Analisis Penggunaan Hasil Deteksi IDS Snort pada *Tools* RITA dalam Mendeteksi Aktivitas *Beacon*

We Muftihaturrahmah Tenri Sau[1)], Sepha Siswantyo[2)]
(1) Politeknik Siber dan Sandi Negara, we.muftihaturrahmah@bssn.go.id
(2) Politeknik Siber dan Sandi Negara, sepha.siswantyo@poltekssn.ac.id

Abstrak

Meningkatnya berbagai macam ancaman dan serangan, mengharuskan sistem keamanan informasi juga lebih ditingkatkan. Intrusion Detection System (IDS) sebagai salah satu sistem untuk melakukan deteksi dan pencegahan, juga harus ditingkatkan kemampuannya dalam mengamankan jaringan. Saat ini, jenis IDS yang berbasis signature masih memiliki kekurangan, yaitu tidak mampu mendeteksi ancaman atau serangan yang belum diketahui, seperti serangan aktivitas beacon yang biasanya dilakukan oleh malware berjenis ransomware atau trojan. Oleh karena itu, diperlukan pendetekatan atau tools lain untuk melengkapi kekurangan dari IDS jenis ini. Real Intelligence Threat Analytics (RITA) adalah tools berbasis anomali yang melakukan deteksi aktivitas beacon melalui analisis statistik dan algoritme K-means clustering di dalam sebuah lalu lintas jaringan. Pada penelitian ini akan dilakukan analisis terhadap penggunaan IDS Snort pada tools RITA dalam mendeteksi aktivitas beacon dengan menggunakan metode eksperimen yang diperinci dalam tujuh tahap penelitian. Pengujian terhadap deteksi aktivitas beacon dilakukan terhadap 3 buah PCAP dan skenario aktivitas beacon (live beaconing) selama 1 jam. Setelah melakukan pengujian dilakukan analisis terhadap hasil pengujian. Hasil dari penelitian menunjukkan bahwa IDS Snort dapat digunakan sebagai data input RITA dimana terlebih dahulu format log IDS Snort harus diubah menjadi format log IDS Bro/Zeek yang berbentuk TSV/JSON. Sehingga tools RITA dapat dijadikan sebagai solusi alternatif untuk mendeteksi aktivitas beacon pada IDS Snort.

Kata kunci: *beacon*, IDS, intrusi, RITA, Snort.

1. PENDAHULUAN

Intrusion Detection System (IDS) terbagi menjadi IDS berbasis *signature* dan IDS berbasis anomali dalam melakukan analisis. Snort adalah salah satu IDS *open source* berbasis *signature* yang paling banyak digunakan [1]. IDS yang berbasis *signature*, akan melakukan deteksi dengan membandingkan *signature* dengan paket yang ditangkap dari *traffic* untuk mengenali adanya kemungkinan intrusi. Kelemahan dari jenis ini adalah jika terdapat serangan atau ancaman yang belum didefinisikan di dalam basis data *signature*, maka hal tersebut tidak akan terdeteksi sebagai intrusi [2].

Salah satu intrusi yang sulit untuk dideteksi oleh IDS berbasis *signature* adalah intrusi yang melakukan aktivitas *beacon*. Intrusi ini sulit dideteksi karena dapat dianggap sebagai aktivitas yang normal, namun dibalik hal tersebut, intrusi ini melakukan aktivitas untuk menghubungi penyerang dalam meminta arahan mengenai hal yang akan dilakukan selanjutnya. Aktivitas tersebut yang dikatakan sebagai aktivitas *beacon* [3]. Aktivitas *beacon* biasanya dilakukan oleh *malware* berjenis *trojan* dan *ransomware* seperti *Trojan Remote Access* dan *WannaCry*, serangan *Denial of Service* (DoS), maupun serangan yang melakukan pencurian data [4].

Real Intelligence Threat Analytics (RITA) adalah sebuah *tools* yang diciptakan oleh Active Countermeasures yang memiliki fungsi untuk mendeteksi adanya aktivitas *beacon*. Latar belakang diciptakannya RITA adalah untuk mendeteksi adanya aktivitas *beacon* yang tidak terdeteksi oleh IDS, dimana pendeteksian dilakukan menggunakan analisis statistik dan algoritme *K-means clustering* terhadap komunikasi yang terjadi dalam sebuah jaringan [4].

Secara standar, RITA menggunakan *log* dari IDS Bro/Zeek, tetapi RITA masih bisa digunakan secara terpisah tanpa harus menggunakan IDS Bro/Zeek [5]. Oleh karena itu, pada penelitian ini dilakukan analisis terhadap penggunaan IDS Snort pada *tools* RITA dalam mendeteksi aktivitas *beacon*. Hal ini bertujuan untuk mengetahui cara melanjutkan hasil deteksi dari IDS Snort ke dalam RITA untuk mendeteksi adanya aktivitas *beacon*, dimana pengujian terhadap deteksi aktivitas *beacon* dilakukan menggunakan 3 buah PCAP dan skenario aktivitas *beacon* (*live beaconing*) menggunakan *tools Build Your Own Botnet* selama satu jam.

2. LANDASAN TEORI

2.1. Aktivitas *Beacon*

Aktivitas *beacon* (*beaconing*) adalah perilaku untuk mengirimkan sinyal secara periodik dari mesin korban terinfeksi ke sebuah *server command & control* (C2) milik penyerang [5]. Hal ini bertujuan untuk menghubungi penyerang untuk meminta arahan mengenai hal yang akan dilakukan selanjutnya. Selain menunjukkan waktu dan aktivitas korban, aktivitas *beacon* juga dapat menjalankan perintah baru pada mesin korban yang terinfeksi tanpa sepengetahuan korban.

