LEMBAR ANALISA

Praktikum Network Security (Sniffing, Spoofing Dan Session Hijacking)

Tanggal Praktikum : 25 Desember 2025

Kelas : 5A

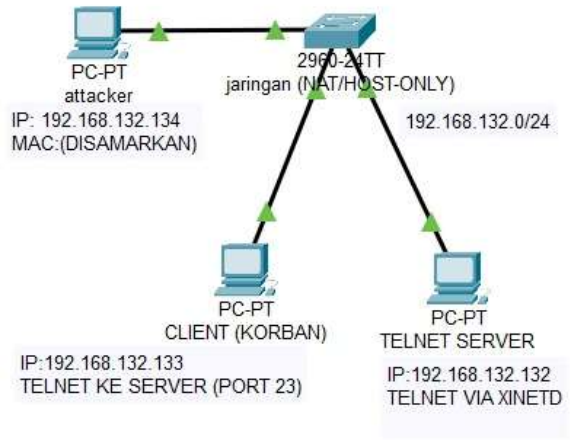
Nama Kelompok : Fathya Shabira A.T 105841111923

: Muh Reyhan Dwi Wijaya 105841112623

: Andi Muhammad Yusuf 105841112923

A. ARP SPOOFING

1. Gambar topologi jaringan beserta dengan IP Addressnya



Gambar diatas menunjukkan topologi ARP spoofing sederhana dengan satu attacker, satu client korban, dan satu telnet server dalam satu LAN 192.168.132.0/24.

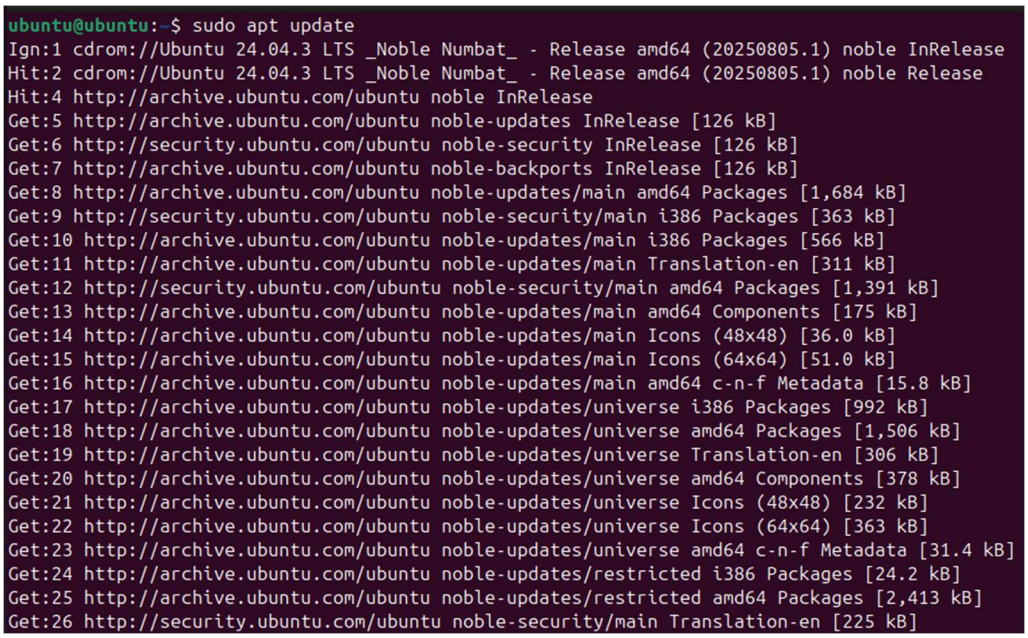
Penjelasan:

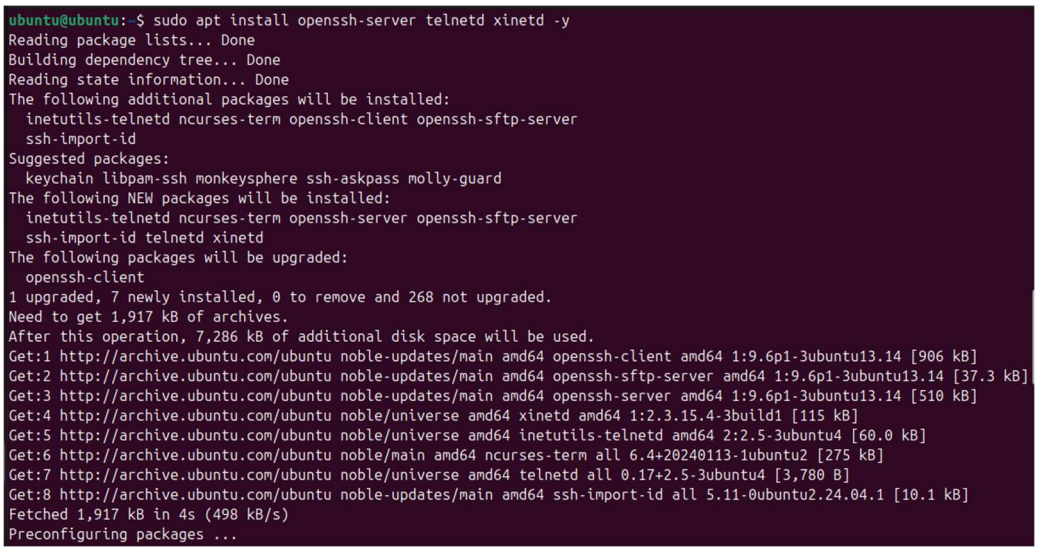
- ≈ Attacker: IP 192.168.132.134, mengirim ARP palsu agar trafik Telnet lewat dirinya.
- ≈ Client (korban): IP 192.168.132.133, melakukan Telnet ke server di port 23.
- ≈ Telnet server: IP 192.168.132.132, layanan Telnet via xinetd sebagai tujuan koneksi.
- ≈ Switch 2960-24TT: Menghubungkan semua host dalam satu broadcast domain, sehingga ARP spoofing efektif.

