Prosiding
ANNUAL RESEARCH SEMINAR 2016
6 Desember 2016, Vol 2 No. 1

ISBN : 979-587-626-0 | UNSRI http://ars.ilkom.unsri.ac.id

Deteksi Serangan *Denial of Service* Menggunakan *Artificial Immune System*

Candra Adi Winanto
Jurusan Sistem Komputer
Fakultas Ilmu Komputer
Universitas Sriwijaya,Inderalaya 30662
Sumatera Selatan Indonesia
09121001042@students.ilkom.unsri.ac.id

Abstrak – Salah satu masalah ayng ada pada bidang komputer security adalah serangan Denial of Service (DoS). Sudah banyak dikembangkan, beberapa metode yang dapat digunakan untuk mendeteksi jenis serangan ini, salah satunya adalah anomaly detection. Pada penelitian ini diterapakan salah satu algoritma Artificial Immune System, yaitu dendritic cell algorithm.Pada penelitian ini menggunakan dataset iscx, dimana serangan DoS dibuat dengan memanfaatkan tools slowloris. Slowloris merupakan salah satu tools yang diguanakan untuk melakukan serangan DoS. Tools slowloris ini, menghabiskan socket yang tersedia pada web server, dan mengirimkan get request yang tidak lengkap.

Kata kunci : Denial of Service, Anomaly Detection, Artificial Immune System, dendritic cell algorithm, slowloris, feature extraction.

I. PENDAHULUAN

Dalam perkembangannya serangan terhadap sistem komputer snagat bervariasi, slah satu serangan yang umum dilakukan oleh penyerang adalah *Denial of Service* (DoS). Pada serangan DoS, penyerang menghalangi akses terhadap *user* pada layanan yang ada pada server, atau memperlambat kinerja sistem itu sendiri. Pada umumnya serangan DoS, dibagi menjadi dua tipe seragan, yaitu : (I). Flooding Attack dan (II) Logic Attack[1].

DoS merupakan serangan yang dapat menyebabkan kerusakan yang serius pada sistem, sehingga diperlukan sebuah sistem yang dapat mendeteksi serangan ini dengan baik, *Intrusion Detection System* (IDS) yang melakukan deteksi menggunakan *signature based* dapat mendeteksi serangan yang telah diketahui dengan efektif, namun untuk jenis serangan yang baru, namun sistem ini tidak mampu mendeteksi serangan lama dengan pola baru[2].

Untuk mengatasi permasalahan tersebut, pada penelitian ini akan diterapkan algoritma *Artificial Immune System* [3], untuk mendeteksi serangan DoS. Algoritma ini akan digunakan untuk mendeteksi serangan DoS pada *Application Layer* dari model OSI, dengan mengklasifikasikan paket data serangan dan paket data normal pada *dataset.*

Pada paper ini disusun dengan format sebagai berikut : pembahasann kajian pustakan pada bagian 2, pada bagian 3 memebahasah metodologi penelitian yang digunakan, bagian 4 membahas mengenai hasil pengamtan sementara, dan kesimpulan serta kegiatan selanjutnya akan dibahas pada bagian 5.

II. KAJIAN PUSTAKA

Pada penelitian[1] membahas permasalahan deteksi serangan DoS yang dapat dikenali dan yang belum dapat dikenali sebelumnya, pada penelitian tersebut menggunakan algoritma *Neighborhood Negative Selection* (NNS) yang merupakan salah satu algoritma *Artificial Immune System.* Dimana uji coba sistem dilakukan secara offline dengan menggunakan dataset dari DARPA tahun 1998. Dari hasil yang didapat, sistem dapat mengidentifikasi dan memilah serangan *Denial of Service* (DoS).

Pada penelitian lain [4] membahas permasalahan deteksi serangan *flooding* TCP-SYN menggunakan algoritma *Dendritic Cell Algorithm* (DCA), dalam penelitian ini sistem disebar menjadi agen-agen monitoring, yang melakukan monitoring pada setiap *host*, dan terdapat pusat pemrosesan data yang memproses hasil dari monitoring pada host. Dari hasil yang didapat, sistem memiliki kemampuan membedakan tingkah laku normal dan abnormal.

