Jurnal Bumigora Information Technology (BITe)
Vol. 7, No. 2, Desember 2025, Hal. 95∼108
E-ISSN: 2685-4066, DOI: 10.30812/bite.v7i2.5226

Pengujian Efektivitas *Intrusion Detection Systems* (IDS) Snort, Suricata, dan Zeek terhadap Serangan SYN Flood

Testing the Effectiveness of Intrusion Detection Systems (IDS) Snort, Suricata, and Zeek against SYN Flood Attacks

I Nyoman Bagus Arya Wirianda*, **Raphael Bianco Huwae**, **Andy Hidayat Jatmika**
Universitas Mataram, Mataram, Indonesia

Informasi Artikel:
Diterima: 19 Juni 2025, Direvisi: 13 Agustus 2025, Disetujui: 4 Desember 2025

Abstrak-
Latar Belakang: Keamanan jaringan adalah aspek penting dalam pengelolaan infrastruktur TI, dengan ancaman utama berupa serangan Denial of Service (DoS), khususnya SYN Flood.
Tujuan: Tujuan penelitian ini adalah untuk mengevaluasi efektivitas tiga Intrusion Detection System (IDS) yaitu Snort, Suricata, dan Zeek dalam mendeteksi serangan TCP SYN Flood. Lingkungan pengujian menggunakan Windows Server 2022 sebagai sistem target untuk mensimulasikan kondisi nyata pada jaringan produksi.
Metode: Penelitian ini menggunakan metode eksperimental yang mencakup tahapan identifikasi masalah, analisis, desain/perancangan, implementasi, pengujian, dan analisis hasil.
Hasil: Hasil penelitian ini adalah Snort memiliki performa tertinggi dalam deteksi serangan, dengan rata-rata 68.25%, diikuti oleh Suricata 61.08% dan Zeek 55.77%. Dalam penggunaan CPU, Snort juga unggul dengan rata-rata 16.3%, sementara Suricata dan Zeek masing-masing menggunakan 24.5% dan 21.7%. Untuk penggunaan RAM, Snort mencatat rata-rata 18.2%, diikuti oleh Zeek 16.6% dan Suricata 24.5%.
Kesimpulan: Kesimpulan pada penelitian ini adalah Snort lebih unggul dalam deteksi jaringan dan efisiensi CPU, sedangkan Zeek lebih efisien dalam penggunaan RAM, dan Suricata memiliki performa menengah dengan penggunaan sumber daya tertinggi.

Kata Kunci: Deteksi Serangan; Efisiensi Sistem; *Intrusion Detection System* (IDS); Keamanan Jaringan; SYN Flood.

Abstract-
Background: *Network security is an essential aspect of IT infrastructure management, with the main threat being Denial-of-Service (DoS) attacks, particularly SYN Flood attacks.*
Objective: *The purpose of this study is to evaluate the effectiveness of three Intrusion Detection Systems (IDS), namely Snort, Suricata, and Zeek, in detecting TCP SYN Flood attacks. The testing environment uses Windows Server 2022 as the target system to simulate real-world conditions on a production network.*
Methods: *This study employs an experimental method comprising the following stages: problem identification, analysis, design/development, implementation, testing, and results analysis.*
Result: *This study shows that Snort performs best in attack detection, with an average of 68.25%, followed by Suricata at 61.08% and Zeek at 55.77%. In terms of CPU usage, Snort also leads with an average of 16.3%, while Suricata and Zeek use 24.5% and 21.7%, respectively. For RAM usage, Snort recorded an average of 18.2%, followed by Zeek at 16.6% and Suricata at 24.5%.*
Conclusion: *This study concludes that Snort is superior in network detection and CPU efficiency. At the same time, Zeek is more efficient with RAM usage, while Suricata has average performance and the highest resource usage.*

Keywords*: Attack Detection; System Efficiency; Intrusion Detection System (IDS); Network Security; SYN Flood.*

Penulis Korespondensi:

I Nyoman Bagus Arya Wirianda,
Program Studi Teknik Informatika, Universitas Mataram,
Email: aryawirianda@gmail.com

How to Cite: I. N. B. A. Wirianda, R. B. Huwae, & A. H. Jatmika, "Pengujian Efektivitas *Intrusion Detection Systems* (IDS) Snort, Suricata, dan Zeek terhadap Serangan SYN Flood," *Jurnal Bumigora Information Technology (BITe)*, vol. 7, no. 2, pp. 95–108, Des 2025. DOI: 10.30812/bite.v7i2.5226.

1. PENDAHULUAN

Dalam era digital saat ini, keamanan siber dapat diartikan sebagai tantangan besar di Indonesia karena dapat membuat serangan terhadap infrastruktur Teknologi Informasi (TI) semakin meningkat di berbagai sektor seperti pemerintahan, perbankan, dan kesehatan [1]. Sistem Keamanan Jaringan merupakan bentuk dari upaya pencegahan dan pengidentifikasi pengguna jaringan yang tidak sah (penyusup) dari suatu lingkup jaringan. Pencegahan itu dapat menghentikan pengguna yang tidak sah tersebut untuk mengakses setiap hal dalam jaringan yang disusupinya. Sehingga sistem keamanan bisa dikatakan sebagai kunci untuk memastikan stabilitas jaringan dan efektivitas data pengguna [2]. *Denial of Service* (DoS) adalah adalah sebuah serangan yang menghabiskan *resource* komputer target sehingga tidak dapat diakses. Serangan DoS dapat dilakukan dengan cara membanjiri *traffic* jaringan dengan banyak data [3]. Salah satu serangan DoS adalah TCP SYN Flood, penyerang akan melakukan serangan pada layer jaringan karena sebuah sistem yang minim perlindungan [4]. Untuk melindungi sistem dari serangan tersebut, penggunaan *Intrusion Detection System* (IDS) menjadi sangat penting. *Intrusion Detection System* atau IDS adalah sebuah sistem yang melakukan pengawasan terhadap traffic jaringan dan pengawasan terhadap kegiatan-kegiatan yang mencurigakan didalam sebuah sistem jaringan [5]. Ada beberapa jenis IDS yang banyak digunakan seperti *Snort*, *Suricata*, *OSSEC*, *Sagan*, *Zeek*, *Solar Winds Logs & Event Manager*, *Open WIP*S dan lain sebagainya, masing-masing sistem ini memiliki keunggulan dan kelemahan tersendiri dalam mendeteksi dan merespons serangan [6].

