



Assignment – A01c

Introduction to Networking Tools

Penulis : BAS

Versi : 1 (20240826-0800)



Riwayat Versi

Setiap “**Versi**” yang dimaksudkan pada riwayat ini dan dijadikan rujukan utama bagi dokumen ini memuat perubahan yang bersifat substantif sehingga perlu diketahui oleh pemangku kepentingan dokumen ini. Dokumen dapat memiliki perubahan non-substantif yang tidak tercatat pada riwayat ini namun tetap tercatat pada riwayat versi yang dikelola Office pada salinan asli dokumen ini.

Riwayat versi ini diurutkan secara kronologis terurut dari versi paling akhir pada baris pertama hingga versi paling awal pada baris terakhir.

Versi	Tanggal dan Waktu	Halaman	Perubahan
1	20240826-0800	Semua	Rilis pertama
2	20240828-2039	18, 19, 22	<ul style="list-style-type: none"> Memperjelas mengenai perintah-perintah yang digunakan untuk melihat tabel ARP pada Windows, macOS, dan Linux Memperjelas kalimat mengenai penjelasan paket “<i>received by filter</i>” (perubahan ditandai dengan highlight kuning)
3	20240831-2033	27, 28	<ul style="list-style-type: none"> Memperjelas mengenai VM Instance mana yang harus digunakan untuk masing-masing soal dari <i>section Network Diagnostics</i> Merevisi soal a dan b untuk <i>sub-section “Traceroute dan ip”</i> pada <i>section Network Diagnostics</i> (perubahan ditandai dengan highlight hijau)

Daftar Isi

 Riwayat Versi	2
 Daftar Isi	3
 Informasi Umum	4
 Ekspektasi Hasil Pembelajaran	4
 Prasyarat	4
 Deskripsi	5
Wireshark	5
Menjalankan Wireshark	5
Mengenal Bagian-Bagian Wireshark	6
Melakukan Packet Capturing dengan Wireshark	7
Mengenal Packet Filter pada Wireshark	8
Melakukan Decrypting HTTP Melalui TLS Packets	9
Menginstal Aplikasi pada VM	16
Informasi dan Konfigurasi Umum pada Jaringan	17
IPConfig	17
ARP	18
Pencarian Nama Domain	20
NSLookup dan DiG	20
Network Browser dan Packet Capture	20
Lynx	20
TCPDump	21
IP Socket dan Informasi Port	23
Netstat/SS	23
Ping	24
Tracert	25
 Spesifikasi	26
 Informasi Pengumpulan Berkas	29
 Peraturan	30

Assignment – A01c

Introduction to Networking Tools

🔍 Informasi Umum

Tipe Tugas	:	Individu
Batas Waktu Pengumpulan	:	Jumat, 06 September 2024 pukul 17.00 WIB (SCeLE)
Format Penamaan Berkas	:	
- Laporan	:	A01_[NPM].pdf (*digabungkan dengan laporan A01a, A01b, A01d)
- Berkas Lainnya	:	
	o	A01_[NPM]_http2.pcapng (Wireshark capture)
	o	A01_[NPM]_aren.pcapng (tcpdump capture)
Tautan Kerangka Laporan	:	Klik Di Sini

☒ Ekspektasi Hasil Pembelajaran

Setelah mengerjakan penugasan ini, mahasiswa diharapkan dapat:

1. Memiliki pemahaman yang baik (C3) terhadap *tools* dan platform *networking*.
2. Mampu menggunakan (C3) program Wireshark.

❖ Prasyarat

1. Mengunduh dan menginstal versi terbaru dari [Wireshark Stable Release](#) sesuai dengan perangkat masing-masing.
 - **Bagi pengguna macOS**, Anda juga perlu menginstal package “Install ChmodBPF.pkg” dan “Add Wireshark to the system path.pkg” agar dapat meng-*capture* packet. (Selengkapnya klik link berikut: [Installing Wireshark under macOS](#) dan [Quick Setup](#)).
 - **Bagi pengguna Linux**, jalankan perintah “`sudo usermod -a -G wireshark`” sebelum menggunakan Wireshark.
 - **Bagi pengguna Windows**, terdapat opsi untuk menginstal Wireshark menggunakan winget dengan membuka terminal (disarankan *powershell*) lalu jalankan perintah “`winget install -e --id WiresharkFoundation.Wireshark`”.

```
PS C:\Users\mocha> winget install -e --id WiresharkFoundation.Wireshark
Found Wireshark [WiresharkFoundation.Wireshark] Version 4.2.6
This application is licensed to you by its owner.
Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Downloading https://2.na.dl.wireshark.org/win64/Wireshark-4.2.6-x64.msi
[██████████] 60.0 MB / 60.0 MB
Successfully verified installer hash
Starting package install...
Successfully installed
PS C:\Users\mocha>
```

Jika mengalami masalah saat menginstal atau menjalankan Wireshark, silakan hubungi Asisten Dosen melalui Server Discord.

2. Menyelesaikan tugas A01a serta menyiapkan dua buah VM di GCP yang dapat diakses dan digunakan sesuai spesifikasi pada tugas A01a.

📘 Deskripsi

Pada tugas ini, Anda akan menggunakan beberapa *networking tools* yang bisa dijalankan pada perangkat yang terhubung melalui jaringan seperti perangkat lokal ataupun *instance VM* yang ada di GCP. Alat tersebut dapat memberikan informasi atau mengatur berbagai aspek terkait jaringan. Pada tugas ini, Anda diharapkan familiar dengan alat tersebut dan tahu cara menggunakannya dengan tepat.

Berikut beberapa penjelasan singkat mengenai beberapa alat yang akan digunakan pada tugas kali ini dan beberapa tugas yang akan datang. Anda dapat mengakses kembali dokumen ini apabila diperlukan.

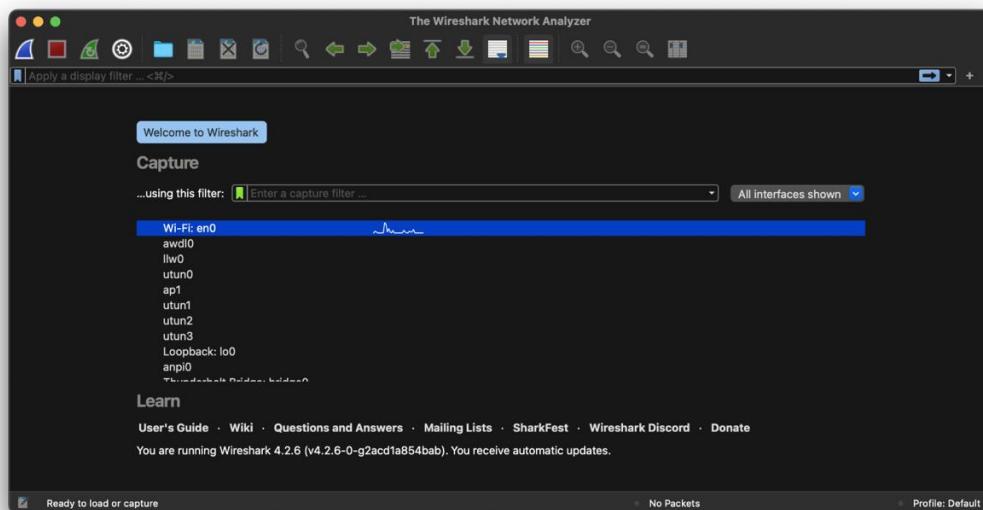
Wireshark

Wireshark merupakan sebuah *network tool* yang digunakan untuk menganalisis *network protocol*. Wireshark memungkinkan kita untuk melihat apa yang terjadi di jaringan pada *microscopic level*. Tools ini merupakan standar *de facto* (dan sering kali *de jure*) di banyak perusahaan komersial dan nirlaba (*non-profit*), lembaga pemerintah, serta lembaga pendidikan. Pada tugas ini, Anda akan mempelajari cara melakukan *packet capturing* sederhana dengan menggunakan Wireshark serta kemudian melakukan analisis mengenai data yang telah di-capture tersebut.



Menjalankan Wireshark

Setelah aplikasi Wireshark telah terunduh dan terinstal pada perangkat Anda, buka aplikasi tersebut. Pastikan aplikasi dapat berjalan dengan lancar. Tampilan awal aplikasi kurang lebih akan terlihat seperti berikut.

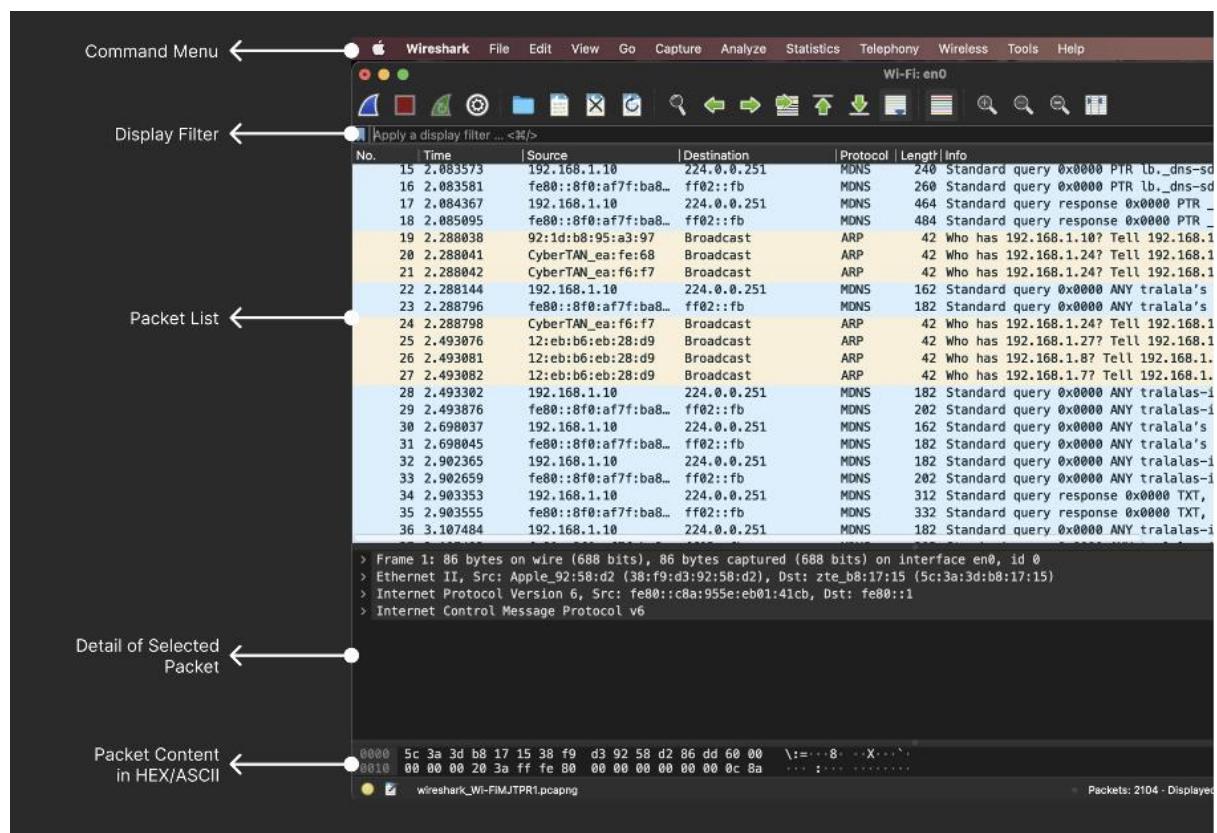


Apabila tampilan awal aplikasi tidak sama persis seperti gambar di atas, Anda tidak perlu khawatir karena masing-masing perangkat mungkin mempunyai *network interfaces* yang berbeda. Hal yang perlu diperhatikan adalah terdapat *section Capture* (di bawah tulisan “Welcome to Wireshark” pada gambar di atas). *Section* tersebut akan menampilkan *interface* apa saja yang bisa Anda *capture*.

Pada tugas ini, kita akan melakukan *capturing* pada *interface* Wi-Fi (dengan asumsi bahwa Anda saat ini mengakses internet menggunakan Wi-Fi). Namun, apabila saat ini Anda menggunakan *interface* lain seperti LAN untuk mengakses internet, maka pilihlah *interface* tersebut. Salah satu cara untuk mengetahui *interface* mana yang harus dipilih adalah dengan melihat keberadaan diagram garis yang ditampilkan di sisi kanan *interface*, misalnya pada gambar di atas terlihat bahwa terdapat diagram garis pada Wi-Fi: en0.

