Analisis Perbandingan Sistem Keamanan Jaringan Menggunakan Snort dan *Netfilter*

Muhammad Suyuti Ma'sum[#1], M. Azhar Irwansyah[#2], Heri Priyanto[#3]
[#]*Program Studi Teknik Informatika Universitas Tanjungpura*
Jl. Prof Dr H. Hadari Nawawi, Kota Pontianak, 78115
[1]muhammadsuyutimasum@yahoo.co.id, [2]irwansyah.azhar@untan.ac.id, [3]heripriyanto.stmt@untan.ac.id

Abstrak - **Penyediaan layanan berbasis *web* di program studi (Prodi) Teknik Informatika Universitas Tanjungpura (UNTAN) berupa sistem yang multi *user* menimbulkan kerentanan pada sistem. Misalnya, terjadi intrusi-intrusi keamanan yang dilakukan oleh sebagian *user* dalam upaya penyerangan terhadap sistem. Upaya pengamanan terhadap sistem informasi telah dilakukan melalui penelitian tentang *monitoring* keamanan jaringan menggunakan snort oleh Asep Fauzi Mutaqin tahun 2015 yang menghasilkan saran untuk dikembangkan dengan menambahkan fungsi *Intrusion Prevension System* (IPS) pada snort. Berdasarkan saran tersebut dilakukan penelitian dengan mengembangkan sistem keamanan jaringan menggunakan snort mode *inline* untuk dapat menjalankan fungsi IPS. Kemudian agar dapat diketahui performa snort mode *inline* diperlukan analisis perbandingan dengan sistem keamanan jaringan lain. *Netfilter* menggunakan *Advanced Policy Firewall* (APF) dan *Mod Evasive* sebagai sistem keamanan jaringan yang digunakan dalam perbandingan performa snort mode *inline* sebagai upaya mengetahui sistem keamanan jaringan yang sesuai untuk diimplementasikan di Prodi Teknik Informatika. Berdasarkan penelitian analisis perbandingan yang telah dilakukan menghasilkan sebagai berikut : (1) Perangkat keras yang digunakan oleh *netfilter* yaitu sebuah *Server Netfilter* sedangkan pada snort menggunakan PC Snort dan *Server* Snort. (2) *Server* Snort menggunakan *memory* sebesar 330668 KiB dan PC Snort menggunakan *memory* sebesar 175488 KiB sedangkan *Server Netfilter* menggunakan *memory* yang lebih besar yaitu 457968 KiB. (3) 2 komponen sistem dan 5 tahap konfigurasi merupakan kebutuhan perangkat lunak pada *netfilter* sedangkan pada snort membutuhkan 8 komponen sistem dan 10 tahap konfigurasi. (4) Persentase pencegahan serangan antara lain : snort 100,00 % lebih baik saat serangan *ping attack* dari pada *netfilter, netfilter* 50,32 % lebih baik saat serangan DoS dari pada snort, dan *netfilter* 91,95 % lebih baik saat serangan *port scanning* dari pada snort.**

Kata Kunci : Keamanan Jaringan, Snort, *Netfilter*, *Advanced Policy Firewall, Mod Evasive,* Prodi Teknik Informatika UNTAN.

I. PENDAHULUAN

Perkembangan dan pemanfaatan teknologi komputer semakin meningkat berdasarkan kebutuhan dalam pertukaran informasi. Di program studi (Prodi) Teknik Informatika Universitas Tanjungpura (UNTAN) telah mengembangkan teknologi komputer dalam pertukaran informasi. Pemanfaatan teknologi komputer pada sistem tersebut digunakan sebagai manajemen informasi, salah satunya yaitu pada sistem repositori. Kemudian teknologi komputer berbasis *web* merupakan teknologi yang digunakan pada sistem repositori tersebut dalam mengembangkan pertukaran informasi di Prodi Teknik Informatika UNTAN.

