Methotika : Jurnal Ilmiah Teknik Informatika
e-ISSN :2776-5792
Vol. 2, No. 2, Oktober 2022, pp. 1-10
http://ojs.fikom-methodist.net/index.php/METHOTIKA

Monitoring Keamanan Jaringan Pada Server Ubuntu Dari Serangan DDoS Menggunakan Snort IDS

Lasma Feronika Nainggolan[1], Naikson F. Saragih[2], Fati G.N. Larosa[3]
[1,2,3]Fakultas Ilmu Komputer, Universitas Methodist Indonesia

Info Artikel

Histori Artikel:

Received, Jul 19, 202
Revised, Jul 30, 2020
Accepted, Aug 4, 2020

Keywords:

Snort IDS,
Ubuntu,
DDoS,
Flooding

ABSTRAK

Salah satu tindak kejahatan pada jaringan komputer adalah serangan *Distributed Denial of Service* (DDoS). Berbagai macam serangan DDoS diantaranya ICMP *flooding* (*ping of death*), TCP *flooding* dan UDP *flooding*. Serangan DDoS menggunakan sejumlah *host* untuk membanjiri komputer korban dengan request informasi sehingga sistem tidak dapat bekerja dengan normal yang berakibat sistem tidak dapat memberikan layanan sebagaimana mestinya. Serangan DDoS juga dapat digunakan untuk melumpuhkan sebuah server. Salah satu server yang kerap digunakan adalah server Ubuntu. Oleh sebab itu maka dibutuhkanya sebuah fitur IDS yang terdapat pada server Ubuntu untuk memonitoring serangan DDoS. Perangkat lunak IDS yang dapat digunakan adalah snort. *Snort* berfungsi untuk memproteksi server linux Ubuntu sebagai OS untuk menjalankan *snort*. Dengan memanfaatkan sistem proteksi IDS yang terdapat pada *snort* diharapkan dapat membantu memproteksi adanya gangguan serangan terhadap server.

Penulis Koresponden:

Naikson F. Saragih,
Faculty of Computer Science,
Universitas Methodist Indonesia, Medan,
Jl. Hang Tua No.8, Medan - Sumatera Utara.
Email: saragihnaikson@gmail.com

1. PENDAHULUAN

Kemajuan teknologi internet membawa dampak positif untuk berbagai industri, perkembangan ini dapat membantu pertumbuhan industri, dan seiring berkembangnya teknologi komputer, komputer tidak luput dari suatu-suatu serangan atau kejahatan sistem diluar kendali *firewall* yang dapat memantau kejahatan atau serangan diluar kendali dari pemilik komputer tersebut. Salah satu tindak kejahatan pada jaringan komputer adalah serangan *Distributed Denial of Service* (DDoS). Serangan *Distributed Denial of Service* (DDoS) adalah serangan yang mungkin bisa sering kita jumpai diantara serangan-serangan lainnya.

Serangan DoS menggunakan sejumlah *host* untuk membanjiri komputer korban dengan *request* informasi sehingga sistem tidak dapat bekerja dengan normal yang berakibat sistem tidak dapat memberikan layanan sebagaimana mestinya. Berbagai macam serangan DDoS diantaranya ICMP *flooding* (*ping of death),* TCP *flooding* dan UDP *flooding* [1]. Salah satu bentuk dari serangan

DDoS yang sering digunakan adalah *ping of death*, dimana penyerangan akan memanfaatkan komunikasi ICMP untuk dibanjiri oleh paket data yang diminta, sehingga membuat sistem server menjadi lambat. Serangan lainya adalah jenis serangan TCP *flooding* dan UDP *flooding* yang menbanjari paket-paket TCP dan UDP secara terus menerus melalui client fiktif hingga membuat server menjadi *down* [2]. Oleh sebab itu, solusi yang dapat digunakan adalah dengan dibutuhkannya rancangan sistem yang dapat menjaga jaringan itu sendiri. Sistem jaringan komputer harus dilengkapi dengan sistem yang dapat mendeteksi adanya penyusupan atau intrusi (*intrusion*).