Mengenal Bagian-Bagian Wireshark

Wireshark mempunyai beberapa bagian dengan fungsi masing-masing. Berikut ini merupakan penjelasan singkat mengenai bagian-bagian pada Wireshark yang mungkin akan Anda gunakan.

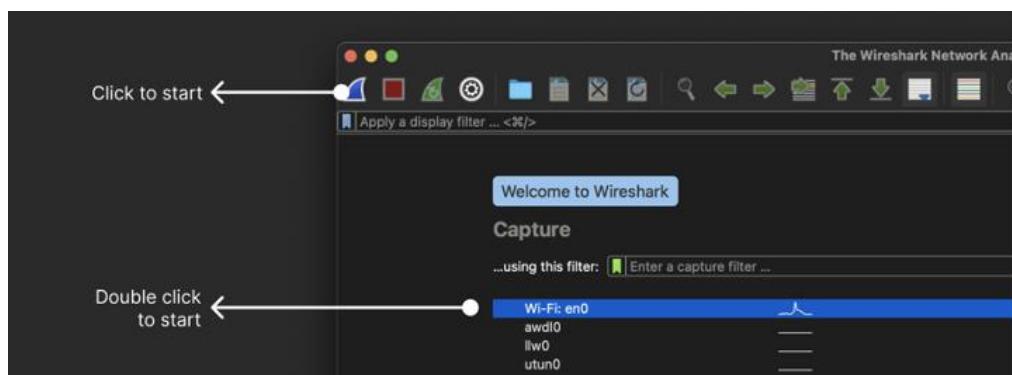


- Command Menu:** Bagian ini terdiri dari semua *command* yang dapat digunakan dalam Wireshark. Yang perlu diperhatikan adalah tab File dan juga Toolbar di bawahnya. Tab File digunakan untuk membuka berkas packet capture yang sudah pernah disimpan (Open), menyimpan packet capture yang baru dilakukan (Save As), dan lainnya. Sementara itu, Toolbar di bawah berisi pintasan perintah seperti Start/Capture Packet (ikon Sirip Hiu Biru), Stop Capturing (ikon Kotak Merah), dan Find Packet (ikon Kaca Pembesar).

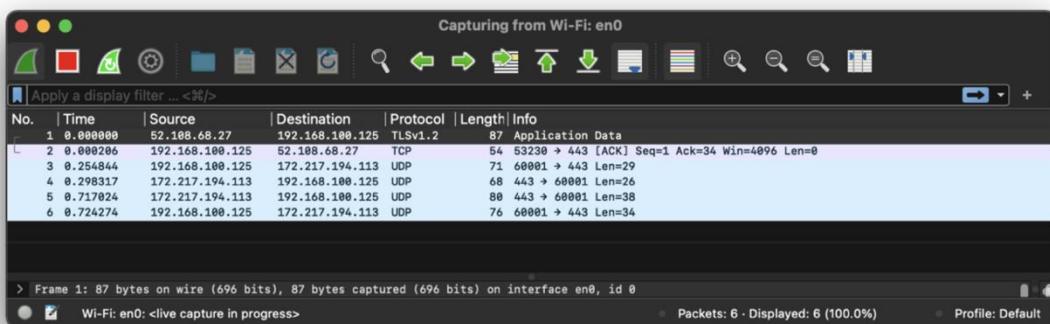
- **Display Filter:** Display Filter digunakan untuk memfilter packet sehingga hanya packet yang sesuai filter yang akan ditampilkan. Ikon pita biru di area paling kiri adalah pintasan Filter, digunakan untuk mencari filter yang umum digunakan/dibuat secara *custom*. Detail mengenai penggunaan filter ini akan dibahas pada bagian [Mengenal Packet Filter pada Wireshark](#)
- **Packet List:** Bagian ini menampilkan semua *network packet* yang telah ter-capture serta menampilkan *overview* setiap *packet*, termasuk *source IP addresses*, *destination IP addresses*, dll. Jika Display Filter digunakan, Packet List hanya akan menampilkan *packet* yang lolos filter.
- **Packet Detail:** Bagian ini menunjukkan isi *network packet* dalam *human-readable format*. Beberapa informasi yang dapat dilihat di sini adalah *frame Ethernet* dan rincian pada setiap *layer*.
- **Packet Content:** Bagian ini menampilkan konten dari packet dalam bentuk heksadesimal dan ASCII.

Melakukan Packet Capturing dengan Wireshark

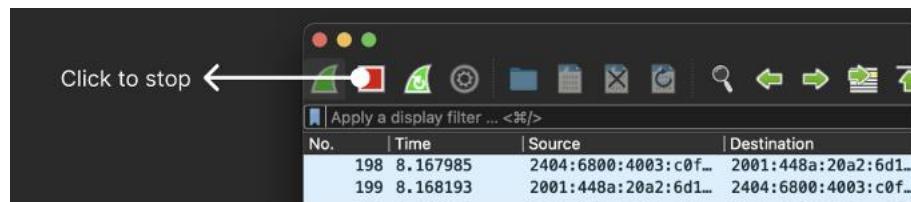
Untuk memulai *capturing*, pilih salah satu interface (interface terpilih akan ter-highlight warna biru) kemudian tekan ikon Start. Sebagai alternatif, Anda juga langsung dapat melakukan *double-click* pada *interface* dan sesi *capturing* akan langsung dimulai.



Sesi *capturing*-pun dimulai. Anda dapat melihat bahwa Wireshark akan meng-*capture* semua *network packets* yang melewati *interface* ini secara *real-time*.



Setelah beberapa detik, hentikan sesi *capturing* dengan menekan ikon Stop di sudut kiri atas (sebelah ikon Start) seperti berikut:



Setelah sesi *capturing* dihentikan, data yang telah *di-capture* dapat langsung disimpan sebagai *network capture file* (.pcapng) dengan menekan File > Save As > network_packets.pcap (atau sesuai instruksi yang diberikan).

Harap diperhatikan dalam tugas ini dan tugas berikutnya, simpanlah hasil capture hanya jika Anda yakin bahwa hasil tangkapan tersebut telah berisi packets yang diperlukan sesuai dengan keperluan analisis dan ketentuan tugas. Anda perlu meng-*capture* data secara mencukupi (beberapa detik saja). Jika meng-*capture* terlalu cepat, packet-packet yang diperlukan mungkin saja belum ter-*capture* secara menyeluruh. Sebaliknya, jika terlalu lama, ukuran file dapat menjadi sangat besar sehingga Anda mungkin saja tidak akan dapat mengunggahnya ke SCeLe. Jika Anda tidak yakin dengan data yang ditangkap, Anda dapat mengulang sesi *capturing* sesuai keinginan.

Mengenal Packet Filter pada Wireshark

Setelah melakukan *capturing*, kita bisa melakukan *filtering* pada tabel *packet* yang telah didapatkan. Proses *filtering* dilakukan dengan memanfaatkan **Display Filter**. Untuk menggunakan Display Filter, Anda cukup mengetik filter yang ingin dilihat.

Tipe-Tipe Filter

1. Protocol

Filter untuk protokol digunakan untuk melakukan *packet filtering* berdasarkan tipe protokolnya. Masing-masing *packet* hanya memiliki satu protokol, tetapi dapat dienkapsulasi oleh *packet* lain dengan protokol yang berbeda. Beberapa filter protokol antara lain: HTTP, HTTPS, TCP, UDP, ARP, dan TLS.

tcp						
No.	Time	Source	Destination	Protocol	Length	Info
5	0.027436	192.168.100.125	152.118.148.78	TCP	78	54155 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=6
6	0.027603	192.168.100.125	152.118.148.78	TCP	78	54156 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=6
7	0.035516	152.118.148.78	192.168.100.125	TCP	74	443 → 54155 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MS
8	0.035516	152.118.148.78	192.168.100.125	TCP	74	443 → 54156 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MS

2. Port Address

Jika Anda ingin melakukan *packet filtering* berdasarkan *port* tertentu, Anda dapat menggunakan *port filter* melalui *display filter*. Terdapat dua macam port, yaitu port TCP dan port UDP. Untuk menggunakannya, Anda perlu memasukkan tipe protokolnya dengan huruf kecil, diikuti dengan titik beserta nomor *port*-nya. [Contoh: **tcp.port == 54155**]

tcp.port == 54155						
No.	Time	Source	Destination	Protocol	Length	Info
5	0.027436	192.168.100.125	152.118.148.78	TCP	78	54155 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=6
7	0.035516	152.118.148.78	192.168.100.125	TCP	74	443 → 54155 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MS
9	0.035579	192.168.100.125	152.118.148.78	TCP	66	54155 → 443 [ACK] Seq=1 Ack=1 Win=131584 Len=0 TSval=
11	0.035727	192.168.100.125	152.118.148.78	TLSv1.2	583	Client Hello

Frame 5: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface en0, id 0
 > Ethernet II, Src: Apple_52:5b:d3 (3c:06:30:52:5b:d3), Dst: HuaweiTe_be:5a:55 (1c:ae:cb:be:5a:55)
 > Internet Protocol Version 4, Src: 192.168.100.125, Dst: 152.118.148.78
 > Transmission Control Protocol, Src Port: 54155, Dst Port: 443, Seq: 0, Len: 0
 Source Port: 54155

0000	1c ae cb b
0010	00 40 00 0
0020	94 4e d3 b
0030	ff ff 48 0
0040	08 0a b1 0

3. IP Address

Anda juga dapat memfilter packet berdasarkan alamat IP-nya. Jenis filter ini digunakan saat Anda ingin menganalisis paket dari *source* tertentu atau ke *destination* tertentu. Untuk melakukan ini, Anda perlu terlebih dahulu mengetahui IP Address dari *source* atau *destination* yang ingin dicek.

Misalkan kita ingin menganalisis packet yang berasal dari atau menuju ke SCeLe. Oleh karena itu, kita perlu terlebih dahulu mencari tahu IP Address dari scele.cs.ui.ac.id ini. Hal ini dapat dilakukan dengan menggunakan **command “ping [domain name]”** pada Terminal/Command Prompt perangkat Anda atau dengan menggunakan bantuan situs [nslookup](#).

```
(base) bastian@Bastians-Mac ~ % ping scele.cs.ui.ac.id
PING scele.cs.ui.ac.id (152.118.148.78): 56 data bytes
```

Didapatkan informasi bahwa IP Address dari scele.cs.ui.ac.id adalah 152.118.148.78. Dengan begitu, kita dapat menuliskan **ip.addr == 152.118.148.78** pada *display filter*.

No.	Time	Source	Destination	Protocol	Length	Info
5	0.027436	192.168.100.125	152.118.148.78	TCP	78	54155 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=6
6	0.027603	192.168.100.125	152.118.148.78	TCP	78	54156 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=6
7	0.035516	152.118.148.78	192.168.100.125	TCP	74	443 → 54155 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MS
8	0.035516	152.118.148.78	192.168.100.125	TCP	74	443 → 54156 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MS
9	0.035579	192.168.100.125	152.118.148.78	TCP	66	54155 → 443 [ACK] Seq=1 Ack=1 Win=131584 Len=0 TSval=
10	0.035612	192.168.100.125	152.118.148.78	TCP	66	54156 → 443 [ACK] Seq=1 Ack=1 Win=131584 Len=0 TSval=

4. Multiple Filter

Selain menerapkan *single filter*, kita juga dapat melakukan *multiple filter* pada Wireshark. Hal ini dapat dilakukan dengan menggunakan simbol **&&** (untuk logika AND), simbol **||** (untuk logika OR), atau dengan kata-kata langsung seperti **and**, **not**, dan lainnya. Contohnya sebagai berikut:

No.	Time	Source	Destination	Protocol	Length	Info
11	0.035727	192.168.100.125	152.118.148.78	TLSv1.2	583	Client Hello
12	0.035825	192.168.100.125	152.118.148.78	TLSv1.2	583	Client Hello
19	0.045845	152.118.148.78	192.168.100.125	TLSv1.2	1466	Server Hello
30	0.068418	152.118.148.78	192.168.100.125	TLSv1.2	1466	Ignored Unknown Record

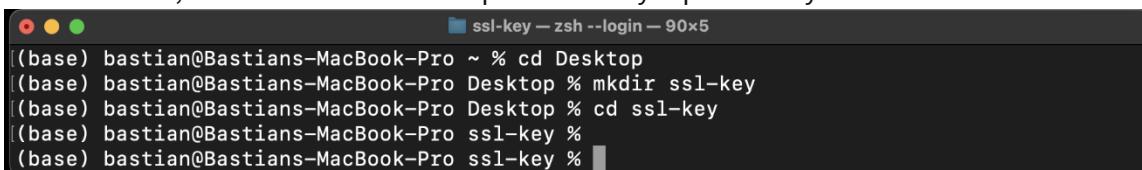
Melakukan Decrypting HTTP Melalui TLS Packets

Saat ini, sebagian besar HTTP packets dienkripsi melalui TLS demi alasan keamanan. Oleh karena itu, saat kita mencoba mengakses situs web dengan protokol HTTP dan menangkap packet menggunakan Wireshark, packet HTTP tidak akan muncul dalam packet list. Agar packet muncul, kita perlu melakukan decrypting.

