BAB II
DASAR TEORI

1.1 TINJAUAN PUSTAKA

Dalam tinjauan pustaka ini menjadi salah satu acuan penulis untuk melengkapi penelitian, sehingga dapat menjadi tinjauan pustaka ketika mengevaluasi penelitian yang dilakukan. Mengenai beberapa studi banding yang tidak dapat dipisahkan dari topik penelitian, berikut adalah beberapa topik terkait:

No	Judul	Tahun	Penulis	Isi Penelitian	Perbedaan
1	Implementasi *Intrusion Detection System* (IDS) Menggunakan SNORT untuk mendeteksi Serangan pada Server	-	Ali Farhan, L.A. Syamsul Irfan Akbar dan A.Sjamsjiar	Pada penelitian ini bertujuan untuk mendeteksi dua serangan yang akan dilakukan, yaitu serangan ddos dengan alat DNS loic dan spoof dengan alat ettercap, setelah serangan terdeteksi, snort akan memberikan laporan secara *real time* sehingga sistem	1.Peneliti melakukan perbandingan performansi *tools* Snort, Suricata dan Zeek 2.Pengujian dengan jenis serangan ICMP *Flood* dan SYN *Flood* 3. Parameter performansi yang digunakan pada penelitian ini menggunakan parameter *Quality of Service* (QOS)

				admin jaringan dapat melakukan tindakan pengamanan di server.	
2	Implementasi Keamanan IDS/IPS Dengan SNORT dan IPTABLES pada Server	2020	Giovanni Tambunan, IGN Mantra	Pengamanan ditentukan memakai beberapa elemen fitur yang tersedia seperti melakukan Snort, IPTables, SSH, Firewall dan lainnya.	1.Peneliti melakukan perbandingan performansi *tools* Snort, Suricata dan Zeek 2.Pengujian dengan jenis serangan ICMP *Flood* dan SYN *Flood* 3. Parameter performansi yang digunakan pada penelitian ini menggunakan parameter *Quality of Service* (QOS)
3	Implementasi adan Analisa Keamanan Jaringan IDS (*Instrusion Detection System*) Menggunakan Suricata Pada Web Server	2020	Elsa Stephani, Fitri Nova dan Ervan Asri	Penelitian ini berfungsi untuk mengoptimalkan ancaman dengan cepat dan dapat mendeteksi serangan yang akan masuk	1.Peneliti melakukan perbandingan performansi *tools* Snort, Suricata dan Zeek 2.Pengujian dengan jenis serangan ICMP *Flood* dan SYN *Flood* 3. Parameter performansi yang digunakan pada penelitian ini menggunakan parameter *Quality*

					of Service (QOS)
4	Penerapan *Intrusion Prevention System* (IPS) Suricata Sebagai Pengamanan Dari Serangan *Distributed Denial Of Service* (DDos)	2020		Hasil penelitian ini bahwa IPS Suricata mampu mendeteksi serangan DDoS dan mampu memblokir akses serangan tersebut dengan memanfaatkan fitur firewall yaitu IPTables	1.Peneliti melakukan perbandingan performansi *tools* Snort, Suricata dan Zeek 2.Pengujian dengan jenis serangan ICMP *Flood* dan SYN *Flood* 3.Parameter performansi yang digunakan pada penelitian ini menggunakan parameter *Quality of Service* (QOS)
5	*A brief study and comparison of Snort and Bro Open Source Network Intrusion Detection Systems*	2012	Pritika Mehra	Melakukan *study* dan komparasi Snort dan Bro *open source* sebagai *Network Intrusion Detecion System*, komparasi Snort dan Bro diuji berdasarkan parameter seperti *speed, signatures,operatinng system capability*.	1.Peneliti melakukan perbandingan performansi *tools* Snort, Suricata dan Zeek 2.Pengujian dengan jenis serangan ICMP *Flood* dan SYN *Flood* 3.Parameter performansi yang digunakan pada penelitian ini menggunakan parameter *Quality of Service* (QOS)

1.2 DASAR TEORI

Dalam penelitian kali ini, penulis membutuhkan ide tentang topik penelitian, dan pada sub bab Teori Dasar, penulis mempertimbangkan ide dari berbagai sumber, seperti buku, surat kabar, artikel berita dan *website*. Pada bab ini, penulis menjelaskan teori jaringan komputer, teknik keamanan jaringan komputer dan tools yang berkaitan dengan penelitian ini.

