Analisa Performa Raspberry Pi sebagai *Intrusion Detection System*: Studi Kasus IDS Pada *Server Web*

Yohanes Priyo Atmojo
Sekolah Tinggi Manajemen Informatika dan Teknik Komputer (STMIK) STIKOM Bali
Jl. Raya Puputan No. 86 Renon - Denpasar, 0361-2444445
e-mail: yohanes@stikom-bali.ac.id

Abstrak

Dalam keamanan sebuah sistem, terdapat tiga aspek keamanan yaitu kerahasiaan, integritas, dan ketersediaan. Pada objek Web Server, sering kali faktor ketersediaan dan integritas, adalah faktor yang paling berpengaruh. Hal ini disebabkan oleh beragamnya user yang mengakses web server sehingga sulit untuk melakukan pengamanan apabila web server tersebut tersedia untuk publik. Salah satu cara pengamanannya adalah menggunakan Intrusion Detection System agar dapat meminimalkan dampak yang diakibatkan oleh kegagalan sebuah web server yang disebabkan oleh celah keamanan yang ada. IDS yang digunakan adalah IDS open source Bro dan Snort yang diinstall pada perangkat Raspberry Pi 3. Objek pengujian penelitian ini adalah dari segi performa dari sistem IDS yang dibuat serta diujikan dengan beberapa skenario yang menyimulasikan adanya serangan yang ditujukan ke web server. Hasil penelitian ini adalah Raspberry Pi dapat digunakan sebagai IDS, namun pada intensitas serangan yang tinggi didapat bahwa IDS BRO mengalami kendala, yaitu crash pada pertengahan pengujian sebagai akibat dari habisnya resource CPU dari Raspberry Pi.

Kata kunci: *Raspberry Pi, Intrution Detection System, Web Server.*

Abstract

In the security of a system, there are three aspects of security: confidentiality, integrity, and availability. In Web Server objects, often the availability and integrity factor, are the most influential factor. This is due to the variety of users who access the web server so it is difficult to do security if the web server is available to the public. One way to secure it is to use Intrusion Detection System in order to minimize the impact that caused by the failure or the existing security hole of a web server. IDS that used are the open source IDS, Bro and Snort that installed on device Raspberry Pi 3. The object of testing this research is in terms of performance of the IDS system created and tested with some scenarios that simulate the attack directed to the web server. The results of this study are Raspberry Pi can be used as IDS, but at the high intensity of attack found that the IDS BRO experience constraints, ie crashes in the middle of testing as a result of the lack of CPU resources from Raspberry Pi.

Keywords*: Raspberry Pi, Intrution Detection System, Web Server.*

1. Pendahuluan

Server web adalah salah satu *server* yang paling sering mengalami serangan, baik berupa serangan ke *web platform*, serangan ke *web application*, serangan ke *database*, serangan ke *web client*, maupun serangan pada *transport* dan serangan ketersediaan informasi pada suatu *website* [1]. Hasil ini juga didukung oleh data dari statistik insiden tahun 2016 yang dibuat oleh GOV-CSIRT (*Goverment Computer Security Incident Response Team*), khusus di Indonesia kasus *website defacement* terhadap *website* pemerintahan dengan domain .go.id terbilang cukup tinggi, yaitu pada statistik triwulan pertama di tahun 2016 serangan *deface* pada domain .go.id sebanyak 42% dan meningkat pada triwulan kedua dimana jumlah kasus *defacement* meningkat hingga 66.8% [2]. Salah satu Serangan yang umum terjadi pada *server web* adalah *flood attack* [3], dan *SQL Injection* [4].

IDS (*Intrution Detecting System*) adalah sebuah perangkat lunak atau perangkat keras yang bekerja secara otomatis untuk memonitor kejadian pada jaringan komputer dan menganalisis masalah keamanan jaringan [5]. Contoh aplikasi IDS yang digunakan adalah Snort [6] dan BRO [7]. Penelitian mengenai IDS dengan menganalisis perbandingan kinerja dari *Instrusion Detection System* (IDS) Snort dan Suricata dalam mendeteksi serangan *Denial of Service* (DoS), dengan hasil kedua *tools* IDS tersebut mempunyai cara dan

