CESS
(Journal of Computing Engineering, System and Science) 8(1) January 2023 123-137
e-ISSN: 2502-714x
p-ISSN: 2502-7131

Contents list available at www.jurnal.unimed.ac.id

CESS
(Journal of Computing Engineering, System and Science)

journal homepage: https://jurnal.unimed.ac.id/2012/index.php/cess

Implementasi *Intrusion Prevention System* (IPS) Sebagai Sistem Keamanan *Server* Berbasis *Website* dan Aplikasi *Mobile*

Implementation of Intrusion Prevention System (IPS) as a Website-Based Server Security System and Mobile Application

Rayco William[1], Ikhwan Ruslianto[2*], Uray Ristian[3]

[1,2,3]Rekayasa Sistem Komputer, Fakultas MIPA Universitas Tanjungpura
Jalan Prof. Dr. H. Hadari Nawawi Pontianak
e-mail: [1]h1051171082@student.untan.ac.id,
[2]ikhwanruslianto@siskom.untan.ac.id,[3]eristian@siskom.untan.ac.id

A B S T R A K

Server merupakan pusat penyedia layanan dan penyimpanan data didalam jaringan komputer. Sebuah server dikelola oleh *administrator server* yang bertugas memonitoring keamanan *server*. Dalam bertugas terdapat kekurangan pada cara mendeteksi serangan, lambatnya informasi terjadi serangan dan penanganan serangan pada *server*. Pada penelitian ini dibuat sistem keamanan *server* dengan mengimplementasikan *Intrusion Prevention System* (IPS) berbasis aplikasi *website* dan *mobile*. Deteksi serangan berfokus pada protokol ICMP dan TCP dengan *latency* waktu sistem merespon serangan 99,89ms (Sangat Baik). Sistem penanganan serangan berhasil dilakukan menggunakan Iptables terhadap IP penyerang yang berhasil terdeteksi oleh sistem Suricata melalui aplikasi *website* dan *mobile*, untuk diberikan aksi yang terbagi menjadi *Drop*, *Reject* dan *Accept*. *Administrator* dapat dengan cepat melakukan pencegahan yang diperlukan setelah menerima notifikasi otomatis saat *server* mendapat serangan melalui Telegram dengan kecepatan rata-rata 3,41detik. Pada *server* lokal kondisi awal kinerja CPU berkisar 10-19%, naik saat terjadi *ping attack* menjadi 21,6%,*memory* 41,7%, dan *disk* 19,6%.*Port scanning* kenaikan CPU 85,9%,*memory* 41,9%, dan *disk* 20,3%.*Ping of death* kenaikan *CPU* 90,4%, *memory* 42,9% dan *disk* 20,8%. Berdasarkan pengujian serangan yang telah dilakukan peningkatan berlebihan terdapat pada serangan *ping of death* yang mengakibatkan kinerja *server* meningkat menjadi 90,4%, jika serangan terjadi dalam waktu cukup lama maka kondisi *server* akan menjadi hang (rusak).

Kata Kunci: *Keamanan Server, Intrusion Prevention System (IPS), Suricata, Iptables, Notifikasi Otomatis.*

A B S T R A C T

Server is a center for providing services and storing data in a computer network. A server is managed by server administrator who has a duty of monitoring security server. While on duty,

**Penulis Korespondensi:*
email: ikhwanruslianto@siskom.untan.ac.id

there are deficiencies in detecting attacks, the slow information about the attacks, and how to handle attacks on the server. In this research, a server security system was created by implementing an Intrusion Prevention System (IPS) based on website and mobile applications. Attack detection focuses on ICMP and TCP port attacks with the latency time when the system responds to an attack is 99,89 ms (very good). The attack handling system was successfully carried out using Iptables against the attacker's IP that detected by the Suricata system through the website and mobile applications, to be given action which is divided into Drop, Reject and Accept. Administrators can quickly take the necessary precautions after receiving an automatic notification when the server is under attack via Telegram with an average speed is 3.41second. The ping attack, port scanning and ping of death (DoS) attacks resulted in an increase in the performance load on the local server with the initial conditions of CPU performance ranging from 10-19%, increasing when a ping attack occurred to 21,6%, memory 41,7%, and disk 19,6%. Port scanning increased by 85,9% CPU, memory 41,9%, and disk 20,3%. Ping of death increased CPU 90,4%, memory 42,9%, and disk 20,8%. Based on the tests that have been done, an excessive increase is found in the ping of death attack which results in server performance increasing to 90,4%, if the attack occurs for a long time then the server condition will be hang (damaged).

Keywords: *Server Security, Intrusion Prevention System (IPS), Suricata, Iptables, Automatic Notifications.*

1. PENDAHULUAN

Semakin banyak pengguna didalam suatu jaringan internet menyebabkan risiko serangan terhadap *server* menjadi semakin besar. Menurut laporan dari Honeynet Project Badan Siber dan Sandi Negara (BSSN) Indonesia memiliki riwayat serangan siber sebesar 316,1 juta kali pada tahun 2020. Angka tersebut memiliki peningkatan yang signifikan dibanding dengan 98,2 juta riwayat serangan siber di Indonesia pada tahun 2019 [1]. Dipaparkan juga terdapat beberapa *port* yang sering menjadi target serangan seperti *port* 445 yang merupakan protokol *client-server*, berfungsi sebagai layanan untuk berbagi berkas didalam jaringan *server*, dengan detail serangan berjumlah 224.309.543 kali dan *port* 80 yang berfungsi untuk pengguna *server* agar dapat mengakses layanan aplikasi *web* pada *server* melalui *browser* yang terhubung internet, dengan detail informasi serangan berjumlah 10.759.973 kali.

Sebuah jaringan komputer memiliki komponen utama yaitu *server* yang memiliki peran penting sebagai pusat pemberi layanan dan penyimpanan data didalam jaringan komputer [2]. Sebuah *server* umumnya dikelola oleh *administrator server* yang bertugas melakukan monitoring keamanan serta kinerja *server* secara langsung maupun tidak langsung. Dalam pengawasan kinerja *server* tersebut terdapat kekurangan pada cara mendeteksi ancaman serangan, lambatnya informasi terjadinya serangan oleh penyerang dan tindakan yang dapat dilakukan apabila terjadi serangan pada jaringan *server*. Keamanan sebuah *server* sangat diperlukan agar permintaan layanan dalam jaringan *server* tetap dalam kondisi yang baik selama menjalankan tugasnya. Beberapa ancaman serangan yang sering terjadi pada jaringan *server* adalah *ping attack*, *port scanning* dan *denial of service* (DoS). Serangan tersebut dilakukan dengan tujuan mencegah *client* untuk menggunakan sumber daya didalam jaringan *server*. Serangan ini umumnya mengarah pada *port* TCP dan *port* ICMP yang berfungsi bagi *server* untuk dapat memberikan layanan bagi pengguna *server*. Serangan pada *port* tersebut

dapat berdampak buruk bagi kinerja *server* dalam memberikan pelayanan bagi *client* di dalam jaringan *server*.