1.2.1 KEAMANAN JARINGAN

Keamanan jaringan adalah sistem yang mencegah aktivitas yang tidak diinginkan dengan mengidentifikasi pengguna yang tidak memiliki hak akses ke jaringan. Dengan menghubungkan komputer ke komputer lain melalui jaringan kabel atau nirkabel, memungkinkan orang lain dapat mengakses data secara online, mengubah konten, dan bahkan menghapus data. [10].

Gambar 2.1 Segitia CIA [11]

Keamanan data didasarkan pada tiga prinsip dasar keamanan data, yaitu: *confidentiality*, *Integrity*, dan *availability*. adapun Penjelasan dari ketiga prinsip tersebut adalah:

1. *Confidentiality* (kerahasiaan)

   Confidentiality (kerahasiaan) dapat dengan mudah diartikan sebagai faktor kerahasiaan. Kerahasiaan dalam hal ini adalah kerahasiaan informasi maupun data, baik perusahaan dan kerahasiaan pengguna atau pelanggan seperti nasabah.

2. *Integrity* (integritas)

   Integritas Dalam keamanan informasi, yang kami maksud adalah integritas (kesetiaan atau kelengkapan). dan tidak dapat membuat, mengubah, atau

menghapus data tanpa izin. Dengan kata lain, kejujuran merupakan prinsip yang harus dijunjung tinggi untuk menjaga keutuhan informasi.

3. *Availabiity* (Ketersediaan)

   Konsep terbaik yang dapat memastikan dalam memelihara semua perangakat keras, dapat melakukan perbaikan terhadap perangakat keras sesegera mungkin saat diperlukan. Selain itu juga dapat memelihara atau menjaga dilingkungan sistem operasi.

1.2.2 *INTRUSION PREVENTION SYSTEM* (IPS)

Intrusion Prevention System (IPS) adalah perangkat lunak atau perangkat keras yang memantau lalu lintas jaringan. Mengidentifikasi aktivitas yang mencurigakan serta mencegah serangan yang dapat menggangu beroperasinya jaringan. IPS adalah cara paling umum untu membangun sistem keamanan komputer. Ada 2 jenis IPS, yaitu *Host Based Intrusion Prevention System* (HIPS) dan *Network Based Intrusion Prevention System*(NIPS).

1. *Host Intrusion Prevention System* (HIPS)

   Host Intrusion Prevention System (HIPS) sama seperti halnya *Host Based Intrusion Detection System* (HIDS). Program agent HIPS diinstall secara langsung di system yang diproteksi untuk dimonitor aktifitas sistem internal. HIPS juga bisa memantau aliran data dan aktivitas pada aplikasi tertentu. Sebagai contoh HIPS untuk mencegah *Intrusion* pada website misalnya. Dari sisi *security* mungkin solusi HIPS bisa mencegah datangnya ancaman terhadap *host*. Tetapi, dari sisi *performance*, harus diperhatikan apakah HIPS pada system operasi mengakibatkan penggunaan resource komputer host menjadi semakin besar.

2. *Network Intrusion Prevention System* (NIPS)

   Network-based Intrusion Prevention System (NIPS) tidak melakukan pantauan secara khusus di satu *host* saja. Tetapi melakukan pantauan dan proteksi II-16 dalam satu jaringan secara global. NIPS menggabungkan fitur IPS dengan *firewall* dan kadang disebut sebagai *In-Line* IDS atau *Gateway Intrusion Detection System* (GIDS). Sistem kerja IPS yang populer yaitu pendeteksian berbasis *signature*, pendeteksian berbasis anomali, dan *monitoring* file pada sistem operasi *host* [12].

2.2.3 SNORT

Snort adalah perangkat lunak yang dapat mendeteksi penyusup dan juga dapat menganalisis lalu lintas secara real time dan mendeteksi berbagai serangan [13]. Snort

tidak hanya merupakan protokol analisis IDS (*Intrusion Detection System*) atau Sistem Deteksi Intrusi, tetapi kombinasi keduanya dan dapat sangat berguna untuk merespon serangan terhadap *web host* [13].