Pada Wireshark, decrypting packet HTTP yang dienkripsi dengan TLS memerlukan *pre-master secret keys*. Untuk melakukannya, kita perlu menambahkan *environment variable* bernama **SSLKEYLOGFILE** ke perangkat kita. Setelah itu, log file akan dibuat, yang terdiri dari *pre-master keys* saat berkomunikasi dengan HTTP melalui TLS. Kita dapat menemukan file log ini di path yang kita tetapkan sebagai value dari **SSLKEYLOGFILE**.

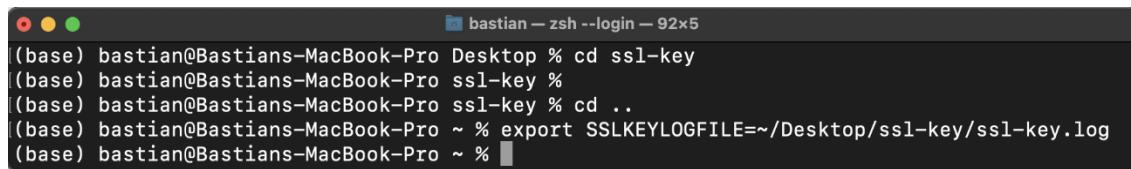
Menambahkan Environment Variable – untuk Pengguna macOS dan Linux

1. Buka Terminal, buatlah folder di Desktop untuk menyimpan ssl-key terlebih dahulu.



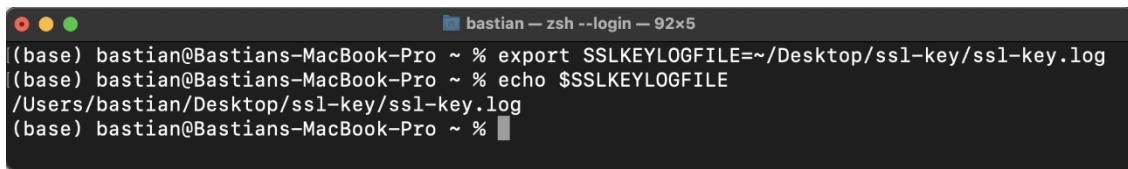
```
ssl-key - zsh --login - 90x5
(base) bastian@Bastians-MacBook-Pro ~ % cd Desktop
(base) bastian@Bastians-MacBook-Pro Desktop % mkdir ssl-key
(base) bastian@Bastians-MacBook-Pro Desktop % cd ssl-key
(base) bastian@Bastians-MacBook-Pro ssl-key %
(base) bastian@Bastians-MacBook-Pro ssl-key %
```

2. Ketik command berikut: **export SSLKEYLOGFILE=<path to ssl-key.log file>**



```
bastian - zsh --login - 92x5
(base) bastian@Bastians-MacBook-Pro Desktop % cd ssl-key
(base) bastian@Bastians-MacBook-Pro ssl-key %
(base) bastian@Bastians-MacBook-Pro ssl-key % cd ..
(base) bastian@Bastians-MacBook-Pro ~ % export SSLKEYLOGFILE=~/Desktop/ssl-key/ssl-key.log
(base) bastian@Bastians-MacBook-Pro ~ %
```

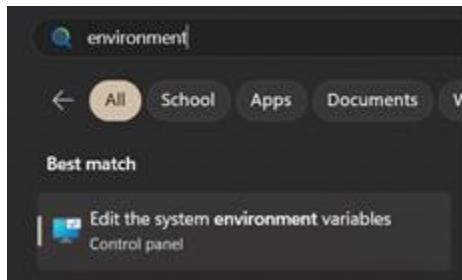
3. Selanjutnya, ketik **echo \$SSLKEYLOGFILE**. Path yang telah Anda tentukan untuk variabel SSLKEYLOGFILE akan tercetak.



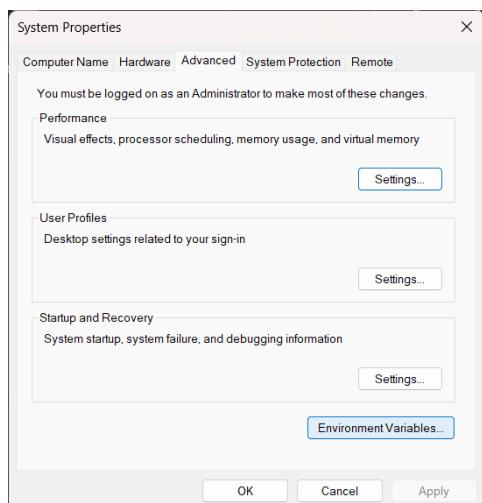
```
bastian - zsh --login - 92x5
(base) bastian@Bastians-MacBook-Pro ~ % export SSLKEYLOGFILE=~/Desktop/ssl-key/ssl-key.log
(base) bastian@Bastians-MacBook-Pro ~ % echo $SSLKEYLOGFILE
/Users/bastian/Desktop/ssl-key/ssl-key.log
(base) bastian@Bastians-MacBook-Pro ~ %
```

Menambahkan Environment Variable – untuk Pengguna Windows

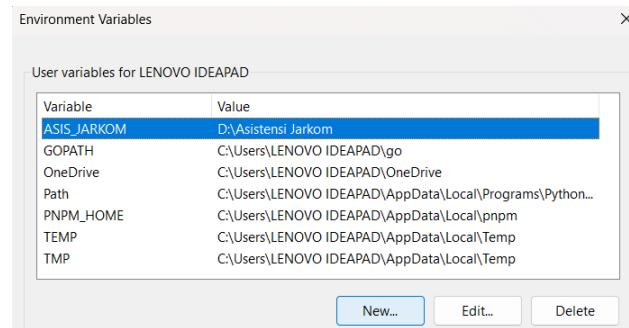
1. Pada search bar, cari “Edit the system environment variables” and klik hal tersebut.



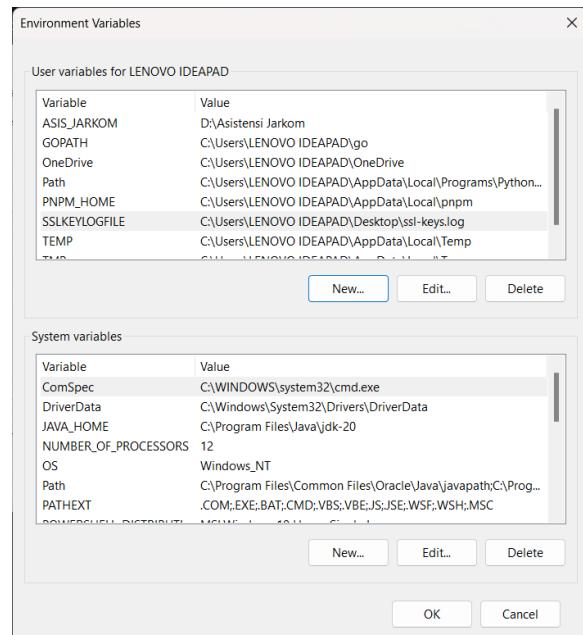
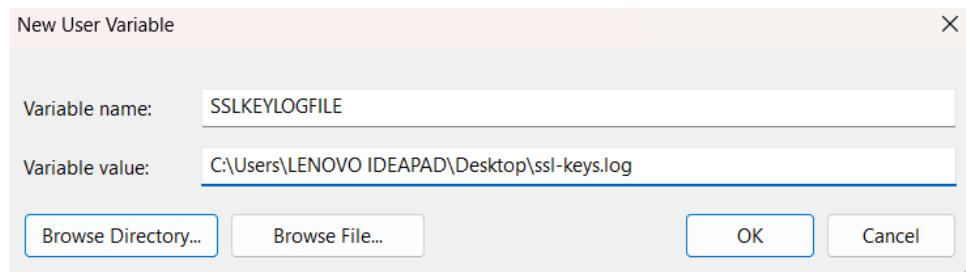
2. Klik tab Advanced, lalu klik Environment Variables....

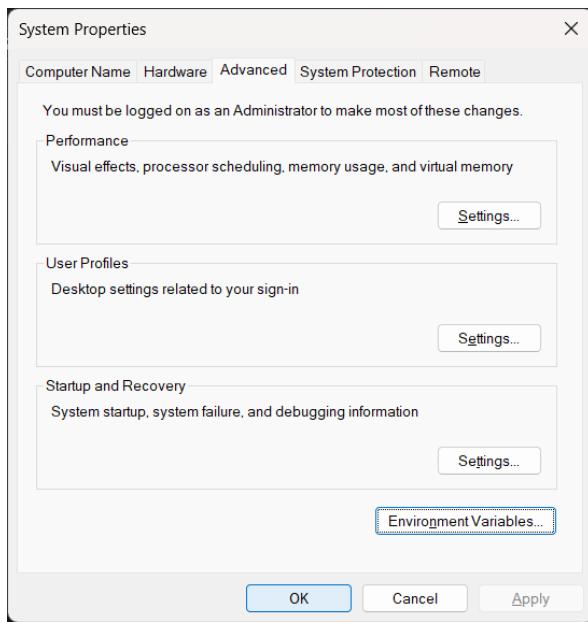


3. Klik tombol New... untuk membuat user variable baru.



4. Isikan kolom variable name dengan SSLKEYLOGFILE dan variable value dengan path di mana Anda ingin menyimpan file log. Lalu, klik OK hingga Anda keluar dari system properties.

