PENERAPAN METODE PORT KNOCKING DAN INTRUSION DETECTION (IDS) SYSTEM PADA VIRTUAL PRIVATE SERVER (VPS)

Fajar[1], Naikson Fandier Saragih[2], Indra Kelana Jaya[3]

[123]Universitas Methodist Indonesia

Jl. Hang Tuah No. 8, Medan, Sumatera Utara
E-mail :saragihnaikson@gmail.com [2]

ABSTRACT

Virtual Private Server security needs to be guaranteed by implementing the necessary security methods according to possible attack threats. Increased security to guarantee or reduce the risk of cyber attacks that occur on servers, one of which is DDoS (Distributed Denial Of Service). The Port Knocking method can set open and close logic ports in computer networks with knock commands (knocking) using Knockd and Intrusion Detection System using Snort which can monitor network packet traffic. The detection system reads suspicious activity from network computer systems and detects network computer activity that may carry out attacks and further information is conveyed through notifications using Telegram bots. Trial attacks 9 times using the LOIC system 100 percent successfully detect attacks with notifications sent to telegrams. Trial tapping port 21 (ftp), 22 (ssh), 23 (telnet), 80 (http) using knockd 3 times worked 100 percent with a duration of 5 seconds.

Keywords: Port Knocking, Intrusion Detection System, VPS, DDos, Knockd, Snort.

I. PENDAHULUAN

Seiringnya perkembangan zaman teknologi informasi semakin juga berkembangnya kejahatan yang dilakukan para hacker atau cracker pada dunia internet sehingga semakin pula dibutuhkan keamanan-keamanan pada sebuah sistem yang ada pada jaringan komputer, pada penelitian sebelumnya sudah banyak penelitian dengan melakukan pengamanan-pengamanan pada sistem, salah satu keamanan yang dapat dilakukan dengan cara memonitoring jaringan terhadap serangan-serangan yang muncul pada jaringan komputer, termasuk serangan DDoS (Distributed Denial of Service) serangan ini membuat sumber daya jaringan tidak tersedia bagi user yang dituju dengan mengganggu layanan host yang terhubung ke internet untuk sementara atau tanpa batas, maka dari itu keamanan yang dilakukan adalah dengan mendeteksi adanya serangan itu melalui metode keamanan IDS (Intrusion Detection System) pada sistem operasi Ubuntu, monitoring yang dilakukan adalah dengan memberi notifikasi atau pemberitahuan adanya serangan melalui Bot Telegram (Nainggolan et al., 2022), selain itu juga sudah dilakukan penelitian metode keamanan pada Mikrotik Router yang sama halnya memonitoring jaringan melalui Bot Telegram, hanya saja bedanya terdapat pada wadah implementasi yang dilakukan, pada penelitian sebelumnya ini dilakukan pada Mikrotik (Simanjuntak et al., 2022), dari referensi penelitian sebelumnya tersebut maka dalam penelitian ini dilakukan penerapan dua metode keamanan yang bergabung dalam satu sistem operasi Ubuntu yang dimana diterapkan metode keamanan IDS (Intrusion Detection System) yang dilakukan juga pemberian Bot Telegram sebagai penghubung notifikasi serangan dan Port Knocking yang dimana Port Knocking dapat mengatur buka tutup Port dalam jaringan komputer, pada penelitian sebelumnya juga Port Knocking hanya dilakukan pada port 22 (ssh) saja, namun pada penilitian ini dicoba menambah

autentikasi port 21 (ftp), 22 (ssh), 23 (telnet), 80 (http). Pada umumnya dasar keamanan server hanya menggunakan firewall default yang dapat memberikan perlindungan pada server, namun keamanan firewall saja tidak mampu menjadikan server terjamin keamanannya, maka dari itu diperlukan peningkatan keamanan untuk menjamin atau mengurangi resiko peluang terjadinya serangan cyber yang terjadi pada server, serangan yang dimaksud adalah DDoS (Distributed Denial Of Service) yang merupakan penyerangan terhadap server yang dapat menyebabkan lalu lintas jaringan atau sistem menjadi terganggu (Geges & Wibisono, n.d.) Ada beberapa cara untuk meningkatkan keamanan dalam mencegah serangan cyber dan mengatur hak akses dalam server, yang pertama adalah menggunakan port knocking dan yang kedua adalah menggunakan intrusion detection system, menurut (Mulyanto et al., 2021). Port Knocking merupakan metode keamanan jaringan yang dapat mengatur buka tutup port logic dalam jaringan komputer dengan perintah knock (pengetukan). Menurut (Suwaryo et al., n.d.). Intrusion Detection System merupakan metode keamanan jaringan yang dapat memantau lalu lintas paket jaringan, membaca aktivitas mencurigakan dari sistem komputer jaringan dan mendeteksi aktivitas komputer jaringan yang mungkin melakukan serangan atau eksperimen untuk menganalisis ada tidaknya penyusupan. Untuk menerapkan Intrusion Detection System, penelitian ini menggunakan snort yang mampu dalam membentuk logging paket-paket dan analisis trafik- trafik secara real time dalam jaringan yang berbasis TCP/IP (Efendi & Kusuma, n.d.). Karena keterbatasan mengakses komputer server fisik, ada acara lain untuk memanfaatkan virtualisasi server yang secara real memalui akses remote yaitu Virtual

Private Server (VPS) yang merupakan server dengan teknologi virtualisasi. Virtualisasi ini memungkinkan pengguna untuk menginstal sistem operasi dan perangkat lunak lainnya sesuai kebutuhan pengguna dan Virtual Private Server memiliki efisiensi yang sangat berpengaruh dalam biaya penyewaan layanan yang sangat murah dibandingkan membangun server fisik mandiri, maupun juga dalam pengaturan yang cepat dan kemudahan dalam pengembangan update sistem operasi pada Virtual Private Server.

II. METODOLOGI

a. Metode

1. Port Knocking

Port Knocking adalah metode otentikasi pengguna berbasis firewall untuk berkomunikasi melalui port terbuka ataupun tertutup. Metode port knocking menggunakan sistem otentikasi yang dirancang khusus untuk koneksi client dan server. port knocking digunakan untuk mengotentikasi pengiriman pesan atau data. Ini menyebabkan informasi yang dikirim dikodekan (dan berpotensi dienkripsi) ke dalam serangkaian nomor port. Dengan menggunakan metode port knocking untuk akuisisi akses jarak jauh dengan menonaktifkan port terbuka untuk perlindungan server dari pemindaian port. Pengguna memiliki akses ke akses port, menutup port dan membuka port dapat dilakukan dengan perintah ketukan yang sudah ditentukan, dengan user diberikan akses untuk mengakses port dan diakhiri dengan menutup port agar firewall menghapus rule yang ditulis sebelumnya untuk membuka port.