Selanjutnya [5] menerapkan algoritma *Dendritic Cell Algorithm* (DCA) untuk medeteksi serangan *port scanning,* secara online menggunakan *tools* Nmap dengan menyertakan juga *traffic* dari penggunaan *Secure Shell* (SSH), Bash, dan *Secure Copy* (SCP). Dari hasil yang diperoleh, sistem mampu mendeteksi serangan *port scanning* dengan

ISBN : 979-587-626-0 | UNSRI http://ars.ilkom.unsri.ac.id

menggunakan *tools* nmap. Dari beberapa ulasan diatas, pendekatan algoritma *Artificial Immune System* (AIS) dapat dimanfaatkan untuk mendeteksi *anomaly* dari serangan *Denial of Service* (DoS).

III. METHODOLOGI PENELITIAN

Pada penelitian ini menggunakan beberapa perangkat lunak dan *dataset iscx* sebagai *dataset* acuan dalam penelitian, serta sebuah PC yang digunakan untuk pengoahan data dengan spesifkasi sebagai berikut :

Tabel 1. Kebutuhan Perangkat Lunak

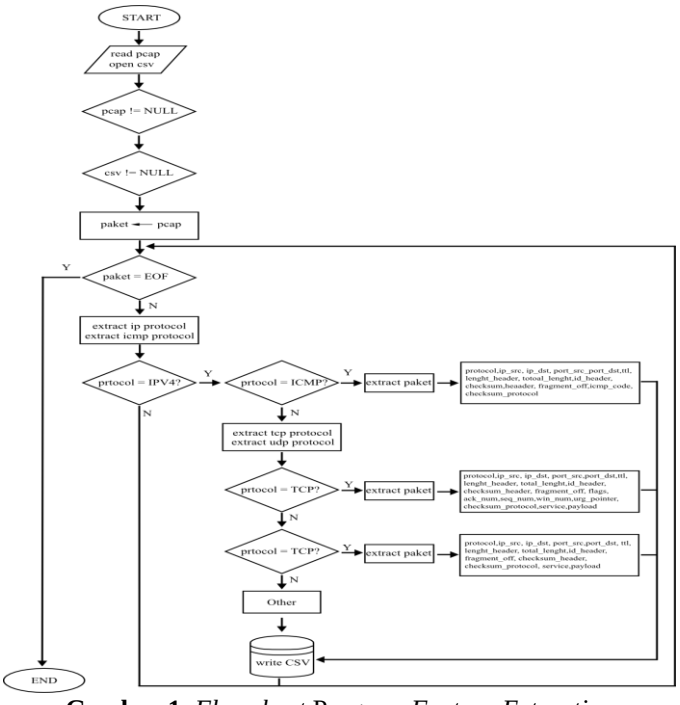
Sistem	*Tools*	Keterangan
IDS	Snort	Versi 2.9.8.2 GRE (Build 335)
Compiler	gcc	Versi Ubuntu 4.8.4-2ubuntu1~14.04.3
Compiler	Python	Versi Python 2.7.6
Text Editor	Geany	Versi 1.23.1
Library	Libpcap-dev	Versi 1.5.3-2

Tabel 2. Kebutuhan Perangakat Keras

Perangkat Keras / SO	Keterangan
CPU	Intel Core i5
RAM	8 GB
HDD	750 GB
Sistem Operasi	Linux

