TASK 5 JARINGAN KOMPUTER

Disusun oleh :

Nama : Ilham Kholfihim M

NIM : 09011281419043

JURUSAN SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA
2016

ANALISIS PERBANDINGAN CAPTURING NETWORK TRAFFIC MENGGUNAKAN WIRESHARK DAN NETSTAT

Capturing atau Menangkap trafik jaringan sangat berguna bagi administrator untuk menyelesaikan masalah-masalah yang muncul di jaringan, termasuk masalah keamanan. Hal ini berdasarkan fakta bahwa sejumlah penyerangan dimulai dengan penggunaan penyadap untuk melihat trafik jaringan dengan harapan melihat data-data penting yang ditransmisikan oleh user. Karena itu mempelajari kapabilititas dan keterbatasan software-software capturing packet menjadi bagian yang penting dari kemampuan networking.

Pada kegiatan ini saya akan membandingkan trafik jaringan yang dilalui oleh http request dengan menggunakan Wireshark dan Netstat (Network Statistic) pada command line terminal.

HTTP Requests merupakan Permintaan-permintaan dari source ke destination berisikan informasi tentang macam-macam data yang user inginkan.

Siklus dari HTTP Request umumnya terlihat seperti ini:

1. Seorang pengguna mengunjungi URL dari sebuah situs web.
2. Hal ini menciptakan permintaan yang diarahkan ke web server melalui internet (jaringan DNS itu, router dan switch) melalui HTTP (Hypertext Transfer Protocol).
3. Web server menerima permintaan HTTP dan merespon pengguna dengan halaman web (atau isi) yang diminta.

Setiap kali kita klik link dan mengunjungi halaman web, di balik layar kita membuat suatu permintaan, dan menerima respon dari web server. Perhatikan bahwa permintaan HTTP dapat dilakukan melalui berbagai macam jaringan.

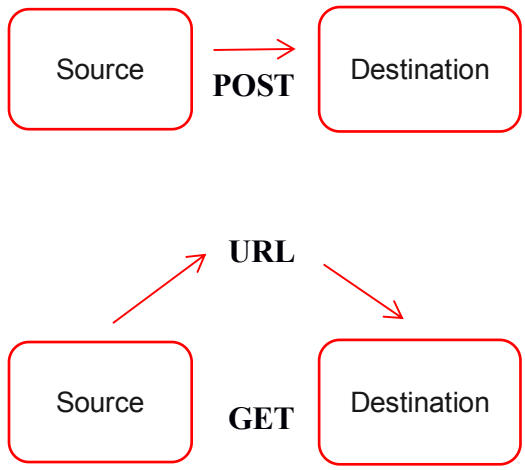
Salah satu item informasi yang dienkapsulasi pada permintaan HTTP adalah sebuah method. Hal ini yang memberitahu server macam-macam permintaan yang dibuat, Ada tiga protokol yang biasa digunakan : GET, POST, dan PUT.

1. GET adalah method HTTP paling sederhana dan digunakan sebagian besar untuk meminta resource tertentu dari server, dapat berupa halaman web, file gambar grafis, atau sebuah dokumen, dan lain-lain.

2. POST, Jenis permintaan ini didesain seperti browser dapat membuat permintaan kompleks dari server. Mereka didesain sehingga user, melalui browser, dapat mengirim banyak data ke server. Form kompleks secara umum dicapai dengan menggunakan permintaan POST, sebagaimana form sederhana yang memelukan proses upload file ke server.

   Satu perbedaan yang nyata antara method GET dan POST terletak pada cara mengirimkan data ke server. Seperti yang dinyatakan sebelumnya, GET hanya

menambahkan data ke URL yang akan mengirim. POST, di sisi lain, mengenkapsulasi atau menyembunyikan data di dalam body pesan (message body) yang dikirim. Ketika server menerima permintaan dan menentukan bahwa itu merupakan sebuah permintaan POST, dapat dilihat dari body pesan data tersebut.



