ANALISIS PERBANDINGAN KEAKURATAN DETEKSI SERANGAN DAN EFISIENSI PEMAKAIAN CPU RESOURCES DARI TOOLS PENDETEKSI SERANGAN SNORT DAN SURICATA YANG DI PASANG DI WEB SERVER

Dhanar Restu Arrasy, Agustinus Noertjahyana
Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto No.121-131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) – 8417658
Email:dhanararrasy@gmail.com, agust@petra.ac.id

ABSTRAK

Analisis perbandingan keakuratan deteksi serangan dan efisiensi pemakaian CPU Resources dari tools pendeteksi serangan *Snort* dan *Suricata* yang di pasang di Web Server.

Snort dan *Suricata* merupakan IDS tools yang digunakan untuk mendeteksi serangan jaringan. *Snort Snort* adalah Open Source Intrusion Prevention System (IPS) menggunakan serangkaian aturan yang membantu menentukan aktivitas jaringan berbahaya dan menggunakan aturan tersebut untuk menemukan paket yang cocok dengannya dan menghasilkan peringatan bagi pengguna. *Suricata* adalah mesin pendeteksi ancaman sumber terbuka independen terkemuka. Dengan menggabungkan deteksi intrusi (IDS), pencegahan intrusi (IPS), pemantauan keamanan jaringan (NSM) dan pemrosesan PCAP, *Suricata* dapat dengan cepat mengidentifikasi, menghentikan, dan menilai serangan paling canggih.

Dari hasil penelitian ini bahwa pada saat *flood DOS Attack* Snort memiliki rata – rata 93,5 % sedangkan Suricata memiliki 94,2 % di bagian CPU pada *port* 80. Ini menjelaskan bahwa Suricata unggul efisiensi di banding dengan Snort pada saat *flood* DOS Attack. Sedangkan untuk bagian port 443 Snort memiliki rata – rata 94,5 % dan Suricata memiliki 95,67 % di bagian CPU pada *port* 443 membuktikkan bahwa Suricata unggul dalam aktifitas CPU. Untuk bagian memori Snort memiliki rata – rata 19.9 % untuk pemakaian memori atau efisien memori dibanding dengan Suricata yang emmiliki pemakaian memori rata – rata 69,3 % pada port 80. Pada port 443 Snort memiliki pemakaian memori rata – rata 30,3 % dibanding dengan Suricata yang memiliki rata – rata memori dengan pemakaian 30,5 % dalam berarti Suricata memiliki pemakaian lebih sedikit dibanding dengan Snort

Kata Kunci : Cybersecurity, Snort , Suricata, IDS

ABSTRACT

Analyze attack detection accuracy and CPU Resource usage efficiency of Snort and Suricata attack detection tools installed on the Web Server..

Snort and Suricata are IDS tools used to detect network attacks. Snort Snort is an Open Source Intrusion Prevention System (IPS) using a set of rules that help determine malicious network activity and uses those rules to find packets that match it and generate alerts for users. Suricata is a leading independent open source threat detection engine. By combining intrusion detection (IDS), intrusion prevention (IPS), network security monitoring (NSM) and PCAP processing, Suricata can quickly identify, stop, and assess the most advanced attacks.

From the results of this study that at the time of flood DOS Attack Snort had an average of 93.5% while Suricata had 94.2% in the CPU section on port 80. This explains that Suricata is superior in efficiency compared to Snort during flood DOS Attack. As for the port 443, Snort has an average of 94.5% and Suricata has 95.67% in the CPU section on port 443, proving that Suricata is superior in CPU activity. For the memory section, Snort has an average of 19.9% for memory usage or is efficient compared to Suricata which has an average memory usage of 69.3% on port 80. On port 443 Snort has an average memory usage of 30.3% compared to with Suricata which has an average memory usage of 30.5%, it means that Suricata has less usage than Snort

Keyword: Cybersecurity, Snort , Suricata, IDS

1. PENDAHULUAN

Teknologi *Server* semakin kesini berkembang dengan pesat seiring dengan berjalannya waktu. Dengan adanya perkembangan tersebut banyak sekali kemanfaatan yang didapatkan sebagai pengguna teknologi. Salah satu keuntungannya adalah keamanan *server* menjadi semakin kuat susah ditembus oleh serangan penyusup, *hacking*, *virus*, *malware* dan serangan berbahaya lainnya [2].

Intrusion Detection Sytem penting untuk keamanan komprehensif, tetapi ada beberapa hal untuk menggunakan IDS secara efektif. Saat memantau dan menganalisis lalu lintas jaringan untuk aktivitas yang mencurigakan atau berpotensi berbahaya dan sangat penting untuk memiliki personel TI dengan pengetahuan dan keterampilan untuk membuat keputusan dan mengambil tindakan yang diperlukan berdasarkan peringatan IDS jaringan.

Snort adalah *open source* Intrusion Prevention System (IP*S*) terkemuka di dunia. Snort IDS menggunakan serangkaian aturan yang membantu menentukan aktivitas jaringan berbahaya dan menggunakan aturan tersebut untuk menemukan paket yang cocok dengannya dan menghasilkan *alert* bagi pengguna bila terjadi serangan jaringan. Snort didasarkan pada *libpcap* (untuk pengambilan paket perpustakaan), sebuah alat yang banyak digunakan di *sniffer* dan penganalisa lalu lintas TCP / IP. Melalui

analisis protokol serta pencarian dan pencocokan konten, Snort mendeteksi metode serangan, termasuk penolakan layanan, *buffer overflow*, dan pemindaian *port* tersembunyi. Ketika perilaku mencurigakan terdeteksi, Snort mengirimkan peringatan *real-time* ke *syslog*, *file alert* terpisah, atau ke jendela *pop-up*. Sementara Snort telah menjadi salah satu solusi paling populer untuk menggagalkan serangan jaringan, peningkatan kompleksitas serangan dan lanskap penyebaran yang berubah membutuhkan solusi baru. "Ketika kami mulai memikirkan seperti apa rupa generasi IP*S* berikutnya, kami memutuskan untuk memulai dari awal," tulis perusahaan dalam pengumuman rilisnya [5].

