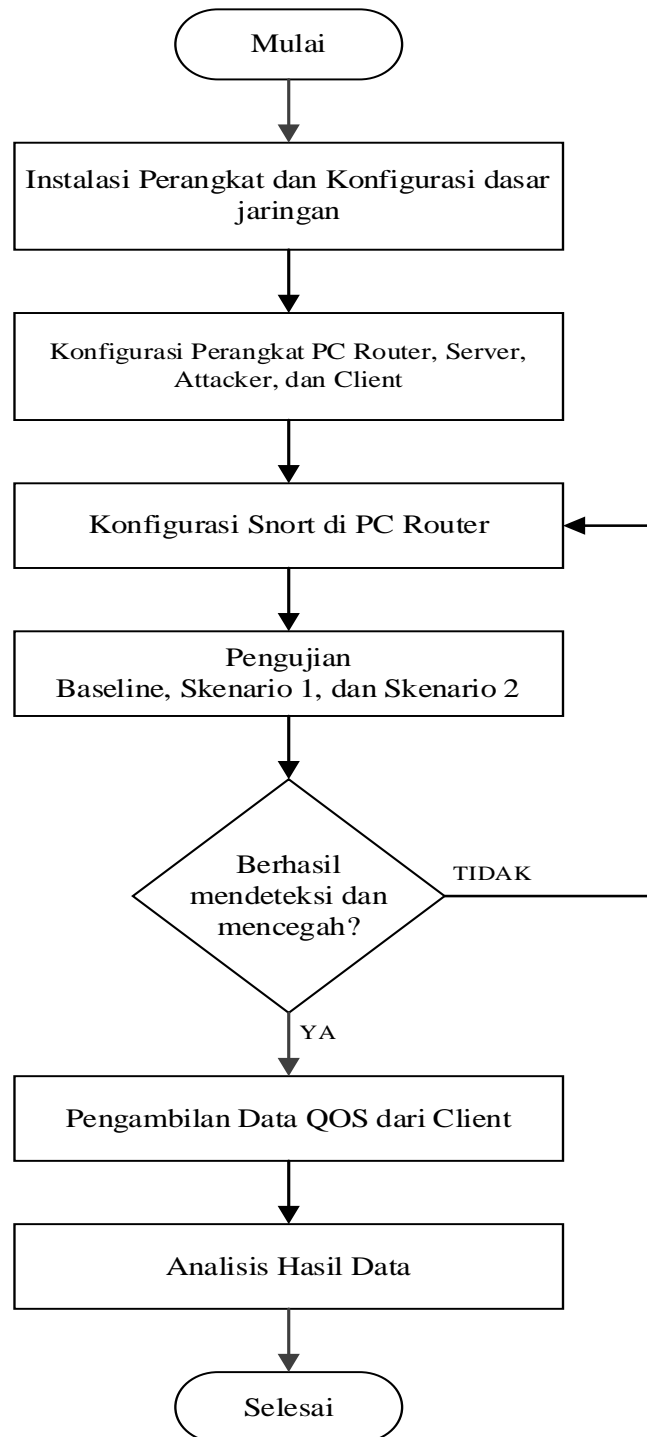


BAB III METODE PENELITIAN

3.1 ALUR PENELITIAN

Diagram alur penelitian ini ditampilkan dalam bentuk *flowchart* seperti pada gambar 3.1.



Gambar 3. 1 Flowchart Alur Penelitian

Gambar 3.1 merupakan *flowchart* alur penelitian yang akan dilakukan. Tahap pertama yaitu Instalasi perangkat dan konfigurasi dasar jaringan dengan memastikan dapat terhubung antar perangkat. Tahap kedua yaitu konfigurasi Perangkat *PC Router, Server, Attacker, dan Client*. Konfigurasi pada *PC Router* meliputi *instalasi snort* dan konfigurasi *file snort.conf* serta konfigurasi *rules snort*, konfigurasi pada *Server* yaitu membuat *file* FTP berisi *file* yang nantinya akan diunduh oleh *client*, konfigurasi *attacker* meliputi *instalasi hping3*, dan konfigurasi *client* meliputi *instalasi wireshark* untuk mengukur parameter QoS.

Dilanjutkan dengan pengujian, pengujian yang dilakukan terbagi menjadi 3 yaitu skenario 1 pengujian *baseline* pada saat *server* dalam keadaan normal dan *client* mengakses layanan yang disediakan serta mengunduh *file* yang tersedia, kemudian skenario 2 yaitu pada saat serangan ICMP, UDP, dan SYN *flood* dijalankan namun *snort* belum aktif dan *client* mengakses layanan yang disediakan serta mengunduh file yang tersedia, dan skenario 3 yaitu pada saat serangan ICMP, UDP, dan SYN *flood* dijalankan dan *snort* sudah aktif kemudian *client* mengakses layanan yang disediakan serta mengunduh file yang tersedia. Untuk skenario 2 dan 3 apabila serangan terdeteksi dan berhasil dicegah oleh *snort* maka akan dilanjutkan untuk pengambilan Data QoS dari sisi *client*, namun jika *alert* tidak muncul maka harus dilakukan konfigurasi ulang. Setelah skenario 1, 2, dan 3 berhasil diambil datanya maka selanjutnya dilakukan proses perhitungan untuk mengetahui nilai QoS dan Analisis Hasil Data.