Beberapa jenis *malware* dan serangan yang biasanya melakukan aktivitas *beacon* adalah sebagai

berikut. *Trojan* berjenis *trojan remote access*, *malware* berjenis *ransomware*, serangan *Denial of Service* (DoS), dan serangan pencurian data.

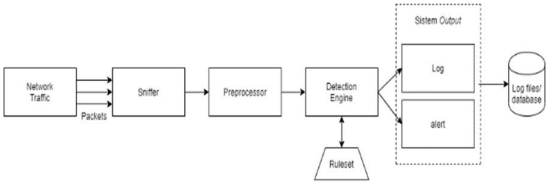
2.2. *Intrusion Detection System*

Intrusion Detection System (IDS) atau sistem pendeteksi intrusi adalah sebuah sistem untuk mendeteksi adanya tindakan atau akses tidak sah di dalam suatu jaringan [1]. IDS akan mengumpulkan data-data dari sensor jaringan yang kemudian akan memberikan peringatan jika terjadi intrusi pada jaringan tersebut.

Berdasarkan cara menganalisis sebuah intrusi, IDS terbagi menjadi analisis berbasis *signature* dan berbasis anomali. IDS berbasis *signature* adalah IDS yang paling banyak digunakan saat ini, yaitu IDS yang melakukan pendeteksian intrusi berdasarkan pola atau *string* yang sesuai dengan ancaman ataupun serangan yang diketahui. Sedangkan IDS berbasis anomali adalah IDS yang membandingkan perilaku kondisi normal dan abnormal dari suatu jaringan untuk menentukan adanya intrusi pada suatu sistem [6]. Salah satu contoh IDS berbasis *signature* adalah Snort dan Suricata, sedangkan IDS berbasis anomali adalah Zeek.

2.3. Snort

Snort adalah sebuah sistem pendeteksi intrusi *open source* yang diciptakan oleh Roesch dan pertama kali muncul pada 22 Desember 1998. IDS ini dapat menganalisis lalu lintas jaringan dan paket *logging* secara *real time* dengan 3 mode utama, yaitu sebagai *packet sniffer*, *packet logger*, dan IDS jaringan [1]. Arsitektur dari IDS Snort digambarkan pada Gambar 1.



Gambar 1. Arsitektur Snort

Untuk dapat menjalankan fungsinya, Snort memiliki 4 komponen utama pada arsitekturnya, yaitu [7]:

1. *Sniffer*

Sniffer adalah perangkat yang berfungsi untuk masuk ke dalam jaringan. Tujuan utama dari komponen ini adalah untuk melakukan *sniffing*, atau menangkap paket-paket yang sedang melintasi jaringan.

2. *Preprocessor*

Tugas utama dari komponen ini adalah untuk melakukan penyaringan atau pengecekan terhadap paket-paket yang telah ditangkap. Komponen ini akan menentukan jenis dari paket yang ditangkap, apakah berupa paket HTTP, paket hasil *scanning port* atau paket jenis lainnya.

3. *Detection Engine* (Mesin pendeteksi)

Pada tahap ini, mesin pendeteksi akan melakukan pencocokan antara paket dengan *rules* yang telah didefinisikan. Jika paket cocok dengan *rules* yang dibuat, maka paket tersebut akan diteruskan ke sistem *output* untuk menghasilkan *alert.*

4. Sistem *output*

Jika paket sesuai dengan *rules*, maka sistem *output* akan mengeluarkan peringatan atau *alert* dari IDS. Selain menghasilkan *alert*, Snort juga akan menghasilkan *log* berbentuk teks yang secara tetap tersimpan dalam *file log* pada direktori /var/log/snort. Format *log* dari IDS Snort dapat disimpan dalam berbagai format, yaitu tcpdump, csv, dan Unified2 [8].

2.4. *Basic Analysis and Security Engine*

Basic Analyisis and Security Engine (BASE) adalah *tools* berbasis *web interface* yang digunakan untuk menampilkan hasil analisis intrusi (*alert*) yang dideteksi oleh Snort [1]. *Tools* ini berfungsi sebagai visualisasi dari hasil deteksi Snort untuk mempermudah pengguna dalam menganalisis hasil deteksi. BASE diciptakan oleh sekelompok pengembang dan penerjemah yang didasarkan pada kode dari proyek ACID (*Analysis Console for Intrusion Databases*) [1]. BASE dilisensikan di bawah GNU *General Public License* (GPL) yang dapat berjalan pada varian Unix maupun Windows [9].

2.5. *Real Intelligence Threat Analytics* (RITA)

RITA adalah sebuah *tools* yang diciptakan oleh John Strand untuk menganalisis lalu lintas jaringan dengan menggunakan *log* dari IDS Bro/Zeek [7] [10]. RITA memiliki 3 fungsi utama, yaitu untuk mendeteksi aktivitas *beacon*, mendeteksi *Domain Name System* (DNS) *tunneling*, dan memeriksa *blacklist* untuk mencari *domain* dan *host* yang mencurigakan [11]. Gambar 2 di bawah ini merupakan alur kerja dari RITA.