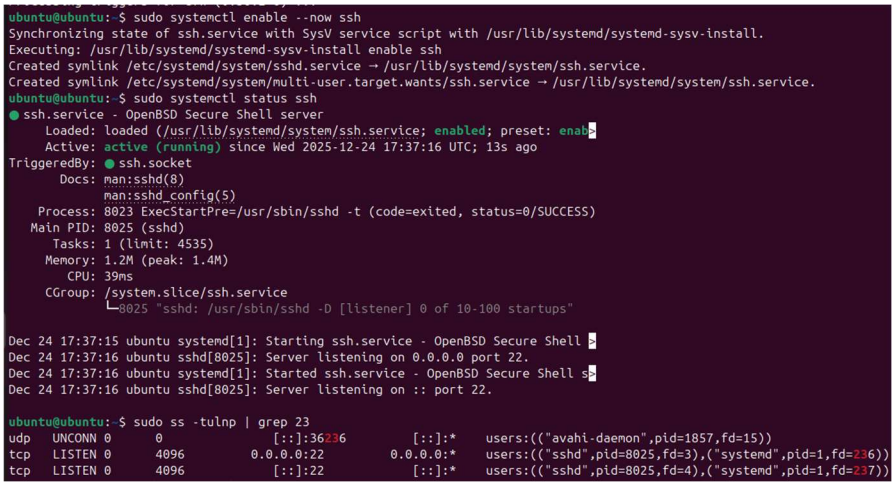
2. **Instal aplikasi telnet dan ssh pada Server dan lakukan tes koneksi dari client**

   Gambar-gambar dibawah ini menjelaskan langkah instal dan aktivasi layanan Telnet dan SSH di Ubuntu server.

   a. **Install telnet dan ssh**







Penjelasan:

≈ **sudo apt update**

Meng-update daftar paket Ubuntu sebelum instal.

≈ **sudo apt install openssh-server telnetd xinetd -y**

Menginstal SSH server, Telnet server, dan xinetd sebagai service manager Telnet.

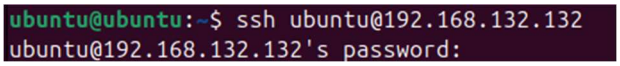
≈ **sudo systemctl enable --now ssh**

Mengaktifkan dan langsung menyalakan service SSH.

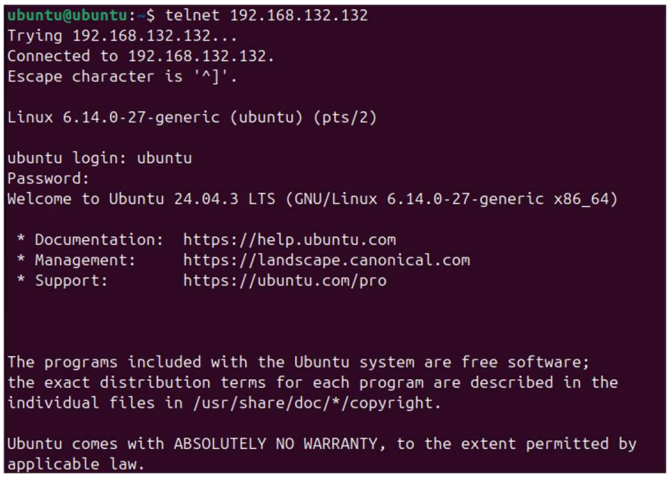
≈ **sudo systemctl status ssh + sudo ss -tulpn | grep 22**

Mengecek bahwa SSH running dan port 22 sudah terbuka.

b. Tes koneksi dari client



Gambar itu menunjukkan dua jenis *tes koneksi* dari client ke server.