Snort adalah NIDS yang bekerja dengan menggunakan signature detection, dan bekerja sebagai pelacak dan pencatat paket. Snort awalnya dibuat oleh Marti Roesh dan kemudian menjadi sebuah *open source* [13]. Snort dapat dioperasikan dengan 3 mode yaitu:

1. *Paket Sniffer*
   Ini digunakan untuk membaca paket data dari jaringan dan menampilkan status aliran tanpa henti pada kontrol (layar).
2. *Packet Logger*
   berfungsi untuk mencatat paket-paket kedalam *disk*.
3. NIDS (*Network Intrusion Detection System*) mode
   Dalam mode ini, snort mendeteksi serangan dari jaringan komputer.

2.2.4 SURICATA

Suricata merupakan sebuah mesin yang dapat pendeteksi ancaman/serangan *open source*. Mesin Suricata dapat mendeteksi serangan jaringan sistem deteksi intrusi dan segera melakukan online *intrusion prevention* System (IPS), *network security monitoring* (NSM) dan konfigurasi offline PCAP [14].

Suricata menganalisis jalur jaringan dengan menggunakan aturan dan korelasi yang kuat serta menggunakan *signature language* dan mendukung penggunaan skrip LUA yang kuat sehingga dapat mendeteksi serangan. Dengan menggunakan format *input* seperti YAML dan JSON begitu juga dengan *output*, integrasi dengan seperti SIEM, Splunk, Logstah/Elasticsearch, Kibana, dan database lainnya menjadi mudah. Proyek pengembangan pengguna Suricata berfokus pada keamanan, kegunaan, dan kinerja [14].

Suricata dirancang sebagai sistem *multi-threaded* yang dapat menggunakan banyak *core*. Di mesin yang sama, Snort ditemukan bekerja lebih baik daripada Suricata. Suricata menunjukkan performa tinggi di mesin multicore dengan aturan terbanyak untuk Suricata. Suricata dapat dengan mudah mengatur volume lalu lintas tanpa mengurangi jumlah pesanan. Suricata juga dikenal karena kemampuannya memantau visibilitas lapisan aplikasi dan mempercepat penjelajahan HTTP. Oleh karena itu, Suricata juga dapat memeriksa jalur HTTP terlepas *port* yang digunakan dan tidak bergantung pada nomor *port* untuk mengidentifikasi jalur jaringan. Suricata juga memungkinkan

inspeksi dalam aliran log dan karenanya dapat mengekstrak *file* dari sesi HTTP untuk pemeriksaan lebih lanjut [14].

2.2.5 ZEEK

Zeek merupakan penganalisis lalu lintas jaringan yg bersifat pasif. Banyak operator jaringan menggunakan Zeek sebagai alat pemantau keamanan jaringan untuk menyelidiki aktivitas yang mencurigakan atau berbahaya. Manfaat pertama yang didapat pengguna baru Zeek yaitu koleksi log yang sangat luas yang menggambarkan aktivitas jaringan. Log ini mungkin tidak berisi semua rekaman untuk semua antarmuka kabel yang terlihat, tetapi juga transkrip lapisan aplikasi [15].

Zeek menggunakan libpcap untuk menangkap paket di jaringan, dan kemudian mesin dapat menerima lalu lintas yang difilter dari libpcap. Proses ini melakukan berbagai pemeriksaan integritas untuk memastikan bahwa paket dibentuk dengan benar dan mendeteksi kesalahan pada *header* IP. Fragmen IP dirakit untuk proses ini karena penganalisa dapat menggunakan seluruh paket IP untuk mengirimkan kejadian ke lapisan kebijakan. *Policy Script Interpreter* dalam skrip Zeek dengan metode deteksi *independen*. zeek menggunakan metode analisis jaringan yang menggunakan deteksi serangan mencurigakan [15].

2.2.6 DDOS (*DISTRIBUTED DENIAL OF SERVICES*)

DdoS terjadi ketika *hacker* memdati jaringan dengan melalui jalur untuk membebani sistem dan mengganggu kinerja jaringan. Serangan semacam itu biasanya digunakan untuk membuat situs web atau aplikasi offline sementara dan dapat berlangsung selama berhari-hari atau bahkan lebih lama.

DDoS atau *Distributed Denial of Services* menurut definisi adalah penolakan layanan terdistribusi. DDoS adalah jenis serangan yang menjadi jalur jaringan Internet hingga ke server, jaringan, atau sistem. Biasanya, serangan ini dilakukan menggunakan beberapa komputer *host* yang menyerang hingga komputer target menjadi tidak dapat digunakan. [16]. Beberapa jenis serangan DDOS adalah sebagai berikut [16]:

1. *Ping of Death*

   Ping of Death adalah serangan yang mengirimkan beberapa *ping* berbahaya ke komputer dengan alamat IP hingga 65535 *byte*, paket normal biasanya 84 *byte*. Hal ini menyebabkan *buffer overflows* pada target, mengakibatkan *denial of service* untuk paket yang valid untuk *host* tersebut yang mengakibatkan *crash* pada komputer.