Suricata adalah mesin pendeteksi ancaman jaringan sumber terbuka yang menyediakan kemampuan termasuk *Intrusion Detection Sytem* (IDS) dan pemantauan keamanan jaringan. Suricata adalah perangkat lunak IDS, IPS, dan monitoring untuk jaringan yang berkinerja tinggi. Suricata adalah perangkat lunak *open source* yang dikelola yayasan non-profit, Open Information Security Foundation (OISF). Suricata dikembangkan oleh OISF dan vendor pendukungnya [3]. Suricata bekerja sangat baik dengan inspeksi paket yang mendalam dan pencocokan pola yang membuatnya sangat berguna untuk deteksi ancaman dan serangan. Suricata termasuk *open source* yang baik bagi pengguna. Dirancang agar kompatibel dengan komponen keamanan jaringan yang ada, Suricata menampilkan fungsionalitas output terpadu dan opsi *library* yang dapat dicolokkan untuk menerima panggilan dari aplikasi lain.Rilis awal Suricata berjalan pada platform Linux 2.6 yang mendukung konfigurasi pemantauan lalu lintas inline dan pasif yang mampu menangani beberapa tingkat lalu lintas gigabit. Linux 2.4 didukung dengan fungsionalitas konfigurasi yang lebih rendah, seperti tidak ada opsi sebaris.

Dari pemilihan IDS banyak orang awam yang menggunakan IDS dari segi deteksi saja. Namun tidak tahu apa yang terjadi pada *server* ketika terkena serangan terjadi. Dan ini kemungkinan bisa dimanfaatkan oleh pihak yang tidak berwenang untuk melakukan sebuah kerusakan *server* dan pencuriaan *data*. Sehingga dapat dimanfaatkan oleh pihak tidak berwenang untuk melakukan analisis kelemahan dan bentuk *server* dari target penetrasi. Oleh karena itu dibutuhkan sebuah penelitian untuk membandingkan kinerja dari kedua IDS tersebut.

Tujuan penelitian ini untuk membandingkan kedua IDS menggunakan sistem operasi Linux dengan pengujian serangan menggunakan DOS (Denial Of Service) yang akan menyerang pengujian *device server*, kemudian IDS Snort dan Suricata yang telah terpasang pada pengujian *device server* memberikan peringatan jika terjadi serangan. Dalam menentukan hasil perbandingan, digunakan parameter - parameter yang akan menjadi acuan yaitu jumlah serangan yang terdeteksi dan efektivitas deteksi serangan dari kedua IDS tersebut. Sehingga bisa mengetahui kinerja dari kedua *tools* tersebut.

2. DASAR TEORI

2.1 Snort

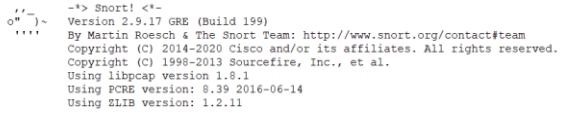
Snort merupakan aplikasi atau perangkat lunak berbasis *Open Source* yang memiliki keunggulan untuk mengetahui adanya indikasi penyusupan pada jaringan berbasis *TCP*/IP secara real time . Jika terindikasi adanya penyusupan, *Snort* akan melakukan pencatatan atau logging terhadap paket-paket yang telah terdeteksi sebagai intrusi berdasarkan aturan yang telah ditetapkan (Muttaqin, 2020).

Pada penelitian Lukman dan Melati (2020), bahwa Snort lebih unggul dalam pemakaian CPU dan efisiensi dan Suricata lebih baik di efektifitas serangan dari uncaptured paket. Dan menyerang DOS dengan waktu, tidak berdasarkan dengan paket yang diluncurkan misalkan 100 paket yang diluncurkan.

Secara prinsip, Snort memerankan tiga fungsi utama :

1. Penangkal program – program sniffer paket – paket seperti tcpdump.
2. Packet logger berguna untuk melakukan debug pada lalu lintas jaringan
3. Sebagai system pencegah intrusi untuk sistem – sistem jaringan

Pada implementasinya, *Snort* memiliki aturan yang telah dikonfigurasi untuk mendeteksi intrusi dalam sebuah jaringan. Terdapat sebuah *database* yang mencakup aturan-aturan yang memiliki fungsi tertentu sehingga dapat digunakan sesuai dengan kebutuhan. Setiap paket yang lalu lintas jaringan akan dianalisa dengan cara melakukan pencocokkan terhadap aturan yang telah ditetapkan. *Rules Snort* menyediakan yang mendeteksi serangan dan aktivitas berbahaya. *Snort* dapat menulis aturan khusus seperti peringatan, *log*, putus koneksi, dll. *Rules* memiliki sintaks sederhana. Juga, dapat menulis semua aturan dalam *file* konfigurasi, dan dapat mengedit apa yang dibutuhkan dari sistem lain [1] . Untuk tampilan Snort versi dapat dilihat di Gambar 1



Gambar 1 Suricata Version

2.2 Suricata

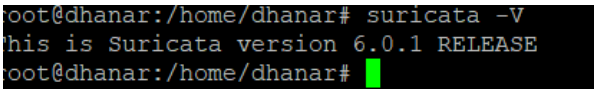
Suricata adalah IDS, IPS, dan monitoring engine untuk jaringan yang berkinerja tinggi. *Suricata* adalah *Open Source* dan dimiliki oleh masyarakat yang dikelola yayasan non-profit, Open Information Security Foundation (OISF). *Suricata* dikembangkan oleh OISF dan vendor pendukungnya (*Suricata* -ids,2020-2). Tiga (3) alasan utama mengapa kita perlu mencoba *Suricata* :

1. Highly Scalable - *Suricata* adalah multi threaded. Ini berarti dapat menjalankan satu instance dan akan menyeimbangkan beban pengolahan di setiap prosesor pada sensor *Suricata* yang dikonfigurasi untuk menggunakan. Hal ini memungkinkan perangkat keras komoditas untuk mencapai kecepatan 10 gigabyte pada lalu lintas real tanpa mengorbankan cakupan *Rule*set.