Penjelasan:

≈ Perintah ssh ubuntu@192.168.132.132

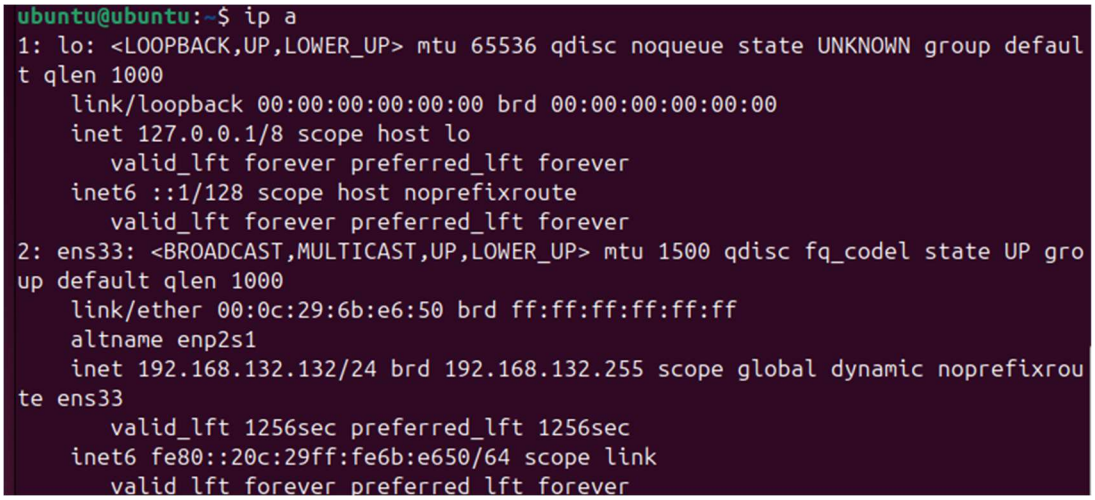
Client mencoba login ke server via SSH dan diminta memasukkan password user *ubuntu*, menandakan port 22 dan layanan SSH di server sudah aktif.

≈ Perintah telnet 192.168.132.132

Client terhubung ke server via Telnet, diminta login dan password, lalu berhasil masuk ke shell Ubuntu 24.04.3 LTS, menandakan layanan Telnet di server berjalan dengan baik.

3. Catat MAC Address dari komputer client dan server

a. Client



Gambar ini menunjukkan konfigurasi IP dan MAC Address pada client dengan perintah ip a.

Penjelasan:

≈ Interface ens33 dalam keadaan UP dan mendapat IP 192.168.132.132/24 dengan broadcast 192.168.132.255, artinya client terhubung ke jaringan 192.168.132.0/24 dan bisa berkomunikasi dengan server serta attacker.

≈ MAC Address client ditampilkan sebagai 00:0c:29:6b:e6:50, nilai ini yang nantinya akan dimanipulasi mapping-nya di tabel ARP ketika ARP spoofing dilakukan.

b. Server

```
ubuntu@ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:6b:e6:50 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.132.132/24 brd 192.168.132.255 scope global dynamic noprefixroute ens33
       valid_lft 1271sec preferred_lft 1271sec
    inet6 fe80::20c:29ff:fe6b:e650/64 scope link
       valid_lft forever preferred_lft forever
```

Gambar ini menunjukkan konfigurasi IP dan MAC Address pada server dengan perintah ip a

Penjelasan:

≈ Interface ens33 berstatus UP dengan IP 192.168.132.132/24 dan broadcast 192.168.132.255, artinya server berada di jaringan yang sama dengan client dan attacker sehingga bisa jadi target Telnet/SSH serta ARP spoofing.

≈ MAC Address server adalah 00:0c:29:6b:e6:50, nilai ini yang seharusnya direferensikan oleh client; setelah ARP spoofing, mapping IP server di tabel ARP client akan mengarah ke MAC attacker, bukan lagi ke MAC ini.

4. Catat MAC Address setelah dilakukan arp spoofing dengan tool hunt (poin 1.e), bandingkan dengan MAC address sebelumnya.

a. Sebelum Di Arp Spoofing

Gambar ini menunjukkan isi tabel ARP/server sebelum dilakukan ARP spoofing dengan perintah ip neigh.

```
ubuntu@ubuntu:~$ ip neigh
192.168.132.134 dev ens33 lladdr 00:0c:29:46:ae:83 REACHABLE
192.168.132.2 dev ens33 lladdr 00:50:56:ed:1c:56 REACHABLE
192.168.132.254 dev ens33 lladdr 00:50:56:ed:23:18 STALE
192.168.132.133 dev ens33 lladdr 00:0c:29:33:ad:34 DELAY
```

Setiap IP (192.168.132.134 attacker, 192.168.132.2 gateway, 192.168.132.254, 192.168.132.133 client) masih terhubung ke MAC Address aslinya, dan statusnya REACHABLE/STALE/DELAY yang berarti komunikasi berlangsung normal tanpa manipulasi ARP

b. Setelah Di Arp Spoofing

Gambar ini menampilkan kondisi tabel ARP setelah dilakukan ARP spoofing dengan perintah ip neigh.

```
ubuntu@ubuntu:~$ ip neigh
192.168.132.134 dev ens33 lladdr 00:0c:29:46:ae:83 STALE
192.168.132.2 dev ens33 lladdr 00:50:56:ed:1c:56 REACHABLE
192.168.132.254 dev ens33 lladdr 00:50:56:ed:23:18 STALE
192.168.132.133 dev ens33 lladdr 00:0c:29:46:ae:83 REACHABLE
```

Terlihat bahwa IP client 192.168.132.133 dan IP attacker 192.168.132.134 sekarang sama-sama menunjuk ke MAC Address attacker 00:0c:29:46:ae:83, sehingga trafik yang seharusnya langsung ke client dapat dialihkan melewati attacker (man-in-the-middle).