Intrusion Detection System (IDS) merupakan sistem yang dapat melakukan monitoring keamanan didalam lalu lintas jaringan *server* dan memberikan respon deteksi berupa peringatan tanda-tanda kemungkinan ancaman serangan didalam jaringan *server*. Di sisi lain, *Intrusion Prevention System* (IPS) merupakan sistem yang memiliki kelebihan dibanding IDS yaitu tidak hanya mendeteksi ancaman serangan tetapi juga melakukan pencegahan terhadap ancaman serangan didalam jaringan *server* [3].

Intrusion Prevention System (IPS) adalah salah satu pilihan untuk meningkatkan keamanan pada sebuah jaringan *server*. Penggunaan IPS dilakukan untuk membantu *administrator* dalam melakukan monitoring, menganalisis dan mengambil tindakan terhadap ancaman serangan yang sedang terjadi dalam jaringan *server*. IPS digunakan karena mampu mendeteksi ancaman berbahaya dalam jaringan *server* dan memberikan peringatan kepada *administrator* serta mampu memblokir penyusup yang terdeteksi oleh sistem IPS [4].

Penelitian terkait dengan judul Implementasi *Intrusion Prevention System* (IPS) Menggunakan Snort dan Iptables Pada Monitoring Jaringan Lokal Berbasis *Website* [5]. Dimana penelitian ini menerapkan sistem keamanan jaringan lokal dengan menggunakan IPS Snort dan Iptables yang membantu melindungi ancaman serangan DoS *ping of death* dan *port scanning*. Hasil dari penelitian ini sistem yang dibangun berhasil mendeteksi serangan sebesar 90% untuk serangan *ping of death* dan 85% pada serangan *port scanning*.

Penelitian terkait berikutnya dengan judul "Implementasi *Intrusion Prevention System* (IPS) Suricata Sebagai Pengamanan dari Serangan *Distributed Denial Of Service* (DDoS)"[6]. Penelitian ini membangun sistem keamanan *intrusion prevention system* (IPS) menggunakan suricata untuk mencegah serangan *distributed denial of service* (DDoS) yang dikombinasikan menggunakan aplikasi *ELK Stack* sebagai penampil hasil *log* serangan. Hasil dari penelitian ini IPS yang digunakan mampu mendeteksi serangan *DoS* dan mampu memblokir ancaman serangan dengan memanfaatkan fitur *firewall* yaitu Iptables.

Berdasarkan pembahasan sebelumnya, dibuatlah penelitian dengan judul "Implementasi *Intrusion Prevention System* (IPS) Sebagai Sistem Keamanan *Server* Berbasis *Website* dan Aplikasi *Mobile*". Pada penelitian ini dibangun sistem keamanan *server* dengan menerapkan metode *Intrusion Prevention System* (IPS) yang dikombinasikan dengan sistem monitoring berbasis aplikasi *Website,* agar memudahkan *administrator server* dalam melakukan monitoring keamanan serta kondisi kesehatan dari *server*. Ditambahkan sistem yang dapat melakukan tindakan pencegahan terhadap *IP address* penyerang yang berhasil terdeteksi oleh sistem deteksi IPS berbasis aplikasi *Mobile* Android. Pada sistem keamanan *server* ini terdapat sistem notifikasi otomatis yang berfungsi sebagai peringatan dini apabila terjadi serangan oleh penyusup terhadap *server* dengan menggunakan *bot* Telegram. Dengan adanya sistem yang dibuat diharapkan dapat membantu *administrator server* untuk dapat berperan penuh dalam melakukan monitoring lalu lintas jaringan dan mencegah ancaman serangan didalam jaringan *server.*

2. DESAIN PENELITIAN

2.1. Alat dan Bahan

Alat dan bahan yang digunakan untuk keperluan realisasi pada penelitian ini dapat dilihat pada Tabel 1 kebutuhan perangkat keras dan Tabel 2 kebutuhan perangkat lunak. Perangkat keras dan perangkat lunak yang dibutuhkan, kemudian dipersiapkan untuk dilakukan

konfigurasi agar dapat terhubung dengan tujuan dapat saling berkomunikasi didalam satu jaringan *server* yang sama guna menunjang keperluan pada penelitian ini.

Tabel 1. Kebutuhan Perangkat Keras

No	Perangkat Keras	Spesifikasi	Keterangan
1	1 Buah Laptop	Dell Inspiron 3442, Sistem Operasi Ubuntu 20.04, Intel(R) Core (TM) i3-4005U CPU @ 1.70GHz, Ram 8 GB DDR3L, SSD 128 GB.	Digunakan sebagai *Server.*
2	1 Buah Laptop	Asus X453MA, Sistem Operasi Windows 10, Intel(R) Celeron CPU, Ram 4 GB DDR3L, SSD 128 GB, HDD 500GB.	Digunakan sebagai *Client* untuk menyerang *Server*.
3	1 Buah *Acces Point*	TP-Link, TLWA5110G.	Digunakan sebagai penghubung LAN.
4	1 Buah *Router*	TP-Link Wireless WR840N.	Digunakan sebagai penghubung WLAN.
5	1 Buah *Switch*	TP-Link TL-SF1008D.	Digunakan sebagai Penghubung LAN.
6	Kabel UTP dan Konektor RJ 45	Cat 6, 1000Mbps/ 1Gbps, 200 Mhz.	Digunakan sebagai kabel Penghubung.
7	*Smartphone*	Realme C2, Android 9.0, Chipset Mediatek MT6762, CPU Octa-core 2.0 GHz Cortex-A53, Ram 2GB.	Digunakan untuk menerima notifikasi dan melakukan tindakan aksi secara *mobile*.

Tabel 2. Kebutuhan Perangkat Lunak

No	Perangkat Lunak	Keterangan
1	Ubuntu LTS 20.04	Sistem operasi yang digunakan *server*.
2	Suricata	*Tools* yang digunakan sebagai deteksi serangan.
3	Iptables	Digunakan sebagai sistem penanganan serangan.
4	*Framework* Laravel	Digunakan untuk membangun sistem monitoring serangan berbasis *Website*.
5	Flutter	Digunakan untuk membangun aplikasi *Mobile.*
6	Nmap	*Tools attack* untuk melihat *port* yang terbuka pada *server*.
7	Virtualbox	Digunakan untuk menguji sistem operasi *server*.
8	MySQL	Digunakan sebagai *database* penyimpanan data.
9	Telegram	Digunakan sebagai penerima notifikasi otomatis jika *server* diserang.