Tinjauan pustaka menunjukkan bahwa sejumlah penelitian telah dilakukan untuk mengevaluasi performa sistem keamanan jaringan, khususnya dalam mendeteksi serangan SYN Flood menggunakan *Intrusion Detection System* (IDS). Penelitian yang dilakukan oleh Lukman dan tim [7] pada tahun 2020 mengenai Analisis Perbandingan Kinerja Snort Dan Suricata Sebagai *Intrusion Detection System* Dalam Mendeteksi Serangan SYN Flood Pada Web Server Apache. Hasil dari penelitian tersebut yaitu Snort kemampuan deteksi serangan SYN Flood lebih tinggi dibanding Suricata. Selanjutnya, penelitian Kalabo et al. [8] melakukan analisis performa TCP SYN Flood pada IDS Snort dan Suricata menggunakan *packet generator Scapy*, dan menemukan bahwa Snort unggul dalam akurasi dan kecepatan deteksi. Bada et al. [9] melakukan analisis menyeluruh terhadap Snort, Suricata, dan Bro (Zeek) dengan mengukur akurasi, false positive/negative, dan *throughput* pada berbagai serangan DoS, yang menggambarkan performa masing masing sistem dalam konteks realistik. Perdigón Llanes [10] menambahkan temuan bahwa Snort lebih hemat CPU, sementara Suricata unggul efisiensi deteksi dalam konteks probe dan DoS (termasuk SYN Flood).

Meskipun berbagai penelitian sebelumnya telah membandingkan performa *Intrusion Detection System* (IDS) seperti Snort, Suricata, dan Zeek, sebagian besar studi tersebut memiliki keterbatasan dalam ruang lingkup, baik dari sisi jumlah sistem yang diuji maupun parameter evaluasi yang digunakan. Beberapa penelitian hanya membandingkan dua IDS, tanpa menyertakan Zeek yang semakin populer karena kemampuan analisis berbasis log dan visibilitas trafik yang mendalam. Selain itu, kebanyakan penelitian menggunakan sistem target berbasis Linux atau skenario simulasi terbatas, tanpa mempertimbangkan lingkungan Windows Server yang umum digunakan di jaringan produksi nyata. Terdapat gap atau kesenjangan yang belum diselesaikan oleh penelitian sebelumnya yaitu belum adanya perbandingan komprehensif antara Snort, Suricata, dan Zeek pada target Windows Server 2022 dengan pengukuran performa yang mencakup deteksi serangan dan efisiensi penggunaan sumber daya (CPU dan RAM) secara bersamaan. Perbedaan penelitian ini dengan sebelumnya adalah pengujian dilakukan pada ketiga IDS *open-source* tersebut dalam skenario serangan TCP SYN Flood terhadap Windows Server 2022. Tujuan penelitian ini adalah mengevaluasi dan membandingkan performa Snort, Suricata, dan Zeek dalam mendeteksi serangan TCP SYN Flood serta mengukur efisiensi penggunaan sumber daya pada lingkungan Windows Server 2022. Kontribusi penelitian ini adalah memberikan acuan berbasis data bagi praktisi dan peneliti dalam memilih IDS yang sesuai berdasarkan kebutuhan performa deteksi dan efisiensi sumber daya, sekaligus memperkaya literatur terkait implementasi IDS pada platform Windows Server yang masih jarang dibahas, sehingga mendukung pengembangan sistem keamanan jaringan yang lebih tangguh dan relevan dengan kebutuhan infrastruktur modern.

2. METODE PENELITIAN

Identifikasi masalah merupakan tahap awal dalam melakukan penelitian, dengan mengidentifikasi masalah kita bisa mengetahui masalah apa yang akan kita bahas dalam penelitian ini [11]. Permasalahan utama dalam penelitian ini adalah efektivitas IDS Snort, Suricata, dan Zeek dalam mendeteksi serangan TCP SYN Flood pada Windows Server 2022. Serangan SYN Flood mengeksploitasi proses *three-way handshake* dengan mengirim paket SYN berlebihan tanpa menyelesaikan koneksi, sehingga menghabiskan sumber daya server dan mengganggu layanan [12]. Karena kompleksitas serangan yang meningkat, diperlukan evaluasi kinerja ketiga IDS tersebut dalam mendeteksi dan merespons serangan SYN Flood pada Windows Server 2022.

2.1. Analisis Kebutuhan

Proses analisis permasalahan terkait dengan penelitian untuk menentukan kebutuhan sistem, termasuk perangkat keras dan perangkat lunak. Perangkat keras yang dibutuhkan dalam penelitian ini dalam menjalankan fungsi IDS dapat dilihat pada Tabel 1. Kemudian perangkat lunak yang dibutuhkan dalam sistem bisa dilihat pada Tabel 2.

Tabel 1. Kebutuhan Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	*Processor*	Intel(R) Core(TM) i3-1005G1 CPU @ 1.20GHz 1.20 GHz
2	*Memory*	4 GB RAM
3	Penyimpanan	SSD 256 GB
4	VGA/*Screen*	Intel® UHD Graphics

Tabel 2. Kebutuhan Perangkat Lunak

No	Perangkat Lunak	Keterangan
1	VirtualBox versi 7.1.8	*Software* virtualiasi
2	Kali Linux versi 2025.1a	Sistem operasi penyerang (*attacker*)
3	Windows Server 2022	Sistem operasi (*target*)
4	Ubuntu versi 20.04.06	Sistem operasi (IDS)
5	Snort versi 2.9.7.0	*Software* IDS
6	Suricata versi 7.0.10	*Software* IDS
7	Zeek versi 7.0.7	*Software* IDS

2.2. Snort

Snort adalah salah satu *software* keamanan jaringan berbasis IDS. Snort bersifat *open-source* sehingga dapat dimodifikasi sesuai dengan kebutuhan. Dalam pengoperasiannya, Snort dapat berjalan pada mode *Packet Sniffer*, *Packet Logger* dan *Network Intrusion Detection* [13]. Snort bisa berjalan dalam berbagai mode, termasuk sebagai sistem deteksi intrusi yang menggunakan *signature-based detection*, yang membuatnya cocok untuk mendeteksi pola serangan TCP SYN Flood secara spesifik. Selain itu, Snort sudah banyak digunakan secara luas dan memiliki dokumentasi lengkap sehingga memudahkan proses instalasi, konfigurasi, dan analisis hasil dalam penelitian ini.

2.3. Suricata

Suricata adalah perangkat IDS/IPS *open-source* yang mendeteksi serangan menggunakan metode *signature-based* [14]. Suricata dipilih karena mendukung *multi-threading*, sehingga lebih cepat dalam memproses lalu lintas jaringan dibandingkan IDS lain. Selain itu, Suricata kompatibel dengan aturan Snort, memudahkan pengujian dan perbandingan dalam penelitian ini.

2.4. Zeek

Zeek (sebelumnya Bro) adalah IDS *open-source* yang fokus pada analisis mendalam lalu lintas jaringan secara real-time. Zeek unggul dalam mendeteksi aktivitas mencurigakan yang sulit dideteksi oleh metode konvensional, termasuk pola serangan yang tidak bergantung pada signature [15]. Zeek dipilih karena pendekatan

analisis perilaku dan *scripting* yang fleksibel memungkinkan deteksi serangan SYN Flood dengan analisis konteks paket, melengkapi metode *signature-based* dari Snort dan Suricata dalam penelitian ini.