5. Catat proses terjadinya session hijackings

a. Telnet client-server

- **Telnet Dari Client**

```
ubuntu@ubuntu:~$ telnet 192.168.132.132
Trying 192.168.132.132...
Connected to 192.168.132.132.
Escape character is '^]'.

Linux 6.14.0-27-generic (ubuntu) (pts/3)

ubuntu login: ubuntu
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-27-generic x86_64)

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

Gambar ini menunjukkan bahwa client berhasil membuka sesi Telnet ke server pada IP 192.168.132.132 sebelum session hijacking terjadi.

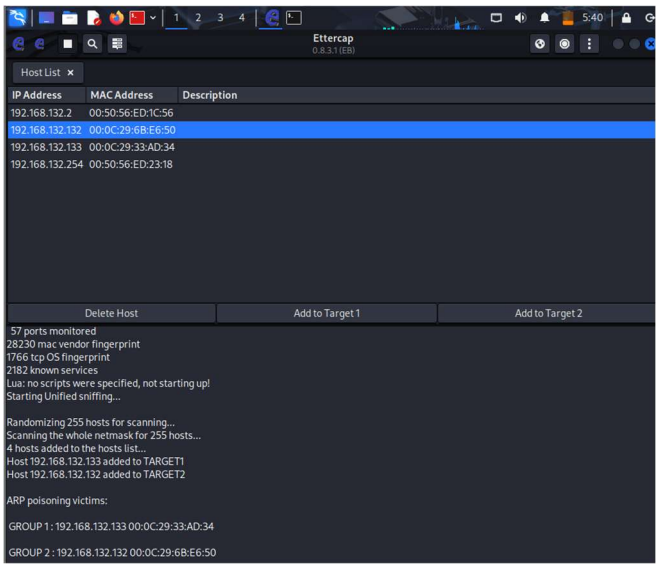
Penjelasan:

≈ Client menjalankan telnet 192.168.132.132 dan tersambung ke server, muncul banner Linux Ubuntu serta prompt login.

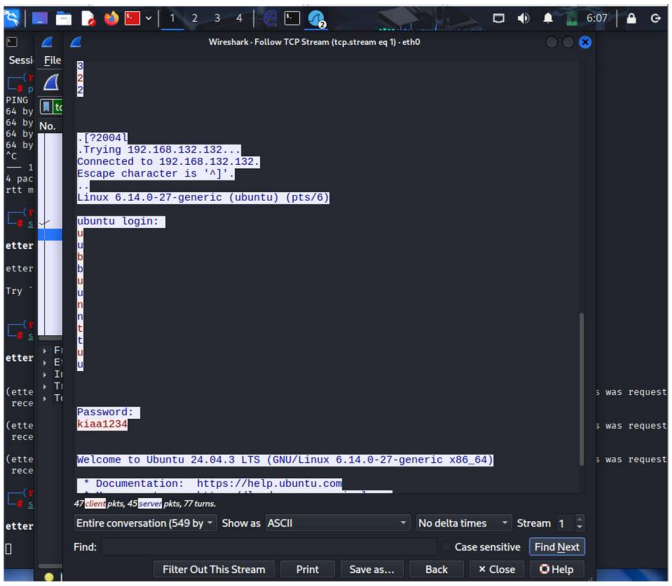
≈ Setelah ARP spoofing dilakukan, aliran paket Telnet client–server dialihkan melalui attacker; attacker dapat menangkap kredensial dan mengambil alih sesi tersebut, sementara dari sisi client tampak seperti koneksi Telnet biasa tanpa indikasi gangguan.

b. Amati pada komputer attacker

Pada komputer attacker terlihat dua hal penting saat ARP spoofing dan session hijacking berlangsung.



Di Ettercap, attacker memindai jaringan dan menemukan host 192.168.132.132 (server) dan 192.168.132.133 (client), lalu menambahkannya sebagai *Target1* dan *Target2* sehingga Ettercap melakukan ARP poisoning terhadap keduanya dan seluruh trafik Telnet client–server lewat attacker

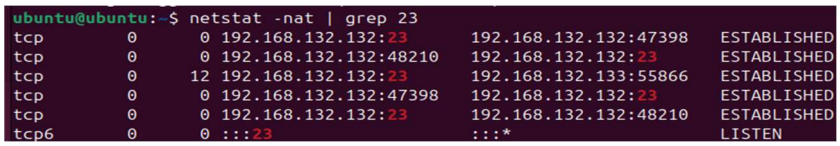


Di Wireshark (Follow TCP Stream), attacker dapat melihat isi sesi Telnet secara utuh dalam bentuk teks: proses koneksi ke 192.168.132.132, prompt ubuntu login:,

input username, hingga password kiaa1234 serta banner welcome Ubuntu, yang menunjukkan bahwa sesi Telnet berhasil disadap dan dapat di-*hijack*.

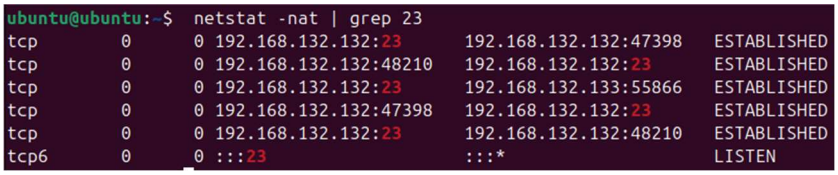
c. Catat koneksi client-server setelah dilakukan hijacking dgn netstat -nat

- **Client**



Pada client, terlihat beberapa entri TCP dari 192.168.132.132:23 (Telnet server) ke port sumber berbeda (misal 47398, 48210, 55866) dengan status ESTABLISHED, menandakan sesi Telnet ke server masih aktif meskipun trafiknya sudah dialihkan lewat attacker.

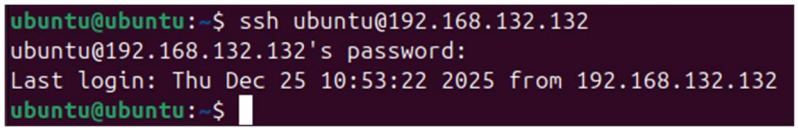
- **Server**



Pada server, ditampilkan entri yang sama: koneksi dari IP client ke 192.168.132.132:23 juga berstatus ESTABLISHED serta ada satu entri LISTEN di :::23, yang menunjukkan Telnet server tetap mendengarkan koneksi baru di port 23 sambil mempertahankan sesi yang sedang di-*hijack*.

d. Ulangi langkah diatas jika yang dijalankan aplikasi ssh

- **Aplikasi Ssh**



Perintah ssh ubuntu@192.168.132.132 dijalankan, lalu setelah memasukkan password, muncul pesan Last login: Thu Dec 25 10:53:22 2025 from 192.168.132.132 yang menandakan login berhasil dan sesi shell aman (terenkripsi) sudah aktif antara client dan server.

- **Cek Koneksi Ssh**
  - **Client**

