KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)
Terakreditasi Nomor 204/E/KPT/2022 | Vol. 5, No. 3, Juli (2024), pp. 901-912

Penggunaan *Snort* Sebagai Sistem Pendeteksi Serangan Pada Jaringan Menggunakan Notifikasi Telegram (Kasus Dinas Komunikasi Informatika Dan Persandian Kabupaten Sukabumi)

Anggun Fergina[1], Sultan Alif Nur Ikhsan[2], Zaenal Alamsyah[3]
[1,2,3]Universitas Nusa Putra, Sukabumi, Indonesia
E-mail: [1]anggun.fergina@nusaputra.ac.id, [2]sultan.alif_ti20@nusaputra.ac.id, [3]zaenal.alamsyah@nusaputra.ac.id

Abstract

Network security has become very important to protect individuals, companies, and agencies from threats such as cyberattacks, and data theft. Therefore, understanding the importance of network security is essential. Based on the interview results, there is a server in the Sukabumi District Command Center managed by DISKOMINFOSAN Sukabumi District, the server is used as an application server. The server does not have a network security monitoring system that provides alerts when there is an attempted attack on the server in real time. One way to improve security on the server is to use an Intrusion Detection System (IDS). IDS is a system intended to detect suspicious activity or attacks on the network. One of the main goals of IDS is to provide warnings against security threats that may occur. Snort is one of the open-source IDS tools. Snort was created to identify network attacks and provide realtime alerts to administrators when identifying certain behaviors or attack patterns. In this study the authors used the SPDLC development method. Security Policy Development Life Cycle (SPDLC) is a system development method that focuses on network security. After testing, it can be concluded that snort can be used as an IDS installed on ubuntu server 22.04, with the rules that have been made snort can detect when someone tries port scanning to the server using masscan and can detect ping attacks aimed at the server in real time. With the script that has been created, snort can send alerts to network administrators using telegram in realtime so that these alerts can be followed up immediately.

***Keywords*:** *Intrusion Detection System (IDS), Security Policy Development Life Cycle (SPDLC), Server, Snort, Telegram*

Abstrak

Keamanan jaringan menjadi sangat penting untuk melindungi individu, perusahaan, dan instansi dari ancaman seperti serangan siber, dan pencurian data. Oleh karena itu, memahami pentingnya keamanan jaringan sangat penting. Berdasarkan hasil wawancara, terdapat server di Command Center Kabupaten Sukabumi yang dikelola oleh DISKOMINFOSAN Kabupaten Sukabumi, server tersebut digunakan sebagai server aplikasi. Pada server tersebut belum memiliki sistem monitoring keamanan jaringan yang memberikan alert ketika terjadi percobaan penyerangan di server tersebut secara realtime. Salah satu cara untuk meningkatkan keamanan pada server tersebut adalah dengan menggunakan Intrusion Detection System (IDS). IDS adalah sistem yang dimaksudkan untuk mendeteksi aktivitas mencurigakan atau serangan pada jaringan. Salah satu tujuan utama IDS adalah untuk memberikan peringatan terhadap ancaman keamanan yang mungkin terjadi. Snort adalah salah satu tools IDS bersifat open-source. Snort dibuat untuk mengidentifikasi serangan jaringan dan memberikan peringatan secara realtime kepada administrator ketika mengidentifikasi perilaku atau pola serangan tertentu. Dalam penelitian ini penulis menggunakan metode pengembangan SPDLC . Security Policy Development Life Cycle (SPDLC) adalah sebuah metode pengembangan sistem yang berfokus pada keamanan jaringan. Setelah melakukan pengujian, dapat disimpulkan bahwa snort dapat digunakan sebagai IDS yang dipasang

ISSN: 2720-992X

di ubuntu server 22.04, dengan rules yang telah dibuat snort dapat mendeteksi ketika ada yang mencoba port scanning ke server menggunakan masscan dan dapat mendeteksi ping attack yang ditujukan ke server secara realtime. Dengan script yang telah dibuat, snort dapat mengirimkan alert kepada network administrator menggunakan telegram secara realtime agar alert tersebut dapat langsung ditinjaklanjuti.

Kata Kunci : *Intrusion Detection System (IDS), Security Policy Development Life Cycles (SPDLC), Server, Snort, Telegram*