2. Identifikasi Protocol - Protokol yang paling umum secara otomatis dikenali oleh *Suricata* saat stream komunikasi dimulai, sehingga memungkinkan penulis *Rules* untuk menulis aturan untuk protokol, tidak ke *port* yang diharapkan. Hal ini membuat *Suricata* Malware Command dan Control Channel tidak seperti yang lain. Saluran off HTTP CnC, yang biasanya meluncur tepat oleh sebagian besar sistem IDS, di *Suricata* ini merupakan hal yang mudah. Selain itu, berkat kata kunci khusus dapat mencocokan field protokol yang berkisar dari http URI sampai SSL certificate identifier.

3. Identifikasi *File*, MD5 Checksum, dan *File* Extraction - *Suricata* dapat mengidentifikasi ribuan jenis *file* yang melintasi jaringan. Tidak hanya dapat mengidentifikasi, tetapi jika memutuskan ingin melihat lebih jauh dapat menandai untuk diekstraksi dan *file* akan ditulis ke disk dengan *file data* meta yang menggambarkan situasi penangkapan dan aliran [4].

Pada penelitian Satria (2019), Suricata merupakan IDS yang dapat mendeteksi aktifitas ancaman serangan pada jaringan yang dibantu dengan *rules* yang telah ada. Cara kerja dari Suricata adalah ketika adanya penyerangan Suricata akan melakukan pengecekan paket/serangan yang ada melalui *rules* yang dibuat. Ketika serangan terdeteksi maka Suricata akan membuat *log* serangan yang dilakukan, Suricata juga dapat melakukan deteksi otomatis pada *layer 7* yaitu aplikasi seperti *DNS*, HTTP, IMAP, FTP, dan SMTP. Sehingga Suricata dapat memberikan solusi untuk meningkatkan keamanan dalam Cloud Computing. Untuk melihat Suricata versi dapat dilihat di Gambar 2



Gambar 2 Suricata Version

Rule yang digunakan untuk pengujian ini sama yang digunakan oleh *Snort*. Hanya saja pada penelitian sebelumnya tidak dicantumkan rule yang digunakan. Analisis Perbandingan Kinerja *Snort* Dan *Suricata* Sebagai Intrusion Detection System Dalam Mendeteksi Serangan Syn Flood Pada Web Server Apache (Lukman dan Melati, 2020). Dalam penelitian ini analisis kinerjanya menggunakan IDS Web Server yang di testing menggunakan HPing3 dengan menjabarkan CPU dan RAM Resource yang dipakai . Sehingga penelitian ini menggunakan CLI. Hasil penelitian *Snort* lebih unggul dalam pemakaian CPU dan efisiensi dan *Suricata* lebih baik di efektifitas serangan dari uncaptured paket. Dan menyerang DOS dengan waktu, tidak berdasarkan dengan paket yang diluncurkan misalkan 100 paket yang diluncurkan.

2.3 Rules dan DOS Commands

Rule yang digunakan untuk pengujian ialah sama untuk Suricata dan Snort. Rule yang diguanakan

1. alert tcp any any -> $HOME_NET (80 atau 443) (msg:"DOS"; sid:100;rev:1;classtype:attempted-dos;) (Rule tipe 1)
   - Rule ini digunakan untuk mendeteksi semua alert dos yang menyerang port 80 atau 443. Rule ini akan mengeluarkan alert apabila ip home dan port 80 diserang.
2. alert tcp any any -> $HOME_NET (80 atau 443) (msg:" DOS"; sid:100;rev:1;classtype:attempted-dos;flags:S;flow:stateless;) (Rule tipe 2)
   - Rule ini digunakan untuk mendeteksi semua alert dos yang menyerang port 80 atau 443 terfokus pada SYN (mendeteksi sesuai souce dos attack)
3. alert tcp any any -> $HOME_NET (80 atau 443) (msg:"SYN FLOOD Detected!";flow:stateless;flags:S;threshold:type both, track by_dst,count 2, seconds 4;sid:1000004;rev:1;) (Rule tipe 3)
   - Rule sama halnya dengan aturan flag:s hanya saja ada filter untuk setiap alert terjadi. Alert akan keluar jika ketentuan ada 2 paket terpenuhi dalam 4 detik terdeteksi. Dan di hitung berdasarkan waktu periode. Misal jika dalam 4 detik memiliki 4 alert yang sama, maka 1 alert akan dimunculkan.
4. alert tcp any any -> $HOME_NET (80 atau 443) (msg:"SYN FLOOD Detected!";flow:stateless;flags:S;threshold:type limit, track by_dst,count 2, seconds 4;sid:1000004;rev:1;) (Rule tipe 4)
   - Rule ini sama seperti *threshold* hanya saja beda dari limit yaitu waktu periode dan angka paket. Rule trigger sama seperti *threshold* sebelumnya, hanya saja jika ada 4 alert yang sama dalam 4 detik maka menghasilkan 2 alert.