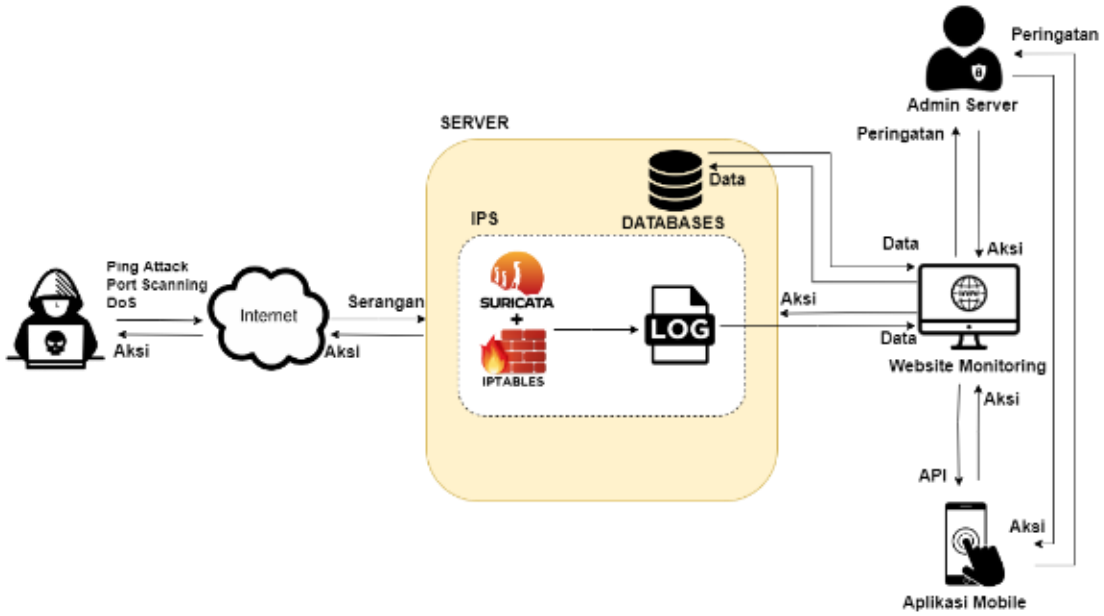
2.2. Metode Penelitian

Metode penelitian yang digunakan dalam penerapan *intrusion prevention system* (IPS) berbasis aplikasi *website* dan *mobile* sebagai sistem keamanan *server* adalah sebagai berikut:

a. Studi Literatur
   Peneliti mengumpulkan dan mempelajari konsep-konsep mengenai kebutuhan penelitian seperti *Intrusion Prevention System* (IPS), *Server*, Analisis QoS, Pengembangan aplikasi *website* dan *mobile* yang bersumber dari buku, *website* dan jurnal.
b. Metode Pengumpulan Data
   Pengumpulan data pada penelitian ini dilakukan dengan melakukan observasi terhadap objek penelitian yaitu *Intrusion Prevention System* (IPS) berbasis aplikasi dalam menangani serangan yang telah ditentukan yaitu *ping attack*, *port scanning* dan *ping of death* (DoS).
c. Analisis Kebutuhan
   Peneliti melakukan analisis kebutuhan guna menunjang keberhasilan penelitian berupa kebutuhan perangkat keras dan perangkat lunak.
d. Perancangan Sistem
   Kebutuhan yang telah dipersiapkan kemudian dilakukan perancangan sedemikian rupa seperti melakukan rancangan sistem IPS, aplikasi *website* dan *mobile* serta rancangan sistem notifikasi otomatis jika *server* mendapat serangan.
e. Implementasi
   Peneliti menerapkan rancangan sistem yang telah dibuat seperti melakukan instalasi, konfigurasi dari perangkat keras dan perangkat lunak.
f. Pengujian
   Peneliti melakukan pengujian terhadap sistem *intrusion prevention system* (IPS) berbasis aplikasi *website* dan *mobile* yang kemudian diambil kesimpulan terhadap sistem yang telah dibangun.

2.3. Perancangan Sistem

Sistem yang dibangun bertujuan sebagai sistem keamanan *server* dengan menggunakan metode *intrusion prevention system* (IPS). Sistem yang telah dibangun kemudian diuji dengan serangan *ping attack*, *port scanning* dan *ping of death* (DoS). Ketika penyusup melakukan serangan terhadap *server*. Sistem IPS akan mendeteksi serangan melalui Suricata yang kemudian aktivitas serangan akan disimpan berupa *log*, data *log* mengandung informasi berupa waktu dan tanggal serangan, sumber IP penyerang, pesan serangan dan klasifikasi serangan. Data *log* serangan ini akan ditampilkan pada aplikasi *website* dan *mobile* untuk dilakukan tindakan lebih lanjut. Tindakan yang dapat dilakukan terbagi menjadi tiga aksi yaitu *Drop*, *Reject* dan *Accept*. Ketika aksi telah dilakukan sistem akan menyimpan data berupa IP penyerang, waktu, tipe dan status aksi yang telah dilakukan kedalam *database* MySQL. *Administrator server* akan mendapatkan notifikasi otomatis yang memberikan informasi telah terjadi serangan pada *server* melalui aplikasi Telegram. Rancangan sistem IPS secara umum dapat dilihat pada Gambar 1.