1. Pendahuluan

Dinas Komunikasi Informatika dan Persandian (DISKOMINFOSAN) Kabupaten Sukabumi adalah salah satu lembaga ditingkat pemerintahan daerah yang bertugas untuk mengelola infrastruktur teknologi informasi pemerintah daerah, yang mencakup pengembangan sistem informasi, manajemen data, dan keamanan data. Adapun bagian dari Dinas Komunikasi Informatika dan Persandian yakni Command Center Kabupaten Sukabumi. Command Center merupakan pusat pengendalian atau pemantauan yang mengkoordinasikan berbagai kegiatan atau informasi dari berbagai sumber. Kemajuan teknologi telah menimbulkan tantangan keamanan yang signifikan. Keamanan jaringan menjadi sangat penting untuk melindungi individu, perusahaan, dan instansi dari ancaman seperti serangan siber, dan pencurian data. Oleh karena itu, memahami pentingnya keamanan jaringan sangat penting. Menurut Laporan Tahunan Monitoring Keamanan Siber 2021, Badan Siber dan Sandi Negara (BSSN) mencatat 1.637.973.022 trafik anomali dengan 242.066.168 anomali tertinggi pada bulan Desember, 264 kasus phishing, 5.940 kasus *web defacement* dengan 727 kasus tertinggi pada bulan Maret di Indonesia [1]. Berdasarkan hasil wawancara, terdapat server di Command Center Kabupaten Sukabumi yang dikelola oleh DISKOMINFOSAN Kabupaten Sukabumi, server tersebut digunakan sebagai server aplikasi. Pada server tersebut belum memiliki sistem monitoring keamanan jaringan yang memberikan *alert* ketika terjadi percobaan penyerangan di server tersebut secara *realtime*. Salah satu cara untuk meningkatkan keamanan pada server tersebut adalah dengan menggunakan *Intrusion Detection System* (IDS). *Intrusion Detection System* adalah sistem yang dimaksudkan untuk mendeteksi aktivitas mencurigakan atau serangan pada jaringan atau sistem komputer [2]. Salah satu tujuan utama *Intrusion Detection System* adalah untuk memberikan peringatan terhadap ancaman keamanan yang mungkin terjadi. *Snort* adalah salah satu *tools Intrusion Detection System* bersifat *open-source*. *Snort* dibuat untuk mengidentifikasi serangan jaringan dan memberikan peringatan secara *realtime* kepada *administrator* atau sistem keamanan ketika mengidentifikasi perilaku atau pola serangan tertentu [3].

2. Metodologi Penelitian

Penelitian kualitatif merupakan metode penelitian yang difokuskan pada upaya mendeskripsikan keadaan, sifat, atau hakikat suatu fenomena atau nilai tertentu pada suatu objek. Metode ini menghasilkan data deskriptif dalam bentuk kata-kata, baik yang ditulis maupun lisan, yang diperoleh dari narasumber atau perilaku yang menjadi fokus pengamatan. Penjelasan tersebut menekankan pada jenis data yang dikumpulkan selama proses penelitian, yakni data deskriptif kualitatif, yang bertujuan untuk memberikan Gambaran dan menggali makna dari suatu fenomena [4].

2.1. Landasan Teori

2.1.1. *Intrusion Detection System*

Sistem deteksi intrusi atau *Intrusion Detection System* (IDS) merupakan sistem yang dapat menganalisis data secara *realtime* untuk mendeteksi dan mencatat terjadinya serangan siber. Sistem deteksi intrusi (IDS) ini dapat memberikan peringatan kepada

administrator ketika terjadi serangan atau penyalahgunaan jaringan; peringatan ini bahkan dapat mengungkapkan alamat IP dari sistem penyerang [5].

2.1.2. Snort

Snort adalah *tools* IDS berbasis *opensource* yang memiliki kemampuan untuk mengidentifikasi dan mengirimkan *alert* jika terdapat indikasi penyusupan pada jaringan secara *realtime*. Jika ditemukan, *Snort* akan menggunakan aturan yang telah ditetapkan untuk mencatat atau melacak paket-paket yang telah diidentifikasi sebagai intrusi [6]. *Packet Capture*, *Packet Decoder*, *Prepocessing*, *Detection Engine*, dan *Output Module* merupakan komponen yang ada pada snort, komponen-komponen tersebut bekerjasama untuk mendeteksi serangan [7].

2.1.3. Keamanan Jaringan

Keamanan jaringan merupakan upaya untuk melindungi integritas, kerahasiaan, dan aksesibilitas informasi yang disimpan, diproses, dan ditransmisikan melalui jaringan komputer. Ini mencakup serangkaian tindakan dan teknologi yang bertujuan untuk mencegah orang yang tidak dimaksud mengakses, menyalahgunakan, atau mengganggu jaringan dan sistem yang terhubung [8].

2.1.4. Server

Server merupakan perangkat atau sistem komputer berfungsi untuk menyediakan layanan, sumber daya, atau informasi kepada komputer lain yang terhubung ke jaringan. Fungsi utama server termasuk penyimpanan dan distribusi data, mengelola akses pengguna, dan menyediakan berbagai layanan seperti email, hosting web, basis data, dan berbagai aplikasi kepada pengguna atau perangkat yang terhubung ke jaringan [9].

2.1.5. *Elasticsearch*

Elasticsearch adalah salah satu basis data klasifikasi NoSQ. Database ini berfokus pada database mesin pencari, bersifat open source, bebas digunakan, dan memiliki tujuan atau kemampuan yang dapat diperluas sebagai mesin pencari dan mesin analisis teks. *Elasticsearch* juga memudahkan untuk menyimpan, mengambil, dan menganalisis data dari kumpulan data yang besar dengan cepat dan *realtime* [10].

2.1.6. *Filebeat*

Filebeat adalah agen pengiriman data ringan yang digunakan untuk mengumpulkan dan mengirimkan log dan data lainnya ke pusat seperti *Elasticsearch* [11].

2.1.7. *Kibana*

Kibana adalah aplikasi opensource untuk analisis dan visualisasi data yang bekerja secara terintegrasi dengan *Elasticsearch*. *Kibana* dapat diakses melalui browser dan menawarkan fitur untuk menyajikan data dalam bentuk diagram, Tabel, dan analisis data tingkat lanjut. Selain itu, *Kibana* memungkinkan pembuatan *dashboard* yang menyajikan hasil kueri *Elasticsearch* secara *realtime* [12].

