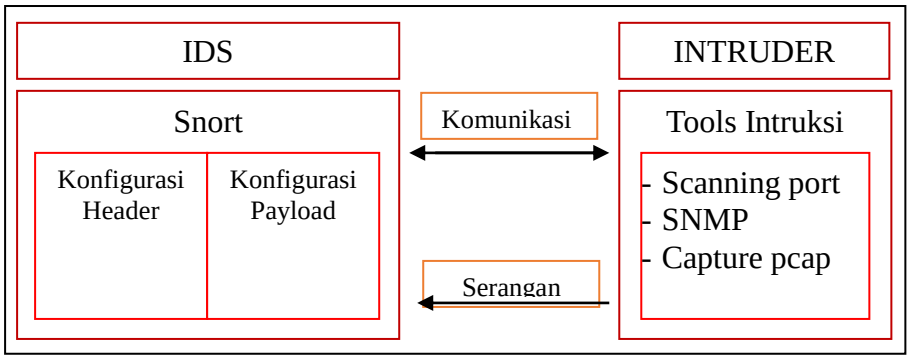
Nama : Riki Andika
NIM : 09011181320015

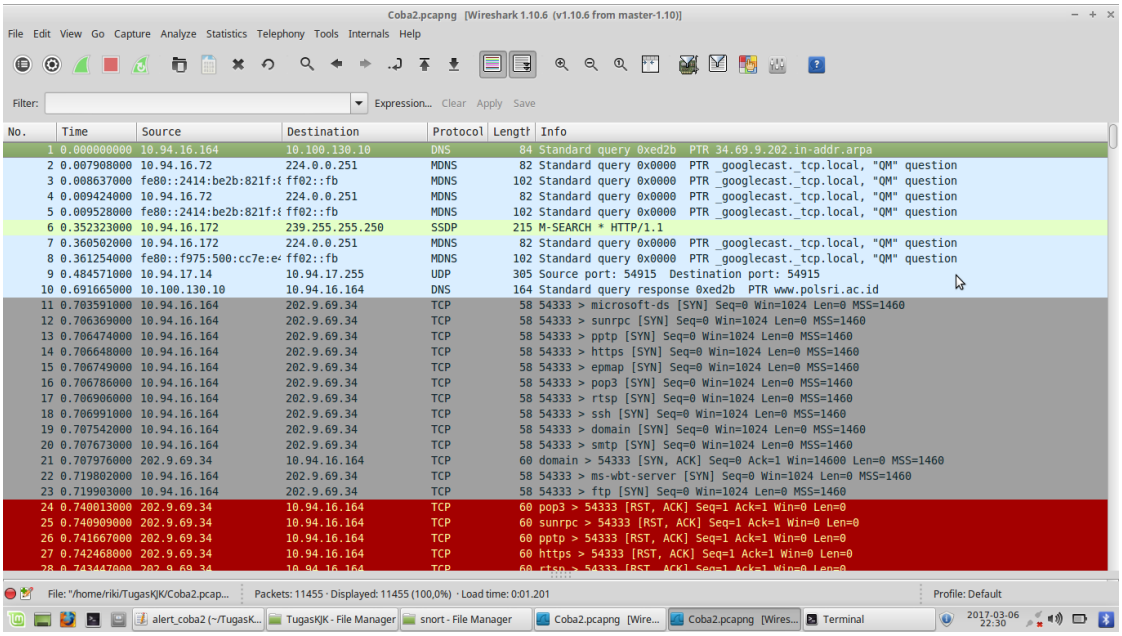
Intrusion Detection System (IDS) adalah sebuah sistem yang melakukan pengawasan terhadap traffic jaringan dan pengawasan terhadap kegiatan-kegiatan yang mencurigakan didalam sebuah sistem jaringan, yang selanjutnya akan disingkat dengan nama IDS. Jika ditemukan kegiatan-kegiatan yang mencurigakan berhubungan dengan traffic jaringan maka IDS akan memberikan peringatan kepada sistem atau administrator jaringan. Dalam banyak kasus IDS juga merespon terhadap traffic yang tidak normal/ anomali melalui aksi pemblokiran seorang user atau alamat IP (Internet Protocol). IDS sendiri muncul dengan beberapa jenis dan pendekatan yang berbeda yang intinya berfungsi untuk mendeteksi traffic yang mencurigakan didalam sebuah jaringan. Beberapa jenis IDS adalah : yang berbasis jaringan (NIDS) dan berbasis host (HIDS).