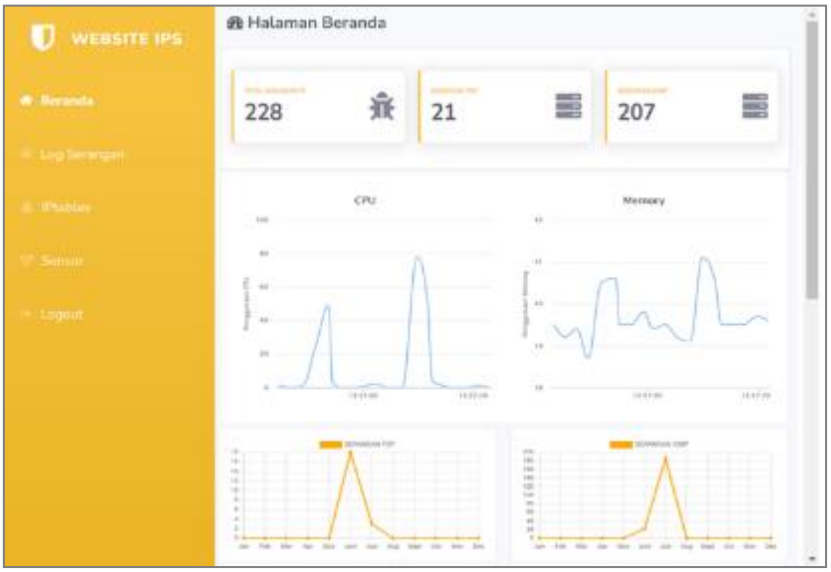


Gambar 1. Perancangan Sistem

3. HASIL DAN PEMBAHASAN

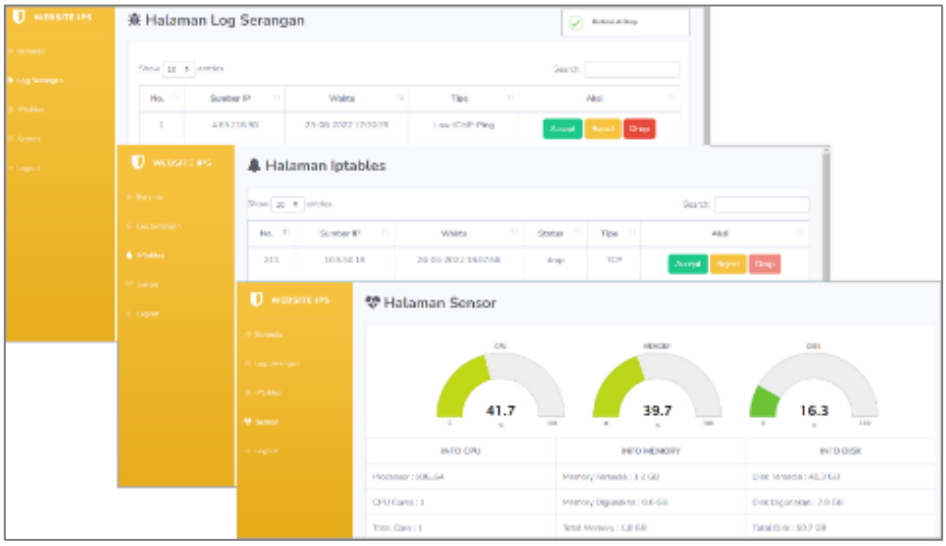
3.1 Implementasi *Website* IPS

Penerapan *website* IPS berfungsi sebagai *interface* yang akan digunakan untuk memudahkan *administrator server* dalam memonitoring keamanan *server*. Terdapat lima *menu* pada aplikasi *website* yaitu beranda, *log* serangan, Iptables, sensor dan *logout* Aplikasi website dimulai dengan antarmuka halaman *login* yang berfungsi sebagai autentikasi pengguna yaitu *administator server* yang tersedia pada *database*. Kemudian setelah berhasil *login* pengguna akan dialihkan pada halaman beranda, halaman beranda merupakan halaman utama ketika pengguna berhasil melakukan *login*, halaman beranda memiliki fungsi menampilkan informasi berupa data jumlah total serangan IP yang telah dilakukan tindakan aksi, informasi jumlah serangan *port* TCP dan ICMP, informasi penggunaan CPU dan *memory* serta grafik *history* aktivitas serangan bulanan yang telah dilakukan tindakan aksi. Halaman beranda pada aplikasi *website* dapat dilihat pada Gambar 2.



Gambar 2. Halaman Beranda.

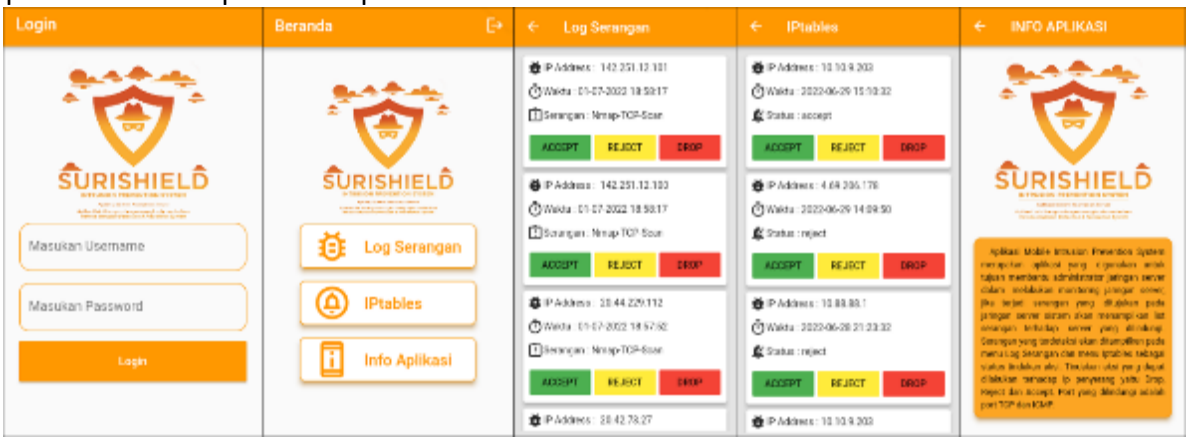
Menu Halaman yang berfungsi untuk melihat data serangan yang tercatat oleh sistem adalah halaman *log* serangan. *Log* serangan yang tercatat meliputi sumber IP penyerang, waktu terjadi serangan, tipe serangan dan aksi yang dapat dilakukan terhadap IP penyerang. Halaman ini merupakan menu utama yang berfungsi untuk melakukan Aksi terhadap IP penyerang, Aksi terbagi menjadi *Drop*, *Reject* dan *Accept*. Ketika aksi berhasil dilakukan, data akan tersimpan pada database dan pengguna akan dialihkan ke halaman Iptables untuk melihat status Aksi. Pengguna juga dapat melihat kinerja *server* pada *server* yang mencakup CPU, *memory* dan *disk* pada halaman sensor. Halaman tersebut dapat dilihat pada Gambar 3.



Gambar 3. Halaman *Log*, Iptables dan Sensor.

3.1.1. Implementasi Aplikasi *Mobile*

Penerapan aplikasi *mobile* berfungsi bagi *administrator server* untuk dapat melakukan aksi terhadap IP penyerang yang berhasil terdeteksi oleh sistem secara *mobile*. Aplikasi dimulai dengan menu halaman *login* pengguna yaitu *administrator server*. Setelah berhasil melakukan *login username* dan *password* pengguna, *admin* akan dialihkan pada halaman beranda, halaman ini menampilkan tiga *menu* halaman yang pertama yaitu *log* serangan yang berfungsi untuk melakukan aksi terhadap penyerang yang berhasil terdeteksi sistem, kemudian *menu* Iptables yaitu *menu* yang berfungsi untuk melihat status aksi yang telah dilakukan sebelumnya, kemudian info aplikasi yang berfungsi sebagai halaman yang memberikan informasi fungsi dari aplikasi *mobile* yang telah dibuat. Hasil Implementasi aplikasi *mobile* dapat dilihat pada Gambar 4.



