

## Analisa Performa Intrusion Detection System (IDS) Snort d/dan Suricata Terhadap Serangan TCP SYN Flood

Edi H. Kalabo<sup>1</sup>, Syaifuddin<sup>2</sup>, Fauzi Dwi Setiawan Sumadi<sup>3</sup>

<sup>1,2,3</sup>Universitas Muhammadiyah Malang

h.\_437270@webmail.umm.ac.id<sup>1</sup>, syaifuddin@umm.ac.id<sup>2</sup>, fauzisumadi@umm.ac.id<sup>3</sup>

### Abstrak

Serangan TCP SYN Flood adalah serangan untuk mencegah pengguna yang sah menggunakan sumber daya tertentu seperti layanan web, jaringan, atau host. Peretas dengan sengaja memblokir ketersediaan sumber daya untuk pengguna resminya. Dalam beberapa tahun terakhir ini Snort dan Suricata telah menjadi IDS berbasis open source yang mengalami kemajuan pesat. Namun, telah terjadi masalah dalam kalangan user untuk memilih kedua IDS tersebut dalam hal mendeteksi serangan TCP SYN Flood dengan packet generator scapy. Pada penelitian terdahulu sudah membandingkan antara IDS tersebut dalam mendeteksi serangan TCP SYN Flood dengan packet generator hping3. Terdapat beberapa metode untuk digunakan didalam pengujian ini. Metode-metode tersebut adalah meluncurkan packet-packet dengan jumlah yang berbeda menggunakan tcpreplay. Digunakannya metode-metode ini dengan tujuan untuk mengetahui performa dari DS Snort dan IDS Suricata dalam menangani serangan dari TCP SYN Flood. Parameter-parameter yang akan diuji di penelitian tersebut ialah akurasi deteksi dan kecepatan deteksi. Pada pembahasan tersebut telah didapatkan bahwa IDS Snort unggul untuk aspek-aspek seperti akurasi deteksi dan kecepatan deteksi. Sedangkan IDS Suricata tidak unggul dalam aspek-aspek tersebut.

**Kata Kunci:** TCP SYN Flood, IDS, Snort, Suricata, Scapy

### Abstract

A TCP SYN Flood attack is an attack to prevent authorized users from using certain resources such as web services, networks, or hosts. Hackers intentionally block the availability of resources for their authorized users. In recent years Snort and Suricata have become open source based IDSs that are experiencing rapid progress. However, there has been a problem among users to choose the two IDSs in terms of detecting TCP SYN Flood attacks with the scapy packet generator. Previous research has compared the two IDSs in terms of detecting TCP SYN Flood attacks with packet generator hping3. There are several methods applied in the test. Those methods are launching different number of packets using tcpreplay. This method is used to determine the performance of IDS Snort and IDS Suricata in handling TCP SYN Flood. The parameters that will be tested in this research are detection accuracy and detection speed. From the discussion, it was found that IDS Snort excels in aspects such as detection accuracy and detection speed. Meanwhile, IDS Suricata does not excel in these aspects.

**Keywords:** TCP SYN Flood, IDS, Snort, Suricata, Scapy

### 1. Pendahuluan

TCP (Transmission Control Protocol) ialah protokol yang sangat umum digunakan dalam dunia internet, dikarenakan kelebihan dari protokol TCP ialah terjadinya koreksi ketika adanya kesalahan. Saat memilih protokol TCP, maka proses pengiriman paket akan terjamin keamanannya. Hal tersebut dikarenakan oleh adanya bagian untuk sebuah metode yang disebut flow control. Flow control menentukan kapan paket data harus dikirim kembali dan menentukan kapan menghentikan aliran data dari paket sebelumnya, sampai paket data tersebut sudah berhasil dikirim. Hal tersebut dikarenakan jika paket data berhasil dikirim, maka akan terjadi suatu tabrakan. Ketika hal ini terjadi, maka klien akan meminta kembali paket dari server sampai seluruh paket berhasil dikirim dan sama dengan aslinya [1].