Gambar 2. Alur Kerja RITA

RITA akan mengambil data dari IDS sebagai input yang digunakan dalam melakukan analisis. Data tersebut berasal dari *log* IDS ataupun dari PCAP. RITA hanya dapat memproses *file log* dengan format IDS Bro/Zeek yang berbentuk TSV atau JSON [5]. Setelah memasukkan data menjadi *dataset*, data tersebut dapat dianalisis oleh RITA dan hasil akhirnya berupa *output* yang ditampilkan pada terminal ataupun dalam bentuk *web interface.*

RITA menggunakan modul *Discrete Fast Fourier Transform* (DFFT) sebagai modul untuk menganalisis *log* koneksi [4]. Modul ini yang dapat mengidentifikasi seberapa sering komunikasi ke *server* C2 terjadi. Data tersebut yang selanjutnya akan

menjadi salah satu paramater dalam menentukan adanya aktivitas *beacon.*

RITA mendefiniskan beberapa parameter yang digunakan untuk menganalisis aktivitas *beacon*, parameter ini ditampilkan pada tabel hasil analisis RITA. Tabel 1 menunjukkan parameter serta ketentuan yang digunakan RITA dalam melakukan analisis aktivitas *beacon.*

Tabel 1 Parameter RITA

No.	Parameter	Ketentuan
1.	*Connections*	20
2.	*Size Mode*	Semakin kecil semakin baik
3.	*Interval Skew*	Semakin mendekati 0 semakin baik
4.	*Size Skew*	Semakin mendekati 0 semakin baik
5.	*Interval Dispersion*	Semakin mendekati 0 semakin baik
6.	*Size Dispersion*	Semakin mendekati 0 semakin baik
7.	*Score*	Semakin mendekati 1 semakin baik

2.6. *Build Your Own Botnet*

Build Your Own Botnet (BYOB) adalah *tools open source* yang memiliki fungsi untuk membangun dan mengoperasikan botnet dasar [12]. *Tools* ini didesain bagi peneliti dan pengembang untuk memperdalam dan meningkatkan pemahaman mengenai keamanan jaringan, dimana pengguna dapat langsung menggunakan *tools* tanpa harus membuat *Remote Access Trojan* (RAT) atau *server* C2 dari awal.

Fitur utama dari *tools* ini adalah membentuk akses ke mesin target secara jarak jauh tanpa sepengetahuan pemilik atau yang disebut juga dengan *Remote Access Trojan* (RAT) [13]. Melalui akses tersebut, penyerang dapat melakukan berbagai hal terhadap mesin target.

2.7. Penelitian Terkait

Aydin et al. pada tahun 2009 dalam penelitiannya menambahkan pendekatan berbasis anomali *Packet Header Anomaly Detector* (PHAD) dan *Network Traffic Anomaly Detector* (NETAD) untuk menambah kemampuan deteksi dari Snort yang berbasis *signature* [2]. Hasil penelitian membuktikan bahwa PHAD dan NETAD dapat menambah kemampuan deteksi dari Snort khususnya dalam mendeteksi serangan yang belum diketahui.

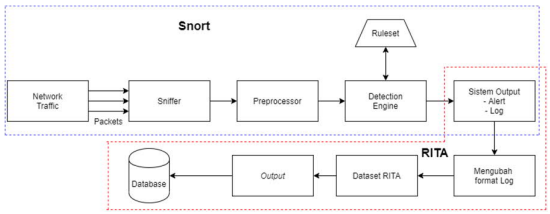
Penelitian yang sama juga dilakukan oleh J. Gomez et al. pada tahun 2009 dengan menciptakan H-Snort, yaitu integrasi antara Snort dan pendekatan berbasis anomali [14]. Hasil penelitian membuktikan bahwa sistem deteksi dari Snort semakin meningkat, dimana *false alarm* juga semakin berkurang.

Pada awal tahun 2019, D. Haselhorst melakukan penelitian dengan menambahkan *tools* RITA ke dalam Security Onion untuk menambah kemampuan deteksi dari Security Onion, khususnya untuk mendeteksi adanya aktivitas menghubungi *server* C2 [4]. Hasil dari penelitian Haselhorst membuktikan bahwa integrasi antara Security Onion, Zeek, dan RITA dapat meningkatkan kemampuan untuk menganalisis jumlah koneksi, durasi koneksi, dan jumlah data, sehingga dapat mendeteksi aktivitas *beacon* dengan efektif.

3. METODE PENELITIAN

Metode penelitian yang digunakan adalah metode eskperimen. Metode ini dipilih karena adanya perlakuan terhadap hasil deteksi dari IDS Snort berupa penambahan *tools* RITA. Pengaruh dari perlakuan tersebut yang selanjutnya akan diteliti untuk melihat kemampuan deteksi, khususnya dalam mendeteksi adanya aktivitas *beacon.*

Eksperimen alur kerja Snort dan *tools* RITA yang dirancang digambarkan pada Gambar 3.