Gambar 4. *Menu* Pada Aplikasi *Mobile*

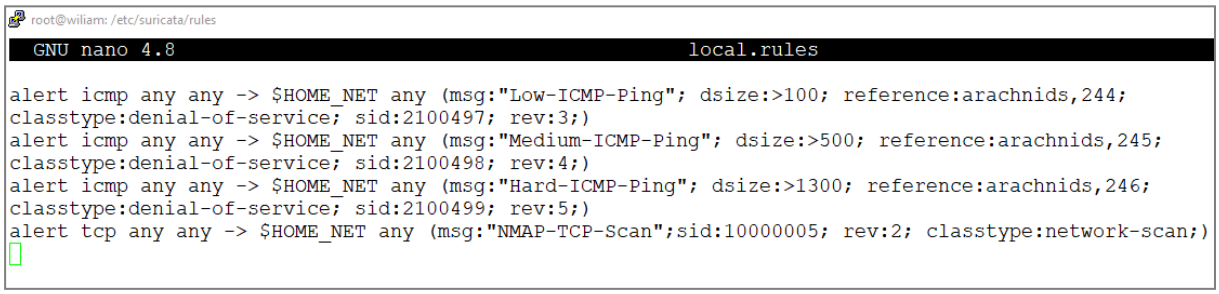
3.1.2. Implementasi Notifikasi Otomatis

Penerapan aplikasi Telegram digunakan untuk menerima notifikasi jika terjadi serangan pada *server* yang berhasil terdeteksi oleh sistem. Sistem akan mengirimkan notifikasi terjadi serangan melalui aplikasi Telegram sebagai informasi terjadi ancaman serangan terhadap *server*. Notifikasi berisi informasi waktu sistem mendeteksi serangan dan nama tipe serangan yang dilakukan oleh penyerang. Hasil implementasi notifikasi dapat dilihat pada Gambar 5.

Gambar 5. Notifikasi Terjadi Serangan

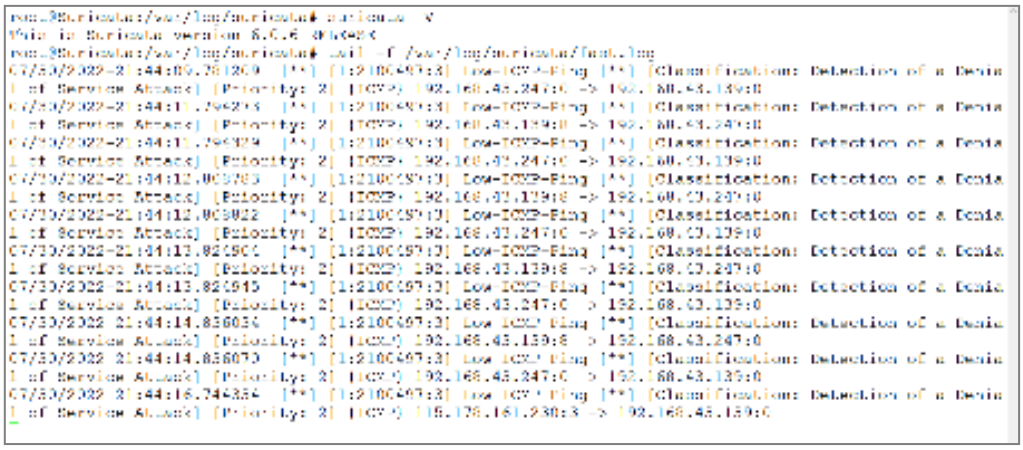
3.2. Pengujian Sistem

Log serangan yang ditampilkan pada aplikasi *website* dan *mobile* merupakan *output* dari keberhasilan deteksi serangan oleh sistem Suricata. Dilakukan pengujian beberapa kali untuk menguji apakah sistem dapat bekerja dengan baik saat terjadi serangan didalam jaringan server. Pengujian dimulai dengan penerapan *rule* serangan dan sistem deteksi atau *intrusion detection system* (IDS) suricata. Penerapan *rule* serangan berfungsi untuk mendeteksi *port* yang menjadi target serangan. *Rule* merupakan aturan pada sistem suricata yang akan selalu terpanggil pada saat sistem suricata dijalankan. Penerapan *rule* serangan dapat dilihat pada Gambar 6.



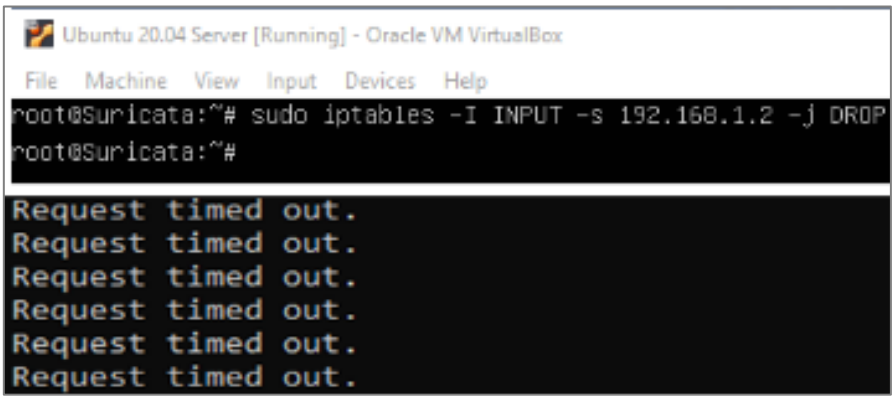
Gambar 6. *Rule* Serangan.

Sistem suricata dapat berjalan dengan baik dalam mendeteksi adanya serangan terhadap *server*. Suricata menghasilkan *log* atau *output* berupa *alert* terjadi serangan oleh penyusup dengan memberikan detail informasi berupa waktu serangan, jenis serangan, *port*, dan sumber IP penyerang. *Alert* yang dihasilkan dalam bentuk *console* tersebut menyesuaikan dengan *rule* serangan yang telah diterapkan. Hasil dari deteksi serangan dapat dilihat pada Gambar 7.



Gambar 7. Suricata *Mode Console*.

Informasi *log* serangan tersebut kemudian dapat dilakukan tindakan berupa aksi dengan memanfaatkan Iptables, tindakan aksi yang dapat dilakukan terbagi menjadi *drop*, *reject* dan *accept*. Salah satu penerapan Iptables terhadap IP penyerang yang berhasil terdeteksi oleh sistem dapat dilihat pada Gambar 8.



Gambar 8. Aksi *Drop* dan Hasil yang didapatkan oleh penyerang.

3.3. Pengujian Serangan Pada *Server* Lokal

Tahapan pengujian serangan pada *server* lokal terbagi menjadi tiga serangan yaitu *ping attack*, *port scanning* dan *ping of death* (DoS). Serangan tersebut dilakukan pada *server* lokal dengan tujuan mengetahui dampak serangan terhadap kinerja *server* lokal. Pengujian dilakukan berulang sebanyak 20 kali untuk mendapatkan hasil data serangan yang baik.

3.3.1. Serangan *Ping Attack*

Ping Attack merupakan teknik serangan yang dilakukan dengan melakukan ping terhadap *server*, tindakan ini dilakukan dengan tujuan untuk mengetahui apakah target dari *server* dalam keadaan aktif atau tidak [7]. Pengujian dilakukan dengan cara melakukan *request* ping pada alamat IP *server*. Hasil dari *ping attack* dapat dilihat pada Tabel 3.