2. Intrusion Detection System

Intrusion Detection System (IDS) adalah sebuah sistem yang dapat mendeteksi aktivitas mencurigakan dalam sistem atau jaringan. Ketika aktivitas mencurigakan ditemukan pada lalu lintas jaringan, IDS akan memberi peringatan ke sistem, atau administrator jaringan. IDS pada dasarnya adalah sebuah sistem dapat menganalisis data deteksi, perekaman (log) dan menghentikan penyalahgunaan dan penyerangan. Secara umum jenis-jenis IDS antara lain (Ulfa, n.d.):

a. NIDS (Network Intrusion Detection System)

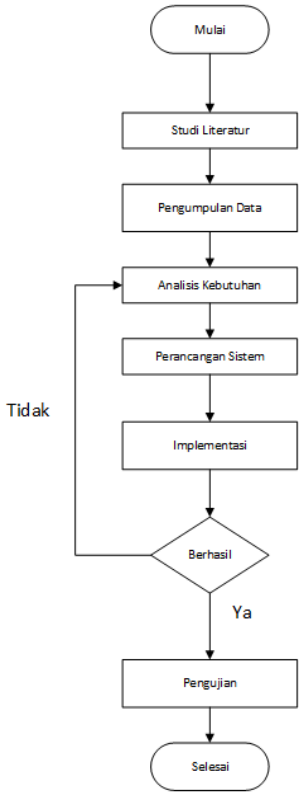
NIDS adalah sebuah strategi cara yang efektif untuk memeriksa lalu lintas masuk/keluar atau lalu lintas antar host antara segmen jaringan lokal. NIDS biasanya dikembangkan depan dan belakang Firewall dan gateway VPN yang dibuat khusus untuk efektivitas perangkat keamanan dan memperkuat keamanan jaringan.

b. HIDS (Host Intrusion Detection System)

HIDS melakukan monitoring pada perangkat komputer tertentu yang ada dalam jaringan. HIDS akan memantau kejadian seperti kesalahan login berkali-kali dan melakukan pengecekan pada file.

b. Alur Penelitian

Untuk melakukan proses penelitian sehingga mendapatkan hasil yang di inginkan maka peneliti membuat alur penelitian seperti yang ditunjukan pada Gambar 1



Gambar 1 Alur Penelitian

III. HASIL DAN PEMBAHASAN

3.1 Implementasi Sistem

1. Tahapan ini dilakukan dengan menyewa jasa layanan VPS dari provider herza.id. Sistem yang disewa ialah VPS KVM dengan sistem operasi Ubuntu 20.04. Adapun spesifikasi VPS yang di sewa dapat dilihat pada Tabel 1

Tabel 1 Spesifikasi VPS KVM

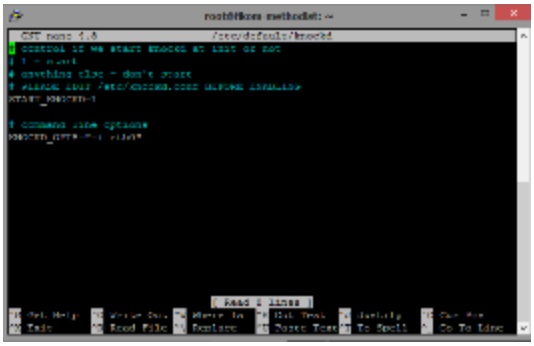
CPU yg digunakan	Core 4
Ram	4 GB
Harddisk	80 GB
Bandwidth	Unlimited
Sistem Operasi	Ubuntu 20.04

Selanjutnya dilakukan penerapan metode Port Knocking dan IDS dan dilakukan juga pengujian.

a. Implementasi Metode Port Knocking

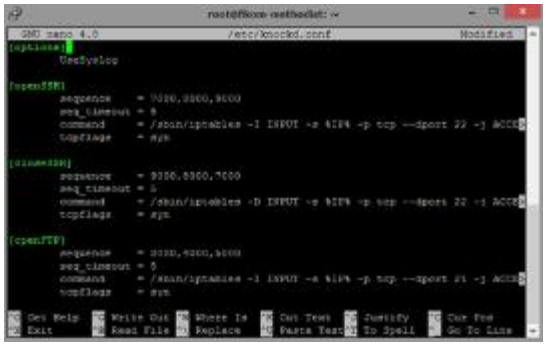
Implementasi metode port knocking dimulai dari instalasi knockd dengan perintah apt install knock lalu pembukaan pada port 21, 22, 23, 80, dimana port 21 adalah port FTP yaitu File Transfer Protocol, port 22 adalah port SSH yaitu Secure Shell, Port 23 adalah port telnet dan port 80 adalah port HTTP yaitu port web server, lalu mengkonfigurasi file etc/knockd.default lalu mengubah start_knockd=0, angka 0

diubah menjadi angka 1 dan knockd_opts=1 eth0" pada etho mengikuti interface jaringan pada virtual private server.



Gambar 2 Konfigurasi /etc/default/knockd

Lalu membuat konfigurasi rule otentikasi pada etc/knockd.conf

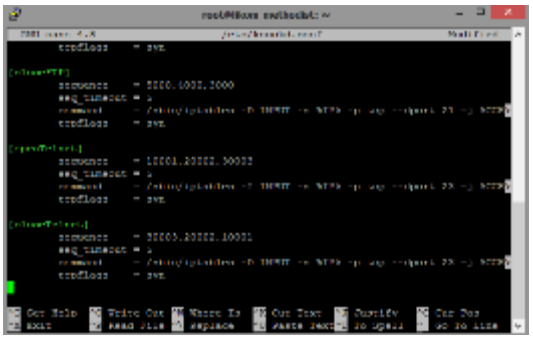


Gambar 3 Konfigurasi /etc/knockd.conf

Terdapat beberapa bagian yaitu options, openSSH, closeSSH dan openFTP, pada option dapat digunakan untuk memberikan log aktivitas IP yang berusaha untuk masuk ke dalam knock, berikutnya pada openSSH terdapat "sequence = 7000,8000,9000" yang merupakan kode ketukan sebagai autentikasi untuk membuka port pada server SSH, berikutnya terdapat "seq_timeout = 5" artinya waktu dalam proses berjalannya autentikasi, berikutnya terdapat "command = sbin/iptables -I INPUT -s %IP% -p tcp -dport 22 -j ACCEPT" merupakan perintah yang akan dilakukan apabila ketukan yang dilakukan benar dan akan diterima.