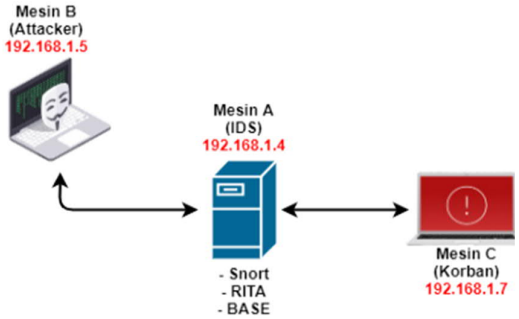
Gambar 3. Alur Kerja Snort dan RITA

Pertemuan antara alur kerja Snort dan RITA berada pada sistem *output* yang dihasilkan berupa *file log* yang dihasilkan oleh Snort. Pertemuan dari alur kerja tersebut dipilih karena *tools* RITA melakukan analisis dengan melihat statistik dari keseluruhan *log*, sehingga *tools* ini hanya dapat digunakan ketika sudah terdapat *log* dari sebuah IDS. Oleh karena itu, *tools* RITA akan digunakan setelah Snort menghasilkan sebuah *file log.*

4. RANCANGAN PENELITIAN

4.1. Lingkungan Penelitian

Eksperimen ini menggunakan 1 unit perangkat keras dengan beberapa mesin virtualisasi VirtualBox. Gambar 4 menggambarkan lingkungan penelitian yang digunakan.



Gambar 4. Skema Jaringan Aktivitas *Beacon*

Spesifikasi dari perangkat keras yang digunakan diberikan pada Tabel 2, spesifikasi mesin virtualisasi diberikan pada Tabel 3, dan spesifkasi perangkat lunak yang digunakan diberikan pada Tabel 4.

Tabel 2. Spesifikasi Perangkat Keras

No.	Informasi	Keterangan
1.	Merk/Tipe	HP Pavilion 13-an0014tu
2.	*Processor*	Intel Core i7-8565U Processor
3.	Memori (RAM)	8 GB DDR4-2400 SDRAM
4.	*Processor* Grafis	Intel UHD *Graphics* 620
5.	*Hard Disk*	512 GB PCIe NVMe M.2 SSD
6.	Sistem Operasi	Windows 10 *Home Single Language* 64

Tabel 3. Spesifikasi Virtualisasi Mesin A

No.	Informasi	Mesin A	Mesin B	Mesin C
1.	Sistem Operasi	Ubuntu 18.04	Kali Linux	Windows 10
2.	Memori (RAM)	4024 MB	3072 MB	1048 MB
3.	*Processor*	2 CPU	2 CPU	1 CPU
4.	Memori Video	16 MB	16 MB	16 MB
5.	*Hard Disk*	100 GB	80 GB	80 GB
6.	Alamat IP	192.168.1.4	192.168.1.5	192.168.1.7

Tabel 4. Spesifikasi Perangkat Lunak

No	Informasi	Versi	Mesin	Keterangan
1.	Snort	2.9.15.1	Mesin A	IDS
2.	RITA	3.2.0	Mesin A	*Tools* tuntuk mendeteksi aktivitas *beacon*
3.	BASE	1.4.5	Mesin A	Virtualisasi Snort
4.	BYOB	1.0	Mesin B	*Tools* untuk melakukan aktivitas *beacon*

4.2. Skenario Aktivitas *Beacon* (*Live Beaconing*)

Pengujian aktivitas *beacon* dilakukan dengan menggunakan tiga buah PCAP dan skenario *live beaconing* selama 1 jam. Berikut adalah langkah-langkah yang dilakukan ketika pengujian.

a) Menjalankan PCAP dan melakukan skenario aktivitas *beacon*
b) IDS Snort mendeteksi intrusi dari PCAP dan skenario aktivitas *beacon*
c) Melanjutkan hasil deteksi Snort ke dalam *tools* RITA
d) Melakukan analisis aktivitas *beacon* menggunakan RITA

Tiga buah PCAP yang digunakan, 2 buah pcap berasal dari Netresec [15] dan 1 buah pcap dari Active Countermeasures [16] dengan spesifikasi masing-masing PCAP diberikan pada Tabel 5.

Tabel 5. Spesifikasi *File* PCAP

No.	Nama PCAP	Informasi		
		Jumlah paket	Ukuran *file*	Nama PCAP
1.	Maccdc2012_00012.pcap	3.190.917	1,04 GB	PCAP A
2.	maccdc2011_00007.pcap	10.000.000	1,79 GB	PCAP B
3.	Sample-1500.pcap	4.156.081	1,66 GB	PCAP C

Skenario aktivitas *beacon* yang dilakukan pada penelitian ini menggunakan *tools* BYOB (*Build Your Own Botnet)* dengan langkah-langkah sebagai berikut:

1. Penyerang membuat *server* botnet
   Tahap ini adalah langkah pertama, dimana penyerang membuat *server* botnet menggunakan port 1446. Tahap ini dilakukan dengan menjalankan perintah : python server.py --port 1446.

2. Penyerang membuat *client* botnet
   Tahap kedua, penyerang akan membuat *file* untuk *client* yang nantinya akan dijalankan di mesin korban. *File* ini yang akan membentuk *channel* CnC antara *client* dan *server*. Perintah yang dijalankan adalah : python client.py --name runini.py --freeze 192.168.1.5 1446. Perintah *freeze* bertujuan untuk membuat *file executable* yang akan dijalankan pada mesin Windows.

3. Mengirimkan *file* botnet ke mesin *client.*
   Pada tahap ini, diskenariokan bahwa penyerang telah memiliki akses ke mesin korban, sehingga penyerang dapat mengirimkan *file* runini.exe ke mesin korban.

4. Korban menjalankan *file* runini.exe
   Setelah *file* berada di mesin korban, diskenariokan korban menjalankan *file* tersebut, sehingga menciptakan *channel* CnC antara penyerang dan korban. Ketika *file* tersebut berhasil dijalankan, maka penyerang dapat melakukan berbagai hal tanpa sepengetahuan korban.