Pada tahun 2019 sebuah perusahaan jaringan melakukan penelitian dengan mengamati beberapa serangan jaringan besar-besaran termasuk yang terbesar yang pernah tercatat, yaitu mencapai 580 Mbps, ukuran serangan dengan tidak lebih dari 1.000 permintaan per detik (RPS). Serangan SYN Flood akan terjadi bila suatu host hanya mengirimkan paket SYN saja secara

terus-menerus tanpa memperdulikan mengirimkan paket ACK sebagai balasan tersebut dapat menimbulkan host tujuan akan terus menunggu paket tersebut dan menyimpannya kedalam back log [3]. Intrusion Detection System (IDS) adalah sebuah cara yang dapat digunakan dalam mendeteksi aksi yang mencurigakan dalam jaringan. Ada dua jenis IDS berlisensi open-source yang sedang marak digunakan saat ini, yaitu Snort dan Suricata.

Snort dan Suricata merupakan dua jenis IDS yang berlisensi open-source dan cukup ramai dipakai oleh pengguna dalam memantau jaringan komputer. Tetapi, kedua jenis IDS tersebut pasti memiliki kelemahan dan keunggulan masing-masing, terutama dalam hal mendeteksi serangan dari TCP SYN Flood. Maka dari itu, dibutuhkan sebuah pengujian dalam hal untuk mengetahui hasil akhir performa dari kedua IDS dalam mendeteksi serangan TCP SYN Flood [4]

Pada kajian pustaka yang dipelajari selama penelitian berlangsung, kebanyakan dari penelitian sebelumnya hanya memusatkan penelitian pada satu jenis serangan dan satu jenis IDS saja. Alasan tersebutlah yang penulis jadikan dasar untuk memilih penelitian tersebut, dengan mengacu pada penelitian yang dilakukan oleh Emir dan Lukman, penulis ingin mengetahui performa kedua IDS tersebut jika dilakukan dengan menggunakan jenis serangan yang berbeda juga menggunakan simulasi paket yang juga berbeda. Pada penelitian ini penulis akan membuat simulasi paket menggunakan scrapy dikarenakan keunggulan Scapy dalam memanipulasi paket lebih powerful dan dapat menerjemahkan beberapa kode dalam satu variable.

## 2. Kajian Teori

Ada beberapa penelitian sebelumnya yang berkaitan dengan penelitian ini telah dibahas pada bab pendahuluan dalam latar belakang. Pada kajian pustaka yang telah penulis pelajari, umumnya penelitian terdahulu hanya terpusat pada satu subjek penelitian. Jika ditemukan ada perbandingan diantara IDS Snort dan IDS Suricata, maka hal tersebut hanya sebatas membahas performa kedua IDS dalam mendeteksi *malicious activity* saja, dan tidak terfokus pada satu jenis serangan saja dengan skala paket yang mungkin jauh lebih besar. [5]

Snort dan Suricata dipilih pada studi ini karena penulis merasa Snort dan Suricata memiliki fungsi yang sebanding, pada *set* aturan deteksi dan sintaks. Kedua IDS berada dibawah lisensi *Open-source* tetapi kedua IDS ini pastinya memiliki kelebihan dan kelemahan masing-masing, terkhusus dalam hal menangani serangan dari TCP SYN Flood. Penelitian yang pernah dilakukan oleh Raza Shah dari jurnalnya yang berjudul *Performance Comparison of Intrusion Detection Systems and Application of Machine Learning to Snort System*, disimpulkan bahwa IDS Snort dan Suricata dapat menentukan dan memeriksa 10 Gbps lalu lintas jaringan serta untuk mengklasifikasikan lalu lintas yang sah dan berbahaya dengan benar. Sedangkan menurut Risyad, dalam jurnalnya dengan judul *Perbandingan Performa Intrusion Detection System (IDS) Snort Dan Suricata Dalam Mendeteksi Serangan TCP SYN Flood*, disimpulkan bahwa IDS Snort dan Suricata mampu mendeteksi serangan TCP SYN Flood dengan memanipulasi paket menggunakan *Hping3*.

