Network Security: SNORT

Intrusion detection adalah proses mendeteksi penggunaan yang tidak sah, atau serangan terhadap suatu jaringan komputer. *Intrusion Detection Systems* (IDS) dirancang dan digunakan untuk membantu dalam menghalangi atau mengurangi ancaman, kerusakan yang dapat ditimbulkan dari aktivitas *hacking* [1]. IDS merupakan kombinasi perangkat lunak atau perangkat keras yang dapat melakukan deteksi penyusupan pada sebuah jaringan. IDS mendeteksi adanya aktifitas mencurigakan berdasarkan *signature based* dan *anomali based* yang merupakan metode analisis *event* IDS. *Signature based* dapat diumpamakan seperti virus yang memiliki ciri khas. *Signature based* menggunakan pendekatan dengan cara pencocokan kejadian (*event*) dengan jenis serangan yang telah dikenal pada *database* IDS. Teknik ini sangat efektif dan merupakan metode utama yang digunakan pada beberapa perangkat atau produk IDS untuk mendeteksi serangan, sedangan *anomali based* berdasakan kejanggalan yang terjadi pada pola lalulintas jaringan yang diawasi, *Anomaly based* menggunakan pendekatan dengan cara mengidentifikasi perilaku atau aktivitas yang tidak biasa yang terjadi pada suatu *host* atau jaringan. *Anomaly based* membentuk perilaku dasar pada sebuah kondisi jaringan "normal" dengan profil pengguna tertentu kemudian mengukur dan membandingkannya ketika aktivitas jaringan berjalan tidak "normal". [1, 2].

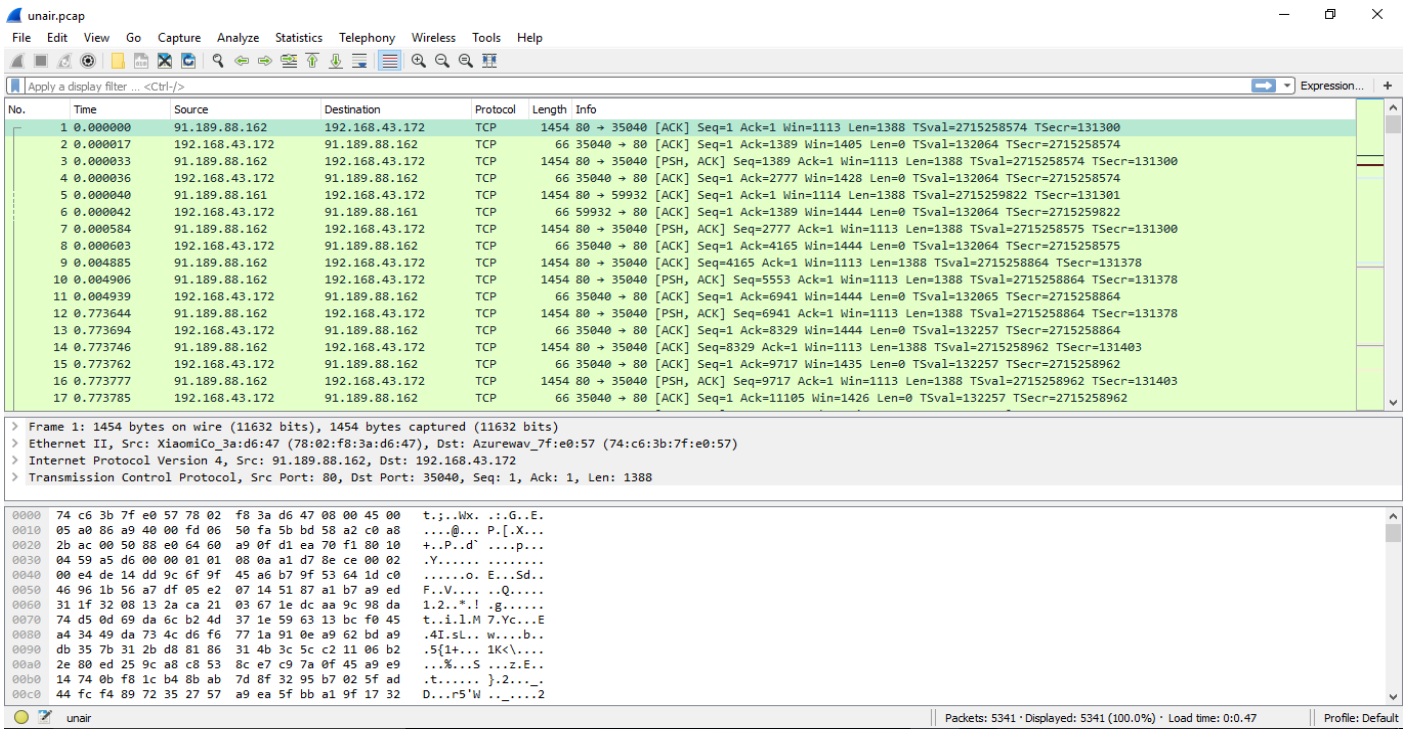
Berikut adalah *tools* yang digunakan dalam mengerjakan tugas *snort network security*:

- Wireshark: digunakan untuk *capture paket*
- Nmap: digunakan untuk melakukan *scanning* target
- Snort: digunakan untuk deteksi IDS
- Program counteralert.py: digunakan untuk mendapatkan total dan *priority* dari hasil log alert snort.

➢ Wireshark dan Nmap

Wireshark merupakan sebuah *Network Packet Analyzer*. *Network Packet Analyzer* akan mencoba menangkap paket-paket jaringan dan berusaha untuk menampilkan semua informasi di paket tersebut sedatail mungkin. Sedangkan Nmap merupakan tools yang digunakan untuk melakukan *scanning* pada target. Target dalam percobaan ini merupakan website Universitas Airlangga www.unair.ac.id.

Pada saat melakukan scanning, kita harus menjalankan Wireshark secara bersamaan untuk mengcapture paket. Paket dari hasil *scanning* dari hasil wireshark berjumlah 5341.