```
ubuntu@ubuntu:~$ netstat -nat | grep 22
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0     52 192.168.132.132:22      192.168.132.132:52022   ESTABLISHED
tcp        0      0 192.168.132.132:52022   192.168.132.132:22      ESTABLISHED
tcp        0      0 192.168.132.132:22      192.168.132.132:37914   ESTABLISHED
tcp        0      0 192.168.132.132:37914   192.168.132.132:22      ESTABLISHED
tcp6       0      0 :::22                   :::*                    LISTEN
```

Client menjalankan ssh ubuntu@192.168.132.132, memasukkan password, dan berhasil login ke server; ini menandakan layanan SSH di server aktif dan koneksi terenkripsi end-to-end

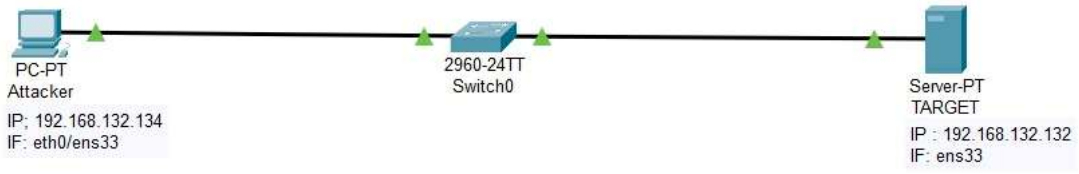
  - **Server**

```
ubuntu@ubuntu:~$ netstat -nat | grep 22
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 192.168.132.132:22      192.168.132.132:52022   ESTABLISHED
tcp        0      0 192.168.132.132:52022   192.168.132.132:22      ESTABLISHED
tcp        0      0 192.168.132.132:22      192.168.132.132:37914   ESTABLISHED
tcp        0      0 192.168.132.132:37914   192.168.132.132:22      ESTABLISHED
tcp6       0      0 :::22                   :::*                    LISTEN
```

Hasil netstat -nat | grep 22 di sisi client dan server: terlihat koneksi TCP dari client ke port 22 server dengan status ESTABLISHED serta entri LISTEN pada 0.0.0.0:22/:::22, yang menunjukkan SSH server siap menerima koneksi baru meskipun satu sesi sedang berjalan

B. IP SPOOFING

1. Gambar topologi jaringan beserta dengan IP Addressnya



Gambar ini adalah topologi sederhana untuk IP spoofing hanya dengan attacker dan satu server target yang terhubung lewat switch.

Penjelasan:

≈ PC-PT Attacker memiliki IP 192.168.132.134 dengan interface eth0/ens33, dan akan mengirim paket dengan IP sumber palsu menuju server.

≈ Server-PT Target memiliki IP 192.168.132.132 dengan interface ens33, menjadi korban paket spoofing yang diteruskan oleh switch 2960-24TT dalam satu segmen jaringan yang sama.

2. **Jalankan beberapa tool ip spoofing dan catat apa yang terjadi**

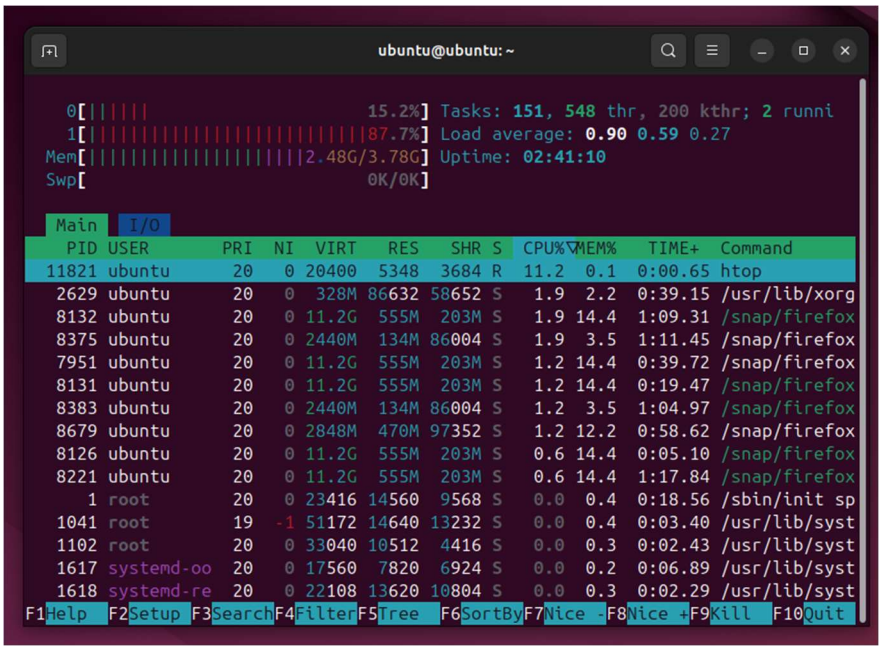
Kedua gambar menunjukkan efek serangan ICMP flood (pod_spoofing) ke server.

a. pod_spoofing