Sistem tersebut yang dikenal dengan istilah *Snort*. *Snort* merupakan *tool* yang berbasis *Intrusion Detection System* (IDS) yang dapat memonitor jaringan yang berdampak serangan dan menyimpan serangan pada log [3]. *Intrusion Detection System* (IDS) diterapkan karena mampu mendeteksi paket-paket berbahaya pada jaringan, bekerja sebagai pendeteksi aktivitas yang mencurigakan pada jaringan [4][5]. Dengan menggunakan *Intrusion Detection System* (IDS) sebagai proteksi jaringan masih belum mangkus/mumpuni dalam menjaga jaringan tersebut. Penambahan *mode inline* atau *Intrusion Prevention System* menjadi salah satu cara yang mampu mendeteksi serangan dan melakukan *Drop* pada serangan [6].

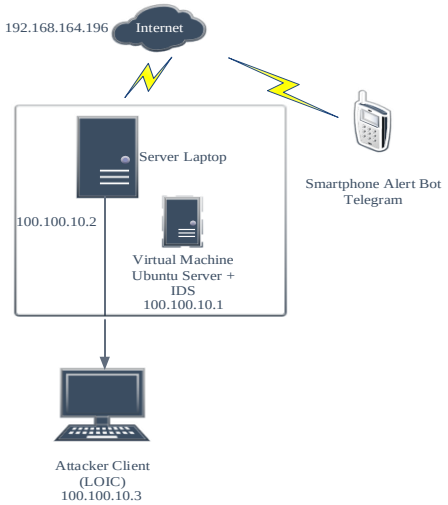
Pada penelitian ini, dilakukan serangan DDoS jenis ICMP *flooding* (*ping of death)*, TCP *flooding* dan UDP *flooding* untuk melihat sejauh mana pengukuran dari kinerja dari keamanan data dari suatu simulasi serangan terhadap server pada OS Ubuntu Server. Serangan DDoS akan dilakukan terhadap server yang akan dibangun dengan menggunakan beberapa *host* agar dapat memaksimalkan pengujian serangan terhadap server Ubuntu serta mengetahui ketahanan server dari serangan DDoS. Untuk menambah keamanan monitoring serangan, maka digunakan fitur notifikasi untuk memberikan peringat kepada admin server ketika adanya serangan DDoS. Peringatan notifikasi serangan DDoS dapat dilakukan dengan konfigurasi *alert bot* telegram pada *snort* ubuntu. Dengan adanya notifikasi *bot* telegram dalam sistem tersebut maka akan memberikan kemudahan bagi *admin* dalam monitoring jaringan secara *mobile*, dan memungkinkan *admin* mendapatkan notifikasi bila terjadi *attacker*.

2. METODE PENELITIAN

Metode atau tahapan yang digunakan untuk menghasilkan suatu pengetahuan baru adalah menggunakan metode PPDIOO (*Prepare and Plan, Design, Implementation, Operate and Optimize*) [7]. PPDIOO merupakan metode desain jaringan dengan pendekatan *network lifecycle*, sehingga tahapan awal adalah melalukan perancangan topologi jaringan, kemudian konfigurasi awal sistem operasi ubuntu *server* yang sudah terinstall dengan melakukan pemberian ip *address* serta konfigurasi instalasi *snort*.

2.1 Rancangan Topologi

Adapun sistem yang akan di bangun dapat digambarkan dengan topologi yang dibangun sesuai pada penelitian ini seperti ditunjukkan pada gambar 1 berikut:



Gambar 1. Topologi Sistem IDS Yang Akan di bangun

Berdasarkan pada gambar 1 di atas, dapat dijelaskan dengan pengalamatan IP pada tabel berikut ini:

Tabel 1. Pengalamatan Ip Address

Hardware/Software Network	*Port Ethernet*	Alamat IP / IP Address
Internet melalui Hotspot	-	Address 192.168. 164.196
Laptop Server	Ether 1	*Dynamic Host Configuration Protocol* (DHCP)
	Ether 2	*Address* 100.100.10.2
Virtual Machine Ubuntu Server + IDS	Ether 0	Bridge : *Wireless Card* (Mengikuti koneksi dari Ether 1) *Address* 192.168.164.196
	Ether 1	*Bridge* : LAN Card (Mengikuti koneksi dari Ether 2) *Address* 100.100.10.1
Client Terhubung Jaringan Lokal	*Port switch*	*Address* 100.100.10.2-100.100.10.254 *Gateway* 100.100.10.1

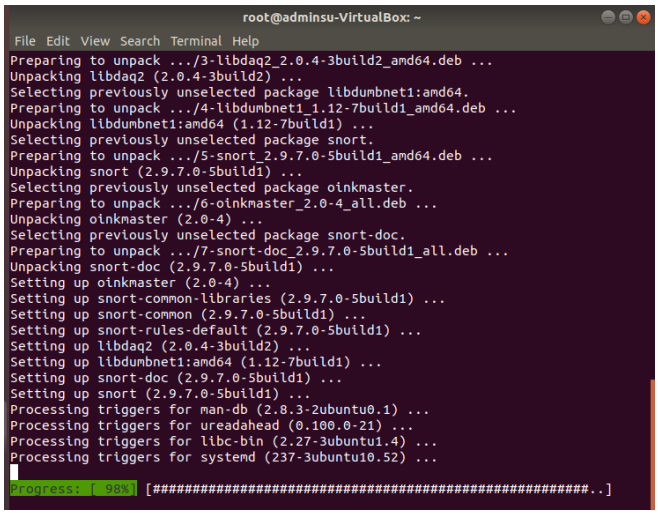
2.2 Konfigurasi Snort IDS

Implementasi sistem terdiri dari proses konfigurasi sistem server ubuntu yang terdiri dari:

1. Konfigurasi Snort IDS
   Setelah dilakukanya konfigurasi IP Address selanjutnya melakukan installasi paket snort dengan script sebagai berikut:

```
=>  Menginstall Paket Snort
# nano apt-get install snort

=>  Restart Paket Snort
# /etc/init.d/snort restart

=>  Membuat penyimpanan untuk Snort
# mkdir /etc/Snort
# mkdir /etc/Snort/rules

=>  Membuat beberapa file
# touch /etc/Snort/rules/local.rules

=> Membuat ruang penyimpanan log
# mkdir /var/log/Snort

=> Memberikan hak akses
# chmod -R 777 /etc/Snort
# chmod -R 777 /var/log/Snort
# chmod -R 777 /etc/Snort/so_rules
```

Berdasarkan *script* di atas, berikut konfigurasi installasi paket *snort* pada ubuntu seperti pada Gambar 2:



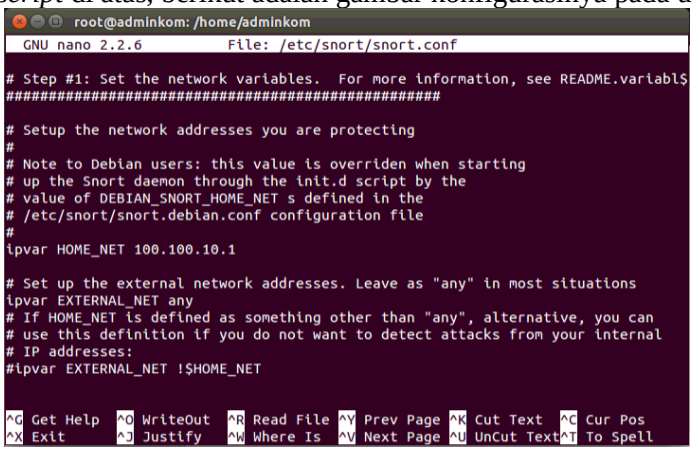
Gambar 2. Proses Installasi *Snort*

2. Konfigurasi *Rule Snort* IDS

Setelah installasi dan konfigurasi *snort* berhasil dilakukan, selanjutnya adalah melakukan konfigurasi *rule snort* IDS dengan kententuan *script* di bawah ini:

```
=> Merubah IP Address Paket Snort
# nano /etc/snort/snort.conf  => Lakukan perubahan script ipvar HOME NET dengan Ip Address Server (100.100.10.1)
```