2. UDP *Flooding*

   User Datagram Protocol (UDP) adalah protokol Internet yang mengirim pesan melalui jaringan ke komputer lain tanpa menunggu konfirmasi dari komputer target. Dalam serangan DDoS jenis ini, target membanjiri paket UDP, sehingga *port* membanjiri komputer target dengan *port* acak. Ini menyebabkan komputer berulang kali memeriksa *port* yang diminta, dan jika *port* tidak ditemukan, komputer target merespons dengan informasi bahwa *port* yang dikirim tidak ditemukan. Proses ini sangat memakan sumber daya komputer Anda (CPU dan memori), pada akhirnya dapat membuat komputer Anda tidak dapat diakses.

3. SYN *Flooding*

   SYN *Flood* adalah serangan yang mengirimkan permintaan SYN dalam jumlah besar ke *host* target tetapi tidak menanggapi SYN ACK. Dalam skenario ini, pengirim meminta beberapa SYN palsu yang menyebabkan komputer menunggu pemberitahuan untuk setiap permintaan, sehingga komputer target menghabiskan sumber daya dan tidak dapat tersambung.

4. ICMP (Ping) *Flood*

   ICMP *Flood* melakukan serangan dengan membanjiri sumber daya komputer target dengan paket ICMP *Echo Request*, biasanya mengirimkan paket dengan cepat tanpa menunggu respon dari komputer target. Jenis serangan ini dapat menghabiskan *bandwidth* yang keluar dan masuk karena komputer target selalu berusaha merespon paket ICMP *Echo Reply*.

5. *Smurf Attack*

   Smurf Attack merupakan penyerangan yang memanfaatkan ICMP *echo request*, yang sering digunakan untuk mengirim identitas kedalam sebuah jaringan. semua komputer yang terhubung ke jaringan juga akan menanggapi permintaan tersebut.

2.2.7 WIRESHARK

Wireshark merupakan penganalisa paket jaringan yang tersedia secara *open source* dan menangkap paket data yang melewati jaringan dan menampilkannya kedalam format yang dimengerti. Wireshark juga dianggap sebagai pisau milik tentara Swiss karena dapat digunakan dalam berbagai situasi seperti pemecahan masalah terhadap layanan keamanan, jaringan, dan program pelatihan internal. Wireshark dapat membaca data secara langsung dari koneksi Ethernet, Token Ring, FDDI, Serial (PPP dan SLIP), LAN

wireless 802.11, dan ATM. Alat ini dapat menangkap paket data/informasi yang berjalan di jaringan.

Alat ini dapat menangkap paket data/informasi yang berjalan di jaringan. Semua jenis paket data dalam format log yang berbeda dapat ditangkap dan dianalisis dengan mudah. Karena seringkali *tool* ini juga bisa digunakan untuk *sniffing* (menangkap informasi penting seperti *password* email atau akun lainnya) dengan cara menangkap dan menganalisa paket-paket yang berjalan di dalam jaringan. menangkap paket-paket data/informasi yang berjalan dalam jaringan [17].

Wireshark mendukung banyak paket/format file tangkapan, termasuk cap dan erf. Selain itu, alat dekripsi bawaan dapat mengekstraksi paket terenkripsi dari banyak protokol yang biasa digunakan kedalam jaringan Internet, termasuk WPA/WPA2 dan WEP. Salah satu keunggulan Wireshark yaitu distribusi pengembangannya yang bersifat lintas platform, sehingga pengguna atau user Macintosh dan Linux juga dapat menggunakan Wireshark.

2.2.8 *QUALITY OF SERVICE* (QOS)

Quality of Service (QOS) adalah metode untuk mengukur kualitas jaringan dalam upaya untuk menentukan kualitas layanan. QoS digunakan untuk mengukur karakteristik kinerja tertentu dan dikaitkan dengan sebuah layanan [18].

