Jurnal Teknologi Informatika dan Terapan Vol. 05, No 02, Juli – Desember 2018 ISSN: 2358-838X
DOI: https://doi.org/10.25047/jtit.v5i2.92

IMPLEMENTASI SENSOR MONITORING PADA JARINGAN WI-FI (HOTSPOT) BERBASIS SNORT

Ahmad Faisol [1], Imam Izzat Muttaqin[2]

[1]*Dosen Teknik Informatika ITN Malang*
Jl. Raya Karanglo Km. 2 Singosari - Malang
[1]mzfais@lecturer.itn.ac.id

[2]*Mahasiswa Teknik Elektro ITN Malang*
Jl. Raya Karanglo Km. 2 Singosari - Malang
[2]modunglanceng@gmail.com

Abstract

The wireless networking application besides giving the simplicity in communication or data exchange, also has a weakness in security system. Every user's tools that connected to the wireless network must be ready towards the appearance of the destruction or attack, because wireless network doesn't have a clear defense track. On this research, monitoring sensor application based on Snort is being suggested as one of the solutions that can be used to help the network arrangement in monitoring the condition of the network and analyze every dangerous package that is in the network. Snort will detect the intruder and analyze the package that cross the network directly and recording into the data storage media. Monitoring sensor is using rule-base system that will detect every package based on the directions that has defined to the direction data collection. The result of the research shown every new data package that entry the sensor, so the change of the event's amount of the monitoring sensor will change automatically based on the admin's arrangements. More data that cross the sensor, can influence the activity from the server that shown by the slow response from the web server.

Keywords— monitoring sensor, network security, snort, wireless networking, wlan.

PENDAHULUAN

Penerapan jaringan nirkabel memang memberikan banyak manfaat terutama kemudahan dalam berkomunikasi atau bertukar data. Akan tetapi, karena tidak memiliki jalur pertahanan yang jelas, model jaringan ini memiliki kerentanan dari segi keamanan. Sehingga setiap komputer yang terhubung dengan jaringan nirkabel harus selalu siap terhadap adanya gangguan atau serangan yang mungkin terjadi.

Ada beberapa metode yang sering digunakan untuk pengamanan suatu jaringan komputer. Salah satu solusi yang dapat diusulkan adalah pemasangan sensor yang dapat melakukan *monitoring* untuk memantau kondisi jaringan dan menganalisis setiap paket berbahaya yang dikirimkan melalui jaringan tersebut.

Berdasarkan permasalahan tersebut, penulis menggunakan *Snort* sebagai sensor untuk me-*monitoring* kondisi jaringan nirkabel, yang berfungsi untuk mendeteksi setiap paket berdasarkan aturan-aturan yang sudah didefinisikan pada kumpulan data aturan. Jika terdeteksi sebuah paket berbahaya, maka sistem akan langsung memberikan peringatan kepada kepada pengatur jaringan tentang kondisi jaringan saat itu.

Ruang lingkup pada penelitian ini adalah untuk menerapkan sensor sistem *rule base* yang dapat memantau kondisi keamanan

jaringan nirkabel dengan menggunakan *Snort*.

TINJAUAN PUSTAKA

A. Jaringan Nirkabel (WLAN)

Teknologi *Wireless* berarti sebuah teknologi yang tidak menggunakan kabel (nirkabel) untuk melakukan pertukaran data. Sehingga, jaringan nirkabel atau *Wireless* LAN (WLAN) dapat diartikan sebagai sebuah alat yang berfungsi untuk menghubungkan ke jaringan internet tanpa menggunakan kabel [1].

WLAN menggunakan dua macam teknik modulasi, yaitu *Orthogonal Frequency Division Multiplexing* (OFDM) dan *Direct Sequence Spread Spectrum* (DSSS). OFDM akan menyebabkan kecepatan pengiriman data lebih tinggi dibandingkan dengan DSSS, tetapi DSSS lebih sederhana daripada OFDM sehingga akan lebih murah dalam implementasinya. Standar yang lazim digunakan untuk WLAN adalah 802.11 yang ditetapkan oleh IEEE pada akhir tahun 1990. Standar 802.11 kemudian terbagi lagi menjadi beberapa jenis, yakni 802.11, 802.11b, dan 802.11g yang dibedakan oleh frekuensi dan kecepatannya.