2.1.8. *Virtualbox*

Oracle VM VirtualBox merupakan sebuah aplikasi virtualisasi yang memungkinkan pengguna untuk menjalankan sistem operasi tambahan di dalam sistem operasi utama. Fungsinya terletak pada virtualisasi sistem operasi, serta mampu membuat simulasi jaringan komputer yang sederhana. Penggunaan VirtualBox mencakup penggunaan pada server, desktop [13].

2.1.9. *Port Scanning*

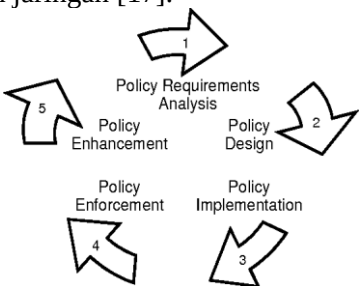
Port scanning adalah cara untuk menemukan *port* terbuka pada komputer atau jaringan yang rentan terhadap serangan. Penyerang dapat menggunakan informasi ini untuk mengeksploitasi kerentanan dan mendapatkan akses ke komputer atau jaringan [14] [15].

2.1.10. *Telegram*

Telegram merupakan aplikasi yang memungkinkan penggunanya mengirim pesan dengan cepat dan aman. Selain itu, Telegram sangat ringan, sederhana, dan gratis. Bot Telegram adalah akun khusus yang tidak memerlukan nomor telepon pendaftaran tambahan ke Server Telegram. Akun ini bertindak sebagai antarmuka antara kode program dan server Telegram. Telegram adalah salah satu aplikasi yang mendukung bot [16].

2.2. Metode Pengembangan Sistem

Dalam penelitian ini penulis menggunakan metode pengembangan SPDLC. *Security Policy Development Life Cycle* (SPDLC) adalah sebuah metode pengembangan sistem yang berfokus pada keamanan jaringan [17].



Gambar 1. *Security Policy Development Life Cycle (SPDLC)*

Ada lima tahapan pada metode pengembangan menggunakan *Security Policy Development Life Cycle* (SPDLC), yaitu: Analysis, Design, Implementation, Enforcement, dan Enchancement. Akan tetapi pada penelitian kali ini, hanya sampai pada tahap Enforcement atau tahap pengujian saja.

2.2.1. *Analysis*

Pada tahap ini melakukan analysis terkait aset informasi yang perlu dilindungi, termasuk data, perangkat keras, dan jaringan, dan mengidetifikasi risiko keamanan berdasarkan kemungkinan dan dampak dari ancaman.

2.2.2. *Design*

Setelah melakukan *analysis*, tahap selanjutnya adalah *design*. Pada tahap ini dilakukan perancangan topologi jaringan di Command Center Kabupaten Sukabumi.

2.2.3. *Implementation*

Tahap selanjutnya adalah implementasi, pada tahap ini melingkupi instalasi dan konfigurasi komponen sistem IDS *snort* dengan notifikasi melalui telegram di server Command Center Kabupaten Sukabumi.

2.2.4. *Enforcement*

Setelah tahap implementasi, dilanjutkan dengan tahap pengujian (*enforcement*), dimana penulis melakukan kegiatan operasional dan mengamati apakah sistem yang dibangun dan diimplementasikan yaitu sistem IDS dengan notifikasi telegram sudah berfungsi dengan baik. dan penulis melakukan pengujian terhadap sistem yang telah dibangun.

2.2.5. *Enchancement*

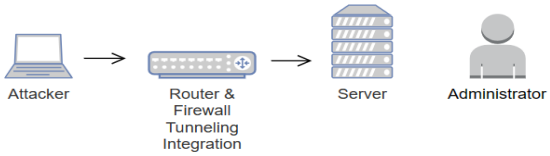
Tahap terakhir dari SPDLC (*Security Policy Development Life Cycle*) adalah evaluasi (*enchancement*). Pada tahap ini, perbaikan dilakukan pada sistem yang telah ditetapkan.

3. Hasil dan Pembahasan

3.1. *Analysis*

3.1.1. *Analysis* Sistem Berjalan

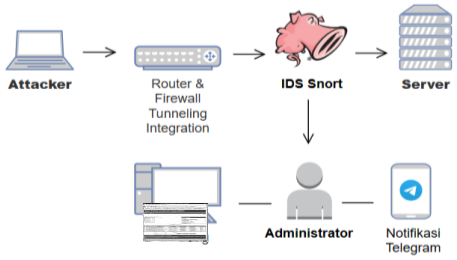
Ketika terdapat serangan ke server oleh *attacker*, seorang *network administrator* tidak mendapatkan peringatan atau *alert* secara *realtime* jika ada serangan ke server.



Gambar 1. *Analysis* Sistem Berjalan

3.1.2. *Analysis* Sistem Usulan

Berdasarkan hasil analisis dibutuhkan *Intrusion Detection System (*IDS) pada server menggunakan *tools snort*. *Snort* digunakan untuk dapat memonitor lalu lintas jaringan dan memberikan laporan status sistem secara *realtime* melalui telegram sebagai media notifikasi kepada *administrator* jaringan jika terjadi serangan ke server. *Administrator* jaringan juga dapat melihat hasil pemantauan peringatan dari IDS *snort* melalui antarmuka web pada situs web.



Gambar 2. *Analysis* Sistem Usulan

3.1.3. Kebutuhan *Hardware* dan *Software*

Adapun kebutuhan *Hardware* dan *Software* pada penelitian ini dapat dilihat pada Tabel 1 dan Tabel 2.

