PRAKTIKUM NETWORK SECURITY (SNIFFING, SPOOFING DAN SESSION HIJACKING)

Nama Kelompok : Iryan Tegar (105841113023)
Andini Febrianti (105841113223)
Putri Amelia Nur (105841114423)
Kelas : 5A Advanced Network Security And Protocol

1. PENDAHULUAN

Keamanan jaringan komputer merupakan aspek penting dalam menjaga kerahasiaan, integritas, dan ketersediaan data. Namun, pada praktiknya masih banyak protokol jaringan dasar yang tidak memiliki mekanisme keamanan bawaan, seperti ARP (Address Resolution Protocol) dan IP (Internet Protocol). Kondisi ini membuka peluang bagi penyerang untuk melakukan manipulasi identitas jaringan, salah satunya melalui teknik ARP Spoofing dan IP Spoofing.

Praktikum ini dilakukan untuk memahami secara langsung bagaimana serangan ARP Spoofing dapat digunakan sebagai pintu masuk untuk session hijacking, khususnya pada protokol yang tidak terenkripsi seperti Telnet. Selain itu, praktikum ini juga mengeksplorasi berbagai metode IP Spoofing, seperti Ping of Death, SYN Flood, Teardrop, dan Land Attack, yang berfokus pada gangguan layanan (Denial of Service) di layer transport dan network.

Dengan melakukan simulasi serangan pada lingkungan jaringan lokal berbasis virtual machine, diharapkan mahasiswa dapat melihat perbedaan perilaku jaringan antara protokol aman (SSH) dan protokol tidak aman (Telnet), serta memahami pentingnya penerapan mekanisme mitigasi keamanan jaringan secara menyeluruh.

2. Tujuan praktikum

Tujuan dari praktikum ARP Spoofing, Session Hijacking, dan IP Spoofing ini adalah sebagai berikut:

a. Memahami konsep dasar ARP Spoofing dan bagaimana serangan tersebut dapat memanipulasi tabel ARP pada client dan server.

b. Menganalisis proses terjadinya session hijacking pada komunikasi client–server menggunakan protokol Telnet.
c. Membandingkan keamanan protokol Telnet dan SSH berdasarkan hasil sniffing lalu lintas jaringan menggunakan Wireshark.
d. Menguji dan mengamati dampak berbagai metode IP Spoofing seperti Ping of Death, SYN Flood, Teardrop, dan Land Attack terhadap kinerja sistem target.
e. Mengidentifikasi protokol dan layer jaringan yang paling rentan terhadap serangan spoofing.
f. Mengetahui dan memahami metode mitigasi yang dapat diterapkan untuk mencegah serangan ARP Spoofing dan IP Spoofing pada jaringan komputer.

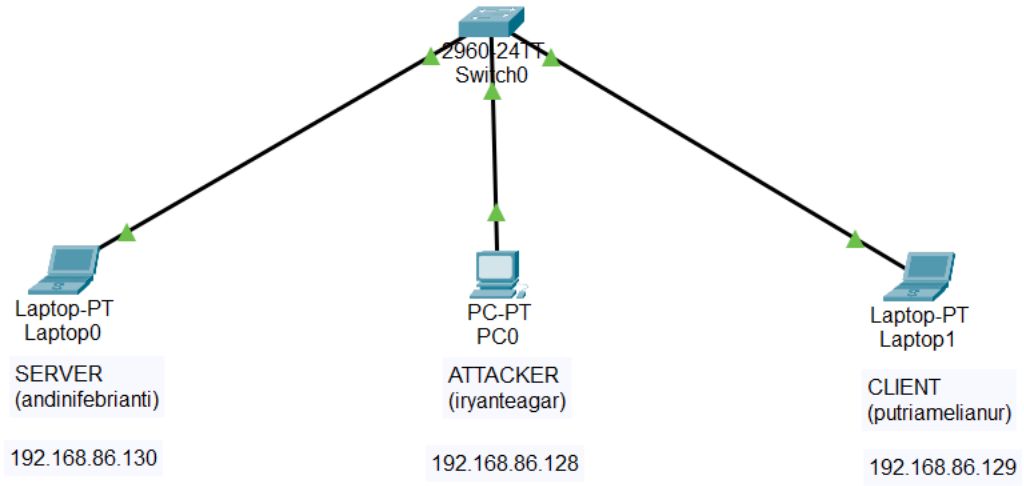
3. Tools yang digunakan

Kategori	**Nama Tools / Layanan**	**Fungsi / Kegunaan**
Perangkat Lunak	VMware Workstation	Platform virtualisasi untuk menjalankan beberapa mesin virtual (server, client, dan attacker).
Perangkat Lunak	Ubuntu Server	Berfungsi sebagai server target yang menjalankan layanan Telnet dan SSH.
Perangkat Lunak	Ubuntu Desktop	Digunakan sebagai client untuk melakukan koneksi ke server.
Perangkat Lunak	Kali Linux	Digunakan sebagai mesin attacker untuk melakukan ARP Spoofing, IP Spoofing, dan session hijacking.
Perangkat Lunak	Wireshark	Digunakan untuk melakukan sniffing dan analisis lalu lintas jaringan.

Perangkat Lunak	Hping3	Digunakan untuk melakukan serangan IP Spoofing seperti Ping of Death, SYN Flood, dan LAND Attack.
Perangkat Lunak	Netcat (nc)	Digunakan untuk simulasi session hijacking dan pembuatan koneksi backdoor.
Layanan Jaringan	Telnet	Digunakan untuk pengujian komunikasi client–server yang tidak terenkripsi.
Layanan Jaringan	SSH (Secure Shell)	Digunakan untuk pengujian komunikasi client–server yang terenkripsi.