Berdasarkan pada *script* di atas, berikut adalah gambar konfigurasinya pada ubuntu server:

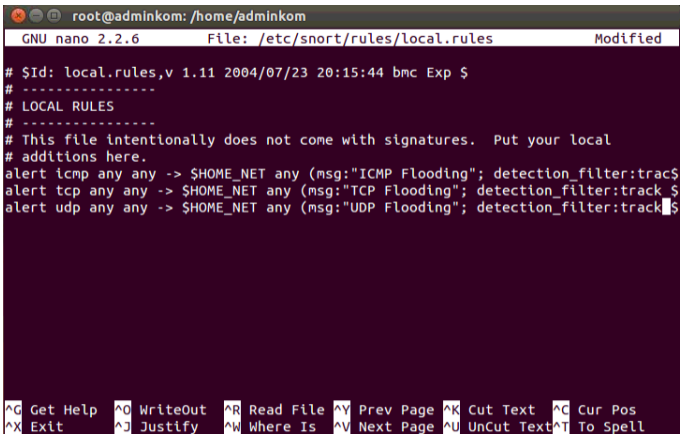


Gambar 3. Konfigurasi *Snort.Conf*

Selanjutnya melakukan penambahan *rules* serangan DDoS dengan *script* seperti di bawah ini:

```
=> Menambahkan Rules Snort
# nano /etc/snort/rules/local.rules  => Lakukan penambahan script rules serangan DDoS seperti berikut:
alert icmp any any -> $HOME_NET any (msg:"ICMP Flooding"; detection_filter:tracks by_src, count 30, second 60; sid1000006; rev2;)
alert tcp any any -> $HOME_NET any (msg:"TCP Flooding"; detection_filter:tracks by_src, count 30, second 60; sid1000006; rev2;)
alert udp any any -> $HOME_NET any (msg:"UDP Flooding"; detection_filter:tracks by_src, count 30, second 60; sid1000003; rev1;)
```

Berdasarkan pada *script* di atas, berikut adalah hasil penambahan *script* konfigurasi *rules snort* IDS pada ubuntu server seperti ditunjukkan pada gambar 4:



Gambar 4. Konfigurasi *Rules Snort*

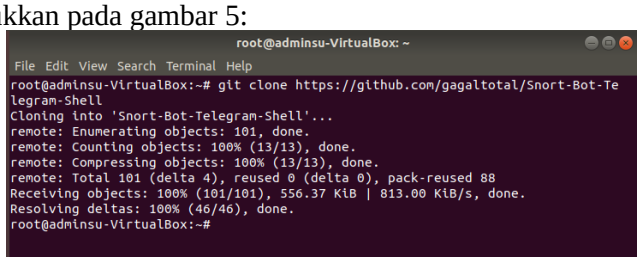
Berdasarkan pada gambar 4 di atas, konfigurasi *rules snort* berdasarkan serangan DDoS yang terdiri dari serangan DDoS ICMP *Flooding* (*Ping of Death*), TCP *Flooding* dan UDP *Flooding*.

3. Konfigurasi *Bash Shell* Bot Telegram *Snort* IDS

Selanjutnya adalah melakukan konfigurasi *Bash Shell* yang akan menjadi *command* informasi ketika adanya serangan serta penghubung monitoring serangan kepada aplikasi bot telegram. Adapun *script* konfigurasi *Bash Shell* sebagai berikut:

```
=> Install Paket Bash Shell Snort
# git clone https://github/gagaltotal/Snort-Bot-Telegram-Shell => berfungsi untuk mendownload pack bash shell bot telegram pada github.
```

