TUGAS I & II

ADVANCED NETWORK SECURITY

OLEH:

Hermi Nur Safitri	105841116223
Ummul Mu'minin	105841117323
Ririn Yulandari	105841117923

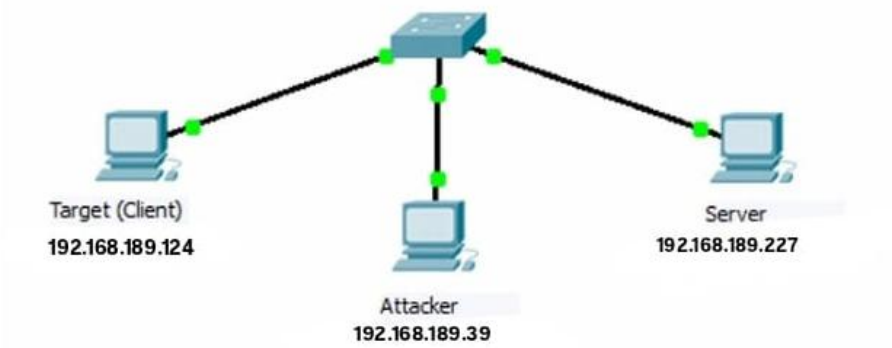
PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH MAKASSAR

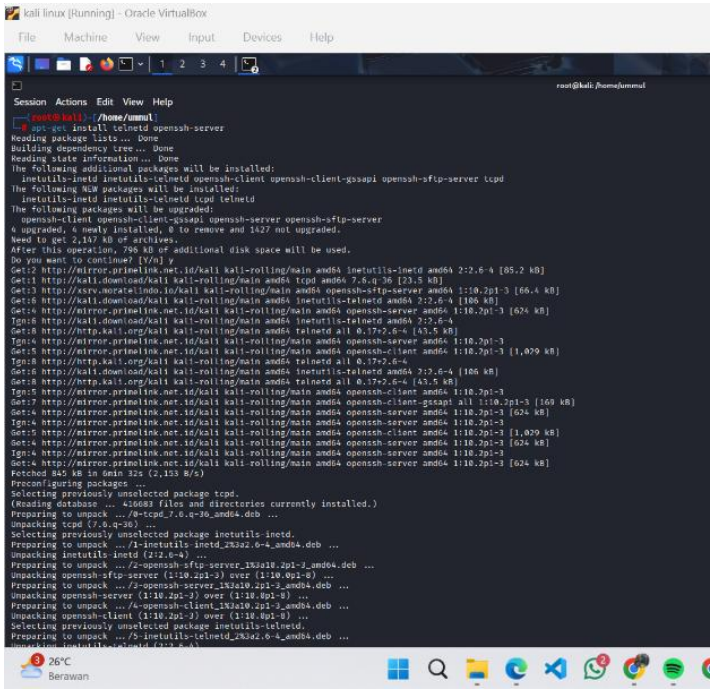
2025

1. ARP Spoofing
   a. Gambar topologi jaringan beserta dengan IP Addressnya

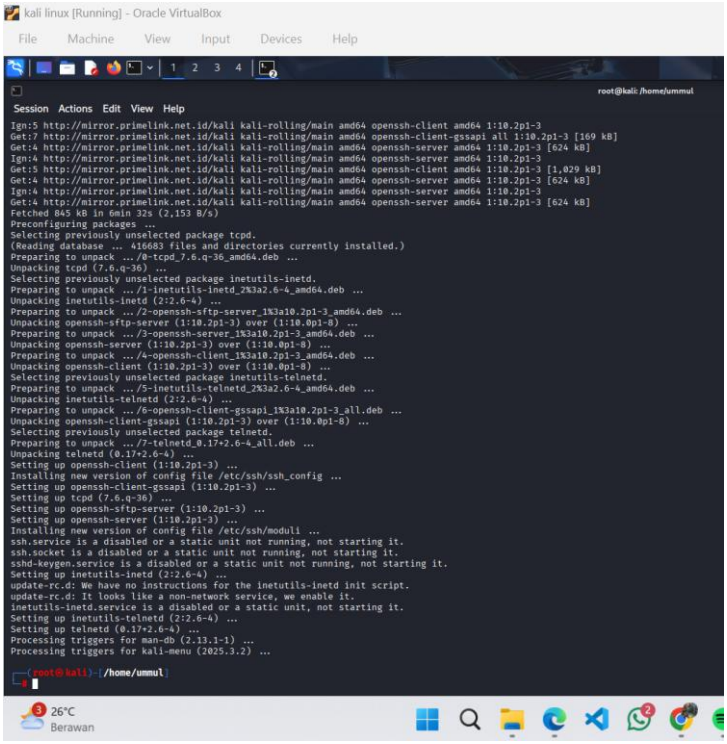


Topologi jaringan yang digunakan dalam praktikum ini menerapkan skenario serangan *Man-in-the-Middle* (MITM) yang melibatkan tiga entitas utama, yaitu Target (Client) dengan alamat IP 192.168.189.124, Server dengan alamat IP 192.168.189.227, dan Attacker dengan alamat IP 192.168.189.39. Ketiga perangkat tersebut terhubung dalam satu segmen jaringan lokal melalui sebuah *switch* pusat yang berfungsi sebagai titik distribusi data.