3. PUT, Berfungsi untuk meng-upload representasi dari sumber tertentu
4. HTTP Response, HTTP merespon dari server yang berisi headers dan body pesan, seperti yang permintaan HTTP lakukan. Mereka menggunakan kumpulan header yang berbeda, meskipun demikian disini kita tidak perlu terlalu dalam membahasnya secara detail. Cukup dengan mengatakan bahwa headers berisi informasi tentang protokol HTTP yang digunakan pada server, sebagaimana tipe dari isi yang dienkapsulasi ke dalam body pesan. Nilai dari tipe isi adalah MIME-type. Ini akan memberitahu browser jika pesan berisi HTML, gambar, atau tipe lainnya.

1. WIRESHARK

Wireshark, atau dulunya dikenal sebagai Ethereal, adalah salah satu tool yang sangat powerfull sebagai senjata para analis keamanan jaringan. Sebagai analyzer packet jaringan, Wireshark dapat digunakan sebagai peer di dalam suatu jaringan dan mengamati trafik secara detail dalam berbagai level, mulai dari header packet hingga bit yang menyusan suatu paket. Karena wireshark menggunakan GUI, banyak pengguna memperoleh kemudahan grafis dalam menggunakan informasi yang terkandung di dalamnya.

2. NETSTAT

Netstat (network statistics) merupakan program berbasis teks yang berfungsi untuk memantau koneksi jaringan pada suatu komputer, baik itu jaringan lokal (LAN) maupun jaringan internet.

Berikut ini keterangan dari output netstat :

1. **Proto**. Kolom proto menunjukan jenis protokol yang dipakai bisa TCP atau UDP.
2. **Local Address**. Kolom ini menjelaskan alamat dan nomor port yang ada di komputer anda yang mana saat itu sedang aktif melakukan koneksi.
3. **Foreign Address**. Kolom ini menunjukan koneksi yang dituju oleh local address beserta nomor portnya. Contoh diatas saya sedang menghubungi server google melalui http (port 80) yang artinya saya sedang browsing google.
4. **State**. Kolom ini menunjukan status dari koneksi yang sedang terjadi. **ESTABLISED** artinya sudah terhubung dengan komputer lain dan siap mengirimkan data.

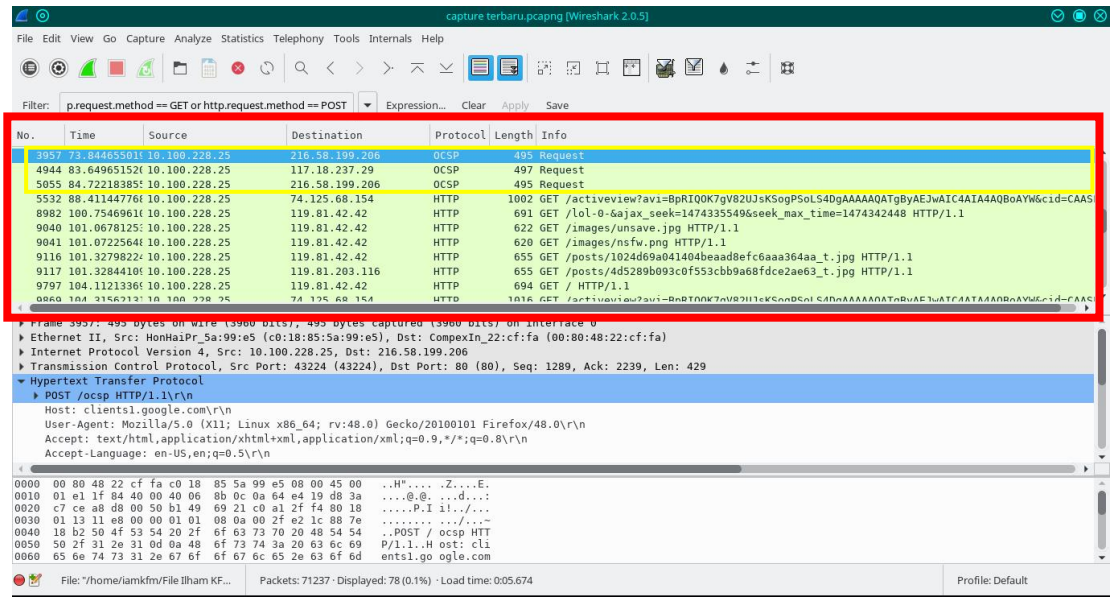
State yang biasa terjadi :

- **LISTENING** -> siap untuk melakukan koneksi.
- **SYN_SENT** -> mengirimkan paket SYN
- **SYN_RECEIVED** -> menerima paket SYN
- **ESTABLISHED** -> koneksi terjadi dan siap mengirimkan data.
- **TIME_WAIT** -> sedang menunggu koneksi

Perlu diperhatikan jika muncul state SYN_SENT dalam jumlah yang banyak dan terus menerus, efeknya koneksi internet anda menjadi sangat lambat.