3.2 PERANGKAT YANG DIGUNAKAN

Perangkat yang digunakan dalam penelitian ini terbagi menjadi perangkat keras dan perangkat lunak.

Berikut adalah spesifikasi perangkat yang akan digunakan dalam penelitian ini:

1. Perangkat keras

Spesifikasi perangkat keras yang digunakan tertera pada Tabel 3.1

Tabel 3.1 Spesifikasi Perangkat Keras

No	Perangkat	Spesifikasi	Keterangan
1	1 PC	Prosesor Intel Core i7, RAM 8 GB, Hard disk 80 Gb	Sebagai <i>Router</i>

No	Perangkat	Spesifikasi	Keterangan
2	1 PC	Prosesor Intel Core i7, RAM 4 GB, Hard disk 80 Gb	Sebagai <i>attacker</i>
3.	1 PC	Prosesor Intel Core i7, RAM 4 GB, Hard disk 80 Gb	Sebagai <i>Server</i>
4.	1 PC	Prosesor Intel Core i7, RAM 4 GB, Hard disk 80 Gb	Sebagai <i>Client</i>

2. Perangkat lunak

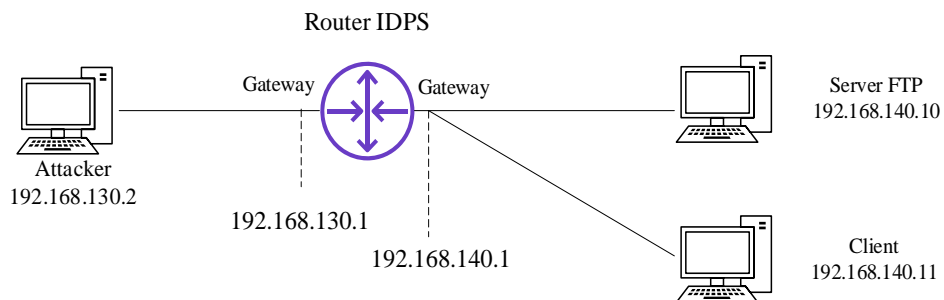
Spesifikasi Perangkat Lunak yang digunakan tertera pada Tabel 3.2

Tabel 3. 2 Spesifikasi Perangkat Lunak

No	Perangkat Lunak	Keterangan
1.	Windows	Sistem Operasi <i>Server</i> dan <i>Client</i>
2.	Linux Ubuntu	Sistem Operasi <i>Router</i> dan <i>Attacker</i>
3.	Snort	<i>Tools</i> IDPS
4.	Hping	<i>Tools</i> <i>attacker</i>
5.	Wireshark	Untuk mengetahui <i>traffic</i>

3.3 TOPOLOGI JARINGAN

Topologi jaringan pada penelitian ini terdiri dari 4 perangkat. Topologi jaringan pada penelitian ini ditunjukkan pada Gambar 3.2



Gambar 3. 2 Topologi Jaringan

Pada Gambar 3.2 tertera topologi yang digunakan dalam penelitian ini,

perangkat pertama berfungsi sebagai PC Router yang menghubungkan *attacker*, *server*, dan *client*. Perangkat kedua berfungsi sebagai *attacker* yang melakukan serangan data flooding, *perangkat* ketiga berfungsi sebagai *Server FTP*, dan perangkat keempat berfungsi sebagai *Client*.

3.4 SKENARIO PENGUJIAN

Pada penelitian ini dilakukan beberapa pengujian untuk melihat kinerja IDPS Snort dalam mengatasi serangan ICMP *flood*, UDP *flood*, dan SYN *flood*. Serangan yang dijalankan *attacker* menggunakan aplikasi Hping3. Pengujian akan dilakukan dengan 4 skenario agar dapat melihat performansi IDPS snort. Skenario pengujian ditampilkan pada tabel dibawah:

Tabel 3. 3 Skenario Pengujian

Skenario	Pengujian	Aktifitas	Kondisi <i>server</i> FTP	IDPS Snort
1	Pengujian <i>Baseline</i>	Unduh film	Kondisi <i>server</i> normal, belum terkena serangan	OFF
2.1	Pengujian ICMP <i>Flood</i>	Unduh film	Kondisi server FTP mengalami serangan ICMP <i>Flood</i>	OFF
2.2	Pengujian ICMP <i>Flood</i>	Unduh film	Kondisi server FTP mengalami serangan ICMP <i>Flood</i>	ON
3.1	Pengujian UDP <i>Flood</i>	Unduh film	Kondisi server FTP mengalami serangan UDP <i>Flood</i>	OFF
3.2	Pengujian UDP <i>Flood</i>	Unduh film	Kondisi server FTP mengalami serangan UDP <i>Flood</i>	ON
4.1	Pengujian SYN <i>Flood</i>	Unduh film	Kondisi server FTP mengalami serangan SYN <i>Flood</i>	OFF
4.2	Pengujian SYN <i>Flood</i>	Unduh film	Kondisi server FTP mengalami serangan SYN <i>Flood</i>	ON

Tabel 3.3 menjelaskan skenario pengujian yang dilakukan pada penelitian ini. Pengujian *baseline* dilakukan saat *server* dalam keadaan normal dan belum menerima serangan, pengukuran QoS dimulai pada saat *client* mengunduh *file* yang disediakan oleh Server FTP dan diakhiri saat file sudah terunduh secara sempurna.

Pada skenario 2.1 pengujian ICMP Flood dilakukan dengan mengirimkan paket ICMP dengan panjang 300 byte secara berulang-ulang pada saat snort belum aktif dan disaat bersamaan client mencoba untuk mengunduh file yang disediakan oleh server FTP. pengukuran QoS dimulai pada saat client mengunduh file dan diakhiri saat file sudah terunduh secara sempurna.

Pada skenario 2.2 pengujian ICMP Flood dilakukan dengan mengirimkan paket ICMP dengan panjang 300 byte secara berulang-ulang pada saat snort sudah aktif dan disaat bersamaan client mencoba untuk mengunduh file yang disediakan oleh server FTP. pengukuran QoS dimulai pada saat client mengunduh file dan diakhiri saat file sudah terunduh secara sempurna.

Pada skenario 3.1 pengujian UDP Flood dilakukan dengan mengirimkan paket UDP dengan panjang 300 byte secara berulang-ulang pada saat snort belum aktif dan disaat bersamaan client mencoba untuk mengunduh file yang disediakan oleh server FTP. pengukuran QoS dimulai pada saat client mengunduh file dan diakhiri saat file sudah terunduh secara sempurna.

Pada skenario 3.2 pengujian UDP Flood dilakukan dengan mengirimkan paket UDP dengan panjang 300 bytes secara berulang-ulang pada saat snort sudah aktif dan disaat bersamaan client mencoba untuk mengunduh file yang disediakan oleh server FTP. pengukuran QoS dimulai pada saat client mengunduh file dan diakhiri saat file sudah terunduh secara sempurna.

Pada skenario 4.1 pengujian SYN Flood dilakukan dengan mengirimkan paket SYN menuju port 80 secara berulang-ulang pada saat snort belum aktif dan disaat bersamaan client mencoba untuk mengunduh file yang disediakan oleh server FTP. pengukuran QoS dimulai pada saat client mengunduh file dan diakhiri saat file sudah terunduh secara sempurna.

Pada skenario 4.2 pengujian SYN Flood dilakukan dengan mengirimkan paket SYN menuju port 80 secara berulang-ulang pada saat snort sudah aktif dan disaat bersamaan client mencoba untuk mengunduh file yang disediakan oleh server FTP. pengukuran QoS dimulai pada saat client mengunduh file dan diakhiri saat file sudah terunduh secara sempurna.

Pada penelitian ini dilakukan pengukuran nilai Quality of Service meliputi throughput, packet loss, delay, dan jitter yang dihasilkan dari setiap scenario yang telah dilakukan. Pengambilan data QoS menggunakan aplikasi Wireshark yang terpasang pada client dan filter yang dipilih yaitu FTP-DATA.

3.5 KONFIGURASI PERANGKAT

Pada tahap ini, peneliti melakukan instalasi dan konfigurasi perangkat yang digunakan selama proses penelitian. Konfigurasi pada penelitian ini meliputi:

1. Konfigurasi Dasar Jaringan
 - a. Konfigurasi *Router* IDPS

Router IDPS berfungsi sebagai jembatan antar network, Router IDPS tidak memiliki IP sendiri melainkan memiliki IP dari dua interface yang digunakan sebagai Gateway. Konfigurasi jaringan pada Router IDPS sebagai berikut:

```
GNU nano 4.8                                01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  #renderer: NetworkManager
  renderer: networkd
  ethernet:
    ens38:
      addresses: [192.168.130.1/24]
      routes:
        - to: 192.168.140.1
    ens39:
      addresses: [192.168.140.1/24]
      routes:
        - to: 192.168.130.1
```

Gambar 3. 3 Konfigurasi jaringan Router IDPS

Gambar 3.3 menunjukkan konfigurasi jaringan pada Router IDPS, Pengaturan IP statis dapat diedit pada direktori */etc/netplan*. Router IDPS terdiri dari *interface* ens38 dengan alamat gateway 192.168.130.1 dan *interface* ens39 dengan alamat gateway 192.168.140.1. Agar kedua network dapat saling terhubung, maka dibutuhkan proses *routing* statis dengan menambahkan perintah routes to.