   b. Instalasi aplikasi telnet dan ssh



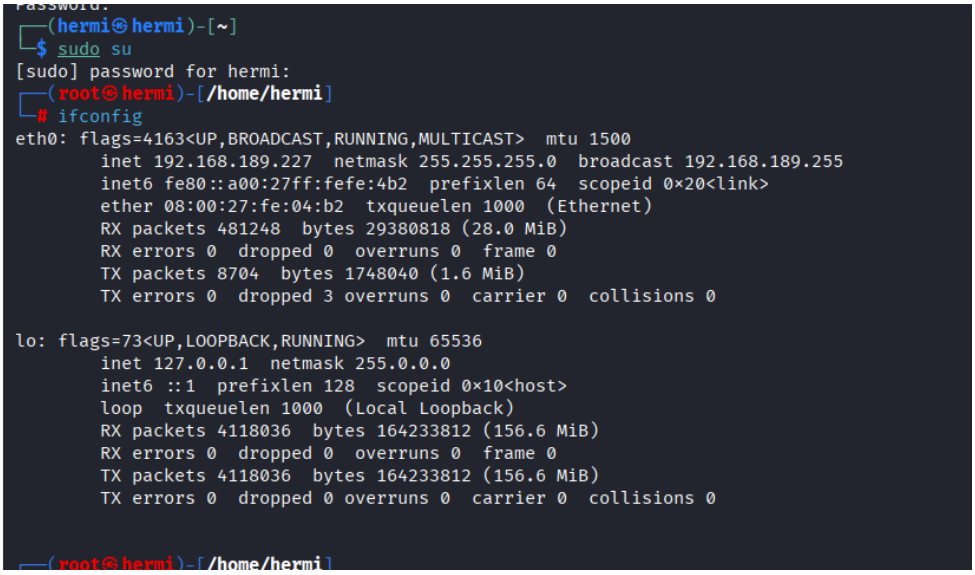
Proses konfigurasi layanan jaringan dimulai dengan melakukan instalasi paket telnetd dan openssh-server pada sistem operasi Kali Linux melalui terminal. Dengan menggunakan perintah apt-get install, sistem secara otomatis mengunduh paket-paket yang diperlukan beserta dependensinya, seperti inetutils-inetd dan tcpd, dari repositori resmi. Setelah seluruh paket berhasil diunduh, sistem melakukan proses *unpacking* dan konfigurasi awal agar layanan Telnet dan SSH siap untuk diaktifkan. Tahapan ini sangat krusial untuk menyediakan protokol komunikasi jarak jauh yang akan digunakan dalam pengujian konektivitas dan analisis keamanan jaringan pada sesi praktikum ini.



Setelah proses pengunduhan selesai, sistem melanjutkan ke tahap konfigurasi dan penyelesaian instalasi paket. Terminal menunjukkan bahwa seluruh komponen pendukung, termasuk openssh-client, openssh-server, dan telnetd, telah berhasil dikonfigurasi dan dipasang pada sistem. Proses ini juga melibatkan pembuatan file konfigurasi standar seperti /etc/ssh/ssh_config dan pembaruan pemicu (*triggers*) untuk menu sistem. Meskipun beberapa layanan seperti ssh.service dan inetutils-inetd.service terdeteksi dalam kondisi *disabled* atau tidak langsung berjalan secara otomatis setelah instalasi, seluruh paket telah terpasang dengan sukses.

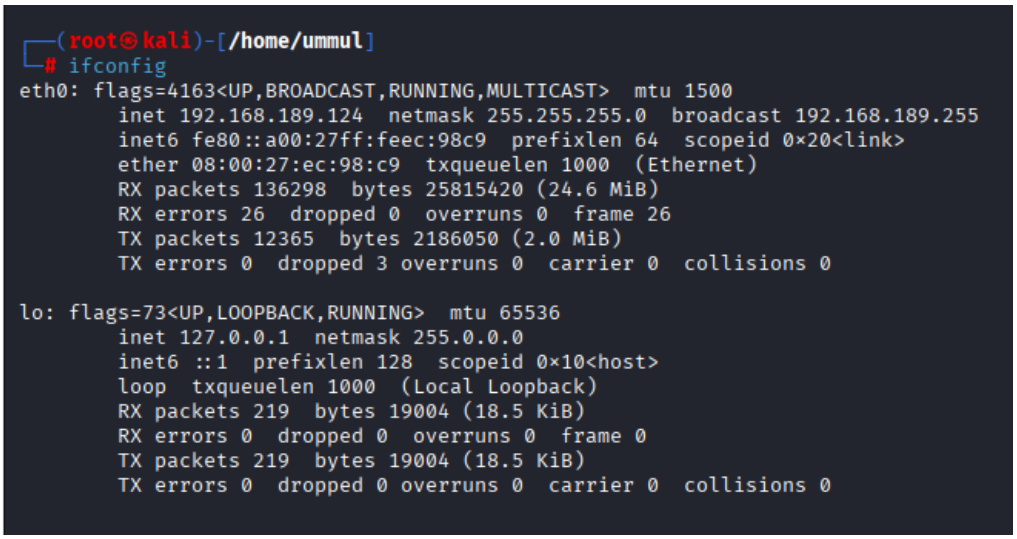
c. Catat MAC Address dari komputer client dan server

- Komputer Server (08:00:27:ec:98:c9)



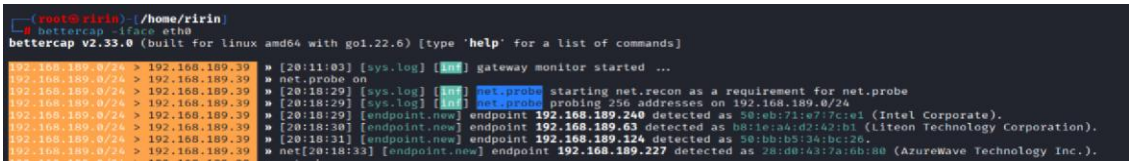
Setelah seluruh paket layanan terpasang, langkah selanjutnya adalah melakukan verifikasi identitas jaringan pada mesin Kali Linux menggunakan perintah ifconfig. Berdasarkan *output* terminal, terdeteksi bahwa antarmuka jaringan utama yaitu eth0 memiliki alamat fisik atau MAC Address 08:00:27:ec:98:c9 yang tercantum pada baris ether. Selain itu, melalui perintah ini juga dapat diketahui bahwa mesin telah memperoleh alamat IP v4 192.168.189.124 dengan netmask 255.255.255.0. Informasi ini sangat penting untuk tahap praktikum berikutnya, karena MAC address dan alamat IP tersebut akan digunakan sebagai referensi utama dalam melakukan konfigurasi tabel routing, pengalamatan statis, maupun untuk keperluan audit keamanan jaringan

- Komputer Client (08:00:27:fe:04:d2)