B. Intrusion Detection System (IDS)

IDS adalah sebuah aplikasi perangkat lunak atau perangkat keras yang bekerja secara otomatis untuk memonitor kejadian pada jaringan komputer dan menganalisis masalah keamanan jaringan [2]. Terdapat 2 jenis IDS, yaitu :

1. Network – based IDS (NIDS)

NIDS akan melakukan pemantauan terhadap seluruh bagian pada jaringan dengan mengumpulkan paket – paket data yang terdapat pada jaringan tersebut serta melakukan analisa dan menentukan apakah paket – paket tersebut merupakan paket normal atau paket serangan.

2. Host – based IDS (HIDS)

HIDS hanya melakukan pemantauan pada perangkat komputer tertentu dalam jaringan. HIDS biasanya akan memantau kejadian seperti kesalahan login berkali – kali dan melakukan pengecekan pada file.

Hal yang perlu diperhatikan pada implementasi IDS adalah perihal *false positive* dan *false negative*. *False positive* adalah peringatan serangan yang dihasilkan oleh IDS akan sebuah paket normal pada sistem yang dimonitor. *False negative* adalah sebuah serangan yang benar - benar terjadi namun terlewatkan oleh IDS sehingga IDS tidak akan menghasilkan peringatan apapun atas serangan tersebut. IDS dapat melewatkan serangan karena serangan tersebut tidak dikenali oleh IDS atau karena penyerang berhasil menggunakan sebuah metode serangan yang dapat menghindari IDS.

C. Snort

Snort merupakan suatu perangkat lunak untuk mendeteksi penyusup dan mampu menganalisa paket yang melintasi jaringan secara langsung dan melakukan pencatatan ke dalam penyimpanan data serta mampu mendeteksi berbagai serangan yang berasal dari luar jaringan [3]. *Snort* dikembangkan oleh Marty Roesch dan tersedia gratis di www.snort.org. *Snort* bisa digunakan pada sistem operasi linux, Windows, BSD, solaris dan sistem operasi lainnya.

Snort memanfaatkan perangkat tcpdump untuk mengambil dan menganalisis paket data terhadap sekumpulan jenis serangan yang sudah terdefinisi. *Snort* dapat berjalan dalam tiga mode antara lain :

- Paket *sniffer*, melihat paket yang lewat di jaringan.
- Paket *logger*, mencatat semua paket yang lewat di jaringan untuk di analisis.
- NIDS, mendeteksi serangan yang dilakukan melalui jaringan komputer dengan konfigurasi dari berbagai aturan

DOI: https://doi.org/10.25047/jtit.v5i2.92

yang akan membedakan sebuah paket normal dengan paket serangan.

D. Komponen *Snort*

Snort terdiri dari komponen – komponen yang mempunyai tugas dan fungsinya sendiri – sendiri yaitu [4]:

1. *Packet Capture Library*

Packet capture library adalah sebuah perangkat lunak yang terpisah yang mengambil paket data dari NIC. Paket – paket itu adalah paket data Lapisan Data Link (OSI model) yang biasanya disebut frame yang masih belum diproses. Pada sistem Linux dan UNIX, *Snort* menggunakan libpcap, sedangkan pada sistem Windows, *Snort* menggunakan winpcap.

2. *Packet decoder*

Packet decoder mengambil frame lapisan 2 (Data Link) yang dikirimkan oleh packet capture library dan kemudian memecahnya. Pertama – tama komponen ini membaca kode sandi terhadap frame lapisan 2, kemudian paket lapisan 3 (*protocol* IP), lalu kemudian paket lapisan 4 (paket TCP atau UDP). Setelah proses selesai dilakukan, *snort* mempunyai semua informasi masing – masing protokol untuk pemrosesan lebih lanjut.