Pada closeSSH terdapat terdapat "sequence = 9000,8000,7000" merupakan ketukan autentikasi untuk menutup pengaksesan port SSH, berikutnya terdapat "seq_timeout = 5" artinya waktu dalam proses berjalannya autentikasi, berikutnya terdapat "command = sbin/iptables -D INPUT -s %IP% -p tcp -dport 22 -j ACCEPT" merupakan perintah untuk menutup kembali akses ssh port 22 dengan ketukan autentikasi yang telah ditentukan,

Pada bagian openFTP terdapat "sequence = 3000,4000,5000" merupakan ketukan autentikasi untuk membuka akses port FTP, berikutnya "seq_timeout = 5" artinya waktu dalam proses berjalannya autentikasi, berikutnya "command = sbin/iptables -I INPUT -s %IP% -p tcp -dport 21 -j ACCEPT" merupakan perintah untuk membuka akses port FTP, untuk autentikasi closeFTP, openTelnet dan closeTelnet.



Gambar 4 Konfigurasi /etc/knockd.conf

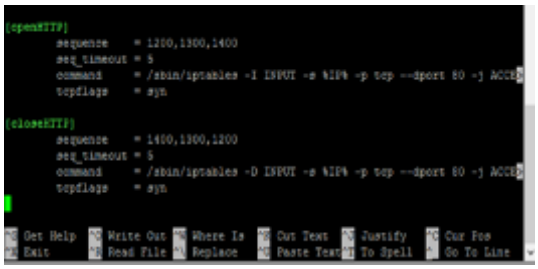
Selanjutnya pada bagian closeFTP terdapat "sequence = 5000,4000,3000" merupakan ketukan autentikasi untuk menutup akses port FTP, berikutnya "seq_timetout = 5" artinya waktu dalam proses berjalannya autentikasi, berikutnya "command = sbin/iptables -D INPUT -s %IP% -p tcp -dport 21 -j ACCEPT" merupakan perintah untuk menutup akses port FTP.

Berikutnya pada openTelnet terdapat "sequence = 10003,20002,30003" yang merupakan kode ketukan sebagai autentikasi untuk membuka port pada telnet, berikutnya terdapat "seq_timeout = 5" artinya waktu dalam proses berjalannya autentikasi, berikutnya terdapat "command = sbin/iptables -I INPUT -s %IP% -p tcp -dport 23 -j ACCEPT" merupakan perintah yang akan dilakukan apabila ketukan yang dilakukan benar dan akan diterima.

Pada closeTelnet terdapat "sequence = 30003,20002,10001" merupakan ketukan autentikasi untuk menutup pengaksesan port telnet, berikutnya terdapat "seq_timeout = 5" artinya waktu dalam proses berjalannya autentikasi, berikutnya terdapat "command = sbin/iptables -D INPUT -s %IP% -p tcp -dport 23 -j ACCEPT" merupakan perintah untuk menutup kembali akses telnet port 23 dengan ketukan autentikasi yang telah ditentukan,

Selanjutnya Pada bagian openHTTP terdapat "sequence = 1200,1300,1400" merupakan ketukan autentikasi untuk membuka pengaksesan port HTTP, berikutnya terdapat "seq_timeout = 5" artinya waktu dalam proses berjalannya autentikasi, berikutnya terdapat "command = sbin/iptables -I INPUT -s %IP% -p tcp -dport 80 -j ACCEPT" merupakan perintah untuk membuka akses HTTP port 80 dengan ketukan autentikasi yang telah ditentukan,

Selanjutnya Pada bagian closeHTTP terdapat "sequence = 1400,1300,1200" merupakan ketukan autentikasi untuk menutup pengaksesan port HTTP, berikutnya terdapat "seq_timeout = 5" artinya waktu dalam proses berjalannya autentikasi, berikutnya terdapat "command = sbin/iptables -D INPUT -s %IP% -p tcp -dport 80 -j ACCEPT" merupakan perintah untuk menutup akses HTTP port 80 dengan ketukan autentikasi yang telah ditentukan, untuk melihat hasil dari konfigurasi ketukan open HTTP dan close HTTP



Gambar 5 Konfigurasi /etc/knockd.conf

b. Metode Intrusion Detection System

Selanjutnya dilakukan penerapan metode IDS dengan menginstall Snort, Snort merupakan perangkat lunak yang umumnya diinstal pada system operasi linux yang berfungsi sebagai alat pendeteksi serangan yang masuk terhadap jaringan interface pada Snort, setelah melakukan instalasi maka dilakukan konfigurasi :

a. Menginstal Paket Snort

```
# apt-get install snort
```

b. Restart paket Snort

```
# etc/init.d/snort restart
```

c. Membuat penyimpanan untuk Snort

```
# mkdir /etc/snort
# mkdir /etc/snort/rules
```

d. Membuat file pada rules

```
# touch /etc/snort/rules/local.rules
```

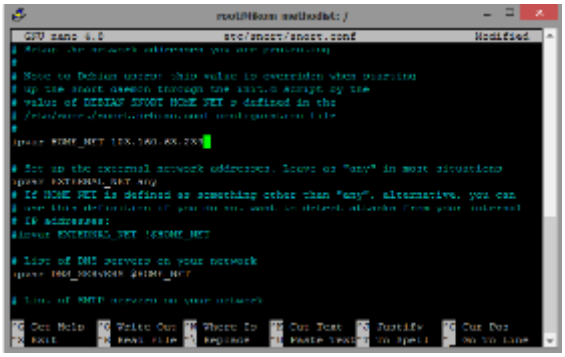
e. Membuat ruang penyimpanan log snort

```
# mkdir /var/log/snort
```

f. Perintah untuk memberikan hak akses

```
# chmod -R 777 /etc/Snort
# chmod -R 777 /var/log/Snort
# chmod -R 777 /etc/Snort/so_rules
```