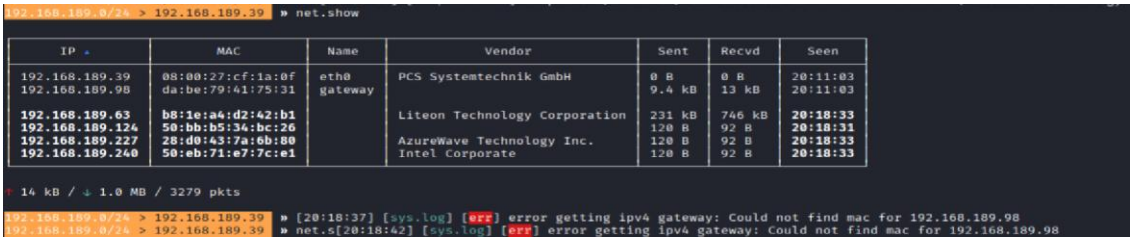


Tahap selanjutnya dilakukan pada sisi komputer client untuk memverifikasi konfigurasi jaringan yang akan digunakan dalam berinteraksi dengan server. Dengan menjalankan perintah ifconfig di terminal, diperoleh informasi bahwa antarmuka jaringan eth0 pada client memiliki alamat IP v4 192.168.189.124 dengan subnet mask 255.255.255.0. Selain itu, tercatat bahwa alamat fisik atau MAC Address untuk perangkat client ini adalah 08:00:27:ec:98:c9, yang ditemukan pada baris ether. Data pengalamatan ini memastikan bahwa client berada dalam segmen jaringan yang sama dengan server, sehingga pengujian layanan Telnet dan SSH dapat dilanjutkan untuk menguji konektivitas antar-host.

d. Proses Arp Spoofing



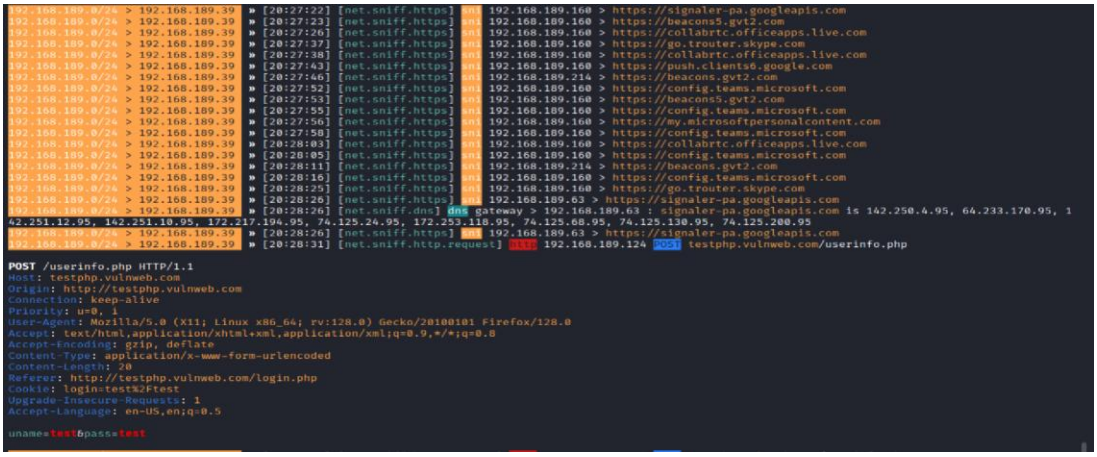
Proses ARP Spoofing dimulai dengan menjalankan *tool* Bettercap pada antarmuka eth0 menggunakan perintah bettercap -iface eth0. Modul net.probe kemudian diaktifkan untuk memindai seluruh *host* yang terhubung dalam jaringan 192.168.189.0/24. Melalui log aktivitas, sistem berhasil mengidentifikasi target pada alamat IP 192.168.189.124 beserta MAC address-nya, yang menandakan tahap persiapan *Man-in-the-Middle* (MITM) telah siap untuk dieksekusi.



Setelah proses pemindaian aktif selesai, perintah net.show dijalankan untuk menampilkan tabel inventaris seluruh perangkat yang terdeteksi di jaringan. Tabel ini menyajikan informasi detail berupa **Alamat IP**, **MAC Address**, nama *vendor* perangkat, serta statistik trafik data. Dari daftar tersebut, target utama dengan IP 192.168.189.124 telah tervalidasi keberadaannya, sehingga memudahkan penyerang untuk menentukan parameter target yang spesifik sebelum melakukan eksploitasi atau peracunan ARP.

Proses dilanjutkan dengan mengaktifkan fitur penyadapan menggunakan perintah net.sniff on dan menetapkan target spesifik melalui perintah set arp.spoof.targets 192.168.189.124. Dengan konfigurasi ini, Bettercap berhasil meracuni tabel ARP target dan mulai menangkap aktivitas trafik data secara *real-time*. Pada log terminal, terlihat sistem mampu memantau berbagai kueri DNS dan koneksi keluar dari perangkat target, yang menandakan serangan *Man-in-the-Middle* (MITM) sedang berlangsung dengan sukses.

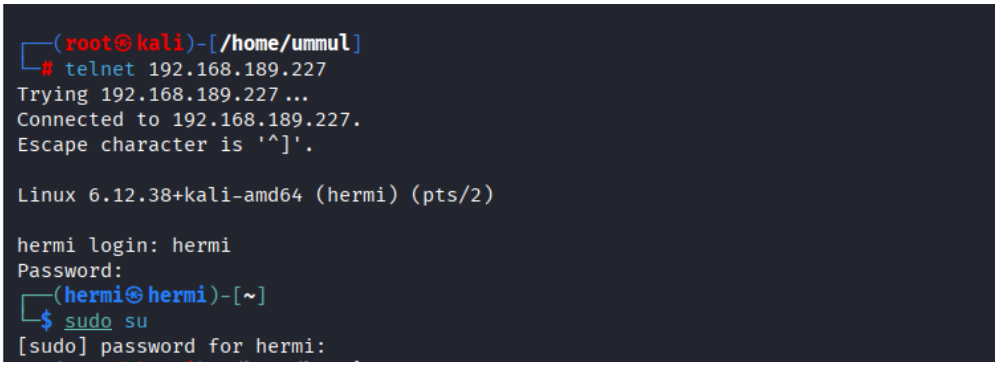
Proses diakhiri dengan menjalankan perintah arp.spoof on untuk mulai meracuni tabel ARP target. Bettercap berhasil memposisikan diri di tengah jalur komunikasi, yang dibuktikan dengan tertangkapnya log permintaan HTTP GET secara *real-time* dari perangkat target 192.168.189.124. Terlihat aktivitas akses ke situs testphp.vulnweb.com berhasil disadap sepenuhnya, menandakan serangan *Man-in-the-Middle* telah berhasil mengekspos trafik data yang tidak terenkripsi.