Dan untuk DOS tool nya yaitu Hping3 dari Kali Linux dengan 2 commands. Yaitu :

1. Hping3 -S -c 100 -p 80 [target ip]
   a. S : Syn paket
   b. c : berapa paket yang dikirimkan
   c. p : *port* berapa yang diserang
2. Hping3 -S -p [port] –flood [target ip]
   a. S : Syn paket
   b. --flood : mengirim paket tanpa batas
   c. p : *port* berapa yang diserang

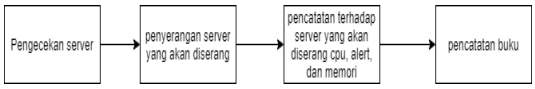
3. IMPLEMENTASI SISTEM

Pada bab ini dijelaskan tentang implementasi sistem sesuai dengan teori dan desain sistem pada bab-bab sebelumnya. Implementasi sistem meliputi 3 implementasi aplikasi, yaitu implementasi os linux, implementasi os kali linux, implementasi aplikasi *Snort* dan implementasi aplikasi *Suricata*

3.1 Implementasi Penelitian

Dalam implementasi penelitian ada beberapa tahap yang dilakukan pada saat pengujian :

1. Tahap pertama melakukan pengecekan server yang akan diserang dan server untuk menyerang
2. Tahap kedua melakukan penyerangan server yang akan diserang
3. Tahap ketiga melakukan pencatatan terhadap server yang akan diserang seperti halnya memori, cpu, dan alert
4. Tahap terakhir pencatatan pada buku untuk penelitian yang telah terjadi.



Gambar 3 Tahap penelitian

Gambar 3 merupakan alur tahap penelitian yang dilakukan.

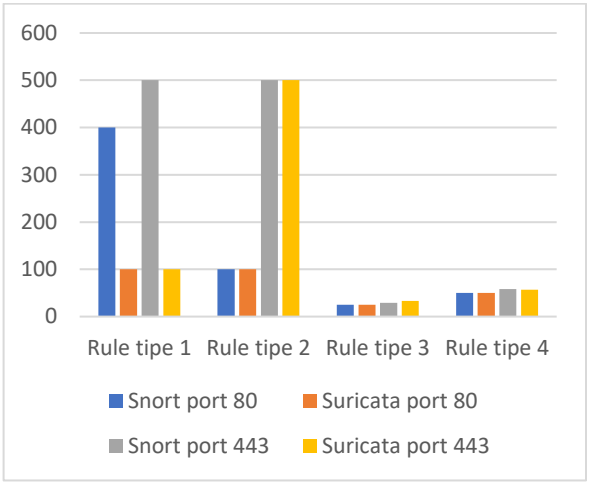
4. HASIL PENELITIAN

4.1 Pengetesan serangan DOS attack

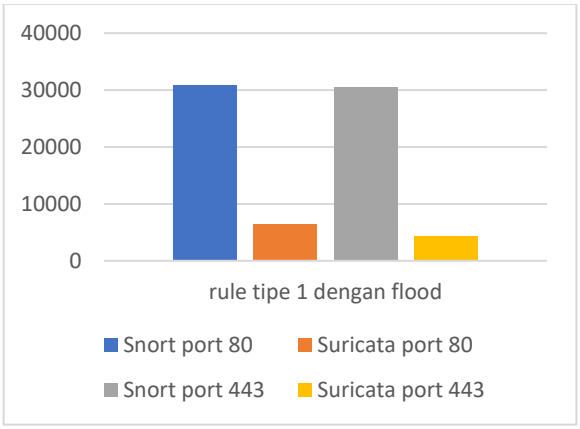
Pengetesan rule dilakukan dengan menggunakan rule yang sama. Menggunakan rule yang sama dengan tujuan sebanding tanpa penambahan atau pengurangan.untuk melakukan tahap ini dibutuhkan serangan DOS.

Pengujian dilakukan yaitu 4 rule yang berbeda akan mendetetksi serangan selama 100 detik dengan tidak flood. Tetapi flood akan tetap dilakukan dengan menggunakan rule tipe 1. Jadi dilakukan 5 pengujian yaitu 4 rule dengan dos 1 serangan per detik dan 1 rule

dengan tipe 1 untuk FLOOD DOS. Hasil untuk port 80 dapat dilihat dari gambar statistik Gambar 4 dan untuk port 443 dapat dilihat dari gambar statistik Gambar 5.



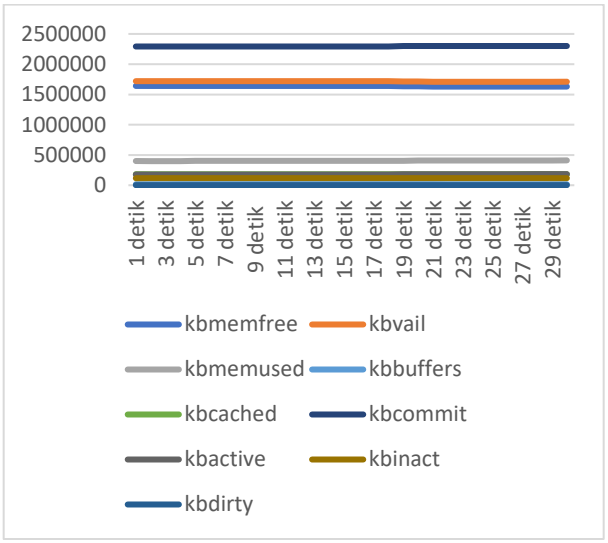
Gambar 4 Total alert Rule Snort Dan suricata pada saat DOS menyerang server



Gambar 5 total alert Rule Snort dan Suricata pada saat FLOOD DOS Attack

4.2 Statistik CPU dan Memori pada saat Flood DOS Attack

Kumpulan semua memori telah di catat pada saat pengujian terjadi tetapi pada saat terjadi flood, memori dan cpu

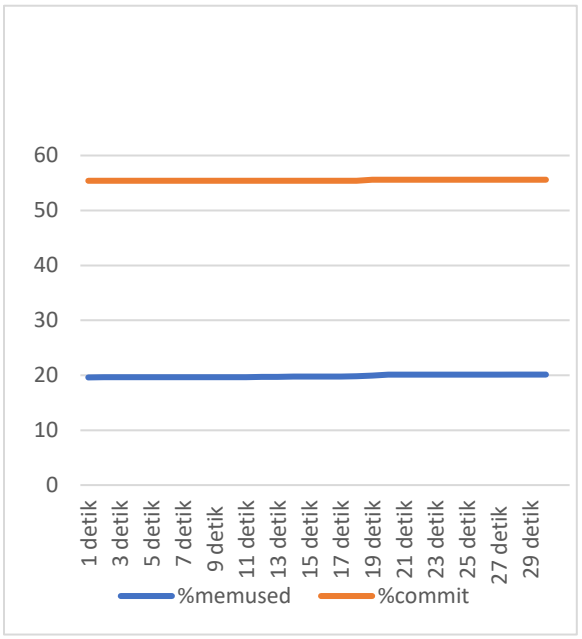


Gambar 6 memori Snort pada saat Flood DOS Attack menyerang port 80

pada Gambar 6 diatas merupakan *Snort* memory server pada saat FLOOD DOS Attack.