ANALISA

Capturing data menggunakan wireshark dapat dilihat pada gambar di bawah ini



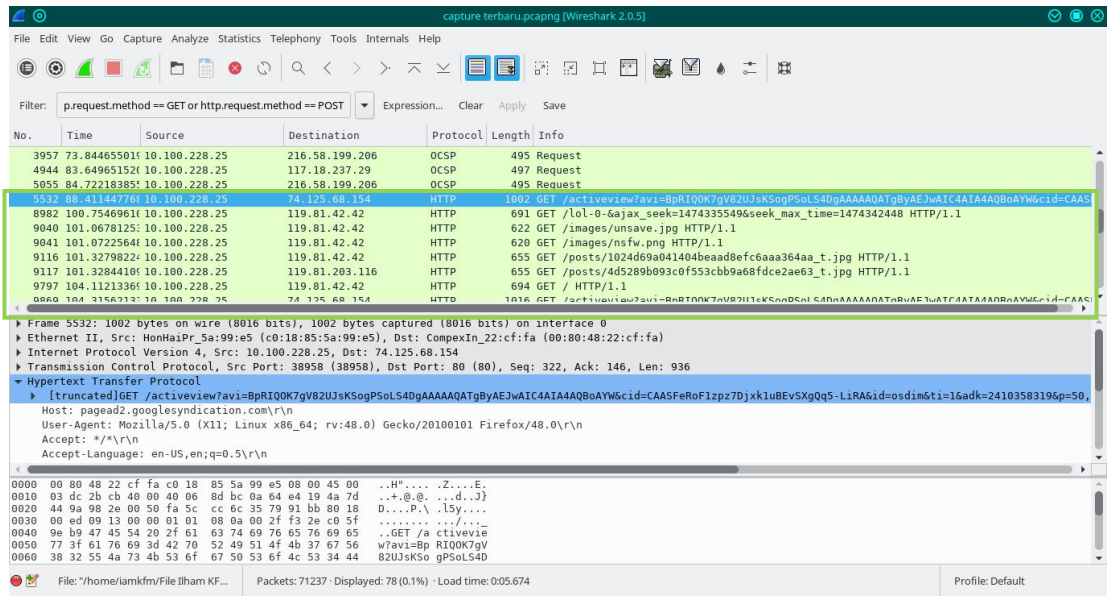
Hasil dari capturing dapat dilihat pada kotak berwarna merah, menampilkan bermacam-macam informasi, yaitu :

1. **No,** menampilkan informasi nomor paket ke-n yang diperoleh pada saat melakukan interkoneksi.
2. **Time,** menampilkan informasi waktu mengakses paket dimulai dari start capturing.
3. **Source,** menampilkan informasi IP dari sumber data atau pengakses.
4. **Destination,** menampilkan informasi IP dari tujuan data tersebut.
5. **Protokol,** menampilkan informasi Protokol yang digunakan pada saat mengakses data tersebut.
6. **Info,** menampilkan informasi aktivitas yang dilakukan oleh data seperti request, send, post, get, dan lain sebagainya.

Pada posisi pointer yang berada didalam kotak berwarna kuning terlihat aktivitas **request** yang sedang dilakukan oleh alamat IP **10.100.228.25** yang merupakan alamat IP yang saya gunakan yang merupakan IP Static Jaringan Hotspot di Fakultas Ilmu Komputer Universitas Sriwijaya yang meminta paket data ke alamat tujuan yaitu suatu alamat website http://1cak.com dengan alamat IP **74.125.68.154**

Dari proses capturing dapat dilihat suatu kondisi request dimana alamat website yang saya tuju tidak langsung diakses dari jaringan lokal yang saya gunakan, akan tetapi request dilakukan melalui beberapa rute yang melalui port **ocsp** ke alamat IP **216.58.199.206** lalu selanjutnya ke alamat IP **117.18.237.29** dan proses request terakhir ke alamat IP **216.58.199.286**, alamat-alamat tersebut merupakan alamat ISP dan juga alamat Server dari website yang saya tuju.