Berdasarkan pada *script* di atas, berikut adalah hasil installasi *bash shell* telegram pada ubuntu server seperit ditunjukkan pada gambar 5:

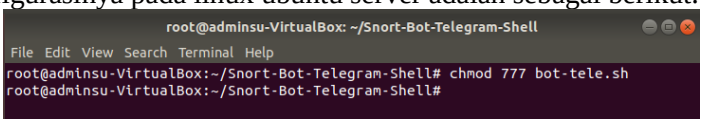


Gambar 5. Proses Installasi *Bash Shell Snort* Bot Telegram

Selanjutnya berikan akses penuh adalah modifikasi script penghubung snort ubuntu dengan bot telegram dengan sscript sebagai berikut:

```
=> Memberikan Izin Modifikasi Bot Tele
# chmod 777 bot-tele.sh
```

Adapun hasil konfigurasinya pada linux ubuntu server adalah sebagai berikut:



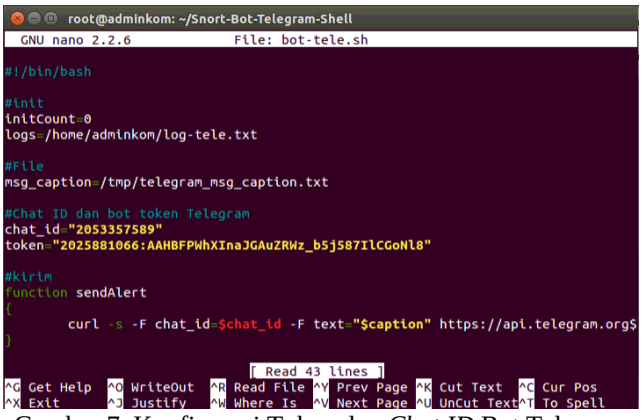
Gambar 6. Perubahan Izin Modifikasi *Bash Shell Snort* Bot Telegram

Berdasarkan pada gambar 6 di atas, selanjutnya adalah konfigurasi isi direktori bot-tele.sh dengan script di bawah ini:

```
=> Membuka Direktori Bot-Tele
# nano bot-tele.sh
Tambahkan dan mofikasi script bot tele menyesuaikan token dan chat ID bot telegram yang telah dikonfigurasi pada gambar 4. Dan 4.1.
#!/bin/bash
#init
initCount=0
```

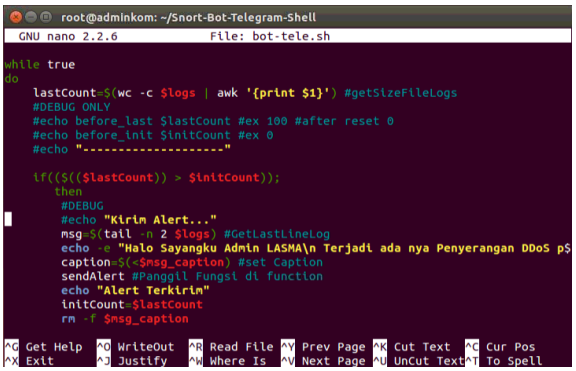
```
logs=/home/ghost666/log-tele.txt
#File
msg_caption=/tmp/telegram_msg_caption.txt
#Chat ID dan bot token Telegram
chat_id="2053357589"
token="2025881066:AAHBFWhXInaJGAuZRWz_b5j587IlCGoNl8"
function sendAlert{
      curl -s -F chat_id=$chat_id -F text="$caption"
https://api.telegram.org/bot$token/sendMessage #> /dev/null 2&>1
}while true
do
   lastCount=$(wc -c $logs | awk '{print $1}') #getSizeFileLogs
   #DEBUG ONLY
   #echo before_last $lastCount #ex 100 #after reset 0
   #echo before_init $initCount #ex 0
   #echo "--------------------"
   if(($(($lastCount)) > $initCount));
     then
      #DEBUG
      #echo "Kirim Alert..."
      msg=$(tail -n 2 $logs) #GetLastLineLog
      echo -e "Halo Sayangku Admin Lasma\n Terjadi ada nya Penyerangan
DDoS pada Server!!!\n\nServer Time : $(date +"%d %b %Y %T")\n\n"$msg >
$msg_caption #set Caption / Pesan
      caption=$(<$msg_caption) #set Caption
      sendAlert #Panggil Fungsi di function
      echo "Alert Terkirim"
      initCount=$lastCount
      rm -f $msg_caption
      sleep 1
   fi
   sleep 2 #delay if Not Indication
done
```