Tabel 3. Serangan *Ping Attack*

No.	Sumber IP	Tujuan	Kondisi	*Port*	Kinerja *server* naik hingga (%)		
					CPU	***Memory***	***Disk***
1	103.154.110.50	10.110.0.183	IDS	*ICMP*	21,6	41,4	18,7
2	103.154.110.50	10.110.0.183	IDS	*ICMP*	16,4	41,3	18,7
3	103.154.110.50	10.110.0.183	IDS	*ICMP*	13	41,4	18,7
4	103.154.110.50	10.110.0.183	IDS	*ICMP*	17,5	41,4	18,7
5	103.154.110.50	10.110.0.183	IDS	*ICMP*	22,2	41,4	18,7
6	103.154.110.50	10.110.0.183	IDS	*ICMP*	11,9	41,4	18,7
7	103.154.110.50	10.110.0.183	IDS	*ICMP*	8,5	41,5	18,7
8	103.154.110.50	10.110.0.183	IDS	*ICMP*	17,8	41,4	18,7
9	103.154.110.50	10.110.0.183	IDS	*ICMP*	17,9	41,4	18,7
10	103.154.110.50	10.110.0.183	IDS	*ICMP*	8,5	41,4	18,7
11	103.154.110.50	10.110.0.183	IDS	*ICMP*	17,4	41,7	19,6
12	103.154.110.50	10.110.0.183	IDS	*ICMP*	9,3	41,1	19,6
13	103.154.110.50	10.110.0.183	IDS	*ICMP*	11,4	41,4	19,6
14	103.154.110.50	10.110.0.183	IDS	*ICMP*	14,9	41,2	19,6
15	103.154.110.50	10.110.0.183	IDS	*ICMP*	15,9	41,2	19,6
16	103.154.110.50	10.110.0.183	IDS	*ICMP*	17,5	41,1	19,6
17	103.154.110.50	10.110.0.183	IDS	*ICMP*	18,2	41,2	19,6
18	103.154.110.50	10.110.0.183	IDS	*ICMP*	11,9	41,2	19,6
19	103.154.110.50	10.110.0.183	IDS	*ICMP*	16,7	41,2	19,6
20	103.154.110.50	10.110.0.183	IDS	*ICMP*	11,6	41,2	19,6
Rata-Rata Kenaikan (%)					15	41,3	19,1

Berdasarkan pengujian serangan *ping attack* yang telah dilakukan sebanyak 20 kali terhadap komputer *server* lokal, didapatkan nilai kenaikan tertinggi penggunaan *CPU* dari *server* menjadi 21,6% dengan nilai rata-rata dari total pengujian sebesar 15%, Kenaikan tertinggi penggunaan *memory* adalah 41,7% dengan nilai rata-rata sebesar 41,3%, dan kenaikan tertinggi pada *disk* adalah 19,6% dengan nilai rata-rata sebesar 19,15%.

3.3.2. Serangan *Port Scanning*

Port Scanning merupakan jenis serangan yang dapat digunakan untuk mendapatkan informasi dari *server* seperti status *port* yang terbuka dan layanan yang dijalankan pada *port* layanan di *server* [8]. Serangan *port scanning* dilakukan dengan menggunakan *tools* zenmap melalui komputer *client* yang berperan sebagai penyerang. Hasil dari pengujian serangan *port scanning* dapat dilihat pada Tabel 4.

Tabel 4. Serangan *Port Scanning*

No.	Sumber IP	Tujuan	Hasil *Port Scanning*	Kinerja *server* naik hingga (%)		
			TCP *Port* 80 dan 22	*CPU*	*Memory*	*Disk*
1	103.154.110.50	10.110.0.183	✔	46,8	41,8	19,4
2	103.154.110.50	10.110.0.183	✔	65,4	40,6	19,4
3	103.154.110.50	10.110.0.183	✔	63,5	41,6	19,4
4	103.154.110.50	10.110.0.183	✔	42,2	41,6	19,4
5	103.154.110.50	10.110.0.183	✔	30,6	40,6	19,4
6	103.154.110.50	10.110.0.183	✔	56,8	41,7	19,7
7	103.154.110.50	10.110.0.183	✔	62,1	41,6	19,7
8	103.154.110.50	10.110.0.183	✔	42,7	40,6	19,7
9	103.154.110.50	10.110.0.183	✔	70,9	41,7	19,7
10	103.154.110.50	10.110.0.183	✔	60,8	40,6	19,7
11	103.154.110.50	10.110.0.183	✔	68,2	41,7	19,7
12	103.154.110.50	10.110.0.183	✔	79,5	41,7	19,7
13	103.154.110.50	10.110.0.183	✔	70,7	41,7	19,7
14	103.154.110.50	10.110.0.183	✔	66,4	40,7	20,3
15	103.154.110.50	10.110.0.183	✔	72,9	41,7	20,3
16	103.154.110.50	10.110.0.183	✔	58,2	41,7	20,3
17	103.154.110.50	10.110.0.183	✔	85,9	41,9	20,3
18	103.154.110.50	10.110.0.183	✔	69,3	41,7	20,3
19	103.154.110.50	10.110.0.183	✔	65,5	41,7	20,3
20	103.154.110.50	10.110.0.183	✔	85,9	41,9	20,3
	Rata-Rata Kenaikan (%)			62,2	41,3	19,8

Berdasarkan pengujian yang dilakukan oleh *client* sebagai penyerang dengan melakukan serangan *port scanning (intense scan)*. Pengujian serangan dilakukan sebanyak 20 kali untuk setiap serangan, didapat hasil dari pengujian *intense scan* pada *server* berupa kenaikan kinerja CPU dengan menjadi 85,9% dengan rata-rata dari total pengujian adalah 62,2%. Kenaikan tertinggi penggunaan *memory* adalah 41,9% dengan nilai rata-rata sebesar 41,3%, dan kenaikan tertinggi pada *disk* adalah 20,3% dengan nilai rata-rata sebesar 19,8%. Setiap hasil dari *scanning* didapat informasi *port* yang terbuka yaitu TCP 80 dan TCP 22.

3.3.3. Serangan *Ping of Death* (DoS)

Serangan *Ping of Death* merupakan salah satu jenis serangan DoS, Pada pengujian serangan ini dilakukan dengan cara melakukan *request* paket secara berkala dengan beban paket secara besar ke *server server* lokal [9]. Perintah yang digunakan adalah "Ping -I 65500 (alamat IP *server*)", perintah ini memberikan beban *request* paket yang maksimal sebesar 65500 *bytes* menuju *server*. Hasil dari pengujian *ping of death* dapat dilihat pada Tabel 5.

