Penerapan dan Analisis *Network Security Snort* Menggunakan *Intrusion Detection System* pada Serangan *UDP Flood*

Karina Fitriwulandari Ilham*, Erick Irawadi Alwi, Farniwati Fattah*****
*,*** Prodi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Muslim Indonesia
** Prodi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Muslim Indonesia
*karinaaa2816@gmail.com, **erick.alwi@umi.ac.id, ***farniwati.fattah@umi.ac.id

ABSTRACT

The web server is one type of server that was most frequently attacked because the web server is the most vital means for storing databases. The most frequently used attack is Denial of Service (DoS). DoS is an attack technique against a system by consuming data resources from a server so that the server cannot be accessed again. Therefore, this study aims to design a network security system by implementing an Intrusion Detection System (IDS) using the Snort application as a detection system in the event of a DoS attack on the web server. by implementing Snort on a web server will help network administrators to ensure that the web server is protected from various threats. The results of this study indicates that the Snort rule that has been created has been successfully detecting attacks that have been tested through the LOIC tools.

Keywords: IDS, Snort, LOIC, DoS.

1. Pendahuluan

Perkembangan teknologi informasi yang sangat pesat saat ini, membuat teknologi berperan sangat penting dalam kehidupan sehari-hari, khususnya jaringan komputer. Dengan adanya jaringan komputer kita dapat dengan mudah mengakses suatu informasi dan data. Di era saat ini hampir seluruh manusia mengandalkan jaringan komputer untuk menyimpan data dan informasi.

Server adalah komputer yang mendukung aplikasi dan telekomunikasi dalam jaringan, dan mendistribusikan peralatan *software* dan *database* diantara berbagai terminal kerja dalam jaringan [1]. *Server* juga menjalankan perangkat lunak administratif yang mengontrol akses ke jaringan dan sumber daya yang dikandungnya [2]. *Server* dengan keamanan jaringan yang lemah dapat dimanfaatkan oleh oknum yang tidak bertanggung jawab untuk mencuri suatu data atau informasi. *Web server* adalah host yang paling ditargetkan dan diserang di lingkup jaringan suatu organisasi [2]. Oleh karena itu, diperlukan suatu sistem keamanan jaringan yang dapat memantau apakah *web server* sedang diserang atau tidak. Salah satu cara untuk meningkatkan keamanan *web server* adalah dengan mengimplementasikan *Intrusion Detection System* (IDS) Snort. IDS adalah sistem yang mampu mendeteksi serangan dan ancaman yang terjadi pada sebuah jaringan komputer. IDS akan memberikan peringatan dini kepada administrator jaringan ketika terjadi sebuah aktivitas mencurigakan terjadi di jaringan komputer. Snort adalah aplikasi keamanan yang fungsinya untuk mendeteksi intrusi-intrusi jaringan (penyusupan, serangan, pemindaian, dan lain-lain), dan sekaligus untuk melakukan pencegahan [3]. Jika aktivitas mencurigakan terkait dengan *traffic* jaringan ditemukan, IDS akan memperingatkan sistem atau administrator jaringan [4]. Snort adalah perangkat lunak untuk mendeteksi penyusup yang dapat menganalisis paket data yang melintasi jaringan secara *real time* dan mencatatnya ke dalam *database*, serta dapat mengidentifikasi berbagai serangan dari luar jaringan [5]. Snort dapat mendeteksi penyusup dan mampu menganalisis paket data yang melewati jaringan secara *real time* dan melakukan *login* ke *database* serta mendeteksi berbagai serangan dari luar jaringan [6].

Salah satu bentuk serangan yang bersifat merugikan yaitu serangan *Denial of Service* (DoS) jenis *User Datagram Protocol* (*UDP*) *Flood*. DoS adalah teknik serangan terhadap sistem yang dapat membuat sistem tersebut tidak dapat dikases. UDP *Flood* dapat menyebabkan komputer *server* yang menjadi target mengalami *error* akibat banyaknya jumlah paket yang diterima komputer *server* tersebut [2]. UDP *Flood* merupakan serangan yang tidak memperhatikan apakah paket yang dikirim diterima atau tidak oleh komputer *server* [7]. UDP *Flood* ini akan mengirimkan karakter yang akan menguji jaringan korban, sehingga terjadi aliran data

yang tidak perlu dalam jaringan korban tersebut [8]. *Tool* yang digunakan untuk melakukan penyerangan UDP *Flood* yaitu LOIC. LOIC adalah sebuah *tool* atau aplikasi yang fungsinya untuk melumpuhkan *server* dengan cara mengirimkan paket sebanyak-banyaknya sesuai keinginan penyerang ke komputer *server* yang dituju melalui ip *server* komputer target [9].