Adapun hasil konfigurasi token *API Key* dan chat ID dari bot telegram yang menghubungkan *bash shell* sebagai media monitoring serangan DDoS pada ubuntu server dapat dilihat pada gambar sebagai berikut:



Gambar 7. Konfigurasi Token dan *Chat ID* Bot Telegram

Setelah token *API Key* dan chat ID bot telegram dimasukan, berikut adalah *script* tampilan pesan yang akan dikirimkan *snort* IDS apabila terjadi serangan DDoS pada server ubuntu:



Gambar 8. Konfigurasi Pesan Monitoring Serangan DDoS

Berdasarkan pada gambar 8 di atas, pesan monitoring *snort* IDS dari serangan DDoS dapat dimodifikasi sesuai keiinginan.

2.3 Skenario Pengujian

Berikut adalah skenario pengujian dalam penelitian ini yang terlihat seperti ditunjukkan pada tabel berikut:

Tabel 2. Skenario Pengujian

Jenis Serangan	Jenis Server	Aplikasi	IP Tujuan Serangan	Notifikasi
ICMP *Flooding*	Ubuntu	CMD Bat	100.100.10.1	Bot Telegram
TCP *Flooding*	Ubuntu	LOIC	100.100.10.1	Bot Telegram
UDP *Flooding*	Ubuntu	LOIC	100.100.10.1	Bot Telegram

3. HASIL DAN PEMBAHASAN

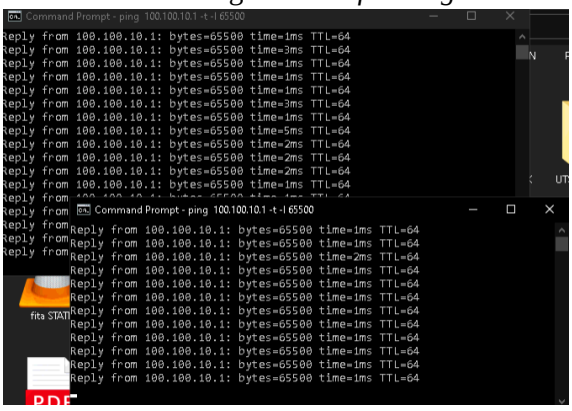
Pada pembahasan ini dilakukan pengujian dengan melakukan penyerangan dari serangan DoS (*Distributed Denial of Service*) yaitu ICMP *flooding*, TCP *flooding* serta UDP *flooding* yang bekerja dengan mengirimkan paket ICMP, TCP dan UDP dengan jumlah yang sangat besar ke server sehingga dapat mampu membuat sebuah server *crash* atau server tidak bekerja dengan baik. Hasil ini akan menunjukkan serangan ICMP, TCP dan UDP *Flooding* dengan CMD bat dan LOIC dalam menyerang server yang terdapat *Snort* IDS.

3.1. Pengujian Sistem

Dalam pengujian ini akan menggunakan serangan DDoS dengan jenis serangan ICMP *flooding* menggunakan *bat* cmd sedangkan TCP *flooding* dan UDP *flooding* menggunakan aplikasi LOIC.