4. ARP SPOOFING & SESSION HIJACKING

a. Gambar topologi jaringan beserta dengan IP Addressnya



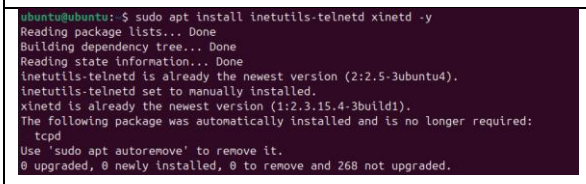
Gambar 1.1 Topologi Jaringan untuk Pengujian ARP Spoofing (Client–Server–Attacker)

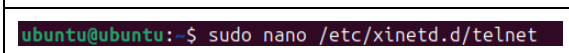
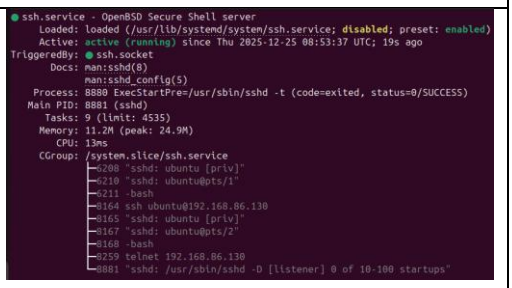
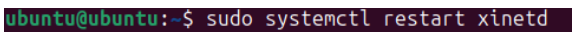
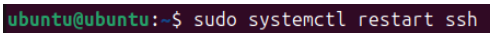
Gambar ini memperlihatkan topologi jaringan yang digunakan pada pengujian ARP Spoofing, di mana seluruh perangkat terhubung dalam satu jaringan lokal melalui

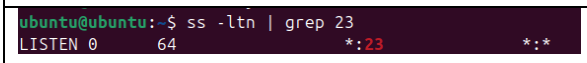
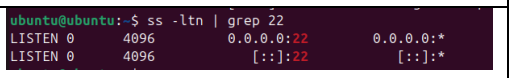
sebuah switch. Terdapat tiga host utama, yaitu server dengan alamat IP 192.168.86.130, client dengan alamat IP 192.168.86.129, dan attacker dengan alamat IP 192.168.86.128. Server dan client berfungsi sebagai pihak yang melakukan komunikasi layanan (Telnet dan SSH), sedangkan attacker berada di tengah jaringan dengan tujuan melakukan penyadapan lalu lintas data. Topologi ini memungkinkan attacker melakukan manipulasi tabel ARP sehingga komunikasi antara client dan server dapat dialihkan dan diamati, yang menjadi dasar untuk proses ARP spoofing dan session hijacking.

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

- **Ubuntu server (andinifebrianti)**

telnet	**ssh**
 *Gambar 1.2 Proses Instalasi Layanan* Telnet pada Server Ubuntu Gambar ini menunjukkan proses instalasi layanan Telnet pada sistem operasi Ubuntu menggunakan perintah sudo apt install inetutils-telnetd xinetd -y. Berdasarkan output terminal, dapat dilihat bahwa paket inetutils-telnetd dan xinetd sudah terpasang pada sistem, sehingga tidak dilakukan instalasi ulang.	 *Gambar 1.3 Proses Instalasi dan* Konfigurasi Layanan SSH pada Server Ubuntu Gambar ini menampilkan proses instalasi ulang layanan SSH pada server Ubuntu menggunakan perintah sudo apt install --reinstall openssh-server -y. Dari output terminal terlihat bahwa paket openssh-server berhasil diunduh, dipasang, dan dikonfigurasi tanpa adanya error. Layanan SSH ini digunakan sebagai media komunikasi yang lebih aman antara client dan server. Instalasi SSH diperlukan untuk melakukan pengujian lanjutan, khususnya dalam mengamati

	perbedaan perilaku komunikasi jaringan sebelum dan sesudah dilakukan ARP spoofing serta proses session hijacking pada protokol yang terenkripsi.
 Gambar 1.4 Proses Pengeditan Konfigurasi Telnet melalui Xinetd Gambar ini menunjukkan perintah sudo nano /etc/xinetd.d/telnet yang digunakan untuk membuka dan mengedit file konfigurasi layanan Telnet pada sistem Ubuntu. Pengaturan ini dilakukan untuk memastikan layanan Telnet aktif dan dapat menerima koneksi dari komputer client. Tahap konfigurasi ini penting sebelum melakukan pengujian ARP spoofing, karena Telnet akan menjadi salah satu protokol yang digunakan untuk mengamati proses penyadapan komunikasi dan session hijacking oleh attacker.	 *Gambar 1.5 Status Layanan SSH dan Koneksi Aktif antara Client dan Server* Gambar ini menampilkan hasil pengecekan status layanan SSH menggunakan perintah service ssh status. Dari output terlihat bahwa layanan OpenBSD Secure Shell (SSH) berada dalam kondisi active (running), yang menandakan bahwa server siap menerima koneksi SSH.
 *Gambar 1.6 Restart Layanan Xinetd pada Server Ubuntu* Gambar ini menunjukkan proses me-restart layanan xinetd pada sistem operasi Ubuntu menggunakan perintah sudo systemctl restart xinetd. Layanan xinetd (Extended	 *Gambar 1.7 Restart Layanan SSH pada Server Ubuntu* Gambar ini memperlihatkan proses me-restart layanan SSH (Secure Shell) pada sistem operasi Ubuntu menggunakan perintah sudo systemctl

Internet Services Daemon) berfungsi sebagai *super-server* yang mengelola berbagai layanan jaringan berbasis TCP/IP, seperti Telnet. Restart layanan ini dilakukan setelah instalasi atau perubahan konfigurasi layanan jaringan agar pengaturan terbaru dapat diterapkan. Pada konteks praktikum ARP Spoofing, langkah ini diperlukan untuk memastikan layanan Telnet/SSH pada server berjalan dengan baik sehingga koneksi dari client dapat diuji sebelum dan sesudah dilakukan serangan.	restart ssh. Layanan SSH berfungsi untuk menyediakan akses jarak jauh yang aman melalui mekanisme enkripsi antara client dan server.
 *Gambar 1.8 Pemeriksaan Status Layanan Telnet pada Server* Gambar ini menunjukkan hasil perintah ss -ltn | grep 23 yang digunakan untuk memeriksa status port Telnet (port 23) pada server Ubuntu. Output LISTEN menandakan bahwa server sedang mendengarkan koneksi masuk pada port 23, yang berarti layanan Telnet telah aktif dan siap menerima koneksi dari client.	 *Gambar 1.9 Pemeriksaan Status Layanan SSH pada Server* Gambar ini menampilkan hasil perintah ss -ltn | grep 22 yang digunakan untuk mengecek status port SSH (port 22) pada server Ubuntu. Output LISTEN pada alamat IPv4 (0.0.0.0:22) dan IPv6 ([::]:22) menunjukkan bahwa layanan SSH aktif dan siap menerima koneksi dari client melalui kedua protokol jaringan.

Tabel 1.1 Hasil Instalasi, Konfigurasi, dan Status Layanan Telnet dan SSH pada Server Ubuntu

Tabel ini menyajikan rangkuman hasil tahapan instalasi, konfigurasi, serta pengujian status layanan Telnet dan SSH pada server berbasis sistem operasi Ubuntu.

Proses instalasi Telnet dilakukan menggunakan paket *inetutils-telnetd* dan *xinetd*, sedangkan layanan SSH dipasang dan dikonfigurasi menggunakan *openssh-server*.