- b. Konfigurasi *Attacker*

Konfigurasi jaringan pada attacker yaitu dengan menambahkan IP secara manual dengan perintah *nano 01-network-manager-all.yaml*

```

GNU nano 4.8                                01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  renderer: networkd
  ethernets:
    ens38:
      addresses: [192.168.130.2/24]
      gateway4: 192.168.130.1

```

Gambar 3. 4 Konfigurasi jaringan Attacker

Gambar 3.4 menunjukkan konfigurasi jaringan Attacker, penambahan IP statis dapat di edit melalui direktori /etc/netplan. Attacker menggunakan interface ens38 dengan alamat IP 192.168.130.2.

c. Konfigurasi *Server FTP*

Konfigurasi jaringan Server dengan menambahkan ip statis seperti gambar dibawah:

```

Microsoft Windows [Version 10.0.19042.985]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : 
   Link-local IPv6 Address . . . . . : fe80::bc47:f0d3:3b37:e148%4
   IPv4 Address. . . . . : 192.168.140.10
   Subnet Mask . . . . . : 255.255.255.0
   Default Gateway . . . . . : 192.168.140.1

Ethernet adapter Bluetooth Network Connection:

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

```

Gambar 3. 5 Konfigurasi jaringan server FTP

Gambar 3.5 menunjukkan konfigurasi server FTP dengan alamat IP 192.168.140.10 dan menggunakan interface ens39 sebagai gateway.

d. Konfigurasi *Client*

Konfigurasi jaringan Client dengan menambahkan ip statis seperti gambar dibawah:

```

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : 
   Link-local IPv6 Address . . . . . : fe80::25bc:9d4d:e34b:5670%8
   IPv4 Address. . . . . : 192.168.140.11
   Subnet Mask . . . . . : 255.255.255.0
   Default Gateway . . . . . : 192.168.140.1

Ethernet adapter Bluetooth Network Connection:

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

```

Gambar 3. 6 Konfigurasi jaringan Client

Gambar 3.6 menunjukkan konfigurasi server FTP dengan alamat IP 192.168.140.11 dan menggunakan interface ens39 sebagai gateway.

2. Instalasi *Snort*

Pada tahap ini konfigurasi yang dilakukan meliputi instalasi dan konfigurasi *rules Snort*.

a. *Update dan Upgrade System*

Sebelum memulai menginstall *Snort*, pastikan *system* sudah *terupdate* dan *terupgrade* dengan menggunakan perintah:

Tabel 3. 4 *Update dan Upgrade System*

1	<code>apt update -y</code>
2	<code>apt upgrade -y</code>

Tabel 3.4 menunjukkan perintah untuk melakukan update dan upgrade system yang terdapat di Ubuntu.

b. Instalasi *Depedencies*

Instalasi *Depedencies* dapat dilakukan dengan perintah:

Tabel 3. 5 Instalasi *Depedencies*

1	<code>apt install openssh-server ethtool build-essential libpcap-dev libpcrc3-dev libdumbnet-dev bison flex zlib1g-dev openssl libssl-dev autoconf</code>
---	---

Tabel 3.5 menunjukkan perintah untuk menginstal *Depedencies* yang digunakan dalam *snort*. *Depedencies* atau *Required build tools* merupakan program yang mengotomatiskan pembuatan aplikasi yang dapat dieksekusi dari kode sumber

c. Instalasi *Data Acquisition library (DAQ)*

Instalasi *DAQ* yang akan digunakan oleh *snort* dapat menggunakan perintah:

Tabel 3. 6 Instalasi *DAQ*

1	<code>cd Snort-Installation-Files</code>
2	<code>wget https://www.snort.org/downloads/snort/daq-2.0.7.tar.gz</code>
3	<code>tar -zxvf daq-2.0.7.tar.gz</code>
4	<code>cd daq-2.0.7</code>
5	<code>./configure --enable-nfq=yes && make && make install</code>

Tabel 3.6 menunjukkan Instalasi DAQ, DAQ merupakan sebuah *library* yang dapat digunakan dalam IDS maupun IPS. DAQ bertanggung jawab untuk menangkap paket jaringan yang keluar masuk pada sistem. Ketika paket datang, DAQ akan mengambil data dari paket tersebut untuk kemudian diberikan kepada IDS/IPS untuk dianalisis lebih.

d. Instalasi *Snort*

Pada tahap instalasi *Snort*, diawali dengan melakukan *update* dan *upgrade system*, *install dependencies*, dan *instal* DAQ yang dibutuhkan oleh *Snort*. Jika semuanya sudah terinstal, maka dilanjutkan dengan instalasi *snort* dengan perintah dibawah ini:

Tabel 3. 7 Instalasi *Snort*

1	<code>cd Snort-Installation-Files</code>
2	<code>wget https://www.snort.org/downloads/snort/snort-2.9.20.tar.gz</code>
3	<code>tar -xvzf snort-2.9.20.tar.gz</code>
4	<code>cd snort-2.9.20</code>
5	<code>./configure --enable-sourcefire && make && make install</code>
6	<code>Snort -V</code>

Tabel 3.7 menunjukkan perintah untuk instalasi *snort* secara manual, untuk memastikan *snort* sudah *terinstall* pada perangkat dapat menggunakan perintah `snort -V`

```

root@cantikaw-Ubuntu:/home/cantikaw# snort -V
    ,,_      -*> Snort! <*-
  o"  )~    Version 2.9.20 GRE (Build 82)
   "'      By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
          Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.

```

Gambar 3. 7 Tampilan *Snort*

Gambar 3.7 menunjukkan tampilan awal *snort*, apabila *snort* sudah berhasil *terinstall* maka akan muncul halaman *snort* beserta versinya.

2. Konfigurasi *Snort*

Konfigurasi utama terdapat pada file *snort.conf* yang terletak di direktori `/etc/snort/snort.conf`

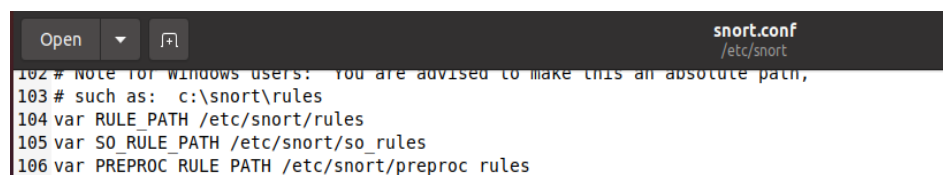
```

44 # Setup the network addresses you are protecting
45 ipvar HOME_NET 192.168.140.0/24
46
47 # Set up the external network addresses. Leave as "any" in most situations
48 ipvar EXTERNAL_NET !$HOME_NET
49

```

Gambar 3. 8 Konfigurasi snort.conf

Gambar 3.8 Menunjukkan isi pada file *snort.conf*. Tahap pertama yang perlu dikonfigurasi pada file *snort.conf* yaitu menentukan alamat HOME_NET, alamat HOME_NET merupakan *network* yang ingin dilindungi oleh *Snort* IDPS. Pengisian *network* berguna untuk menjadikan *snort* dapat berjalan sebagai mode IDPS, pada penelitian ini *network* yang ingin dilindungi yaitu 192.168.140.0/24. EXTERNAL_NET merupakan alamat network yang dianggap mencurigakan atau perlu diawasi, pada penelitian ini alamat HOME_NET diisi !\$HOME_NET yang berarti *network* selain 192.168.140.0/24 dianggap sebagai penyerang sehingga perlu diawasi.



```

102 # NOTE FOR windows users: You are advised to make this an absolute path,
103 # such as: c:\snort\rules
104 var RULE_PATH /etc/snort/rules
105 var SO_RULE_PATH /etc/snort/so_rules
106 var PREPROC_RULE_PATH /etc/snort/preproc_rules

```

Gambar 3. 9 Konfigurasi Direktori

Gambar 3.9 menunjukkan konfigurasi direktori, konfigurasi ini mengisi RULE_PATH, SO_RULE_PATH, dan PREPROC_RULE_PATH yang semuanya terletak dalam direktori */etc/snort*.



```

544
545 # site specific rules
546 include $RULE_PATH/local.rules

```

Gambar 3. 10 Konfigurasi lokasi rules Snort

Gambar 3.10 menunjukkan konfigurasi lokasi rules snort, yang digunakan, untuk menentukan *rules* yang digunakan yaitu dengan menghapus tanda # pada lokasi *rules* yang dikonfigurasi. Pada penelitian ini *rules* yang digunakan yaitu *local rules*, terletak pada direktori */etc/snort/rules/local.rules*.