Penelitian sebelumnya telah membahas mengenai implementasi IDS untuk mendeteksi adanya serangan DoS pada *web server*. Pada penelitian [5] membahas mengenai implementasi IDS Snort sebagai keamanan jaringan di SMAN 1 Cikesual. Namun penelitian tersebut masih menggunakan *Virtual Machine* untuk melakukan pengujian sistem IDS Snort, hal itu menyebabkan PC menjadi *hang* dikarenakan VM bisa memakan cukup banyak lokasi *hardware* pada PC selain itu juga memakan ruang penyimpanan data *virtual* yang lumayan besar dan mengambil sebagian fungsi *processor* dan RAM, akibatnya IDS Snort tidak dapat bekerja secara maksimal dalam memonitoring *traffic* jaringan. Maka dari itu, penelitian ini akan menggunakan *real machine* agar IDS Snort dapat bekerja secara maksimal untuk mendeteksi serangan *UDP Flood*.

Penulis bermaksud mengimplementasi sistem keamanan jaringan menggunakan IDS Snort untuk mendeteksi serangan *UDP Flood*. Pada penelitian ini akan dilakukan penyerangan *UDP Flood* menggunakan LOIC sebagai *tools* penyerangan. Lalu melakukan penyerangan dengan jumlah *threads* yang berbeda yaitu 500 dan 1000 *threads* kemudian, akan dilihat keefektifan *rule* dalam mendeteksi serangan *UDP Flood*.

2. Metode Penelitian

Metode penelitian yang digunakan pada penelitian ini yaitu *action research*. *Action research* atau metode penelitian tindakan merupakan metode penelitian yang malakukan praktik sekaligus berteori atau menggabungkan teori sekaligus melaksanakan praktik [10]. Pada penelitian ini terdapat terdapat 5 tahapan penelitian yang digunakan yaitu diagnosis (*diagnosing*), rancangan tindakan (*action planning*), pelaksanaan tindakan (*action taking*) dan evaluasi (*Evaluating*).

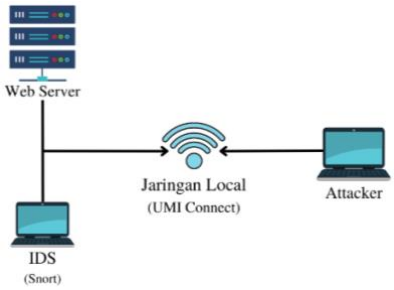
2.1 Tahap Diagnosis (*Diagnosing*)

Pada tahapan ini penulis mengindentifikasi masalah-masalah pokok yang ada. Seperti pada masalah keamanan jaringan pada *web server*. Untuk pengembangan pada tahap ini peneliti mengintifikasi kebutuhan untuk mengatasi masalah keamanan jaringan pada *web server*.

2.2 Tahap Rancangan Tindakan (*Action Planning)*

Pada tahapan ini penulis memahami pokok masalah yang ada kemudian membuat skema rencana tindakan yang tepat untuk menyelesaikan permasalahan yang ada. Pada tahap ini akan dibuat desain topologi jaringan, rancangan alur penelitian sistem, skenario penyerangan, kebutuhan perangkat lunak dan kebutuhan perangkat keras.