Ada IDS yang bekerja dengan cara mendeteksi berdasarkan pada pencarian ciri-ciri khusus dari percobaan yang sering dilakukan. Cara ini hampir sama dengan cara kerja perangkat lunak antivirus dalam mendeteksi dan melindungi sistem terhadap ancaman. Kemudian ada juga IDS yang bekerja dengan cara mendeteksi berdasarkan pada pembandingan pola traffic normal yang ada dan kemudian mencari ketidaknormalan traffic yang ada. Ada IDS yang fungsinya hanya sebagai pengawas dan pemberi peringatan ketika terjadi serangan dan ada juga IDS yang bekerja tidak hanya sebagai pengawas dan pemberi peringatan melainkan juga dapat melakukan sebuah kegiatan yang merespon adanya percobaan serangan terhadap sistem jaringan dan komputer, berikut topologi dari penggunaan IDS pada scanning website target;



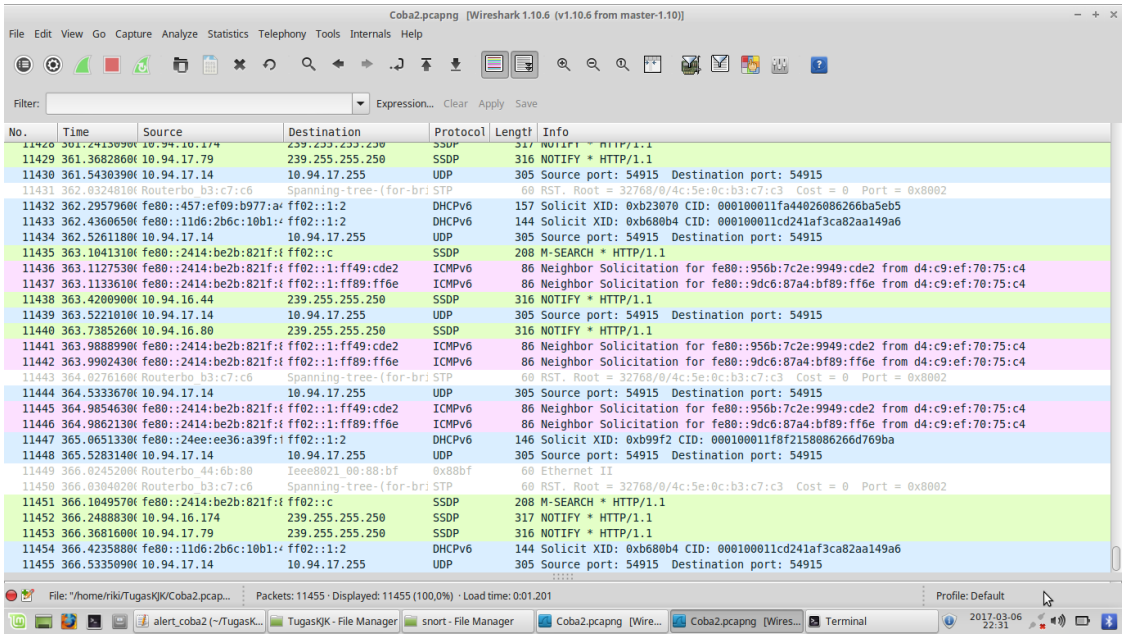
Gambar 1. Topologi Jaringan Scanning

Pada tugas ini pengambilan data dengan menggunakan aplikasi wireshark dalam rentan waktu selama 366 detik atau sama dengan 6 menit, dengan menggunakan tools NMAP yang digunakan untuk melakukan scaning pada target, dengan alamat domain target www.polsri.ac.id (202.9.69.34) yang dicapture dengan menggunakan aplikasi wireshark yang manghasilkan capture sebanyak 11455 paket. Berikut hasil capture yang telah dilakukan;



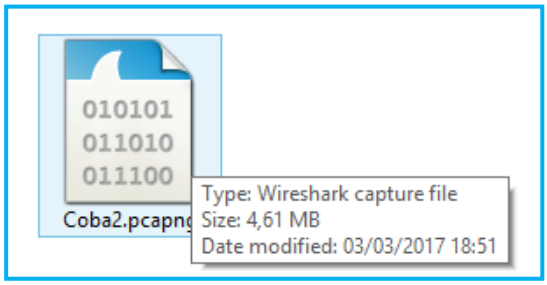
Gambar 2.a. Hasil Capture 1

Gambar diatas menunjukkan hasil capture paket-paket yang melakukan komunikasi antara server dengan client, yang dimulai dalam satu detik dengan menghasilkan file capture sebanyak 63 paket yang tercapture.