Penyadapan mencapai tahap krusial ketika Bettercap berhasil menangkap permintaan HTTP POST dari target 192.168.189.124 ke halaman login testphp.vulnweb.com/userinfo.php. Karena protokol HTTP tidak terenkripsi, seluruh data sensitif yang dikirimkan melalui formulir login dapat terbaca dengan jelas oleh penyerang. Pada bagian bawah log terminal, terlihat informasi kredensial berupa nama pengguna (uname=test) dan kata sandi (pass=test) yang berhasil disadap secara langsung, membuktikan efektivitas serangan *Man-in-the-Middle* dalam mencuri informasi otentikasi pada jaringan yang tidak aman.

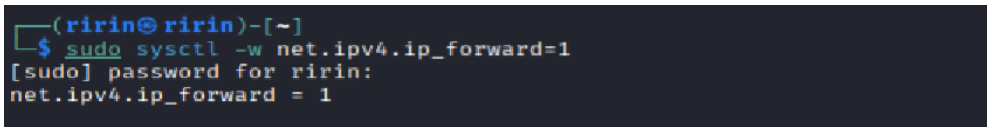
e. Sesion Hijacking

- Telnet client-server

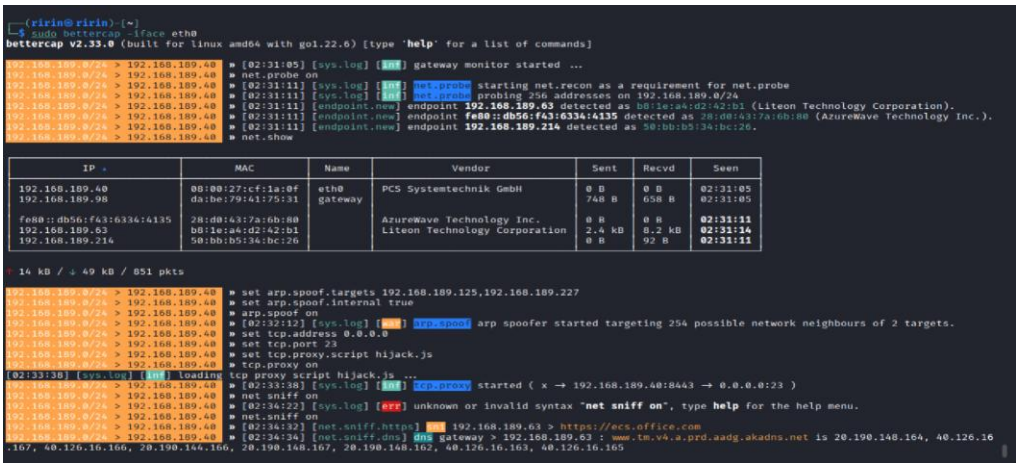


Proses pengujian dilanjutkan dengan melakukan simulasi akses jarak jauh menggunakan protokol **Telnet** ke alamat IP 192.168.189.227. Karena Telnet mengirimkan data dalam bentuk teks biasa (*plain text*) tanpa enkripsi, seluruh aktivitas login yang dilakukan oleh pengguna—termasuk nama pengguna (**hermi**) dan kata sandi—dapat dipantau sepenuhnya oleh penyerang yang telah memposisikan diri di tengah jaringan. Pada terminal, terlihat sesi berhasil terbentuk hingga pengguna mendapatkan akses *shell* dan mencoba meningkatkan hak akses menggunakan perintah sudo su.

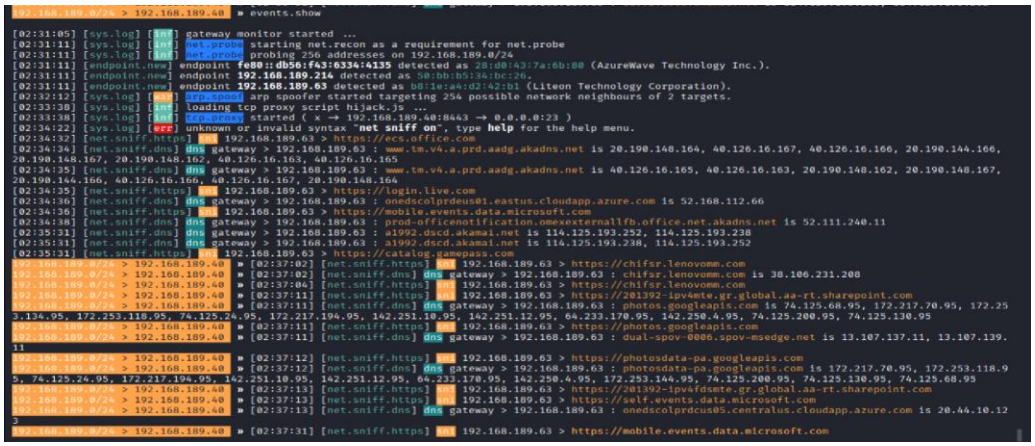
- Proses Hijacking



Untuk menjalankan serangan *Session Hijacking* secara efektif, komputer penyerang harus dikonfigurasi agar dapat meneruskan paket data yang disadap kembali ke tujuannya. Hal ini dilakukan dengan mengaktifkan fitur IP Forwarding melalui perintah sudo sysctl -w net.ipv4.ip_forward=1. Dengan mengubah nilai parameter tersebut menjadi 1, mesin penyerang resmi bertindak sebagai jembatan (*gateway*) yang transparan.