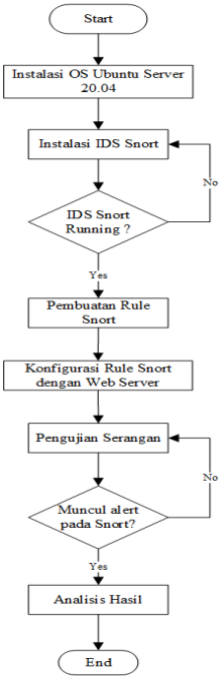
2.2.1 Desain Topologi Jaringan



Gambar 1. Desain Topologi Jaringan

Pada Gambar 1 merupakan desain topologi jaringan. Pada topologi tersebut terdapat 2 PC yaitu, PC 1 berperan sebagai pendeteksi adanya serangan pada *web server* (IDS). Kemudian PC 2 berperan sebagai *attacker* yang akan melakukan serangan pada *web server*. Topologi ini mensimulasikan bahwa pengujian serangan dilakukan melalui komputer *attacker* yang terhubung pada jaringan lokal yang sama dengan *web server*. Lalu *attacker* akan melakukan penyerangan dengan cara mengakses *ip addre*s dan IDS akan melakukan monitoring jaringan terhadap paket-paket yang masuk pada sistem.

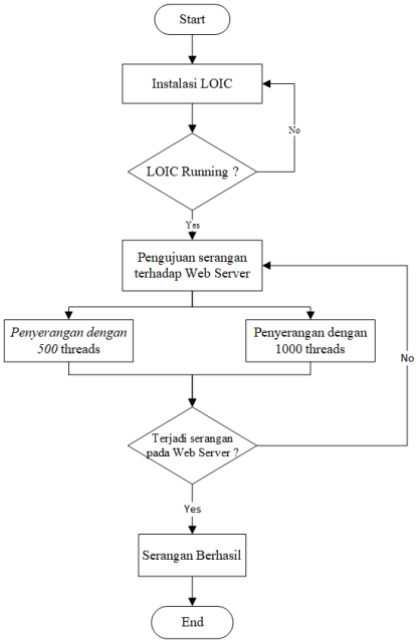
2.2.2 Desain Alur Penelitian Sistem



Gambar 2. Desain Alur Penelitian

Pada Gambar 2 merupakan desain alur penelitian sistem yang akan dilakukan pada penelitian ini, dimana menggunakan OS Ubuntu *Server*. Alur pengujian tersebut akan dilakukan pada PC 1 (PC IDS). *Tools* yang digunakan yaitu Snort sebagai sistem IDS yang akan memonitoring *traffic* jaringan. Objek serangan yang digunakan yaitu *web server*. Seperti pada gambar , dimulai dengan menginstal OS Ubuntu *Server* 20.04. Lalu menginstal IDS Snort, setelah itu membuat *rule* pada Snort dilanjutkan dengan konfigurasi IDS Snort dengan *web server*. Setelah itu melakukan pengujian serangan pada *web server*. Jika berhasil, maka muncul *alert* pada IDS Snort. Tahap berikutnya yaitu, mengambil kesimpulan hasil anilisis sistem *alert* yang dibuat.

2.2.3 Desain Skenario Penyerangan



Gambar 3. Desain Skenario Penyerangan

Pada Gambar 3, merupakan desain alur pengujian penyerangan pada *web server* menggunakan *tools* LOIC dengan serangan DoS jenis *UDP Flood*. Penyerangan ini dilakukan pada PC 2 (PC *attacker*). Proses pertama yang dilakukan yaitu instalasi LOIC, lalu jika berhasil maka langsung menguji serangan pada *web server* dengan mengirimkan jumlah *threads* yang berbeda yaitu 500 dan 1000 *threads*.

2.2.4 Kebutuhan Perangkat Keras (*Hardware*)

Tabel 1. Daftar kebutuhan *hardware*

Nama Perangkat	Spesifikasi	Fungsi
Server	*Processor*: GenuineIntel Common KVM *processor dual core* RAM: 2048 MB	Sebagai *Server* yang akan diserang
Laptop MSI Modern 14	*Processor*: Intel *Core* i7 RAM: 16 GB SSD: 500 GB	Sebagai laptop *attacker*

2.2.5 Kebutuhan Perangkat Lunak (*Software*)

Tabel 2. Daftar kebutuhan *software*

Perangkat Lunak	Keterangan
OS Ubuntu Server 20.04	Sebagai sistem operasi *server*
OS Kali Linux 2022.2	Sebagai sistem operasi *attacker*
Snort	*Tools* yang digunakan sebagai sistem monitoring jaringan
LOIC	*Tools* yang digunakan untuk melakukan penyerangan
PuTTY	*Tools* digunakan untuk mengakses *server*

2.3 Tahap Pelaksanaan Tindakan (*Action Taking*)

Pada tahap ini peneliti mengimplementasikan skema rencana yang telah dibuat dengan harapan dapat menyelesaikan masalah. Pada tahap ini peneliti melakukan uji coba atau mengimplemntasikan IDS Snort serta melakukan konfigurasi Snort untuk mendeteksi serangan *UDP Flood*.