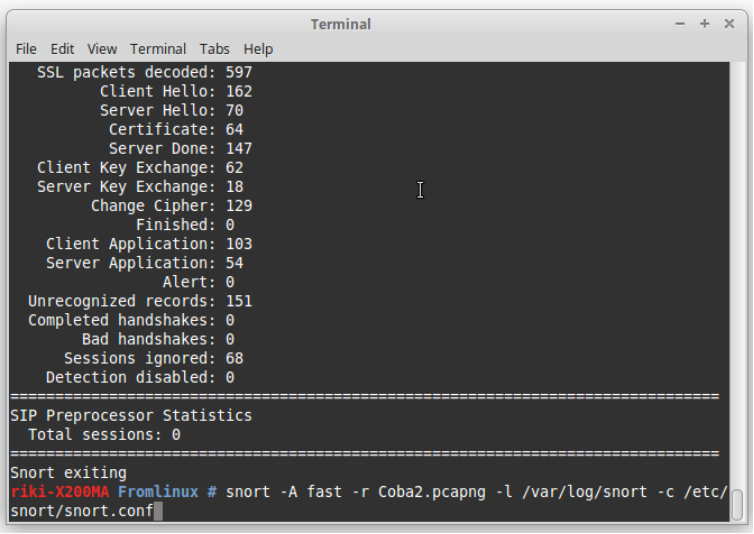
Gambar 2.a Hasil Capture 2

Banyaknya paket yang tercapture dengan menggunakan aplikasi wireshark, dapat dilihat pada gambar 2.a, dalam waktu 6 menit mengcapture paket yang yang saling berkomunikasi sebanyak 11455 paket. File pcap yang diperoleh di compile dengan menggunakan tools snort yang digunakan untuk mendeteksi serangan pada suatu network, snort dapat mendeteksi serangan sebesar 100Mbps. Output yang dihasilkan berupa report dan alert. Ada banyak variasi output yang dihasilkan snort, seperti teks (ASCII), XML, syslog, tcpdump, binary format, atau Database (MySQL, MsSQL, PostgreSQL, dsb). Berikut screenshot hasil pcap yang diperoleh dengang besar 4,61 MB.



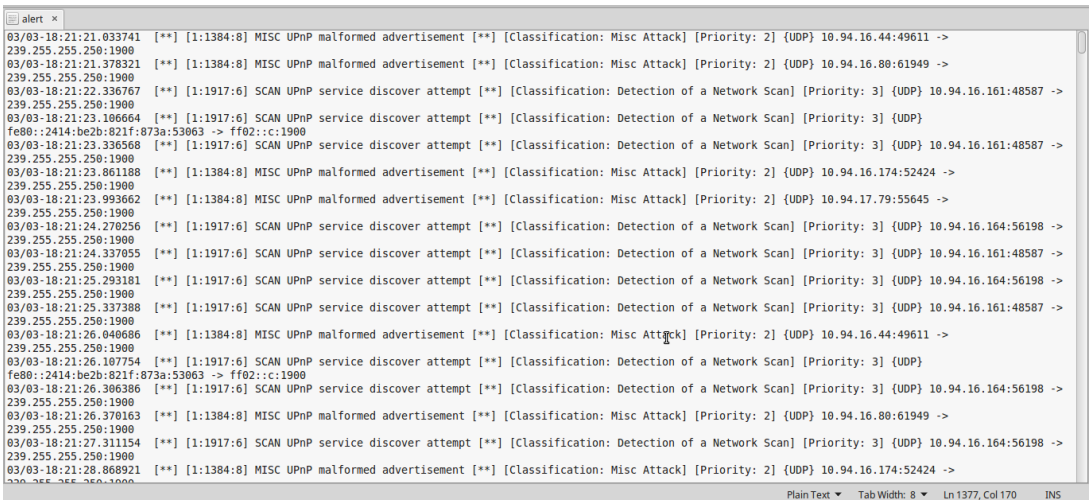
Gambar 3. Hasil capture file pcap

Kompile file pcap dengan menggunakan snort akan menghasilkan file alert yang berisi informasi-informasi paket yang terdapat dalam capture dari file pcap yang dilakukan dengan menggunakan wireshark, dengan perintah snort –A fast –r namafile.pcapng –l /var/log/snort/ -c /etc/snort/snort.conf. Perintah tersebut akan menghasilkan file alert yang akan tersimpan pada folder /var/log/snort. Berikut screenshot compile dari file pcap dengan menggunakan snort;