5. Penyerang melakukan berbagai aktivitas
   Setelah terhubung, penyerang dapat melakukan hal apapun sesuai dengan fitur yang disediakan oleh *tools* BYOB. Gambar 5 menunjukkan fitur-fitur yang dapat dilakukan menggunakan *tools* BYOB.

Gambar 5. Fitur *Tools* BYOB

4.3. Proses Penggabungan Snort dan RITA

Untuk melanjutkan proses Snort ke dalam RITA, *output log* dari Snort akan dijadikan sebagai *dataset* RITA. Sebelum dijadikan *dataset*, format *log* harus diubah terlebih dahulu menjadi format *log* IDS Bro/Zeek yang berbentuk TSV atau JSON. Ketika *file log* telah sesuai, maka *file* dapat digunakan sebagai *dataset* yang menjadi input dari RITA. Berikut adalah tahapan untuk melanjutkan proses Snort ke RITA.

a. Mengubah *file log* Snort atau PCAP menjadi format *log* Bro/Zeek

Secara standar, RITA menggunakan format *log* IDS Bro/Zeek sebagai data yang akan dianalisis. Oleh karena itu, jika ingin menggunakan RITA maka *log* dari IDS ataupun *file* PCAP harus diubah terlebih dahulu menjadi format *log* Bro/Zeek. Untuk mengubah *file* PCAP atau *log* IDS menjadi bentuk format *log* Bro/Zeek dapat dilakukan dengan perintah : bro -r nama_*file*log/PCAP local "Log::default_rotation_interval = 1 day". Setelah perintah tersebut dijalankan, maka akan terdapat banyak *file log*. Untuk mempermudah proses import ataupun analisis dari RITA, maka *file-file log* tersebut disatukan di dalam satu *folder*.

b. Melakukan impor *file log* menjadi *dataset* RITA

Mengubah *file log* menjadi *dataset* RITA, secara manual dapat dijalankan dengan perintah : rita import folder_log nama_dataset. Setelah perintah tersebut dijalankan, maka *file log* akan menjadi *dataset* dari RITA.

c. Menampilkan hasil RITA

Menampilkan hasil analisis dari RITA dapat dilakukan melalui perintah yang dijalankan secara langsung di terminal. Hasil analisis dapat dicetak langsung pada terminal sesuai dengan keinginan pengguna ataupun dapat dijadikan dalam bentuk format *web interface* untuk mempermudah dan mempercantik tampilan analisis. Untuk mencetak *output* dalam bentuk *web interface* dapat dilakukan dengan perintah : rita html-report nama_dataset.

5. HASIL DAN PEMBAHASAN

5.1. Pengujian

Pengujian deteksi aktivitas *beacon* dilakukan dengan menggunakan 3 buah *file* PCAP dan *live beaconing* terhadap IDS Snort. Penggunaan 3 buah PCAP akan dijadikan sebagai skenario aktivitas *beacon*, dimana *file* PCAP akan dijalankan menggunakan *tools* tcpreplay sehingga seolah-olah kejadian pada PCAP tersebut sedang berlangsung. Sedangkan *live beaconing* akan dilakukan secara langsung dengan menggunakan 2 buah mesin dimana *tools* BYOB digunakan sebagai *tools* untuk melakukan skenario aktivitas *beacon.*

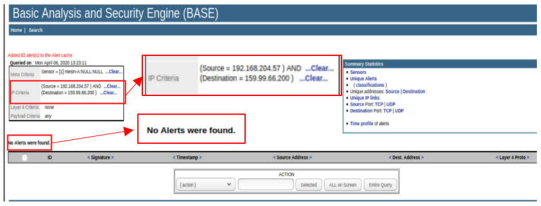
a. PCAP A

Hasil deteksi dari Snort akan ditampilkan melalui visualisasi BASE dan tersimpan sebagai *file log*. Gambar 6 menunjukkan hasil deteksi RITA terhadap PCAP A.

Score	Source	Destination	Connections	Size Mode	Intvl. Skew	Size Skew	Intvl. Dispersion	Size Dispersion
0.835	192.168.204.57	159.99.66.200	37	76	0.000	0.000	0	0
0.676	192.168.202.76	157.55.56.165	48	48	0.000	0.393	1	17
0.675	192.168.202.76	157.55.56.162	42	48	0.000	0.429	1	16
0.674	192.168.202.76	157.56.52.12	56	48	0.000	0.500	1	14
0.673	192.168.202.76	157.55.56.148	62	48	0.000	0.536	1	13

Gambar 6. Hasil Deteksi RITA Terhadap PCAP A

Berdasarkan Gambar 5, dapat dilihat bahwa *score* yang paling tinggi adalah 0.835 dimana RITA mendeteksi aktivitas lalu lintas yang terjadi antara alamat IP 192.168.204.57 dan 159.99.66.200 adalah aktivitas yang dapat dikatakan sebagai aktivitas *beacon.* Setelah mendapatkan hasil deteksi tersebut, Gambar 7 menunjukkan hasil deteksi dari Snort terhadap aktivitas kedua alamat IP tersebut.