Setelah melakukan proses request ke server selanjutnya server mengirimkan kembali paket data ke source yang menyatakan bahwa paket data yang diminta terdapat pada server tersebut.



Lalu kemudian source meminta kembali paket data yang terdapat pada server dengan cara yang sama melalui beberapa rute seperti yang terlihat pada gambar di atas yang ditandai kotak berwarna hijau, Proses yang dilakukan oleh source untuk meminta data ke destination disini merupakan proses GET yang dilakukan melalui port http pada OSI layer proses ini terdapat pada layer ke-3 Layer Network. Pada capturing yang saya lakukan tidak terdapat proses POST karena saya tidak mengisikan form apapun pada website tersebut.

Dengan Menggunakan Wireshark kita bisa mendapatkan beberapa informasi yang lebih mendetail jika dibandingkan dengan menggunakan netstat pada CLI. Beberapa informasi yang diperoleh dapat dilihat pada gambar dibawah ini.

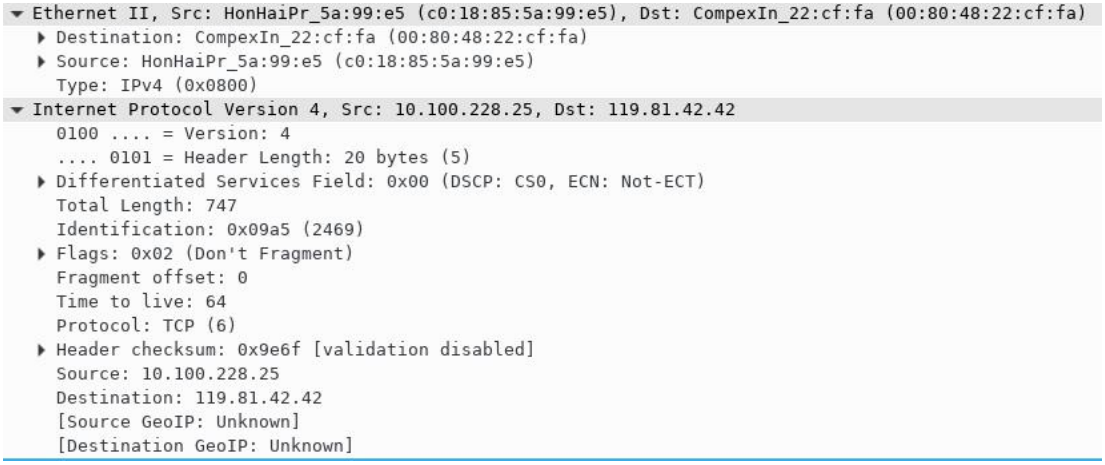
1. Informasi Frame

```
▾ Frame 10717: 761 bytes on wire (6088 bits), 761 bytes captured (6088 bits) on interface 0
    Interface id: 0 (wlp7s0b1)
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 20, 2016 10:34:36.640360566 WIB
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1474342476.640360566 seconds
    [Time delta from previous captured frame: 0.000202685 seconds]
    [Time delta from previous displayed frame: 0.157305632 seconds]
    [Time since reference or first frame: 106.800791806 seconds]
    Frame Number: 10717
    Frame Length: 761 bytes (6088 bits)
    Capture Length: 761 bytes (6088 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:http]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
```