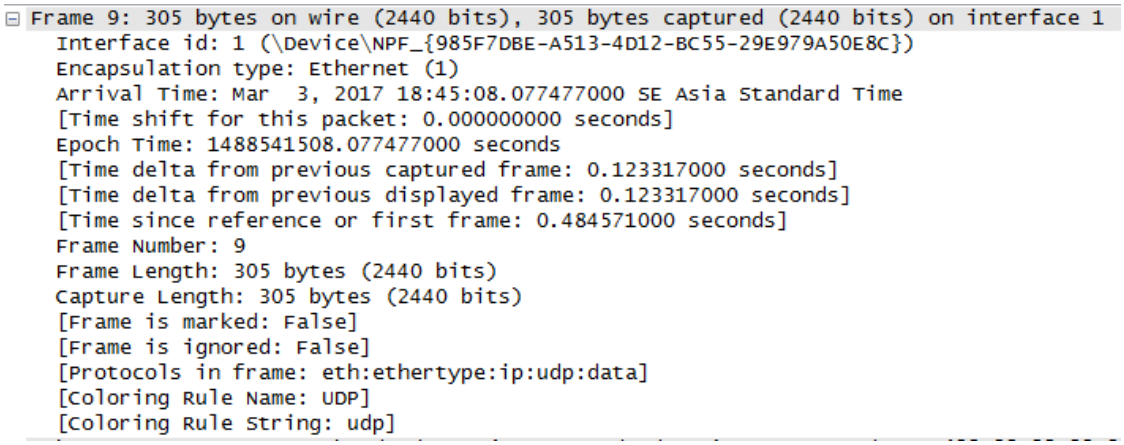
Gambar 4. Compile file pcap dengan menggunakan snort

Web yang dilakukan scanning dipantau selama kurang lebih 6 menit pengintaian, ada banyak sekali paket data yang mengalir. Terdapat beberapa paket data yang mengalir merupakan ancaman, sehingga IDS akan memberikan warning (alert). Beberapa alert yang muncul pada IDS Snort seperti yang ditunjukan pada Gambar 5. Baris pertama sampai baris terahir pada menunjukan adanya beberapa serangan yang diluncurkan. Dengan hasil alert yang diperoleh dari file pcap yang telah dicompile dengan menggunakan snort menghasilkan sebanyak 1386 snort yang dihasilkan, dengan klasifikasi yang berbeda-beda sebanyak 20. Berikut hasil alert yang diperoleh;



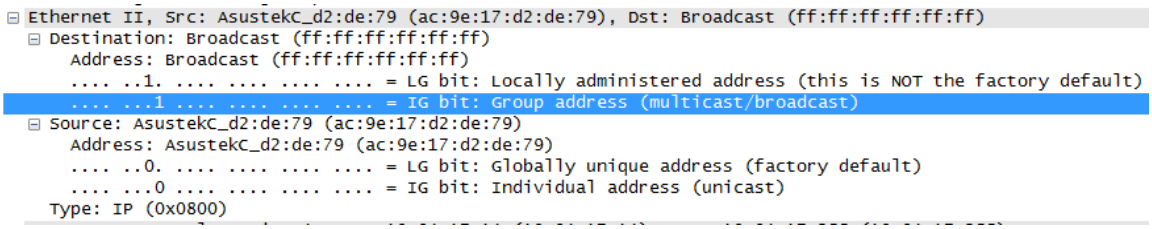
Gambar 5. Hasil snort berupa alert

Gambar 6 menunjukkan salah satu frame paket data yang mengalir pada jaringan. Paket data ini di-capture pada tanggal 03 maret 2017, dengan panjang frame 305 bytes (bits). Protokol jaringan yang digunakan adalah UDP.