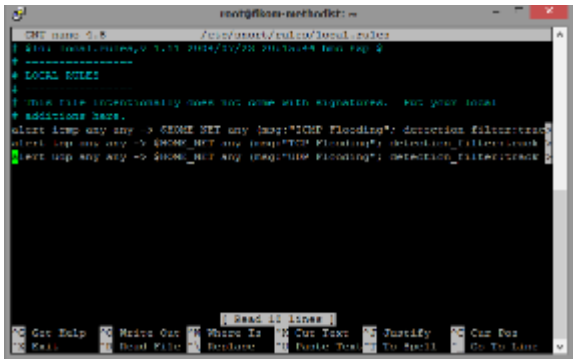
Setelah itu dilakukan konfigurasi IP Address interface pada Snort, dengan perintah `nano etc/snort/snort.conf` lalu mengubah "ipvar HOME_NET" yang sebelumnya dalam status *any* lalu diubah menjadi *IP Address* VPS



Gambar 6 Konfigurasi etc/snort/snort.conf

Setelah konfigurasi *IP Address* snort sudah dilakukan selanjutnya melakukan penambahan rules pesan peringatan serangan *DDoS* dengan *script*

```
alert icmp any any -> $HOME_NET any (msg:"ICMP Flooding"; detection_filter:track by_src, count 30, seconds 60; sid:10000013; rev:2;)
alert tcp any any -> $HOME_NET any (msg:"TCP Flooding"; detection_filter:track by_src, count 30, seconds 60; sid:1000006; rev:2;)
alert udp any any -> $HOME_NET any (msg:"UDP Flooding"; detection_filter:track by_src, count 30, seconds 60; sid:1000003; rev:1;).
```

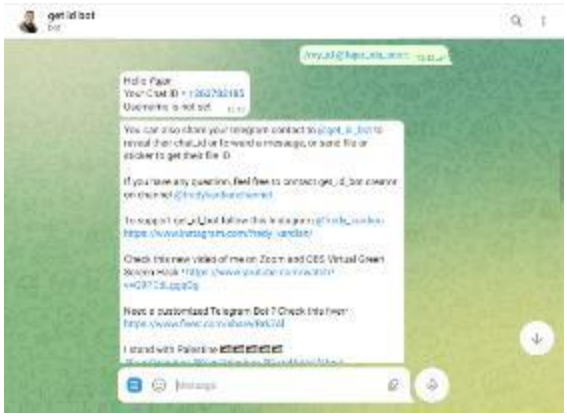


Gambar 7 Konfigurasi etc/snort/rules/local.rules

Setelah melakukan konfigurasi rules pesan peringatan serangan DDoS, maka dilakukan konfigurasi atau penerapan tambahan yaitu dengan membuat notifikasi serangan masuk kepada Bot Telegram, Langkah pertama adalah dengan melakukan instalasi Bot Telegram pada channel Bot Father lalu melakukan identifikasi Token API

Gambar 8 Konfigurasi Token ID Bot

Setelah mendapatkan Token ID atau Token API lalu diidentifikasi juga Chat ID akun Telegram yang kita pakai melalui channel get id bot.



Gambar 9 Konfigurasi Chat ID

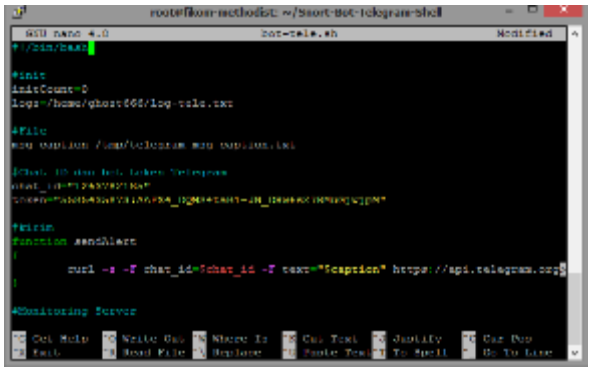
Selanjutnya dilalukan konfigurasi Bash Shell Bot Telegram yang berfungsi sebagai penghubung antara Snort dengan Bot Telegram, dengan mengetik perintah :

`# git clone https://github.com/gagaltotal/Snort-Bot-Telegram-Shell`

Selanjutnya melakukan perintah untuk memberikan hak akses penuh untuk memodifikasi script penghubung Snort *virtual private server* dengan script `# chmod 777 bot-tele.sh`, selanjutnya adalah konfigurasi isi direktori bot-tele.sh dengan perintah membuka direktori terlebih dahulu `#nano bot-tele.sh` dan memodifikasi dengan script dibawah ini :

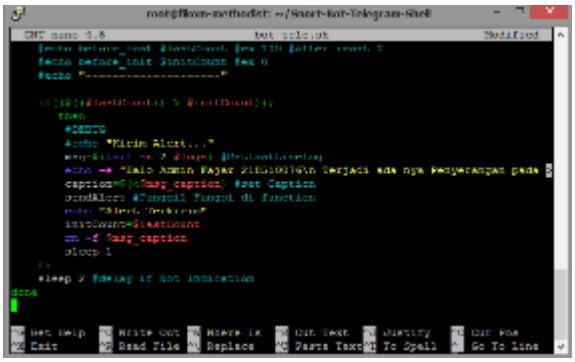
```
#!/bin/bash
#init
initCount=0
logs=/home/fajar/log-tele.txt
#File
msg_caption=/tmp/telegram_msg_captio
n.txt
#Chat ID dan bot token Telegram
chat_id="1263782185"
token="5585435873:AAFX4_DQMS4taB1-
JN_DRw6sZ7RMuGjwjpM"
#kirim
function sendAlert
{
 curl -s -F chat_id=$chat_id -F
text="$caption"
https://api.telegram.org/bot$token/s
endMessage #> /dev/null 2&>1
}
#Monitoring Server
while true
do
 lastCount=$(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 Admin Fajar 218510076\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
```

Berdasarkan pada *script* "bot-tele.sh" diperlukan modifikasi atau penginputan Chat ID dan Token Bot Telegram yang sudah didapat pada konfigurasi dan implementasi dalam membuat Bot baru yang sebelumnya sudah dilakukan, maka dapat dilihat pada Gambar dibawah ini Chat ID dan Token Bot Telegram yang sudah didapat sebelumnya diinput ke dalam *script* pada *file* "bot-tele.sh"



Gambar 10 Konfigurasi bot-tele.sh

Selanjutnya adalah *script* bagaimana pesan notifikasi apabila ada serangan *DDos Flooding* yang akan dikirim kepada Bot Telegram yang sudah terhubung juga dengan snort *VPS.*



Gambar 11 Konfigurasi bot-tele.sh

c. Pengujian

Setelah melakukan penerapan metode Port Knocking dan IDS maka tahap selanjutnya dilakukan Pengujian.