2.5. Windows Server 2022

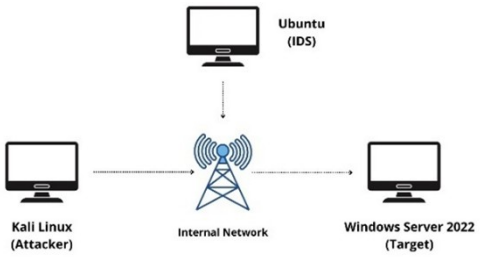
Windows Server 2022 adalah sistem operasi server terbaru yang banyak digunakan dalam lingkungan enterprise. Sistem ini tetap memiliki potensi kerentanan, termasuk terhadap serangan TCP SYN Flood yang menyasar proses TCP *handshake* [16]. Pemilihan Windows Server 2022 sebagai target dalam penelitian ini didasari oleh arsitekturnya yang kompleks serta penerapannya dalam layanan penting seperti *web server* dan *Active Directory*, sehingga cocok untuk menguji efektivitas IDS dalam situasi realistis dan lingkungan jaringan modern.

2.6. Serangan SYN Flood

Serangan SYN flood merupakan serangan yang akan membanjiri server dengan paket SYN sehingga server akan secara terus menerus mengirimkan kembali paket SYN-ACK. Efek dari metode ini adalah server tidak dapat melayani request yang lain dan resource dari server tersebut akan terus menerus meningkat [7]. Karena pola serangannya konsisten dan berulang, SYN Flood termasuk jenis serangan yang mudah dikenali oleh sistem deteksi intrusi berbasis *signature*, sehingga cocok digunakan dalam pengujian IDS.

2.7. Desain/Perancangan

Pada tahap desain/perancangan dilakukan proses perancangan gambaran topologi jaringan. Pada tahap ini juga membahas mengenai mekanisme dari alur perancangan agar dapat beroperasi sesuai kebutuhan. Perancangan sistem IDS ini dilakukan dengan merancang topologi yang akan digunakan sebagai simulasi dan pengujiannya.



Gambar 1. Topologi Jaringan

Gambar 1 menggambarkan topologi yang akan digunakan pada saat proses implementasi. Pada topologi diatas terdapat PC *attacker*, PC target dan PC untuk mendeteksi serangan yang terhubung dalam satu jaringan internet lokal. Semua PC akan berjalan pada *software* VirtualBox dan menggunakan sistem operasi Linux. Lalu akan dilakukan instalasi *software* Snort, Suricata dan Zeek pada PC untuk mendeteksi serangan, PC attacker sendiri akan dilakukan instalasi HPing3 sebagai tool dalam melakukan penyerangan.

Pengujian ini dilakukan secara bergantian pada tiap IDS agar penggunaan sumber daya pada VM dapat maksimal. Dalam melakukan analisis dan pengujian tentunya harus sesuai dengan parameter-parameter yang akan dijadikan acuan untuk melakukan perbandingan seperti 1) Jumlah serangan yang terdeteksi, Pada pengujian ini dilakukan serangan TCP SYN Flood untuk menyerang target, serangan yang dilakukan dalam waktu 30 detik dalam setiap percobaan serangan. Peneliti melakukan 20 kali percobaan yang sama untuk menghasilkan analisa data yang lebih akurat. 2) Penggunaan *resource*, data yang akan dijadikan sampel untuk melakukan perbandingan selanjutnya dalam deteksi masing-masing IDS adalah penggunaan RAM dan CPU.

3. HASIL DAN PEMBAHASAN

3.1. Implementasi

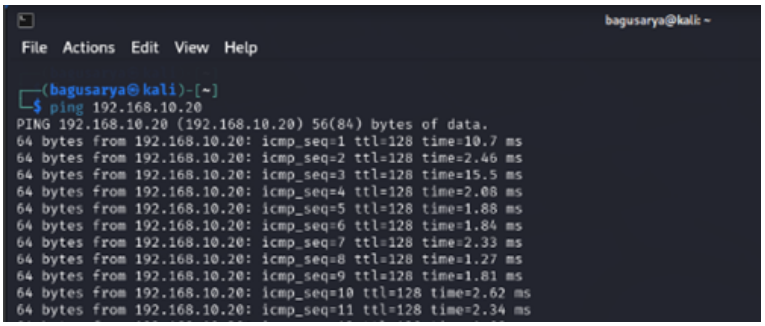
Tahap implementasi merupakan tahap selanjutnya dari proses analisis dan perancangan sistem. Pada tahap ini, dilakukan berbagai aktivitas seperti instalasi, konfigurasi, pengujian sistem, hasil simulasi, analisis hasil pengujian berdasarkan variabel-variabel tertentu yang telah ditentukan sebelumnya. Seluruh kegiatan implementasi dilakukan secara sistematis dan berurutan untuk memastikan hasil pengujian sistem bersifat valid dan akurat.

Langkah awal proses implementasi adalah melakukan instalasi dan konfigurasi terhadap perangkat lunak dan sistem operasi yang akan digunakan dalam proses pengembangan dan pengujian sistem. Proses ini mencakup instalasi pada sistem sebagai pendeteksi maupun pada sistem yang bertindak sebagai penyerang (*attacker*). Adapun tahapan instalasi dan konfigurasi dilakukan dengan urutan sebagai berikut:

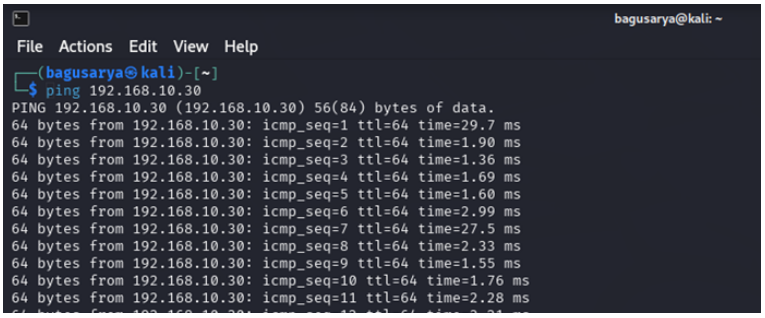
a. Instalasi aplikasi VirtualBox pada OS Windows.
b. Instalasi dan konfigurasi VM dengan OS Kali Linux, Windows Server 2022 dan Ubuntu pada VirtualBox.
c. Instalasi dan konfigurasi Hping3 pada VM *attacker.*
d. Instalasi dan konfigurasi IDS Snort, Suricata dan Zeek pada VM Pendeteksi.
e. Konfigurasi jaringan lokal pada VirtualBox.
f. Konfigurasi IP setiap VM seperti VM Penyerang, VM Target dan VM Pendeteksi.