Gambar 7. Filter Hasil Deteksi Snort Terhadap PCAP A

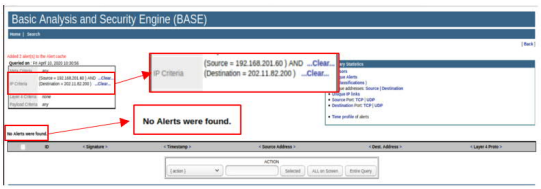
b. PCAP B

Setelah melanjutkan hasil deteksi Snort menjadi data input RITA, Gambar 8 menunjukkan hasil deteksi dari RITA terhadap PCAP B. Berdasarkan nilai tertinggi pada gambar tersebut, dapat ditarik kesimpulan bahwa lalu lintas yang terjadi antara alamat IP 192.168.201.60 dan 202.11.82.200 adalah aktivitas *beacon.*

Score	Source	Destination	Connections	Size Mode	Intvl. Skew	Size Skew	Intvl. Dispersion	Size Dispersion
0.845	192.168.201.60	202.11.82.200	345	76	0.000	0.000	0	0
0.844	192.168.201.59	159.99.66.200	332	76	0.000	0.000	0	0
0.841	192.168.203.67	192.162.24.2	223	44	0.000	0.000	0	0
0.841	192.168.203.67	192.162.24.26	223	44	0.000	0.000	0	0
0.841	192.168.203.67	192.162.204.34	214	44	0.000	0.000	0	0

Gambar 8. Hasil Deteksi RITA Terhadap PCAP B

Setelah mendapatkan alamat IP dari hasil deteksi RITA sesuai Gambar 8, Gambar 9 adalah hasil pencarian menggunakan *filtering* alamat IP pada BASE. Gambar tersebut menunjukkan bahwa Snort tidak dapat mendeteksi lalu lintas yang terjadi antara alamat IP 192.168.201.60 dan 202.11.82.200, sedangkan RITA dapat mendeteksi adanya komunikasi antara kedua alamat IP tersebut.



Gambar 9. Filter Hasil Deteksi Snort Terhadap PCAP B

c. PCAP C

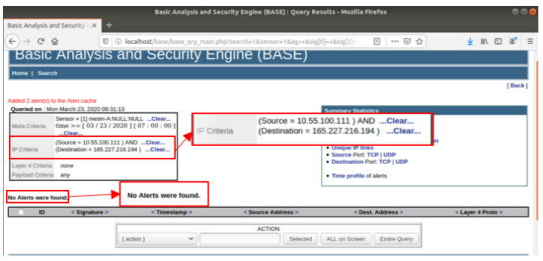
Selain mendeteksi PCAP menggunakan IDS Snort, PCAP juga dianalisis menggunakan *tools* RITA. Gambar 10 menunjukkan hasil deteksi RITA terhadap PCAP C.

Score	Source	Destination	Connections	Size Mode	Intvl. Skew	Size Skew	Intvl. Dispersion	Size Dispersion
1000	10.55.100.111	165.227.216.194	20054	52	0.000	0.000	0	0
1000	192.168.88.2	165.227.88.15	108858	89	0.000	0.000	0	0
0.838	10.55.200.10	205.251.194.64	210	70	0.000	0.000	0	0
0.835	10.55.200.11	205.251.197.77	69	70	0.000	0.000	0	0
0.834	10.55.100.111	34.239.169.214	34	156	0.000	0.000	0	0

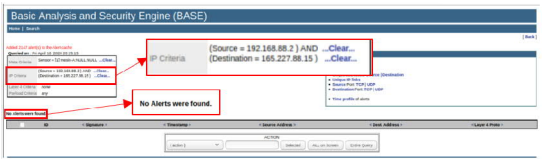
Gambar 10. Hasil Deteksi RITA Terhadap PCAP C

Dari hasil deteksi yang ditunjukkan oleh Gambar 10, dapat dilihat bahwa terdapat 2 lalu lintas jaringan yang menghasilkan nilai sempurna dari hasil perhitungan RITA. Oleh karena itu dapat disimpulkan bahwa pada PCAP C terdapat 2 aktivitas *beacon*, yaitu aktivitas antara alamat IP 10.55.100.111 dan 165.227.216.194, serta aktivitas antara alamat IP 192.168.88.2 dan 165.227.88.15.

Kedua aktivitas tersebut kemudian dicari pada hasil deteksi Snort melalui fitur pencarian menggunakan *filtering* pada visualisasi BASE. Gambar 11 dan Gambar 12 menunjukkan hasil pencarian aktivitas *beacon* terhadap hasil deteksi Snort. Dari kedua gambar tersebut, dapat dilihat bahwa Snort tidak mendeteksi satupun aktivitas antara alamat IP tersebut.



Gambar 11. Filter Hasil Deteksi Snort Terhadap PCAP C (1)



Gambar 12. Filter Hasil Deteksi Snort Terhadap PCAP C (2)

d. *Live Beaconing*

Setelah melakukan skenario aktivitas *beacon* menggunakan BYOB selama satu jam, *log* dari hasil deteksi Snort kemudian dijadikan sebagai data input RITA sesuai dengan langkah-langkah yang telah dijelaskan pada poin 4.3. Hasil analisis dari RITA menunjukkan bahwa terdapat beberapa aktivitas yang dapat dikatakan sebagai aktivitas *beacon*, dimana lalu lintas yang memiliki nilai *score* tertinggi adalah lalu lintas jaringan antara mesin C dan alamat IP 104.111.16.139. Hal tersebut ditunjukkan pada Gambar 13.