3. *Preprocessor*

Preprocessor pada *Snort* memiliki beberapa fitur tambahan yang dapat dimatikan atau dinyalakan. Preprocessor bekerja pada paket yang sudah dibaca kode sandinya dan kemudian melakukan transformasi pada data itu supaya lebih mudah untuk diproses oleh *Snort*.

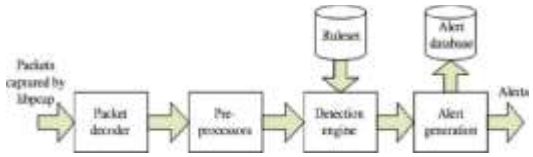
4. *Detection Engine*

Komponen ini mengambil informasi dari packet decoder dan preprocessor yang kemudian memproses data itu pada lapisan *Transport* dan *Application*, membandingkan data yang terkandung dalam paket dengan aturan – aturan yang juga merupakan fitur tambahan dari komponen ini.

5. Output

Ketika *preprocessor* terpancing karena adanya data yang cocok dengan definisi jenis jaringan, *Snort* kemudian menghasilkan peringatan dan kemudian melakukan pencatatan. *Snort* mendukung beberapa macam keluaran, seperti keluaran dalam format teks atau biner. Pencatatan juga bisa dilakukan ke dalam penyimpanan data ataupun syslog.

Cara kerja sistem dari *Snort* untuk mengenali serangan ditunjukkan pada Gambar 1.



Gambar 1. Cara kerja sistem *Snort*

METODE PENELITIAN

Alur pengembangan sistem pada penelitian ini melalui beberapa tahap, antara lain:

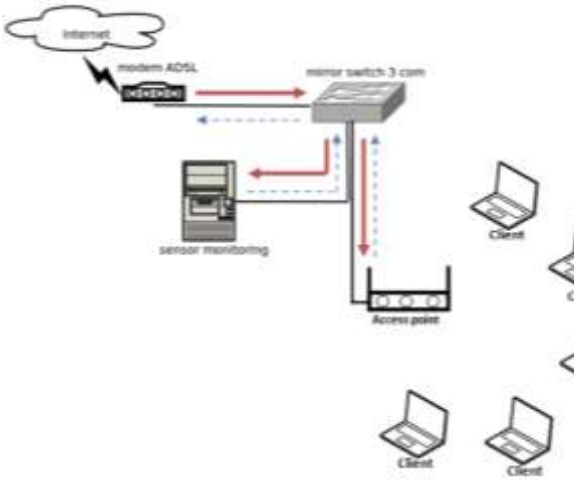
1. Studi pendahuluan
2. Identifikasi dan perumusan masalah
3. Studi pustaka
4. Pengumpulan data (wawancara, survei lapangan, dan studi pustaka)
5. Pengolahan data (analisis kebutuhan sistem)
6. Analisis dan perancangan sistem
7. Implementasi sistem
8. Uji coba sistem
9. Kesimpulan dan saran

Pada tahap perancangan sistem, terdapat beberapa tahapan yang dilakukan, yaitu:

1. Desain sistem monitoring yang berkualitas dan handal
2. Pemilihan *Software*, dalam hal ini adalah pemilihan sistem operasi menggunakan *Ubuntu Server 10.04 LTS*.

3. Pemilihan sensor menggunakan *Snort* yang akan me-*monitoring* jaringan nirkabel.

Sedangkan desain sistem yang digunakan pada penelitian ini ditunjukkan pada Gambar 2.