```
┌──(root㉿kali)-[~]
└─# sudo hping3 -1 192.168.132.132 --flood --data 12000
HPING 192.168.132.132 (eth0 192.168.132.132): icmp mode set, 28 headers + 12000 data bytes
hping in flood mode, no replies will be shown
```

Di sisi attacker, perintah sudo hping3 -1 192.168.132.132 --flood --data 12000 mengirim paket ICMP berukuran besar secara kontinu ke IP server, sehingga membanjiri jaringan dan membuat reply ICMP tidak sempat diproses/ditampilkan.

- **Di server: CPU/network naik, mungkin ICMP reply tidak stabil**.



Di sisi server, tampilan htop menunjukkan penggunaan CPU dan resource sistem meningkat selama serangan berlangsung, yang mengindikasikan beban tambahan akibat harus menangani banyak paket ICMP masuk sehingga performa jaringan/server bisa menjadi tidak stabil.

b. syn_flood

Gambar ini menunjukkan serangan TCP SYN flood ke port SSH (22) pada server dan efeknya di netstat.

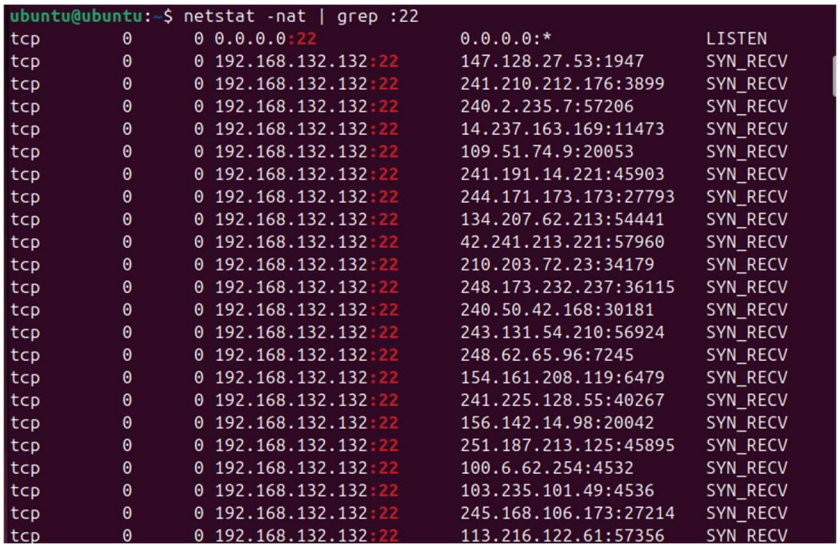
```
┌──(root㉿kali)-[~]
└─# sudo hping3 -S -p 22 --flood --rand-source 192.168.132.132
HPING 192.168.132.132 (eth0 192.168.132.132): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Di sisi attacker, perintah:

sudo hping3 -S -p 22 --flood --rand-source 192.168.132.132

mengirim banyak paket TCP SYN ke port 22 server dengan sumber IP acak (--rand-source), sehingga seolah-olah datang dari banyak host berbeda.