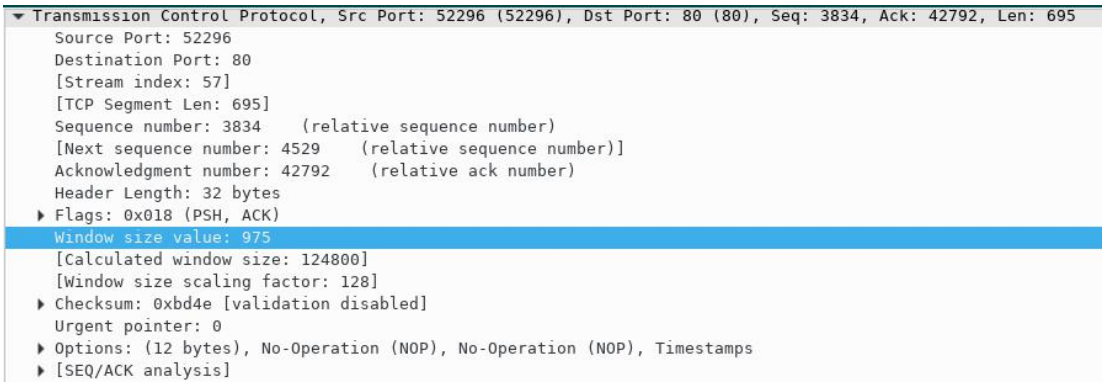
Pada capturing yang saya lakukan dapat diperoleh informasi frame seperti gambar diatas seperti waktu frame dieksekusi, jumlah frame, lebar frame, protokol yang digunakan oleh frame yaitu **ethernet:ip:tcp:http.** Pada OSI Layer aktivitas frame terdapat pada layer

ke-2 yaitu Datalink.

2. Informasi Interface Network yang digunakan seperti IP dan mac address



3. Informasi TCP



4. Informasi port yang digunakan

Pada gambar di atas dapat diperoleh informasi port yang dilalui source untuk sampai ke destination. Dalam hal ini saya mengakses website http://1cak.com dari web browser Mozilla Firefox dan system operasi Linux.

Jika di wireshark menampilkan informasi trafik jaringan secara mendetail lain halnya dengan netstat(network statistic) pada CLI di linux maupun CMD di windows.

Seperti yang telah dijelaskan di awal instruksi netstat menampilkan statistik koneksi jaringan dari suatu komputer. Pada gambar dibawah ini merupakan hasil dari capturing yang telah saya lakukan pada saat terjadinya proses request ke website http://1cak.com

Saya menggunakan website destination yang sama agar dapat dengan mudah memahami perbedaannya menggunakan wireshark dan netstat.

```
[sklibur@fukamuori iamkfm]$ netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0     54 fukamuori:45292         202.58.9.201:https      LAST_ACK
tcp        0      0 fukamuori:58258         edge-star-mini-sh:https ESTABLISHED
tcp        0      1 fukamuori:38600         a23-15-155-27.:www-http FIN_WAIT1
tcp        0      0 fukamuori:38606         a23-15-155-27.:www-http TIME_WAIT
tcp        0      0 fukamuori:50930         sin04s09-in-f206.:https ESTABLISHED
tcp        0      0 fukamuori:37082         202.79.210.118:https    TIME_WAIT
tcp        0      0 fukamuori:48378         104.244.43.145:https    TIME_WAIT
tcp        0      0 fukamuori:45294         202.58.9.201:https      TIME_WAIT
tcp        0      0 fukamuori:47496         sb-in-f95.1e100.n:https ESTABLISHED
tcp        0      0 fukamuori:36644         sc-in-f155.1e100.:https ESTABLISHED
tcp        0      0 fukamuori:51348         kul06s14-in-f194.:https ESTABLISHED
tcp        0      0 fukamuori:49600         luna.archlinux.or:https ESTABLISHED
tcp        0      0 fukamuori:44118         sa-in-f139.1e1:www-http TIME_WAIT
tcp        0      0 fukamuori:33612         74.125.200.84:https     ESTABLISHED
tcp        0      0 fukamuori:54806         74.125.200.94:https     ESTABLISHED
tcp        0      0 fukamuori:54944         74.125.68.157:https     ESTABLISHED
tcp        0      0 fukamuori:48382         104.244.43.145:https    TIME_WAIT
tcp        0      0 fukamuori:49448         74.125.130.156:https    ESTABLISHED
tcp        0      0 fukamuori:59864         216.58.196.46:https     ESTABLISHED
tcp        0      0 fukamuori:42276         104.244.42.136:https    ESTABLISHED
tcp        0      0 fukamuori:42348         216.58.199.198:https    ESTABLISHED
tcp        0      0 fukamuori:54852         74.125.68.157:https     ESTABLISHED
tcp        0      0 fukamuori:38934         172.217.24.228:https    ESTABLISHED
tcp        0      1 fukamuori:38602         a23-15-155-27.:www-http FIN_WAIT1
tcp        0      0 fukamuori:45300         202.58.9.201:https      TIME_WAIT
tcp        0      0 fukamuori:38604         a23-15-155-27.:www-http TIME_WAIT
tcp        0    124 fukamuori:55330         216.58.196.166:https    FIN_WAIT1
```