Perkembangan *website* yang semakin cepat dengan berbagai macam fungsi dan kebutuhan, menuntut peningkatan pada keamanan jaringan *web server*. *Website* Prodi Teknik Informatika UNTAN yang dapat diakses oleh banyak *user* melalui jaringan *Local Area Network* (LAN) berbasis *wired* maupun *wireless fidelity* (Wifi). Memungkinkan *recources website* tersebut dapat diakses oleh banyak *user*. Sedangkan *user* yang dapat mengakses *website* secara bebas merupakan kerentanan bagi *web server* apabila keamanannya tidak diperhatikan dan tidak dikembangkan. Oleh sebab itu keamanan pada *web server* diperlukan untuk mengatasi *user* yang melakukan serangan sehingga terjaminnya ketersedian layanan bagi *user* lain.

Pada penelitian Sistem *Monitoring* Keamanan Jaringan Prodi Teknik Informatika Melalui SMS *Alert* dengan Snort yang dilaksanakan oleh Asep Fauzi Mutaqin pada tahun 2015 telah diterapkan dan telah berhasil. Sistem *monitoring* tersebut mendeteksi serangan-serangan yang masuk di jaringan dengan hasil deteksi pada protokol ICMP berjumlah 567 (0.16%), UDP berjumlah 9.817 (2.81%) dan TCP berjumlah 339.109 (97.03%). Saran dalam penelitian tersebut untuk ditambahkan *iptables* sebagai *Intrussion Prevention System* (IPS) [1]. *Inline mode* merupakan fitur yang dapat digunakan sesuai dengan saran pada penelitian sebelumnya yaitu agar dapat menjalankan *iptables* sebagai *firewall.* Peneliti membandingkan snort dalam *inline mode* dengan *netfilter* untuk mengetahui kebutuhan dan performa snort sebagai pencegah serangan-serangan pada *web server*.

II. URAIAN PENELITIAN

A. Konsep Dasar Keamanan Jaringan

Keamanan jaringan adalah proses untuk mencegah dan mengidentifikasi penggunaan yang tidak sah dari jaringan komputer. Langkah-langkah pencegahan membantu menghentikan pengguna yang tidak sah untuk mengakses setiap bagian dari sistem jaringan komputer. Keamanan jaringan komputer sendiri bertujuan untuk mengantisipasi resiko pada jaringan komputer berupa bentuk ancaman fisik maupun *logic* baik langsung ataupun tidak langsung mengganggu aktivitas yang sedang berlangsung dalam jaringan komputer. Secara umum terdapat 3 hal dalam konsep keamanan jaringan, yaitu [2]:

1. Resiko atau tingkat bahaya (*risk*)

Submitted 25-01-2017; Revised 25-01-2017; Accepted 25-01-2017

Menyatakan seberapa besar kemungkinan dimana penyusup (*intruder)* berhasil mengakses komputer dalam suatu jaringan.

2. Ancaman (*threat*)

Menyatakan sebuah ancaman yang datang dari seseorang yang mempunyai keinginan untuk memperoleh akses ilegal ke dalam suatu jaringan komputer seolah-olah mempunyai otoritas terhadap jaringan tersebut.

3. Kerapuhan sistem (*vulnerability*)

Menyatakan seberapa kuat sistem keamanan suatu jaringan komputer yang dimiliki dari seseorang dari luar sistem yang berusaha memperoleh akses ilegal terhadap jaringan komputer tersebut.

Keamanan sendiri menyangkut 3 elemen dasar yaitu :

1. Keamanan jaringan (*network security*)
2. Keamanan aplikasi (*application security)*
3. Keamanan komputer (*computer security*)

B. Firewall

Firewall adalah suatu cara atau mekanisme yang diterapkan baik terhadap *hardware*, *software*, ataupun sistem dengan tujuan untuk melindungi. Perlindungan dapat dilakukan dengan menyaring, membatasi, atau bahkan menolak suatu atau semua hubungan/kegiatan dari suatu segmen pada jaringan pribadi dengan jaringan luar yang bukan merupakan ruang lingkupnya. Segmen tersebut dapat merupakan sebuah *workstation, server, router,* atau *Local Area Network* [3].