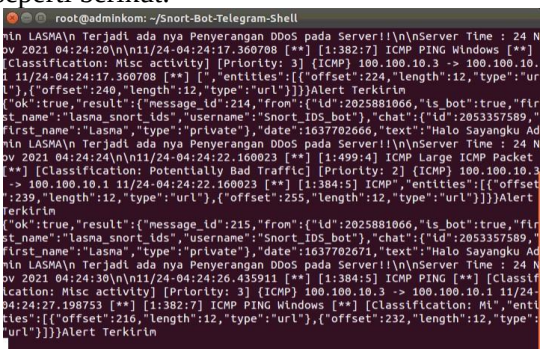
1. Pengujian Serangan ICMP *Flooding*

   Serangan ICMP *flooding* dengan bat cmd pada protokol *ping* bertujuan untuk membanjiri lalu lintas jaringan. Berikut adalah bentuk serangan ICMP *flooding*:



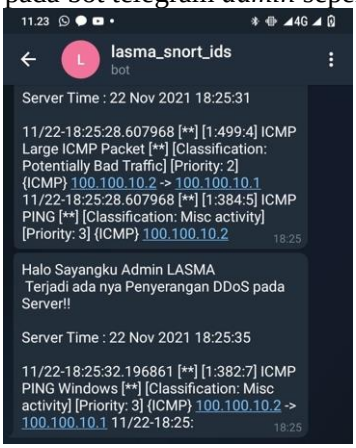
Gambar 9. Serangan ICMP *Flooding*

Adapun fungsi *snort* IDS akan mengirimkan pesan serangan ICMP *flooding* secara terus menerus melalui *bash shell snort* seperti berikut:



Gambar 10. *Bash Shell Snort* Mengirimkan Pesan *Alert* ICMP *Flooding*

Berdasarkan pada *alert bash shell* pada gambar 10, maka *bash shell snort* linux ubuntu akan mengirimkan notifikasi serangan pada bot telegram *admin* seperti pada gambar di bawah ini:



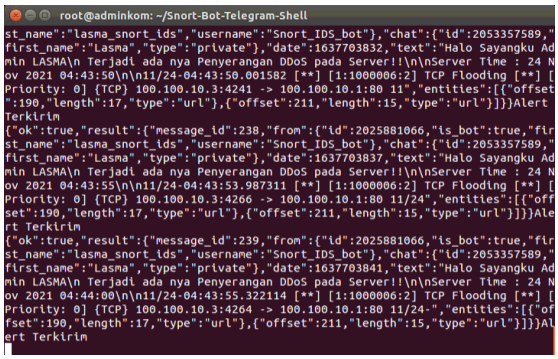
Gambar 11. Notifikasi Serangan DDoS ICMP *Flooding*

2. Pengujian Serangan TCP *Flooding*

Sebuah serangan yang mengarah pada protokol TCP memanfaatkan koneksi protokol yang terhubung dengan server dengan membanjirinya (*flooding*) dan mengirimkan banyaknya paket kedalam TCP, lokasi *buffer* akan mengalami kepenuhan dan mengakibatkan server menjadi tidak bekerja dengan baik. Selanjutnya dilakukan percobaan serangan DDoS TCP *flooding* menggunakan aplikasi LOIC dengan serangan sebagai berikut:

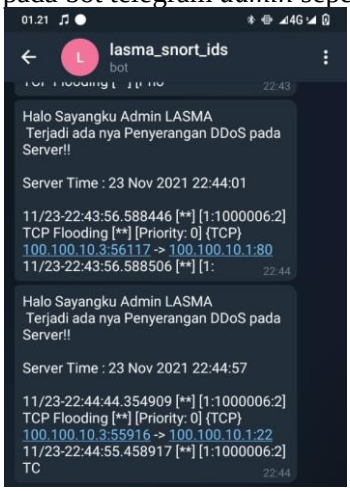
Gambar 12. Pengujian Serangan LOIC TCP *Flooding*

Adapun fungsi *snort* IDS akan mengirimkan pesan serangan TCP *flooding* secara terus menerus melalui *bash shell snort* seperti berikut:



Gambar 13. *Bash Shell Snort* Mengirimkan Pesan *Alert* TCP *Flooding*

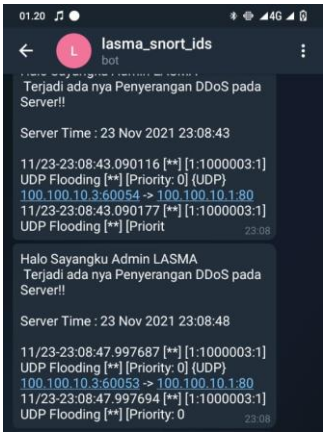
Berdasarkan pada *alert bash shell* pada gambar 11, maka *bash shell snort* linux ubuntu akan mengirimkan notifikasi serangan pada bot telegram *admin* seperti pada gambar di bawah ini:



Gambar 14. Notifikasi Serangan DDoS TCP *Flooding*

3. Pengujian Serangan UDP *Flooding*

   Serangan UDP *flooding* dilakukan dengan cara yang sama seperti TCP *flooding*, sehingga *bash shell snort* linux ubuntu akan mengirimkan notifikasi serangan pada bot telegram *admin* seperti pada gambar di bawah ini:



Gambar 15. Notifikasi Serangan DDoS UDP *Flooding*

Berdasarkan dari hasil pengujian, notifikasi serangan DDoS ICMP *flooding*, TCP *flooding* dan UDP *flooding* berhasil masuk pada bot telegram *admin*, sehingga monitoring keamanan server ubuntu dari serangan DDoS berhasil dilakukan

4. KESIMPULAN

Berdasarkan dari hasil pengujian, adapun kesimpulan yang dapat diberikan pada penelitian ini adalah sebagai berikut:

1. Berdasarkan hasil pengujian, sistem *snort* IDS mampu bekerja efektif sebagai pemberi peringatan dini dari serangan DDoS *flooding* diantaranya ICMP *flooding*, UDP *flooding* dan TCP *flooding* terhadap jaringan server.
2. Penerapan *bash shell snort* mampu merekam serangan DDoS *flooding* diantaranya ICMP *flooding*, UDP *flooding* dan TCP *flooding* kemudian meneruskan *alert* serangan DDoS dengan notifikasi melalui bot telegram secara *realtime* kepada *admin*.

REFERENSI

[1] W. W. Purba and R. Efendi, "Perancangan dan analisis sistem keamanan jaringan komputer menggunakan SNORT," *Aiti*, vol. 17, no. 2, pp. 143–158, 2021, doi: 10.24246/aiti.v17i2.143-158.

[2] S. A. Valianta, T. Salim, and D. Stiawan, "Identifikasi Serangan Port Scanning dengan Metode String Matching," *Annual Research Seminar (ARS)*, vol. 2, no. Fakultas Ilmu Komputer Unsri, pp. 466–471, 2016.

[3] E. K. Dewi, "Analisis Log Snort Menggunakan Network Forensic," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 2, no. 2, pp. 72–79, 2017, doi: 10.29100/jipi.v2i2.370.

[4] A. D. Mulyanto, "Pemanfaatan Bot Telegram Untuk Media Informasi Penelitian," *Matics*, vol. 12, no. 1, p. 49, 2020, doi: 10.18860/mat.v12i1.8847.

[5] A. L. Ginting, J. Napitupulu, and J. Jamaluddin, "Sistem Monitoring Pendeteksian Penyusup Menggunakan Snort pada Jaringan Komputer Fakultas Ekonomi Universitas Methodist Indonesia," pp. 83–87, 2018, doi: 10.31227/osf.io/w5gt7.

[6] Sutarti, A. P. Pancaro, and F. I. Saputra, "Implementasi IDS (Intrusion Detection System) Pada Sistem Keamanan Jaringan SMAN 1 Cikeusal," *Jurnal PROSISKO*, vol. 5, no. 1, pp. 1–8, 2018.

[7] Mitra Unik, S. Soni, and Randra Aguslan Pratama, "Penerapan Metode Htb Dan Diffserv Guna Peningkatan Qos Pada Layanan Video Streaming," *Jurnal Fasilkom*, vol. 9, no. 3, pp. 35–40, 2019, doi: 10.37859/jf.v9i3.1665.