Pada kajian pustaka yang telah dipelajari kebanyakan penelitian sebelumnya hanya terfokus pada satu jenis serangan saja. Alasan tersebutlah dan kemudian dijadikan dasar untuk memilih penelitian ini, karena mengacu pada penelitian yang dilakukan oleh Emir dan Lukman, penulis juga ingin mengetahui performa kedua IDS tersebut kalau diuji dengan menggunakan jenis serangan yang berbeda menggunakan simulasi paket yang berbeda juga. Pada penelitian ini penulis akan membuat simulasi paket menggunakan scrapy. Dalam hal itulah yang melatarbelakangi penelitian ini dilakukan dengan mengacu pada peneliti terdahulu, diharapkan dapat dilihat hasil akhir performa untuk kedua IDS dalam mendeteksi serangan TCP SYN Flood dengan menggunakan packet generator scrapy.

### 2.1 TCP SYN Flood

Serangan SYN Flooding adalah sebuah serangan yang menyerang langsung ke server menggunakan cara mengirim paket data menuju server, kemudian saat server memberikan respon dari paket data yang dikirim dan kemudian memberi tanda sebagai jawaban ke client melalui alamat IP dari pengirim, dan ternyata respon saat diterima oleh server untuk membalas jawaban client dikirim oleh client menggunakan alamat IP yang palsu, kejadian itu terjadi secara kontinyu, menyebabkan server menjadi tak terkendali untuk mengirimkan respon atas paket dari alamat IP palsu.

Dalam cara kerja dari *three-why-handshake* dalam protokol TCP, *client* yang melakukan usaha agar bisa terhubung dengan server berarti mengirimkan sebuah pesan *request* kepada server agar menjadi tanda bahwa client dan server telah terhubung, pesan yang dimaksud ialah SYN. Dan server otomatis menerima kemudian mengonfirmasi *request* dari client dan akan mengirimkan ulang SYN-ACK pada *client*. *Client* yang sudah mendapatkan SYN-ACK dari server kemudian meresponnya dengan pesan ACK, maka saat itu bisa disebut bahwa koneksi dari *client* ke *server* telah terjadi [7]

## 2.2 IDS

*Intrusion Detection System (IDS)* adalah sebuah aturan atau cara agar digunakan dengan tujuan mendeteksi aktivitas yang mencurigakan dalam atau jaringan dan menjadi salah satu komponen penting dalam jaringan IDS dirancang untuk mendeteksi setiap gangguan atau lalu lintas yang tidak bersahabat di jaringan menurut Achmad Hambali dalam skripsinya yang berjudul implementasi *intrusion detection system (IDS)* pada keamanan PC Server terhadap serangan Flooding data, dikatakan bahwa IDS dapat melakukan pemeriksaan dan pengawasan terhadap lalu lintas jaringan *inbound* ataupun *outbound* dalam sebuah jaringan, menjalankan Analisa paket kemudian mencari bukti atas percobaan intrusi (penyusupan) yang dilakukan dalam jaringan tersebut[5].

### 2.2.1 Snort

Snort merupakan IDS yang telah dikembangkan pada tahun 1998 oleh *sourcefire* dan telah menjadi standart *de-facto* untuk IDS selama decade terakhir dan telah banyak dikerahkan dan diselidiki dalam studi penelitian. Snort sendiri merupakan arsitektur *singlethread* seperti yang ditunjukkan pada Gambar 1 yang menggunakan tumpukan TCP/IP untuk menangkap dan memeriksa muatan paket jaringan.

Menurut Adeb Alhomoud dalam jurnalnya yang berjudul *performance Evaluation Study of Intrusion Detection System*, dikatakan bahwa tujuan utama Snort adalah untuk secara efektif menganalisa semua paket yang lewat dalam jaringan.