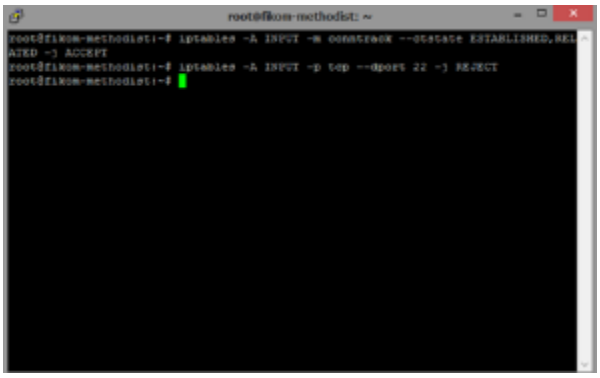
1. Port Knocking

Setelah melakukan implementasi dan konfigurasi pada metode Port Knocking maka tahap selanjutnya adalah dengan menguji bagaimana metode Port Knocking atau knockd bekerja dalam mengatur atau membuka menutup port logic pada virtual private server, penliti melakukan pengujian pada beberapa hari dalam 1 minggu yang dapat dilihat data pada tabel 2

Tabel 2 Pengujian Port Knocking

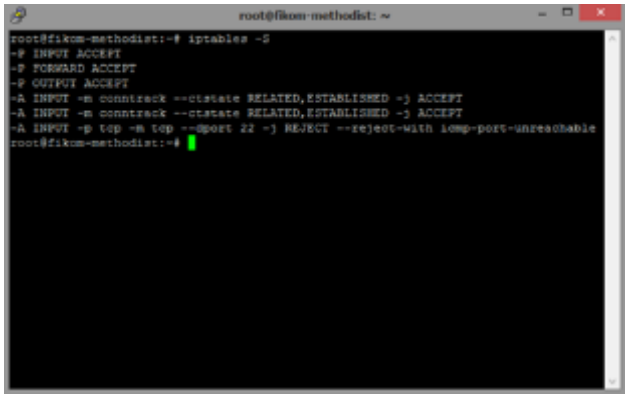
Tanggal Pengujian	Port 21	Port 22	Port 23	Port 80	Keterangan Hitting / Ketukan
2 Sep 2022	Tutup dan Buka	Tutup dan Buka	Tutup dan Buka	Tutup dan Buka	Normal 5 Detik
5 Sep 2022	Tutup dan Buka	Tutup dan Buka	Tutup dan Buka	Tutup dan Buka	Normal 5 Detik
7 Sep 2022	Tutup dan Buka	Tutup dan Buka	Tutup dan Buka	Tutup dan Buka	Normal 5 Detik

dilakukan perintah iptables penutupan port 22 SSH, dengan perintah iptables -A INPUT -p tcp –dport 22 -j REJECT, dari kedua perintah tersebut untuk menutup port 22 SSH dapat dilihat pada Gambar 12.



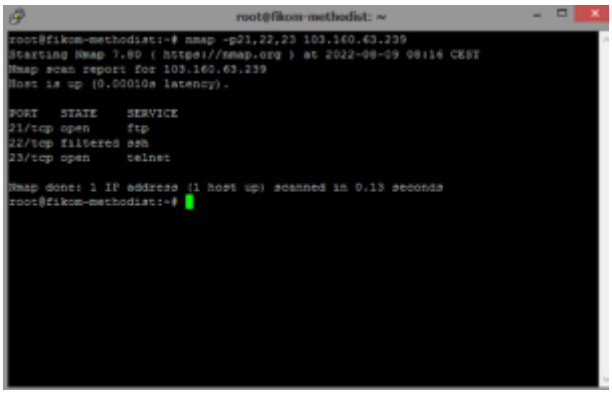
Gambar 12 Perintah Iptables Tutup Port SSH

Berdasarkan pada Gambar 12 dilakukan perintah penutupan port 22 SSH maka dapat dilihat hasil penambahan rules penutupan port 22 SSH pada perintah iptables -S dapat dilihat pada Gambar 13



Gambar 13 Perintah Tutup SSH pada Iptables

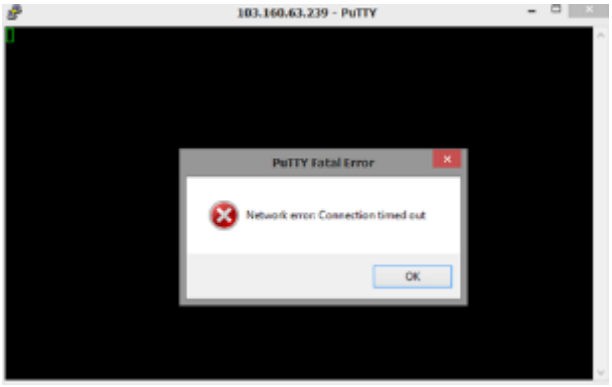
Pada Gambar 13 dapat dilihat bahwa perintah tutup port 22 SSH sudah tersimpan pada iptables dan status port 22 SSH dengan perintah nmap -p21,22,23 103.160.63.239 sudah menjadi filtered yang dapat dilihat pada Gambar 14



Gambar 14 Status SSH Filtered

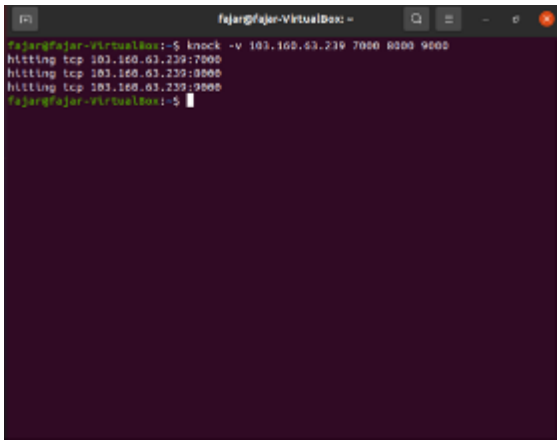
Berdasarkan pada Gambar 14 dapat dilihat bahwa status port 22 SSH dalam keadaan filtered yang berarti tertutup dan dilakukan pengujian untuk terhubung kepada SSH server dengan menggunakan aplikasi putty untuk mengakses SSH melalui IP Address virtual private server

yang dapat dilihat pada Gambar 15 bahwa untuk terhubung kepada port 22 SSH mengalami connection timed out yang artinya adalah gagal terkoneksi akibat penutupan port 22 SSH pada rules iptables yang sudah dilakukan.