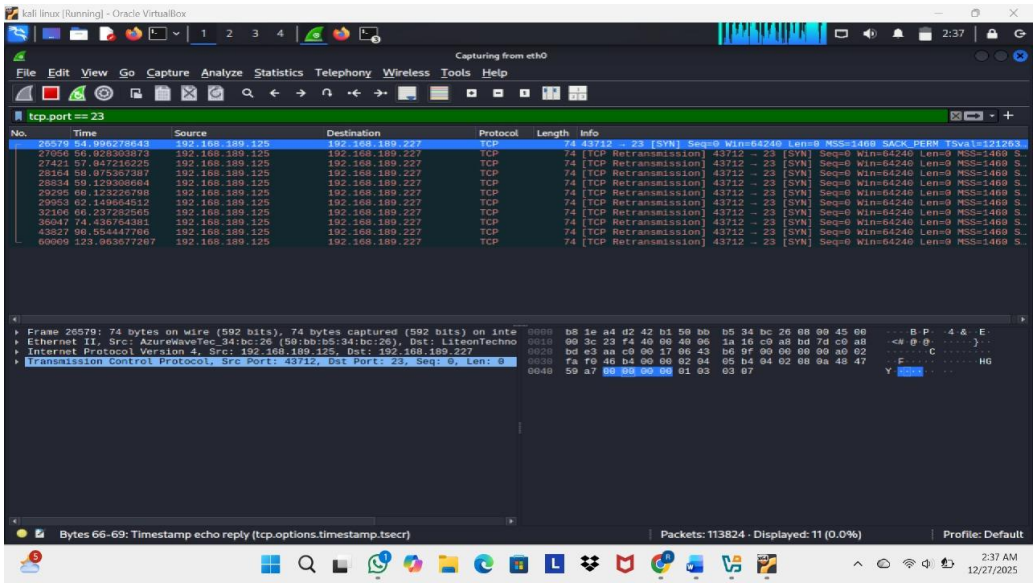


Pada tahap ini, penyerang memuat skrip hijack.js melalui modul tcp.proxy yang dikonfigurasi pada port 23 (Telnet). Perintah set arp.spoof.targets 192.168.189.227 digunakan untuk mengunci target, sementara arp.spoof on memulai proses peracunan ARP. Dengan status tcp.proxy yang telah aktif (*started*), penyerang kini siap melakukan intersepsi dan modifikasi data secara otomatis pada sesi Telnet yang sedang berjalan antara client dan server.



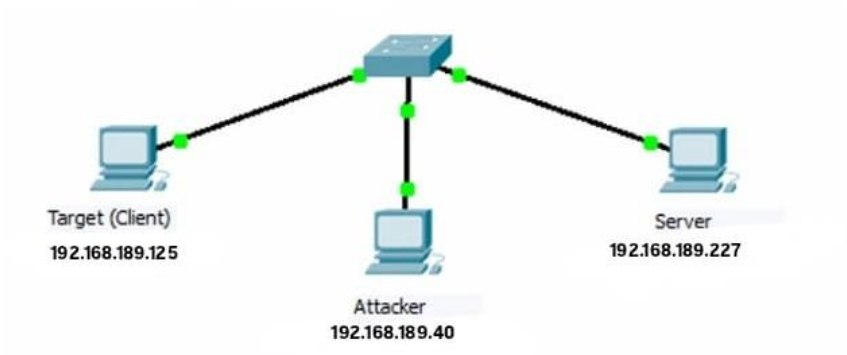
Setelah seluruh modul serangan diaktifkan, terminal Bettercap menampilkan log aktivitas jaringan secara *real-time*. Terlihat bahwa fitur **net.sniff** mulai menangkap berbagai paket data dari target, termasuk kueri DNS dan koneksi HTTPS ke berbagai domain layanan seperti Microsoft Office, Google Photos, dan Lenovo. Meskipun trafik HTTPS terenkripsi, kemampuan penyerang untuk memetakan destinasi trafik target menunjukkan bahwa posisi *Man-in-the-Middle* telah berhasil dipertahankan, memungkinkan penyerang untuk menganalisis pola aktivitas pengguna di dalam jaringan tersebut.

Setelah seluruh parameter serangan dan *forwarding* dikonfigurasi, langkah berikutnya adalah membuka alat analisis protokol jaringan dengan menjalankan perintah sudo wireshark &. Penggunaan simbol ampersand (&) memungkinkan Wireshark berjalan di latar belakang (*background process*), sehingga terminal tetap dapat digunakan untuk memantau log Bettercap secara bersamaan. Melalui Wireshark, penyerang dapat melakukan inspeksi mendalam terhadap paket data Telnet yang telah berhasil dialihkan, guna mengidentifikasi isi komunikasi dan memvalidasi keberhasilan pembajakan sesi secara visual.



Setelah Wireshark terbuka, penyerang menerapkan filter tcp.port == 23 untuk memfokuskan analisis pada lalu lintas Telnet. Pada jendela utama, terlihat deretan paket TCP yang dikirim dari target (192.168.189.125) menuju server (192.168.189.227). Munculnya banyak paket [SYN] dan [TCP Retransmission] mengindikasikan bahwa sesi sedang berusaha dibangun atau telah diinterupsi oleh proses hijacking. Dengan membedah isi paket tersebut, penyerang dapat merekonstruksi seluruh percakapan teks, memvalidasi perintah yang disisipkan melalui skrip hijack.js, dan memastikan kontrol penuh atas sesi jarak jauh tersebut telah berhasil didapatkan.

2. IP Spoofing
   a. Gambar topologi jaringan beserta IP Addressnya

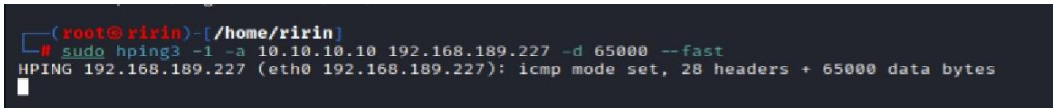


Gambar topologi menunjukkan skema jaringan yang digunakan untuk serangan IP Spoofing. Penyerang (*Attacker*) dengan IP 192.168.189.40 memposisikan diri di antara Target/Client (192.168.189.125) dan Server (192.168.189.227). Dalam