p-ISSN: 2089-1814; e-ISSN: 2460-3694; DOI: 10.30864/eksplora.v8i1.143

keunggulan masing-masing dalam hal penanganan serangan *Denial of Service* (DoS)[8]. Penelitian sebelumnya yang terkait dengan penelitian ini adalah penelitian penggunaan Raspberry Pi sebagai IDS yang menggunakan Snort sebagai IDS tunggal [9]. Penelitian lainnya juga menggunakan Rapsberry Pi sebagai *honeypot* dengan memanfaatkan gabungan aplikasi Snort, Kippo, dan Dionea [10]. Penelitian untuk perbandingan antara Snort dan BRO memberi hasil bahwa Bro merupakan IDS yang bersifat fleksibel dan dapat disesuaikan dengan berbagai topologi jaringan serta mampu digunakan dalam jaringan skala besar, sedangkan Snort lebih unggul dalam hal kesederhanaan, dan interoperabilitas antar sistem operasi yang digunakan [11]. Pada penelitian ini bertujuan untuk menguji kemampuan dari Raspberry Pi dalam apabila digunakan sebagai IDS yang menggunakan Snort dan BRO aplikasi tersebut.

2. Tinjauan Pustaka/*State of the Art*

2.1. *State of the Art*

Penggunaan Raspberry Pi-*Honeypot* sebagai umpan dalam jaringan merupakan solusi yang sederhana dan efisien untuk meningkatkan keamanan jaringan menggunakan Raspberry Pi dan alat *open source*. Pemanfaatan dan pengelolaan Raspberry Pi sebagai *honeypot* adalah salah satu solusi dengan biaya yang efektif dan juga menyediakan integrasi yang mudah [10]. Penggunaan Raspberry Pi sebagai IDS pada *IoT* dengan aplikasi SNORT dan melakukan berapa percobaan untuk menganalisa lalu lintas jaringan pada perangkat *IoT*. Dari percobaan tersebut didapatkan bahwa Raspberry Pi mampu bekerja dengan baik [9]. Penggunaan *cluster* Raspberry Pi telah dilakukan dalam sebuah penelitian yang bertujuan mengamankan sistem *embedded* dengan menggunakan *cluster* Raspberry Pi yang dapat menjalankan beberapa jenis dari IDS dengan cara paralel. Pengujian kelayakan arsitektur yang dibuat, skenarionya adalah menjalankan dua contoh dari Bro IDS pada dua Raspberry Pi. Hasil penelitian menunjukkan bahwa sistem tersebut berjalan secara efektif [12].

2.2. Server

Server adalah sebuah komputer yang diperuntukkan untuk menyediakan satu atau banyak layanan ke komputer lainnya atau perangkat lainnya yang berada di dalam jaringan. Layanan yang ada pada *server* dapat berupa wadah dari konten dan mengendalikan akses ke perangkat keras, perangkat lunak dan sumber daya lainnya yang berada dalam jaringan [13]. *Server* memiliki banyak jenis, ukuran dan fungsi yang berbeda-beda, contohnya seperti sebuah *file server* yang berisi *file* dimana banyak pengguna dapat mengakses *file* tersebut secara terpusat atau sebuah *server database* dimana *server* ini menyimpan data dari sebuah aplikasi yang ada pada jaringan. *Server* biasanya memiliki ukuran dan bentuk yang lebih besar serta memiliki spesifikasi seperti *memory*, prosesor, media penyimpanan, perangkat lunak dan koneksi *internet* yang disesuaikan dengan fungsi dan layanan yang disediakan oleh *server*. *Server* mungkin saja tidak memerlukan monitor dan alat masukan seperti mouse dan *keyboard*, beberapa *server* tidak memiliki sistem *GUI* (*Graphical User Interface*) melainkan hanya berupa *CLI* (*Command Line Interface*), untuk dapat mengendalikan atau mengonfigurasi *server* dapat menggunakan fitur *SSH* (*Secure Shell*) atau menggunakan *remote desktop*.

2.3. SNORT

Snort adalah suatu perangkat lunak untuk mendeteksi penyusup dan mampu menganalisis paket yang melintasi jaringan secara *real time* dan melakukan *logging* ke dalam *database* serta mampu mendeteksi berbagai serangan yang berasal dari luar jaringan. Snort merupakan sebuah produk *open source* yang 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 merupakan *network based IDS* yang menggunakan metode *Signature Based Detection*, menganalisis paket data apakah sesuai dengan jenis serangan yang sudah diketahui olehnya.