Dari gambar diatas dapat diperoleh beberapa informasi yaitu proto, local address, foreign address, dan state yang telah dijelaskan di awal. Dari gambar diatas diperoleh informasi sebagai berikut.

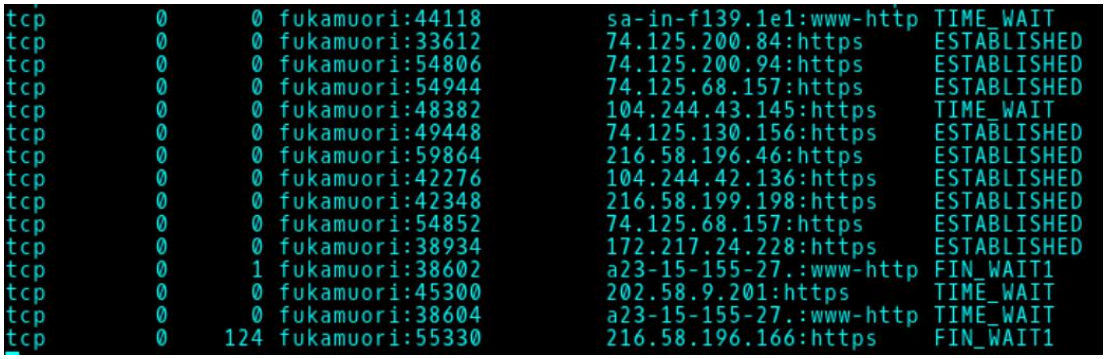
1. Proto
   Protokol yang digunakan yaitu tcp

2. Local Address
   Informasi Local Address yang diperoleh disini berupa nama host yang komputer yang saya gunakan yaitu **fukamuori** beserta nomor port yang digunakan. Biasanya informasi yang ditampilkan berupa alamat IP source karena beberapa pengaturan di kernel linux jadi yang ditampilkan adalah hostname.

3. Foreign Address
   Sama seperti penjelasan alamat IP tujuan pada wireshark sebelumnya alamat IP tujuan melalui beberapa rute. Pada instruksi netstat port alamat tujuan yang dalam hal ini http ditampilkan bersamaan dengan IP-nya.