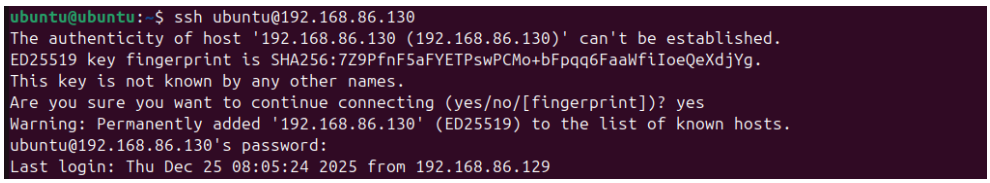
- **Ubuntu client (putri amelia nur)**

telnet

Gambar 1.10 Koneksi Telnet Berhasil dari Client ke Server

Gambar ini menunjukkan proses koneksi Telnet dari komputer client ke server dengan alamat IP 192.168.86.130. Setelah perintah telnet 192.168.86.130 dijalankan, sistem menampilkan status *Connected*, yang menandakan bahwa koneksi berhasil terjalin. Client kemudian melakukan login menggunakan akun pengguna Ubuntu hingga berhasil masuk ke sistem server.

Ssh



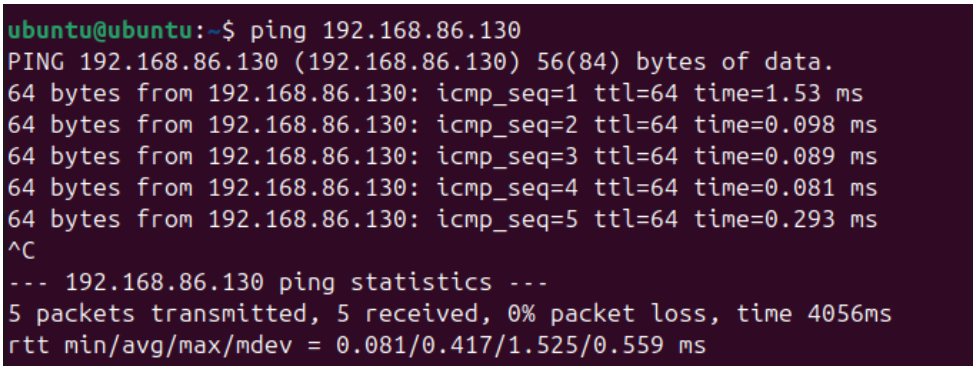
Gambar 1.11 Uji Koneksi SSH dari Client ke Server pada Jaringan Lokal

Gambar menunjukkan proses koneksi Secure Shell (SSH) yang dilakukan dari komputer client menuju server dengan alamat IP 192.168.86.130. Saat koneksi pertama kali dilakukan, sistem menampilkan peringatan bahwa authenticity host belum dikenali, sehingga pengguna diminta untuk memverifikasi fingerprint kunci SSH (ED25519) milik server.

Tabel 1.2 Hasil Pengujian Koneksi Telnet dan SSH dari Client ke Server

Tabel ini menyajikan hasil pengujian koneksi jaringan antara komputer client dan server menggunakan dua protokol komunikasi, yaitu Telnet dan Secure Shell (SSH), pada jaringan lokal. Pengujian Telnet dilakukan dengan perintah telnet 192.168.86.130, yang menunjukkan status Connected serta keberhasilan proses login ke sistem server menggunakan akun pengguna Ubuntu. Hal ini menandakan bahwa layanan Telnet pada server telah berjalan dengan baik dan dapat diakses oleh client.

Proses ping (uji koneksi)



Gambar 1.12 Pengujian Konektivitas Jaringan Client ke Server Menggunakan Perintah Ping (ICMP)

Gambar memperlihatkan hasil pengujian konektivitas jaringan dari komputer client ke server dengan alamat IP 192.168.86.130 menggunakan perintah ping.

Hasil pengujian menunjukkan bahwa seluruh paket ICMP yang dikirim (5 paket) berhasil diterima kembali oleh server dengan packet loss sebesar 0%, yang menandakan koneksi jaringan berjalan normal.

c. Catat MAC Address dari komputer client dan server

	Ip address	**Mac addres**
Client	192.168.86.129	00:0c:29:BE:49:8F
Server	192.168.86.130	00:0c:29:C8:8F:84

Tabel 1.4 Data IP Address dan MAC Address Komputer Client dan Server

Tabel ini menampilkan hasil pencatatan IP Address dan MAC Address dari komputer client dan server pada jaringan lokal sebelum dilakukan serangan ARP Spoofing. Data ini diperoleh untuk mengetahui pemetaan awal antara alamat IP dan alamat fisik (MAC) pada masing-masing perangkat jaringan.

d. Catat MAC Address setelah dilakukan arp spoofing .bandingkan dengan MAC address sebelumnya.

	Sebelum	**Sesudah**
Client	00:0c:29:BE:49:8F	00:0c:29:a4:39:e3
Server	00:0c:29:C8:8F:84	00:0c:29:a4:39:e3

Tabel 1.5 Perbandingan MAC Address Client dan Server Sebelum dan Sesudah ARP Spoofing

Tabel ini menyajikan perbandingan MAC Address pada komputer client dan server sebelum dan sesudah dilakukan serangan ARP Spoofing. Data ini digunakan untuk mengidentifikasi perubahan pemetaan alamat IP ke alamat MAC yang terjadi akibat manipulasi protokol ARP oleh komputer attacker.

e. proses terjadinya session hijacking

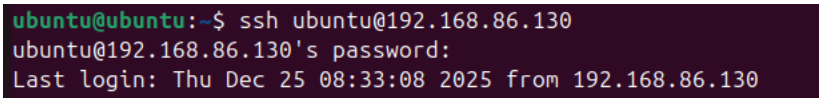
1. Telnet client-server

- Client telnet

Gambar 1.13 Uji Koneksi Telnet dari Client ke Server pada Jaringan Lokal

Gambar menunjukkan proses koneksi Telnet yang dilakukan dari komputer client ke server dengan alamat IP 192.168.86.130. Koneksi berhasil dibangun, ditandai dengan pesan *"Connected to 192.168.86.130"* dan munculnya informasi sistem operasi server Ubuntu 24.04.3 LTS.