- Kbcommit pada poin 1 detik dengan memori 229257kb stabil hingga kenaikan mencapai 2300516 kb pada poin 19 detik
- Kbavail pada poin 1 detik dengan memori 1719668 kb stabil hinga mengalami penurunan 1710416 kb pada poin 30 detik
- Kbmemused mengalami stabil dengan memori 400096 kb hingga poin 30 detik
- Kbactive mengalami kenaikan drastis dengan memori 115760 kb di poin 1 detik naik menjadi 176380 kb di poin 2 detik dan mengalami kenaikan 180780 pada poin 19 detik hingga poin terakhir
- Kbmemsfree mengalami naik turun memori dari 1640828 kb di poin 1 detik
- Kbbuffers mengalami kenaikan sedikit dari 15844 kb di poin 1 detik menuju poin 4 detik dengan memori 15852 kb menuju poin 10 detik 15868 kb menuju 22 detik dengan memori 15878 kb
- Kbcached mengalami kenaikan dari 182500 kb di poin 1 hingga menjadi 183676 di poin 30 detik
- Kbinact mengalami kenaikan sedikit dari 115760 kb di poin 1 detik menjadi 116912 kb di poin 30 detik

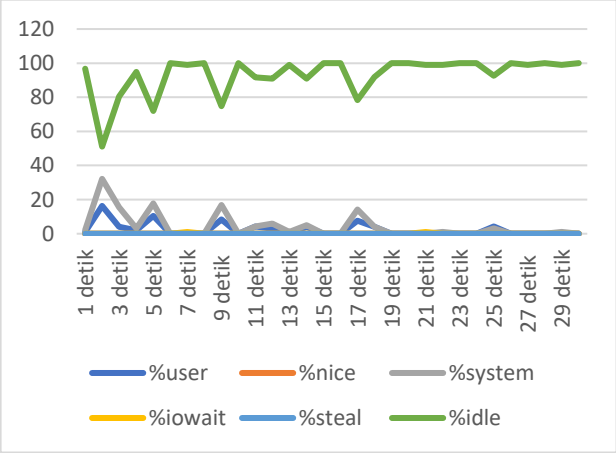
- Kbdirty mengalami kenaikan dari 832 kb di poin 1 detik menjadi 1732 kb di poin 30 detik



Gambar 7 memori persentase Snort pada saat FLOOD DOS Attack menyerang port 80

Penjelasan Gambar 7 :

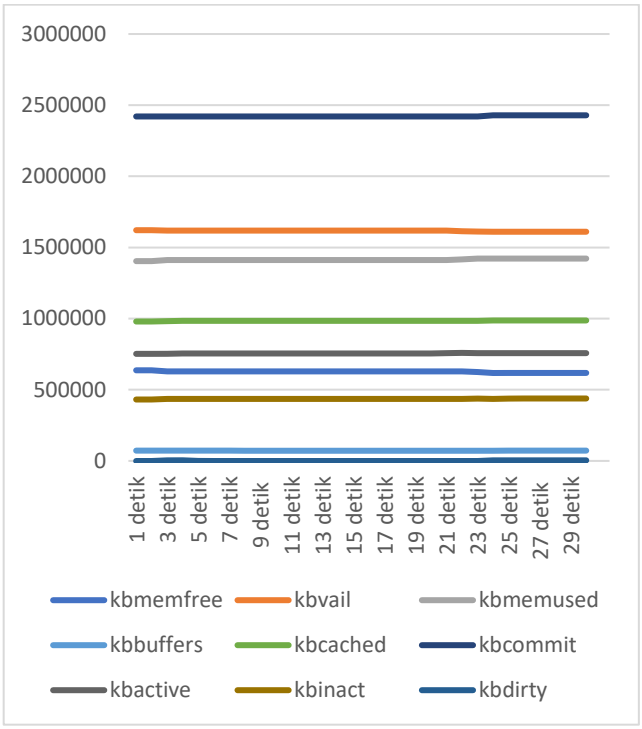
- Penggunaan persentasi memori yang mengalami kenaikan kestabilan mulai dari 55,4% hingga naik sedikit menjadi 55.59%
- Dan untuk commit ialah rata – rata 20.1%



Gambar 8 persentase CPU Snort pada saat FLOOD DOS Attack menyerang port 80

Bagian cpu ada pergerakan gambar pada saat flood terjadi yaitu dibagian idle. Idle akan efek berefek Ketika salah satu system cpu (%user, %nice, %system,%iowait, dan%steal) terjadi perubahan. Penjelasan Gambar 8 :