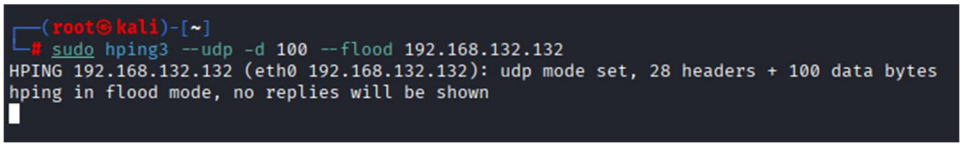
- **Di server: netstat -nat | grep :22 akan banyak baris SYN_RECV, koneksi SSH baru jadi lambat/gagal.**



Di sisi server, output netstat -nat | grep :22 memperlihatkan satu entri 0.0.0.0:22 LISTEN dan puluhan koneksi ke 192.168.132.132:22 berstatus SYN_RECV, artinya server sudah menerima SYN tetapi belum menyelesaikan handshake karena tidak pernah mendapat ACK balik dari IP palsu tersebut . Kondisi ini memenuhi backlog koneksi dan membuat koneksi SSH baru menjadi sangat lambat atau gagal terbentuk

c. teardrop+spoofing

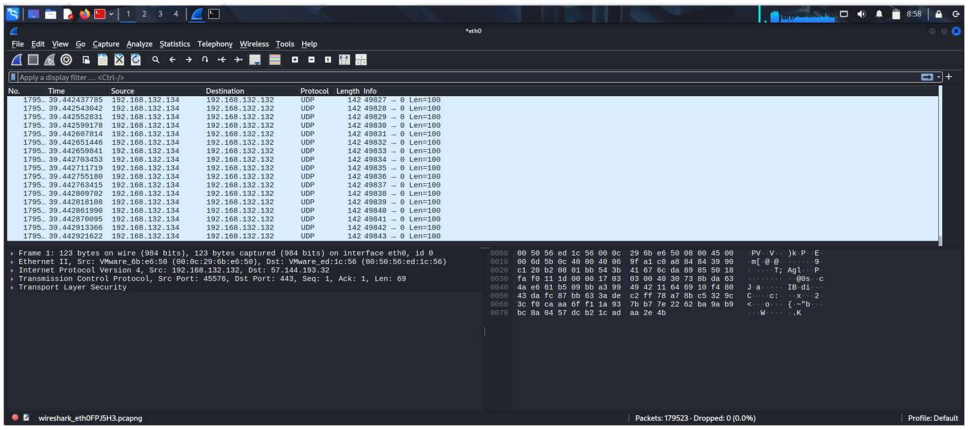
Kedua gambar menggambarkan serangan UDP flood yang dalam konteks praktikum ini dipakai sebagai contoh teardrop+spoofing (membanjiri target dengan paket terfragmentasi/UDP).



Di sisi attacker, perintah:

sudo hping3 --udp -d 100 --flood 192.168.132.132

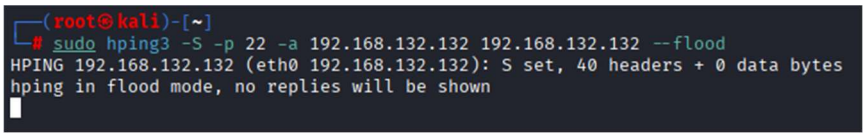
mengirim paket UDP dengan payload 100 byte ke server secara kontinu dalam mode flood, tanpa menunggu balasan (no replies will be shown), sehingga memenuhi jalur jaringan dengan trafik UDP.



Di sisi server yang dipantau dengan Wireshark, terlihat banyak paket UDP dari IP attacker 192.168.132.134 menuju 192.168.132.132 dengan panjang yang sama (Len=100), berulang terus-menerus, yang dapat menyebabkan beban proses fragmentasi/reassembly meningkat dan berpotensi mengganggu stabilitas layanan di server.

d. land_attack

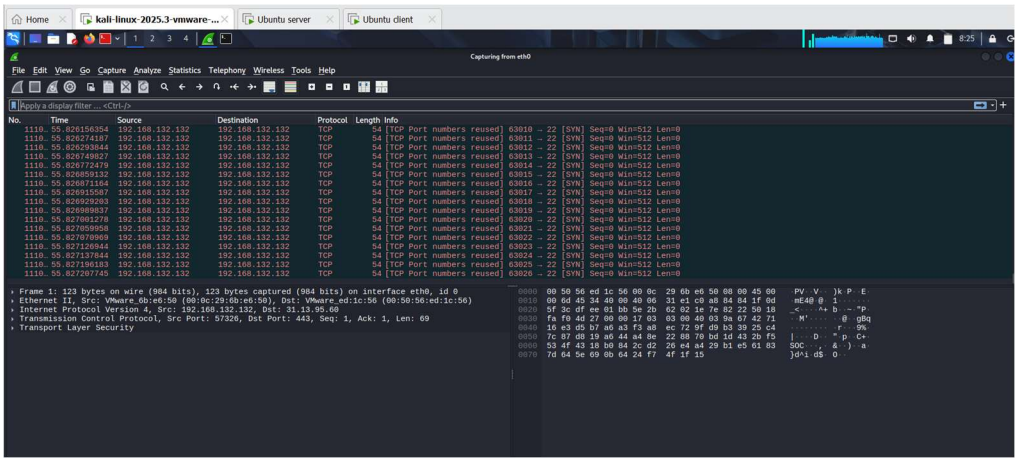
Kedua gambar menggambarkan serangan land attack ke port SSH (22) server.