Tabel 5. Serangan *Ping of Death* (DoS)

No.	Sumber IP	Tujuan	Kondisi	Beban paket dikirim ke *Server*	Kinerja *server* naik hingga (%)		
				Port ICMP	*CPU*	*Memory*	*Disk*
1	103.154.110.50	10.110.0.183	IDS	65500	76,1	41,6	20,3
2	103.154.110.50	10.110.0.183	IDS	65500	81	42,7	20,3
3	103.154.110.50	10.110.0.183	IDS	65500	72,9	41,7	20,3
4	103.154.110.50	10.110.0.183	IDS	65500	78,4	42,7	20,4
5	103.154.110.50	10.110.0.183	IDS	65500	84,5	41,7	20,4
6	103.154.110.50	10.110.0.183	IDS	65500	90,4	42,8	20,5
7	103.154.110.50	10.110.0.183	IDS	65500	89,3	42,8	20,5
8	103.154.110.50	10.110.0.183	IDS	65500	81,5	42,8	20,5
9	103.154.110.50	10.110.0.183	IDS	65500	85,7	41,8	20,6
10	103.154.110.50	10.110.0.183	IDS	65500	86,9	42,9	20,6
11	103.154.110.50	10.110.0.183	IDS	65500	75,7	41,8	20,6
12	103.154.110.50	10.110.0.183	IDS	65500	66,7	42,9	20,7
13	103.154.110.50	10.110.0.183	IDS	65500	75,7	41,9	20,7
14	103.154.110.50	10.110.0.183	IDS	65500	68,9	41,9	20,7
15	103.154.110.50	10.110.0.183	IDS	65500	69,4	41,8	20,7
16	103.154.110.50	10.110.0.183	IDS	65500	78,2	41,9	20,7
17	103.154.110.50	10.110.0.183	IDS	65500	66,2	41,9	20,8
18	103.154.110.50	10.110.0.183	IDS	65500	75	41,8	20,8
19	103.154.110.50	10.110.0.183	IDS	65500	67,1	41,8	20,8
20	103.154.110.50	10.110.0.183	IDS	65500	71,6	41,8	20,8
Rata-Rata Kenaikan (%)					77	42,1	20,5

Berdasarkan pengujian yang telah dilakukan didapatkan nilai kenaikan tertinggi kinerja CPU dari *server* lokal adalah sebesar 90,4% dengan nilai rata-rata dari total pengujian sebesar 77%. Kenaikan tertinggi penggunaan *memory* adalah 42,9% dengan nilai rata-rata pengujian sebesar 42,1%, dan kenaikan tertinggi pada *disk* adalah 20,8% dengan nilai rata-rata pengujian sebesar 20,5%.

3.4. Pengujian Notifikasi Otomatis

Dilakukan pengujian notifikasi telegram yang telah terintegrasi dengan sistem pada *server* sebanyak 20 kali. Pengujian dilakukan dengan tujuan mendapatkan selisih waktu ketika terjadi

serangan terhadap *server* sampai dengan mendapatkan notifikasi melalui aplikasi Telegram. Pengujian dilakukan pada *server* lokal yang telah terintegrasi sistem IPS. Hasil pengujian waktu notifikasi otomatis menggunakan Telegram dapat dilihat pada Tabel 6.

Tabel 6. Notifikasi Otomatis Melalui Telegram.

No.	Sumber IP	IP Tujuan	Waktu (Detik)	Keterangan
1	10.30.11.4	10.30.11.34	3,91	Berhasil Terkirim Notifikasi
2	10.30.11.4	10.30.11.34	3,66	Berhasil Terkirim Notifikasi
3	10.30.11.4	10.30.11.34	4,09	Berhasil Terkirim Notifikasi
4	10.30.11.4	10.30.11.34	3,76	Berhasil Terkirim Notifikasi
5	10.30.11.4	10.30.11.34	2,69	Berhasil Terkirim Notifikasi
6	10.30.11.4	10.30.11.34	3,07	Berhasil Terkirim Notifikasi
7	10.30.11.4	10.30.11.34	3,47	Berhasil Terkirim Notifikasi
8	10.30.11.4	10.30.11.34	3,45	Berhasil Terkirim Notifikasi
9	10.30.11.4	10.30.11.34	2,53	Berhasil Terkirim Notifikasi
10	10.30.11.4	10.30.11.34	3,28	Berhasil Terkirim Notifikasi
11	10.30.11.4	10.30.11.34	3,99	Berhasil Terkirim Notifikasi
12	10.30.11.4	10.30.11.34	3,41	Berhasil Terkirim Notifikasi
13	10.30.11.4	10.30.11.34	3,91	Berhasil Terkirim Notifikasi
14	10.30.11.4	10.30.11.34	2,49	Berhasil Terkirim Notifikasi
15	10.30.11.4	10.30.11.34	2,62	Berhasil Terkirim Notifikasi
16	10.30.11.4	10.30.11.34	3,61	Berhasil Terkirim Notifikasi
17	10.30.11.4	10.30.11.34	3,61	Berhasil Terkirim Notifikasi
18	10.30.11.4	10.30.11.34	4,14	Berhasil Terkirim Notifikasi
19	10.30.11.4	10.30.11.34	3,54	Berhasil Terkirim Notifikasi
20	10.30.11.4	10.30.11.34	3,01	Berhasil Terkirim Notifikasi
Rata-Rata Waktu (Detik)			3,41	Berhasil Terkirim Notifikasi

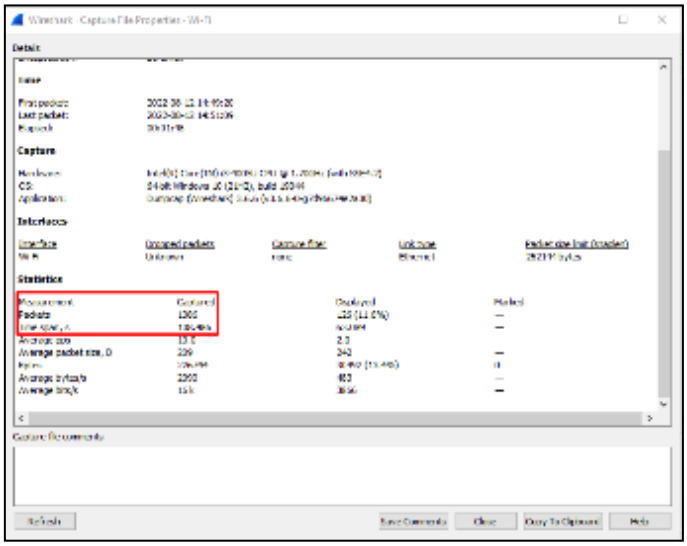
Berdasarkan pengujian yang telah dilakukan sebanyak 20 kali, didapatkan hasil nilai rata-rata pengujian notifikasi berhasil terdeteksi dan terkirim sebesar 3,41 detik. Notifikasi yang terkirim memberikan informasi berupa waktu terjadi serangan, tanggal, sumber IP penyerang dan klasifikasi serangan.

Gambar 8. Notifikasi Berhasil Terkirim.