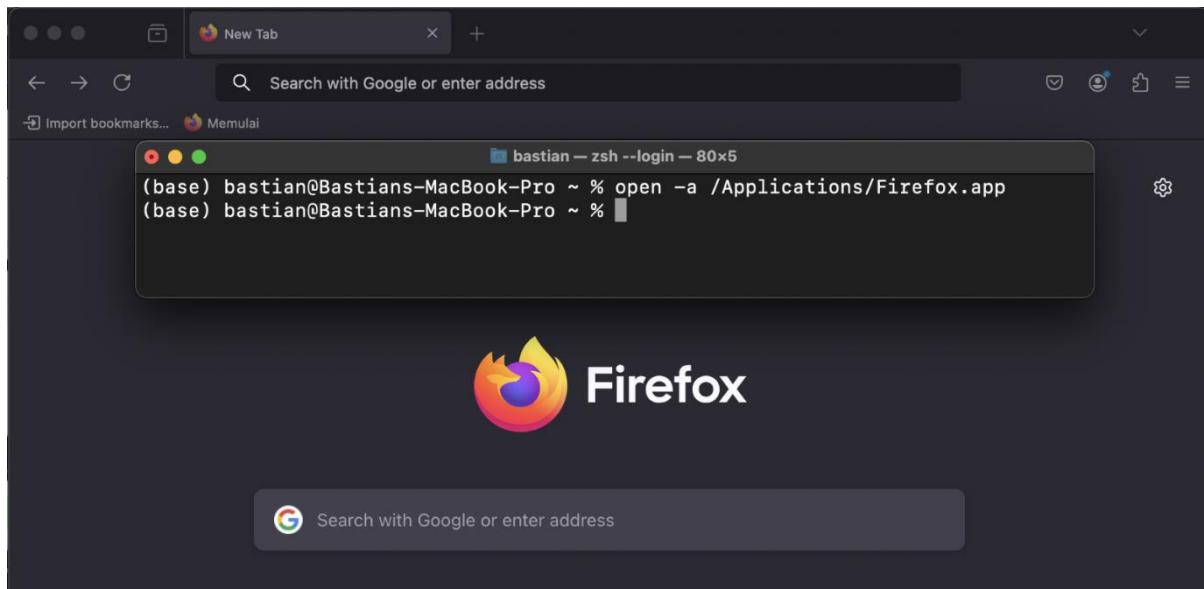




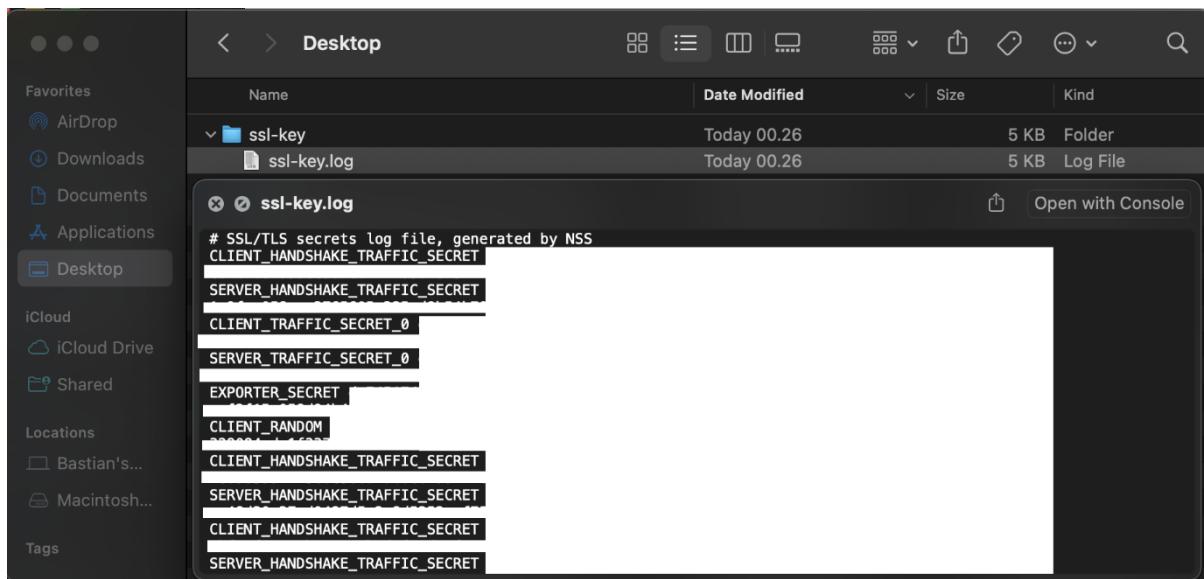
Memastikan Pre-master Keys Terisi

Kita telah mengonfigurasi environment variable SSLKEYLOGFILE. Lalu, bagaimana kita memastikan bahwa konfigurasi tersebut berfungsi seperti yang kita harapkan? Pertama, buka window browser baru **dari browser yang sedang tidak digunakan**. Anda dapat menghentikan browser yang sedang Anda gunakan (*close all tab*) dan membukanya lagi.

Anda disarankan menggunakan Firefox untuk melakukan pengujian. Jika Anda sedang menggunakan Firefox, tutup semua window-nya terlebih dahulu. Kemudian, untuk membuka Firefox, buka Terminal / Command Prompt Anda, dan ketik **open -a /Applications/Firefox.app** (untuk macOS), **firefox** (untuk Linux), atau **start firefox** (untuk Windows). Setelah itu, tekan tombol enter dan window baru dari Firefox akan muncul.

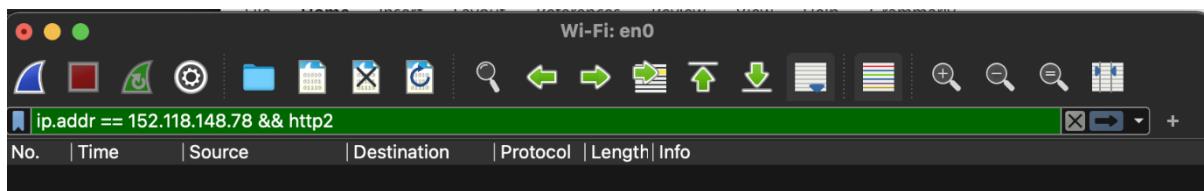


Setelah membuka browser window baru, cobalah untuk mengakses site yang secure, seperti SCeLE. Kemudian, buka directory tempat Anda menyimpan berkas log (nilai variabel SSLKEYLOGFILE). Buka file log, jika tidak empty dan Anda dapat melihat secret keys-nya, berarti Anda sudah benar.

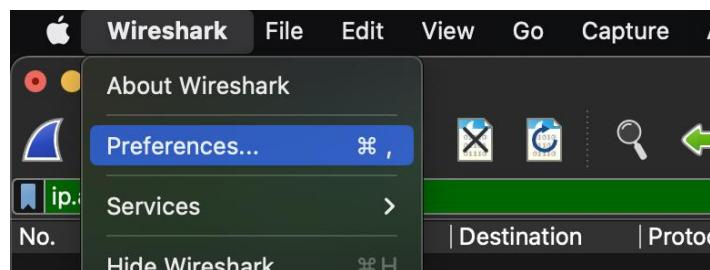


```
# SSL/TLS secrets log file, generated by NSS
CLIENT_HANDSHAKE_TRAFFIC_SECRET
SERVER_HANDSHAKE_TRAFFIC_SECRET
CLIENT_TRAFFIC_SECRET_0
SERVER_TRAFFIC_SECRET_0
EXPORTER_SECRET
CLIENT_RANDOM
CLIENT_HANDSHAKE_TRAFFIC_SECRET
SERVER_HANDSHAKE_TRAFFIC_SECRET
CLIENT_HANDSHAKE_TRAFFIC_SECRET
SERVER_HANDSHAKE_TRAFFIC_SECRET
```

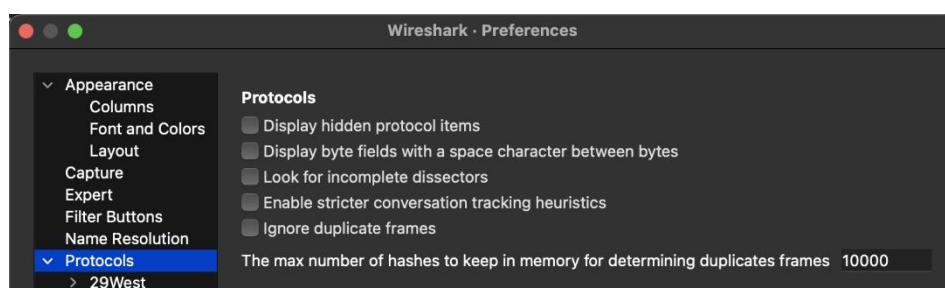
Cobalah untuk memulai sesi capturing. Start capture, lalu bukalah site secure misalnya SCeLE. Setelah beberapa saat, stop sesi capturing. Lalukan filter pada packets sehingga packet list hanya menampilkan HTTP packets dari atau menuju SCeLE. Hasilnya seharusnya seperti berikut.

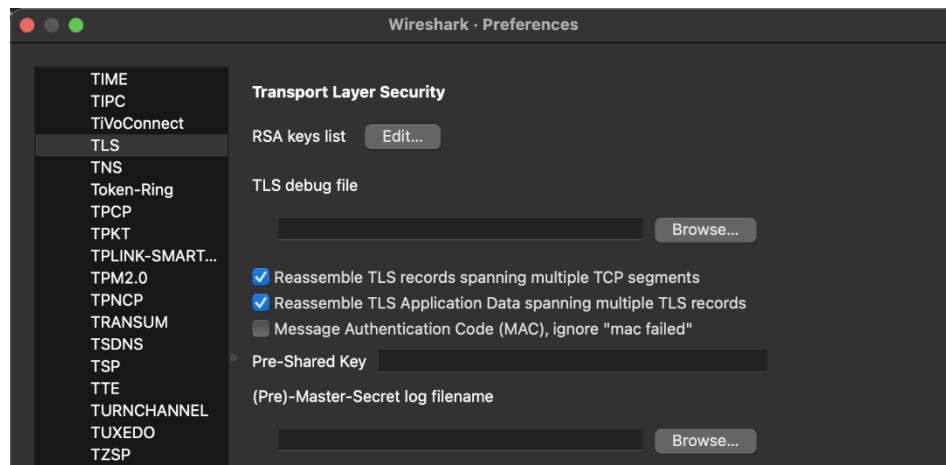


Lho, KOSONG? Ya. *Packet list*-nya kosong karena HTTP packets-nya telah terenkripsi. Sekarang, untuk mendekripsi HTTP packets-nya, kita harus memanfaatkan *pre-master keys* yang telah diisi dan disimpan dalam file log. Untuk melakukannya, buka Preferences... pada Wireshark. (Untuk Windows, hal ini dapat dilakukan dengan klik Edit pada Command Menu lalu pilih Preferences...)

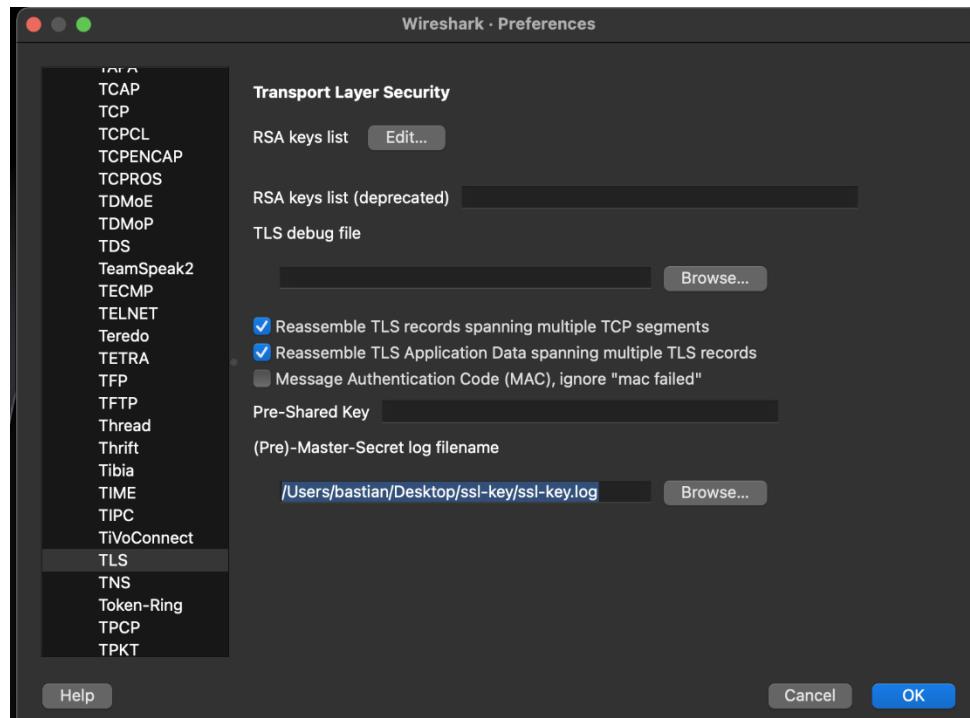


Expand opsi pada Protocols dan cari TLS.