Terdapat banyak produk IDS lainnya seperti *Cisco IDS* dan *ISS Real Secure* serta produk-produk lain yang juga *open source*, Snort dipilih karena memiliki beberapa kelebihan sebagai berikut [14]:

a. Snort mudah dalam konfigurasi. Semua konfigurasi pada Snort mulai dari *file* konfigurasi sampai pada *rules*-nya sudah tersedia dan mudah untuk dilakukan. Bahkan kita dapat menambahkan *rule* sendiri untuk jenis-jenis serangan yang baru.
b. Gratis. Diluncurkan dengan lisensi GNU GPL yang berarti Snort bisa digunakan secara bebas tanpa biaya apa pun.
c. Dapat berjalan pada berbagai macam sistem operasi. Awalnya Snort dikembangkan dalam lingkungan sistem operasi *UNIX*, tetapi Snort dapat digunakan pada sistem operasi yang lainnya.

2.4. BRO

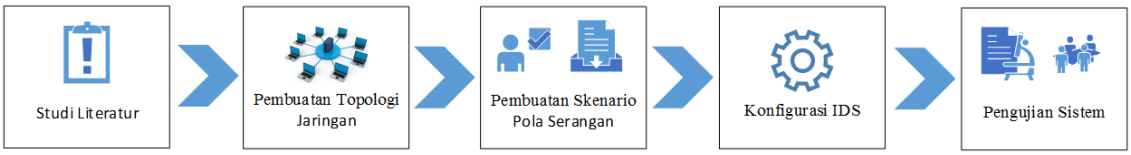
Bro adalah aplikasi *Network Intrusion Detection System* (NIDS) *open-source*, berbasis *UNIX* yang pasif memonitor lalu lintas jaringan dan mencari aktivitas yang mencurigakan mendeteksi Bro mendeteksi aktivitas mencurigakan intrusi melalui proses *parsing* data pertama untuk mengekstrak pola semantik di level aplikasi dan kemudian melaksanakan analisis berdasarkan perilaku aplikasi dan membandingkan aktivitas dengan pola yang dianggap tidak biasa. Analisis dari program Bro meliputi deteksi serangan tertentu termasuk yang didefinisikan oleh tanda tangan, tetapi juga yang didefinisikan dalam hal kejadian dan kegiatan yang tidak biasa (misalnya, *host* tertentu menghubungkan ke layanan tertentu, atau pola upaya untuk melakukan autentikasi). Bro menggunakan bahasa kebijakan khusus yang memungkinkan sebuah situs untuk menyesuaikan operasi Bro menggunakan pola tertentu yang mengizinkan sebuah sistem dapat terhubung dengan aplikasi ini. Jika Bro mendeteksi sesuatu paket berbahaya, maka Bro mencatatnya dalam sebuah log, mengingatkan operator secara *real-time*, menjalankan perintah sistem operasi (misalnya, untuk mengakhiri sambungan atau memblokir koneksi berbahaya secara langsung). Selain itu, rincian *file log* Bro bisa menjadi sangat berguna untuk *forensic* [15].

3. Metode Penelitian

3.1. Alur Penelitian

Dalam melakukan penelitian, tahapan-tahapan yang dilakukan adalah sebagai berikut:

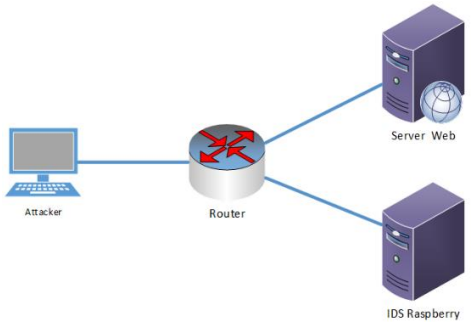
1. Studi literatur, yaitu mempelajari karakteristik *single-board computer*, dalam hal ini menggunakan Raspberry Pi. Literatur yang digunakan lebih banyak menggunakan referensi dari web resmi dari Raspberry Pi dan eLinux. Studi literatur juga mempelajari aplikasi IDS, yaitu Snort dan BRO.
2. Perancangan topologi jaringan pengujian, yaitu mempelajari dan merancang topologi jaringan yang digunakan sebagai bahan pengujian IDS.
3. Pembuatan skenario dan pola serangan terhadap sistem, yaitu pembuatan skenario yang dijalankan pada pengujian serta membuat pola serangan yang disimulasikan pada sistem yang dibangun.
4. Konfigurasi *rule* pada IDS, yaitu melakukan konfigurasi agar IDS mengenali pola serangan yang dijalankan serta membuat optimasi pada program IDS.
5. Pengujian sistem, yaitu melakukan pengujian dari seluruh skenario yang dibuat.