Berikut fitur-fitur yang ada pada Snort IDS:

- Mampu berjalan pada semua jenis Sistem Operasi
- *Single-thread* (Beralur Tunggal)
- Kinerjanya dalam hal memeriksa protokol
- Kinerjanya untuk memeriksa suatu kondisi ataupun *event*
- Kinerjanya dalam hal *me-reassembly* paket-paket
- Mempersiapkan paket *output* dalam bentuk ASCII
- Menampilkan model GUI untuk hasil analisa

### 2.2.2 Suricata

Suricata IDS dikembangkan pada tahun 2010 oleh Open Information Security Foundation (OISF). Suricata dipublikasikan sebagai IDS generasi berikutnya yang mengintegrasikan ide-ide baru seperti *multithreading*, seperti yang ditunjukkan pada gambar 2.5. mengoperasikan Suricata tidaklah jauh berbeda dengan pengoperasian IDS Snort, sebab bisa dijalankan menggunakan *Command lines* dan *berkeley packet filter*. Aturan pada Suricata pun tidak jauh berbeda dengan aturan pada Snort.

Rilis awal Suricata bekerja pada platform linux 2.6 dan mampu mendukung konfigurasi pemantauan lalu lintas inline dan pasif serta mampu menangani beberapa tingkat lalu lintas gigabit jaringan.

Berikut fitur-fitur yang terdapat pada Suricata IDS:

- Dapat berjalan pada semua jenis system operasi
- *Multi-thread*
- Tersedia fitur *Intrusion Prevention Sistem (IPS)*
- Kemampuan untuk memeriksa jenis protokol
- Tersedia fitur *Network Security Monitoring (NSM)*
- Kemampuan untuk memeriksa kondisi/*event*

### 2.3 Scapy

Scapy dikembangkan oleh Philippe Biondi dan *The Scapy Community*. Scapy sendiri ialah sebuah *tools* untuk manipulasi paket yang kuat dan interaktif. Scapy sanggup memalsukan (*building* paket) atau men-*decode* paket-paket dari berbagai protokol-protokol, mengirimkan ke jaringan, menangkapnya ataupun mencocokkan permintaan dan balasan [7].

### 2.4 Tcpreplay

Tcpreplay merupakan *tools* yang *Open-source* atau gratis, dikembangkan oleh Aaron Turner untuk system operasi UNIX yang memberikan kemampuan dalam menggunakan trafik yang telah ditangkap dalam format .pcap untuk menguji berbagai perangkat jaringan, dan memungkinkan anda untuk mengklasifikasikan trafik sebagai clien atau server, menulis header layer 2, 3, atau 4 dan akhirnya mengirimkan ulang kembali trafik ke jaringan.

## 3. Metodologi

Dalam bab ini akan diisi oleh tahapan dalam menyelesaikan penelitian yang berjudul Analisa Performa *Intrusion Detection System* (IDS) Snort dan Suricata terhadap serangan TCP SYN Flood. Pada bab ini berfungsi sebagai panduan alur pengerjaan dalam penelitian yang ditunjukkan agar penelitian berjalan sesuai dengan harapan. Tahapan tersebut berupa langkah-langkah yang akan dilakukan dalam penelitian secara sistematis dan spesifik. Penelitian ini dilalui dalam empat tahap yaitu, Analisa, Desain Perancangan Sistem, Implementasi, dan Pengujian & hasil seperti pada Gambar 1 dibawah ini.



Gambar 1. Flowchart Metode Penelitian

Adapun penjelasan dari flowchart sebagai berikut:

#### a. Analisa

Teori yang terikat disusun beralaskan referensi dari beberapa sumber, referensi yang digunakan dalam penelitian ini berdasarkan dari jurnal, artikel, jurnal konferensi, buku, maupun penelitian sebelumnya baik secara nasional maupun internasional serta dibimbing secara langsung oleh pembimbing tugas akhir pada proses penelitian tersebut.

#### b. Desain perancangan system

Pada arsitektur sistem akan dijelaskan bagaimana proses dari pengumpulan data dari kedua IDS *Snort* dan *Suricata*, yang menyimpan datanya. Pada tahap ini, akan dilakukan dengan merancang desain sistem yang akan dibangun, berupa skema sistem yang menjelaskan rancangan alur kerja system, komponen-komponen, topologi dan tesbed ujicoba.