- Client ssh

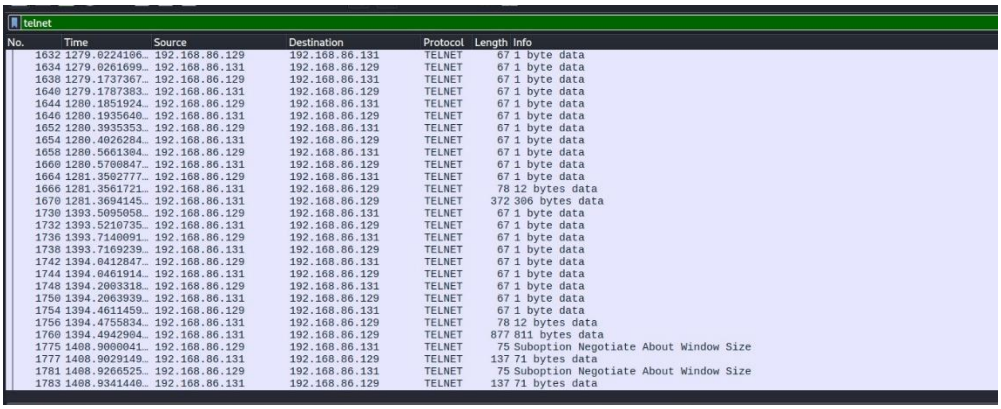


Gambar 1.14 Autentikasi dan Akses Server Menggunakan SSH dari Client

Gambar memperlihatkan proses login ke server menggunakan protokol Secure Shell (SSH) dari komputer client ke alamat IP 192.168.86.130. Setelah memasukkan password user *ubuntu*, client berhasil masuk ke sistem server, yang ditandai dengan munculnya informasi Last login beserta waktu dan alamat IP asal koneksi.

2. Amati pada komputer attacker (wireshark)

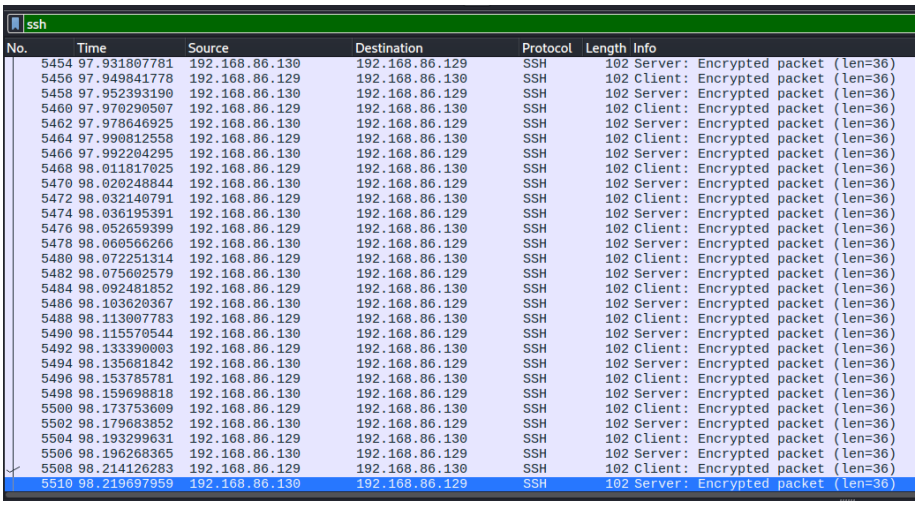
- telnet



Gambar 1.15 Hasil Sniffing Lalu Lintas Telnet Menggunakan Wireshark Setelah ARP Spoofing

Gambar menampilkan hasil penangkapan paket jaringan (sniffing) menggunakan Wireshark pada protokol Telnet antara komputer client (192.168.86.129) dan server (192.168.86.131). Terlihat sejumlah paket Telnet dengan pertukaran data dua arah (*byte data*) yang mengindikasikan adanya sesi komunikasi aktif.

- ssh



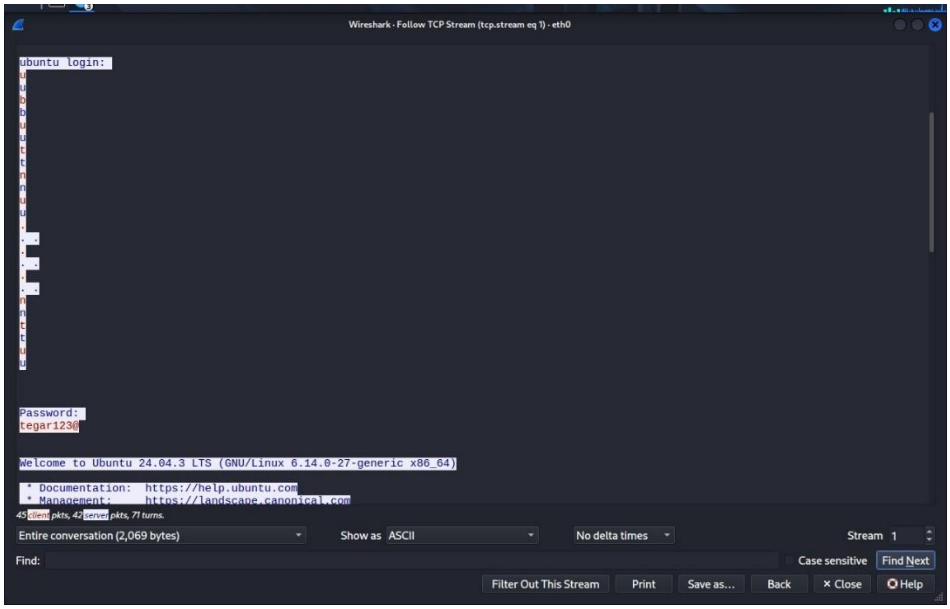
Gambar 1.16 Monitoring Lalu Lintas SSH antara Client dan Server Menggunakan Wireshark

Gambar ini menampilkan hasil pemantauan lalu lintas jaringan menggunakan aplikasi *Wireshark* pada saat komunikasi SSH (Secure Shell)

antara komputer client dengan alamat IP 192.168.86.129 dan server dengan alamat IP 192.168.86.130. Paket-paket yang tertangkap ditandai dengan protokol SSH dan berstatus *Encrypted packet*, yang menunjukkan bahwa data yang dikirimkan telah dienkripsi.