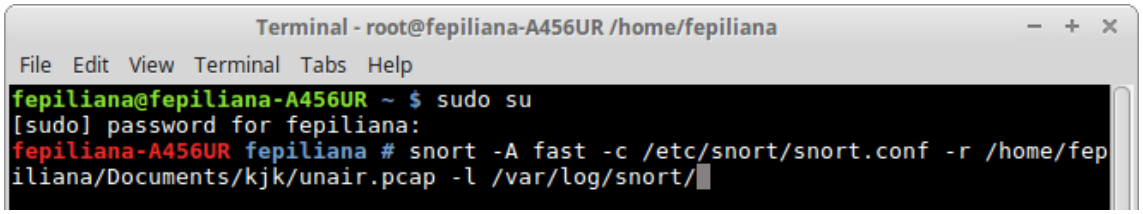
Gambar 1: hasil capture Wireshark *scanning* target

➢ Snort dan Program counteralert.py

Pada pengerjaan tugas *network security*, penulis menggunakan perangkat lunak Snort. Snort adalah salah satu tool atau aplikasi *open source Intrusion Detection Systems* (IDS) terbaik yang tersedia dan dikembangkan hingga saat ini. Pada pengerjaan tugas mata kuliah *network security* Snort dirancang untuk beroperasi berbasis *command line* dengan menggunakan metode *signature based*. Snort menganalisis semua lalu lintas jaringan untuk mengendus (*sniff*) dan mencari beberapa jenis penyusupan dalam

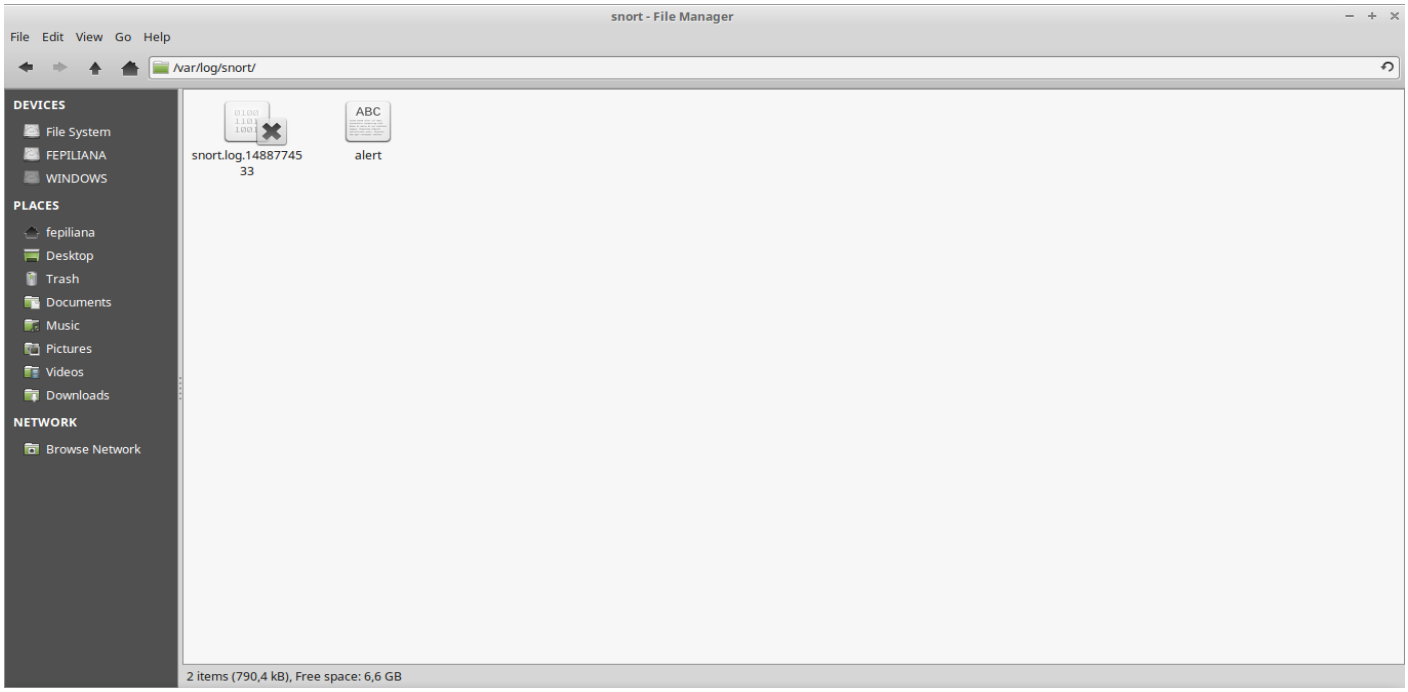
sebuah jaringan [2]. Snort tersusun atas *rules* yang disimpan dalam *signature database* [3]. Output dari Snort berupa *log alert* yang terletak pada direktori **/var/log/snort** di sistem operasi linux.

Untuk menjalankan snort pada sistem operasi Linux, kita gunakan *command* snort –A fast –c /etc/snort/snort.conf –r (masukan_path_file_pcap) –l /var/log/snort**.**



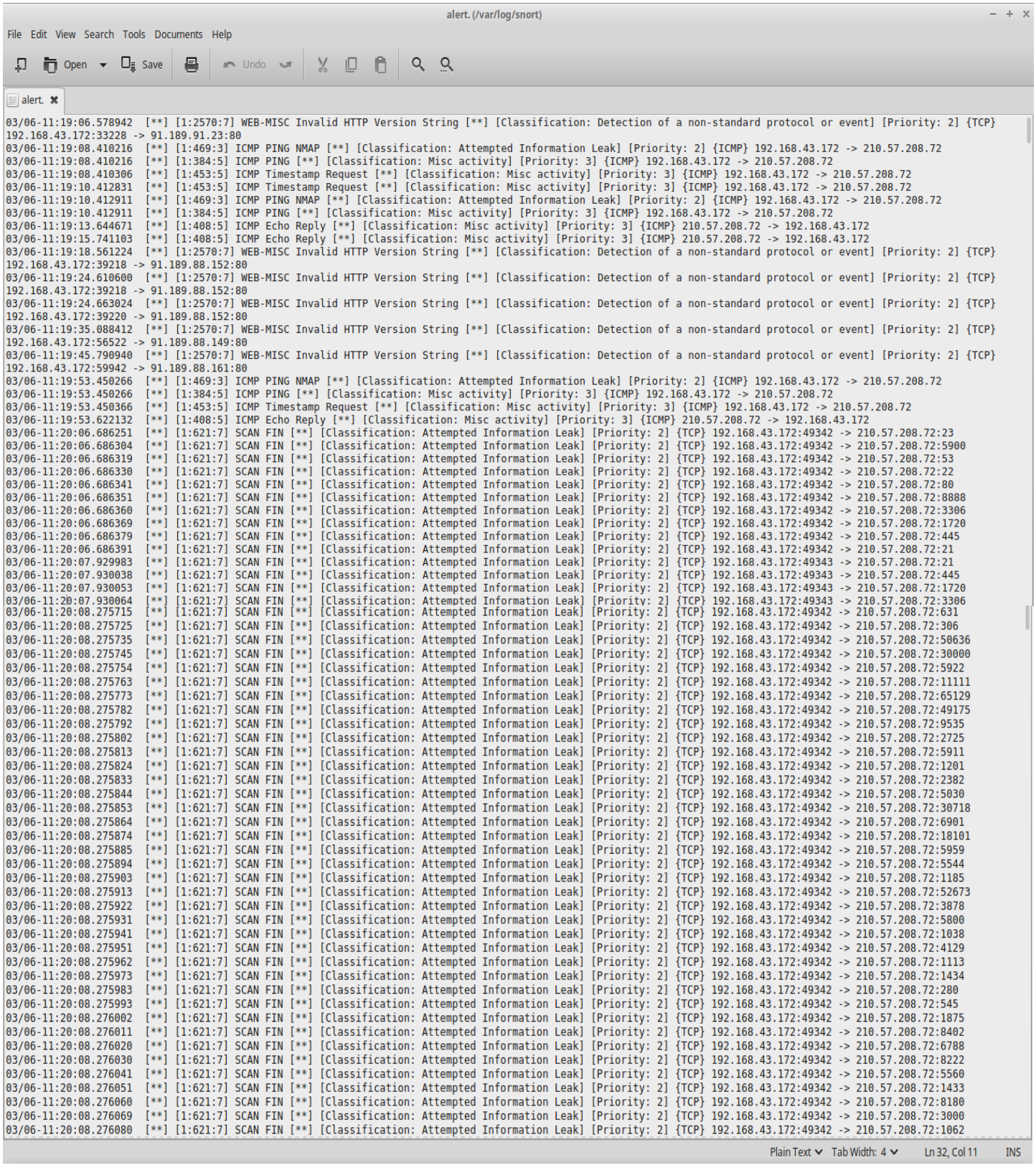
Gambar 2: *command* menjalankan snort.