#### c. Implementasi

Pada sub bab ini akan membahas tentang fungsi-fungsi dari setiap *rules* yang akan diimplementasikan ke dalam kedua IDS dan menjelaskan komponen-komponen dari system.

#### d. Hasil dan Analisa

Pada tahap ini terdapat 5 aktivitas pengujian yang akan dilakukan pada mesin virtual IDS Snort dan IDS Suricata. Aktivitas pengujian yang dimaksud adalah mesin virtual Scapy akan mengirim packet dengan jumlah packet yang berbeda-beda sesuai dengan paket yang tertera pada setiap aktivitas.

## e. Penutup

Memberikan kesimpulan yang memberikan jawaban dari pertanyaan yang diajukan sebelumnya, dikelompokkan jawaban hanya terfokus pada ruang lingkup pertanyaan dan keseluruhan jawaban akan disesuaikan dengan jumlah dari rumusan masalah yang diajukan. Pada bab ini juga berisi saran untuk penelitian kedepannya.

#### 4. Hasil dan Pengujian

##### 4.1 Paramter Pengujian

Pengujian dilakukan beralaskan pada parameter-parameter yang nantinya akan menjadi bahan hitungan dari kedua IDS berdasarkan hasil percobaan. Berikut ini ialah parameter-parameter yang nantinya penguj gunakan untuk penelitian ini, diantaranya adalah Akurasi deteksi dan kecepatan deteksi dan penggunaan sumber daya system.

##### 4.2 Perhitungan Akurasi dan Rata-rata

Metode analisis ini dibuat dalam hal untuk menghitung persentase akurasi dari hasil pengujian berdasarkan akurasi deteksi. Kemudian data-data yang telah didapatkan akan dibuat dalam bentuk persentase angka dengan skala yang menentukan akurasi kedua IDS dalam hal mendeteksi serangan TCP SYN Flood. Untuk mengolah data-data yang telah dihasilkan dari kedua IDS, dapat disimak dalam Persamaan dibawah ini.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$$

- TP = True Positive
- TN = True Negative
- FP = False Positive
- FN = False Negative

Metode perhitungan rata-rata berfungsi sebagai perhitungan secara matematis dengan memperlihatkan jumlah data. Metode tersebut digunakan untuk menghitung nilai rata-rata dari hasil pengujian berdasarkan kecepatan deteksi. Dengan kata lain, semakin dekat nilai standart rata-rata dengan angka 0, maka semakin baik IDS yang digunakan. Untuk menghitung dan mengolah nilai rata-rata, dapat dilihat pada Persamaan dibawah ini.

$$Rata - rata = \sum \frac{X}{n}$$

X = nilai dari masing-masing sample

n = jumlah sample yang diambil dan digunakan sebagai hitungan

##### 4.3 Implementasi system

Setelah mempunyai paket TCP SYN yang sudah tersimpan dalam format .pcap, untuk membuat pengiriman paket sesuai dengan pengujian, yaitu 1000 loop, 2000 loop, 3000 loop, 4000 loop, dan 5000 loop, maka dibutuhkan bantuan *tools* tcpreplay.

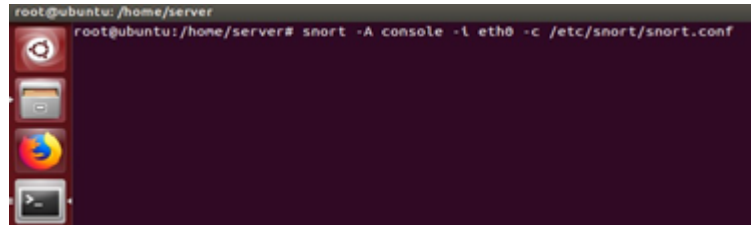


```
root@ubuntu: /home/server/Downloads
root@ubuntu: /home/server/Downloads# tcpreplay --pps=1000 --lntf1=eth0 test1000packets.pcap
Actual: 1000 packets (59370 bytes) sent in 0.999049 seconds
Rated: 59426.5 Bps, 0.475 Mbps, 1000.95 pps
Flows: 4 flows, 4.00 fps, 994 flow packets, 6 non-flow
Statistics for network device: eth0
Successful packets: 1000
Failed packets: 0
Truncated packets: 0
Retried packets (ENOBUFS): 0
Retried packets (EAGAIN): 0
root@ubuntu: /home/server/Downloads#
```