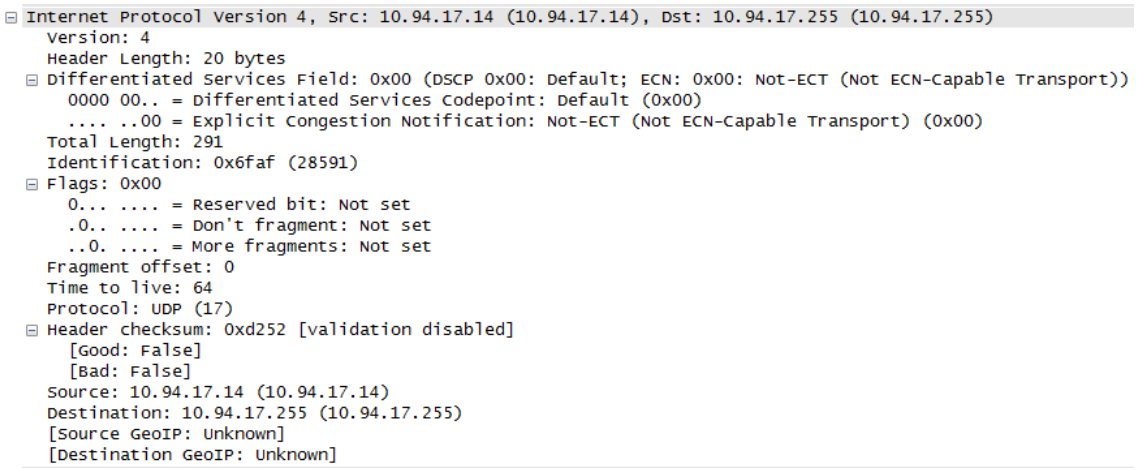
Gambar 6. Frame paket data

Pada range Ethernet berikutnya merupakan lapisan fisik dan data link pada local area network, pada gambar 7 menunjukkan ethernet yang digunakan, baik dari sumber dan tujuan paket data di jaringan yang telah tercapture, berikut hasil screen paket ethernetnya.



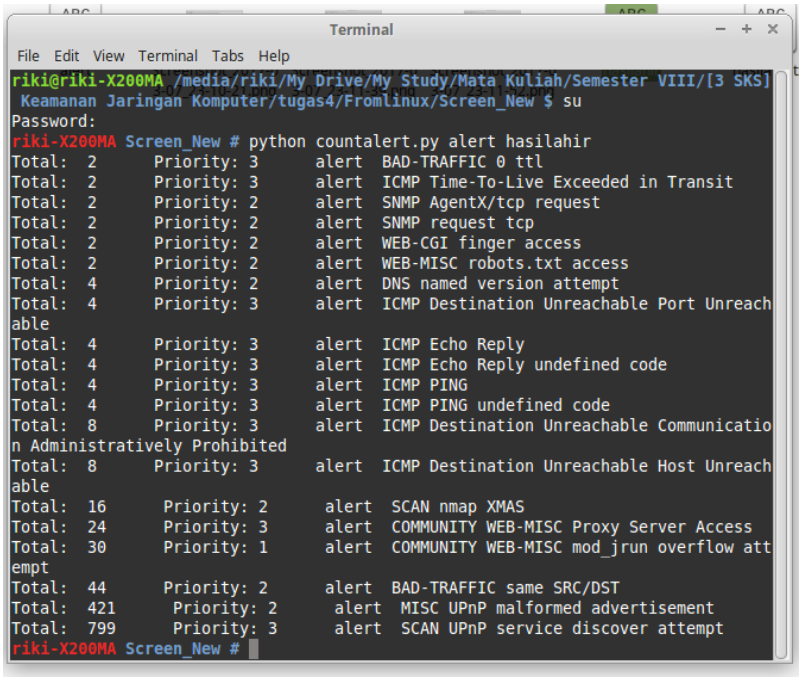
Gambar 7. Ethernet paket data

Paket data yang mengalir pada jaringan menggunakan IP Versi 4, dengan alamat sumber adalah 192.168.1.4 dengan tujuan 223.165.7.209. Informasi detailnya seperti yang ditunjukkan pada gambar 8.