Firewall secara umum diperuntukkan untuk melayani :

1. Mesin/Komputer

Setiap mesin komputer yang terhubung langsung ke jaringan luar atau internet dan menginginkan semua yang terdapat pada komputernya terlindungi.

2. Jaringan

Jaringan komputer yang terdiri lebih dari satu buah komputer dan berbagai jenis topologi jaringan yang digunakan, baik yang dimiliki oleh perusahaan, organisasi dsb.

Firewall mempunyai beberapa tugas :

1. Mengimplementasikan kebijakan *security* di jaringan (*site security policy*) : jika aksi tertentu tidak diperbolehkan oleh kebijakan ini, maka *firewall* harus meyakinkan bahwa semua usaha yang mewakili operasi tersebut harus gagal atau digagalkan. Dengan demikian, semua akses ilegal antar jaringan (tidak diotorisasikan) akan ditolak.
2. Melakukan *filtering* : mewajibkan semua *traffic* yang ada untuk dilewatkan melalui *firewall* bagi semua proses pemberian dan pemanfaatan layanan informasi. Dalam konteks ini, aliran paket data dari/menuju *firewall*, diseleksi berdasarkan IP *address*, nomor *port*, atau arahnya, dan disesuaikan dengan kebijakan *security*.
3. *Firewall* juga harus dapat merekam/mencatat *even-even* mencurigakan serta memberitahu administrator terhadap segala usaha-usaha menembus kebijakan *security*.

C. Netfilter

Netfilter.org adalah *home* bagi perangkat lunak *framework packet filtering* dalam Linux 2.4.x dan seri kernel selanjutnya. *Software* umumnya terkait dengan netfilter.org adalah iptables. *Software* dalam kerangka ini memungkinkan penyaringan paket, *Network Address Translation* (NAT) dan paket *mangling* lainnya. Ini telah dirancang ulang dan sangat ditingkatkan dari penerus sebelumnya yaitu ipchains 2.2.x Linux dan sistem Linux 2.0.x ipfwadm [4].

Netfilter adalah suatu pengaturan dari *hooks* di dalam kernel Linux yang memungkinkan kernel modul untuk mendaftar fungsi *callback* dengan *stack* jaringan. Sebuah fungsi *callback* terdaftar kemudian dipanggil kembali untuk setiap paket yang melintasi *hook* dalam *stack* jaringan. *Iptables* adalah struktur tabel generik untuk mendefinisikan seperangkat pengaturan. Setiap aturan dalam sebuah tabel IP terdiri dari sejumlah pengklasifikasi (pencocokan *iptables*) dan satu tindakan yang terhubung (target *iptables*).

D.Snort

Snort merupakan sebuah aplikasi atau *tool* sekuriti yang berfungsi untuk mendeteksi intrusi-intrusi jaringan (penyusupan, penyerangan, pemindaian, dan beragam bentuk ancaman lainnya), sekaligus juga melakukan pencegahan. Dalam praktiknya, snort sangat andal untuk membentuk *logging* paket-paket dan analisis trafik-trafik secara *real-time* dalam jaringan berbasis TCP/IP [5].

Snort dapat dikonfigurasi untuk mengawasi jaringan dari jenis serangan tertentu, dan menghubungi anggota tim penanggulangan insiden saat serangan benar-benar dilancarkan oleh penyerang. Fitur-fitur inilah yang menjadikan snort sebuah sistem pendeteksi gangguan dan serangan jaringan, yang sangat berguna bagi tim penanggulangan insiden.

Secara prinsip, snort memerankan tiga fungsi utama :

1. Sebagai penangkal program-program *sniffer* paket-paket (seperti *tcpdump*).
2. Sebagai *packet logger* (berguna untuk men-debug trafik-trafik jaringan).
3. Sebagai sistem pencegah intrusi untuk sistem-sistem jaringan.