3. proses pencurian username dan password

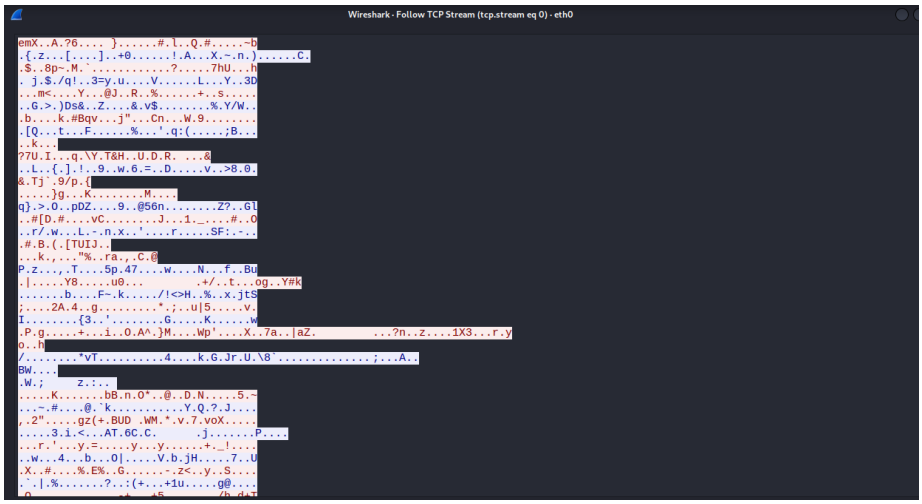
- Telnet



Gambar 1.17 Hasil Session Hijacking pada Koneksi Telnet Menggunakan Fitur Follow TCP Stream di Wireshark

Gambar ini menunjukkan hasil analisis koneksi Telnet antara client dan server yang ditangkap menggunakan aplikasi *Wireshark* melalui fitur Follow TCP Stream. Pada tampilan tersebut terlihat secara jelas proses autentikasi Telnet, mulai dari permintaan login, pengisian username, hingga password yang dikirimkan dalam bentuk *plain text*.

- ssh

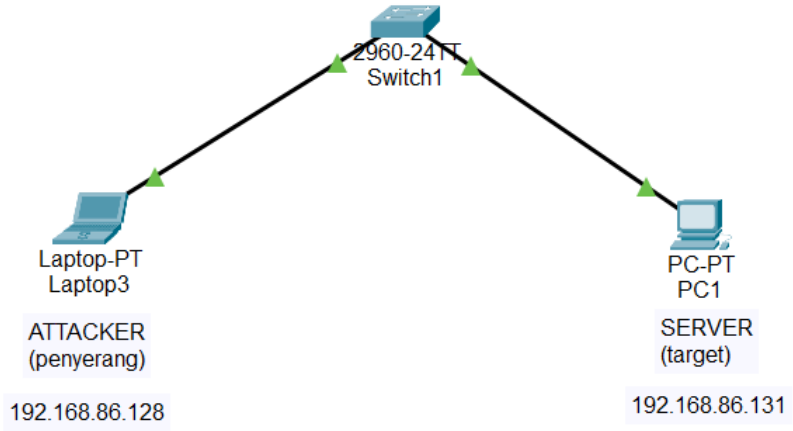


Gambar 1.18 Hasil Analisis Follow TCP Stream pada Koneksi SSH Setelah Session Hijacking

Gambar ini menampilkan hasil analisis lalu lintas SSH (Secure Shell) menggunakan fitur Follow TCP Stream pada aplikasi *Wireshark*. Berbeda dengan hasil pemantauan pada protokol Telnet, data yang ditampilkan pada koneksi SSH terlihat dalam bentuk karakter acak dan tidak terbaca (*ciphertext*). Hal ini disebabkan karena seluruh komunikasi SSH telah dienkripsi menggunakan mekanisme kriptografi yang aman.

5. IP SPOOFING

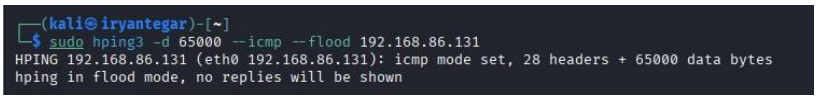
a. Topologi



Gambar 2.1 Topologi Jaringan Pengujian ARP Spoofing dan Session Hijacking

b. Jalankan beberapa tool ip spoofing dan catat apa yang terjadi

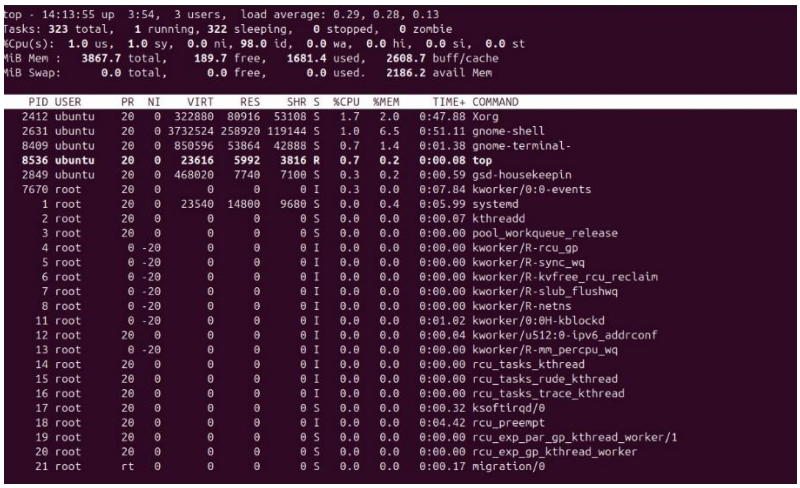
- **Pod Spoofing (Ping of Death)**



Gambar 2.2 Eksekusi Serangan Ping of Death (PoD) menggunakan Tool Hping3

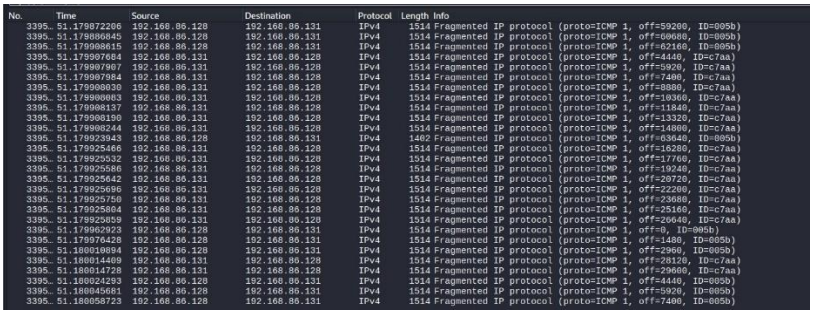
Gambar tersebut menunjukkan terminal pada sistem operasi Kali Linux yang sedang menjalankan serangan Denial of Service (DoS) terhadap target dengan alamat IP 192.168.86.131. Berikut adalah detail teknis dari perintah yang dijalankan:

- Tool yang Digunakan: hping3, sebuah utilitas baris perintah yang mampu merakit dan mengirim paket TCP/IP kustom untuk pengujian keamanan jaringan.
- Target Serangan: Komputer dengan IP Address 192.168.86.131.
- Parameter -d 65000: Menunjukkan bahwa attacker mengirimkan paket dengan ukuran data sebesar 65.000 bytes. Ukuran ini sangat besar dan melebihi batas normal paket ICMP, yang merupakan ciri khas dari serangan Ping of Death.
- Parameter --icmp: Menginstruksikan tool untuk menggunakan protokol ICMP (Internet Control Message Protocol) atau biasa dikenal dengan paket "ping".
- Parameter --flood: Mengaktifkan mode "banjir", di mana paket dikirimkan secepat mungkin tanpa menunggu balasan (reply) dari target. Hal ini bertujuan untuk menghabiskan *bandwidth* jaringan dan sumber daya CPU pada target.
- Status Terminal: Keterangan *"hping in flood mode, no replies will be shown"* mengonfirmasi bahwa serangan sedang berlangsung secara agresif, sehingga sistem tidak menampilkan balasan satu per satu agar proses pengiriman tetap maksimal.



Gambar 2.3 Monitoring Resource System

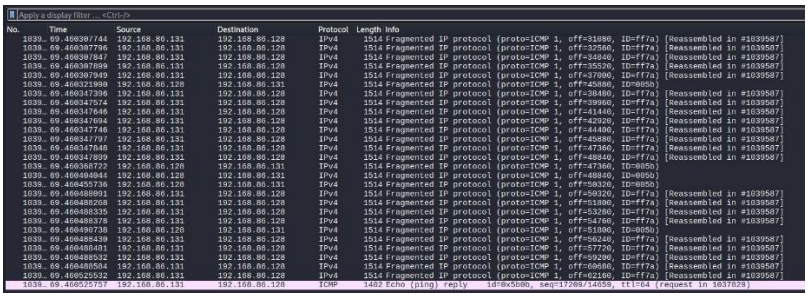
Gambar menampilkan tampilan perintah top di terminal Ubuntu yang digunakan untuk memantau proses yang sedang berjalan di server, termasuk informasi CPU, memori, dan swap. Baris-baris di bagian bawah menunjukkan daftar proses beserta PID, user, penggunaan CPU dan memori, yang dapat membantu menganalisis beban sistem saat dilakukan serangan atau pengujian keamanan jaringan.



Gambar 2.4 Lalu Lintas Paket Terfragmentasi pada Serangan Teardrop dengan IP Spoofing

Gambar menampilkan daftar paket IPv4 terfragmentasi pada aplikasi analisis jaringan (misalnya Wireshark) dengan banyak frame dari sumber 192.168.88.128 menuju tujuan 192.168.88.131 yang menggunakan protokol ICMP. Informasi panjang paket dan offset fragment menunjukkan adanya pengiriman fragmentasi IP yang tidak normal, yang merupakan karakteristik

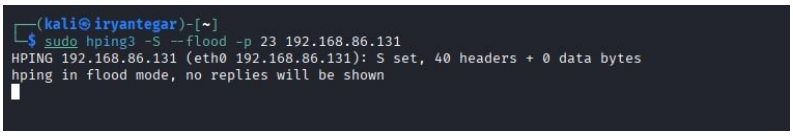
serangan teardrop dengan pemalsuan IP untuk mengganggu atau melumpuhkan host tujuan



Gambar 2.5 Reassembly Paket Fragmentasi ICMP pada Serangan Teardrop

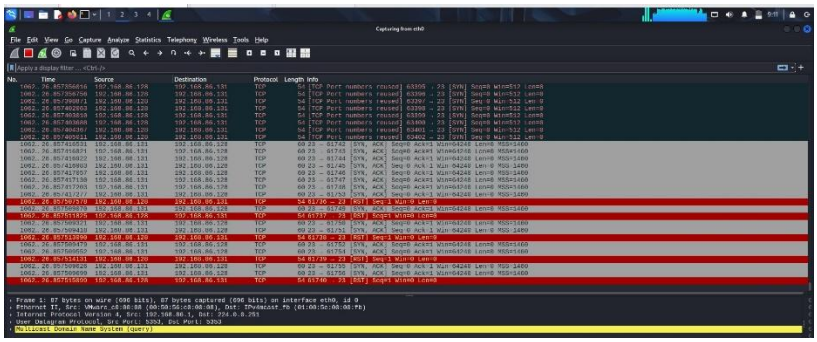
Gambar menunjukkan daftar paket IPv4 terfragmentasi dari sumber 192.168.88.131 ke tujuan 192.168.88.128 yang dianalisis dengan Wireshark, dengan kolom tambahan "Reassembled in …" yang menandakan proses penyusunan ulang fragment. Di bagian bawah tampak paket ICMP Echo (ping) reply yang berhasil direkonstruksi dari beberapa fragment, yang menggambarkan bagaimana host korban harus memproses fragmentasi tidak normal selama serangan teardrop.

- **SYN Flood**



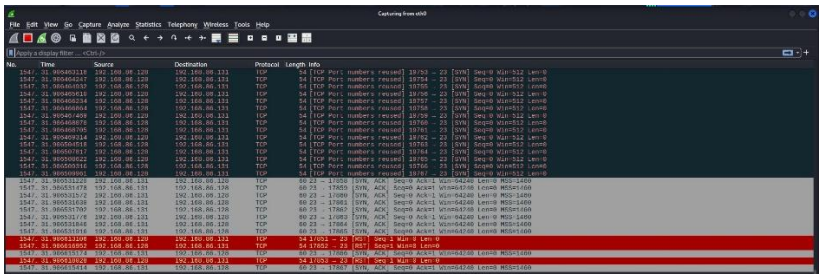
Gambar 2.6 Perintah hping3 untuk Serangan SYN Flood ke Port Telnet

Gambar menampilkan terminal Kali Linux yang menjalankan perintah sudo hping3 -S --flood -p 23 192.168.88.131 untuk mengirim paket TCP SYN secara terus-menerus ke port 23 (Telnet) server 192.168.88.131. Mode flood pada hping3 membuat server dibanjiri permintaan koneksi semu sehingga dapat menyebabkan penurunan kinerja atau penolakan layanan pada layanan Telnet.