3.2. Pengujian

Pada proses pengujian akan dilakukan berdasarkan parameter-parameter yang telah dijabarkan diatas yang digunakan sebagai acuan dalam melakukan perbandingan. Proses pengujian IDS Snort, Suricata dan Zeek, sebelumnya akan dilakukan pengaktifan pada VM *attacker*, VM Target dan VM Pendeteksi. Pada tahap ini dilakukan pengujian konektifitas antara VM *attacker*, VM Target dan VM Pendeteksi, untuk memastikan ketiga VM tersebut berhasil terkoneksi dalam sebuah jaringan lokal. Pengujian dilakukan dengan cara membuka *command prompt* dengan menggunakan perintah "`ping`" dari VM *attacker* ke VM Target dan VM pendeteksi (lihat Gambar 2 dan 3).

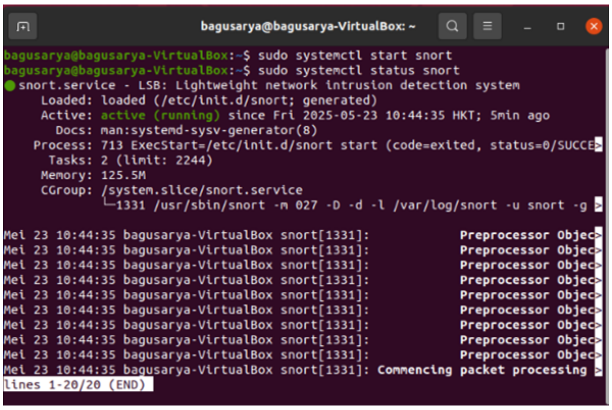


Gambar 2. *Ping* VM *Attacker* ke VM Target



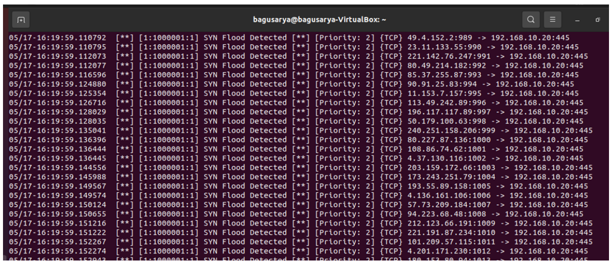
Gambar 3. *Ping* VM *Attacker* ke VM Pendeteksi

Setelah ketiga VM tersebut sudah saling terhubung dengan baik, maka Langkah selanjutnya dilakukan pengujian pertama dengan melakukan pengaktifan *service* IDS Snort untuk memastikan IDS berjalan dengan baik. Untuk mengaktifkan yaitu dengan cara `#sudo systemctl start snort` (Gambar 4).



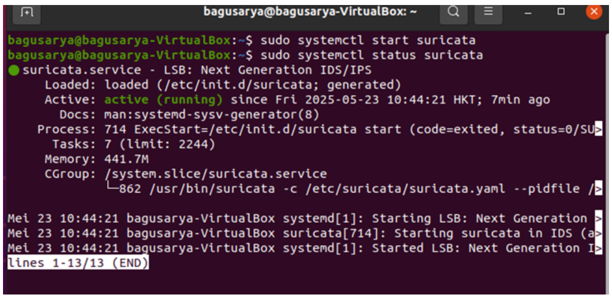
Gambar 4. Pengaktifan IDS Snort

Setelah IDS Snort aktif maka tahap selanjutnya dilakukan konfigurasi untuk pengujian pada VM *attacker* yang sudah terinstal *tool* Hping3 sebagai serangan SYN Flood. Untuk melakukan serangan SYN Flood tersebut dilakukan perintah berikut: `#hping3 -S -p 445 -flood -rand source 192.168.10.20`. Selanjutnya, dilakukan penyerangan pada target. Di sisi VM pendeteksi, Snort akan diaktifkan dengan perintah: `#sudo snort -A console -c /etc/snort/snort.conf -i enp0s3`. Pada saat Snort dijalankan dan VM *attacker* melakukan serangan SYN Flood, Snort berhasil mendeteksi serangan tersebut (Gambar 5).



Gambar 5. Deteksi Serangan oleh Snort

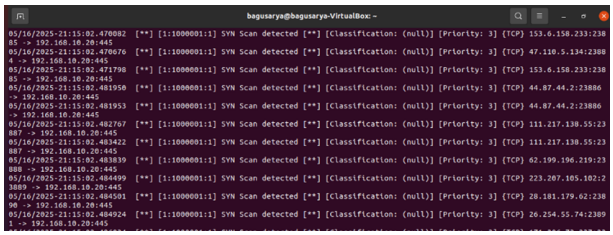
Setelah pengujian pada IDS Snort selesai, maka pengujian kedua dilakukan pada IDS Suricata. Sama seperti pengujian sebelumnya, dilakukan pengaktifan pada *service* IDS Suricata untuk memastikan IDS berjalan dengan baik. Untuk mengaktifkan yaitu dengan cara `#sudo systemctl start suricata` (Gambar 6).



Gambar 6. Pengaktifan IDS Suricata

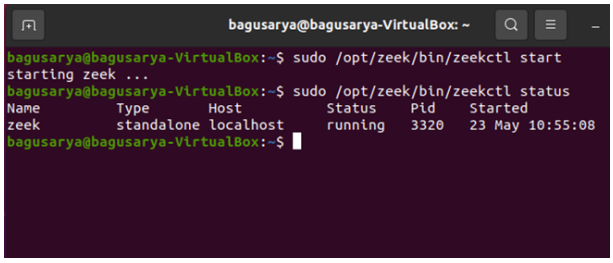
Setelah IDS Suricata aktif maka tahap selanjutnya masih sama seperti pengujian sebelumnya yang dimana dilakukan konfigurasi untuk pengujian pada VM *attacker* yang sudah terinstal *tool* Hping3 seba-

gai serangan SYN Flood. Untuk melakukan serangan SYN Flood tersebut dilakukan perintah berikut: `#hping3 -S -p 445 -flood -rand source 192.168.10.20`, Selanjutnya dilakukan penyerangan pada target. Di sisi VM pendeteksi, Suricata akan diaktifkan dengan perintah: `#sudo tail -f /var/log/suricata/fast .log`. Pada saat Suricata dijalankan dan VM *attacker* melakukan serangan SYN Flood, Suricata berhasil mendeteksi serangan tersebut (Gambar 7).



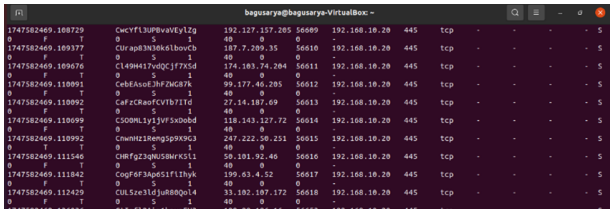
Gambar 7. Deteksi Serangan oleh Suricata

Setelah pengujian pada IDS Suricata selesai, maka pengujian ketiga dilakukan pada IDS Zeek. Sama seperti pengujian sebelumnya, dilakukan pengaktifan pada *service* IDS Zeek untuk memastikan IDS berjalan dengan baik. Untuk mengaktifkan yaitu dengan cara `#sudo opt/zeek/bin/zeekctl start` (Gambar 8).