Gambar 8. IP paket data pada jaringan

Hasil dari alert yang diperoleh memperlihatkan banyaknya dan jenis-jenis dari alert yang ada didalam file pcap yang telah dicaputre, hasil dari alert selanjutnya akan dikelompokkan berdasarkan nama alert yang ada, yang diurutkan dengan menggunakan program pengelompokan alert dengan perintah python countalert.py alert hasilahir.txt dimana kata pertama untuk mengcompile program python, kata kedua memangi program yang akan dieksekusi, kata ketiga nama alert yang akan dihitung dan kata terahir nama file serta format hasil dari perhitungan alert tersebut (*Program countalert.py Eko Arip Winanto*). Berikut hasil screenshot dari pengurutan alert berdasarkan nama dan jumlahnya pada gambar 9.



Gambar 9. Hasil pengelompokkan alert

Dengan hasil file yang dihasilkan dari program tersebut berupa pengurutan berdasarkan jenis alert yang ada dengan tipe file txt, berikut hasilnya dapat dilihat pada gambar 10.

```
Total: 2        Priority: 3         alert  BAD-TRAFFIC 0 ttl
Total: 2        Priority: 3         alert  ICMP Time-To-Live Exceeded in Transit
Total: 2        Priority: 2         alert  SNMP AgentX/tcp request
Total: 2        Priority: 2         alert  SNMP request tcp
Total: 2        Priority: 2         alert  WEB-CGI finger access
Total: 2        Priority: 2         alert  WEB-MISC robots.txt access
Total: 4        Priority: 2         alert  DNS named version attempt
Total: 4        Priority: 3         alert  ICMP Destination Unreachable Port Unreachable
Total: 4        Priority: 3         alert  ICMP Echo Reply
Total: 4        Priority: 3         alert  ICMP Echo Reply undefined code
Total: 4        Priority: 3         alert  ICMP PING
Total: 4        Priority: 3         alert  ICMP PING undefined code
Total: 8        Priority: 3         alert  ICMP Destination Unreachable Communication Administratively Prohibited
Total: 8        Priority: 3         alert  ICMP Destination Unreachable Host Unreachable
Total: 16       Priority: 2         alert  SCAN nmap XMAS
Total: 24       Priority: 3         alert  COMMUNITY WEB-MISC Proxy Server Access
Total: 30       Priority: 1         alert  COMMUNITY WEB-MISC mod_jrun overflow attempt
Total: 44       Priority: 2         alert  BAD-TRAFFIC same SRC/DST
Total: 421      Priority: 2         alert  MISC UPnP malformed advertisement
Total: 799      Priority: 3         alert  SCAN UPnP service discover attempt
```