Gambar 1. Alur penelitian.

3.2. Topologi Jaringan Pengujian Sistem

Gambar 1 merupakan gambaran topologi jaringan yang digunakan dalam penelitian ini. Topologi ini adalah topologi yang menyimulasikan bahwa pengujian serangan dilakukan melalui komputer *Attacker* dan *router* dikonfigurasikan untuk mengaktifkan *port mirroring* [16], sehingga *traffic* data yang mengarah ke *Server Web* akan diterima juga oleh mesin Raspberry Pi yang difungsikan sebagai IDS. Topologi ini menggunakan *range* IP *local*, sehingga memudahkan untuk melakukan *monitoring* terhadap paket yang masuk serta mengurangi adanya paket data yang terkirim dalam jaringan ini.



Gambar 2. Topologi jaringan pengujian sistem.

3.3. Skenario Pengujian

Skenario pengujian dibagi menjadi beberapa skenario yang dilakukan untuk melakukan pengujian terhadap Web Server dan IDS Raspberry PI, antara lain:

Tabel 1. Skenario pengujian.

Parameter Pengujian	Nilai
Alat	Raspberry Pi 3
Sistem Operasi	Arch Linux ARM
Memori RAM	992MB
Memori GPU	32 MB
Aplikasi IDS	Snort dan BRO
Aplikasi Serangan SYN Flood	Hping3[17]
Parameter Serangan SYN Flood	100, 1000, 10000 (dalam paket/detik)
Aplikasi Serangan SQL Injection	Sqlmap[18]
Parameter Serangan SQL Injection	10, 100, 1000 (threads per query)
Pengukuran Performa	Network troughput, CPU load, Memory utilization

3.4. Rules SNORT

Rules SNORT yang digunakan adalah *rules* yang sudah tersedia dari pengembang SNORT itu sendiri, namun karena skenarionya sudah ditentukan, maka *rules* tersebut dispesifikkan untuk 2 buah serangan, yaitu serangan DOS (*Denial of Service*), yang disimulasikan sebagai *syn-flood* menggunakan *tools* hping3. *Rules* SNORT yang digunakan disederhanakan menjadi seperti berikut:

```
alert tcp any any -> $HOME_NET 80 (flags: S; msg:"Possible TCP DoS"; flow:
stateless; threshold: type both, track by_dst, count 70, seconds 10;
sid:10001;rev:1;)
```

Sedangkan untuk serangan *SQL Injection* melalui paket *HTTP* yang disimulasikan menggunakan SQLMap, maka *rules* SNORT dibuat untuk melakukan inspeksi terhadap paket data yang memiliki *user-agent sqlmap*, seperti berikut:

```
alert tcp any any -> $HOME_NET 80 (msg:"INDICATOR-SCAN sqlmap SQL injection
scan attempt"; flow:to_server,established; content:"User-Agent|3A| sqlmap";
fast_pattern:only;        http_header;        metadata:service        http;
reference:url,sqlmap.sourceforge.net;   classtype:web-application-activity;
sid:19779; rev:6;)
```

3.4. Rule Bro

Konfigurasi untuk BRO IDS menggunakan *script default* dari BRO yang sudah tersedia pada *source code* Bro, sehingga dapat langsung digunakan dalam pengujian yang dilakukan, dan cukup mengaktifkan konfigurasi tersebut untuk mendeteksi serangan *syn-flood* dan *SQL injection* saja.