Gambar 8. Pengaktifan IDS Zeek

Setelah IDS Zeek aktif maka tahap selanjutnya masih sama seperti pengujian sebelumnya yang dimana dilakukan konfigurasi untuk pengujian pada VM attacker yang sudah terinstal tool Hping3 sebagai serangan SYN Flood. Untuk melakukan serangan SYN Flood tersebut dilakukan perintah berikut: `#hping3 -S -p 445 -flood -rand source 192.168.10.20`, Selanjutnya dilakukan penyerangan pada target. Di sisi VM pendeteksi, Zeek akan diaktifkan dengan perintah : `#sudo tail -f /opt/zeek/logs/current/conn .log`. Pada saat Zeek dijalankan dan VM attacker melakukan serangan SYN Flood, Zeek berhasil mendeteksi serangan tersebut (Gambar 9).



Gambar 9. Deteksi Serangan oleh Zeek

3.3. Hasil Pengujian

Setelah melakukan pengujian dengan menggunakan IDS Snort, Suricata dan Zeek, maka hasil yang didapatkan dalam waktu 30 detik setiap percobaan serangan dengan melakukan pengambilan sampel sebanyak 20 kali percobaan yang sama untuk menghasilkan hasil analisis yang lebih akurat. Peneliti mengambil beberapa data primer serangan yang digunakan sebagai dasar analisis ketiga IDS, yaitu:

a. Jumlah serangan terdeteksi IDS selama 30 detik setiap kali percobaan, dengan melakukan 20 kali percobaan serangan yang sama.
b. Data CPU sebelum proses pengujian berlangsung oleh masing-masing IDS.
c. Data CPU pada saat proses pengujian berlangsung oleh masing-masing IDS.
d. Data RAM pada sebelum proses pengujian berlangsung oleh masing-masing IDS.
e. Data RAM pada saat proses pengujian berlangsung oleh masing-masing IDS.

Data dari jumlah serangan dan banyak serangan yang terdeteksi dari masing-masing hasil pengujian tiap IDS setelah dilakukan olah data oleh peneliti dapat dilihat pada Tabel 3.

Tabel 3. Data Pengujian Serangan

Uji	Jumlah Paket Serangan			Jumlah Paket Serangan Terdeteksi		
	Snort	Suricata	Zeek	Snort	Suricata	Zeek
1	71340	21708	68172	44961	19046	40973
2	62347	25201	70289	44118	17087	41377
3	65207	25944	73470	49235	17476	43258
4	73715	31136	70572	55755	18816	38911
5	65891	28367	68636	41350	26187	40197
6	71999	29575	68136	57483	21892	40401
7	65220	29957	74798	45747	20970	37802
8	65010	29866	75604	39492	18482	42477
9	65909	29939	70822	40795	19568	41014
10	66415	30983	70151	41642	25571	38777
11	64804	26827	69746	47415	19682	40146
12	67691	31208	70904	46395	23035	38115
13	71512	76079	70835	51339	27678	40209
14	62966	72904	68603	42902	37263	37750
15	67304	72439	68978	44354	35159	38672
16	65410	70191	70105	46289	35338	36172
17	64953	76526	70250	41060	38910	37972
18	64739	69089	70093	42738	37069	38240
19	66918	78053	70130	40460	51409	37118
20	65054	77695	69386	47156	59668	36550

3.4. Rekapitulasi Hasil Pengujian

Rekapitulasi hasil pengujian terhadap tiga jenis IDS, yaitu Snort, Suricata, dan Zeek, dalam mendeteksi serangan SYN Flood dapat dilihat pada Tabel 4. Tabel tersebut menyajikan data jumlah serangan yang masuk, jumlah serangan yang berhasil terdeteksi, serta persentase keberhasilan deteksi untuk masing-masing IDS. Nilai yang ditampilkan meliputi nilai minimum, maksimum, rata-rata, dan total dari hasil pengujian yang dilakukan secara berulang.

Tabel 4. Data Rekapitulasi Pengujian Serangan

	Jumlah Paket Serangan			Jumlah Paket Serangan Terdeteksi			Persentase		
	Snort	Suricata	Zeek	Snort	Suricata	Zeek	Snort	Suricata	Zeek
MIN	62347	21708	68136	39492	17087	36172	63.34%	78.71%	53.09%
MAX	73715	78053	75604	57483	59668	43258	77.98%	76.45%	57.22%
AVERAGE	66720.2	46684.35	70484	45534.3	28515.3	39306.55	68.25%	61.08%	55.77%
TOTAL	1334404	933687	1409680	910686	570306	786131			

3.5. Analisis Rekapitulasi Hasil Pengujian

Hasil rekapitulasi pengujian pada Tabel 4 menunjukkan bahwa jumlah paket serangan SYN Flood yang dikirim oleh *attacker* pada saat melakukan serangan disetiap IDS berbeda-beda. IDS Zeek mencatat rata-rata 70.484 paket, Snort 66.720 paket, dan Suricata 46.684 paket. Meskipun seluruh IDS dijalankan pada mesin virtual dengan konfigurasi yang sama, perbedaan ini lebih disebabkan oleh perbedaan arsitektur dan mekanisme pemrosesan paket yang digunakan oleh masing-masing IDS. Selain itu, waktu pelaksanaan serangan dan kondisi jaringan saat pengujian juga turut memengaruhi jumlah paket yang berhasil dikirim. Jika serangan dilakukan saat beban sistem tinggi atau saat ada aktivitas lain yang berjalan di latar belakang, maka kemungkinan

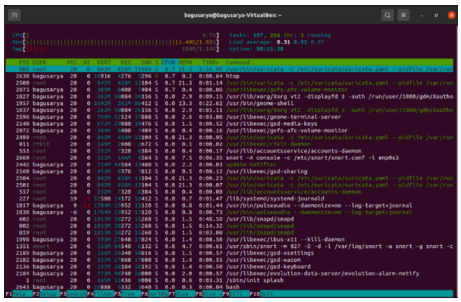
terjadinya *packet loss* akan meningkat. Dengan demikian, jumlah paket yang dikirim tidak hanya mencerminkan intensitas serangan, tetapi juga menunjukkan kemampuan sistem dalam menangani lalu lintas padat di bawah berbagai kondisi teknis.

Kemudian, hasil rekapitulasi pengujian pada Tabel 4 juga menunjukkan bahwa jumlah paket serangan SYN Flood yang berhasil dideteksi pada masing-masing IDS berbeda-beda. Temuan penelitian ini adalah Snort mencatat hasil deteksi tertinggi, dengan rata-rata 45.534 serangan, disusul oleh Zeek sebanyak 39.306, dan Suricata sebanyak 28.515 serangan. Perbedaan ini dipengaruhi oleh pendekatan deteksi yang digunakan. Snort dan Suricata menggunakan metode berbasis *signature*, yang sangat efektif dalam mengenali serangan yang sudah dikenal, seperti SYN Flood. Hal ini menjelaskan mengapa Snort dan Suricata mampu secara langsung mengidentifikasi serangan tersebut meskipun hasil Suricata lebih rendah karena pengaruh konfigurasi atau performa sistem. Sementara itu, Zeek menggunakan pendekatan analisis perilaku, yang lebih fokus pada pencatatan dan mengevaluasi aktivitas jaringan daripada langsung mengklasifikasikan paket sebagai serangan. Hasil penelitian ini didukung oleh penelitian Perdigón Llanes [10] bahwa Zeek lebih menekankan pencatatan log daripada deteksi langsung berbasis *signature*.