Pengenalan paket serangan merupakan tindakan yang harus dilakukan sebelum menentukan parameter yang akan digunakan pada sistem. Pada percobaan ini digunakan dataset ISCX tahun 2010 dengan nama paket 14 juni 2010 dengan ukuran paket sebesar 7,4 GB. Menurut [6] , paket raw data tersebut dengan aliran data L2L sebanyak 25,444, L2R sebanyak 144,127 dengan tag data normal sebanyak 167,609 dan tag data serangan sebanyak 3,771. Dengan komposisi paket data berupa paket TCP sebanyak 122,298, UDP 48,453 dan ICMP 623. Berikutnya paket dipecah menjadi beberapa peket dengan bantuan aplikasi editcap, dengan banyaknya data per file hasil pemecahan adalah 2 juta baris data yang menghasilkan file keluaran sebanyak 5 buah file, tujuan dari pemecahan paket ini adalah agar data hasil feature extraction dapat termuat pada file dengan format csv, yang manan format file ini hanya mampu menampung file dengan ukuran tidak lebih dari 2 GB. Selanjutnya, melakukan konfigurasi snort IDS untuk memvalidasi keberadaan serangan yang menjadi objek penelitian. Aplikasi snort dibangun dari *source code*, dikarenakan snort yang dipasang dengan memanfaatkan *repository* pada sistem tidak dapat membaca paket raw data lebih dari 2 GB, dimana target file raw data yang akan dibaca sebesar 7,4 GB. Selanjutnya, pembacaan raw data menggunakan snort untuk mengetahui serangan yang terdeteksi, pada proses pembacaan raw data ini diterapkan dua kali pengujian dimana pada pengujian pertama tidak menggunakan *rules slowloris*, dan pada pengujian kedua menggunakan *rules slowloris* dengan masing-masing percobaan juga disertakan *rules* jenis serangan lain dengan total *rules* yang diaktifkan selain slowloris berjumlah 56 buah *rules*.

Tahap berikutnya adalah melakukan *feature extraction*, dimana paket yang berbentuk raw data akan diubah menjadi file csv agar mudah dalam proses pembacaan, dan penentuan parameter yang akan digunakan. Dalam proses *feature extraction* ini, digunakan aplikasi dengan *flowchart* sebagaia berikut :



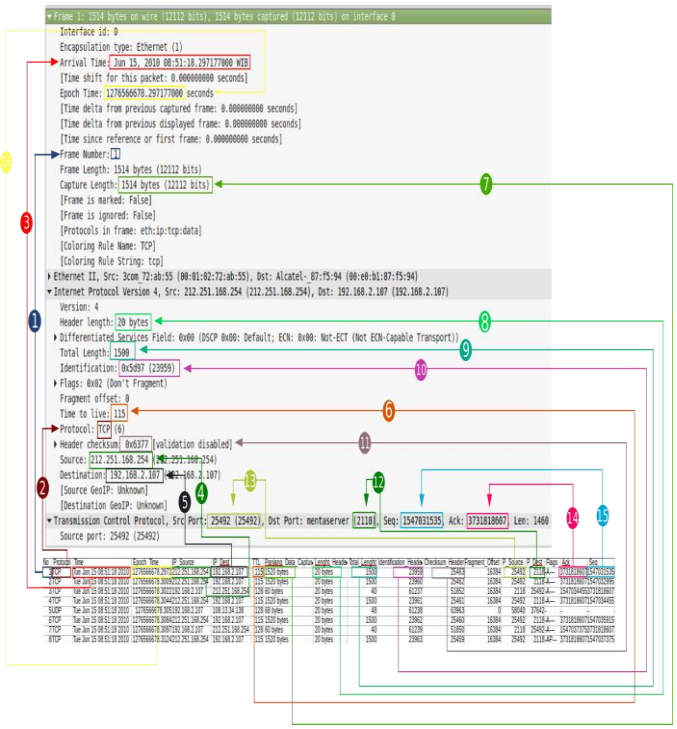
Gambar 1. *Flowchart* Program *Feature Extraction*

Prosiding
ANNUAL RESEARCH SEMINAR 2016
6 Desember 2016, Vol 2 No. 1

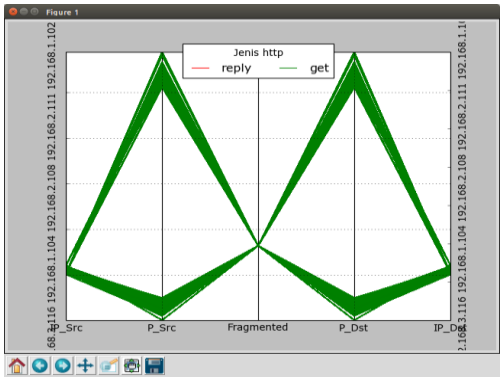
ISBN : 979-587-626-0 | UNSRI http://ars.ilkom.unsri.ac.id