Di sisi attacker, perintah:

sudo hping3 -S -p 22 -a 192.168.132.132 192.168.132.132 --flood

mengirim paket TCP SYN ke port 22 server dengan alamat sumber (-a) dipalsukan sama dengan alamat tujuan, yaitu 192.168.132.132, sehingga server seolah-olah menyerang dirinya sendiri.



Di Wireshark pada server, terlihat banyak paket TCP dengan source dan destination IP sama (192.168.132.132) dan port tujuan 22, berulang dalam jumlah besar; hal ini membuat server membangun koneksi dengan dirinya sendiri dan dapat menguras resource jaringan/CPU, menyebabkan layanan SSH terganggu atau tidak responsif

C. KESIMPULAN

1. ARP Spoofing & Session Hijacking

a. Topologi awal berisi attacker, client korban, dan server (Telnet/SSH) dalam satu jaringan lokal 192.168.132.0/24. Kondisi normal: tabel ARP di client dan server berisi pasangan IP–MAC yang benar, koneksi Telnet/SSH berjalan wajar.

b. Setelah ARP spoofing (menggunakan Ettercap/hunt), entry ARP di korban berubah: IP lawan (client/server) kini menunjuk ke MAC attacker. Artinya seluruh trafik client–server dialihkan melewati attacker (man-in-the-middle).

c. Pada koneksi Telnet, karena protokol ini tidak terenkripsi, attacker bisa:
   - Menyadap seluruh isi sesi (banner, username, dan password) melalui Wireshark (Follow TCP Stream).
   - Melanjutkan atau mengambil alih sesi (session hijacking), sementara di sisi client dan server koneksi masih tampak ESTABLISHED pada netstat -nat

d. Saat skenario yang sama diulang dengan SSH:
   - ARP spoofing tetap memungkinkan attacker berada di tengah jalur komunikasi.
   - Namun isi payload SSH terenkripsi, sehingga attacker tidak bisa membaca username/password atau isi sesi hanya dengan sniffing biasa.

- Ini menunjukkan bahwa enkripsi (SSH) jauh lebih aman dibanding Telnet terhadap serangan sniffing/session hijack berbasis MITM.

2. IP Spoofing & Berbagai Serangan Flood

Pada topologi attacker–server, kamu menguji beberapa jenis serangan berbasis spoofing dengan hping3:

a. ICMP Flood / pod_spoofing
   - hping3 -1 ... --flood --data 12000 mengirim ICMP berukuran besar secara terus-menerus ke server.
   - Di server, penggunaan CPU/network meningkat dan reply ICMP menjadi tidak stabil. Ini menggambarkan efek DoS dengan membanjiri jalur ICMP.

b. TCP SYN Flood ke port 22 (SSH)
   - hping3 -S -p 22 --flood --rand-source ... mengirim banyak paket SYN dari IP sumber acak.
   - Di server, netstat -nat | grep :22 menunjukkan puluhan koneksi berstatus SYN_RECV. Backlog koneksi penuh, membuat koneksi SSH baru sangat lambat atau gagal. Ini adalah contoh klasik TCP SYN flood berbasis IP spoofing.

c. UDP Flood (teardrop+spoofing dalam praktikum)
   - hping3 --udp -d 100 --flood ... mengirim banyak paket UDP berpayload 100 byte ke server.
   - Di Wireshark terlihat arus besar paket UDP dari attacker ke server. Volume tinggi paket ini dapat membebani proses penanganan paket/fragmentasi dan mengganggu layanan lain (model DoS pada layer transport).

d. Land Attack
   - hping3 -S -p 22 -a 192.168.132.132 192.168.132.132 –flood mengirim paket SYN ke server dengan source IP yang sama dengan destination (server “menyerang diri sendiri”).
   - Di Wireshark, source dan destination IP sama. Hal ini membuat server mencoba membalas ke dirinya sendiri berulang-ulang, berpotensi menguras resource dan membuat layanan SSH tidak responsif.

3. Pelajaran Utama dari Praktikum

a. Pemetaan ARP sangat krusial: begitu tabel ARP berhasil dimanipulasi, attacker dapat masuk sebagai man-in-the-middle dan mengontrol/menyadap trafik pada LAN.

b. Protokol tanpa enkripsi (Telnet) sangat rentan: kredensial dan data aplikasi terlihat jelas di sniffing, sehingga mudah disadap dan sesi bisa di-hijack.

c. Protokol terenkripsi (SSH) lebih tahan terhadap sniffing: tetapi tetap bisa terganggu layanannya oleh serangan DoS (SYN flood, land attack) karena masalahnya bukan pada kerahasiaan data, tapi pada ketersediaan layanan.

d. IP spoofing mempermudah DoS: dengan memalsukan IP sumber, attacker:
   - Menyulitkan pelacakan sumber serangan.
   - Dapat membuat banyak koneksi "palsu" dari berbagai alamat, membanjiri port layanan (22, ICMP, UDP) dan menurunkan ketersediaan server.

e. Secara keseluruhan, hasil analisis kami menunjukkan bahwa:
   - Pada lapisan data-link (ARP spoofing), okus serangan adalah penyadapan dan pengambilalihan sesi.
   - Pada lapisan jaringan/transport (IP spoofing + flood), fokusnya adalah menghabiskan resource dan menjatuhkan ketersediaan layanan (DoS).