Gambar 2. Running tcpreplay

Pada pengujian system, IDS Snort dan Suricata akan dijalankan bersamaan dengan tcpreplay. Untuk menjalankan Snort dengan menambahkan beberapa switch -v, -d, -e, -c akan menghasilkan beberapa keluaran yang berbeda. Pada Gambar 3 penulis akan menjalankan snort dalam mode IDS yaitu *Sniffer*, mode untuk melihat header TCP/IP yang lewat.

Pada Gambar 4 bagaimana sistem Suricata akan dijalankan melalui terminal ubuntu dan akan menangkap setiap paket yang terhubung dengan interface yang dipasang suricata, dengan perintah -c console dan perintah -i untuk interface yang akan dimonitoring.




```

root@ubuntu: /home/server
root@ubuntu: /home/server# snort -A console -i eth0 -c /etc/snort/snort.conf

```

Gambar 3. Running Snort



```

root@ubuntu: ~
File Edit View Search Terminal Help
root@ubuntu:~# sudo suricata -c /etc/suricata/suricata.yaml -i ens33 &
[2] 2355
root@ubuntu:~# 24/6/2021 -- 14:18:33 - <Notice> - This is Suricata version 6.0.2
2 RELEASE running in SYSTEM mode
24/6/2021 -- 14:18:34 - <Notice> - all 2 packet processing threads, 4 management
t threads initialized, engine started.

```

Gambar 4. Running Suricata

#### 4.4 Pengujian Akurasi Deteksi

Tujuan dalam pengujian ini ialah untuk mengetahui hasil akurasi deteksi dari kedua IDS tersebut dalam hal mendeteksi serangan menggunakan konfigurasi aturan yang sama. Perihal hasil dari pengujian ini adalah untuk mengetahui apakah kedua IDS dapat memiliki hasil akhir yang sama atau ada perbedaan.

Tabel 1. Persentase Akurasi

Aktivitas	Snort	Suricata
1	99%	98%
2	99%	98%
3	99%	99%
4	99%	99%
5	99%	99%



Gambar 5. Perbandingan Akurasi deteksi IDS Snort dan Suricata

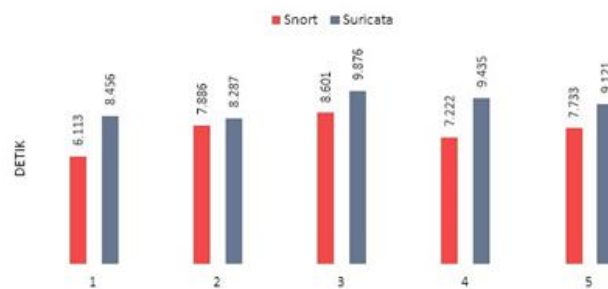
#### 4.5 Kecepatan Deteksi

Pada pengujian kecepatan deteksi IDS snort akhirnya dapat mendeteksi adanya aktivitas jauh lebih cepat daripada IDS Suricata. Dalam Tabel 1 dan Tabel 2 dapat dilihat bahwa IDS Snort dan IDS Suricata kali ini sama-sama dapat mendeteksi semua aktivitas. Untuk tabulasi waktu awal pengujian, pengujian yang dilakukan oleh IDS Snort dilakukan pengujian untuk waktu sekitar jam 15 sore. Kemudian pengujian yang dilakukan oleh IDS Suricata dilakukan pada besoknya sekitaran jam 11 siang.