Snort menggunakan rules yang disimpan dalam file teks yang dapat dimodifikasi dengan editor teks. Rules terbagi-bagi dalam beberapa kategori yang tersimpan dalam file-file yang berbeda-beda. File-file tersebut merupakan bagian dari file konfigurasi Snort, yang secara *default* disimpan dengan nama file konfigurasi snort.conf. Rule snort digunakan untuk mendeteksi adanya serangan atau aktivitas yang dilakukan oleh client-client terhadap server. Untuk melihat hasil alert yang dihasilkan oleh snort, kita dapat lihat pada direktori /var/log/snort.



Gambar 3: hasil alert dari snort.

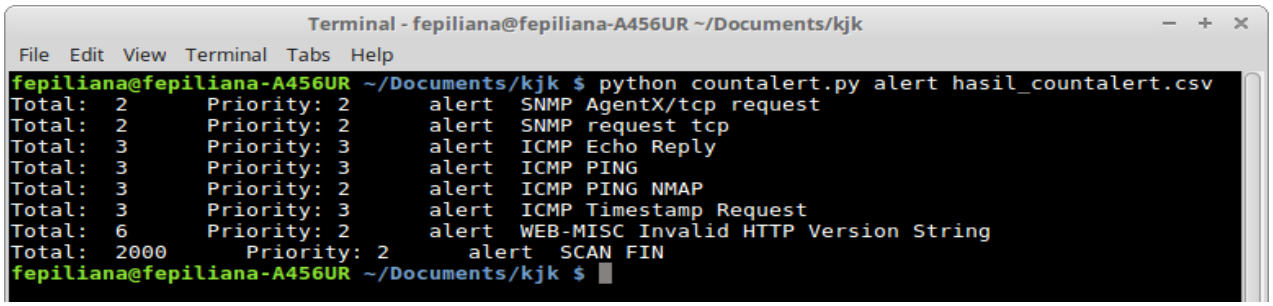
Pada gambar 3, dapat kita lihat terdapat sebuah file yang bernama alert pada direktori /var/log/snort. Berikut ada beberapa isi dari file alert:



Gambar 4: isi file alert dari snort

Berdasarkan gambar 4, ada banyak jenis alert yang dapat kita lihat dari hasil snort. Untuk mendapatkan total dari setiap alert yang dihasilkan, jika dihitung setiap baris akan sangat melelahkan dan tidak menutup kemungkinan akan adanya kekeliruan. Untuk hasil yang lebih efisien, maka penulis menggunakan program counteralert.py untuk mendapatkan total dan *priority* dari setiap jenis alert. Program counteralert.py merupakan program yang berbasis bahasa Python. Untuk menjalankan program tersebut kita ketik di *command* linux :

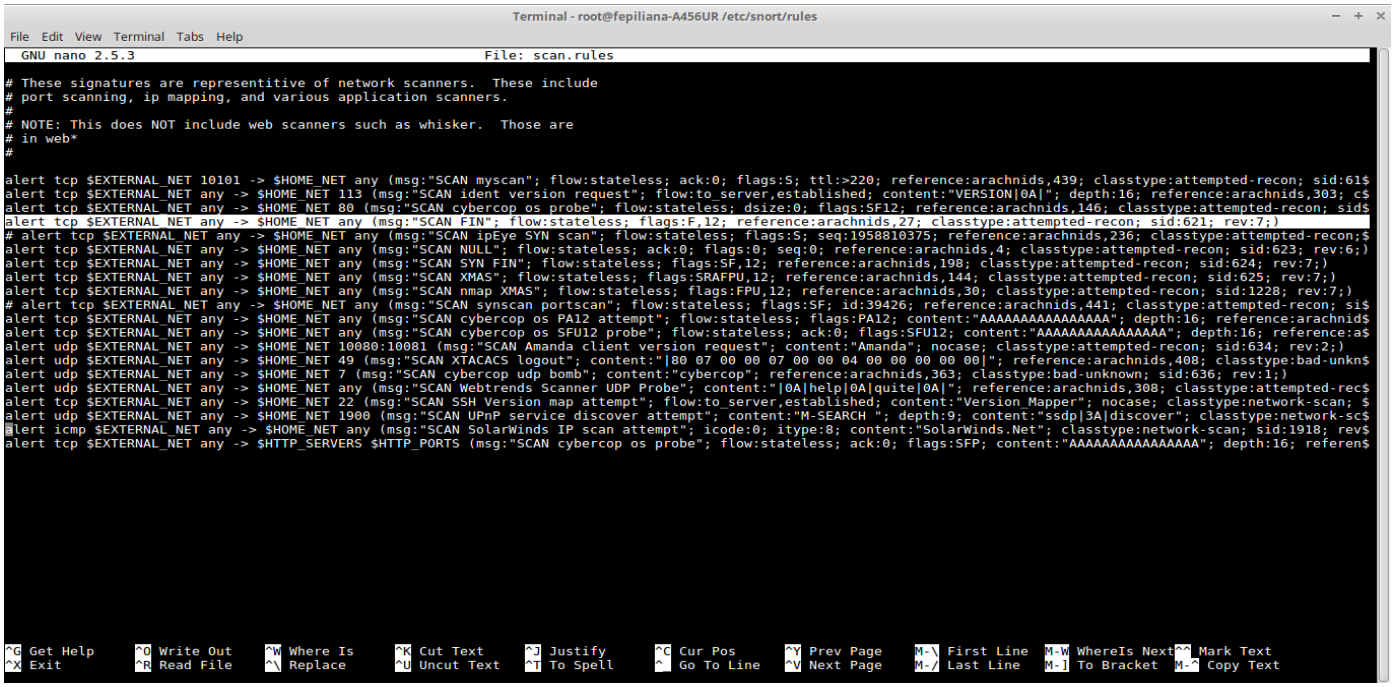
python counteralert.py alert (nama_file_yang_diinginkan).csv