Model *monitoring* QoS terdiri dari beberapa komponen yaitu QoS *monitoring, monitoring application*, *monitored objects*, dan *monitor*. Berikut adalah beberapa penjelas dari setiap model nya [18]:

1. *Monitoring Application*

   adalah antarmuka pengguna untuk administrator jaringan. Komponen ini mengambil informasi lalu lintas dari paket data pemantauan, menganalisisnya, dan mengirimkan hasil analisisnya kepada pengguna.

2. *Qos Monitoring*

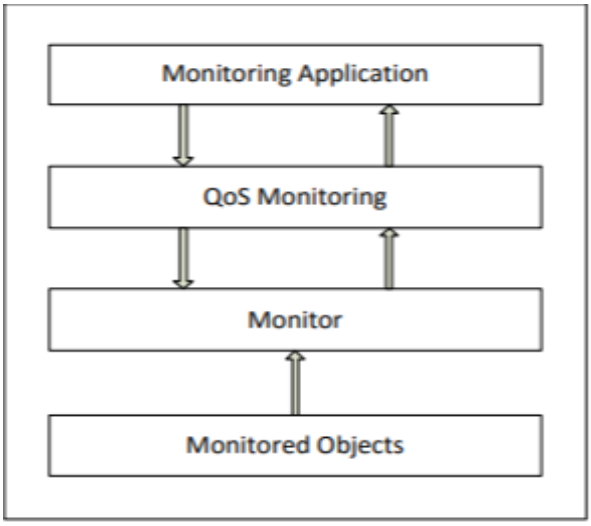
   Menyediakan sistem pemantauan QoS dengan mengambil informasi tentang nilai parameter QoS dari lalu lintas data paket.

3. *Monitor*

   mengumpulkan dan menyimpan informasi lalu lintas paket, yang dikirim ke *monitoring application*. *Monitor* mengukur paket data secara bersamaan dan melaporkan hasilnya ke *monitoring application*.

4. *Monitored Objects*

Ini adalah informasi seperti identitas dan aktivitas yang dipantau di jaringan. Dalam pemantauan QoS, informasi adalah aliran paket yang dipantau secara langsung.



Gambar 2.2 Model *Monitoring* QOS [18]

2.2.9 *THROUGHPUT*

Bandwidth sebenarnya yang diukur pada waktu tertentu saat mengirim sebuah file. Tidak seperti bandwidth, meskipun satuannya sama dalam *bits per second* (bps), tetapi *throughput dapat* lebih baik dalam menggambarkan bandwidth yang aktual pada waktu tertentu dan dalam kondisi tertentu serta jaringan yang digunakan untuk mengunduh ukuran file tertentu. *Throughput* adalah jumlah total paket yang berhasil terdeteksi dalam interval tertentu dibagi dengan durasi interval tersebut. Nilai *throughput* dapat dihitung menggunakan persamaan:

$$Throughput = \frac{Paket\ data\ yang\ diterima\ (bit)}{Waktu\ pengiriman\ paket\ (second)}$$

2.2.10 *PACKET LOSS*

Ini adalah persentase paket yang hilang selama transmisi data. Alasannya banyak hal seperti sinyal lemah di jaringan, kesalahan perangkat keras jaringan, atau radiasi lingkungan. *Packet loss* adalah parameter yang menggambarkan situasi yang menunjukkan jumlah paket yang hilang yang dapat disebabkan oleh tabrakan (*collision*) dan kemacetan (*congestion*) di jaringan.

$$Packet\ Loss = \frac{(Paket\ data\ yang\ dikirim - paket\ data\ yang\ diterima)x100\%}{Paket\ data\ yang\ dikirim}$$

2.2.11 *JITTER*

Jitter adalah perubahan *delay* (perbedaan ruang waktu) antar paket dalam jaringan akibat panjangnya antrian saat memproses data dalam jaringan. Nilai *jitter* dipengaruhi oleh beban trafik dan jumlah paket *(congestion)* dalam jaringan. Lalu lintas tinggi dan lalu lintas jaringan. Untuk menghitung nilai *jitter*, gunakan persamaan:

$$Jitter = \frac{Total\ variasi\ delay}{Total\ paket\ yang\ diterima}$$

2.2.12 *DELAY*

Delay atau *Latency* adalah waktu yang dibutuhkan data untuk melakukan perjalanan dari sumber ke tujuan. Keterlambatan dipengaruhi oleh jarak, media fisik, kemacetan (*Congestion*) waktu pemrosesan yang lama.

$$Delay = \frac{Total\ delay}{Total\ paket\ yang\ diterima}$$