2.4 Tahap Evaluasi (*Evaluating*)

Setelah melakukan uji coba dan mendapatkan hasil yang cukup kemudian peneliti melakukan evaluasi dari hasil implementasi.

3. Hasil dan Pembahasan

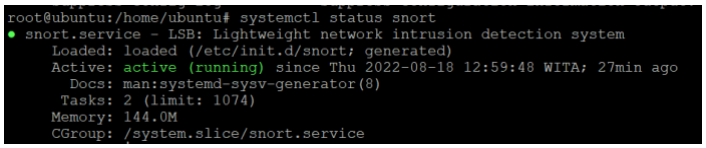
3.1 Pelaksanaan Tindakan (*Action Taking*)

Pada tahap ini peneliti melakukan uji coba atau mengimplemntasikan IDS Snort serta melakukan konfigurasi Snort untuk mendeteksi serangan *UDP Flood*.

3.1.1 Instalasi IDS Snort

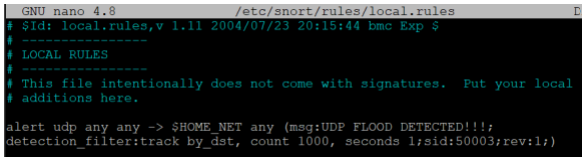
Langkah pertama yang dilakukan yaitu melakukan instalasi IDS Snort pada komputer server. Untuk mengakses server telah dibuat menggunakan tools PuTTY. Kemudian menjalankan perintah #apt-get install snort. Setelah melaukan instalasi Snort, dilanjutkan dengan konfigurasi Snort dengan menjalan perintah #nano /etc/snort/snort.conf.

Untuk mengecek status Snort apakah Snort sudah aktif atau belum maka dapat dijalankan perintah #systemctl status snort. Gambar 4 merupakan *output* dari perintah tersebut



Gambar 4. Status Snort

3.1.2 Pembuatan *Rule* Snort



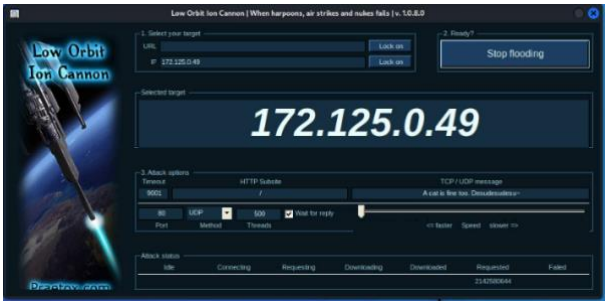
Gambar 5. Rule Snort

Pada Gambar 5 merupakan *library* dari *rule* Snort yang akan digunakan. *Rule* pada Snort menggunakan *rule* yang dibuat dengan menyesuaikan pada jenis serangan yang akan diujikan. Pada Snort *rule* dibuat pada file */etc/snort/rules/local.rules*.

3.1.3 Pengujian Serangan

Pengujian serangan *UDP Flood* menggunakan *tools* LOIC dengan jumlah *threads* yang berbeda yaitu 500 *threads* dan 1000 *threads*.

1) Pengujian serangan 500 *threads*



Gambar 6. Pengujian 500 *threads*

Pada Gambar 6 merupakan tampilan LOIC setelah melakukan uji coba serangan menggunakan metode *UDP Flood*. Dengan jumlah threads yang dikirimkan sebanyak 500 threads dan jumlah paket yang dikirim sebanyak 2.142.580.644 paket.