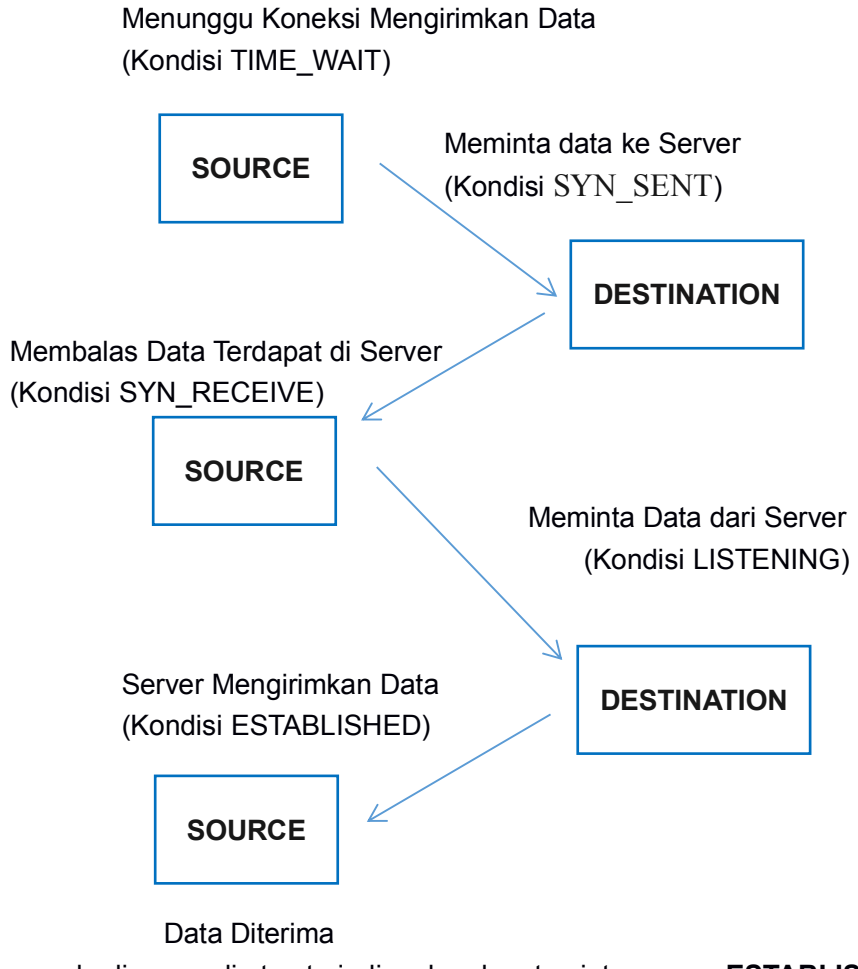
4. State

State inilah yang dapat perbedaannya paling mencolok antara penggunaan wireshark dan netstat. Pada wireshark telah dijelaskan sebelumnya kondisi state ini dijelaskan secara mendetail di dalam setiap paket yang di capture. Sedangkan pada netstat kondisi statistik ini hanya diekspresikan dalam suatu keadaan. Seperti capturing web browser yang saya lakukan dengan instruksi netstat pada gambar di bawah ini



```
tcp        0      0 fukamuori:44118         sa-in-f139.1e1:www-http TIME_WAIT
tcp        0      0 fukamuori:33612         74.125.200.84:https     ESTABLISHED
tcp        0      0 fukamuori:54806         74.125.200.94:https     ESTABLISHED
tcp        0      0 fukamuori:54944         74.125.68.157:https     ESTABLISHED
tcp        0      0 fukamuori:48382         104.244.43.145:https    TIME_WAIT
tcp        0      0 fukamuori:49448         74.125.130.156:https    ESTABLISHED
tcp        0      0 fukamuori:59864         216.58.196.46:https     ESTABLISHED
tcp        0      0 fukamuori:42276         104.244.42.136:https    ESTABLISHED
tcp        0      0 fukamuori:42348         216.58.199.198:https    ESTABLISHED
tcp        0      0 fukamuori:54852         74.125.68.157:https     ESTABLISHED
tcp        0      0 fukamuori:38934         172.217.24.228:https    ESTABLISHED
tcp        0      1 fukamuori:38602         a23-15-155-27.:www-http FIN_WAIT1
tcp        0      0 fukamuori:45300         202.58.9.201:https      TIME_WAIT
tcp        0      0 fukamuori:38604         a23-15-155-27.:www-http TIME_WAIT
tcp        0    124 fukamuori:55330         216.58.196.166:https    FIN_WAIT1
```

Dapat dilihat proses yang terjadi pada saat saya mengakses website http://1cak.com dengan alamat IP **74.125.200.84** yang melalui port tujuan http dimulai dari proses **TIME_WAIT** maksudnya adalah source sedang menunggu koneksi dari destination yang dianalogikan seperti diagram dibawah ini.



Setelah proses pada diagram di atas terjadi maka akan tercipta proses **ESTABLISHED.**

Itulah analisa perbandingan beberapa Informasi yang bisa diperoleh dengan menggunakan wireshark dan netstat masih banyak lagi informasi lainnya yang bisa diperoleh dari wireshark, untuk saat ini karena keterbatasan ilmu yang telah saya miliki, saya hanya bisa menjelaskan sebagian kecil proses 3 way handshake yang terjadi pada dunia jaringan komputer yang dapat dilihat melalui wireshark dan netstat.