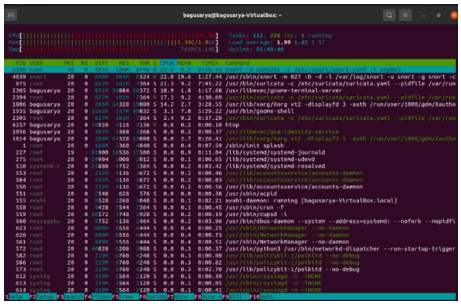
Sementara itu, Snort menunjukkan persentase deteksi tertinggi sebesar 68.25%, disusul oleh Suricata 61.08%, dan Zeek 55.77%. Meskipun Zeek mendeteksi lebih banyak serangan secara absolut (39.306 paket) dibanding Suricata (28.515 paket), persentase deteksi Zeek lebih rendah karena volume lalu lintas serangan yang diterimanya jauh lebih besar. Hal ini menunjukkan bahwa efektivitas deteksi tidak hanya dilihat dari jumlah serangan yang terdeteksi, tetapi juga dari rasio keberhasilannya. Dalam hal ini, Snort terbukti lebih unggul dalam mendeteksi serangan secara konsisten terhadap lalu lintas yang diterima. Hasil penelitian ini didukung oleh penelitian Lukman et al. [7] yang menyimpulkan bahwa Snort unggul dalam hal konsistensi dan efisiensi deteksi terhadap serangan SYN Flood.

3.6. Analisis Pemakaian Sumber Daya (CPU dan RAM)

Aktifitas penggunaan RAM dan CPU sebelum pengujian menggunakan IDS Snort menunjukan nilai yang normal dan saat pengujian serangan berjalan menunjukan peningkatan aktivitas yang dapat dilihat pada Gambar 10 dan Gambar 11.

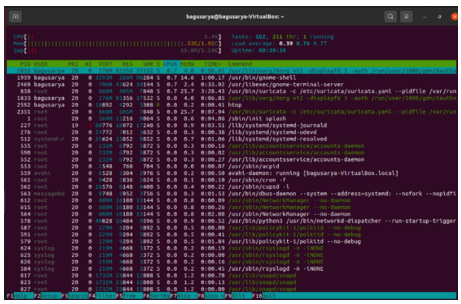


Gambar 10. CPU RAM Snort Sebelum Serangan

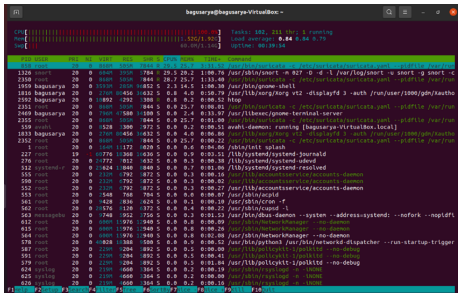


Gambar 11. CPU RAM Snort pada Saat Serangan Berjalan

Setelah itu, untuk data aktifitas penggunaan RAM dan CPU sebelum pengujian menggunakan IDS Suricata menunjukan nilai yang normal dan saat pengujian serangan berjalan menunjukan peningkatan aktivitas yang dapat dilihat pada Gambar 12 dan Gambar 13.

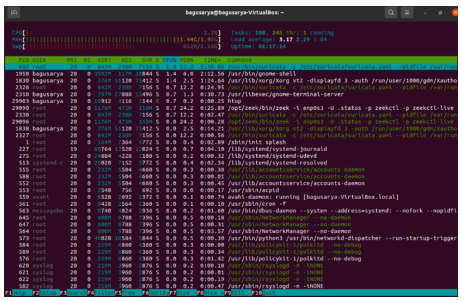


Gambar 12. CPU RAM Suricata Sebelum Serangan

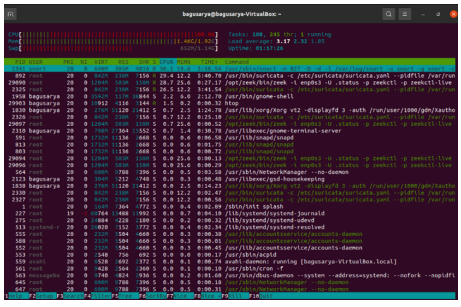


Gambar 13. CPU RAM Suricata pada Saat Serangan Berjalan

Terakhir, untuk data aktifitas penggunaan RAM dan CPU IDS Zeek sebelum pengujian menunjukan nilai yang normal dan saat pengujian serangan berjalan menunjukan peningkatan aktivitas yang dapat dilihat pada Gambar 14 dan Gambar 15.



Gambar 14. CPU RAM Zeek Sebelum Serangan



Gambar 15. CPU RAM Zeek pada Saat Serangan Berjalan

Tabel 5. Data Rekapitulasi Penggunaan CPU

Uji	CPU Usage (%)					
	Start Test			Penggunaan		
	Snort	Suricata	Zeek	Snort	Suricata	Zeek
1	0.1	1	0.7	16	21.9	19.7
2	0.1	0.7	0.3	16.3	18.7	26.9
3	0.1	1	0.3	22	23.6	23.7
4	0	1.3	0.3	15	21.3	22.7
5	0	1.1	0.3	16.3	26.3	22
6	0	3	0.7	16.3	21.7	26.7
7	0	1.7	0.3	13.7	23.3	19.3
8	0	0.7	0.3	17	22.3	24.6
9	0.3	0.7	0.3	15.6	36.2	26.7
10	0	1	0.3	18.3	21.3	19.9
11	0	1.3	0.3	15	22.3	27.3
12	0	2	0.3	13.6	21.3	22
13	0	2	0.3	14.6	27.2	21.3
14	0.7	1.3	0.3	16.9	20.3	16.3
15	1	2	0.3	16.3	23.6	15.6
16	0.7	1.3	0.3	15.7	24.5	22
17	0	1.5	0.3	17.6	22.3	22.3
18	0.3	1.3	0.3	15.3	26.9	19.3
19	0	1.7	0.3	17.9	33.7	18.9
20	0.3	2.3	0.3	17.3	31.9	18.3
MIN	0	0.7	0.3	13.6	18.7	15.6
MAX	1	3	0.7	22	36.2	27.3
AVERAGE	0.18	1.445	0.34	16.335	24.53	21.775
TOTAL	3.6	28.9	6.8	326.7	490.6	435.5

Pada Tabel 5 yaitu data rekapitulasi penggunaan CPU, menunjukkan bahwa seluruh IDS mengalami peningkatan penggunaan CPU selama serangan SYN Flood. Suricata mencatat penggunaan tertinggi dengan rata-rata 24.53% dan maksimum 36.2%, karena menggunakan arsitektur *multi-threading* yang memungkinkan pemrosesan paralel namun lebih membebani CPU. Zeek menyusul dengan rata-rata 21.78% dan maksimum 27.3%. Meski juga mendukung *multi-threading*, beban CPU Zeek dipengaruhi oleh proses analisis perilaku jaringan yang kompleks. Sementara itu, Snort memiliki penggunaan CPU terendah, rata-rata 16.33% dan maksimum 22%, karena menggunakan arsitektur *single-threading* yang lebih ringan namun kurang efisien untuk lalu lintas besar.