2) Pengujian serangan 1000 *threads*

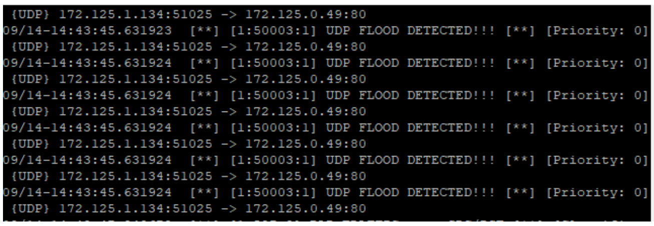
Gambar 7. Pengujian 1000 *threads*

Pada Gambar 7 merupakan tampilan LOIC setelah melakukan uji coba serangan menggunakan metode *UDP Flood*. Dengan jumlah *threads* yang dikirimkan sebanyak 1000 *threads* dan jumlah paket yang dikirim sebanyak 2.143.586.358 paket.

3.2 Hasil Dan Analisis

Setelah melakukan tahap *action planning* maka akan didapatkan hasil analisis, hasil tersebut merupakan tahap *evaluating*. Berikut merupakan hasil dari penelitian ini :

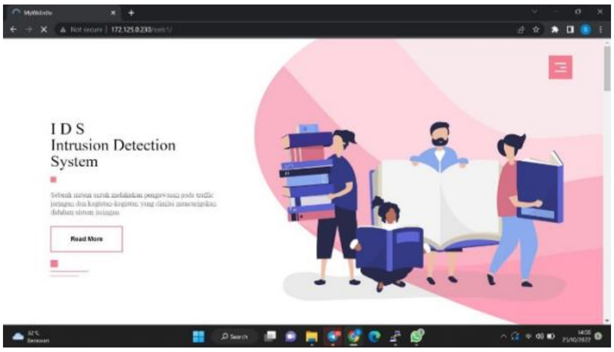
3.2.1 Hasil Monitoring Snort



Gambar 8. Hasil monitoring Snort

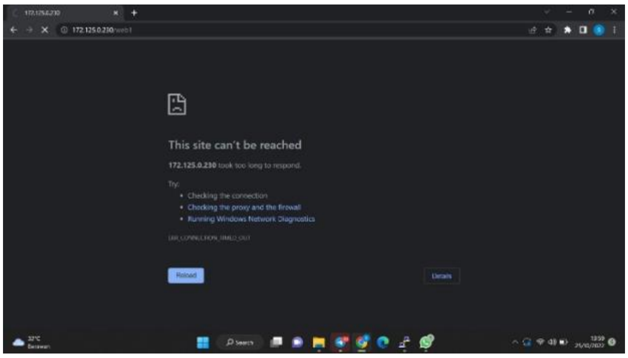
Pada Gambar 8 merupakan hasil monitoring Snort bila terjadi adanya serangan UDP *Flood*. Hasil monitoring tersebut disebut dengan *alert* atau peringatan bila terjadi. *Alert* tersebut menampilkan pesan *alert*, waktu, jenis serangan dan juga informasi *ip address* penyerang sesuai dengan *rule* yang telah dibuat sebelumnya.

3.2.2 Dampak Serangan pada *Web Server*



Gambar 9. Tampilan *web server* sebelum serangan

Pada Gambar 9 merupakan tampilan dari *web server* sebelum terjadinya serangan UDP *Flood.*



Gambar 10. Server down

Pada gambar 10 merupakan tampilan dari *web server* setelah mengalami serangan. *Web server* tersebut mengalami *down* dan susah untuk diakses.

3.2.3 Tabel Hasil Pencatatan dan Akurasi Waktu Serangan

Tabel 3. Hasil pencatatan serangan dan akurasi waktu serangan

Jenis Serangan	Jumlah *Threads* yang Dikirim	Status Serangan pada Snort	Total Durasi Penyerangan	Total Paket yang Terdeteksi
UDP Flood	500 *threads*	Terdeteksi	57 menit	26.116.253 paket
	1000 *threads*	Terdeteksi	1 jam 4 menit	30.066.287 paket

Pada tabel 3 merupakan tabel dari hasil pencatatan serangan yang terdeteksi oleh Snort. Pada tabel tersebut terbukti bahwa Snort dapat mendeteksi jumlah paket yang masuk dan tercatat kedalam *log file*. Dengan adanya bukti hasil pencatatan tersebut dapat membantu administrator jaringan untuk melakukan tindakan sela njutnya dalam menanggulangi jika terjadi kembali serangan UDP *Flood*.