*Tabel 2. Hasil Rata-Rata Kecepatan Deteksi*

	Snort	Suricata
Rata-Rata	7.509 detik	9.034 detik

**PERBANDINGAN KECEPATAN DETEKSI  
PADA SNORT DAN SURICATA**



*Gambar 6. Perbandingan kecepatan deteksi Snort dan Suricata*

#### 4.6 Penggunaan Sumber Daya

Selanjutnya untuk mengetahui penggunaan sumber daya dari kedua IDS dalam menghadapi serangan TCP SYN dari attacker. Untuk penggunaan sumber daya IDS Snort dan IDS Suricata dapat dilihat dari hasil analisis disetiap tabel berdasarkan kalkulasi yang telah dibuat. Dimana IDS Snort mencatat penggunaan RAM dengan kisaran rata-rata 12% dan kenaikan sebesar 4% sampai 5% kecuali pada pengujian pertama dimana IDS Snort mencatat penggunaan ram lebih kecil dibandingkan aktivitas 2, 3, 4, dan 5. Sedangkan pada IDS Suricata, mencatat penggunaan RAM dengan kisaran 1% dan rata-rata kenaikan sebesar 1.4% per aktivitas.

*Tabel 3. Penggunaan sumber daya (RAM) pada Snort*

Banyak Packet	Persentase
1000	1.16%
2000	8.33%
3000	11.10%
4000	15.97%
5000	23.6%

*Tabel 4. Penggunaan Sumber daya (RAM) pada Suricata*

Banyak Packet	Persentase
1000	0.90%
2000	1.11%
3000	1.31%
4000	1.34%
5000	2.38%



Gambar 7. Perbandingan Penggunaan Sumber Daya RAM IDS Snort dan Suricata

## 5. Penutup

### 5.1 Kesimpulan

Berdasarkan hasil penelitian yang sudah dilakukan, kemudian dapat disimpulkan ada beberapa hal mengenai IDS Snort dan IDS Suricata, Performa IDS Snort dan IDS Suricata mendapatkan hasil yang sangat baik dalam mendeteksi TCP SYN Flood. Terbukti saat berhasilnya kedua IDS untuk mendeteksi dan menangani adanya serangan TCP SYN Flood, kedua IDS juga memberikan data yang sama dengan data tabel jumlah pengujian paket. IDS Snort memiliki keakuratan deteksi lebih baik dalam mengukur akurasi deteksi. Hal tersebut terbukti pada pengujian akurasi deteksi dimana IDS Snort menangkap paket yang tidak sesuai *rule* lebih kecil dan mendapatkan packet TCPSYN sebesar 1000, 2000, 3000, 4000, dan 5000 pada pengujian, paket tersebut sesuai dengan jumlah paket yang diuji. Disisi lain IDS Suricata menangkap paket yang tidak sesuai *rule* lebih banyak dari IDS Snort tetapi berhasil menangkap paket yang sesuai *rule* jumlahnya sama dengan IDS Snort yaitu sebesar 1000, 2000, 3000, 4000, dan 5000. Kemudian pada pengujian kecepatan deteksi IDS Snort telah mendeteksi semua jenis aktivitas lebih cepat daripada IDS Suricata. Hal berikut bisa dilihat dengan didaparkannya nilai rata-rata 7.509 detik, sedangkan untuk IDS Suricata mendapatkan nilai rata-rata sebesar 9.034 detik. Angka tersebut menunjukkan bahwa IDS Snort dapat mendeteksi lebih cepat daripada IDS Suricata. Pada penggunaan sumber daya yang dimaksud adalah penggunaan RAM dimana IDS Suricata mengambil lebih sedikit sumber daya dibandingkan dengan IDS Snort. Dilain sisi IDS Snort memakai banyak sekali sumber daya sebab IDS Snort lebih *powerfull* ketika berjalan pada system *single-thread*. Dimana IDS Snort mencatat penggunaan RAM dalam kisaran rata-rata 12% dan kenaikan sebesar 4% sampai 5% kecuali pada pengujian pertama dimana IDS Snort mencatat penggunaan RAM lebih kecil dibandingkan aktivitas 2, 3, 4, dan 5. Sedangkan pada IDS Suricata, mencatat penggunaan RAM dengan kisaran 1% dan rata-rata kenaikan sebesar 1.4% per aktivitas.