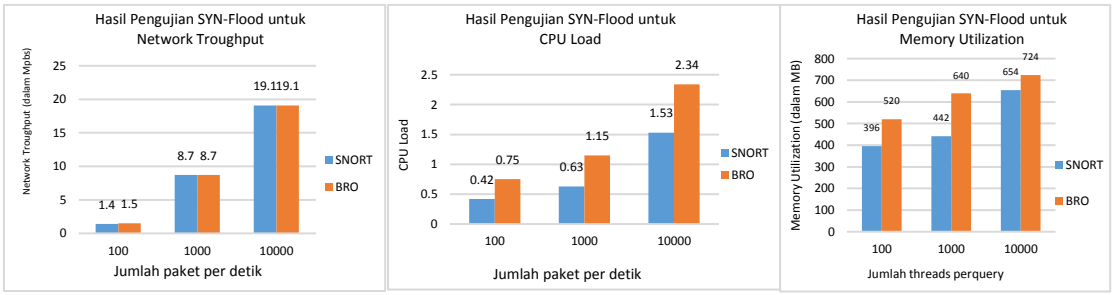
4. Hasil dan Pembahasan

4.1. Hasil Pengujian Serangan SYN *Flood*

Pada pengujian SYN *flood*, dilakukan sebanyak 3 kali serangan yang memiliki intensitas berbeda, yaitu dengan mengirimkan 100 paket per detik, 1000 paket per detik, dan 10000 paket per detik. *Syntax hping3* yang digunakan dalam pengujian adalah sebagai berikut:

```
hping3 -S -i u$X -p 80 $IP_TARGET
```

Variabel $X adalah jumlah paket yang dikirim dalam setiap detik, dimana nilai $X = 1000 untuk 100 paket per detik, $X = 100 untuk 1000 paket per detik, dan $X=10 untuk 10000 paket per detik. IDS Snort dan Bro dijalankan secara bergantian dengan mengulangi proses yang sama, sehingga menghasilkan hasil seperti berikut:



Gambar 3. Hasil Pengujian Serangan SYN Flood

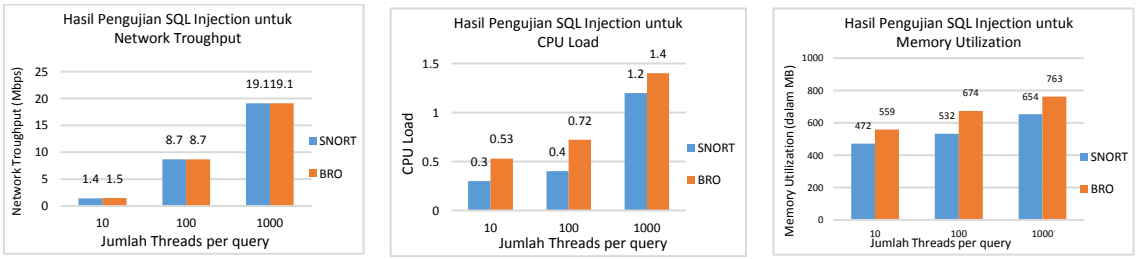
Dari Gambar 3, diperoleh kesimpulan bahwa hasil SNORT dan BRO tidak terlalu jauh di bagian *Network Throughput*, hasil ini disebabkan oleh *bottleneck* pada sisi *Router* RB450G yang digunakan. Pada *CPU Load* dan *Memory Utilization*, terlihat bahwa BRO lebih banyak menghabiskan *resource CPU* dan *memory* karena BRO menjalankan aplikasinya dalam beberapa *threads* secara paralel, berbeda dengan SNORT yang hanya menjalankan 1 *threads* saja. Sebagai catatan, pada pengujian terakhir, dengan opsi 10000 paket per detik, Raspberry Pi sempat mengalami *crash*, sehingga harus dilakukan *restart* sistem dan pengujian diulang lagi dari awal.

4.2. Hasil Pengujian Serangan SQL *Injection*

Pengujian SQL *injection*, dilakukan sebanyak 3 kali serangan yang sama, namun dengan penggunaan *threads* serangan yang berbeda, yaitu dengan menggunakan 10 *threads*, 100 *threads*, 1000 *threads*. *Syntax* sqlmap yang digunakan adalah sebagai berikut:

```
python sqlmap.py --url http://$IP_TARGET/data.php?id=1 --threads $X
```

Pada pengujian *SQL Injection*, *variable* $X adalah jumlah *threads* per *query* yang dijalakan oleh program *sqlmap* pada saat melakukan percobaan serangan *SQL* dengan nilai 10, 100, dan 1000 *threads* per *query*. Hasil yang didapat adalah sebagai berikut.



Gambar 4. Hasil pengujian serangan SQL *injection*.

Dari Gambar 4, diperoleh kesimpulan bahwa hasil yang didapatkan memiliki *trend* yang sama dengan pengujian *SYN-Flood*, dimana BRO masih memerlukan *resource* yang jauh lebih banyak dibanding SNORT, tetapi hasil pengujian tidak menimbulkah *crash* pada sistem.