Snort dapat dikonfigurasikan untuk berjalan pada mode-mode berikut ini :

1. *Sniffer mode*
   Bertugas membaca paket-paket dari jaringan dan menampilkan tampilan dalam bentuk aliran tak terputus pada konsol (layar).
2. *Packet logger mode*
   Bertugas mencatat log dari paket-paket ke dalam disk.
3. NIDS (*Network Intrussion Detection System*) *mode*
   Memiliki konfigurasi kompleks, namun bisa dimodifikasi, yang membuat snort bisa menganalisis arus jaringan untuk dibandingkan dengan rangkaian *ruleset* yang dibuat oleh *user*, sekaligus melakukan beberapa tindakan berdasarkan hal yang diamatinya.
4. *Inline mode*
   Bertugas mengambil paket dari iptable (daripada libpcap) dan menginstruksikan *iptables* untuk menolak atau meneruskan paket tersebut berdasarkan jenis *rule* dari snort yang digunakan.

E. Advanced Policy Firewall (APF)

Advanced Policy Firewall (APF) adalah *iptables* (*netfilter*) berdasarkan sistem *firewall* dirancang berdasarkan kebutuhan penting dari internet yang mengembangkan *server* dan kebutuhan yang unik dari pengembangan *server* berbasis Linux

Korespondensi : Muhammad Suyuti Ma'sum

[6]. Konfigurasi APF dirancang untuk menjadi lebih informatif dan menyajikan kemudahan kepada pengguna untuk mengikuti proses, dari atas ke bawah dari file konfigurasi.

Sisi teknis APF adalah seperti pemanfaatan fitur stabil terbaru dari *iptables* (*netfilter*) proyek untuk menyediakan firewall yang sangat kokoh dan kuat. Penyaringan yang dilakukan oleh APF antara lain sebagai berikut :

- *Static Rule Base Policies*
- *Connection Based Sateful Policies*
- *Sanity Based Policies*

F. Mod Evasive

Mod evasive adalah *evasive maneuvers module* untuk *Apache Web Server* sebagai aksi *evasive* pada saat terjadi HTTP DoS atau serangan DDoS atau serangan *brute force* [6]. Mod evasive juga dirancang sebagai pendeteksi dan *network management tools,* dan bisa secara mudah dikonfigurasi agar terdeksi dengan *ipchains, firewalls, routers,* dan *etcetera. Mod evasive* dapat menghasilkan laporan *abuses* melalui *e-mail* dan fasilitas *syslog.*

Pendeteksian dilakukan dengan membuat sebuah *internal dynamic hash table* dari alamat IP dan URls, dan menolak (*deny*) alamat IP mana saja yang berasal dari :

- *Request* halaman yang sama secara berulang kali pada selang waktu tertentu per detik.
- Membuat lebih dari 50 *concurrent request* pada *child* yang sama per detik.
- Mebuat permintaan-permintaan sewaktu di-*blacklist* secara temporer (pada *blocking list*).

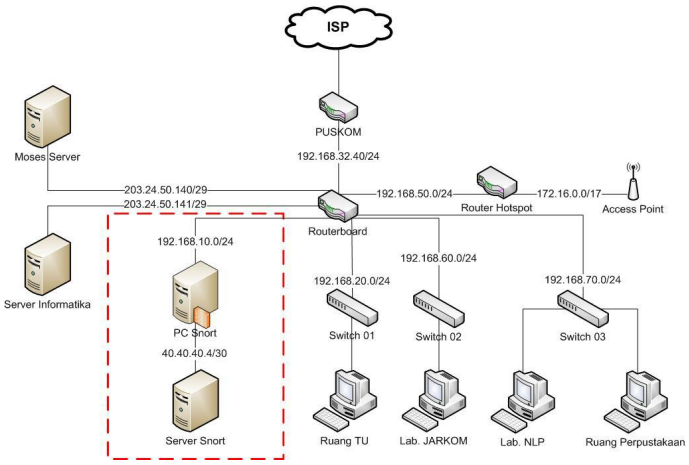
III. PERANCANGAN SISTEM

A. Perancangan Sistem dan Arsitektur Jaringan

Perancangan pada penelitian ini menggunakan *network* 192.168.10.0/24 pada Routerboard di Jaringan Teknik Informatika. Adapun perancangan yang dilakukan, yaitu antara lain sebagai berikut :