Gambar 2.7 Capture Serangan SYN Flood ke Port Telnet di Wireshark

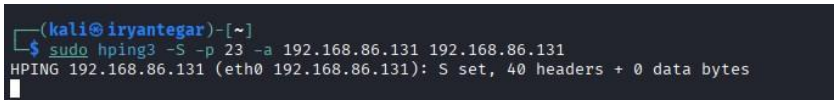
Gambar menampilkan hasil capture Wireshark yang memperlihatkan banyak paket TCP dari host attacker ke server 192.168.88.131 dengan flag SYN menuju port 23/Telnet secara beruntun. Baris-baris yang diberi highlight merah menunjukkan banjir paket SYN tanpa proses three-way handshake yang lengkap, yang menjadi ciri serangan SYN flood dan dapat menyebabkan tabel koneksi server penuh sehingga layanan sulit diakses



Gambar 2.8 Respon Server terhadap Serangan SYN Flood di Wireshark

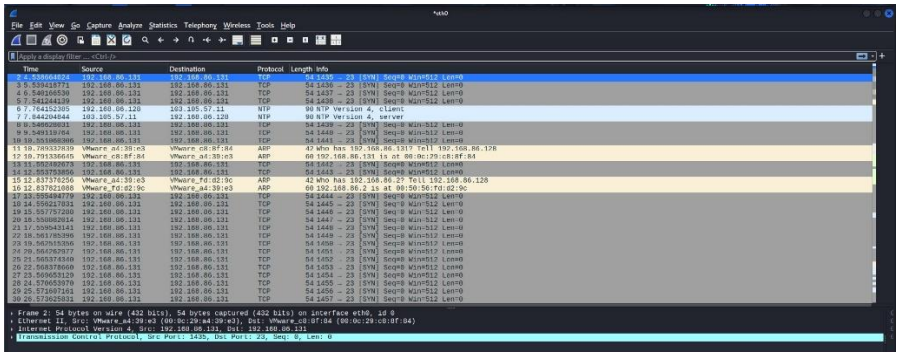
Gambar memperlihatkan hasil capture Wireshark yang menampilkan aliran paket TCP antara attacker dan server 192.168.88.131 pada port 23/Telnet, dengan beberapa paket yang di-highlight merah sebagai deretan SYN dari attacker. Di bagian lain tampak paket RST atau ACK dari server yang menunjukkan usaha server merespons atau memutus koneksi setengah terbuka, sehingga mengilustrasikan dampak serangan SYN flood terhadap proses koneksi TCP.

- **Land Attack**



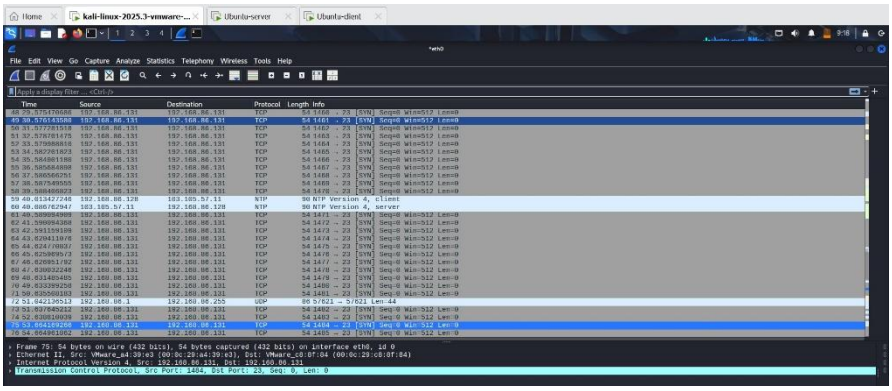
Gambar 2.9 Simulasi LAND Attack dengan hping3 pada Port Telnet

Gambar menampilkan terminal Kali Linux yang menjalankan perintah sudo hping3 -S -p 23 -a 192.168.86.131 192.168.86.131, yaitu pengiriman paket TCP SYN ke port 23/Telnet dengan alamat sumber dan tujuan sama (IP server). Teknik ini dikenal sebagai LAND attack, di mana server dipaksa memproses paket yang "berdialog dengan dirinya sendiri", sehingga dapat mengakibatkan gangguan atau penurunan kinerja layanan jaringan.



Gambar 2.10 Capture LAND Attack di Wireshark dengan IP Sumber dan Tujuan Sama

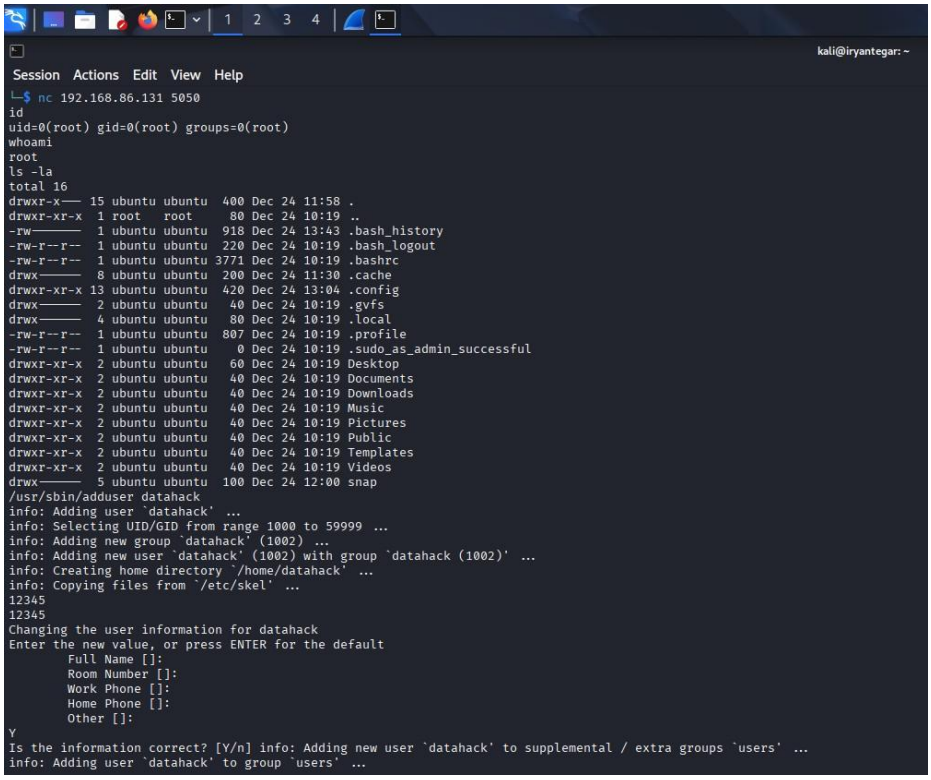
Gambar menunjukkan hasil capture Wireshark yang menampilkan paket TCP dari sumber 192.168.86.131 ke tujuan 192.168.86.131 pada port 23/Telnet, sehingga IP sumber dan tujuan identik. Pola ini sesuai karakteristik LAND attack, di mana host korban menerima paket yang seolah datang dari dirinya sendiri, yang dapat memicu kebingungan pada stack TCP/IP dan berpotensi mengganggu kestabilan layanan jaringan