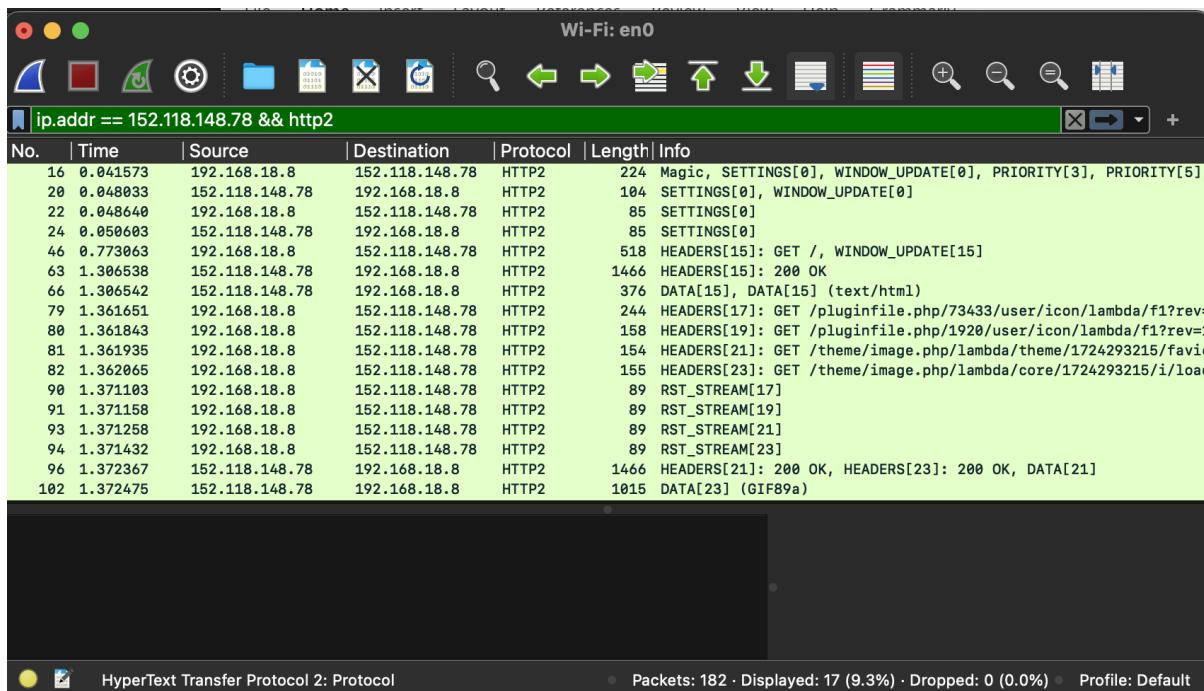




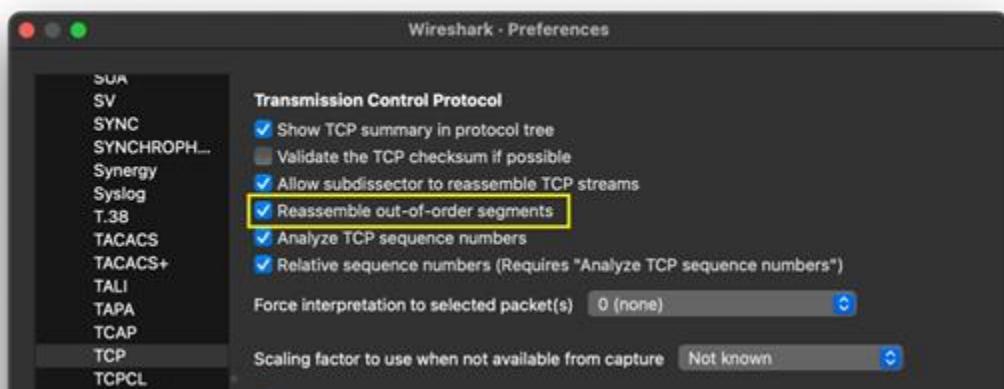
Isikan **(Pre)-Master Secret log filename** dengan value dari variabel SSLKEYLOGFILE variable yang telah Anda tentukan. Lalu, klik OK.



Sekarang, HTTP packets-nya dapat didekripsi. Packet list seharusnya akan menampilkan HTTP packets yang tadi kita capture.



Terkadang, terdapat beberapa packet yang tidak muncul di packet list karena merupakan *out-of-order segments*. Untuk mengatasinya, Anda dapat Kembali membuka Preferences dan memilih protocol TCP. Lalu, centang kotak “Reassemble out-of-order segment” dan klik OK.



Selamat! Anda baru saja mempelajari dasar-dasar penggunaan Wireshark. Semua yang telah Anda pelajari sejauh ini akan berguna tidak hanya dalam tugas ini, tetapi juga di tugas-tugas mendatang. Anda dapat mengakses dokumen ini lagi jika ingin me-review materi kembali. Jika Anda memiliki pertanyaan lebih lanjut atau masalah teknis, jangan ragu untuk mendiskusikannya dengan Asisten Dosen melalui server Discord.

Menginstal Aplikasi pada VM

Sering kali, kita mungkin perlu menginstal beberapa aplikasi baru di virtual machine kita. Di VM Ubuntu, Anda dapat melakukannya dengan command **apt**. Perintah ini juga merupakan solusi yang diberikan oleh VM Ubuntu saat kita mencoba menggunakan aplikasi yang belum kita instal.

```
farkhans-new@New-Ubuntu:~$ cowsay Hello
Command 'cowsay' not found, but can be installed with:
sudo apt install cowsay
farkhans-new@New-Ubuntu:~$ S■
```

Perintah untuk menginstal aplikasi menggunakan **apt** adalah **sudo apt install <app name>**. Anda juga dapat menambahkan parameter **-y** sehingga Anda tidak perlu menjawab "yes" setiap kali instalasi memerlukan persetujuan Anda. Sebagai contoh, kita akan mencoba menginstal **cowsay** di VM Ubuntu. Outputnya akan seperti ini.

```
farkhans-new@New-Ubuntu:~$ sudo apt install cowsay -y
[sudo] password for farkhans-new:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  wmdocker
Use 'sudo apt autoremove' to remove it.
Suggested packages:
  filters cowsay-off
The following NEW packages will be installed:
  cowsay
0 upgraded, 1 newly installed, 0 to remove and 8 not upgraded.
Need to get 18.6 kB of archives.
After this operation, 93.2 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 cowsay all 3.03+dfsg2-8 [18.6 kB]
Fetched 18.6 kB in 2s (11.2 kB/s)
Selecting previously unselected package cowsay.
(Reading database ... 212283 files and directories currently installed.)
Preparing to unpack .../cowsay_3.03+dfsg2-8_all.deb ...
Unpacking cowsay (3.03+dfsg2-8) ...
Setting up cowsay (3.03+dfsg2-8) ...
Processing triggers for man-db (2.10.2-1) ...
farkhans-new@New-Ubuntu:~$
```

Sekarang, mari kita coba aplikasi yang baru saja kita instal. Pada contoh ini, kita akan menjalankan **cowsay "Hello, World!"**

```
farkhans-new@New-Ubuntu:~$ cowsay "Hello, World!"
< Hello, World! >
-----
 \  ^__^
  (oo)\_____
   (__)\       )\/\
    ||----w |
     ||     ||
```

Anda juga dapat memeriksa apakah instalasi Anda berhasil atau tidak dengan command **which**. Ketik **which <app name>** lalu tekan enter. Hal ini akan menampilkan direktori aplikasi Anda. Jika output kosong, aplikasi tersebut mungkin belum terinstall.

```
farkhans-new@New-Ubuntu:~$ which cowsay
farkhans-new@New-Ubuntu:~$
```

cowsay belum terinstall

```
farkhans-new@New-Ubuntu:~$ which cowsay
/usr/games/cowsay
```

cowsay sudah terinstall

Informasi dan Konfigurasi Umum pada Jaringan

IPConfig

IPConfig (atau [ifconfig](#) atau `ip a`) adalah CLI tools yang digunakan untuk menampilkan *network interface* dan melakukan konfigurasi *network settings* pada perangkat Anda. Windows menggunakan IPCConfig, sedangkan macOS menggunakan ifconfig. Jika alat ini digunakan tanpa menyertakan parameter apa pun, opsi *default* akan digunakan dan akan memberikan *output* kurang lebih sebagai berikut.

```
[base] bastian@Bastians-MacBook-Pro ~ % ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
    inet 127.0.0.1 netmask 0xffff0000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8018<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
anpii1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=400<CHANNEL_IO>
```

Contoh ifconfig di macOS

Di Windows, *network interface* yang biasanya digunakan untuk terhubung ke Internet adalah adaptor Wi-Fi. Beberapa informasi penting yang ditampilkan di Windows IPConfig adalah Connection-specific DNS Suffix (biasanya router, dalam hal ini ui.ac.id), IPv4 Address (IP Address perangkat Anda), Subnet Mask, dan Default Gateway. Perhatikan bahwa parameter default Windows tidak menampilkan semua informasi. Untuk menampilkan informasi lengkap dari semua interface, gunakan **ipconfig /all**.

Pada gambar di bawah kanan, beberapa detail baru kini ditampilkan, yaitu Description, Physical Address (MAC Address), Lease Obtained & Expired (timer untuk UI reauthentication), DHCP Server, dan DNS servers. Detail setiap informasi ini akan dipelajari di kemudian hari. Namun, pada tugas ini, bagian yang paling penting adalah IPv4 Address.

```
C:\>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 1:

  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 3:

  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Wireless LAN adapter Wi-Fi:

  Connection-specific DNS Suffix . :
  IPv6 Address . . . . . : 2001:HH8a:2083:809b::f55:3ce2:8c5d:cdae
  Temporary IPv6 Address . . . . . : 2001:HH8a:1:2083:809b:892d:f4b8:9e05 b826
  Link-local IPv6 Address . . . . . : fe80::88a8:6e3c:f85e:e6c4%6
  IPv4 Address . . . . . : 192.168.100.193
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : fe80::1%6
                             192.168.100.1

C:\>
```

Ipconfig

```
Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . . . . . : 
Description . . . . . : Realtek 8822CE Wireless LAN 802.11ac PCI-E NIC
Physical Address . . . . . : A8-93-4A-54-B5-9D
DHCP Enabled . . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
IPv6 Address . . . . . : 2001:448a:2083:809b:cf55:3ce2:8c5d:cdae(PREFERRED)
Temporary IPv6 Address . . . . . : 2001:448a:2083:809b:89d3:fbb8:9e65:8262(PREFERRED)
Link-local IPv6 Address . . . . . : fe80::88e:6e3c:f85e:e6c4%6(Preferred)
IPv4 Address . . . . . : 192.168.100.193(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained . . . . . : Friday, August 18, 2023 1:57:58 PM
Lease Expires . . . . . : Tuesday, August 22, 2023 8:01:52 AM
Default Gateway . . . . . : fe80::1%6
                                         192.168.100.1
DHCP Server . . . . . : 192.168.100.1
DHCPv6 IAID . . . . . : 78156618
DHCPv6 Client DUID . . . . . : 00-01-00-01-29-45-89-E8-7C-8A-E1-BD-6D-34
DNS Servers . . . . . : 2001:4489:208:102::2
                         2001:4489:202:102::2
                         192.168.100.1
NetBIOS over Tcpip. . . . . : Enabled

C:\>|
```

ipconfig /all

```
farkhan_syawal11@vm-2-farkhan-2106709125:~$ 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
        valid_lft forever preferred_lft forever
2: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1460 qdisc mq state UP group default qlen 1000
    link/ether 42:01:0a:8a:00:03 brd ff:ff:ff:ff:ff:ff
    inet 10.138.0.3/32 metric 100 scope global dynamic ens4
        valid_lft 86364sec preferred_lft 86364sec
    inet6 fe80::4001:aff:fe8a:3/64 scope link
        valid_lft forever preferred_lft forever
farkhan_syawal11@vm-2-farkhan-2106709125:~$
```

Contoh ip a di VM GCP

Sementara itu, pada VM di GCP, hanya *loopback* dan *network interface* ens4 yang ditampilkan. Beberapa informasi penting yang ditampilkan adalah *inet* (alamat IP privat dari VM GCP) dan tipe *interface* (BROADCAST, MULTICAST, dll.). Untuk menampilkan default gateway, Anda dapat mengetik **ip r | grep default** pada terminal dan menekan tombol enter.

```
farkhan_syawal11@vm-2-farkhan-2106709125:~$ ip r | grep default
default via 10.138.0.1 dev ens4 proto dhcp src 10.138.0.3 metric 100
farkhan_syawal11@vm-2-farkhan-2106709125:~$
```

Pada contoh di atas, default gateway-nya adalah 10.138.0.1. Sementara itu, 10.138.0.3 adalah private IP address dari VM instance.