Score	Source	Destination	Connections	Size Mode	Intvl. Skew	Size Skew	Intvl. Dispersion	Size Dispersion
0.634	192.168.1.7	104.111.16.139	66	0	0.000	-0.391	0	1281
0.597	192.168.1.7	8.8.4.4	294	0	0.467	-0.641	5	64
0.522	192.168.1.7	118.98.95.99	52	0	0.273	0.617	4	634
0.509	192.168.1.7	118.98.95.91	52	0	0.273	0.693	4	172
0.487	192.168.1.7	23.7.216.221	104	0	0.294	-0.687	12	1121

Gambar 13. Hasil Deteksi RITA Terhadap *Live* Beaconing

Dari hasil analisis RITA yang ditunjukkan pada Gambar 13, selanjutnya dilakukan *filtering* terhadap hasil deteksi Snort untuk melihat apakah aktivitas tersebut dideteksi sebagai *beacon* oleh Snort atau tidak. Gambar 14 menunjukkan bahwa tidak terdapat *alert* terhadap lalu lintas kedua alamat IP tersebut. Sehingga dapat disimpulkan bahwa Snort tidak mendeteksi aktivitas antara alamat IP 192.168.1.7 dan 104.111.16.139 sebagai aktivitas yang berbahaya, sedangkan aktivitas tersebut kemungkinan besar adalah aktivitas dari *malware trojan* yang dijalankan pada mesin C.

Gambar 14. Filter Hasil Deteksi Snort Terhadap *Live* Beaconing

5.2. Pembahasan

Pada bagian ini dijelaskan terkait hasil pengujian terhadap PCAP, dan menjelaskan beberapa hal pada *tools* RITA yang menjadikan *tools* tersebut dapat mendeteksi aktivitas *beacon.*

a. Hasil Deteksi Waktu *Beacon*

Sesuai dengan definisi dari aktivitas *beacon*, yaitu aktivitas dari mesin korban untuk menghubungi *server* C2 secara periodik, pada bagian ini dilakukan analisis PCAP untuk melihat durasi waktu yang dilakukan antara *client* dan *server* C2. Analisis dilakukan dengan melakukan *filtering* alamat IP pengirim dan IP penerima pada *Wireshark*. Alamat IP yang dicari adalah alamat IP yang memiliki nilai *score* tertinggi pada hasil deteksi RITA disetiap pengujian yang dilakukan. Hasil analisis PCAP yang dilakukan terangkum pada Tabel 6.

Tabel 6. Durasi Waktu *Beacon* Setiap Pengujian

No	PCAP	Waktu *Beacon*
1.	A	Setiap 3 menit lalu 2 menit dan berulang
2.	B	Setiap 2 menit lalu 3 menit dan berulang
3.	C	Setiap 14 detik dan 1 menit
4.	*Live Beacnoing*	5 milidetik

b. Format *Log*

Format *log* Snort yang berbentuk tcpdump, csv, atau Unified2 masih berupa data mentah yang belum terstruktur, sehingga harus meneliti dan menganalisis data lebih lanjut. Selain itu diperlukan aplikasi atau *tools* lain untuk mempermudah atau untuk memvisualisasikan pembacaan hasil deteksi.

Pada penelitian ini, format *log* yang digunakan pada Snort adalah tcpdump dan Unified2, sedangkan *tools* RITA hanya dapat memproses *file log* IDS Bro/Zeek yang berbentuk TSV atau JSON. Oleh karena itu, dibutuhkan konversi dari format *log* Snort menjadi format *log* Bro/Zeek.

Format *log Tab-Separated Values* (TSV) dan *JavaScript Object Notation* (JSON) adalah format *log*

terstruktur yang berbentuk kolom dan baris. Format ini memudahkan manusia dan komputer dalam membaca, memproses dan menganalisis data yang diberikan. Selain itu, format ini juga memudahkan komputer untuk mencari data yang unik sesuai parameter yang diinginkan. Hal tersebut yang menjadikan *tools* RITA mampu melakukan analisis untuk menentukan aktivitas *beacon*. Sehingga format *log* dari IDS Snort harus dikonversi menjadi bentuk *log* IDS Bro/Zeek terlebih dahulu untuk menggunakan *tools* RITA yang berbasis anomali.

c. Parameter *Beacon* RITA

RITA hanya menggunakan beberapa parameter dalam melakukan analisis aktivitas *beacon*. Dimana hasil akhir dari analisis yang dilakukan dapat dilihat pada nilai *Score* dari hasil RITA. Gambar 15 menunjukkan bahwa nilai *score* tertinggi dari hasil deteksi RITA terhadap pengujian yang dilakukan, semuanya mendekati atau memenuhi ketentuan dari parameter-parameter yang digunakan untuk menghitung nilai *beacon*.

No.	Parameter	Ketentuan	PCAP A	PCAP B	PCAP C	*Live Beaconing*
1.	*Connections*	20	37	345	20.054	66
2.	*Size Mode*	Semakin kecil semakin baik	76	76	52	0
3.	*Interval Skew*	Semakin mendekati 0 semakin baik	0,00	0,00	0,00	0,00
4.	*Size Skew*	Semakin mendekati 0 semakin baik	0,00	0,00	0,00	-0,391
5.	*Interval Dispersion*	Semakin mendekati 0 semakin baik	0	0	0	0
6.	*Size Dispersion*	Semakin mendekati 0 semakin baik	0	0	0	1281
7.	*Score*	Semakin mendekati 1 semakin baik	0,835	0,845	1	0,634