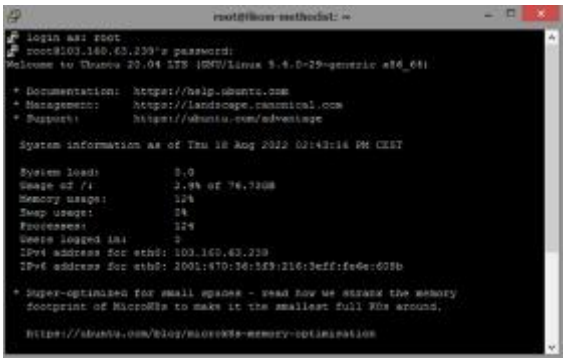
Gambar 15 Koneksi SSH Error

Selanjutnya dilakukan perintah untuk mencoba membuka kembali port 22 SSH dengan perintah knock daripada klien untuk membuka port 22 SSH pada virtual private server yang dimana sudah diterapkan kode autentikasi pada knockd.conf yang dimana kode tersebut atau kode autentikasi ketukan untuk membuka port 22 SSH yang dapat dilihat pada Gambar 16.



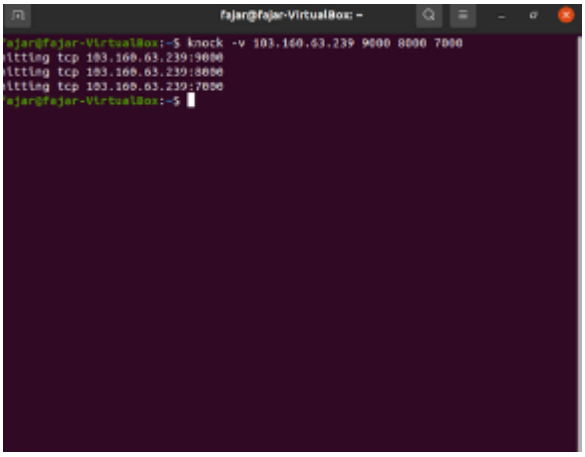
Gambar 16 Knock Membuka Port SSH

Berdasarkan pada Gambar 16 dapat dilihat bahwa pada klien mencoba untuk membuka kembali port 22 SSH dengan kode autentikasi yang sudah diatur pada knockd.conf dengan perintah knock -v 103.160.63.239 7000 8000 9000, pada bagian knock artinya adalah perintah untuk pengetukan port, pada bagian IP Address 103.160.63.239 artinya adalah IP Address target atau virtual private server, berikutnya pada bagian 7000 8000 9000 artinya adalah kode autentikasi untuk membuka port 22 SSH, selanjutnya dapat dilihat hasil hitting tcp pengetukan urutan 7000 8000 9000 untuk membuka port 22 SSH pada virtual private server, untuk melihat hasil pengetukan pembukaan port 22 SSH dapat dilihat pada Gambar 17.



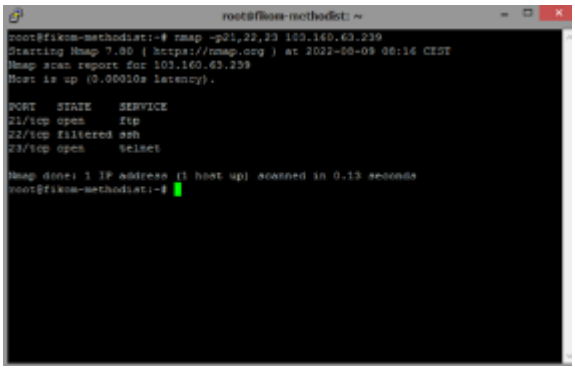
Gambar 17 Konektifitas SSH Terhubung

Dapat dilihat pada Gambar 17 dengan menggunakan putty untuk mencoba terhubung ke port SSH sudah berhasil, yang berarti pengetukan untuk membuka port 22 SSH sudah berhasil dilakukan.
Selanjutnya dilakukan lagi klien untuk mencoba pengetukan untuk menutup port 22 SSH yang dapat dilihat pada Gambar 18.



Gambar 18 Knock Tutup Port SSH

Berdasarkan pada Gambar 18 dapat dilihat klien melakukan pengetukan kepada virtual private server untuk menutup kembali port 22 SSH dengan perintah knock -v 103.160.63.239 9000 8000 7000, pada bagian knock artinya adalah perintah untuk pengetukan port, pada bagian IP Address 103.160.63.239 artinya adalah IP Address target atau virtual private server, berikutnya pada bagian 9000 8000 7000 artinya adalah kode autentikasi untuk menutup port 22 SSH, selanjutnya dapat dilihat hasil hitting tcp pengetukan urutan 9000 8000 7000 untuk menutup port 22 SSH pada virtual private server, untuk melihat hasil pengetukan penutupan port 22 SSH dapat dilihat pada Gambar 19



Gambar 19 Nmap Port SSH status Filtered

Dapat dilihat pada Gambar 19 port 22 SSH dalam keadaan filtered atau tertutup setelah melakukan pengetukan penutupan port 22 SSH, maka dari itu pengetukan untuk menutup port 22 SSH berhasil dilakukan.

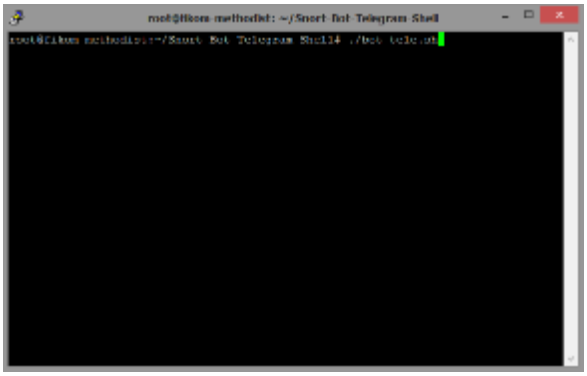
2. Intrusion Detection System

Setelah melakukan implementasi dan konfigurasi pada metode Intrusion Detection System maka tahap selanjutnya adalah dengan menguji bagaimana snort bekerja dalam mendeteksi lalu lintas paket dengan melakukan serangan *DDos Flooding* menggunakan aplikasi LOIC terhadap *virtual private server* dan melihat bagaimana notifikasi atau pemberitahuan apabila adanya serangan *DDos Flooding* terhadap *virtual private server* dan peniliti melakukan pengujian dalam beberapa hari yang dapat dilihat pada Tabel 3