1. Perancangan Sistem dan Arsitektur Jaringan Snort

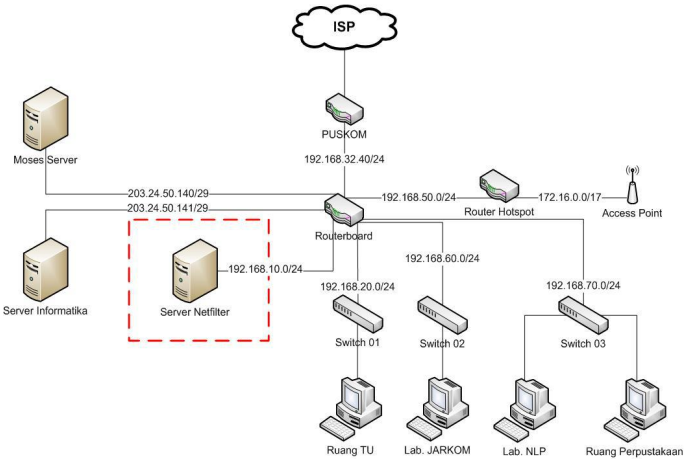
Snort yang dirancang adalah sebagai *router* yang terhubung pada Routerboard. Pengembangan arsitektur jaringan snort yang akan diterapkan dijelaskan pada gambar 1.



Gambar 1. Pengembangan Arsitektur Jaringan Snort

2. Perancangan Sistem dan Arsitektur Jaringan *Netfilter*

Server Netfilter yang dikonfigurasi selain menjalankan *web service* juga menjalankan *netfilter* dan dihubungkan pada Routerboard. Pengembangan arsitektur jaringan *netfilter* yang akan diterapkan dijelaskan pada gambar 2.



Gambar 2. Pengembangan Arsitektur Jaringan *Netfilter*

B. Implementasi Pengembangan Arsitektur Jaringan

Implementasi yang akan dilakukan yaitu dengan menerapkan sistem snort dan sistem *netfilter* seperti pada gambar 1 dan gambar 2.

Implementasi sistem dengan menggunakan Snort yang dilakukan antara lain *update* dan *upgrade kernel*, instalasi paket pedukung Snort, konfigurasi DAQ (*Data Acquisition Library*), konfigurasi snort, konfigurasi *rules snort*, konfigurasi *database snort,* konfigurasi barnyard2, konfigurasi adodb, dan konfigurasi snorby. Kemudian implementase sistem dengan menggunakan *netfilter* yang dilakukan adalah konfigurasi *Advance Policy Firewall* (APF) dan konfigurasi *Mod Evasive*.

C. Pengujian Sistem Snort dan Netfilter

Pada tahap ini dilakukan pengujian terhadap sistem keamanaan yang menggunakan Snort dan sistem keamanaan yang menggunakan *netfilter* untuk memastikan bahwa sistem keamanaan yang telah dikonfigurasi dapat berjalan sesuai dengan perancangan. Berikut pengujian sistem yang akan dilakukan yaitu :

- Pengujian Sistem Snort

Pengujian fungsional snort dilakukan dengan perintah "snort -V", apabila snort telah berjalan maka akan menampilkan notifikasi seperti pada gambar 3.

```
root@netsnort:/home/suyuti# snort -V

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.8.2 GRE (Build 335)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.7.4
           Using PCRE version: 8.35 2014-04-04
           Using ZLIB version: 1.2.8
```

Gambar 3. Pengujian Fungsional Snort

Snort yang telah berjalan kemudian diuji dalam *inline mode* untuk mengetahui snort menjalankan fungsi sebagai IPS (*Intrusion Prevension System*). Pengujian dilakukan dengan menjalankan perintah "snort -T -c /etc/snort/snort.conf -Q -i