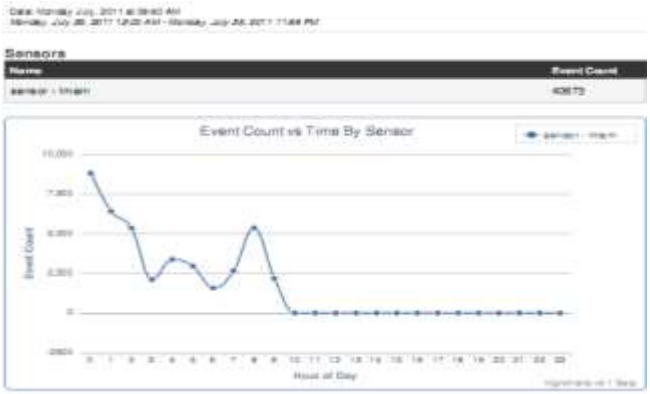
Keterangan : aliran data masuk
aliran data keluar

Gambar 2. Desain Sistem Sensor Monitoring

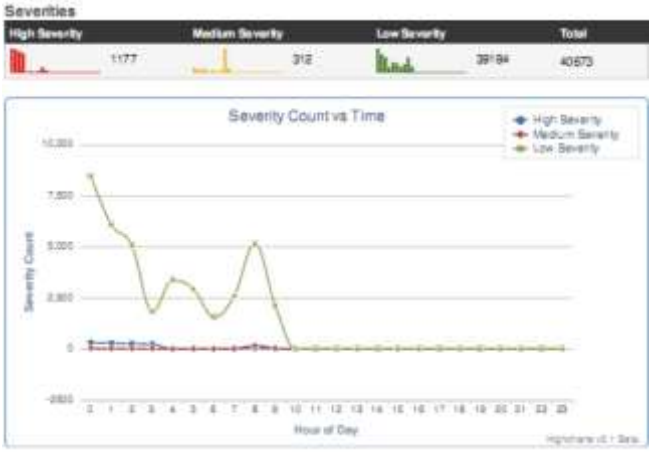
Dari Gambar 2 tampak bahwa aliran data yang masuk dari *router* dikirim melalui *mirror Switch* kemudian paket data tersebut di *copy* oleh server sensor *monitoring* dan diteruskan ke *Access Point* dan disebar pada masing – masing *client*. Fungsi dari server sensor *monitoring* ini hanya meng-*copy* paket data yang lewat tanpa menghalangi aliran paket data yang melintas melalui *mirror switch* maupun *Access Point*.

HASIL DAN PEMBAHASAN

Kinerja Sensor akan dipantau terus oleh pengelola atau admin dan hasilnya dari sistem sensor monitoring akan dilaporkan ke server dengan di-*generate* berupa *PDF*. Tujuan dari *report* atau pelaporan ini adalah sebagai peringatan dari sistem sensor terhadap server. *Report* akan dikirim ke *email administrator* berupa grafik jumlah total *event* dan *real time* yang dikategorikan ke dalam *severity* paket normal ataupun paket – paket yang mencurigakan.



Gambar 3. Grafik Sensor *Event Count*



Gambar 4. Grafik *Severity Event Count*

Gambar 3 dan 4 menunjukkan grafik *event count* dan *severity count* per hari berdasarkan tiga kategori angka dan tampilan skala grafik (*high severity*, *medium severity*, *dan low severity*). Pengujian dilakukan mulai pukul 12 hingga pukul 10 dengan total mencapai 40673 *severity count* dan sejauh itu tingkat *security event* didominasi oleh paket data normal.

Tahap berikutnya adalah menampilkan laporan jumlah aktifitas berdasarkan prosentase nama *event* teratas dan *event count* yang terdaftar dalam *rule base* aplikasi *snort*, seperti ditunjukkan pada Gambar 4 yang berupa http_inspect: LONG HEADER dengan persentasi 33.18% dan jumlah event sebanyak 13424 paket data.