IV. HASIL PENGAMATAN SEMENTARA

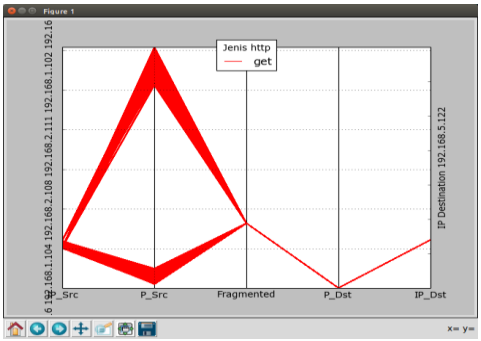
Pada bagian hasil diberikan, visualisasi *trffic http* dari dataset serta korelasi data dari proses *feature extraction* dan data dari *wireshark*. Visualisasi yang dilakukan, menggunakna dataset iscx dengan hanya mengambil *traffic* http dari ip lingkungan *testbed*. Berikut hasil korelasi data hasil *feature extraction* dengan aplikasi *wireshark* dan hasil visualisasi data *http flow* yang terdapat pada *dataset* :



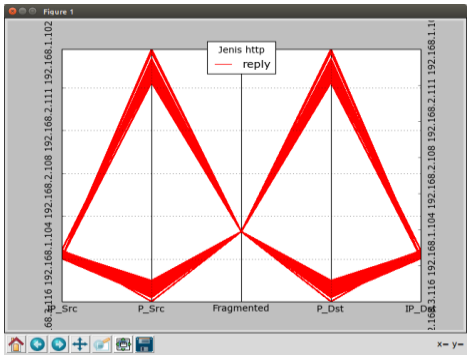
Gambar 2. Hasil Korelasi data



Gambar 3. Visualisasi *Traffic http*



Gambar 4. Visualisasi *Traffic http get*



Gambar 5. Visualisasi *Traffic http response*

Pada gambar 2 terdapat 16 titik persamaan dari file pcap yang dibuka dengan wireshark dan file csv hasil dari *feature extraction*. Keenam belas titik tersebut dapat dijelaskan sebagai berikut :

A. Pada bagian nomor 1,3,7 dan 16 mendeskripsikan ekstraksi bagian frame pada wireshark. Pada bagian frame ini mendeskripsikan pada nomor satu merupakan nomor paket / frame, nomor tiga merupakan detail waktu bulan, tanggal, tahun serta waktu dalam jam dimana paket ter-capture, nomor tujuh mendeskripsikan paket yang ter-capture saat dilakukan sniffing, serta pada nomor 16 mendeskripsikan epoch time, epoch time merupakan unix timestamp dimana katepatan waktu hingga milisecond.

B. Pada bagian nomor 2,4,5,6,8,9,10, dan 11 mendeskripsikan ekstraksi pada bagian Internet Protocol. Pada bagian nomor dua mendeskripsikan jenis protkol yang digunakan oleh paket, nomor empat dan nomor lima mendeskripsikan source ip dan destination ip yang menunjukkan asal dan tujuan paket tersebut, nomor enam mendeskripskan nilaui

ISBN : 979-587-626-0 | UNSRI http://ars.ilkom.unsri.ac.id

TTL (Time to Live) pada pekat tersebut, nomor delapan menunjukkan panjang header pada paket tersebut, nomor sembilan menunjukkan pecket length dari peket, nomor sepuluh menunjukkan nomor identification berupa bilangan hexadecimal serta bilangan decimalnya, serta pada nomor 10 menunjukkan header checksum berupa bilangan hexadecimal.