Korespondensi : Muhammad Suyuti Ma'sum

enp0s15:enx00e04c534458" dan akan menampilkan seperti pada gambar 4.

```
afpacket DAQ configured to inline.
Acquiring network traffic from "enp0s15:enx00e04c534458".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.8.2 GRE (Build 335)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.7.4
           Using PCRE version: 8.35 2014-04-04
           Using ZLIB version: 1.2.8

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 2.6  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>

Snort successfully validated the configuration!
Snort exiting
```

Gambar 4. Pengujian Snort Mode *Inline*

Snort yang menjalankan mode *inline* kemudian dilakukan pengamatan penggunaan *memory* yaitu dengan menjalankan perintah "top" pada PC Snort dan *Server Netfilter*. Hasil dari pengamatan ini yaitu penggunaan *memory* pada PC Snort sebesar 175488 KiB dan penggunaan *memory* pada *Server* Snort sebesar 330668 KiB.

- Pengujian Sistem *Netfilter*

1. Pengujian APF

Pengujian yang dilakukan pada APF dengan cara menjalankan perintah "apf -s" pada *terminal* untuk dapat diketahui bahwa APF telah aktif. Notifikasi yang akan tampil bahwa APF dalam keadaan berjalan seperti pada gambar 5.

```
root@netfilter:~# apf -s
apf(2226): {glob} activating firewall
apf(2266): {glob} determined (IFACE_IN) enp0s15 has address 192.168.10.2
apf(2266): {glob} determined (IFACE_OUT) enp0s15 has address 192.168.10.2
apf(2266): {glob} loading preroute.rules
apf(2266): {resnet} downloading http://cdn.rfxn.com/downloads/reserved.netw
orks
apf(2266): {resnet} parsing reserved.networks into /etc/apf/internals/reser
ved.networks
apf(2266): {glob} loading reserved.networks
apf(2266): {glob} loading bt.rules
apf(2266): {glob} loading deny_hosts.rules
apf(2266): {trust} deny inbound udp 192.168.10.0/24 to port 80
apf(2266): {trust} deny outbound udp 192.168.10.0/24 to port 80
apf(2266): {glob} loading common drop ports
```

Gambar 5. Pengujian APF

2. Pengujian *Mod Evasive*

Pengujian *Mod Evasive* setalah ter-*install* pada *Server Netfilter* dilakukan dengan memverifikasi bahwa *Mod Evasive* telah *enable* dengan menjalankan perintah "apachectl -M | grep evasive". Notifikasi "shared" menandakan *Mod Evasive* telah aktif dengan tampilan seperti pada gambar 6.

```
root@netfilter:~# apachectl -M | grep evasive
 evasive20_module (shared)
root@netfilter:~#
```

Gambar 6. Verifikasi *Mod Evasive*

Kemudian menjalankan *script* test.pl untuk membuat 100 *requests* pada *Server Netfilter* agar dapat diketahui berjalan atau tidaknya sistem keamanan yang terlah ter-*install*. Perintah yang digunakan yaitu "perl /usr/share/doc/libapache2-mod-evasive/examples/test.pl" maka akan terdapat notifikasi 403 yang berarti *server* dapat menolak *requests*. Tampilan pengujian *Mod Evasive* pada gambar 7.

```
root@netfilter:~# perl /usr/share/doc/libapache2-mod-evasive/examples/test.pl
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
```

Gambar 7. Pengujian *Mod Evasive*

3. Penggunaan *Memory* pada *Server Netfilter*

Server Netfilter yang telah menjalankan *netfilter* dilakukan pengamatan penggunaan *memory* yaitu dengan menjalankan perintah "top". Dari pengamatan tersebut dapat diketahui bahwa penggunaan *memory* pada *Server Netfilter* yaitu sebesar 457968 KiB.