Tabel 1 Kebutuhan *Hardware*

No	Komponen	Spesifikasi	Jumlah (pcs)
1	Router	Mikrotik RB5009UG 7x Gigabit Ethernet Ports	1 *pcs*
2	Server	HPE ProLiant DL380 Gen9 Xeon E5-2620v4 8 Core 2,1Ghz 16GB RAM 1.2TB Hard Drive	1 *pcs*
3	Router	TP-Llink Archer C5400 1x Gigabit WAN Port 4x Gigabit LAN Ports	1 *pcs*
4	Switch	TP-Link Gigabit TL-SF-1024 24x 10/100Mbps RJ45 Ports	1 *pcs*
6	Access Point	Ubiquiti UAP AC Pro 2x Gigabit Ethernet RJ45 Port	1 *pcs*
7	Laptop	Intel Core i5-1135g7 16GB RAM 512GB SSD	2 *pcs*

Tabel 2 Kebutuhan *Software*

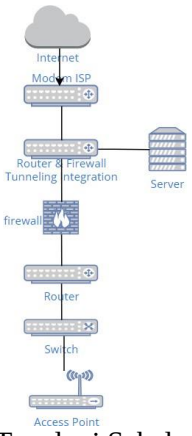
No	Nama	Keterangan
1	Ubuntu Server 22.04	Digunakan untuk menginstall *snort*
2	Kali Linux 2022.2	Digunakan sebagai penyerang server
3	Windows 11	Digunakan sebagai *administrator* jaringan
4	Virtualbox 7.0	Digunakan untuk menginstall kali linux
5	Snort 2.9.15.1	Digunakan sebagai *tools Intrusion detection System*
6	Filebeat 8.13.3	Digunakan untuk mengumpulkan data log dan dikirimkan ke Elasticsearch
7	Elasticsearch 8.13.3	Digunakan untuk melakukan pencarian dan analisis data secara *realtime*
8	Kibana 8.13.3	Digunakan untuk memvisualisasikan data yang tersimpan di Elasticsearch
9	Masscan 1.3.2	Tools untuk pengujian *port scanning*
10	Terminal	Digunakan untuk pengujian *ping attack*
11	Command Prompt (CMD)	Digunakan untuk remote server
12	Telegram	Digunakan untuk mengirimkan *alert* kepada *administrator*

3.2. Design

Pada tahap *design* ini dilakukan perancangan topologi yang digunakan untuk mengimplementasikan IDS.

3.2.1. Topologi Jaringan Sebelum Diterapkan IDS

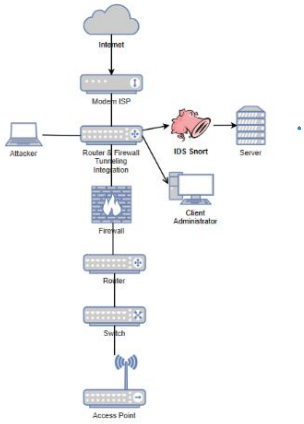
Pada Gambar 4 merupakan topologi jaringan di Command Center Kabupaten Sukabumi sebelum diterapkan snort.



Gambar 3. Topologi Sebelum Diterapkan IDS

3.2.2. Topologi Jaringan Setelah Diterapkan IDS

Pada Gambar 5 dibawah ini merupakan topologi jaringan di Command Center setelah ditetapkan IDS *snort*. Komputer *client administrator* akan digunakan untuk akses server menggunakan CMD, *snort* akan di *install* di *virtual private server* yang terdapat pada server. *Attacker* yang terhubung ke router akan digunakan sebagai penyerang terhadap server dan akan melakukan *port scanning* dan *ping attack* ke server.



Gambar 4. Topologi Jaringan Setelah Diterapkan IDS

Tabel 3 Rincian IP Address

No	Komponen	IP Address
1	Router Integration	114.7.153.1
2	Server (VPS)	114.7.153.78
3	Laptop Attacker	114.7.153.29
4	Client Administrator	114.7.153.30
5	Router	114.7.10.1
6	Switch	114.7.20.1
7	Access Point	114.7.30.1

3.3. Implementation

3.3.1. Install dan Konfigurasi Snort

Langkah awal dalam tahap *implementation* adalah menginstall dan mengkonfigurasi snort pada server dan menambahkan *rules* agar snort dapat mendeteksi *port scanning* dan *ping attack* yang ditujukan ke server.

Langkah 1 (Update package pada ubuntu server)
Sudo apt update && sudo apt upgrade

Langkah 2 (Install snort)
Sudo apt install snort

Langkah 3 (Menambahkan rules)
Sudo nano /etc/snort/rules/local.rules

- *alert* tcp any any -> $*HOME*_NET any (msg:" Terdeteksi *Port Scanning* ke Server!!!"; flags: S; classtype:attempted-recon; sid:1000001; rev:001;)
- *alert* icmp any any -> $*HOME*_NET any (msg:"Terdeteksi Ada Yang Kirim PING Ke Server!!!"; sid:1000002; rev:002;)

Gambar 6. Lagkah-Langkah Install Dan Konfigurasi Snort

3.3.2. Konfigurasi *Script Bash Shell* Telegram

Setelah menginstall dan mengkonfigurasi *snort* pada server, selanjutnya adalah membuat *script bash shell* telegram, *Script* akan digunakan agar telegram bot dapat terhubung dengan *snort*, kemudian membaca log serangan yang berjalan di IDS *snort* dan mengirimkan pemberitahuan serangan ke telegram, dan log *alert* nya akan tersimpan di /*home*/*snort*/*logs*.txt.

