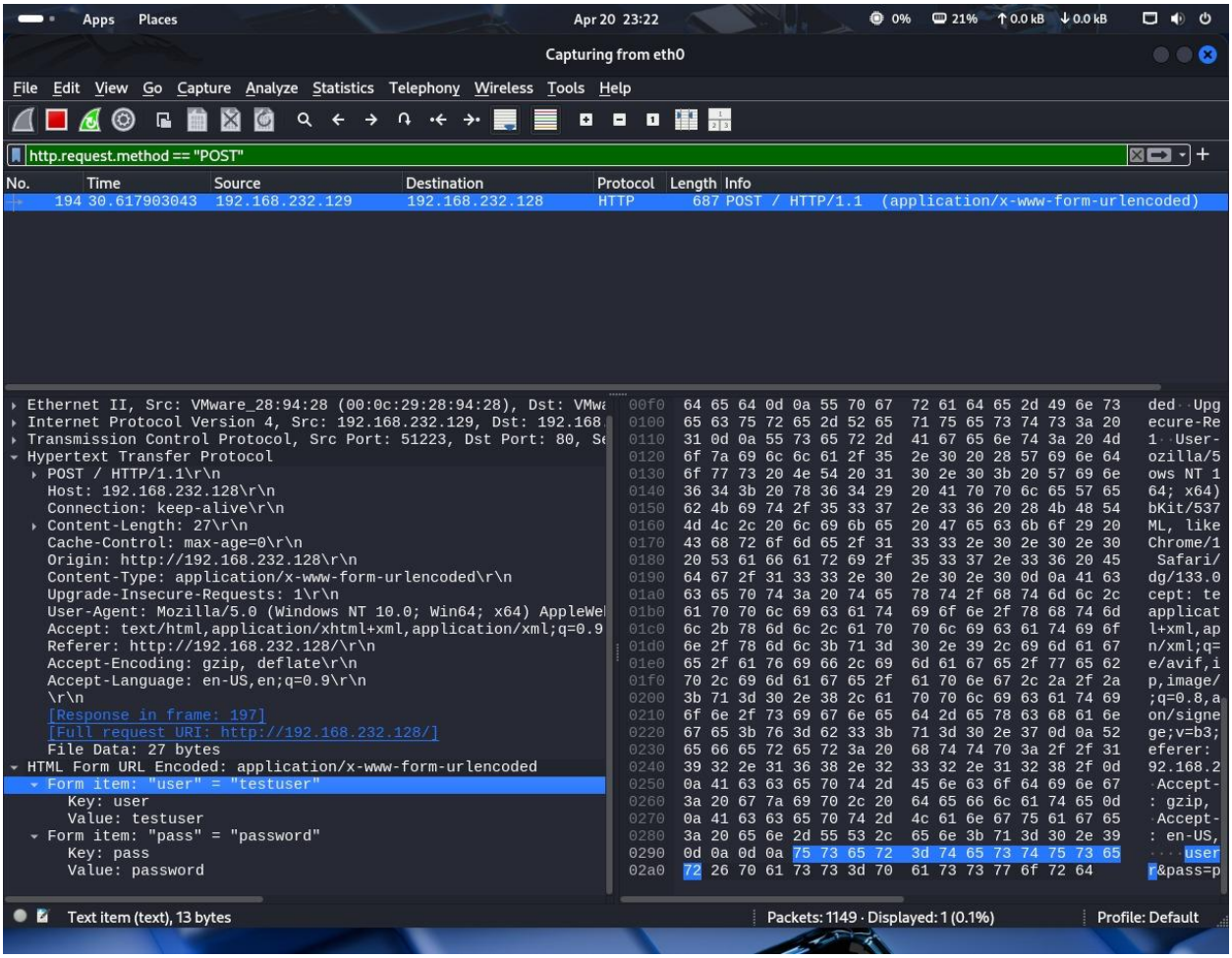
Step 5: Capture Traffic Using Wireshark:

1. Start the Wireshark app on Linux and start capturing packets.



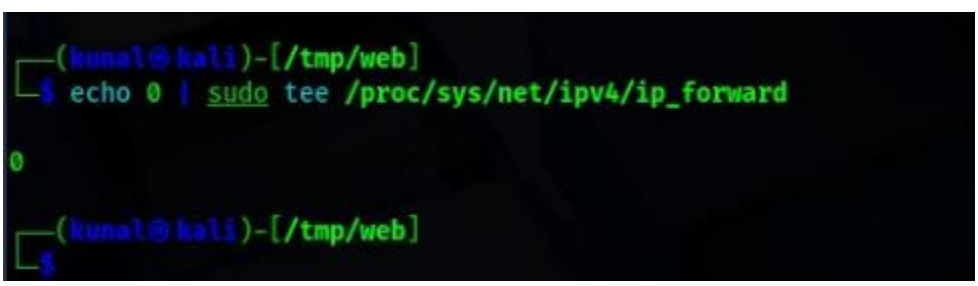
2. Set the filter to: http.request.method == "POST"

This will capture any data inputted by the target.