D. Pengujian Keamanan Snort dan Netfilter

1. Pengujian menggunakan *Ping Attack* (ICMP Traffic)

Pengujian dilakukan dengan cara menyerang *Server* Snort dan *Server Netfilter* oleh *intruder* menggunakan *hping3*. Proses penyerangan yang dilakukan adalah dengan melakukan *ping* menggunakan *hping3* dengan jumlah *count* 1000 (*packets sent* 1000 atau *packet receive* 1000) oleh *intruder* sebanyak 20 kali ke IP *address* pada *server* snort dan *server netfilter*. Hasil yang diperoleh berupa rata-rata persentase pencegahan dengan besaran pada snort adalah 100,00 % dan pada *netfilter* 0,05 %.

2. Pengujian menggunakan DoS/DDoS

Pengujian dilakukan dengan cara menyerang *Server* Snort dan *Server Netfilter* oleh *intruder* menggunakan ApacheBench (ab). Proses penyerangan yang dilakukan adalah dengan mengirim 1000 permintaan menggunakan ab sebanyak 20 kali ke IP *address* pada *server* snort dan *server netfilter*. Hasil yang diperoleh berupa rata-rata persentase pencegahan dengan besaran pada snort adalah 49,45 % dan pada *netfilter* 98,77 %.

3. Pengujian menggunakan *Port Scanning*

Pengujian dilakukan dengan cara menyerang *Server* Snort dan *Server Netfilter* oleh *intruder* menggunakan ZenMap. Proses penyerangan yang dilakukan adalah dengan *scanning* pada *port* dengan interval 1 - 1000 sebanyak 20 kali ke IP *address* pada *server* snort dan *server netfilter*. Hasil yang diperoleh berupa rata-rata persentase *filtered ports* dengan besaran pada snort adalah 0.00 % dan pada *netfilter* 92,00 %.

E. Analisis Perbandingan Sistem Keamanan Snort dan Netfilter

Rincian analisis perbandingan sistem keamanan menggunakan snort dan *netfilter* adalah sebagai berikut :

Korespondensi : Muhammad Suyuti Ma'sum

1. Perangkat keras lebih banyak digunakan oleh sistem keamanan jaringan snort karena selain *server*, sistem juga membutuhkan PC Snort. Sedangkan pada sistem keamanan jaringan *netfilter* hanya menggunakan *server* dimana *web service* dan *netfilter* dijalankan.

Tabel 1 Perbandingan Kebutuhan Perangkat Lunak Sistem

No	Kebutuhan Sistem	Sistem Keamanan Jaringan	
		Snort	*Netfilter*
1	Komponen Sistem	➢ Apache2 ➢ PHP5 ➢ Libmysql-5.6.31 ➢ Snort 2.9.8.2 ➢ Daq-2.0.6 ➢ Barnyard2-2-1.14-336 ➢ Pulledpork-0.7.2-194 ➢ Snorby-2.6.2	➢ APF ➢ *Mod Evasive*
2	Tahap Konfigurasi Sistem	➢ Konfigurasi PC Snort ➢ *Update* dan *upgrade* Kernel ➢ Instalasi paket pendukung snort ➢ Konfigurasi Snort ➢ Konfigurasi directori *rules* snort ➢ Konfigurasi Banryard2 ➢ Konfigurasi PulledPork ➢ Konfigurasi SystemD ➢ konfigurasi Snorby ➢ Konfiurasi Snort mode *inline*	➢ *Update* dan *upgrade* kernel ➢ Iinstalasi paket APF ➢ Konfigurasi APF ➢ Instalasi *mod evasive* ➢ Konfigurasi *mod evasive*

2. Berdasarkan pengamatan pada pengujian sistem snort dan *netfitler* dapat diketahui bahwa penggunaan *memory* pada *Server* Snort lebih sedikit dari pada *Server Netfilter*. Oleh karena itu snort lebih baik digunakan dalam mengoptimalkan *memory* pada *server*. Penggunaan *memory* sistem keamanan jaringan snort dan *netfilter* ditampilkan pada tabel 2.