DOI: https://doi.org/10.25047/jtit.v5i2.92

Top 15 Signatures

Signature Name	Percentage	Eve
http_inspect: LONG HEADER	33.18%	134
stream5: Reset outside window	29.03%	117
http_inspect: NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP R...	13.27%	536
stream5: Limit on number of overlapping TCP packets reached	10.02%	405
stream5: Bad segment, overlap adjusted size less than/equal 0	4.41%	178
ET P2P BitTorrent DHT nodes reply	2.56%	103
stream5: TCP Small Segment Threshold Exceeded	2.28%	921
stream5: FIN number is greater than prior FIN	1.82%	737
ET SCAN Behavioral Unusual Port 445 traffic, Potential Scan or...	1.32%	535
stream5: TCP Timestamp is missing	0.9%	366
ET RBN Known Russian Business Network IP TCP (296)	0.26%	104
stream5: TCP window closed before receiving data	0.2%	80
ET CURRENT_EVENTS HTTP contacting a suspicious *.co.cc domain	0.15%	62
ET WEB_SERVER Exploit Suspected PHP Injection Attack (cmd=)	0.15%	62
http_inspect: NON-RFC DEFINED CHAR	0.15%	61
http_inspect: OVERSIZE REQUEST-URI DIRECTORY	0.1%	39
smtp: Attempted data header buffer overflow	0.06%	25
ET P2P BitTorrent DHT ping request	0.06%	24
stream5: Data sent on stream not accepting data	0.04%	18
stream5: TCP Timestamp is outside of PAWS window	0.04%	15

Gambar 5. *Report Signature Name*

Pada Gambar 6 dan 7 menunjukkan laporan jumlah 10 *event* teratas berdasarkan alamat IP asal terhadap tujuan. Sebagai contoh *client* meminta layanan internet terhadap *host name* berupa IP tujuan 192.168.40.5 dengan perbandingan layanan sebanyak 25.05% dan total *client* melakukan transaksi layanan internet sebanyak 8635 paket data.

Top 10 Source Addresses

Source IP Address	Percentage	Even
192.168.40.5	78.57%	2708
192.168.40.7	7.97%	2747
174.122.138.170	1.89%	653
174.121.62.122	1.67%	577
209.90.137.220	1.39%	479
184.73.216.187	1.16%	399
184.73.180.184	1.13%	391
184.73.213.48	1.02%	350
173.194.49.81	0.79%	272
173.194.49.74	0.73%	251

Gambar 6. *Report Source IP Address*

Top 10 Destination Addresses

Destination IP Address	Percentage	Event Count
192.168.40.5	25.05%	8635
192.168.40.7	6.41%	2209
118.96.36.22	5.53%	1907
118.96.36.21	2.63%	906
184.72.180.184	2.05%	705
184.73.216.187	1.94%	667
184.73.213.48	1.89%	652
118.96.36.156	1.86%	641
118.96.36.32	1.76%	606
74.125.66.147	1.65%	569

Gambar 7. *Report Destination IP Address*

KESIMPULAN

1. Setiap aliran paket data dari *router* menuju *mirror switch* dan *access point*, maka sistem sensor *monitoring* akan bekerja pada server yaitu dengan meng-*copy* atau menggandakan paket data yang melalui swith 3 com.

2. Setiap paket data baru yang masuk pada sensor, maka perubahan jumlah *event* pada sensor *monitoring* akan berubah secara otomatis dan akan dilaporkan kepada administrator.

3. Semakin banyak paket data yang melewati sensor, akan berpengaruh terhadap kinerja server yang ditunjukkan dengan *loading web server* yang lama.

DAFTAR PUSTAKA

Wahidin. 2008. *Jaringan wireless untuk orang awam*. Palembang: Maxikom.

Beale, Jay. 2003. *Snort 2.0 Intrusion Detection*. Masachusset: Syngress Publishing, Inc.

Rafiudin, Rahmat. 2010. *Mengganyang Hacker dengan SNORT*. Surabaya: ANDI OFFSET.

Snort Teams. Desember 7, 2011. *Snort User Manual 2.9.2*. Columbia: Sourcefire, Inc.

(2001) Jogja Linux Website. [Online]. Tersedia: http://jogjalinux.or.id/berita/arsip/2010/01/14/kustumisasi-konfigurasi-IDS-snort.

(2011) Blog Snort. [Online]. Tersedia: http://blog.snort.org/2011/02/ubuntu-1004-install-guide-for-snort.html.