ARP

Perintah arp menampilkan dan memodifikasi tabel alamat IP-physical yang digunakan oleh ARP (*address resolution protocol*). Perintah arp -a (di Windows/mac) akan menampilkan semua entri ARP yang saat ini ada di tabel ARP.

```
C:\Users\LENOVO IDEAPAD>arp -a

Interface: 192.168.56.1 --- 0x4
  Internet Address          Physical Address          Type
  224.0.0.22                01-00-5e-00-00-16      static
  239.255.255.250           01-00-5e-7f-ff-fa      static

Interface: 192.168.100.193 --- 0x7
  Internet Address          Physical Address          Type
  192.168.100.1              34-1e-6b-01-30-c2     dynamic
  224.0.0.22                01-00-5e-00-00-16      static
  239.255.255.250           01-00-5e-7f-ff-fa      static

C:\Users\LENOVO IDEAPAD>
```

Parameter **-v** (hanya di Windows) akan mengaktifkan *verbose mode* dan akan menampilkan informasi tambahan untuk setiap entri. Perintah ini akan menampilkan keseluruhan isi ARP table dari network interfaces perangkat Anda.

```
C:\Users\LENOVO IDEAPAD>arp -av

Interface: 127.0.0.1 --- 0x1
    Internet Address      Physical Address      Type
    224.0.0.22                           static

Interface: 192.168.56.1 --- 0x4
    Internet Address      Physical Address      Type
    224.0.0.22          01-00-5e-00-00-16      static
    239.255.255.250        01-00-5e-7f-ff-fa      static

Interface: 192.168.100.193 --- 0x7
    Internet Address      Physical Address      Type
    192.168.100.1         34-1e-6b-01-30-c2      dynamic
    224.0.0.22          01-00-5e-00-00-16      static
    239.255.255.250        01-00-5e-7f-ff-fa      static

Interface: 0.0.0.0 --- 0xffffffff
    Internet Address      Physical Address      Type
    224.0.0.22          01-00-5e-00-00-16      static

Interface: 0.0.0.0 --- 0xffffffff
    Internet Address      Physical Address      Type
    224.0.0.22          01-00-5e-00-00-16      static

Interface: 0.0.0.0 --- 0xffffffff
    Internet Address      Physical Address      Type
    224.0.0.22          01-00-5e-00-00-16      static

C:\Users\LENOVO IDEAPAD>
```

Pada Linux, Untuk menampilkan tabel ARP, jalankan command **ip neigh show**.

```
farkhan_syawalli1@vm-2-farkhan-2106709125:~$ ip neigh show
10.138.0.1 dev ens4 lladdr 42:01:0a:8a:00:01 REACHABLE
farkhan_syawalli1@vm-2-farkhan-2106709125:~$ 
```

Entri dalam tabel ARP dapat dihapus. Di Windows, Anda dapat melakukannya dengan menggunakan command arp -d. Sementara itu, untuk pengguna macOS dan Linux, Anda dapat menggunakan sudo ip -s -s neigh flush all. Harap perhatikan bahwa saat Anda menghapus entri ARP, seringkali tabel tidak akan sepenuhnya kosong, mungkin masih ada entri untuk default gateway. Jika Anda tertarik, Anda dapat mengeksplor lebih lanjut tentang ARP di [arp\(8\): change system ARP cache - Linux man page \(die.net\)](#) atau sumber lainnya.

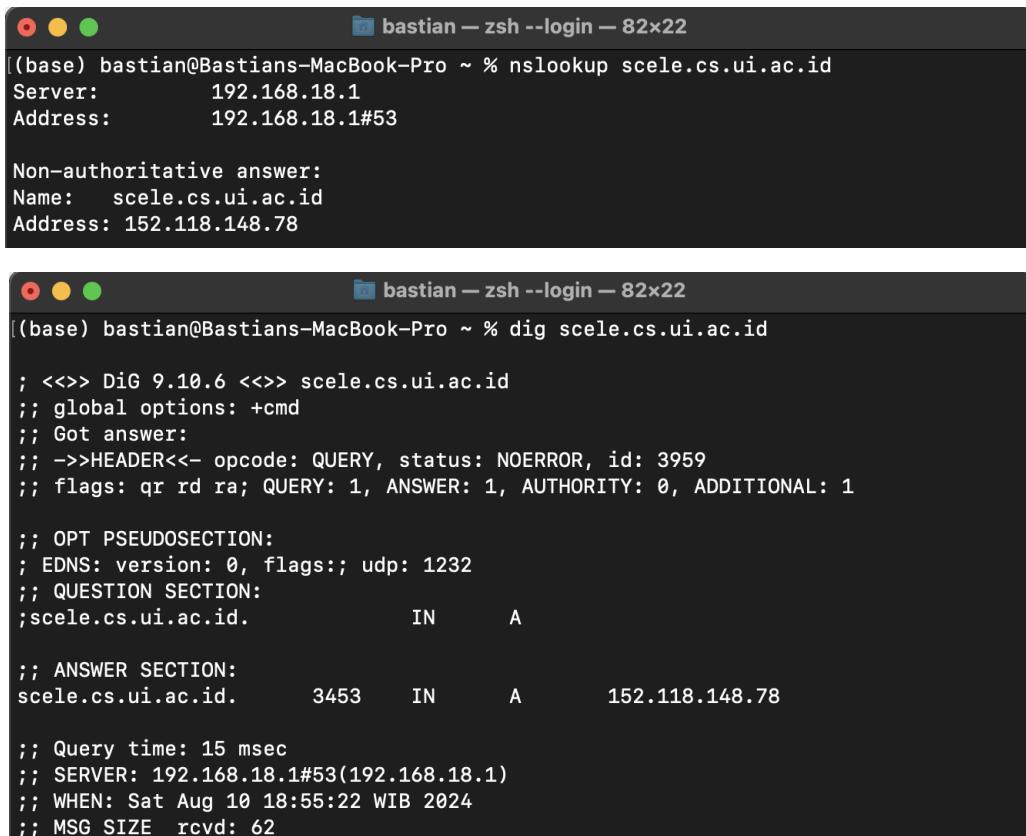
```
farkhan_syawalli1@vm-2-farkhan-2106709125:~$ ip neigh show
10.138.0.1 dev ens4 lladdr 42:01:0a:8a:00:01 REACHABLE
farkhan_syawalli1@vm-2-farkhan-2106709125:~$ sudo ip -s -s neigh flush all
10.138.0.1 dev ens4 lladdr 42:01:0a:8a:00:01 ref 1 used 30/0/30 probes 1 REACHABLE

*** Round 1, deleting 1 entries ***
*** Flush is complete after 1 round ***
farkhan_syawalli1@vm-2-farkhan-2106709125:~$ ip neigh show
10.138.0.1 dev ens4 lladdr 42:01:0a:8a:00:01 REACHABLE
farkhan_syawalli1@vm-2-farkhan-2106709125:~$ 
```

Pencarian Nama Domain

NSLookup dan DiG

NSLookup (Name Server Lookup) dan DiG (Domain information Groper) menampilkan informasi yang dapat digunakan untuk memeriksa dan mencari tahu infrastruktur DNS (Domain Name System). Keduanya melakukan pencarian DNS dan menampilkan respons dari *name server* yang dikueri. Kebanyakan DNS administrators menggunakan DiG untuk melakukan *troubleshooting* permasalahan DNS karena fleksibilitas, mudah digunakan, dan *output* yang jelas. Sementara itu, alat sejenis selain DiG cenderung mempunyai fungsionalitas yang lebih sedikit. Untuk menjalankan perintah ini, cukup jalankan **nslookup** atau **dig** dan diikuti dengan domain.



The image shows two screenshots of a macOS terminal window. Both screenshots have a title bar 'bastian — zsh --login — 82x22'. The first screenshot shows the output of the command 'nslookup scele.cs.ui.ac.id'. It displays the server address (192.168.18.1), port (53), and a non-authoritative answer for the name scele.cs.ui.ac.id with IP 152.118.148.78. The second screenshot shows the output of the command 'dig scele.cs.ui.ac.id'. It displays a detailed DNS query response, including the question section (scele.cs.ui.ac.id), answer section (IP 152.118.148.78), and various time and server-related fields.

```
(base) bastian@Bastians-MacBook-Pro ~ % nslookup scele.cs.ui.ac.id
Server:      192.168.18.1
Address:     192.168.18.1#53

Non-authoritative answer:
Name:   scele.cs.ui.ac.id
Address: 152.118.148.78

(base) bastian@Bastians-MacBook-Pro ~ % dig scele.cs.ui.ac.id
; <>> DiG 9.10.6 <>> scele.cs.ui.ac.id
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3959
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;scele.cs.ui.ac.id.      IN      A

;; ANSWER SECTION:
scele.cs.ui.ac.id.    3453    IN      A      152.118.148.78

;; Query time: 15 msec
;; SERVER: 192.168.18.1#53(192.168.18.1)
;; WHEN: Sat Aug 10 18:55:22 WIB 2024
;; MSG SIZE  rcvd: 62
```

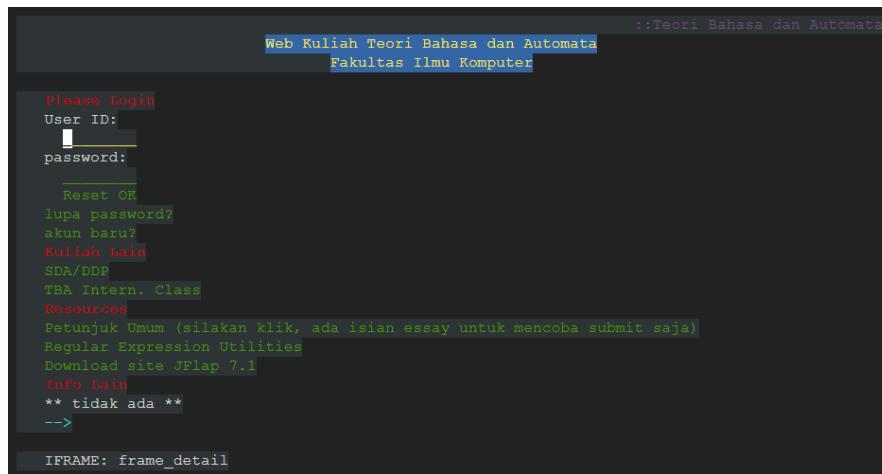
Perhatikan bahwa meskipun keduanya menampilkan informasi DNS, informasi yang ditampilkan DiG lebih informatif. Beberapa informasi penting yang ditampilkan adalah Question Section dan Answer Section. Bagian ini akan menampilkan informasi sesuai kueri dan respons yang didapat. Pada contoh di atas, query DNS “menanyakan alamat IP dari SCeLE Fasilkom UI” dan server DNS memberikan respons alamat IP-nya, yakni 152.118.148.78. Informasi lainnya akan dipelajari nanti, karena hanya pertanyaan dan jawaban DNS yang merupakan bagian yang relevan untuk tugas ini. Informasi lebih dapat dilihat di [nslookup | Microsoft Learn](#), [dig\(1\): DNS lookup utility - Linux man page \(die.net\)](#) atau sumber lainnya.

Network Browser dan Packet Capture

Lynx

Lynx adalah klien World Wide Web (WWW) berfitur lengkap untuk pengguna yang menjalankan perangkat yang memiliki tampilan *cursor-addressable* dan *character-cell* seperti terminal vt100,

emulator vt100 yang dijalankan pada Windows 95/NT atau Macintosh, dan perangkat *curses-oriented* lainnya. Lynx akan menampilkan dokumen HTML yang berisi tautan ke *file* yang berada di sistem lokal, serta *file* yang berada di sistem jarak luar yang menjalankan server Gopher, HTTP, FTP, WAIS, dan NNTP. Versi Lynx saat ini berjalan di Unix, VMS, Windows 95/NT, 386DOS dan OS/2 EMX. Lynx dapat digunakan untuk mengakses informasi di World Wide Web, atau untuk membangun sistem informasi yang tujuannya adalah untuk akses lokal. Untuk menjalankan Lynx, cukup ketik lynx diikuti dengan URL situs web, misalnya lynx aren.cs.ui.ac.id/tba/index.php.



Gunakan tombol panah untuk menavigasi tombol dan elemen HTML lainnya. Untuk keluar, tekan q dan enter. Dalam tugas ini, Lynx digunakan bersama dengan TCPDump untuk simulasi network capture. Informasi selengkapnya dapat dilihat di [lynx\(1\) - Linux man page \(die.net\)](#) atau sumber lain.

TCPDump

Tcpdump mencetak deskripsi isi packet pada sebuah *network interface* yang cocok dengan ekspresi *boolean*. Tcpdump juga bisa dijalankan dengan parameter -w, yang memungkinkan Anda untuk menyimpan data paket ke dalam *file* untuk dianalisis nantinya, dan/atau dengan parameter -r, yang memungkinkan Anda untuk melihat isi dari *file* paket yang tersimpan. Pada kebanyakan kasus, hanya paket yang cocok dengan ekspresi yang akan diproses tcpdump.

Jika dijalankan tanpa parameter -c, tcpdump akan terus menyadap paket sampai dihentikan oleh sinyal SIGINT (dibuat sendiri, misalnya dengan menekan tombol karakter untuk interupsi proses seperti ctrl + c) atau sinyal SIGTERM (khususnya dibuat dengan perintah kill(1)). Jika dijalankan dengan parameter -c, proses penyadapan paket akan berhenti apabila ada sinyal SIGINT atau SIGTERM atau banyak paket yang disadap sudah sesuai dengan parameter yang diberikan.

Ketika tcpdump selesai menyadap paket, tcpdump akan menampilkan banyaknya:

- paket “captured” (banyaknya paket yang diterima dan diproses tcpdump);
- paket “received by filter” (bergantung pada OS yang digunakan saat menjalankan tcpdump, dan mungkin juga bergantung pada konfigurasi OS – jika sebuah filter dispesifikasikan di *command line*, pada beberapa OS, tcpdump akan menghitung semua paket, terlepas paketnya cocok atau tidak dengan filter, serta terlepas tcpdump sudah atau belum membaca dan memproses paketnya (jika paketnya ternyata memang cocok dengan filter)).