Langkah 1 (download script bash shell telegram)
Git clone https://github.com/theozebula/telegram-bot-for-snort.git

Langkah 2 (Membuka file alert-bot.sh)
Sudo nano alert-bot.sh

Langkah 3 (isi chat_id dan token)
Isi chat_id dan token, menggunakan chat_id dan token bot telegram yang telah dibuat

Langkah 4 (Perintah agar script dapat access permission untuk dijalankan)
Chmod 777 alert-bot.sh

Gambar 7. Langkah-Langkah Konfigurasi *Script Bash Shell* Telegram

```
 GNU nano 6.2                                                alert-bot.sh
#!/bin/bash

#Initialization
initCount=0
logs=/home/snort/logs.txt

#Telegram temporary message
message=/tmp/message.txt

#Chat id and bot telegram token
chat_id="-1002034277793"
token="6835105104:AAGBY_OU8LUkdcawGFGih1L4P1MbHgH62DU"

#Send Alert Function
function sendAlert
{
        curl -s -F chat_id=$chat_id -F text="$text" https://api.telegram.org/bot$token/sendMessage
}

#Running the program
while true
do
    lastCount=$(wc -c $logs | awk '{print $1}') #getSizeFileLogs

    if(($($lastCount)) > $initCount));
        then
        msg=$(tail -n 2 $logs) #GetLastLineLog
        echo -e "Halo Admin!\n Terdeteksi Percobaan Penyerangan Pada Server!!!\n\nServer Time : $(date +"%d %b %Y %T")\n\n"$msg > $message
        text=$(<$message)
        sendAlert
        echo " Alert sent!"
        initCount=$lastCount
        rm -f $message
        sleep 1
    fi
    sleep 2
done
```

Gambar 8. *Script Bash Shell* Telegram

3.3.3. *Install* dan Konfigurasi *Elasticsearch*, *Kibana*, dan *Filebeat*

Setelah konfigurasi *script bash shell* telegram, langkah selanjutnya adalah meng*install Elasticsearch*. *Elasticsearch* tidak tersedia di paket bawaan Ubuntu. Untuk memasangnya, perlu menggunakan APT (*Advance Package Tool*) setelah menambahkan sumber paket *Elasticsearch*. Setelah itu *install elasticsearch*, konfigurasi *elasticsearch*.yml dan mengaktifkan layanan *elasticearch*. Pada Gambar 9 dibawah ini merupakan langkah-langkah meng*install* dan mengkonfigurasi *Elasticsearch*.

Langkah 1 (Impor *public* gpg *key*, diperlukan untuk verifikasi integritas paket *Elasticsearch*)
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /usr/share/keyrings/elasticsearch-keyring.gpg

Langkah 2 (Perintah agar sistem dapat mengunduh dan menginstal paket dari repositori melalui koneksi HTTPS yang aman)
sudo apt-get install apt-transport-https

Langkah 3 (Perintah untuk menambahkan definisi repositori paket *Elasticsearch* ke sistem ubuntu)
echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-8.x.list

Langkah 4 (Install Elasticsearch)
sudo apt-get install elasticsearch

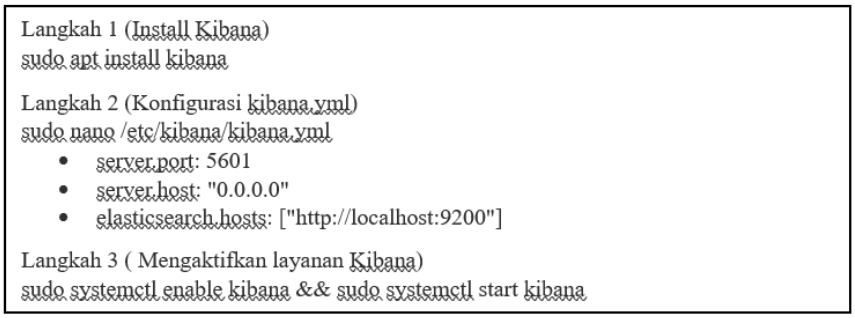
Langkah 5 (Konfigurasi elasticsearch.yml)
sudo nano /etc/elasticsearch/elasticsearch.yml

- network.host: 0.0.0.0
- http.port: 9200

Langkah 6 (Mengaktifkan layanan Elasticsearch)
sudo systemctl enable elasticsearch && sudo systemctl start elasticsearch

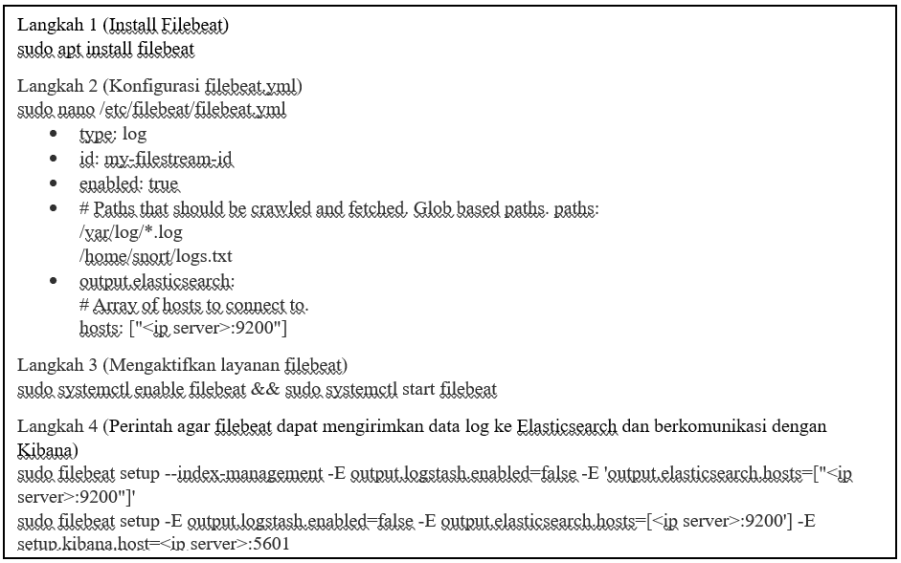
Gambar 9. Lagkah-Langkah *Install* Dan Konfigurasi *Elasticsearch*