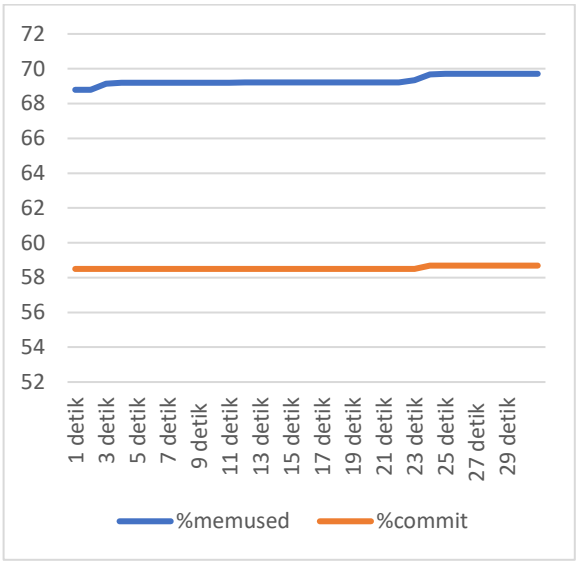
- idle berkurang menjadi menjadi 51.09% karena dari %user (16.3%) dan %system (32.16%) di poin 2 detik
- idle kembali berkurang menjadi 71.88% karena dari %user (10.42%) dan %system (17.71%) di poin 5 detik
- idle kembali berkurang menjadi 74.74% karena dari %user (8.42%) dan %system (16.84%) di poin 9 detik
- idle kembali berkurang menjadi 78.26% karena dari %user (8.42%) dan %system (16.84%) di poin 17 detik



Gambar 9 memori Suricata pada saat FLOOD DOS Attack menyerang port 80

Penjelasan Gambar 9 memori *Suricata* :

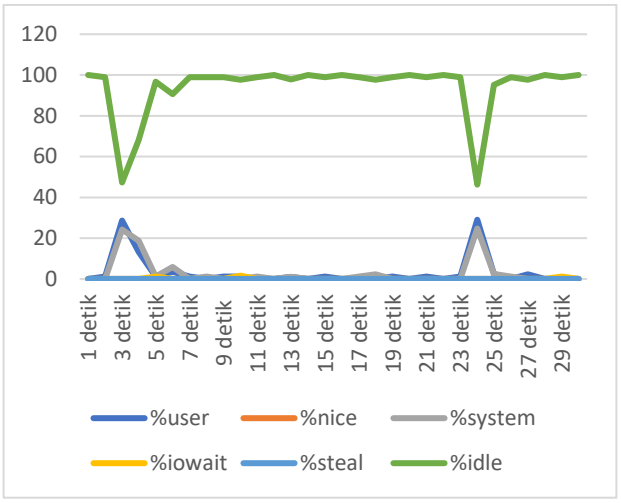
- Kbmemsfree mengalami penurunan dari memori 636880 di poin 1 detik di poin 12 detik dengan memori 628172 kbkb dan mengalami penurunan di poin 25 detik dengan memori 618248 kb
- Kbcommit mengalami kenaikan dari 2420688 kb di poin 1 naik pada 2428632 kb di poin 24 detik
- Kbavail mengalami penurunan dari 1621344 kb di poin 1 detik menjadi 1610460 kb di poin 30 detik
- Kbmemused mengalami kenaikan dari 1404044 kb di poin 1 detik menjadi 1422056 kb di poin 23 detik
- Kbbuffers mengalami naik sedikit dari 72332 kb di poin 1 detik
- Kbcached mengalami kenaikan sedikit dari 979852 kb di poin 1 detik menjadi 987100 kb di poin 24 detik
- Kbactive mengalami kenaikan dari 752424 kb di poin 1 detik menjadi 756676 kb di poin 24 detik
- Kbinact mengalami kenaikan dari 431596 kb di poin 1 detik menjadi 438852 kb di poin 27 detik
- Kbdirty mengalami kenaikan dari 51 kb di poin 1 detik menjadi 3000 di 26 detik



Gambar 10 persentase memori Suricata pada saat FLOOD DOS Attack menyerang port 80

Penjelasan Gambar 10 persentasi memori :

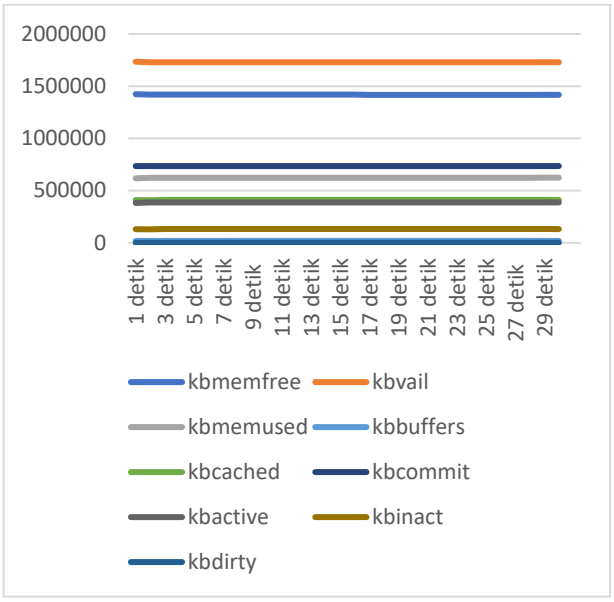
- %commit mengalami kenaikan sedikit dari 58,5 % di poin 1 detik menjadi 58,69% di poin 24 detik
- %memused mengalami kenaikan dari 68.79% di poin 1 detik menjadi 69.71% di poin 24 detik



Gambar 11 persentase CPU Suricata pada saat FLOOD DOS Attack menyerang port 80

Penjelasan Gambar 11 :

- idle berkurang menjadi menjadi 47.25% karena dari %user (28.57%) dan %system (24.18%) di poin 3 detik
- idle kembali berkurang menjadi 46.24% karena dari %user (29.03%) dan %system (24.73%) di poin 24 detik



Gambar 12 memori Snort pada saat FLOOD DOS Attack menyerang port 443

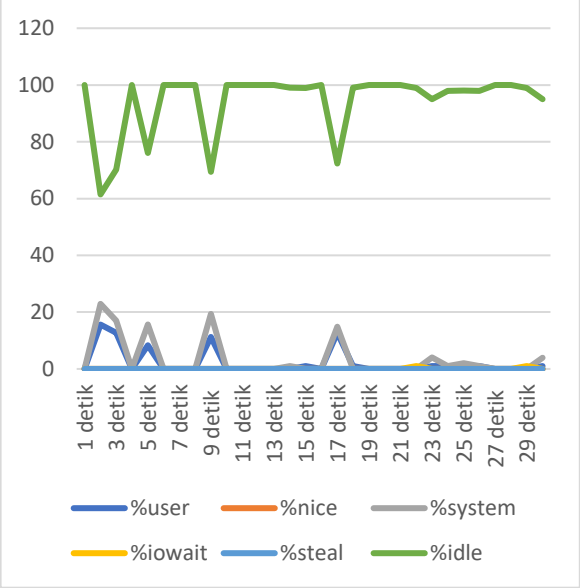
Penjelasan Gambar 12 :