Pada beberapa OS lainnya, tcpdump hanya menghitung paket yang cocok dengan filter

terlepas tcpdump sudah atau belum membaca dan memprosesnya, dan pada OS lainnya, tcpdump hanya menghitung paket yang cocok dengan filter dan diproses oleh tcpdump);

- paket “*dropped by kernel*” (banyaknya paket yang di-*drop*, bisa terjadi karena kurangnya *buffer space*, oleh mekanisme penyadapan paket pada OS yang digunakan, jika OS memberikan informasi tersebut ke aplikasi; jika tidak, informasi tersebut akan ditampilkan dengan 0).

Untuk melakukan penyadapan paket dengan tcpdump, cukup jalankan sudo tcpdump.

Perintah `default` digunakan untuk menyadap paket pada semua *interface* dan menampilkannya di terminal. Untuk menghentikan proses, cukup tekan tombol Ctrl/Command + C. Anda dapat menambahkan parameter `-w` agar tcpdump dapat menyimpan paket di suatu *file*. Anda juga dapat melakukan penyadapan pada *interface* tertentu. Tcpdump juga menyediakan opsi untuk mengecek *interface* yang tersedia menggunakan parameter `-D`.

```
██████████ bastian - zsh --login - 89x19
(base) bastian@Bastiens-MacBook-Pro ~ % sudo tcpdump -D
1.en0  [Up, Running]
2.awd10  [Up, Running]
3.llw0  [Up, Running]
4.utun0  [Up, Running]
5.ap1  [Up, Running]
6.utun1  [Up, Running]
7.utun2  [Up, Running]
8.lo0  [Up, Running, Loopback]
9.anp1  [Up, Running]
10.bridge0  [Up, Running]
11.anp1i  [Up, Running]
12.en1  [Up, Running]
13.en2  [Up, Running]
14.en3  [Up, Running]
15.en4  [Up, Running]
16.gif0  [none]
17.stf0  [none]
(base) bastian@Bastiens-MacBook-Pro ~ %
```

VM GCP menggunakan ens4 sebagai *network interface* utama. Perintah untuk menyadap paket dari *interface* ens4 dan menyimpannya di suatu file adalah sudo tcpdump -i ens4 -w [filename].pcap.

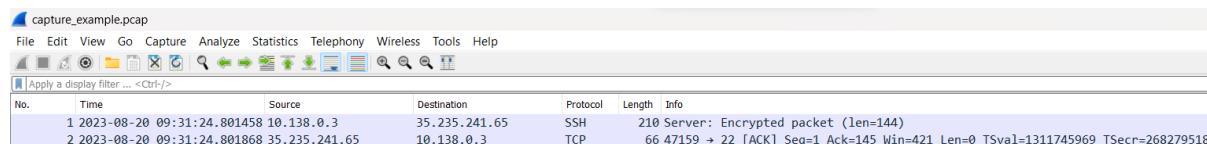
```
farkhan_syawal11@vm-2-farkhan-2106709125:~$ sudo tcpdump -i ens4 -w capture_example.pcap
tcpdump: listening on ens4, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C2 packets captured
3 packets received by filter
0 packets dropped by kernel
```

Tcpdump akan menyadap paket dari *interface* ens4 dan menyimpannya di capture_example.pcap. Tekan tombol Ctrl/Command + C untuk menghentikan proses. File-nya pun tersimpan dan sudah

dapat dilihat. File tersebut dapat disimpan di perangkat lokal dan dibuka di Wireshark atau langsung dibuka menggunakan tcpdump. Gunakan perintah sudo tcpdump -r [filename].pcap untuk membaca isi file menggunakan tcpdump. Pada contoh ini, perintahnya menjadi sudo tcpdump -r capture_example.pcap.

```
farkhan_syawali1@vm-2-farkhan-2106709125:~$ sudo tcpdump -r capture_example.pcap
reading from file capture_example.pcap, link-type EN10MB (Ethernet), snapshot length 262144
02:31:24.801458 IP vm-2-farkhan-2106709125.c.jarkom-farkhan.internal.ssh > 35.235.241.65.47159
: Flags [P.], seq 1162402330:1162402474, ack 2685546879, win 501, options [nop,nop,TS val 2682
79518 ecr 1311745942], length 144
02:31:24.801868 IP 35.235.241.65.47159 > vm-2-farkhan-2106709125.c.jarkom-farkhan.internal.ssh
: Flags [.], ack 144, win 421, options [nop,nop,TS val 1311745969 ecr 268279518], length 0
```

Tampilan dari tcpdump sulit dibaca. Oleh karena itu, kita menggunakan Wireshark. Unduh file tersebut dan buka di Wireshark.



Sekarang paket hasil penyadapan lebih mudah dibaca dan dipahami. Informasi lebih lanjut dapat dilihat di [tcpdump\(8\): dump traffic on network - Linux man page \(die.net\)](#), [Visio-tcpdump.vsd \(packetlife.net\)](#) atau sumber lainnya.

IP Socket dan Informasi Port

Netstat/SS

[Netstat](#) (or [ss di linux modern](#)) adalah network inspector tool yang menampilkan active TCP connections, port tempat komputer mendengarkan, Ethernet statistics, IP routing table, IPv4 statistics (untuk IP, ICMP, TCP, and UDP protocols), dan IPv6 statistics (untuk IPv6, ICMPv6, TCP over IPv6, dan UDP over IPv6 protocols). Digunakan tanpa parameter, perintah ini menampilkan koneksi TCP aktif. Di Linux, netstat sudah tidak digunakan lagi dan diganti dengan ss. macOS dan Windows masih menggunakan netstat. Jika tools ini digunakan tanpa parameter, konfigurasi default akan digunakan.

```
bastian -- zsh --login -- 99x20
(base) bastian@Bastians-MacBook-Pro ~ % netstat
Active Internet connections
Proto Recv-Q Send-Q Local Address          Foreign Address        (state)
tcp4      0      0 192.168.18.8.50582    52.123.252.31.https  ESTABLISHED
tcp4      0      0 192.168.18.8.50579    20.189.173.16.https  ESTABLISHED
tcp4      0      0 192.168.18.8.50578    13.107.138.10.https  ESTABLISHED
tcp4      0      0 192.168.18.8.50577    sc-in-f139.1e100.https ESTABLISHED
tcp4      0      0 192.168.18.8.50576    sc-in-f139.1e100.https ESTABLISHED
tcp4      0      0 192.168.18.8.50575    52.108.79.28.https  ESTABLISHED

farkhan_syawali1@vm-2-farkhan-2106709125:~$ ss
Netid State      Recv-Q Send-Q           Local Address:Port           Peer Address:Port  Process
u_dgr ESTAB      0      0           /run/chrony/chronyd.sock 16689          * 0
u_dgr ESTAB      0      0           /run/systemd/notify 14340          * 0
u_dgr ESTAB      0      0           /run/systemd/journal/dev-log 180          * 0
u_dgr ESTAB      0      0           /run/systemd/journal/socket 182          * 0
u_str ESTAB      0      0           * 16998                  * 15971
u_str ESTAB      0      0           /run/systemd/journal/stdout 15966          * 16966
u_dgr ESTAB      0      0           * 14341                  * 14342
u_str ESTAB      0      0           * 16758                  * 16762
u_dgr ESTAB      0      0           * 20005                  * 20004
u_dgr ESTAB      0      0           * 19975                  * 180
```

Pada gambar di atas, netstat pada macOS hanya akan menampilkan koneksi aktif yang dimulai selama sesi. Sementara itu ss akan menampilkan semua koneksi aktif pada saat dijalankan.

Perhatikan bahwa di ss terdapat soket non-TCP dan UDP yang berjalan. Untuk hanya menampilkan koneksi TCP dan UDP beserta PID terkaitnya, jalankan **netstat -anv -p TCP -p UDP** (untuk macOS), **ss -aetu** (untuk Linux), dan **netstat -ano** (untuk Windows)

```
(base) bastian@Bastians-MacBook-Pro ~ % netstat -anv -p TCP -p UDP
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         (state)      rhiwat shiwat    pid   epid   state   options
udp4       0      0 192.168.18.8.56301     74.125.24.94.443        1048576 29040    826      0 0x0102 0x00000000
udp4       0      0 192.168.18.8.58351     74.125.68.95.443        1048576 29040    826      0 0x0102 0x00000000
udp4       0      0 *.*.59380          *.*.              786896 9216      345      0 0x0100 0x00000000
udp4       0      0 *.*.*              *.*.              786896 9216      387      0 0x0000 0x00000000
udp4       0      0 *.*.*              *.*.              786896 9216      557      0 0x0080 0x00000000
udp4       0      0 *.*.62228          *.*.              524288 9216      593      0 0x0100 0x00000000
udp4       0      0 *.*.53039          *.*.              524288 9216      593      0 0x0100 0x00000000
udp4       0      0 *.*.*              *.*.              786896 9216      623      0 0x0000 0x00000000
udp46      0      0 *.*.5953           *.*.              786896 9216      826      0 0x0100 0x00000204
farkhan sysctl@vm-2-farkhan:~$ ss -aetu
(farkhan sysctl@vm-2-farkhan:~$ ss -aetu
Netid State Recv-Q Send-Q Local Address:Port          Peer Address:Port      Process
udp  UNCONN 0      0 127.0.0.53:domain          0.0.0.0:*          uid:101 ino:16520 sk:5e cgroup:/system.slice/systemd-resolved.service <-
udp  UNCONN 0      0 10.138.0.3:en4:bootpc        0.0.0.0:*          uid:100 ino:16481 sk:5f cgroup:/system.slice/systemd-networkd.service <-
udp  UNCONN 0      0 127.0.0.1:323            0.0.0.0:*          ino:16687 sk:60 cgroup:/system.slice/chrony.service <-
udp  UNCONN 0      0 [*]:323             [*]:*              ino:16689 sk:61 cgroup:/system.slice/chrony.service vconsole:1 <-
tcp  LISTEN 0      4096 127.0.0.1:5353           0.0.0.0:*

```

Perintah ini sangat berguna untuk memeriksa apakah port khusus sedang digunakan oleh suatu proses, atau untuk menghentikan proses menggunakan port tertentu.

Ping

Ping menggunakan datagram *ECHO_REQUEST* yang wajib dalam protokol ICMP untuk menghasilkan respons *ICMP ECHO_RESPONSE* dari sebuah *host* atau *gateway*. Datagram *ECHO_REQUEST* (atau “ping”) memiliki *header IP* dan ICMP yang diikuti oleh struktur *timeval* yang kemudian paket akan diisi oleh sejumlah byte “pad” yang acak. Ping secara fungsi utamanya digunakan untuk memeriksa apakah *public VM* atau *website* tersebut aktif atau tidak. Untuk menggunakan Ping anda dapat menambahkan perintah “ping [nama *public VM* atau *website*]” pada terminal seperti pada contoh berikut:

```
(base) bastian@Bastians-MacBook-Pro ~ % ping www.youtube.com
PING youtube-ui.l.google.com (64.233.170.190): 56 data bytes
64 bytes from 64.233.170.190: icmp_seq=0 ttl=57 time=18.988 ms
64 bytes from 64.233.170.190: icmp_seq=1 ttl=57 time=18.439 ms
64 bytes from 64.233.170.190: icmp_seq=2 ttl=57 time=26.720 ms
64 bytes from 64.233.170.190: icmp_seq=3 ttl=57 time=18.941 ms
64 bytes from 64.233.170.190: icmp_seq=4 ttl=57 time=18.074 ms
64 bytes from 64.233.170.190: icmp_seq=5 ttl=57 time=17.891 ms
64 bytes from 64.233.170.190: icmp_seq=6 ttl=57 time=18.511 ms
64 bytes from 64.233.170.190: icmp_seq=7 ttl=57 time=26.864 ms
^C
--- youtube-ui.l.google.com ping statistics ---
8 packets transmitted, 8 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 17.891/20.553/26.864/3.619 ms
(base) bastian@Bastians-MacBook-Pro ~ %
```

```
C:\>ping www.youtube.com

Pinging youtube-ui.l.google.com [2404:6800:4003:c06::be] with 32 bytes of data:
Reply from 2404:6800:4003:c06::be: time=288ms
Reply from 2404:6800:4003:c06::be: time=267ms
Reply from 2404:6800:4003:c06::be: time=278ms
Reply from 2404:6800:4003:c06::be: time=337ms

Ping statistics for 2404:6800:4003:c06::be:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 267ms, Maximum = 337ms, Average = 292ms
```

Pada gambar di atas, kita dapat melihat bahwa YouTube merespons Ping kita secara konsisten. Hal ini berarti bahwa YouTube dapat diakses dan siap melayani pengguna. Perhatikan bahwa di Linux dan macOS, konfigurasi Ping default diatur untuk berhenti hanya ketika terputus (dengan CTRL / ^C) atau dihentikan, berbeda dengan paket Windows yang mana hanya menampilkan 4 saja. Anda juga dapat memilih ukuran paket yang akan dikirim dan jumlah paket yang akan dikirim menggunakan -l dan -n. Informasi selengkapnya dapat dilihat pada [ping\(8\) - Linux man page \(die.net\)](#) atau sumber lainnya.