4. Kesimpulan

Berdasarkan alur skenario pengujian sistem yang telah dibuat maka kesimpulan dari penlitian ini adalah sebagai berikut :

1) Berdasarkan *rule* yang telah dibuat, Snort dapat mendeteksi adanya serangan *UDP Flood* dengan jumlah intesitas serangan yang berbeda yaitu 500 t*hreads* dan 1000 *threads*.
2) *Log file* Snort berhasil mencatat paket yang masuk dan durasi waktu penyerangan. Untuk penyerangan dengan intensitas 500 *threads log file* Snort mencatat total paket yang masuk yaitu sebanyak 26.116.253 paket. Dan untuk penyerangan dengan intensitas 1000 *threads log file* Snort mencatat total paket yang masuk yaitu sebanyak 30.066.287 paket.
3) Dari hasil analisis diperoleh bahwa setiap jumlah *threads* serangan yang telah diimplementasikan untuk membuat *server down* membutuhkan waktu yang berbeda.

5. Saran

Adapun saran penulis sebagai pengembangan penelitian ini yaitu :

1) Mengembangkan *rule* yang telah ada untuk dapat mendeteksi diluar dari serangan DoS.
2) Menambahkan fitur *block* serangan untuk keamanan *web server* yang lebih maksimal.

Daftar Pustaka

[1] G. Tambunan and M. IGN, "Implementasi Keamanan Ids / Ips Dengan Snort Dan," *Semin. Nas. Mhs. Ilmu Komput. dan Apl. Jakarta-Indonesia, 28 Januari 2020 IMPLEMENTASI*, pp. 10–16, 2020.

[2] Fahmi Bagaskara Perdana, M. . Dr. Ir. Rendy Munadi, and M. . Arif Indra Irawan, S.T., "Implementasi Sistem Keamanan Jaringan Menggunakan Suricata Dan Ntopng," *e-Proceeding Eng.*, vol. 6, no. 2, p. 4076, 2019.

[3] M. H. Dar and S. Z. Harahap, "Implementasi Snort Intrusion Detection System (Ids) Pada Sistem Jaringan Komputer," *J. Inform.*, vol. 6, no. 3, pp. 14–23, 2017, doi: 10.36987/informatika.v6i3.1619.

[4] E. S. J. Atmadji, B. M. Susanto, and R. Wiratama, "Pemanfaatan IPTables Sebagai Intrusion Detection System (IDS) dan Intrusion Prevention System (IPS) Pada Linux Server," *Teknika*, vol. 6, no. 1, pp. 19–23, 2017, doi: 10.34148/teknika.v6i1.55.

[5] A. P. Pancaro and F. I. Saputra, "IMPLEMENTASI IDS (INTRUSION DETECTION SYSTEM) PADA SISTEM KEAMANAN JARINGAN SMAN 1 CIKEUSAL," vol. 5, no. 1, 2018.

[6] Y. P. Atmojo, "Analisa Performa Raspberry Pi sebagai Intrusion Detection System: Studi Kasus IDS Pada Server Web," *Eksplora Inform.*, vol. 8, no. 1, p. 24, 2018, doi: 10.30864/eksplora.v8i1.143.

[7] K. Ramadhani, M. Yusuf, H. E. Wahanani, J. T. Informatika, and F. T. Industri, "Anomali Perubahan Traffic Jaringan Berbasis Cusum," *Penelitian*, no. pendeteksian dini serangan UDP FLOOD, pp. 1–9, 2008.

[8] 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.

[9] P. Panggabean, "Analisis Network Security Snort Metode Intrusion Detection System Untuk Optimasi Keamanan Jaringan Komputer," *Jursima*, vol. 6, no. 1, p. 1, 2018, doi: 10.47024/js.v6i1.107.

[10] Hasan, "Action Research : Desain Penelitian Integratif untuk Mengatasi Permasalahan Masyarakat," *AKSES J. Ekon. dan Bisnis*, vol. 4, no. 8, p. 12, 2009, [Online]. Available: https://publikasiilmiah.unwahas.ac.id/index.php/AKSES/article/view/523