3. Konfigurasi Rules Snort

Rules snort yang digunakan tersimpan pada file *local.rules* yang terletak pada direktori */etc/snort/rules*. file *local.rules* berisikan aturan *custom* yang dibuat oleh peneliti menyesuaikan dengan kebutuhan. Penelitian ini

menggunakan serangan *ICMP Flood*, *UDP Flood*, dan *SYN Flood*. *Drop*, merupakan perintah yang digunakan untuk menolak atau membuang paket yang sesuai dengan *rules* yang sudah dibuat, *drop* bersifat agresif dalam menangani paket yang dianggap berbahaya.

```

1 #ICMP RULES
2 drop icmp !$HOME_NET any -> 192.168.140.10 any (msg:"Warning!! ICMP Flood Attempt!"; dsize:>100;
  classtype:attempted-dos; detection_filter: track by_src, count 50, seconds 1; sid:2000002; rev:1;)
3
4 #UDP RULES
5 drop udp !$HOME_NET any -> 192.168.140.10 any (msg:"Warning!! UDP Flood Attempt!"; dsize:>100;
  classtype:attempted-dos; detection_filter: track by_src, count 300, seconds 1; sid:3000002; rev:1;)
6
7 #SYN RULES
8 drop tcp !$HOME_NET any -> 192.168.140.10 any (msg:"Warning!! SYN Flood Attempt!"; flags:S;
  classtype:attempted-dos; detection_filter: track by_src, count 1000, seconds 1; sid:4000002; rev:1;)

```

Gambar 3. 11 Rules Snort

Gambar 3.11 menunjukkan konfigurasi *rules* yang menjadi acuan *snort* dalam mendeteksi dan mencegah serangan yang masuk. Umumnya *rules snort* yang dimodifikasi sesuai kebutuhan terletak pada file *local.rules*, Pada penelitian ini isi dari *local.rules* yaitu berisi mengenai *rules* untuk serangan *ICMP flood*, *UDP flood*, dan *SYN/TCP flood*. Berikut penjelasan dari setiap *rules* yang telah dibuat:

```

1 #ICMP RULES
2 drop icmp !$HOME_NET any -> 192.168.140.10 any (msg:"Warning!! ICMP Flood Attempt!"; dsize:>100;
  classtype:attempted-dos; detection_filter: track by_src, count 50, seconds 1; sid:2000002; rev:1;)

```

Gambar 3. 12 Rules ICMP Flood

Gambar 3.12 Rules ICMP pada baris 2 bertujuan untuk memberikan alert dan mendrop paket yang berasal dari alamat selain HOME_NET dari port manapun yang menuju alamat 192.168.140.10 port manapun, dengan panjang paket lebih dari 100 bytes. Filter drop akan otomatis aktif jika ada 50 paket yang dikirim oleh alamat yang sama dalam 1 detik maka paket itu dianggap sebagai serangan dan akan di drop, sehingga akan muncul pemberitahuan “Warning!! ICMP Flood Attempt!”.

```

4 #UDP RULES
5 drop udp !$HOME_NET any -> 192.168.140.10 any (msg:"Warning!! UDP Flood Attempt!"; dsize:>100;
  classtype:attempted-dos; detection_filter: track by_src, count 300, seconds 1; sid:3000002; rev:1;)

```

Gambar 3. 13 Rules UDP Flood

Gambar 3.13 Rules UDP pada baris 5 bertujuan untuk memberikan alert dan mendrop paket yang berasal dari alamat selain HOME_NET dari port manapun yang menuju alamat 192.168.140.10 port manapun, dengan panjang

paket lebih dari 100 bytes. Filter drop akan otomatis aktif jika ada 300 paket yang dikirim oleh alamat yang sama dalam 1 detik maka paket itu dianggap sebagai serangan dan akan di drop, sehingga akan muncul pemberitahuan “Warning!! UDP Flood Attempt!”.

```
7 #SYN RULES
8 drop tcp !$HOME_NET any -> 192.168.140.10 any (msg:"Warning!! SYN Flood Attempt!"; flags:S;
  classtype:attempted-dos; detection_filter: track by_src, count 1000, seconds 1; sid:4000002; rev:1;)
```

Gambar 3. 14 Rules SYN Flood

Gambar 3.14 Rules SYN pada baris 8 bertujuan untuk memberikan alert dan mendrop paket yang berasal dari alamat selain HOME_NET dari port manapun yang menuju alamat 192.168.140.10 port manapun, dengan kondisi paket TCPflags berjenis SYN(S). Filter drop akan otomatis aktif jika ada 1000 paket yang dikirim oleh alamat yang sama dalam 1 detik maka paket itu dianggap sebagai serangan dan akan di drop, sehingga akan muncul pemberitahuan “Warning!! SYN Flood Attempt!”.

3.6 PENGUJIAN SERANGAN

Proses pengujian serangan terbagi menjadi dua skenario. Skenario pertama yaitu pada saat belum terinstall *snort* beserta konfigurasinya dan skenario kedua yaitu pada saat sudah terinstall *snort* beserta konfigurasi yang dibutuhkan. Pengujian ini dilakukan dengan mengirimkan serangan *Denial of Service* (DoS) dari PC Attacker menuju PC Server. Supaya *Snort* dapat bekerja sebagai IDPS, maka *Snort* harus berjalan pada mode *inline*.