5. Simpulan

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa baik SNORT maupun BRO IDS dapat berjalan baik pada Raspberry Pi, namun pada saat pengujian dengan intensitas serangan yang tinggi, didapatkan kendala pada BRO IDS, bahkan berakibat *crash* terhadap Raspberry Pi. Hasil selisih pada pada *CPU Load* dan *Memory Utilization* mencapai rata-rata di atas 50 % SNORT jauh lebih sedikit memerlukan *resource* dibanding BRO IDS. Hasil ini sejalan dengan penelitian yang terdahulu yang juga menyimpulkan hasil yang sama [11].

Daftar Pustaka

[1] Scambray Joel, Liu Vincent, Sima Caleb. *Hacking Exposed Web Applications: Web Application Security Secrets And Solutions Third Edition*. 2011:11

[2] Gov-CSIRT, *Data Statistik Serangan terhadap Domain .go.id yang Direspon Tahun 2016*. http://govcsirt.kominfo.go.id/statistik-insiden-respondomain-go-id/. Diakses terakhir 20 Juli 2017.
[3] S.S. Chapade, K.U Pandey, D.S Bhade, *Securing Cloud Servers Against Flooding Based DDOS Attacks*, International Conference on Communication Systems and Network Technologies (CSNT), pp. 524-528, 2013.
[4] P. Kumar, *R.K.A survey on SQL injection attacks, detection and prevention techniques*, First International Conference, pp.1-5, 2012.
[5] John R. Vacca. *Computer and Information Security Handbook*. Waltham: Elsevier, Inc. 2013:92-94
[6] Liao, Hung-Jen, et al. *Intrusion detection system: A comprehensive review*. Journal of Network and Computer Applications, 2013, 36.1: 16-24.
[7] Ganesh Kumar Varadarajan. *Web Application Attack Analysis Using Bro IDS*. 2012. https://www.sans.org/reading-room/whitepapers/detection/web-application-attack-analysis-bro-ids-34042. Terakhir diakses tanggal 20 Maret 2017
[8] Hirrandi, raiman, D. S. Sali Alas Majapahit, and D. S. Doddy Ferdiansyah. *Analisis Perbandingan Kinerja Intrusion Detection System (IDS) Snort dan Suricata Dalam Mendeteksi Serangan Denial of Service Pada Server Linux*. Diss. Fakultas Teknik Universitas Pasundan, 2017.
[9] Alessandro Sforzin, Mauro Conti, F´elix G´omez M´armol, Jens-Matthias Bohli. *RPiDS: Raspberry Pi IDS A Fruitful Intrusion Detection System for IoT*. 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress. 2016
[10] Surendra Mahajan, Akshay Mhasku Adagale, Chetna Sahare. *Intrusion Detection System Using Raspberry PI Honeypot in Network Security*. IJSC. 2016
[11] Mehra, Pritika. *A brief study and comparison of snort and bro open source network intrusion detection systems*. International Journal of Advanced Research in Computer and Communication Engineering, 2012, 1.6: 383-386.
[12] Mohamed Salim LMIMOUNI, Khalid BOUKHDIR, Hicham MEDROMI, Siham BENHADOU. *Using a Cluster for Securing Embedded Systems*. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 9, 2016
[13] Misty E. Vermaat, Susan L. Sebok, Steven M. Freund, Jennifer T. Campbell, Mark Frydenberg, *Discovering Computers, Essentials* ©. 2016:116
[14] Setiawan Junior, Abraham Nethanel. Perancangan dan Implementasi Intrusion Detection System pada Jaringan Nirkabel BINUS University. Skripsi. Jakarta: BINUS University. 2009.
[15] Ganesh Kumar Varadarajan. *Web Application Attack Analysis Using Bro IDS*. 2012. https://www.sans.org/reading-room/whitepapers/detection/web-application-attack-analysis-bro-ids-34042. Terakhir diakses tanggal 12 Pebuari 2018
[16] Frattura, David E.; graham, Richard W.; roese, John. *Method for network traffic mirroring with data privacy*. U.S. Patent No 8,239,960, 2012.
[17] Buchanan, Bill, et al. *A methodology to evaluate rate-based intrusion prevention system against distributed denial-of-service (DDoS)*. Cyberforensics 2011, 2011.
[18] Clarke-Salt, Justin. *SQL injection attacks and defense*. Elsevier, 2009.