Sebagai tambahan, dilakukan perhitungan rasio antara rata-rata penggunaan CPU dan rata-rata jumlah serangan yang diterima. Hasilnya menunjukkan bahwa Snort memiliki rasio terendah sebesar 0.02%, diikuti oleh Zeek 0.03%, dan Suricata 0.05%, yang mengindikasikan bahwa Snort lebih efisien dalam penggunaan CPU terhadap beban serangan yang diterima.

Tabel 6. Data rekapitulasi Penggunaan RAM

Uji	RAM Usage (%)					
	Start Test			Penggunaan		
	Snort	Suricata	Zeek	Snort	Suricata	Zeek
1	8.5	4.4	2	18.6	24.4	15.8
2	8.9	4.3	2.2	19	24.3	16.7
3	8.5	4.3	2.2	19.6	24.3	17.2
4	5.3	4.3	2.2	14.5	24.3	16.4
5	4.5	4.3	2.2	14.7	24.3	16.8
6	5	4.3	2.2	15.2	24.3	16.7
7	6.4	4.3	2.2	16.3	24.3	16.8
8	7.2	4.3	2.2	17.2	24.3	17.3
9	8.4	4.3	2.2	18.6	24.3	16.5
10	9.3	4.3	2.2	19.3	24.3	16.6
11	9.3	4.3	8.7	19.4	24.3	18.7
12	9.4	4.3	2.1	19.4	24.3	16.3

Uji	RAM Usage (%)					
	Start Test			Penggunaan		
	Snort	Suricata	Zeek	Snort	Suricata	Zeek
13	9.4	4.3	2.2	19.5	24.5	16.3
14	9.5	4.5	2.2	19.5	24.6	16.3
15	9.5	4.6	2.2	19.5	24.6	16.6
16	9.5	4.6	2.2	19.5	24.6	16.2
17	9.5	4.5	2.2	19.5	24.5	16.3
18	9.5	4.6	2.2	19.5	24.6	16.3
19	9.5	4.5	2.2	19.5	25.7	16.4
20	9.5	5.7	2.2	19.5	25.7	16.2
MIN	4.5	4.3	2	14.5	24.3	15.8
MAX	9.5	5.7	8.7	19.6	25.7	18.7
AVERAGE	8.33	4.45	2.51	18.39	24.525	16.62
TOTAL	166.6	89	50.2	367.8	490.5	332.4

Pada Tabel 6 yaitu data rekapitulasi penggunaan RAM, menunjukkan bahwa seluruh IDS mengalami peningkatan penggunaan RAM selama serangan. Suricata mencatat penggunaan tertinggi dengan rata-rata 24.52% dan maksimum 25.7%, mencerminkan konsumsi memori yang besar akibat arsitektur *multi-threading* dan kemampuan pemrosesan paralel yang intensif. Snort, meski berbasis *single-threading*, menunjukkan rata-rata penggunaan RAM 18.39% dan maksimum 19,6%, lebih tinggi dari Zeek. Hal ini karena Snort menyimpan banyak data *signature* dalam memori untuk pencocokan pola. Sementara itu, Zeek mencatat penggunaan RAM terendah dengan rata-rata 16.62%, dan maksimum 18.7%. Hal ini menunjukkan bahwa Zeek lebih efisien dalam konsumsi memori dibandingkan Snort dan Suricata. Efisiensi ini berasal dari pendekatan analisis berbasis log yang lebih ringan terhadap memori, meskipun tetap menjalankan proses inspeksi lalu lintas jaringan secara mendalam.

Sebagai pelengkap, dilakukan juga perhitungan rasio antara rata-rata penggunaan RAM dan rata-rata jumlah serangan yang diterima oleh masing-masing IDS. Hasilnya menunjukkan bahwa Zeek memiliki rasio terendah sebesar 0.02%, diikuti oleh Snort 0.03%, dan Suricata 0.05%. Hal ini menguatkan bahwa Zeek paling efisien dalam penggunaan RAM, meskipun deteksi *real-time* nya tidak seoptimal Snort dan Suricata.

4. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa masing-masing *Intrusion Detection System* (IDS) memiliki karakteristik dan keunggulan dalam mendeteksi serangan SYN Flood. Snort menunjukkan performa terbaik dengan tingkat deteksi tertinggi yaitu 68.25%, serta penggunaan CPU paling rendah sebesar 16.33% dan RAM 18.39%. Hal ini menunjukkan bahwa pendekatan berbasis *signature* yang diterapkan Snort sangat efektif dalam mengenali pola serangan yang sudah dikenal serta efisien dalam penggunaan sumber daya. Suricata menempati posisi kedua dengan rata-rata deteksi 61.08%, namun mencatat penggunaan CPU dan RAM tertinggi, masing-masing sebesar 24.53% dan 24.52%. Kelebihan utama Suricata terletak pada arsitektur *multi-threading* yang mendukung pemrosesan paralel, menjadikannya cocok untuk sistem jaringan berskala besar yang membutuhkan performa tinggi, meskipun dengan kebutuhan resource yang lebih besar. Zeek mencatat deteksi rata-rata 55,77%, dengan penggunaan RAM paling rendah (16.62%) dan konsumsi CPU sebesar 21.77%. Meskipun tidak se-efektif Snort dan Suricata dalam deteksi *real-time*, Zeek unggul dalam pencatatan log dan analisis perilaku jaringan, sehingga sangat ideal digunakan untuk kebutuhan forensik jaringan dan monitoring lalu lintas secara menyeluruh.

Berbeda dari penelitian sebelumnya yang umumnya menggunakan sistem target berbasis Linux, penelitian ini menguji performa IDS pada platform Windows Server 2022 untuk mencerminkan kondisi jaringan produksi aktual. Hasil pengujian yang dilakukan pada sistem target Windows Server 2022 mencerminkan kondisi realistis pada lingkungan produksi yang umum digunakan di berbagai sektor. Maka dari itu pada pengujian ini IDS Snort lebih unggul dalam pendeteksian serangan dan efisiensi penggunaan CPU. Suricata unggul dalam efektivitas

penanganan lalu lintas jaringan akibat dukungan arsitektur *multi-threading*, meskipun dengan konsumsi *resource* yang lebih tinggi. Zeek menunjukkan keunggulan pada penggunaan RAM yang paling rendah dan sangat cocok digunakan untuk analisis lalu lintas jaringan secara mendalam (*logging*). Dengan mempertimbangkan tingkat deteksi, efisiensi penggunaan *resource*, serta pendekatan deteksi yang digunakan, Snort menjadi pilihan IDS paling optimal dalam konteks serangan SYN Flood.