Gambar 5: hasil *count* alert

Berdasarkan hasil dari gambar 5, dapat kita lihat bahwa alert SCAN FIN memiliki total 2000 yang terdapat pada file alert dari snort. SCAN FIN merupakan alert untuk mendeteksi *port scanning.*

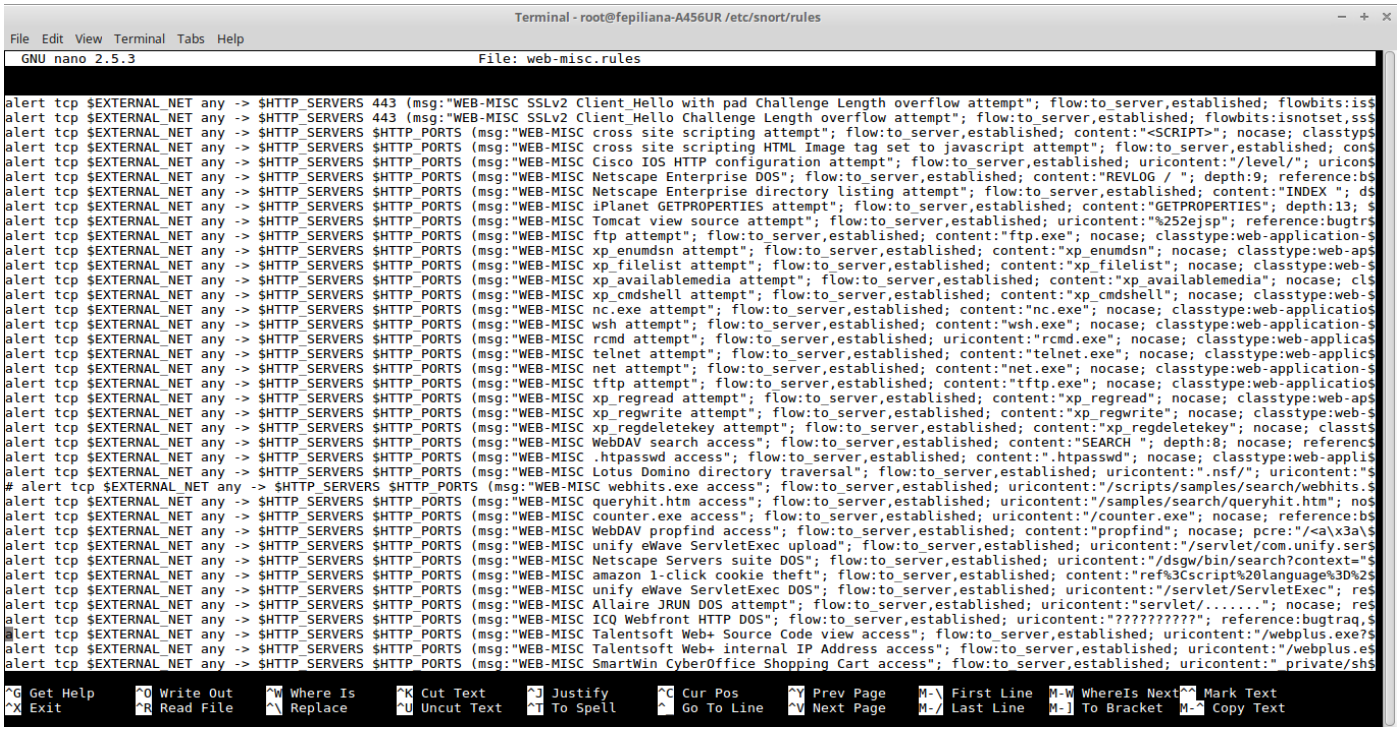
Berikut adalah beberapa rules alert snort yang dicocokan dengan gambar 3 dan gambar 4, sehingga dapatkanlah hasil alert pada gambar 5:



Gambar 6: rules alert SCAN FIN

Alert SCAN FIN

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN FIN"; flow:stateless; flags:F,12; reference:arachnids,27; classtype:attempted-recon; sid:621; rev:7;)



Gambar 7: rules alert WEB-MISC

Alert WEB-MISC

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 443 (msg:"WEB-MISC SSLv2 Client_Hello with pad Challenge Length overflow attempt"; flow:to_server,established; flowbits:is$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 443 (msg:"WEB-MISC SSLv2 Client_Hello Challenge Length overflow attempt"; flow:to_server,established; flowbits:isnotset,ss$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC cross site scripting attempt"; flow:to_server,established; content:"<SCRIPT>"; nocase; classtyp$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC cross site scripting HTML Image tag set to javascript attempt"; flow:to_server,established; con$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC Cisco IOS HTTP configuration attempt"; flow:to_server,established; uricontent:"/level/"; uricon$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC Netscape Enterprise DOS"; flow:to_server,established; content:"REVLOG / "; depth:9; reference:b$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC Netscape Enterprise directory listing attempt"; flow:to_server,established; content:"INDEX "; d$

Alert WEB-MISC

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC xp_availablemedia attempt"; flow:to_server,established; content:"xp_availablemedia"; nocase; cl$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC xp_cmdshell attempt"; flow:to_server,established; content:"xp_cmdshell"; nocase; classtype:web-$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC nc.exe attempt"; flow:to_server,established; content:"nc.exe"; nocase; classtype:web-applicatio$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC wsh attempt"; flow:to_server,established; content:"wsh.exe"; nocase; classtype:web-application-$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC rcmd attempt"; flow:to_server,established; uricontent:"rcmd.exe"; nocase; classtype:web-applica$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC telnet attempt"; flow:to_server,established; content:"telnet.exe"; nocase; classtype:web-applic$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC net attempt"; flow:to_server,established; content:"net.exe"; nocase; classtype:web-application-$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC tftp attempt"; flow:to_server,established; content:"tftp.exe"; nocase; classtype:web-applicatio$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC xp_regread attempt"; flow:to_server,established; content:"xp_regread"; nocase; classtype:web-ap$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC xp_regwrite attempt"; flow:to_server,established; content:"xp_regwrite"; nocase; classtype:web-$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC xp_regdeletekey attempt"; flow:to_server,established; content:"xp_regdeletekey"; nocase; classt$