Tabel 2 Penggunaan *Memory* Sistem Keamanan

No	Nama Perangkat Keras	Sistem Keamanan Jaringan	
		Snort	*Netfilter*
1	PC Snort	175488 KiB	-
2	*Server*	330668 KiB	457968 KiB

3. Berdasarkan kebutuhan perangkat lunak pada perancangan dan implementasi dapat diketahui bahwa sistem keamanan jaringan snort lebih banyak membutuhkan perangkat lunak dari pada *netfilter*. Kebutuhan perangkat lunak pada sistem terdiri dari komponen sistem dan tahap konfigurasi sistem. Perbandingan kebutuhan perangkat lunak sistem ditampilkan pada tabel 1.
4. Berdasarkan pengujian menggunakan *ping attack,* DoS, dan *port scanning netfilter* memiliki hasil keamanan yang lebih baik dengan persentase rata-rata 63,92% dari pada snort dengan persentase rata-rata 49,83%. Perbandingan hasil pengujian keamanan sistem ditampilkan pada tabel 3.

Tabel 3 Perbandingan Hasil Pengujian Keamanan Sistem

No.	Serangan	Persentase Rata-rata Hasil Pengujian Sistem Keamanan Jaringan	
		Snort	*Netfilter*
1	*Ping Attack*	100,00%	0,00%
2	DoS	49,45%	99,77%
3	*Port Scanning*	0,05%	92,00%

IV. KESIMPULAN/RINGKASAN

Setelah dilakukan analisis perbandingan sistem keamanan jaringan menggunakan snort dan *netfilter* dapat disimpulkan bahwa antara lain sebagai berikut :

1. Perangkat keras yang digunakan oleh *netfilter* yaitu sebuah *Server Netfilter* sedangkan pada snort menggunakan PC Snort dan *Server* Snort.
2. *Server* Snort menggunakan *memory* sebesar 330668 KiB dan PC Snort menggunakan *memory* sebesar 175488 KiB sedangkan *Server Netfilter* menggunakan *memory* lebih besar yaitu 457968 KiB.
3. 2 komponen sistem dan 5 tahap konfigurasi merupakan kebutuhan perangkat lunak pada *netfilter* sedangkan pada snort membutuhkan 8 komponen sistem dan 10 tahap konfigurasi.
4. Persentase pencegahan serangan sebagai berikut :
   - Snort 100,00 % lebih baik saat serangan *ping attack* dari pada *netfilter.*
   - *Netfilter* 50,32 % lebih baik saat serangan DoS dari pada snort.
   - *Netfilter* 91,95 % lebih baik saat serangan *port scanning* dari pada snort..

DAFTAR PUSTAKA

[1] Mutaqin, Asep Fauzi. 2015. *Rancang Bangun Sistem Monitoring Keamanan Jaringan Prodi Teknik Informatika Melalui SMS Alert dengan Snort* (Skripsi). Pontianak: Universitas Tanjungpura.

[2] Diarta, Etana. 2013. *Sistem Monitoring Deteksi Penyusup Dalam Jaringan Komputer Menggunakan Snort Pada Ubuntu 12.04 Berbasis Sms Gateway* (Skripsi). Yogyakarta: AMIKOM.

[3] Pratama, Putu Agus E. 2014. *Handbook Jaringan Komputer*. Bandung: Infomatika.

[4] Bapuji, Yeldi dan M.L. Ravi Chandra. 2013. *Design of a Next Generation Firewall Based on Netfilter. International Journal of Scientific Engineering and Technology Research, Volume. 02, IssueNo.15, Pages:1652-1658.*

[5] Rafiudin, Rahmat. 2010. *Mengganyang Hacker dengan Snort*. Yogyakarta: Andi.

[6] Pribadi, Harijanto. 2007. *Firewall Melindungi Jaringan dari DDoS Menggunakan Linux + Mikrotik*. Yogyakarta: ANDI.

Korespondensi : Muhammad Suyuti Ma'sum