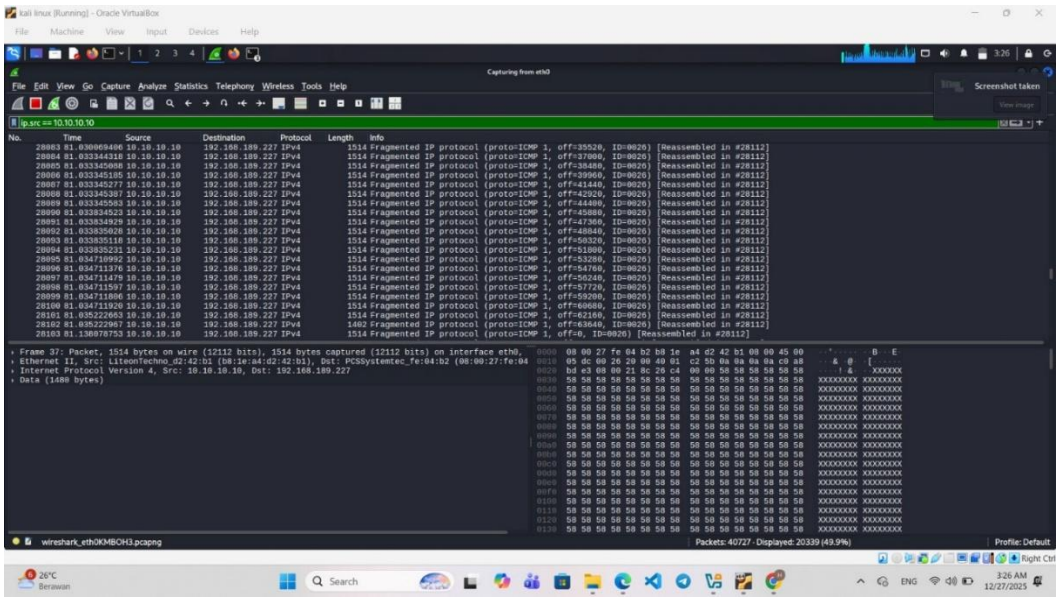
skenario ini, penyerang memanipulasi paket data sehingga seolah-olah paket tersebut berasal dari IP yang sah (misalnya, menyamar sebagai Client saat mengirim ke Server, atau sebaliknya).

b. Menjalankan beberapa tool ip spoofing

- Pod_spoofing



Pada terminal laptop penyerang, perintah sudo hping3 -1 -a 10.10.10.10 192.168.189.227 -d 65000 --fast dijalankan untuk melakukan serangan Ping of Death yang dikombinasikan dengan IP Spoofing. Penyerang menggunakan opsi -a untuk memalsukan identitasnya menjadi IP 10.10.10.10 saat mengirimkan paket ICMP (ping) ke server target 192.168.189.227. Dengan ukuran paket abnormal sebesar 65.000 byte yang dikirim secara cepat (--fast), serangan ini bertujuan untuk membanjiri (*flooding*) sumber daya server sekaligus menyembunyikan lokasi asli penyerang di dalam jaringan.



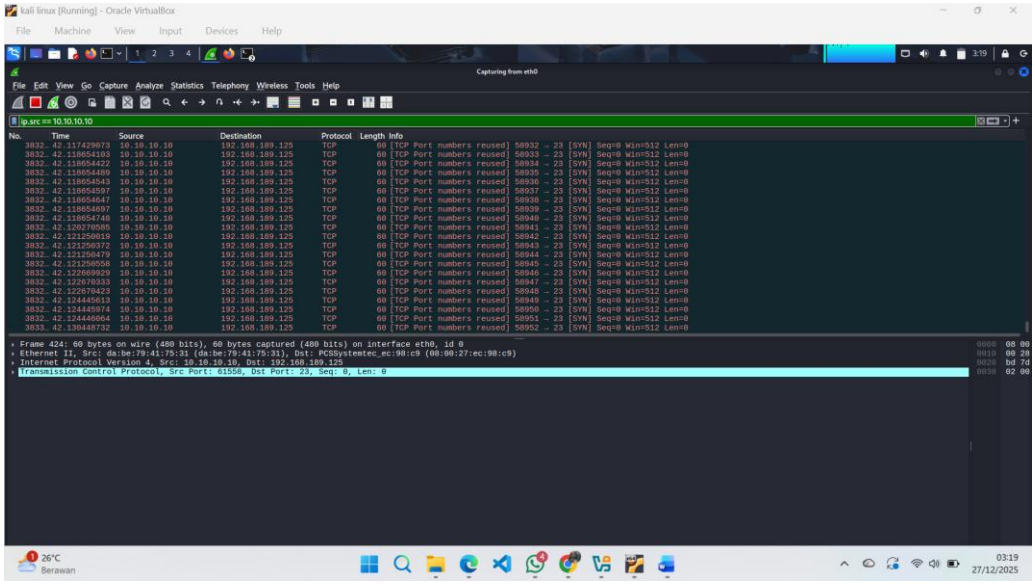
Pada sisi target, Wireshark digunakan untuk memverifikasi serangan dengan menerapkan filter ip.src == 10.10.10.10. Terlihat banjir paket data yang masuk secara masif dari IP palsu tersebut menuju server 192.168.189.227. Paket-paket tersebut terdeteksi sebagai Fragmented IP protocol, di mana satu paket besar dipecah menjadi bagian-bagian kecil berkuran 1514 byte agar dapat melewati jaringan. Karena ukuran total paket yang dikirimkan tidak wajar (sangat besar), sistem target dipaksa bekerja keras untuk menyusun kembali (*reassembled*)

fragmen tersebut, yang pada akhirnya dapat mengakibatkan kelumpuhan sistem atau *buffer overflow*.