Gambar 2.11 Lalu Lintas Normal TCP Telnet antara Client dan Server

Gambar menampilkan capture Wireshark yang memperlihatkan alur koneksi TCP dari client 192.168.88.128 ke server 192.168.88.131 pada port 23/Telnet dengan pola three-way handshake dan pertukaran data yang terlihat berurutan. Tampak paket SYN, SYN-ACK, dan ACK diikuti segmen data Telnet yang menandakan sesi komunikasi client-server berjalan normal sebelum atau di luar aktivitas serangan spoofing

- **Backdoor + spoofing**



Gambar 2.12 Session Hijacking Telnet dengan Netcat sebagai Attacker

Gambar menunjukkan terminal Kali Linux yang terhubung ke server 192.168.86.131 melalui netcat pada port 5050, di mana prompt sudah berada sebagai user root dan perintah whoami, ls -la, dan adduser datahack dapat dijalankan. Kondisi ini menggambarkan keberhasilan session hijacking, karena attacker dapat mengambil alih sesi dan menjalankan perintah administratif di server seolah-olah sebagai pengguna sah.

Gambar 2.13 Verifikasi Akun Baru Hasil Session Hijacking di File /etc/passwd

Gambar memperlihatkan perintah cat /etc/passwd | grep datahack yang menampilkan entri user datahack:x:1002:1002:...:/home/datahack:/bin/bash di sistem. Tampilan ini menjadi bukti bahwa akun datahack berhasil dibuat pada server oleh attacker melalui sesi yang telah di-hijack, sehingga memperkuat dokumentasi keberhasilan serangan session hijacking.

6. ANALISIS TRANSPORT LAYER DAN PROTOKOL

Dalam praktikum IP Spoofing yang telah dilakukan, sebagian besar serangan (seperti *SYN Flood* dan *Land Attack*) memanfaatkan protokol TCP (Transmission Control Protocol) pada Layer Transport.

Hal ini dikarenakan karakteristik TCP yang bersifat Connection-Oriented (berorientasi koneksi) dan Stateful. Untuk memulai komunikasi, TCP mewajibkan proses *Three-Way Handshake* (SYN -> SYN-ACK -> ACK). Serangan spoofing mengeksploitasi mekanisme ini dengan memalsukan IP pengirim pada tahap awal (SYN). Karena server wajib "mengingat" permintaan koneksi tersebut dan menunggu balasan, penyerang dapat membanjiri memori server dengan permintaan palsu. Berbeda dengan UDP yang *connectionless* (kirim dan lupakan), serangan pada TCP lebih efektif membuat target mengalami kelelahan sumber daya (*resource exhaustion*).

7. PERBEDAAN METODE IP SPOOFING

Berdasarkan hasil pengujian, berikut adalah perbedaan mendasar dari metode serangan yang digunakan:

- Pod_spoofing (Ping of Death): Berfokus pada manipulasi ukuran paket. Serangan ini mengirimkan paket ICMP yang ukurannya melebihi batas maksimum protokol IP (> 65.535 bytes) untuk memicu *buffer overflow* dan membuat sistem target *crash*.
- Syn_flood: Berfokus pada manipulasi handshake TCP. Serangan ini membanjiri target dengan paket ber-flag SYN palsu tanpa pernah mengirimkan paket ACK penutup, menyebabkan memori server penuh oleh koneksi setengah terbuka (*half-open*).

- Teardrop + Spoofing: Berfokus pada manipulasi fragmentasi paket. Paket IP dikirim dalam potongan-potongan (fragmen) dengan nilai *offset* yang tumpang tindih (*overlapping*), bertujuan membingungkan sistem operasi target saat menyusun ulang paket.
- Land_attack: Berfokus pada manipulasi identitas alamat. Penyerang merekayasa paket agar IP Pengirim (*Source*) sama persis dengan IP Penerima (*Destination*), sehingga target terjebak dalam *looping* karena membalas paket ke dirinya sendiri.

8. METODE PENANGGULANGAN (MITIGASI)

Untuk mengamankan jaringan dari serangan ARP Spoofing dan IP Spoofing, dapat diterapkan langkah-langkah berikut:

a. Mitigasi ARP Spoofing (Layer 2):
   - Static ARP Entry: Mendaftarkan pemetaan IP-ke-MAC Address secara manual di tabel ARP komputer penting (Server/Gateway) agar tidak bisa ditimpa oleh attacker.
   - DHCP Snooping & Dynamic ARP Inspection: Mengaktifkan fitur keamanan pada Switch Manageable untuk memvalidasi paket ARP yang lewat.

b. Mitigasi IP Spoofing (Layer 3):
   - Ingress & Egress Filtering: Mengonfigurasi Router/Firewall untuk menolak paket masuk dari luar yang mengklaim memiliki IP internal (Ingress) dan menolak paket keluar yang IP sumbernya tidak sesuai dengan subnet lokal (Egress).
   - Enkripsi Protokol: Menggunakan layanan terenkripsi seperti SSH (pengganti Telnet) dan HTTPS (pengganti HTTP) untuk mencegah penyadapan data (*sniffing*) dan pembajakan sesi.

9. KESIMPULAN

Berdasarkan praktikum yang telah dilaksanakan, dapat disimpulkan bahwa kegiatan ini berhasil mendemonstrasikan kerentanan fatal pada protokol jaringan dasar, seperti ARP dan IP, yang tidak memiliki mekanisme autentikasi bawaan, di mana *attacker* terbukti mampu memanipulasi identitas paket untuk mengelabui korban. Secara spesifik, serangan *IP Spoofing*—terutama metode *SYN Flood* dan *Land Attack*—terbukti sangat efektif dalam mengganggu ketersediaan layanan (*availability*), hal ini terkonfirmasi melalui indikator lonjakan trafik visual pada pemantauan EtherApe serta kegagalan koneksi (*timeout*) pada target saat diuji menggunakan Netcat. Oleh karena itu, sistem keamanan jaringan tidak dapat hanya bergantung pada konfigurasi standar (*default*), melainkan mutlak memerlukan implementasi langkah mitigasi lanjut seperti penggunaan protokol terenkripsi (misalnya SSH) dan penerapan *filtering* paket di router untuk meminimalisir dampak dari serangan manipulasi identitas tersebut.