Dalam mode *inline*, *Snort* tidak hanya dapat berfungsi sebagai *Intrusion Detection System* (IDS) yang mengamati lalu lintas dan memberikan *alert* apabila terdapat serangan, tetapi juga memiliki kemampuan untuk mencegah serangan yang masuk. Untuk mengubah *snort* menjadi mode *inline* dapat menggunakan perintah:

```
root@cantikaw-Ubuntu:/etc/snort/rules# snort -Q --daq nfq --daq-mode inline --daq-var queue=0 -c /etc/snort/snort.conf -A Console -l /var/log/snort/
```

Gambar 3. 15 Perintah mode inline

Gambar 3.15 berisi perintah *snort* untuk berjalan dalam *sebagai Intrusion Detection System* (IDS) atau *Intrusion Prevention System* (IPS). Keterangan lengkap pada tabel diatas adalah sebagai berikut:

- a) *Snort*: perintah untuk menjalankan aplikasi *snort*

- b) `-Q`: opsi untuk mengaktifkan mode inline
- c) `--daq nfq`: menentukan jenis DAQ yang digunakan, `nfq` merupakan jenis DAQ
- d) `--daq-mode inline`: menandakan snort berjalan dalam mode inline
- e) `--daq-var queue=0`: mengatur variabel DAQ "`queue`" dengan nilai 0, yang menentukan nomor antrean NFQUEUE yang akan digunakan oleh *Snort* untuk mengambil paket dari antrean tersebut
- f) `-c /etc/snort/snort.conf`: lokasi dan nama file konfigurasi Snort yang akan digunakan.
- g) `-A Console`: menentukan output dari Snort yang akan ditampilkan di terminal.
- h) `-l /var/log/snort/`: lokasi direktori log di mana log dan hasil deteksi *Snort* akan disimpan

Jenis serangan yang di lakukan dalam penelitian ini meliputi:

1. Serangan ICMP Flood

Pengujian ICMP Flood menggunakan tools Hping3 yang terinstall pada PC attacker, perintah untuk menjalankan ICMP Flood menggunakan kode `-I` yang berarti ICMP, keterangan lengkap dari *script* diatas adalah sebagai berikut:

```
root@cantikaw-Ubuntu:/home/cantikaw# hping3 -I --data 300 --flood 192.168.140.10
HPING 192.168.140.10 (ens38 192.168.140.10): icmp mode set, 28 headers + 300 data bytes
hping in flood mode, no replies will be shown
```

Gambar 3. 16 Perintah pengujian ICMP Flood

Gambar 3.16 menunjukkan perintah untuk menjalankan serangan ICMP flood, keterangan lengkap dari *script* diatas adalah sebagai berikut:

- a) *Hping3*: perintah untuk menjalankan hping3
- b) `-I`: jenis paket yang akan dikirimkan, jenis paket yang dikirim yaitu ICMP
- c) `--data 300`: panjang data (payload) dalam paket ICMP yang akan dikirimkan
- d) `--flood`: untuk membanjiri tujuan dengan paket secepat mungkin tanpa menunggu respon.
- e) 192.168.140.2: ip server yang akan diserang

```
root@cantikaw-Ubuntu: /home/cantikaw × root@cantikaw-Ubuntu: /home/cantikaw ×
07/29-13:04:00.457513 [Drop] [**] [1:2000002:1] Warning!! ICMP Flood Attempt! [**]
[Classification: Attempted Denial of Service] [Priority: 2] {ICMP} 192.168.130.2 ->
192.168.140.10
07/29-13:04:00.457513 [Drop] [**] [1:2000002:1] Warning!! ICMP Flood Attempt! [**]
[Classification: Attempted Denial of Service] [Priority: 2] {ICMP} 192.168.130.2 ->
192.168.140.10
07/29-13:04:00.457513 [Drop] [**] [1:2000002:1] Warning!! ICMP Flood Attempt! [**]
[Classification: Attempted Denial of Service] [Priority: 2] {ICMP} 192.168.130.2 ->
192.168.140.10
```

Gambar 3. 17 Alert ICMP Flood

Gambar 3.17 menunjukkan *alert* yang tercatat pada saat *snort* berjalan

dalam mode *inline*, *alert* ini muncul pada saat menerima serangan ICMP *flood* sesuai dengan konfigurasi yang tersimpan di *local.rules*.