- kbmemfree mengalami penurunan dari 1423152 kb di poin 1 detik menjadi 1416912 di poin 29 detik
- kbavail mengalami penurunan dari 1733952 kb di poin 1 detik menjadi 1729168 di poin 30 detik
- kbmemused mengalami kenaikan dari 61772kb di poin 1 detik menjadi 624012 kb di poin 29 detik
- kbbuffers mengalami kenaikan dari 18684 kb di poin 1 detik menjadi 18716 kb di poin 29 detik
- kbcached mengalami kenaikan dari 409436 kb di poin 1 detik menjadi 410864 kb di poin 30 detik
- kbcommit stabil dengan memori 734876 kb
- kbactive mengalami kenaikan dari 38320 kb di poin 1 detik menjadi 387080 kb di poin 30 detik
- kbinactive mengalami kenaikan dari 130480 kb di poin 1 detik menjadi 131884 kb di poin 30 detik
- kbdirty mengalami kenaikan dari 0 kb di poin 1 detik menjadi 1552 kb di poin 30 detik

Gambar 13 persentase CPU Snort pada saat FLOOD DOS Attack menyerang port 443

Penjelasan Gambar 13 persentasi memori :

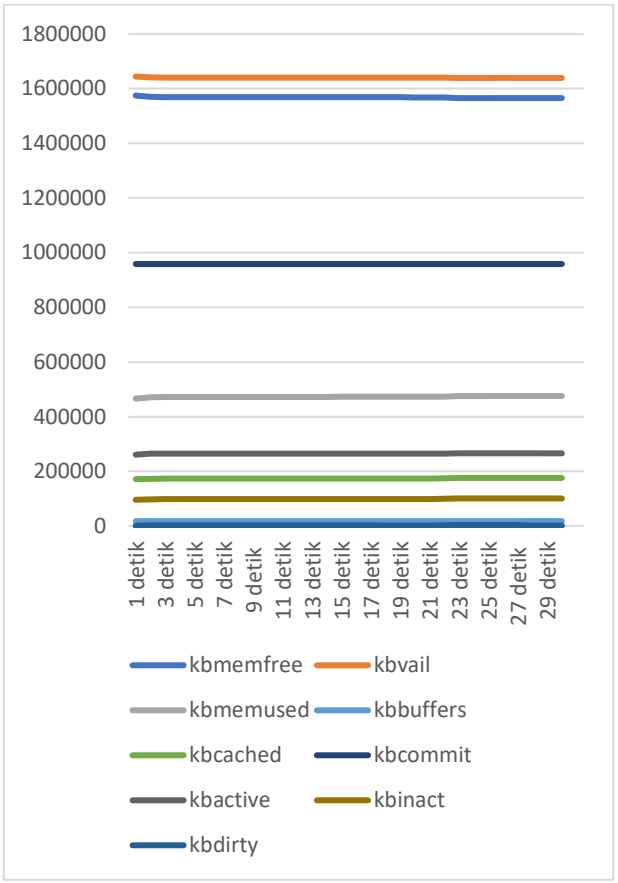
- %commit stabil dengan persentase 17.76%
- %memused mengalami kenaikan sedikit dari 30.27% di poin 1 detik menuju 30.57% di poin 23 detik



Gambar 14 persentase CPU Snort pada saat FLOOD DOS Attack menyerang port 443

Penjelasan Gambar 14 :

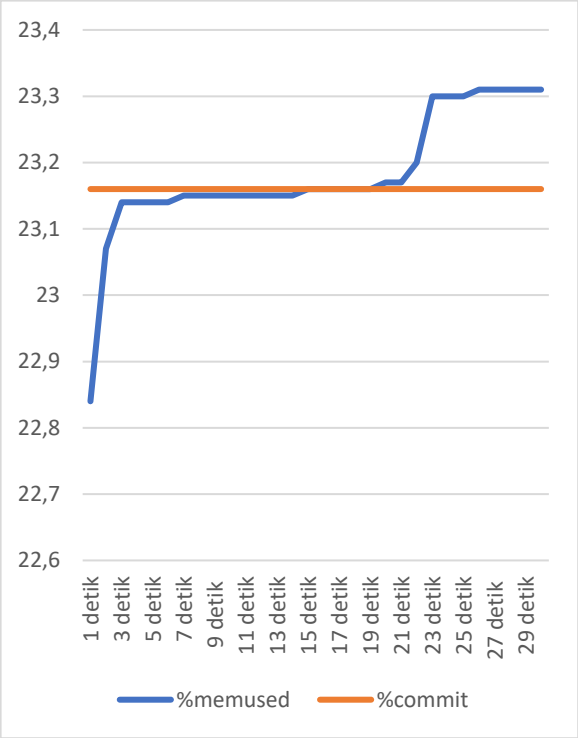
- idle berkurang menjadi menjadi 61.46% karena dari %user (15.62%) dan %system (22.92%) di poin 2 detik
- idle kembali berkurang menjadi 69.39% karena dari %user (11.22%) dan %system (19.39%) di poin 9 detik
- idle kembali berkurang menjadi 72.34% karena dari %user (12.77%) dan %system (14.89%) di poin 17 detik



Gambar 15 memori Suricata pada saat FLOOD DOS Attack menyerang port 443

Penjelasan Gambar 15 :