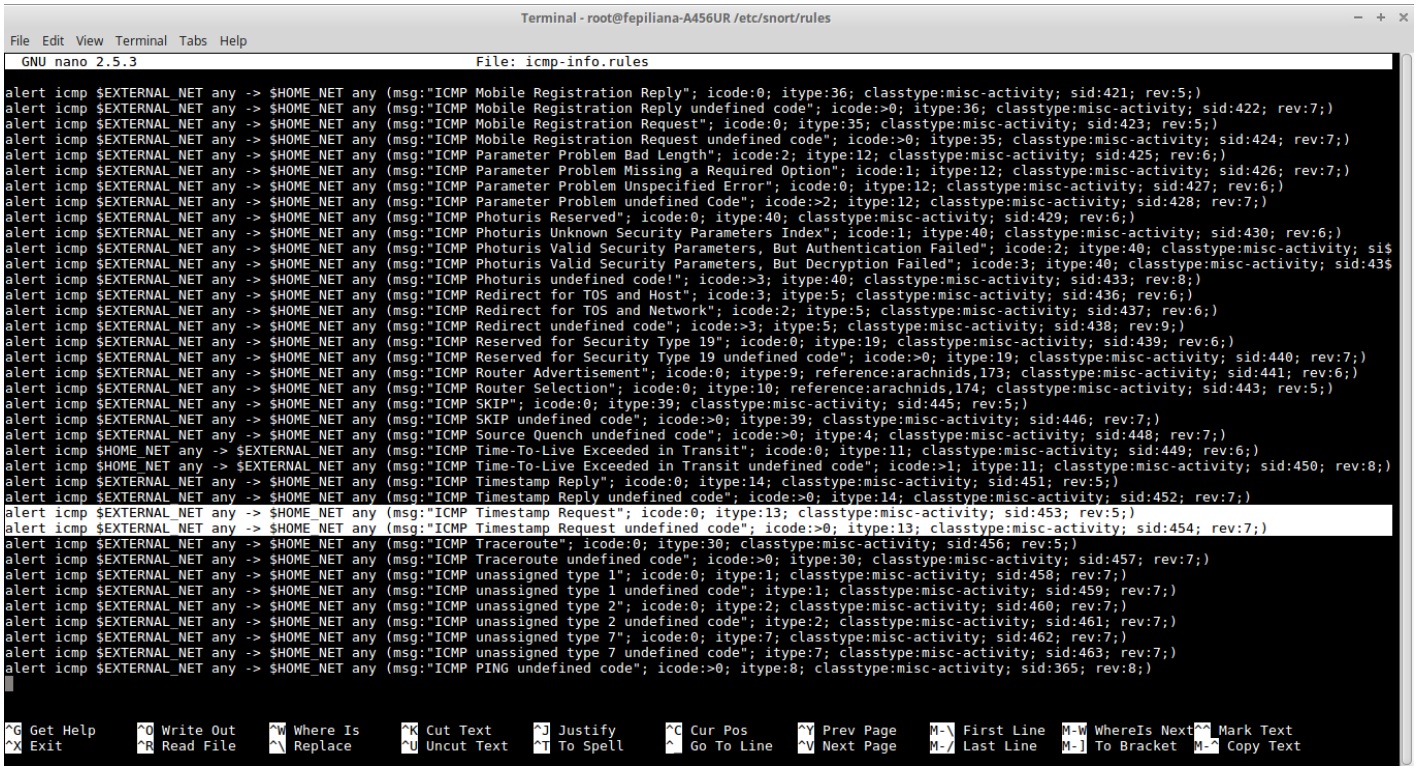
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC .htpasswd access"; flow:to_server,established; content:".htpasswd"; nocase; classtype:web-appli$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC Lotus Domino directory traversal"; flow:to_server,established; uricontent:".nsf/"; uricontent:"$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC webhits.exe access"; flow:to_server,established; uricontent:"/scripts/samples/search/webhits.$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC queryhit.htm access"; flow:to_server,established; uricontent:"/samples/search/queryhit.htm"; no$

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC counter.exe access"; flow:to_server,established; uricontent:"/counter.exe"; nocase; reference:b$

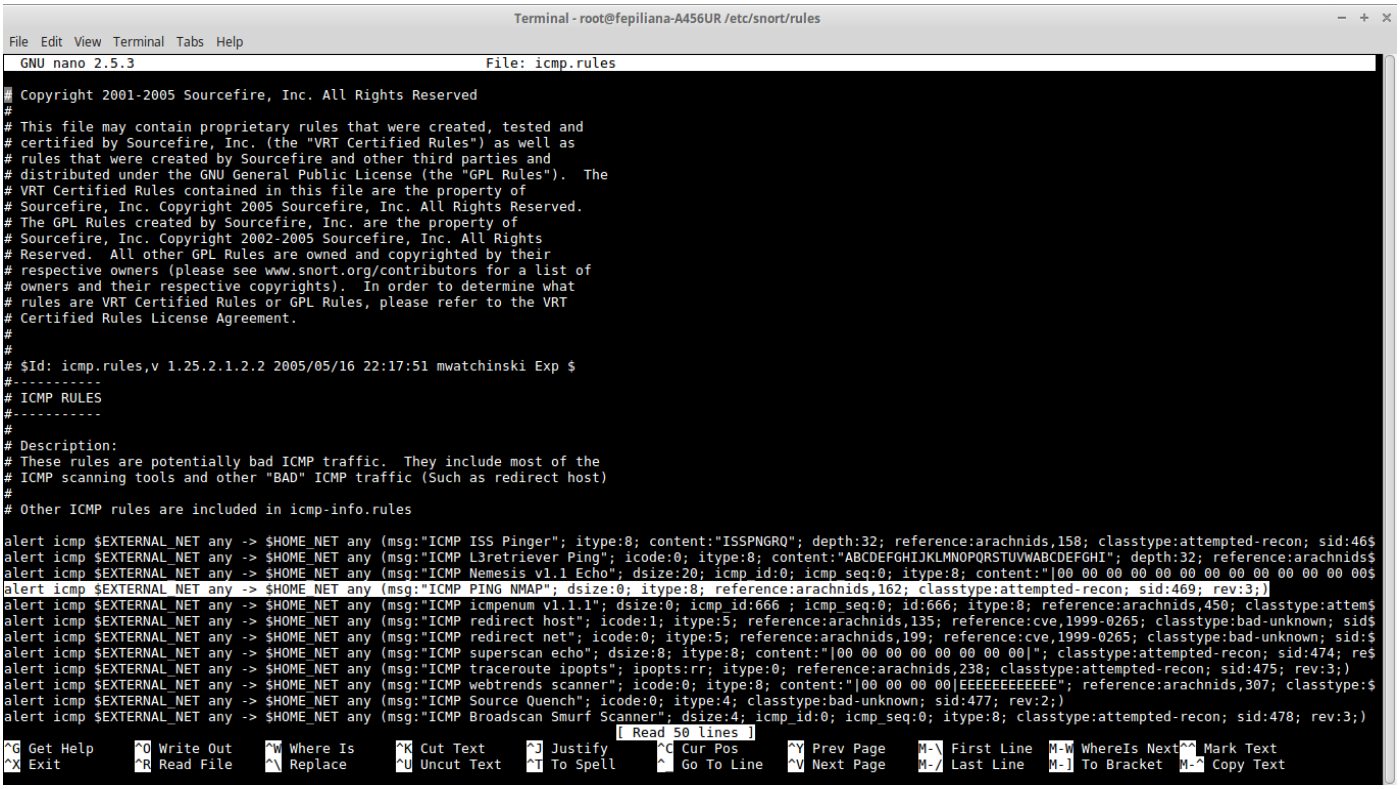


Gambar 8: rules alert ICMP Timestamp

Alert ICMP Timestamp

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP Timestamp Request"; icode:0; itype:13; classtype:misc-activity; sid:453; rev:5;)