Saran untuk penelitian selanjutnya, disarankan untuk mengembangkan pengujian dengan menambahkan variasi jenis serangan lainnya, seperti UDP Flood, ICMP Flood, dan serangan berbasis aplikasi, guna mengukur fleksibilitas dan kemampuan adaptif masing-masing IDS. Selain itu, kombinasi IDS dan *Intrusion Prevention System* (IPS) juga perlu dipertimbangkan untuk menciptakan solusi keamanan jaringan yang lebih menyeluruh dan responsif terhadap berbagai jenis ancaman. Evaluasi terhadap performa dalam jaringan skala besar atau terdistribusi juga bisa menjadi fokus pengembangan di masa mendatang.

DAFTAR PUSTAKA

[1] A. Irawan et al., "Tantangan dan Strategi Manajemen Keamanan Siber di Indonesia berbasis IoT," *Journal Zetroem*, vol. 6, no. 1, pp. 114–119, Apr. 3, 2024. DOI: 10.36526/ztr.v6i1.3376.

[2] O. Rivaldi dan N. L. Marpaung, "Penerapan Sistem Keamanan Jaringan Menggunakan Intrusion Prevention System Berbasis Suricata," *INOVTEK Polbeng - Seri Informatika*, vol. 8, no. 1, p. 141, Jun. 17, 2023. DOI: 10.35314/isi.v8i1.3269.

[3] M. R. H. Tambunan dan S. N. Neyman, "Implementasi Firewall pada Linux untuk Pencegahan Dari Serangan DoS," *Journal of Technology and System Information*, vol. 1, no. 4, p. 10, Jun. 13, 2024. DOI: 10.47134/jtsi.v1i4.2648.

[4] S. Munawarah, K. Kurniabudi, dan E. A. Winanto, "Deteksi Serangan DDoS SYN Flood Pada Jaringan Internet of Things (IoT) Menggunakan Metode Deep Neural Network (DNN)," *Jurnal Informatika dan Rekayasa Komputer (JAKAKOM)*, vol. 4, no. 1, pp. 982–991, Apr. 30, 2024. DOI: 10.33998/jakakom.2024.4.1.1710.

[5] H. Alamsyah, R. Riska, dan A. Al Akbar, "Analisa Keamanan Jaringan Menggunakan Network Intrusion Detection and Prevention System," *JOINTECS (Journal of Information Technology and Computer Science)*, vol. 5, no. 1, p. 17, Jan. 25, 2020. DOI: 10.31328/jointecs.v5i1.1240.

[6] A. Khaliq dan S. Novida Sari, "Pemanfaatan Kerangka Kerja Investigasi Forensik Jaringan untuk Identifikasi Serangan Jaringan Menggunakan Sistem Deteksi Intrusi (IDS)," *Jurnal Nasional Teknologi Komputer*, vol. 2, no. 3, pp. 150–158, Aug. 18, 2022. DOI: 10.61306/jnastek.v2i3.52.

[7] L. Lukman dan M. Suci, "Analisis Perbandingan Kinerja Snort Dan Suricata Sebagai Intrusion Detection System Dalam Mendeteksi Serangan Syn Flood Pada Web Server Apache," *Respati*, vol. 15, no. 2, p. 6, Jul. 10, 2020. DOI: 10.35842/jtir.v15i2.343.

[8] E. H. Kalabo, S. Syaifuddin, dan F. D. S. Sumadi, "Analisa Performa Intrusion Detection System (IDS) Snort Dan Suricata Terhadap Serangan TCP SYN Flood," *Jurnal Repositor*, vol. 4, no. 3, Jan. 16, 2024. DOI: 10.22219/repositor.v4i3.31108.

[9] G. K. Bada, W. K. Nabare, dan D. K. K. Quansah, "Comparative Analysis of the Performance of Network Intrusion Detection Systems: Snort, Suricata and Bro Intrusion Detection Systems in Perspective," *International Journal of Computer Applications*, vol. 176, no. 40, pp. 39–44, Jul. 15, 2020. DOI: 10.5120/ijca2020920513.

[10] R. Perdigón-Llanes, "Evaluación de Snort y Suricata para la detección de sondeos de redes y ataques de denegación de servicio," *Revista científica de sistemas e informática*, vol. 2, no. 2, e363, Jul. 20, 2022. DOI: 10.51252/rcsi.v2i2.363.

[11] T. Purnama, Y. Muhyidin, dan D. Singasatia, "Implementasi Intrusion Detection System (IDS) Snort sebagai Sistem Keamanan Menggunakan Whatsapp dan Telegram sebagai Media Notifikasi," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 14, no. 2, pp. 358–369, Sep. 1, 2023. DOI: 10.51903/jtikp.v14i2.726.

[12] W. Haniyah et al., "Simulasi Serangan Denial of Service (DoS) menggunakan Hping3 melalui Kali Linux," *Journal of Internet and Software Engineering*, vol. 1, no. 2, p. 8, Jun. 11, 2024. DOI: 10.47134/pjise.v1i2.2654.

[13] P. P. Insani, I. Kanedi, dan A. A. Akbar, "Implementation of Snort as a Wireless Network Security Detection Tool Using Linux Ubuntu," *Jurnal Komputer, Informasi dan Teknologi*, vol. 3, no. 2, pp. 443–458, Dec. 31, 2023. DOI: 10.53697/jkomitek.v3i2.1488.

[14] A. R. Zain et al., "Implementasi Intrusion Detection System (IDS) Suricata dan Management Log Elk Stack untuk Pendeteksian Kegiatan Mining," *Jurnal Poli-Teknologi*, vol. 22, no. 1, pp. 23–29, Jan. 31, 2023. DOI: 10.32722/pt.v22i1.4974.

[15] S. Haas, R. Sommer, dan M. Fischer, "Zeek-Osquery: Host-Network Correlation for Advanced Monitoring and Intrusion Detection," en, in *ICT Systems Security and Privacy Protection*, M. Hölbl, K. Rannenberg, dan T. Welzer, Eds., vol. 580, Cham: Springer International Publishing, 2020, pp. 248–262. DOI: 10.1007/978-3-030-58201-2_17.

[16] I. P. A. E. Pratama, "TCP SYN Flood (DoS) Attack Prevention Using SPI Method on CSF: A PoC," *Bulletin of Computer Science and Electrical Engineering*, vol. 1, no. 2, pp. 63–72, Aug. 7, 2020. DOI: 10.25008/bcsee.v1i2.7.