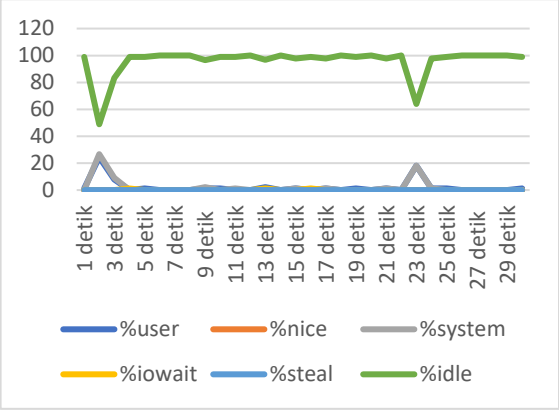
- kbmemsfree mengalami penurunan dari 1574732 kb di poin 1 detik menjadi 1565276 kb di poin 26 detik
- kbavail mengalami penurunan dari 1644168 kb di poin 1 detik menjadi 1639440 kb di poin 30 detik
- kbmemused mengalami kenaikan dari 466192 kb di poin 1 detik menjadi 475648 kb di poin 26 detik
- kbbuffers mengalami kenaikan dari 17880 kb di poin 1 detik menjadi 17928 kb di poin 28 detik
- kbcached mengalami kenaikan dari 171334 kb di poin 1 detik menjadi 175908 kb di poin 29 detik
- kbcommit mengalami kestabilan dengan memori 958464 kb
- kbactive mengalami kenaikan dari memori 261004 kb di poin 1 detik menjadi 265880 di poin 29 detik
- kbinact mengalami kenaikan dari memori 96164 kb di poin 1 detik menjadi 265880 kb di poin 29 detik
- kbdirty mengalami naik dan turun memori



Gambar 16 persentase memori Suricata pada saat FLOOD DOS Attack menyerang port 443

Penjelasan Gambar 16 persentasi memori :

- %memused mengalami kenaikan sedikit dari 22.84% di poin 1 detik menjadi 23.31% di poin 25 detik
- %commit mengalami kestabilan pada 23.16%



Gambar 17 persentase CPU Suricata pada saat FLOOD DOS Attack menyerang port 443

Penjelasan Gambar 17 :

- idle berkurang menjadi menjadi 48.89% karena dari %user (24.44%) dan %system (26.67%) di poin 2 detik
- idle kembali berkurang menjadi 64.04% karena dari %user (17.98%) dan %system (17.98%) di poin 23 detik

5. KESIMPULAN DAN SARAN

Pada bab ini menjelaskan tentang kesimpulan yang diperoleh dalam Pengujian Snort dan Suricata untuk mengetahui perbandingan tools Trading dengan menggunakan serangan jaringan dan sejumlah saran untuk pengembangan lebih lanjut.

5.1 Kesimpulan

Dari hasil pengujian dan analisis yang sudah dilakukan pada bab 5, maka dapat diperoleh beberapa kesimpulan sebagai berikut :

Dari hasil pengujian dan analisis yang sudah dilakukan pada bab 5, maka dapat diperoleh beberapa kesimpulan sebagai berikut :

1. Alert tipe rule 1 Snort sangat tinggi di banding dengan Suricata.
2. Untuk rule tipe 2 Snot dan Suricata seimbang,
3. Tetapi untuk rule tipe 3 dan 4 bagian port 443 itu lebih unggul sedikit dibanding dengan port 80.
4. Pada saat *flood DOS Attack* Snort memiliki rata – rata 93,5 % sedangkan Suricata memiliki 94,2 % di bagian CPU pada *port* 80. Ini menjelaskan bahwa Suricata unggul efisiensi di banding dengan Snort pada saat *flood* DOS Attack
5. Sama seperti poin sebelumnya, Snort memiliki rata – rata 94,5 % dan Suricata memiliki 95,67 % di bagian CPU pada *port* 443 membuktikkan bahwa Suricata unggul dalam aktifitas CPU
6. Untuk bagian memori Snort memiliki rata – rata 19.9 % untuk pemakaian memori atau efisien memori dibanding dengan Suricata yang emmiliki pemakaian memori rata – rata 69,3 % pada port 80
7. Pada port 443 Snort memiliki pemakaian memori rata – rata 30,3 % dibanding dengan Suricata yang memiliki rata – rata memori dengan pemakaian 30,5 % dalam berarti Suricata memiliki pemakaian lebih sedikit dibanding dengan Snort

5.2 Saran

Saran yang diberikan untuk penyempurnaan dan pengembangan lebih lanjut adalah sebagai berikut :

1. Untuk Snort :
   a. Perlu ditingkatkan kestabilan CPU
2. Untuk Suricata :
   a. Perlu dikurangkan penggunaan memori pada saat serangan DOS terjadi
   b. Untuk statistic Suricata perlu di perjelas.

DAFTAR PUSTAKA

[1] Acar, E. (2020, March 27). What is Snort? - Emrullah Acar. Medium (Retrieved 4 Februari 2021). https://medium.com/@acaremrullahkku/what-is-Snort547916bece5f

[2] Eril (2019, November 27). 13 TIPS Menjaga Keamanan Server Website. SSL Certificate Murah. https://gudangssl.id/tIPS-menjaga-keamanan-Server/

[3] Kuswanto, Dwi (2014). Unjuk Kerja Intrusion Prevention Sistem(IPS) berbasis Suricata pada Jaringan Local Area Network Laboratorium TIA+ Tehnik Informatikan Universitas Trunojoyo, NERO Vo.1 No.2, Hal 73-81. (Retrieved 4 Februari 2021)

[4] Rafiudin, R. (2010). Mengganyang Hacker dengan Snort (1st ed.). Penerbit Andi.

[5] Sharma, M. (2021, January 20). Open Source security software Snort gets a major upgrade. TechRadar (Retrieved 4 Februari 2021). https://www.techradar.com/news/open-source-security-software-Snort-gets-amajor-upgrade