- Syn_flood

```
┌──(root㉿ririn)-[/home/ririn]
└─# sudo hping3 -S 192.168.189.125 -a 10.10.10.10 -p 23 --flood
HPING 192.168.189.125 (eth0 192.168.189.125): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Pada terminal laptop penyerang, dijalankan perintah sudo hping3 -S 192.168.189.125 -a 10.10.10.10 -p 23 --flood. Penyerang mengirimkan paket dengan *flag* **SYN** (permintaan koneksi) secara masif menuju port 23 (Telnet) milik target. Dengan menggunakan opsi --flood, paket dikirim secepat mungkin tanpa menunggu balasan, sementara opsi -a 10.10.10.10 digunakan untuk memalsukan alamat IP asal agar identitas penyerang tidak terlacak.

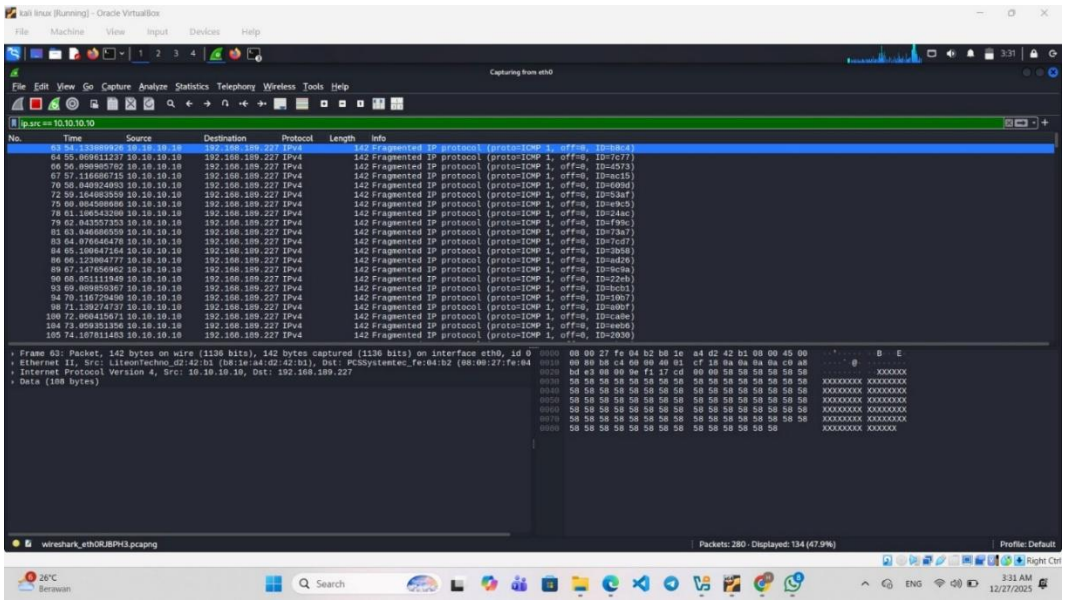


Pada sisi target, **Wireshark** menangkap banjir paket **TCP [SYN]** yang berasal dari IP palsu 10.10.10.10. Terlihat kolom *Info* dipenuhi dengan permintaan koneksi berulang dalam waktu yang sangat singkat. Karena penyerang tidak pernah menyelesaikan proses *three-way handshake*, server target akan terus mengalokasikan sumber daya untuk koneksi "setengah terbuka" ini. Akibatnya, antrean koneksi server menjadi penuh, yang menyebabkan layanan Telnet tidak dapat diakses oleh pengguna sah (**Denial of Service**).

- Teardrop_spoofing

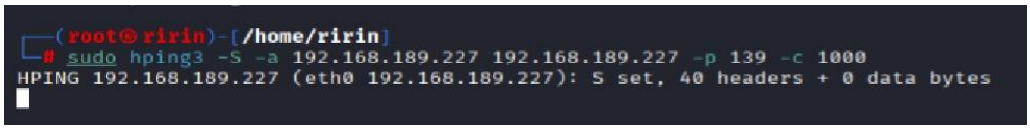
```
┌──(root㉿ririn)-[/home/ririn]
└─# sudo hping3 -1 -a 10.10.10.10 192.168.189.227 -x -y -d 100
HPING 192.168.189.227 (eth0 192.168.189.227): icmp mode set, 28 headers + 100 data bytes
```

Pada terminal penyerang, dijalankan perintah sudo hping3 -1 -a 10.10.10.10 192.168.189.227 -x -y -d 100. Serangan ini menggunakan teknik IP Fragment Overlap. Penyerang mengirimkan paket ICMP dengan opsi -x dan -y untuk memanipulasi *offset* fragmen paket. Dengan memalsukan IP asal menjadi 10.10.10.10, penyerang mengirimkan fragmen-fragmen paket yang tumpang tindih (*overlapping*) ke server target.

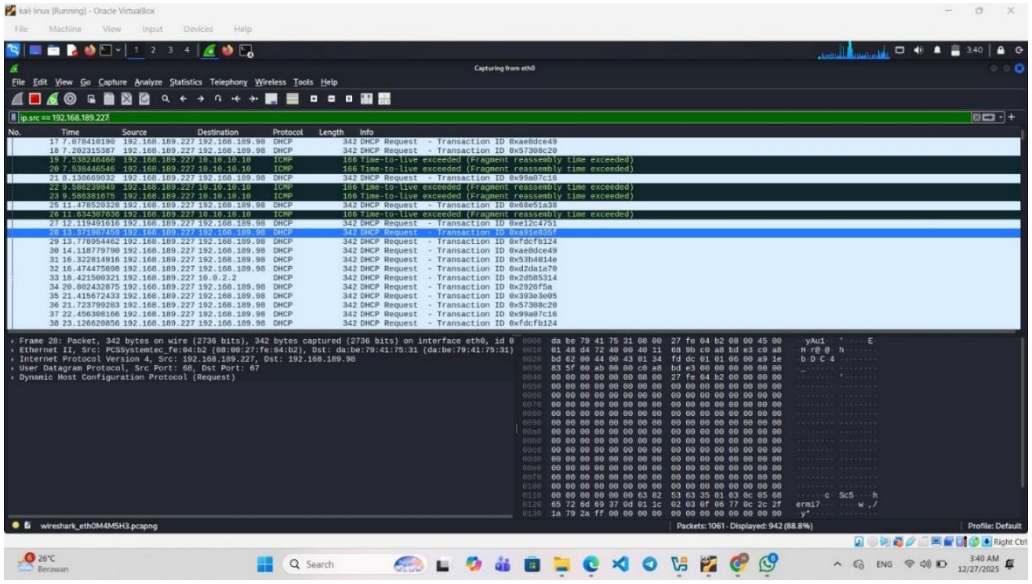


Pada Wireshark di sisi target, terlihat paket-paket yang ditandai sebagai Fragmented IP protocol. Karena penyerang telah memanipulasi nilai *offset* sehingga fragmen yang satu menumpuk di atas fragmen yang lain, sistem operasi target akan mengalami kesulitan saat mencoba menyusun kembali (*reassemble*) paket-paket tersebut. Kegagalan sistem dalam menangani fragmen yang tumpang tindih ini dapat menyebabkan *buffer overflow* atau kegagalan sistem (*crash*), yang merupakan ciri khas dari serangan Teardrop.

- Land_attack



Pada terminal penyerang, dijalankan perintah sudo hping3 -S -a 192.168.189.227 192.168.189.227 -p 139 -c 1000. Penyerang mengirimkan 1000 paket TCP SYN dengan alamat IP asal (-a) yang dipalsukan agar identik dengan alamat IP tujuan, yaitu 192.168.189.227. Dengan menargetkan port 139 (NetBIOS), penyerang mencoba mengeksploitasi cara sistem menangani koneksi yang berasal dari dirinya sendiri.



Meskipun gambar Wireshark yang Anda lampirkan sebelumnya menunjukkan paket terfragmentasi, mekanisme Land Attack yang sebenarnya terlihat dari sinkronisasi IP asal dan tujuan yang sama. Pada sisi target, paket ini memaksa sistem untuk membalas permintaan koneksi ke alamat IP-nya sendiri. Hal ini menciptakan perulangan (*infinite loop*) di dalam *network stack* target yang dapat menyebabkan sistem menjadi sangat lambat, menguras sumber daya CPU, atau bahkan mengalami *crash* total karena terjebak memproses balasan untuk dirinya sendiri.

c. Hasil pengamatan

Dalam pengamatan serangan Ping of Death (PoD), laptop penyerang mengirimkan paket ICMP berukuran abnormal sebesar 65.000 byte dengan identitas palsu, yang menyebabkan target menerima banjir paket terfragmentasi seukuran 1514 byte. Hal ini memaksa sistem target bekerja sangat keras untuk menyusun kembali fragmen tersebut, yang berisiko memicu kelumpuhan sistem atau *buffer overflow*. Selanjutnya, pada serangan SYN Flood, penyerang membanjiri port Telnet target dengan permintaan koneksi (flag SYN) tanpa pernah menyelesaikannya, sehingga antrean koneksi server menjadi penuh dan mengakibatkan layanan tidak dapat diakses oleh pengguna sah.