Setelah menginstall *Elasticsearch*, langkah sealanjutnya adalah meng*install* dan mengkonfigurasi *Kibana*. Pada Gambar 10 dibawah ini merupakan langkah-langkah meng*install* dan mengkonfigurasi *Elasticsearch*.



Gambar 10. Lagkah-Langkah *Install* Dan Konfigurasi *Kibana*

Setelah menginstall *Kibana*, langkah sealanjutnya adalah meng*install*, mengkonfigurasi dan mengaktifkan layanan *Filebeat*. Pada Gambar 11 dibawah ini merupakan langkah-langkah meng*install* dan mengkonfigurasi *Filebeat*.



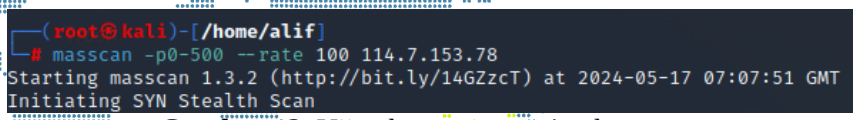
Gambar 11. Lagkah-Langkah *Install* Dan Konfigurasi *Filebeat*

3.4. Enforcement

Pada tahap ini akan dilakukan uji coba penyerangan terhadap sistem yang telah dibuat. Teknik penyerangan *port scanning* dan *ping attack* akan digunakan untuk uji coba penyerangan terhadap sistem. Pada server diperlukan perintah untuk menjalankan *snort* dan menyimpan log *alert* pada /*home*/*snort*/*logs.txt*, dengan perintah "sudo *snort* -i <network interface (bisa enp0s3/ens18/dll)> -c /etc/*snort*/*snort*.conf -l /var/log/*snort* -d -A console > /*home*/*snort*/*logs*.txt" akan menjalankan *snort* dan menyimpan log *alert* pada /*home*/*snort*/*logs*.txt. Selanjutnya, agar log *alert* yang tersimpan dapat diteruskan ke telegram, maka diperlukan perintah yaitu dengan perintah "./*alert*-bot.sh".

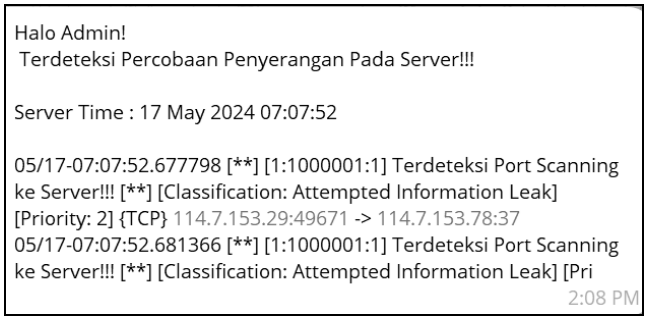
3.4.1. *Port Scanning*

Pada Gambar 12 merupakan uji coba *port scanning* ke server menggunakan *tools* msscan. Dengan perintah "msscan -p0-500 –rate 100 *ip address*", maka masscan akan melakukan pemindaian pada *port* 0 – 500 dengan kecepatan 100 paket/detik ke *ip address* server.



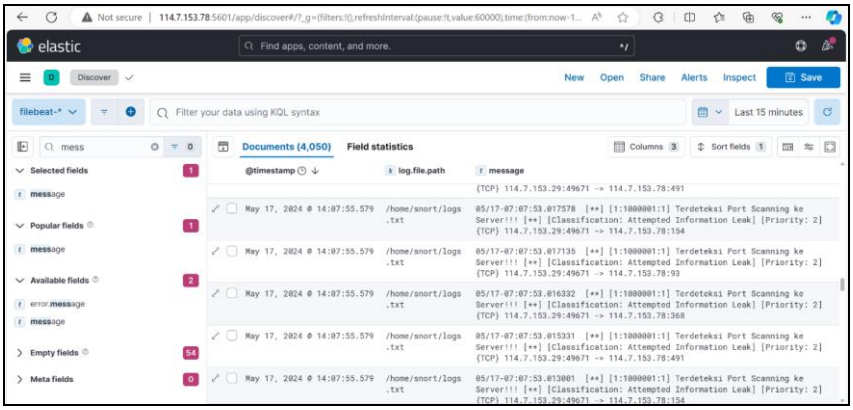
Gambar 12. Uji coba *port scanning* ke server

Pada Gambar 13 dibawah ini merupakan *alert* yang dikirimkan melalui telegram. Ini menandakan bahwa *snort* dapat mendeteksi *port scanning* yang ditujukan ke server dengan menggunakan *tools masscan*.