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP Timestamp Request undefined code"; icode:>0; itype:13; classtype:misc-activity; sid:454; rev:7;)



Gambar 9: rules alert ICMP PING NMAP

Alert ICMP PING NMAP

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING NMAP"; dsize:0; itype:8; reference:arachnids,162; classtype:attempted-recon; sid:469; rev:3;)

Gambar 9: rules alert ICMP PING

Alert ICMP PING

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING *NIX"; itype:8; content:"|10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F|"; depth:32; classtype:misc-act$

Alert ICMP PING

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING BSDtype"; itype:8; content:"|08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17|"; depth:32; reference:arach$

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING BayRS Router"; itype:8; content:"|01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F|"; depth:32; reference:ara$

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING BeOS4.x"; itype:8; content:"|00 00 00 00 00 00 00 00 00 00 00 00 08 09 0A 0B|"; depth:32; reference:arach$

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING Cisco Type.x"; itype:8; content:"|AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD|"; depth:32; reference:$

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING Delphi-Piette Windows"; itype:8; content:"Pinging from Del"; depth:32; reference:arachnids,155; classtype$

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING Flowpoint2200 or Network Management Software"; itype:8; content:"|01 02 03 04 05 06 07 08 09 0A 0B 0C 0D $

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING IP NetMonitor Macintosh"; itype:8; content:"|A9| Sustainable So"; depth:32; reference:arachnids,157; clas$

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING LINUX/*BSD"; dsize:8; id:13170; itype:8; reference:arachnids,447; classtype:misc-activity; sid:375; rev:6$

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING Microsoft Windows"; itype:8; content:"0123456789abcdefghijklmnop"; depth:32; reference:arachnids,159; cla$

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING Network Toolbox 3 Windows"; itype:8; content:"================"; depth:32; reference:arachnids,161; class$

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING Ping-O-MeterWindows"; itype:8; content:"OMeterObeseArmad"; depth:32; reference:arachnids,164; classtype:m$

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING Pinger Windows"; itype:8; content:"Data|00 00 00 00 00 00 00 00 00 00 00 00|"; depth:32; reference:arachn$

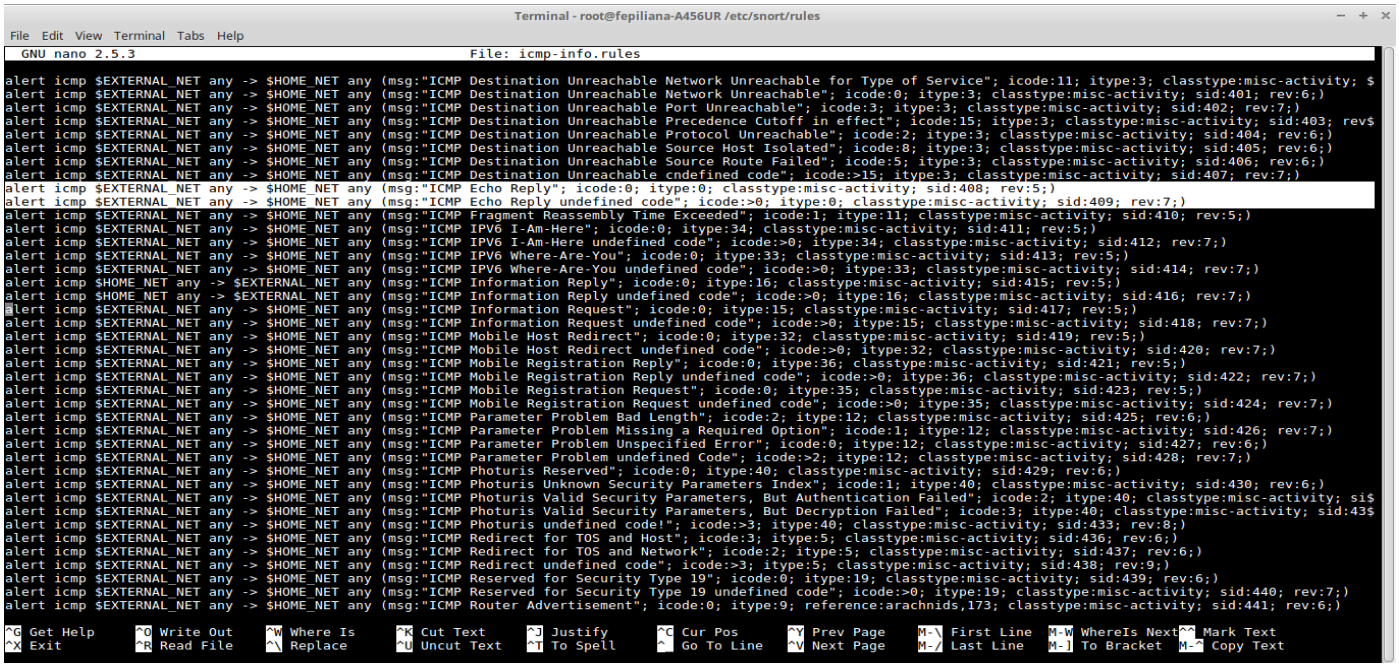
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING Seer Windows"; itype:8; content:"|88 04|"; depth:32; reference:arachnids,166; classtype:mis$

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING Sun Solaris"; dsize:8; itype:8; reference:arachnids,448; classtype:misc-activity; sid:381; rev:6;)

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING Windows"; itype:8; content:"abcdefghijklmnop"; depth:16; reference:arachnids,169; classtype:misc-activity$

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP traceroute"; itype:8; ttl:1; reference:arachnids,118; classtype:attempted-recon; sid:385; rev:4;)

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING"; icode:0; itype:8; classtype:misc-activity; sid:384; rev:5;)