Tabel 3 Pengujian Intrusion Detection System

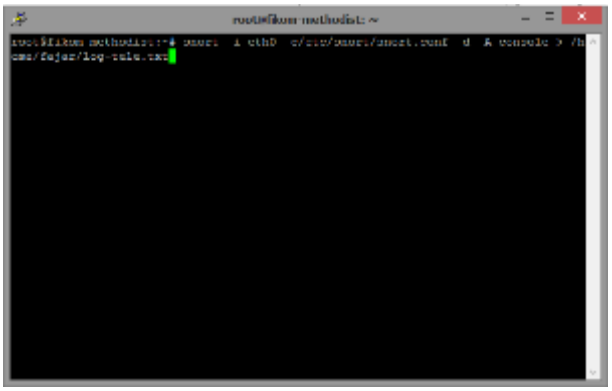
Tanggal Pengujian	Jenis Serangan	Persen CPU VPS	Notifikasi Berhasil ke Telegram
18 Juli 2022	TCP Flooding	15%	Berhasil
18 Juli 2022	ICMP Flooding	10%	Berhasil
18 Juli 2022	UDP Flooding	25%	Berhasil
4 Agustus 2022	TCP Flooding	23%	Berhasil
4 Agustus 2022	UDP Flooding	18%	Berhasil
4 Agustus 2022	ICMP Flooding	28%	Berhasil
25 Agustus 2022	TCP Flooding	21%	Berhasil
25 Agustus 2022	UDP Flooding	26%	Berhasil
25 Agustus 2022	ICMP Flooding	17%	Berhasil

Dan sebelum melakukan pengujian terlebih dahulu mengaktifkan bot-tele.sh yang berfungsi sebagai pengirim notifikasi kepada bot telegram nantinya dan dapat dilihat pada Gambar 20



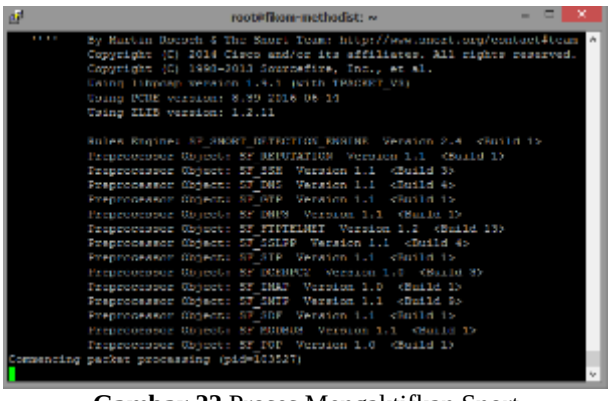
Gambar 20 Program Bot Telegram Aktif

Setelah mengaktifkan bot-tele.sh tahapan selanjutnya adalah mengaktifkan snort untuk memantau lalu lintas paket yang berjalan pada snort virtual private server dan akan terhubung dengan bot-tele.sh sebagai alat pemberian notifikasi atau pemberitahuan di bot telegram, dan dapat dilihat pada Gambar 21



Gambar 21 Mengaktifkan Sistem Snort

Setelah melakukan command untuk mengaktifkan snort, maka tampilan hasil yang menandakan bahwa snort berhasil aktif dan berjalan dengan baik dapat dilihat pada Gambar 22

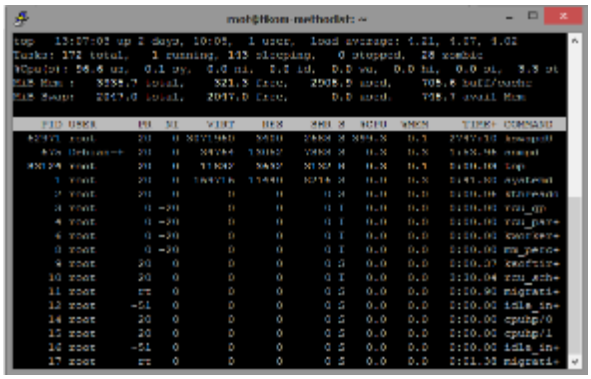


Gambar 22 Proses Mengaktifkan Snort

Dalam pengujian ini akan menggunakan serangan DDoS dengan jenis serangan TCP Flooding menggunakan aplikasi LOIC dengan mengatur method TCP dan memasukkan IP Address yang akan diserang yaitu IP Address virtual private server, pada serangan ini mengarah kepada protocol TCP dengan membanjirinya (Flooding), sebelum melakukan penyerangan DDoS Flooding, pada Gambar 23 dapat dilihat kondisi normal CPU virtual private server.

Gambar 23 Kondisi CPU VPS sebelum serangan TCP Flooding

Pada Gambar 23 kinerja CPU Virtual Private Server hanya terlihat 1% digunakan, dalam kondisi seperti itu dapat dipastikan tidak ada serangan atau lalu lintas data yang banyak atau padat pada Virtual Private Server, dapat dilihat juga pada perintah "top" untuk melihat status latar belakang pada virtual private server yang dapat dilihat pada Gambar 24



Gambar 24 Perintah Top Sebelum pengujian Serangan TCP Flooding

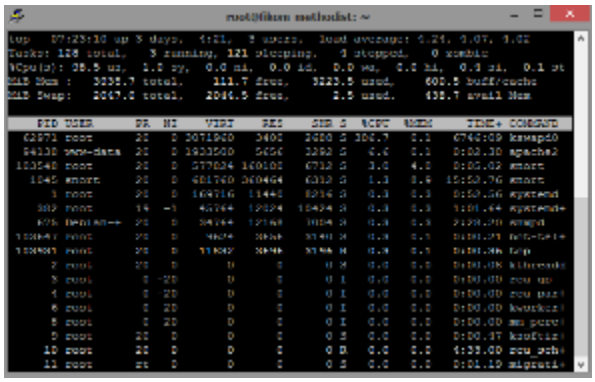
Pada Gambar 24 dapat dilihat bahwa perintah "top" menunjukkan status cpu dalam persen yaitu 0,3%. Selanjutnya dilakukan serangan TCP Flooding dapat dilihat pada Gambar 25