Gambar 13. *Alert* dari telegram ketika terdeteksi *port scanning* ke server

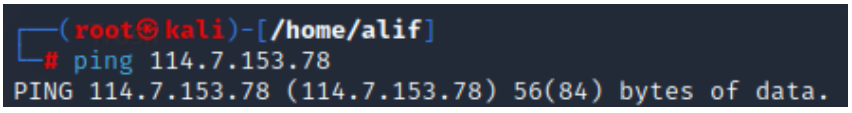
Pada Gambar 14 merupakan *alert* pada *dashboard monitoring*. Ini menandakan bahwa *Filebeat* berhasil menerima log yang tersimpan di /*home*/*snort*/*logs*.txt dan meneruskan nya untuk diolah di *Elasticsearch* dan divisualisasikan oleh *Kibana*. Terdapat pesan "Terdeteksi *Port Scanning* ke Server!!!".



Gambar 14. Tampilan *dashboard* pada saat terjadi *port scanning*

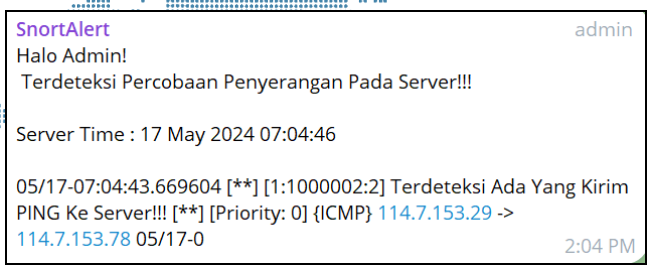
3.4.2. *Ping Attack*

Pada Gambar 15 dibawah ini merupakan uji coba *ping attack ke* server menggunakan terminal. Dengan perintah "ping *ip address*", akan mengirim paket ICMP tanpa henti ke alamat *ip address* server.



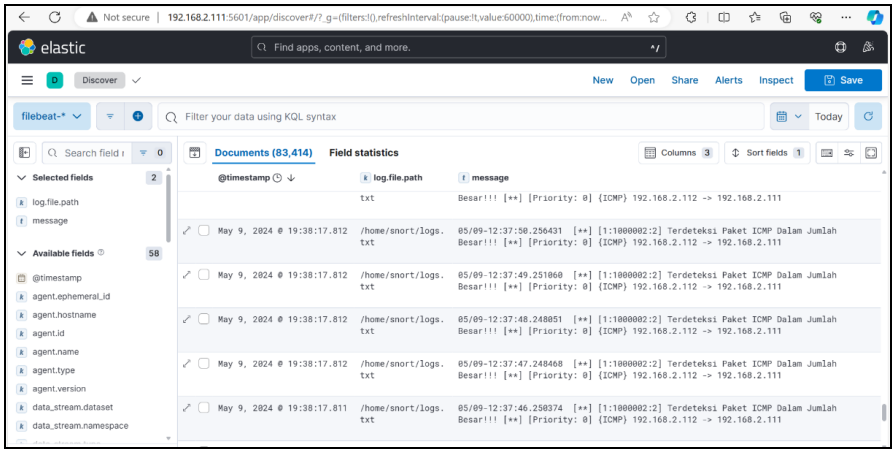
Gambar 15. Uji coba *ping attack* ke server

Pada Gambar 16 merupakan *alert* yang dikirimkan melalui telegram. Ini menandakan bahwa *snort* dapat mendeteksi *ping attack* yang ditujukan ke server dengan menggunakan terminal.



Gambar 16. *Alert* dari telegram ketika terdeteksi *ping attack* ke server

Pada Gambar 17 merupakan *alert* pada *dashboard monitoring*. Ini menandakan bahwa *Filebeat* berhasil menerima log yang tersimpan di /*home*/*snort*/*logs*.txt dan meneruskan nya untuk diolah di *Elasticsearch* dan divisualisasikan oleh *Kibana*. Terdapat pesan "Terdeteksi Ada Yang Kirim PING Ke Server!!!".



Gambar 17. Tampilan *dashboard* ketika terjadi *ping attack*

4. Kesimpulan

Setelah melakukan pengujian, dapat disimpulkan bahwa snort dapat digunakan sebagai *Intrusion Detection System* yang dipasang di ubuntu server 22.04, dengan rules yang telah dibuat snort dapat mendeteksi ketika ada yang mencoba *port scanning* ke server menggunakan masscan dan dapat mendeteksi *ping attack* yang ditujukan ke server secara realtime. Dengan *script bash shell* yang telah dibuat, *snort* dapat mengirimkan *alert* kepada *network administrator* menggunakan telegram secara *realtime* agar dapat langsung ditinjaklanjuti *alert* tersebut. *Elasticsearch*, *Filebeat*, dan *Kibana* dapat digunakan sebagai *dashboard monitoring*.

Daftar Pustaka

[1] M. Y. Samad and Pratama Dahlian Persadha, "Memahami Perang Siber dan Peran Badan Intelijen Negara Dalam Menangkal Ancaman di Siber," *J. IPTEKKOM J. Ilmu Pengetah. Teknol. Inf.*, vol. 24, no. 2, pp. 135–146, 2022, doi: 10.17933/iptekkom.24.2.2022.135-146.

[2] I. A. S. Dewi Paramitha, G. M. A. Sasmita, and I. M. S. Raharja, "Analisis Data Log IDS Snort dengan Algoritma Clustering Fuzzy C-Means," *Maj. Ilm. Teknol. Elektro*, vol. 19, no. 1, p. 95, 2020, doi: 10.24843/mite.2020.v19i01.p14.