### 5.2 Saran

Saran yang dapat penulis berikan tertuju pada penelitian selanjutnya adalah lebih terfokus pada *output* dengan tampilan *Graphic User Interface* pada kedua IDS, dan mencoba dengan menggunakan *Packet generator* lain selain Scapy. Sebagai contoh dapat menggunakan Nping, TRex atau dengan mengimplementasikan sendiri menggunakan bahasa pemrograman tertentu.

### Referensi

- [1] L. Xiaoming, "Denial of Service (DoS) attack with UDP Flood."
- [2] M. Muqorobin, Z. Hisyam, M. Mashuri, H. Hanafi, and Y. Setiyantara, "Implementasi Network Intrusion Detection System (NIDS) Dalam Sistem Keamanan Open Cloud Computing," *Maj. Ilm. Bahari Jogja*, vol. 17, no. 2, pp. 1–9, 2019, doi:10.33489/mibj.v17i2.205.
- [3] A. H. Hambali and S. Nurmiati, "Implementasi Intrusion Detection System (IDS) Pada Keamanan PC Server Terhadap Serangan Flooding Data," *Sainstech J. Penelit. dan Pengkaj. Sains dan Teknol.*, vol. 28, no. 1, pp. 35–43, 2018, doi: 10.37277/stch.v28i1.267.



- [4] Igal Zeifman, "Global DDoS Threat Landscape Q1 2017," *Imperva Incapsula*, p. 1, 2017, [Online]. Available: <https://www.incapsula.com/ddos-report/ddos-report-q1-2016.html%0Ahttps://www.incapsula.com/ddos-report/ddos-report-q1-2017.html>.
- [5] S. Khadafi, B. D. Meilani, and S. Arifin, "Sistem Keamanan Open Cloud Computing Menggunakan Ids (Intrusion Detection System) Dan Ips (Intrusion Prevention System)," *J. IPTEK*, vol. 21, no. 2, p. 67, 2017, doi: 10.31284/j.iptek.2017.v21i2. 207.
- [6] N. Dietrich, "Snort 2.9.9.x on Ubuntu 14 and 16," p. 3, 2015.
- [7] E. Risyad, M. Data, and E. S. Pramukantoro, "Perbandingan Performa Intrusion Detection System (IDS) Snort Dan Suricata Dalam Mendeteksi Serangan TCP SYN Flood," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 9, pp. 2615–2624, 2018.
- [8] Lukman and M. Suci, "Analisis Perbandingan Kinerja Snort Dan Suricata Sebagai Intrusion Detection System Dalam Mendeteksi Serangan Syn Flood Pada Web Server Apache," *J. Teknol. Inf.*, vol. XV, no. 2, pp. 6–15, 2020.
- [9] P. Biondi, "Scapy Documentation," vol. 469, no. 4, pp. 155–203, 2017, [Online]. Available: <http://dx.doi.org/10.1016/j.physrep.2008.09.003>.
- [10] S. A. Raza Shah and B. Issac, "Performance Comparison of Intrusion Detection Systems and Application of Machine Learning to Snort System," *arXiv*, 2017.
- [11] F. Informatika, U. Telkom, and W. Fathoni, *Deteksi Penyusupan Pada Jaringan Komputer Menggunakan Ids Snort Intrusion Detection in Computer Networks Using Ids Snort*. 2015.
- [12] A. Alhomoud, R. Munir, J. P. Disso, I. Awan, and A. Al-Dhelaan, "Performance evaluation study of Intrusion Detection Systems," *Procedia Comput. Sci.*, vol. 5, pp. 173–180, 2011, doi: 10.1016/j.procs.2011.07.024.
- [13] S. Sinha, *Beginning Ethical Hacking with Kali Linux*. 2018.
- [14] P. S. (IAIN S. A. S. Ningsih, "Bab ii kajian teori," *Bab li Kaji. Teor.*, no. 1, pp. 23–35, 2011.
- [15] AppNeta, "Welcome to Tcpreplay," 2013. [online]. Available: <http://tcpreplay.synfin.net/>. [Accessed: 07-Jun-2021]