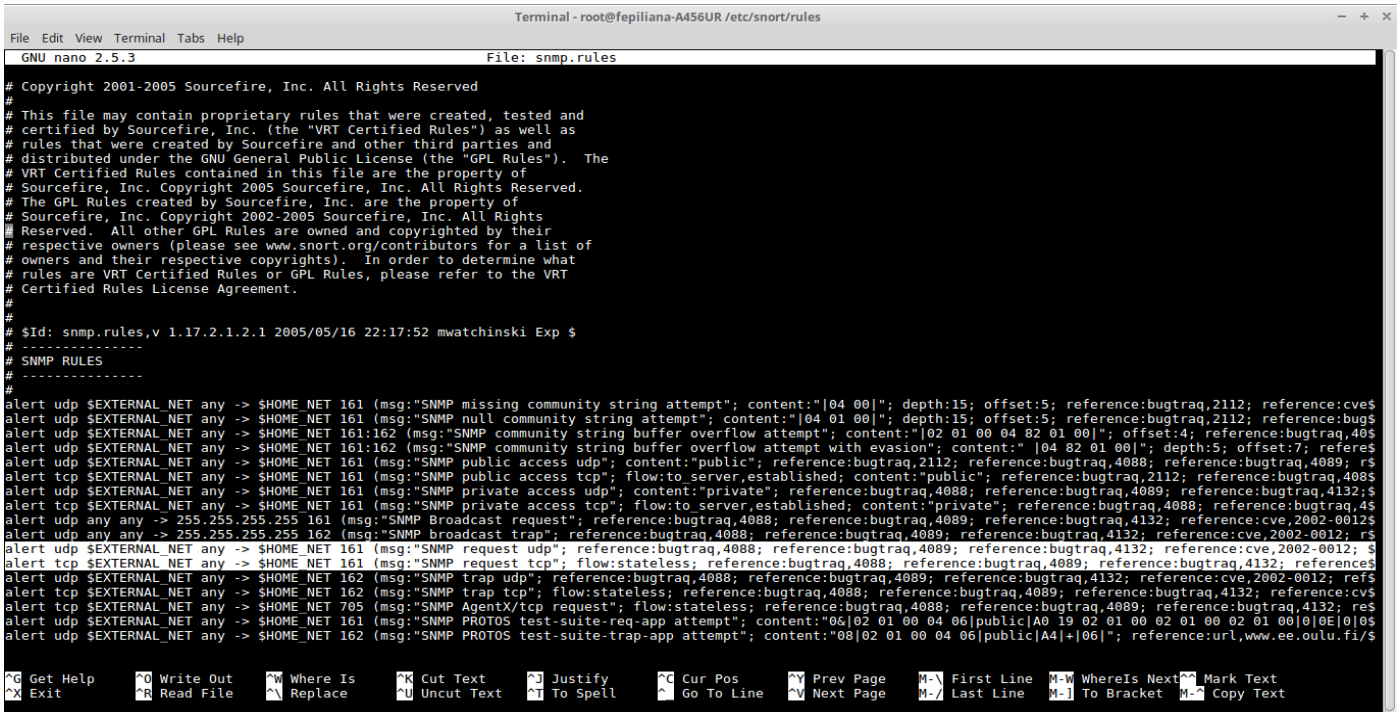


Gambar 10: rules alert ICMP Echo Reply

Alert ICMP Echo Reply

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP Echo Reply"; icode:0; itype:0; classtype:misc-activity; sid:408; rev:5;)

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP Echo Reply undefined code"; icode:>0; itype:0; classtype:misc-activity; sid:409; rev:7;)

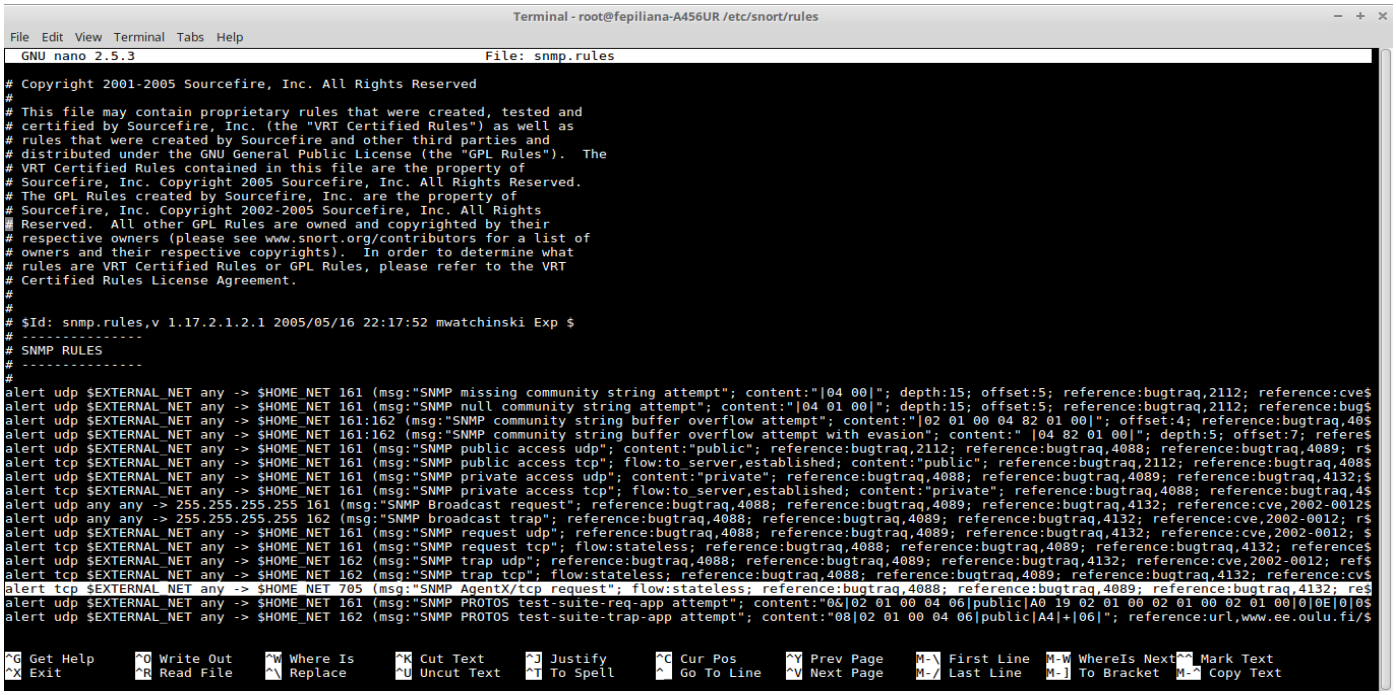


Gambar 11: rules alert SNMP Request

Alert SNMP Request

alert udp $EXTERNAL_NET any -> $HOME_NET 161 (msg:"SNMP request udp"; reference:bugtraq,4088; reference:bugtraq,4089; reference:bugtraq,4132; reference:cve,2002-0012; $

alert tcp $EXTERNAL_NET any -> $HOME_NET 161 (msg:"SNMP request tcp"; flow:stateless; reference:bugtraq,4088; reference:bugtraq,4089; reference:bugtraq,4132; reference$