Gambar 25 Pengujian Serangan TCP Flooding

Pada pengujian Gambar 25 dapat dilihat bahwa pengujian menggunakan aplikasi LOIC dengan method serangan TCP yaitu serangan yang mengarah kepada protocol TCP dengan membanjiri lalu lintas paket data sehingga membuat kinerja CPU Virtual Private Server menjadi lebih padat dapat dilihat pada Gambar 26

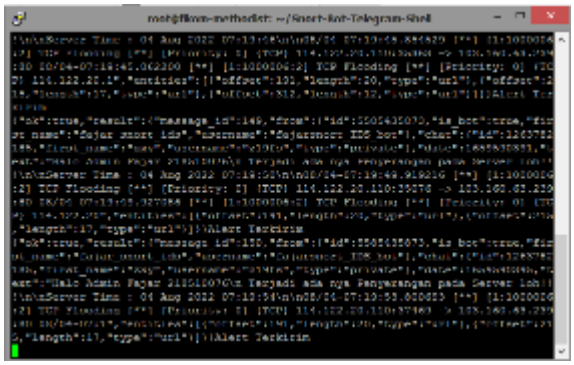
Gambar 26 Status CPU Ketika Serangan TCP Flooding

Pada Gambar 26 dapat dilihat bahwa pengujian serangan DDoS TCP Flooding dapat mempengaruhi kinerja CPU Virtual Private Server menjadi 18%, status CPU pada Virtual Private Server nantinya akan dapat terus bertambah dikarenakan threads yang ada pada aplikasi LOIC yang artinya adalah ada 10 serangan fiktif atau 10 serangan client yang mengirimkan paket TCP yang sangat banyak secara bersamaan dan terus menerus, pada perintah *top* setelah melakukan tindakan serangan TCP Flooding dapat dilihat bahwa status persen pemakaian cpu virtual private server menjadi naik 6,6% yang dapat dilihat pada Gambar 27



Gambar 27 Perintah Top Setelah melakukan Serangan TCP Flooding

Selanjutnya dengan pengujian serangan yang dilakukan sehingga membuat snort berjalan pada Virtual Private Server dan memberikan informasi lalu lintas paket yang berjalan pada Virtual Private Server yang dapat dilihat pada Gambar 28



Gambar 28 Snort Memberikan informasi Serangan TCP Flooding

Berdasarkan Gambar 28 dapat dilihat Snort sudah memberikan notifikasi atau pemberitahuan bahwa terjadi adanya lalu lintas paket TCP yang mengarah kepada virtual private server atau mengarah kepada IP Address virtual private server, dan pada pemberitahuan tersebut juga terkirim ke Bot Telegram yang dapat dilihat pada Gambar 29

Gambar 29 Notifikasi Serangan DDoS TCP Flooding

Berdasarkan pada Gambar 29 dapat dilihat bahwa notifikasi serangan TCP Flooding berhasil masuk pada bot telegram, sehingga snort yang berjalan sebagai perangkat lunak dalam intrusi detection system berhasil dilakukan.

IV. KESIMPULAN

Berdasarkan pembahasan pada bab-bab sebelumnya dari proses konfigurasi dan implementasi sistem maka dapat diambil kesimpulan sebagai berikut :

1. *Port Knocking* telah dapat diterapkan pada Virtual Private Server dengan sistem operasi Ubuntu Server dan Ubuntu Client serta menggunakan tools *iptables* dan *knockd* dalam mengatur akses *port logic* FTP, SSH, Telnet dan HTTP Server pada server dengan membuat ketukan dalam perintah membuka dan menutup port sebagai batasan akses pengguna ketika port tidak sedang mau digunakan dan akan digunakan.
2. *Intrusion Detection System* telah dapat diterapkan pada Virtual Private Server dengan Sistem Operasi Ubuntu Server serta menggunakan tools *snort* dan Snort Bot Telegram Shell untuk menghubungkan *snort* dengan Bot Telegram dan *Intrusion Detection System* dapat berjalan dengan baik dengan *snort* sebagai perangkat lunak yang mendeteksi adanya *intrusi* atau serangan *DDoS (TCP Flooding, UDP Flooding, ICMP Flooding)* kepada server ditambah dengan tindak lanjut menerapkan notifikasi kepada Bot Telegram sehingga memudahkan administrator server dalam mengawasi aktifitas yang terjadi terhadap server.
3. *Virtual Private Server* mampu menggantikan server fisik dalam menerapkan metode keamanan *Port Knocking* dan *Intrusion Detection System* dan menjadi hemat biaya untuk perbandingan dalam membangun komputer server fisik daripada berlangganan layanan *Virtual Private Server*.

Pada penelitian ini penulis memberikan saran-saran yang perlu dalam pengembangan berikutnya, ialah sebagai berikut :

1. Penambahan port-port untuk sumber daya lainnya yang ada di server seperti Email Server, Network Monitoring serta autentikasi pada port logic lainnya didalam Port Knocking, dan implementasi menggunakan multiplatform sistem operasi, serta penggunaan Artificial Intelligence untuk pengaturan Port Knocking.
2. Melengkapi rules terhadap serangan lainnnya di dalam snort dan penggunaan Artificial Intelligence untuk menggunakan snort.

V. DAFTAR PUSTAKA

[1] Efendi, N. P., & Kusuma, J. F. (n.d.). *Sistem keamanan jaringan menggunakan snort*.

[2] Geges, S., & Wibisono, W. (n.d.). *CLIENT PUZZLE*. 53–67.

[3] Mulyanto, Y., Julkarnain, M., & Afahar, A. J. (2021). Implementasi Port Knocking Untuk Keamanan Jaringan Smkn 1 Sumbawa Besar. *Jinteks*, *3*(2), 326–335.

[4] Nainggolan, L. F., Saragih, N. F., & Larosa, F. G. N. (2022). *Monitoring Keamanan Jaringan Pada Server Ubuntu Dari Serangan DDoS Menggunakan Snort IDS*. *2*(2), 1–10.

[5] Simanjuntak, A. G., Saragih, N. F., & Purba, M. J. (2022). *Pengamanan Mikrotik Routerboard Dari Serangan Keamanan Dengan Notifikasi Bot Telegram*. *2*(1), 29–37.

[6] Suwaryo, N., Nawangsih, I., & Rejeki, S. (n.d.). *No Title*.

[7] Ulfa, M. (n.d.). *DI JARINGAN UNIVERSITAS BINA DARMA*. *12*, 105–118.