C. Pada bagian nomor 12,13,14, dan 15 mendeskripsikan ekstraksi pada bagian protokol TCP. Pada bagian nomor dua belas dan tigabelas menunjukkan port asal dan tujuan dari paket, nomor empat belas menunjukkan Ack number, serta nomor 15 menunjukkan Sequence number.

Pada gambar 3 menggambarkan *traffic http* pada *dataset* dengan ip pada lingkungan testbed, sedangkan *traffic http* yang mengarah keluar dari *testbed* tidak dimasukkan pada visualisasi. Pada gambar 3 terlihat bahwa *traffic http* pada datset bersumber dari beberpa ip, dengan rentang port yang dibuka untuk melakukan pengaksesan dan response berada pada 2 rantang titik utama, yaitu pada rentang titik dibawah nilai 10000 dan diatas nilai 50000, serta keseluruhan paket memiliki nilai *fragment* yang menngacu pada satu titik yang sama.

Pada gambar 4 memvisualisasikan *http get* yang mana terlihat, pada saat *get request* terdapat beberapa ip yang melakukannya dengan port sumber berada pada nilai dibawah 10000 dan diatas 50000, dengan satu titik *fragment* yang menandakan paket dikirim secara terpisah pisah, dengan satu ip dan port tujuan.

Pada gambar 5 memvisualisasikan *http response* yang dapat dilihat polanya mirip dengan pola gambar 3.

V. KESIMPULAN DAN KEGIATAN SELANJUTNYA

Dari data visualisasi yang dilakukan, dihasilkan :

b) *Traffic* http pada lingkungan testbed dengan ip kelas c, memiliki pola satu alamat ip memiliki banyak port terbuka yang melakukan koneksi pada server http.
c) Seluruh data *traffic* memiliki nilai *fragment* yang menandakan bahwa paket dikirim per paket.
d) Port yang digunakan pada *traffic* berkisar berkumpul pada port dibawah 10000 dan port diatas 50000.
e) Pada *traffic* http *request get* memiliki satu server tujuan, dimana banyak *host* yang melakukan akses.

Pada kegiatan selanjutnya akan dilakukan :

c) Penelususran pattern DoS yang disebabkan oleh *tools slowloris* pada *dataset ISCX*.
d) Perancangan dan pembangunan sistem dengan menggunakan *Algortima Dendritic Cell.*
e) Penghitungan akurasi serangan dengan menggunakan *Confusion Matrix.*

PENGHARGAAN

Seluruh kegiatan penelitian ini didukung oleh Fakultas Ilmu Komputer Jurusan Sistem Komuter Universitas Sriwijaya dan Laboratorium *Computer Network and Information Security* (COMNETS).

DAFTAR PUSTAKA

[1] D. Wang, L. He, Y. Xue, and Y. Dong, "Exploiting Artificial Immune systems to detect unknown DoS attacks in real-time," *Proc. - 2012 IEEE 2nd Int. Conf. Cloud Comput. Intell. Syst. IEEE CCIS 2012*, vol. 2, pp. 646–650, 2012.

[2] I. Kashyap, "Study and Analysis of Network based Intrusion Detection System," vol. 2, no. 5, 2013.

[3] D. Dasgupta, S. Yu, and F. Nino, "Recent advances in artificial immune systems: Models and applications," *Appl. Soft Comput. J.*, vol. 11, no. 2, pp. 1574–1587, 2011.

[4] N. B. I. Al-Dabagh and I. A. Ali, "Design and implementation of artificial immune system for detecting flooding attacks," *Proc. 2011 Int. Conf. High Perform. Comput. Simulation, HPCS 2011*, pp. 381–390, 2011.

[5] S. Anandita, Y. Rosmansyah, B. Dabarsyah, and J. U. Choi, "Implementation of Dendritic Cell Algorithm as an Anomaly Detection Method for Port Scanning Attack," *Int. Conf. Inf. Technol. Syst. Innov.*, 2015.

[6] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Secur.*, vol. 31, no. 3, pp. 357–374, 2012.