Tracert

Tracert/traceroute berfungsi untuk melacak rute paket-paket yang diambil dari jaringan IP dalam perjalannya menuju tujuan *host* tertentu. Tracert menggunakan *time to live* (TTL) pada protocol IP dan berusaha untuk mendapatkan respons ICMP *TIME_EXCEEDED* dari setiap *gateway* sepanjang jalur menuju *host*. Parameter utama yang dibutuhkan pada tracert adalah nama atau alamat IP dari tujuan *host*. *Optional packet length* merupakan ukuran total dari *probing packet* (secara *default* memiliki 60 byte untuk IPv4 dan 80 byte untuk IPv6). Ukuran yang ditentukan bisa diabaikan dalam beberapa situasi atau ditingkat sampai mendapatkan nilai minimal. Pada lingkungan jaringan yang modern, metode *traceroute* yang tradisional tidak selalu bisa diterapkan karena penggunaan *firewall* yang cukup luas seperti menyaring port UDP atau ICMP echoes.

```
bastian — zsh --login — 89x13
(base) bastian@Bastians-MacBook-Pro ~ % traceroute www.google.com
traceroute to forcesafesearch.google.com (216.239.38.120), 64 hops max, 52 byte packets
 1  192.168.18.1 (192.168.18.1)  3.875 ms  3.455 ms  3.141 ms
 2  * * 10.57.0.1 (10.57.0.1)  16.381 ms
 3  103.175.229.60 (103.175.229.60)  7.002 ms
     103.175.229.197 (103.175.229.197)  6.758 ms
     103.175.229.208 (103.175.229.208)  6.736 ms
 4  172.16.11.12 (172.16.11.12)  5.726 ms  5.667 ms  5.349 ms
 5  103.47.134.209 (103.47.134.209)  6.946 ms  7.309 ms  6.385 ms
 6  142.250.175.134 (142.250.175.134)  19.231 ms *  20.017 ms
 7  * * *
 8  any-in-2678.1e100.net (216.239.38.120)  30.113 ms  18.005 ms  19.358 ms
(base) bastian@Bastians-MacBook-Pro ~ %
```

```
C:\>tracert www.google.com
Tracing route to forcesafesearch.google.com [64:ff9b::d8ef:2678]
over a maximum of 30 hops:
 1      5 ms       4 ms       2 ms  2400:9800:300:2f05::26
 2      *          *          * Request timed out.
 3     29 ms      36 ms      40 ms  2400:9800:a:410::1
 4     35 ms      26 ms      23 ms  64:ff9b::70d7:24ea
 5     51 ms      44 ms      *      64:ff9b::4a7d:76f9
 6     56 ms      41 ms      46 ms  64:ff9b::4a7d:76f8
 7     63 ms      60 ms      43 ms  64:ff9b::d155:f8d1
 8     54 ms      49 ms      46 ms  64:ff9b::480e:e865
 9     57 ms      40 ms      47 ms  64:ff9b::d8ef:2678

Trace complete.
```

Untuk informasi lebih lanjut dapat Anda lihat pada [traceroute\(8\) – Linux man page \(dia.net\)](#) atau sumber lainnya.

Spesifikasi

Pada bagian ini, Anda akan mendemonstrasikan kemampuan Anda dalam menggunakan networking tools dengan melakukan beberapa tugas. Bagian tugas ini akan dinilai dan **Anda akan diminta untuk melaporkan proses Anda sesuai dengan instruksi yang diberikan**. Harap luangkan waktu untuk membaca spesifikasi terlebih dahulu dan pastikan bahwa Anda telah memiliki salinan lembar jawaban sendiri.

[30] Packet Capture and Decrypt with Wireshark

- a. Tambahkan *environment variable* SSLKEYLOGFILE baru ke *local device* Anda. Screenshot *environment variable* yang telah ditambahkan tersebut.
 - Untuk pengguna macOS dan Linux, Anda dapat men-screenshot output **echo \$SSLKEYLOGFILE** pada Terminal Anda.
 - Untuk pengguna Windows, Anda dapat men-screenshot list dari user variables dan *highlight*-lah variable yang baru saja Anda tambahkan.
- b. Screenshot direktori yang Anda gunakan untuk menyimpan file log. Pastikan Anda tidak mengakses apa pun sejak Anda menambahkan variable SSLKEYLOGFILE.
- c. Buka Wireshark dan browser baru (*close* dahulu semua tab-nya jika sebelumnya telah terbuka). Mulailah sesi capturing pada wireshark, lalu kunjungi link berikut (harap ganti <NPM> dengan NPM Anda sendiri): <https://http2lab.jarkom.cs.ui.ac.id/<NPM>/short>

Setelah situs web terbuka, screenshot page tersebut, termasuk URL-nya. Tunggu beberapa detik dan hentikan capturing. Simpan packet yang ditangkap menggunakan format nama berikut: **A01_[NPM]_http2.pcapng**.
- d. Screenshot direktori yang Anda gunakan untuk menyimpan file log.

- e. Buka file packet yang dicapture. Filter packet sehingga packet list menunjukkan HTTP/2 packets dari atau ke http2lab.jarkom.cs.ui.ac.id. Screenshot hasilnya.
- f. Sekarang, gunakan file log sehingga Anda dapat mendekripsi *encrypted packets* tersebut. Gunakan filter yang sama seperti sebelumnya dan screenshot hasilnya. Jika hasilnya tidak muat dalam satu screenshot, Tambahkan screenshot lain di bawahnya

[40] Packet Capture with TCPDump and Analysis

- a. Buka VM Anda pada GCP. Lakukan perintah untuk menampilkan IP dari ens4 network interface (*primary network interface*). Screenshot output-nya.
- b. Mulailah sesi capturing menggunakan tcpdump. Capture hanya HTTP packets yang melewati *instance primary network interface* Anda, dan dari atau ke server aren. Simpan hasil capture dalam file bernama **A01_[NPM]_aren.pcapng**.
- c. Mulai session lainnya dengan instance yang sama seperti sebelumnya. Pada session baru ini, buka <http://aren.cs.ui.ac.id/tba> menggunakan lynx. Setelah situs dibuka, screenshot halaman tersebut.
- d. Hentikan sesi capturing. Screenshot command yang Anda gunakan dan *output message*-nya setelah melakukan abort pada capturing session.
- e. Buka file menggunakan Wireshark. Tanpa filter apa pun, screenshot packets tersebut. Pastikan file name terlihat.
- f. Lakukan analisis terhadap setiap packet di packet list. Berikan penjelasan tentang tujuan dari setiap packet tersebut.

[30] Network Diagnostics

ARP

- a. Pada **VM instance GCP pertama dan kedua Anda**, jalankan command untuk menampilkan *network interfaces* dan ARP *table*-nya. Screenshot kedua output-nya dan pastikan *network interfaces*-nya terlihat
- b. Berikan penjelasan singkat mengenai hasilnya.

DiG

- a. Pada **VM instance GCP pertama Anda**, jalankan **dig youtube.com**. Screenshot output-nya.
- b. Berikan penjelasan singkat mengenai hasilnya.

Ping

- a. Pada **VM instance GCP pertama Anda**, jalankan command ping ke google.com. Gunakan parameter yang benar sedemikian sehingga command ping hanya mengirim tepat empat ECHO_REQUEST packets.
- b. Berikan penjelasan singkat mengenai hasil ping-nya.

Traceroute dan ip

- a. **Screenshot tampilan dashboard VM Instance pada GCP Anda. Pastikan kedua VM Instance terlihat informasinya dengan lengkap, terutama bagian internal IP nya.**

- b. Pada VM instance GCP pertama Anda, jalankan command tracert/traceroute terhadap VM instance GCP kedua. Screenshot output-nya
- e. Berikan penjelasan singkat mengenai hasilnya.

Netstat/ss

- a. Pada VM instance GCP pertama Anda, jalankan command ss dengan parameter sehingga output hanya menampilkan aktivitas dari TCP dan UDP connections. Screenshot output-nya termasuk identitas Anda. Jika terlalu panjang, Anda dapat menambahkan screenshot kembali dan taruh di bawah screenshot sebelumnya.
- b. Berikan penjelasan singkat mengenai hasilnya.

Informasi Pengumpulan Berkas

Lembar jawaban yang harus dikumpulkan untuk tugas ini harus disatukan dengan berkas lainnya (A01a, A01b, A01c, dan A01d). Perlu diperhatikan bahwa A01 ini menggunakan satu *template* lembar jawaban untuk semua bagian dan Anda harus menjawab semua bagian dalam satu dokumen tersebut. Untuk bagian **Introduction to Networking Tools** ini, Anda perlu untuk mengumpulkan berkas berikut:

1. File Laporan (lembar jawaban) yang telah disatukan dengan bagian lain dari A01. Laporan dikumpulkan dalam bentuk file PDF.

Format Penamaan: A01_[NPM].pdf

Contoh: A01_220777777.pdf

2. File *network packet* yang telah di-*capture* dengan *file extension* .pcapng.

Format Penamaan File :

- A01_[NPM]_ http2.pcapng (Wireshark capture)
- A01_[NPM]_ aren.pcapng (tcpdump capture)

Contoh :

- A01_220777777_http2.pcapng
- A01_220777777_aren.pcapng

Kumpulkan berkas bersamaan dengan berkas bagian tugas lainnya pada slot pengumpulan yang sesuai di SCeLE secara langsung. **Tidak perlu digabungkan ke dalam satu folder, dikompres dengan ZIP, atau semacamnya.**

Peraturan

Keterlambatan

Anda diharapkan dapat mengumpulkan hasil pekerjaan yang dilakukan sebelum batas waktu pengumpulan. Jika terdapat kondisi di mana Anda terpaksa terlambat mengumpulkan hasil pekerjaan, terdapat jangka waktu tambahan di mana Anda masih diperbolehkan mengumpulkan hasil pekerjaan dengan konsekuensi tertentu. Jika X adalah durasi setelah batas waktu pengumpulan yang ditetapkan sampai waktu Anda mengumpulkan hasil pekerjaan, Anda akan menerima penalti nilai pekerjaan sebagaimana diatur pada peraturan berikut ini:

- | | |
|---|--|
| • $X < 10$ menit | : Tidak ada penalti |
| • $10 \text{ menit} \leq X < 2 \text{ jam}$ | : 25% penalti |
| • $2 \text{ jam} \leq X < 4 \text{ jam}$ | : 50% penalti |
| • $4 \text{ jam} \leq X < 6 \text{ jam}$ | : 75% penalti |
| • $X \geq 6 \text{ jam}$ | : Cut-off (Pekerjaan anda tidak akan diterima) |

Plagiarisme

Anda diperbolehkan berdiskusi tentang pekerjaan Anda dengan peserta kuliah lain atau pihak lainnya, namun Anda harus memastikan bahwa semua pekerjaan yang dikumpulkan adalah murni hasil pekerjaan Anda sendiri. Anda dilarang keras melakukan tindak plagiarisme atau kecurangan akademik lainnya. Menurut kamus daring Merriam-Webster, plagiarisme berarti:

- Mencuri dan mengklaim (ide atau kata orang lain) sebagai milik sendiri
- Menggunakan hasil (karya/pekerjaan orang lain) sebagai milik sendiri
- Melakukan pencurian literatur/sastra
- Merepresentasikan ulang sebuah ide/produk yang sudah ada sebagai sesuatu yang bersifat baru dan orisinal.

Tim pengajar memiliki hak untuk meminta klarifikasi terkait dugaan ketidakjujuran akademik, terutama plagiarisme, dan memberikan konsekuensi berupa pengurangan nilai hasil pekerjaan atau pencabutan nilai (nilai diubah menjadi nol) untuk hasil pekerjaan yang terkonfirmasi dikerjakan secara tidak jujur.