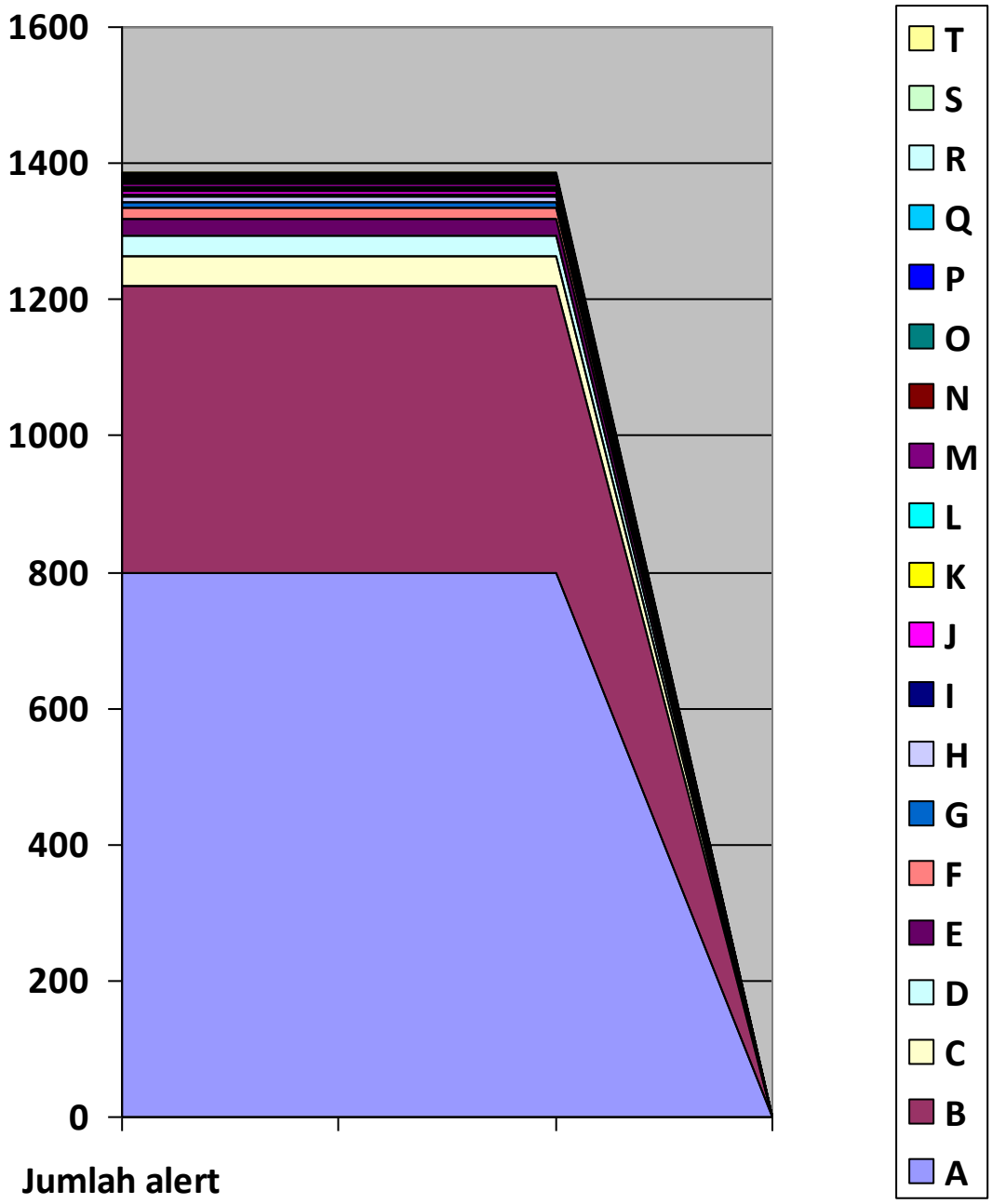
Gambar 10. Hasil dari program

Dari hasil alert yang diperoleh akan dibuat sebuah tabel beserta grafik yang menunjukkan hasil dari alert yag diperoleh, berikut hasil yang telah diperoleh dari scanning yang di capture dengan menggunakan aplikasi wireshark yang disajikan dalam bentuk tabel beserta grafik.

Tabel 1. Hasil pengurutan jumlah alert

No	Nama alert	Jumlah alert
1	BAD-TRAFFIC 0 ttl	2
2	ICMP Time-To-Live Exceeded in Transit	2
3	SNMP AgentX/tcp request	2
4	SNMP request tcp	2
5	WEB-CGI finger access	2
6	WEB-MISC robots.txt access	2
7	DNS named version attempt	4
8	ICMP Destination Unreachable Port Unreachable	4
9	ICMP Echo Reply	4
10	ICMP Echo Reply undefined code	4
11	ICMP PING	4
12	ICMP PING undefined code	4
13	ICMP Destination Unreachable Communication Administratively Prohibited	8
14	ICMP Destination Unreachable Host Unreachable	8
15	SCAN nmap XMAS	16
16	COMMUNITY WEB-MISC Proxy Server Access	24
17	COMMUNITY WEB-MISC mod_jrun overflow attempt	30
18	BAD-TRAFFIC same SRC/DST	44
19	MISC UPnP malformed advertisement	421
20	SCAN UPnP service discover attempt	799
J u m l a h a l e r t		**1387**

Tabel 1 diatas menunjukkan banyaknya alert yang diperoleh, beserta banyaknya jumlah alert dalam alert yang tercapture. Dari tabel diatas akan divisualisasikan dalam bentuk sebuah diagram yang menunnjukkan banyaknya alert yang tercapture.