Pada tahap serangan Teardrop, pengamatan menunjukkan penggunaan teknik *IP Fragment Overlap* di mana penyerang memanipulasi nilai *offset* sehingga fragmen paket saling tumpang tindih. Kegagalan sistem target dalam menangani fragmen yang tidak sinkron ini dapat berujung pada kegagalan sistem atau *crash*.

Terakhir, dalam Land Attack, penyerang memalsukan alamat IP asal agar identik dengan alamat IP tujuan, yang menjebak *network stack* target ke dalam perulangan tanpa akhir (*infinite loop*). Kondisi ini mengakibatkan sistem menjadi sangat lambat atau berhenti berfungsi total karena terus-menerus memproses balasan untuk dirinya sendiri.

3. Kesimpulan
   a. Kesimpulan Hasil Praktikum

      Praktikum ini memberikan pemahaman mendalam bahwa penggunaan protokol komunikasi yang tidak terenkripsi, seperti HTTP dan Telnet, sangat rentan terhadap eksploitasi keamanan di jaringan lokal. Melalui implementasi *Man-in-the-Middle* (MITM) dengan teknik *ARP Spoofing*, terbukti bahwa penyerang dapat mencegat lalu lintas data secara *real-time* dan mencuri kredensial sensitif dalam bentuk teks biasa (*plain text*). Selain itu, berbagai pengujian serangan *Denial of Service* (DoS) menunjukkan bahwa infrastruktur jaringan dapat dengan mudah dilumpuhkan jika tidak memiliki sistem validasi paket yang kuat, sehingga dapat mengakibatkan penghentian layanan secara total bagi pengguna sah.

   b. Perbedaan Metode Percobaan IP Spoofing

      Setiap metode *IP Spoofing* yang diuji memiliki mekanisme serangan yang berbeda dalam mengeksploitasi kelemahan sistem target. Serangan *Ping of Death* (PoD) berfokus pada pengiriman paket ICMP dengan ukuran abnormal (65.000 byte) untuk melampaui kapasitas memori target saat menyusun kembali fragmen paket. Berbeda dengan PoD, *SYN Flood* mengeksploitasi proses *TCP three-way handshake* dengan membanjiri port target menggunakan permintaan koneksi palsu agar antrean layanan menjadi jenuh. Sementara itu, *Teardrop* memanipulasi nilai *offset* agar fragmen paket saling tumpang tindih (*overlap*) yang memicu kegagalan sistem, dan *Land Attack* bekerja dengan menyamakan alamat IP asal dan tujuan untuk menjebak sistem target dalam perulangan balasan tanpa akhir (*infinite loop*).

   c. Tipe Transport Layer pada IP Spoofing

      Tipe *transport layer* utama yang digunakan dalam percobaan ini adalah TCP (Transmission Control Protocol), di samping penggunaan protokol ICMP pada lapisan *network*. Protokol TCP digunakan karena memiliki karakteristik *connection-oriented* yang memungkinkan penyerang mengeksploitasi mekanisme jabat tangan koneksi (seperti pada *SYN Flood*) atau memanipulasi status koneksi melalui port layanan tertentu seperti Telnet (port 23) atau NetBIOS (port 139).

Penggunaan tipe protokol ini sangat efektif dalam *IP Spoofing* karena sistem target secara otomatis akan mengalokasikan sumber daya atau mengirimkan balasan balik berdasarkan aturan protokol TCP/IP yang berlaku, sehingga penyerang dapat dengan mudah memanipulasi perilaku server tanpa perlu identitas asli.

d. Metode Menangkal ARP Spoofing dan IP Spoofing

Untuk menangkal serangan *ARP Spoofing*, administrator jaringan dapat menerapkan Static ARP Entries pada perangkat-perangkat penting atau mengaktifkan fitur Dynamic ARP Inspection (DAI) pada perangkat *switch* untuk memvalidasi setiap paket ARP yang melintas. Sedangkan untuk memitigasi *IP Spoofing*, metode yang efektif meliputi penerapan Ingress dan Egress Filtering pada *firewall* untuk menolak paket dengan alamat asal yang tidak sah, serta penggunaan Unicast Reverse Path Forwarding (uRPF) untuk memverifikasi jalur paket. Langkah pencegahan paling fundamental secara keseluruhan adalah bermigrasi dari protokol tidak aman ke protokol yang terenkripsi, seperti menggunakan SSH sebagai pengganti Telnet dan HTTPS sebagai pengganti HTTP, guna memastikan data tetap terlindungi meskipun terjadi penyadapan di jaringan.