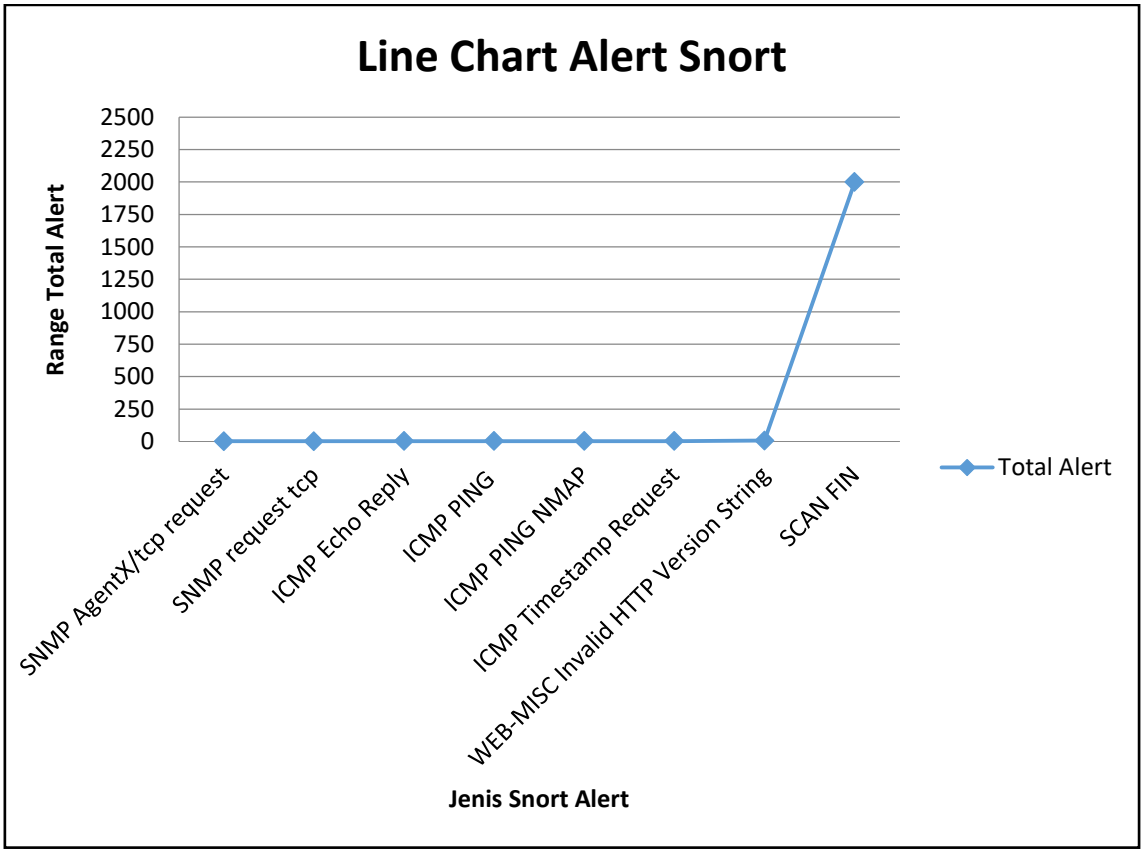
Gambar 11: rules alert SNMP AgentX/tcp

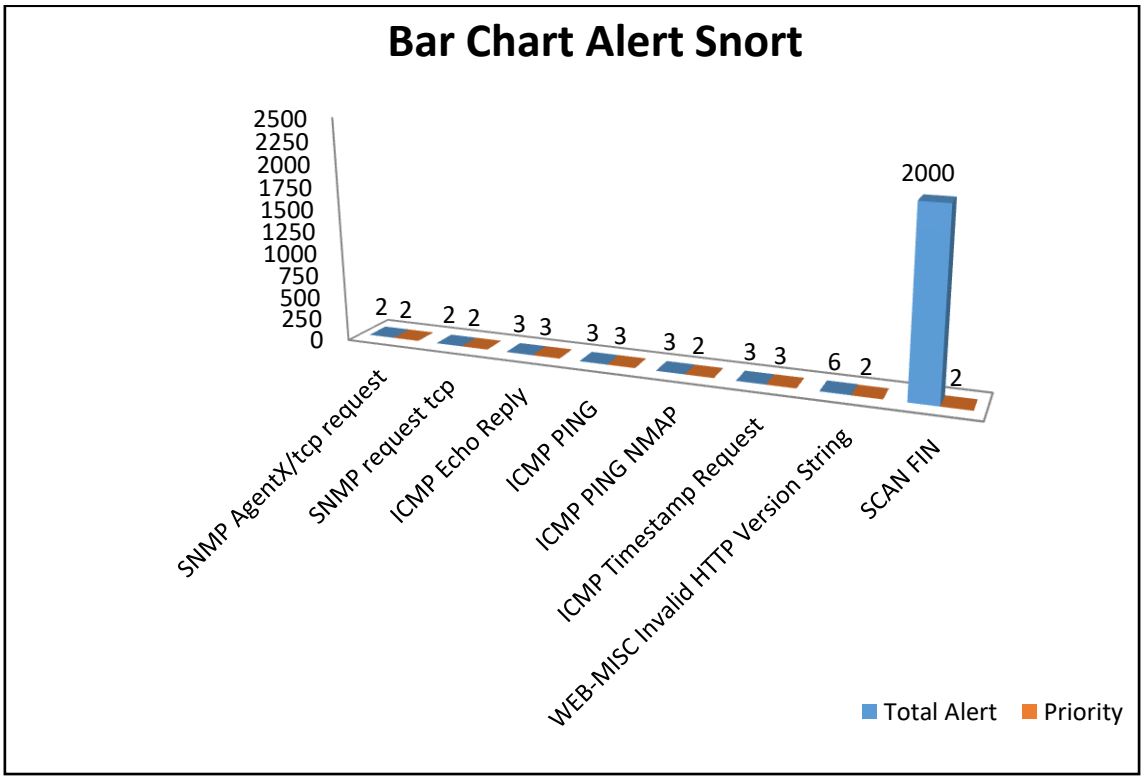
Alert SNMP AgentX/tcp

alert tcp $EXTERNAL_NET any -> $HOME_NET 705 (msg:"SNMP AgentX/tcp request"; flow:stateless; reference:bugtraq,4088; reference:bugtraq,4089; reference:bugtraq,4132; re$

- **Tabel Alert Snort dan Diagramnya**

No	Jenis Alert	Total	Prioritas
1.	SNMP AgentX/tcp request	2	2
2.	SNMP request tcp	2	2
3.	ICMP Echo Reply	3	3
4.	ICMP PING	3	3
5.	ICMP PING NMAP	3	2
6.	ICMP Timestamp Request	3	3
7.	WEB-MISC Invalid HTTP Version String	6	2
8.	SCAN FIN	2000	2




Bar Chart Alert Snort
2500
2250
2000
1750
1500
1250
1000
750
500
250
0
2 2
2 2
3 3
3 3
3 2
3 3
6 2
2000
2
SNMP AgentX/tcp request
SNMP request tcp
ICMP Echo Reply
ICMP PING
ICMP PING NMAP
ICMP Timestamp Request
WEB-MISC Invalid HTTP Version String
SCAN FIN
Total Alert
Priority

Daftar Pustaka

[1] M. Affandi and S. Setyowibowo, “Implementasi Snort Sebagai Alat Pendeteksi Intrusi,” vol. 4, no. 2.

[2] M. Jannah, Hustinawati, and R. Wildani, “Implementasi Intrusion System (Ids) Snort Pada Laboratorium Jaringan Komputer,” *UG J.*, vol. 6 No 5, pp. 1–4, 2012.

[3] M. Ridwan Zalbina, “Sistem Deteksi HTTP Inspect Preprocessor dan Rule Options”, 2016