3.5. *Analysis QoS*

Analisis *Quality of Service* (QoS) merupakan metode pengukuran yang digunakan untuk menentukan kemampuan suatu jaringan dalam memberikan layanan jaringan yang baik

untuk memenuhi kebutuhan suatu layanan [10]. Analisis QoS dilakukan untuk menilai kualitas jaringan yang digunakan dan untuk melihat respons sistem dari *server* terhadap serangan dari penyusup. Pada pengujian ini analisis dilakukan terhadap parameter *Latency* atau *Delay*. *Tools* yang digunakan untuk mengukur QoS adalah aplikasi Wireshark dan persamaan untuk menghitung *latency* yaitu dengan membagi *time span* dan jumlah paket yang berhasil di *capture* oleh Wireshark. Hasil *capture* dari Wireshark pada jaringan yang telah digunakan dilihat pada Gambar 9.



Gambar 9. Hasil *Capture* Paket Wireshark

Latency atau *delay* adalah waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. *Latency* dapat dipengaruhi oleh jarak, media fisik, kongesti atau waktu proses yang lama. Dari tangkapan data yang telah dilakukan dengan Wireshark, maka didapatkan rata-rata *latency* dengan perhitungan sebagai berikut:

$$Latency = \frac{108{,}456}{1086} = 0{,}0998\ s = 99{,}89\ ms$$

Berdasarkan parameter *latency* dari ETSI-TIPHON, besaran *latency* atau *delay* dapat diklasifikasikan sebagai kategori *latency* sangat baik jika <150 ms, bagus jika 150 ms sampai dengan 300 ms, sedang jika 300 ms sampai dengan 450 ms dan buruk jika >450 ms [11]. Didapatkan nilai waktu respon dari sistem sebesar 99,89 ms. Nilai tersebut didapatkan dari hasil perhitungan membagi nilai *time span* dan *packets* berdasarkan data yang berhasil ditangkap menggunakan *tools* Wireshark sebelumnya. Dari hasil yang telah didapatkan menunjukan sistem yang telah dibangun memiliki keandalan yang sangat baik dalam mendeteksi ancaman serangan masuk.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, maka didapat kesimpulan sebagai berikut:

1. Aksi Iptables pada aplikasi berhasil menangani serangan dengan kecepatan rata-rata pada aplikasi *website* adalah 2,85 detik untuk aksi *drop*, 2,59 detik untuk aksi *reject* dan 2,56 detik untuk aksi *accept*. Kemudian aksi Iptables pada aplikasi *mobile* berjalan dengan kecepatan rata-rata 4,53 detik untuk aksi *drop*, 4,48 detik untuk aksi *reject* dan 3,74 detik untuk aksi *accept*. Pengujian keandalan sistem dilakukan dengan menghitung *latency time* sistem dalam merespon serangan yang masuk. Hasil pengujian didapat latency time sistem

dikategorikan sangat baik dalam merespon serangan yang masuk dengan nilai kecepatan respon serangan sebesar 99,89 ms.

2. Serangan *ping attack*, *port scanning* dan *ping of death* (DoS) mengakibatkan peningkatan beban kinerja *server*. Pada *server* lokal rata-rata kinerja awal CPU berkisar 10-19%, saat terjadi serangan *ping attack* terjadi kenaikan kinerja CPU menjadi 21,6%, *memory* menjadi 41,7%, dan *disk* 19,6%. Serangan *port scanning* dengan mode *intense scan* terjadi kenaikan CPU menjadi 85,9%, *memory* menjadi 41,9% dan *disk* 20,3%. Serangan *ping of death* terjadi kenaikan kinerja CPU menjadi 90,4%, *memory* menjadi 42,9% dan *disk* 20,8%. Berdasarkan pengujian serangan yang telah dilakukan peningkatan berlebihan terdapat pada serangan *ping of death* yang mengakibatkan kinerja pada *server* lokal meningkat menjadi 90,4%, jika serangan terjadi dalam waktu cukup lama maka kondisi *server* menjadi *hang* (rusak).
3. Notifikasi otomatis berjalan secara *realtime* melalui *bot* Telegram dan berhasil mengirimkan *alert* yang bersumber dari hasil deteksi sistem IDS Suricata melalui Telegram. Ketika mendapatkan notifikasi, *administrator server* dapat melakukan tindakan dengan cepat untuk mengatasi gangguan serangan yang terjadi dengan nilai rata-rata kecepatan notifikasi berhasil terkirim dan diterima 3,41 detik.

REFERENSI

[1] Badan Siber dan Sandi Negara, *Laporan Tahun 2020 Honeynet Project BSSN - IHP*. 2020.
[2] Z. Husen and M. S. Surbakti, Membangun *Server* dan Jaringan Komputer Dengan *Linux Ubuntu*. Aceh: Syiah Kuala University Press, 2020.
[3] N. Chakraborty, "*Intrusion Detection system and Intrusion Prevention sytem : a Comparative Study*," *Int. J. Comput. Bus. Res.*, vol. 4, no. 2, 2013.
[4] W. Ma'ruf K, "Perancangan dan Implementasi IPS (*Intrusion Prevention System*) Sebagai Pengamanan Jaringan Komputer Berbasis *Snort Inline*," *Univ. Amikom Yogyakarta*, vol. 85, no. 1, pp. 2071–2079, 2016.
[5] R. Suwanto, I. Ruslianto, and M. Diponegoro, "Implementasi *Intrusion Prevention System* (IPS) Menggunakan Snort Dan IPTable Pada Monitoring Jaringan Lokal Berbasis *Website*," *J. Komput. dan Apl.*, vol. 07, no. 1, pp. 97–107, 2019.
[6] I. Adesty, W. A. Prabowo, and M. F. Sidiq, "Penerapan *Intrusion Prevention System* (IPS) Suricata Sebagai Pengamanan Dari Serangan *Distributed Denial of Service* (DDoS)," *Eeasy Chair Prepr.*, p. 2912, 2020.
[7] A. F. Mutaqin, "Rancang Bangun Sistem Monitoring Keamanan Jaringan Prodi Teknik Informatika Melalui SMS *Alert* dengan Snort," *J. Sist. dan Teknol. Inf.*, vol. 1, no. 1, 2016.
[8] S. Sinha, *Beginning Ethical Hacking With Kali Linux : Computational Techniques for Resolving Security Issues*. Apress Publisher, 2018.
[9] D. K. Bhattacharyya and J. K. Kalita, *DDoS Attacks : Evolution, Detection, Prevention, Reaction, and Tolerance*. Boca Raton, London, New York, 2016.
[10] ETSI, *Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); General aspects of Quality of Service (QoS)*, vol. 1. 1999.
[11] H. A. Saputra, P. Pohny, and G. M. Putra, "Analisis QOS Jaringan 4G Dengan Menggunkan Aplikasi Wireshark (Studi Kasus: Tepian Samarinda, Taman Samarinda, dan Taman Cerdas)," *Semin. Ilmu Komput. dan Teknol. Inf.*, vol. 5, no. 1, pp. 13–18, 2020.