Gambar 15. Parameter RITA dan Hasil Deteksi Pengujian

Berdasarkan hasil pengujian terhadap PCAP dan *live beaconing* yang dilakukan, secara keseluruhan dapat dilihat bahwa *tools* RITA dapat digunakan untuk menerima hasil deteksi Snort sebagai data input untuk mendeteksi adanya aktivitas *beacon*, dimana semakin tinggi *score* analisis yang didapatkan, maka komunikasi tersebut semakin dianggap sebagai aktivitas *beacon.*

Sedangkan aktivitas antara alamat IP pada *score* tertinggi yang dideteksi oleh RITA, tidak dapat dideteksi oleh Snort. Hal tersebut dapat dilihat pada penyaringan hasil deteksi Snort melalui BASE. Oleh karena itu, dapat disimpulkan bahwa *tools* RITA dapat digunakan untuk melanjutkan hasil deteksi Snort dalam mendeteksi aktivitas *beacon* yang tidak dideteksi oleh Snort.

6. KESIMPULAN

Dari hasil pengujian terhadap 3 buah PCAP dan skenario aktivitas *beacon* (*live beaconing)* serta analisis yang dilakukan, dapat disimpulkan beberapa hal sebagai berikut:

a. Hasil deteksi Snort dapat dijadikan sebagai data input RITA. Hal ini menyimpulkan bahwa RITA dapat dijadikan sebagai solusi alternatif bagi pengguna IDS Snort yang ingin mengetahui adanya aktivitas *beacon* dengan melanjutkan hasil deteksi Snort ke dalam *tools* RITA.
b. Untuk melanjutkan hasil deteksi Snort ke dalam RITA, format *log* dari IDS Snort harus diubah menjadi format *log* IDS Bro/Zeek yang berbentuk TSV/JSON. Hal ini dapat dilakukan melalui perintah : bro -r *file*log/pcap local "Log::default_rotation_interval = 1 day".
c. Hasil perhitungan *score* dari pengujian menunjukkan jika setiap nilai parameter mendekati ketentuan parameter RITA, maka nilai *score* akan semakin tinggi dan mempengaruhi kemungkinan terjadinya aktivitas *beacon.*

REFERENSI

[1] Y. Tayyebi dan D. S. Bhilare, "A Comparative Study of Open Source Network Based Intrusion Detection System," *Int. J. Comput. Sci. Inf. Technol. IJCSIT*, vol. 9, no. 2, hlm. 23–26, 2018.

[2] M. A. Aydin, A. H. Zaim, dan K. G. Ceylan, "A Hybrid Intrusion Detection System Design for Computer Network Security," *Comput. Electr. Eng.*, vol. 35, no. 3, hlm. 517–526, 2009, doi: 10.1016/j.compeleceng.2008.12.005.

[3] J. Dreijer, "StealthWare - Social Engineering Malware," StealthWare, RP2 Project Thesis, 2015.

[4] D. Haselhorst, "Onion-Zeek-RITA: Improving Network Visibility and Detecting C2 Activity," *Inst.*, 2019.

[5] A. Countermeasure, "Real Intelligence Threat Analytics," *Real Intelligence Threat Analytics*, 2016. https://github.com/activecm/rita (diakses Nov 03, 2020).

[6] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, dan K.-Y. Tung, "Intrusion detection system: A Comprehensive Review," *J. Netw. Comput. Appl. - Elsevier*, vol. 36, hlm. 16–24, 2013, doi: 10.1016/j.jnca.2012.09.004.

[7] A. R. Baker dan J. Esler, *Snort IDS and IPS Toolkit.* Burlington: Syngress Publishing, Inc., 2007.

[8] T. S. Project, *Snort Users Manual 2.9.13*. 2019.

[9] Cook, "BASE, the Basic Analysis and Security Engine," BASE, the Basic Analysis and Security Engine, Nov 23, 2004. https://lwn.net/Articles/112548/ (diakses Mei 18, 2020).

[10] J. Strand, "RITA Real Intelligence Threat Analytics," RITA Real Intelligence Threat Analytics, 2015. https://www.blackhillsinfosec.com/projects/rita/ (diakses Nov 03, 2019).

[11] J. Strand, "RITA - Active Countermeasures," *RITA - Active Countermeasures*, 2016. https://www.activecountermeasures.com/free-tools/rita/ (diakses Mar 11, 2020).

[12] D. V. Myhre, "BYOB (Build Your Own Botnet)," *BYOB (Build Your Own Botnet)*, 2017. https://github.com/malwaredllc/byob (diakses Mei 22, 2020).

[13] J. Budiman, *Analysis on Remote Access Trojan Role in Advance Persistent Threat: A Concern for Cyber Criminal Investigations*. British: BRITISH COLUMBIA INSTITUTE OF TECHNOLOGY, 2016.

[14] J. Gomez, C. Gil, N. Padilla, R. Banos, dan C. Jimenez, "Design of a Snort-Based Hybrid Intrusion Detection System," *IWANN Part II LNCS 5518 Springer-Verl. Berl. Heidelb.*, vol. II, no. LNCS 5518, hlm. 515–522, 2009, doi: 10.1007/978-3-642-02481-8_75.

[15] N. AB, "Capture *file*s from Mid-Atlantic CCDC," *Capture files from Mid-Atlantic CCDC*, 2012. https://www.netresec.com/?page=MACCDC (diakses Apr 10, 2020).

[16] A. Countermeasure, "Threat Hunting Labs Introduction," *Threat Hunting Labs Introduction*, 2019. https://activecm.github.io/threat-hunting-labs/ (diakses Mar 05, 2020).