[3] H. Awal, "Implementasi Intrusion Detection Prevention System Sebagai Sistem Keamanan Jaringan Komputer Kejaksaan Negeri Pariaman Menggunkan Snort Dan Iptables Berbasis Linux," *J. Sains Inform. Terap.*, vol. 2, no. 1, pp. 38–44, 2023, doi: 10.62357/jsit.v2i1.184.

[4] M. Waruwu, "Pendekatan Penelitian Pendidikan: Metode Penelitian Kualitatif,

Metode Penelitian Kuantitatif dan Metode Penelitian Kombinasi (Mixed Method)," *J. Pendidik. Tambusai* , vol. 7, no. 1, pp. 2896–2910, 2023.

[5] F. Riza, "Sistem Deteksi Intrusi pada Server secara Realtime Menggunakan Seleksi Fitur dan Firebase Cloud Messaging," *J. Sistim Inf. dan Teknol.*, vol. 5, pp. 7–9, 2022, doi: 10.37034/jsisfotek.v5i1.161.

[6] G. Tambunan and M. IGN, "Implementasi Keamanan Ids / Ips Dengan Snort Dan IP Tables pada Server," *Semin. Nas. Mhs. Ilmu Komput. dan Apl. Jakarta-Indonesia, 28 Januari 2020 IMPLEMENTASI*, pp. 10–16, 2020.

[7] L. Shuai and S. Li, "Performance optimization of Snort based on DPDK and Hyperscan," *Procedia Comput. Sci.*, vol. 183, no. 2018, pp. 837–843, 2021, doi: 10.1016/j.procs.2021.03.007.

[8] N. A. Santoso, K. B. Affandi, and R. D. Kurniawan, "Implementasi Keamanan Jaringan Menggunakan Port Knocking," *J. Janitra Inform. dan Sist. Inf.*, vol. 2, no. 2, pp. 90–95, 2022, doi: 10.25008/janitra.v2i2.156.

[9] M. A. Husna and P. Rosyani, "Implementasi Sistem Monitoring Jaringan dan Server Menggunakan Zabbix yang Terintegrasi dengan Grafana dan Telegram," *JURIKOM (Jurnal Ris. Komputer)*, vol. 8, no. 6, p. 247, 2021, doi: 10.30865/jurikom.v8i6.3631.

[10] A. Yudhistira and Y. Fitrisia, "Monitoring Log Server Dengan Elasticsearch, Logstash Dan Kibana (Elk)," *Rabit J. Teknol. dan Sist. Inf. Univrab*, vol. 8, no. 1, pp. 124–134, 2023, doi: 10.36341/rabit.v8i1.2975.

[11] H. Khotimah, F. Bimantoro, and R. S. Kabanga, "Implementasi Security Information And Event Management (SIEM) Pada Aplikasi Sms Center Pemerintah Daerah Provinsi Nusa Tenggara Barat," *J. Begawe Teknol. Inf.*, vol. 3, no. 2, pp. 213–219, 2022, doi: 10.29303/jbegati.v3i2.752.

[12] A. Setiyawan, A. Pinandito, and W. Purnomo, "Pengembangan Sistem Informasi Log Management Server Monitoring Menggunakan ELK (Elastic Search, Logstash dan Kibana) Stack pada Aplikasi Padichain di PT. Bank Rakyat Indonesia," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 5, pp. 2142–2151, 2023, [Online]. Available: http://j-ptiik.ub.ac.id

[13] M. K. Anam, D. Sudyana, A. Noviciatie, and N. Lizarti, "Optimalisasi Penggunaan VirtualBox Sebagai Virtual Computer Laboratory untuk Simulasi Jaringan dan Praktikum pada SMK Taruna Mandiri Pekanbaru J-PEMAS STMIK Amik Riau," *http//jurnal.sar.ac.id/index.php/J-PEMAS Optim.*, vol. vol 1, no. 2, pp. 37–44, 2020.

[14] S. Informasi, U. Merdeka, M. Jalan, T. Dieng, and N. Klojen, "Implementasi Honeypot Dionaea Sebagai Uji Kerentanan dan Penunjang Keamanan Jaringan," no. September, pp. 3807–3817, 2023.

[15] A. Fergina, M. I. Setia, M. Yusuf, and ..., "Analisis Monitoring Sistem Keamanan Jaringan Komputer menggunakan Software NMAP (Studi Kasus Jaringan di Universitas Nusa Putra)," *... Ilmu Komput. ...*, 2023, [Online]. Available: http://prosiding.sentimeter.nusaputra.ac.id/index.php/prosiding/article/view/45%0Ahttp://prosiding.sentimeter.nusaputra.ac.id/index.php/prosiding/article/download/45/41

[16] D. K. Hakim and S. A. Nugroho, "Implementasi Telegram Bot untuk Monitoring Mikrotik Router," *Sainteks*, vol. 16, no. 2, pp. 151–157, 2020, doi: 10.30595/st.v16i2.7132.

[17] M. Mukmin, P. Purnawansyah, and M. Hasnawi, "Notifikasi Bot Telegram Untuk Monitoring Jaringan Pada Kementrian Kelautan Dan Perikanan Untia," *Bul. Sist. Inf. dan Teknol. Islam*, vol. 3, no. 2, pp. 127–133, 2022, doi: 10.33096/busiti.v3i2.1162.