Here all the details will be available.

3. Now input this to stop ip forwarding:



This concludes this attack.

Observations & Findings:

1. **Successful Credential Interception**
    - **When the victim accessed an HTTP login page, the attacker was able to capture the username and password in plain text using Wireshark.**
2. **ARP Spoofing Effectiveness**
    - **ARP spoofing allowed the attacker to position themselves between the victim and the gateway without being detected by the victim.**
3. **HTTPS Prevented Credential Leakage**
    - **When the victim accessed a similar login page over HTTPS, credentials were not visible in the captured packets.**
4. **VPN Usage Blocked MITM Attack**
    - **Testing with a VPN on the victim machine encrypted all traffic, making it unreadable to the attacker, even during ARP spoofing.**
5. **Tool Integration Worked Smoothly**
    - **Tools like arpspoof, Ettercap, and Wireshark worked effectively together for simulating and analyzing the MITM attack.**
6. **User Awareness is Critical**
    - **Most users are unaware when ARP spoofing is taking place, emphasizing the need for secure browsing habits and encrypted connections.**

Challenges Faced:

● Initial network configuration issues between attacker and victim VMs (had to switch from NAT to Bridged mode)
● Victim machine's firewall blocked ARP spoofing initially; required temporary firewall adjustments
● Ettercap occasionally crashed or failed to start properly; had to troubleshoot with command-line options
● Difficulty filtering relevant HTTP POST traffic in Wireshark due to background noise in network packets
● HTTPS traffic could not be analyzed without proper SSL stripping, which wasn't in project scope
● VPN testing required external setup and extra configuration time for accurate simulation

Security Recommendations:

● Avoid using HTTP for login pages; enforce HTTPS across all web applications
● Use VPNs when accessing public or untrusted Wi-Fi networks to encrypt traffic
● Implement ARP spoofing detection tools or intrusion detection systems (IDS) in networks
● Use strong SSL/TLS certificates and enable HSTS (HTTP Strict Transport Security)
● Educate users to check for HTTPS and valid certificates before entering credentials
● Segment networks using VLANs to limit broadcast domains and reduce ARP attack surface
● Disable unused services and ports to minimize attack vectors
● Regularly update and patch systems to fix known vulnerabilities in network protocols

Final Deliverables:

● Captured Traffic File (.pcap)

- **A Wireshark capture file containing HTTP traffic intercepted during the MITM simulation.**
- **Includes the full packet trace from the victim machine, showing how credentials were exposed during login.**

- **Screenshots of Key Steps**
  - **Visual documentation of each stage of the attack, such as:**
    - **ARP spoofing using arpspoof and ettercap**
    - **Wireshark interface with HTTP POST data**
    - **Python HTTP server setup on attacker machine**
    - **Victim interacting with the fake login page**
  - **Annotated if necessary for clarity.**
- **Project Documentation (PDF/Word)**
  - **A complete write-up of the project, covering:**
    - **Problem statement, objectives, methodology, tools used**
    - **Step-by-step implementation with commands**
    - **Observations, findings, challenges, and recommendations**
    - **Conclusion and summary of learning**
- **Network Diagram**
  - **A simple visual showing how the MITM attack was set up in the lab environment**
  - **Includes VM details, IP addresses, network mode, and traffic flow paths**
- **Demo Video (Optional)**
  - **A screen recording walking through the attack from start to finish**
  - **Shows real-time output of tools like Wireshark and Ettercap during the interception**
  - **Useful for viva, presentation, or submission if permitted**
- **Presentation Slides**
  - **A summary of the project in a slide deck format**
  - **Useful for classroom or final evaluation presentations**
  - **Covers background, setup, key steps, findings, and recommendations**

Conclusion:

This project successfully simulated a Man-in-the-Middle (MITM) attack to demonstrate how attackers can intercept sensitive information, such as login credentials, over unsecured networks. By using tools like Ettercap, arpspoof, and Wireshark, the project recreated a real-world scenario where a victim unknowingly connected to a compromised public Wi-Fi network and submitted data over an unencrypted HTTP connection.

The simulation clearly showed that without encryption, critical user information can be easily captured and exploited. It also highlighted the dangers of ARP spoofing and the effectiveness of packet sniffing tools in exposing vulnerabilities in network communication.

Beyond executing the attack, the project placed equal emphasis on analyzing the captured data and understanding the consequences of poor security practices. It reinforced the importance of secure protocols (HTTPS), encrypted tunnels (VPNs), and proper network segmentation as key defenses against MITM attacks.

Overall, this hands-on experience deepened the understanding of both offensive and defensive aspects of cybersecurity. It provided valuable insights into how attackers think and operate, while also underlining the critical need for strong security policies, user education, and constant vigilance in today's connected environments.