2. Serangan UDP Flood

Pengujian UDP Flood menggunakan tools Hping3 yang terinstall pada PC attacker, perintah untuk menjalankan UDP Flood yaitu:

```
root@cantikaw-Ubuntu:/home/cantikaw# hping3 -2 --data 300 --flood 192.168.140.10
HPING 192.168.140.10 (ens38 192.168.140.10): udp mode set, 28 headers + 300 data bytes
hping in flood mode, no replies will be shown
```

Gambar 3. 18 Perintah pengujian UDP Flood

Gambar 3.18 menunjukkan perintah untuk menjalankan serangan UDP *flood*, perintah untuk menjalankan UDP Flood menggunakan kode -2 yang berarti UDP, keterangan lengkap dari *script* diatas adalah sebagai berikut:

- a) *Hping3*: perintah untuk menjalankan hping3.
- b) -2: jenis paket yang akan dikirimkan, jenis paket yang dikirim yaitu UDP.
- c) *--data 300*: panjang data (*payload*) dalam paket ICMP yang akan dikirimkan (*byte*).
- d) *--flood*: untuk membanjiri tujuan dengan paket tanpa menunggu respon.
- e) 192.168.140.2: ip server yang akan diserang.

```
root@cantikaw-Ubuntu: /home/cantikaw × root@cantikaw-Ubuntu: /home/cantikaw ×
07/29-13:05:02.740463 [Drop] [**] [1:3000002:1] Warning!! UDP Flood Attempt! [**] [
Classification: Attempted Denial of Service] [Priority: 2] {UDP} 192.168.130.2:27504
-> 192.168.140.10:0
07/29-13:05:02.740464 [Drop] [**] [1:3000002:1] Warning!! UDP Flood Attempt! [**] [
Classification: Attempted Denial of Service] [Priority: 2] {UDP} 192.168.130.2:27505
-> 192.168.140.10:0
07/29-13:05:02.740579 [Drop] [**] [1:3000002:1] Warning!! UDP Flood Attempt! [**] [
Classification: Attempted Denial of Service] [Priority: 2] {UDP} 192.168.130.2:27506
-> 192.168.140.10:0
```

Gambar 3. 19 Alert UDP Flood

Gambar 3.19 menunjukkan alert yang tercatat pada saat *snort* berjalan dalam mode *inline*, alert ini muncul pada saat menerima serangan UDP *flood* sesuai dengan konfigurasi yang tersimpan di *local.rules*.

3. Serangan SYN Flood

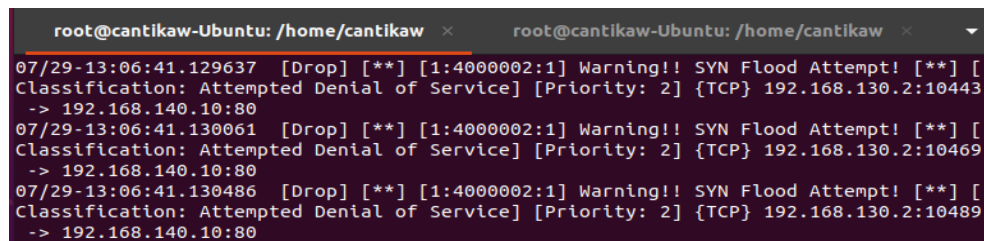
Pengujian SYN Flood menggunakan tools Hping3 yang terinstall pada PC attacker, perintah untuk menjalankan SYN Flood yaitu:

```
root@cantikaw-Ubuntu:/home/cantikaw# hping3 -S --flood -p 80 192.168.140.10
HPING 192.168.140.10 (ens38 192.168.140.10): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Gambar 3. 20 Perintah pengujian SYN Flood

Gambar 3.20 menunjukkan perintah untuk menjalankan serangan SYN flood, perintah untuk menjalankan SYN Flood menggunakan kode -S yang berarti SYN, keterangan lengkap dari *script* diatas adalah sebagai berikut:

- a) *Hping3*: perintah untuk menjalankan hping3.
- b) -S: jenis paket yang akan dikirimkan, jenis paket yang dikirim yaitu SYN.
- c) -p 80: port tujuan port 80.
- d) --flood: untuk membanjiri tujuan dengan paket secepat mungkin tanpa menunggu respon.
- e) 192.168.140.2: ip server yang akan diserang.



```
root@cantikaw-Ubuntu: /home/cantikaw × root@cantikaw-Ubuntu: /home/cantikaw ×
07/29-13:06:41.129637 [Drop] [**] [1:4000002:1] Warning!! SYN Flood Attempt! [**] [
Classification: Attempted Denial of Service] [Priority: 2] {TCP} 192.168.130.2:10443
-> 192.168.140.10:80
07/29-13:06:41.130061 [Drop] [**] [1:4000002:1] Warning!! SYN Flood Attempt! [**] [
Classification: Attempted Denial of Service] [Priority: 2] {TCP} 192.168.130.2:10469
-> 192.168.140.10:80
07/29-13:06:41.130486 [Drop] [**] [1:4000002:1] Warning!! SYN Flood Attempt! [**] [
Classification: Attempted Denial of Service] [Priority: 2] {TCP} 192.168.130.2:10489
-> 192.168.140.10:80
```

Gambar 3. 21 Alert SYN Flood

Gambar 3.21 menunjukkan *alert* yang tercatat pada saat *snort* berjalan dalam mode *inline*, alert ini muncul pada saat menerima serangan SYN flood sesuai dengan konfigurasi yang tersimpan di *local.rules*