Grafik 1. Jumlah alert

Dengan keterangan grafik yang dibuat sebagai berikut;

A : SCAN UPnP service discover attempt

B : MISC UPnP malformed advertisement

C : BAD-TRAFFIC same SRC/DST

D : COMMUNITY WEB-MISC mod_jrun overflow attempt

E : COMMUNITY WEB-MISC Proxy Server Access

F : SCAN nmap XMAS

G : ICMP Destination Unreachable Host Unreachable

H : ICMP Destination Unreachable Communication Administratively Prohibited

I : ICMP PING undefined code

J : ICMP PING

K : ICMP Echo Reply undefined code

L : ICMP Echo Reply

M : ICMP Destination Unreachable Port Unreachable

N : DNS named version attempt

O : WEB-MISC robots.txt access

P : WEB-CGI finger access

Q : SNMP request tcp

R : SNMP AgentX/tcp request

S : ICMP Time-To-Live Exceeded in Transit

T : BAD-TRAFFIC 0 ttl

Pada percobaan scanning ini terdapat paket SYN Flood yang merupakan Denial of Service yang memanfaatkan 'loophole' pada saat koneksi TCP/IP terbentuk. Kernel Linux terbaru (2.0.30 dan yang lebih baru) telah mempunya opsi konfigurasi untuk mencegah Denial of Service dengan mencegah atau menolak cracker mengakses sistem. Pada kondisi normal, client akan mengirimkan paket data yang berupa SYN untuk mensinkronkan diri kepada server. Lalu server mereima request dari client dan akan memberikan jawaban ke client berupa ACK (Acknowledgement) sebagai tanda transakasi sudah dimulai (pengiriman dan penerimaan data), maka client akan mengirimkan kembali sebuah paket SYN lagi.

Serangan ini memenuhi server dengan banyak paket SYN, karena saat pengiriman paket SYN oleh client, maka server juga akan mengirim paket SYN ACK ke client. Paket TCP SYN ACK yang masuk diantrian backlog hanya akan dibuang dari backlog pada saat terjadi time out dari timer TCP yang menandakan bahwa tidak ada respon dari pengirim, paket SYN ini tedak terdeteksi oleh aplikasi snort, sehingga akan

menjadi berbahaya bagi target. Berikut paket SYN yang tercapture oleh wireshark.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	10.94.16.164	10.100.130.10	DNS	84	Standard query 0xed2b PTR 34.69.9.202.in-addr.arpa
2	0.00790800	10.94.16.72	224.0.0.251	MDNS	82	Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
3	0.00863700	fe80::2414:be2b:821	ff02::fb	MDNS	102	Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
4	0.00942400	10.94.16.72	224.0.0.251	MDNS	82	Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
5	0.00952800	fe80::2414:be2b:821	ff02::fb	MDNS	102	Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
6	0.35232300	10.94.16.172	239.255.255.250	SSDP	215	M-SEARCH * HTTP/1.1
7	0.36050200	10.94.16.172	224.0.0.251	MDNS	82	Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
8	0.36125400	fe80::f975:500:cc7e	ff02::fb	MDNS	102	Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
9	0.48457100	10.94.17.14	10.94.17.255	UDP	305	Source port: 54915 Destination port: 54915
10	0.69166500	10.100.130.10	10.94.16.164	DNS	164	Standard query response 0xed2b PTR www.polsri.ac.id
11	0.70359100	10.94.16.164	202.9.69.34	TCP	58	54333→445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
12	0.70636900	10.94.16.164	202.9.69.34	TCP	58	54333→111 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
13	0.70647400	10.94.16.164	202.9.69.34	TCP	58	54333→1723 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
14	0.70664800	10.94.16.164	202.9.69.34	TCP	58	54333→443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
15	0.70674900	10.94.16.164	202.9.69.34	TCP	58	54333→135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
16	0.70678600	10.94.16.164	202.9.69.34	TCP	58	54333→110 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
17	0.70690600	10.94.16.164	202.9.69.34	TCP	58	54333→554 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
18	0.70699100	10.94.16.164	202.9.69.34	TCP	58	54333→22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
19	0.70754200	10.94.16.164	202.9.69.34	TCP	58	54333→53 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
20	0.70767300	10.94.16.164	202.9.69.34	TCP	58	54333→25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
21	0.70797600	202.9.69.34	10.94.16.164	TCP	60	53→54333 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
22	0.71980200	10.94.16.164	202.9.69.34	TCP	58	54333→3389 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
23	0.71990300	10.94.16.164	202.9.69.34	TCP	58	54333→21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
24	0.74001300	202.9.69.34	10.94.16.164	TCP	60	110→54333 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
25	0.74090900	202.9.69.34	10.94.16.164	TCP	60	111→54333 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
26	0.74166700	202.9.69.34	10.94.16.164	TCP	60	1723→54333 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
27	0.74246800	202.9.69.34	10.94.16.164	TCP	60	443→54333 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Gambar 11. Paket syn