IMPLEMENTASI FIREWALL DAN IDS PADA SMOOTHWALL EXPRESS

I Made Dwi Suryadinata[1] **Setia Juli Irzal Ismail[2]** **Mochamad Fachru Rizal[3]**

[1,2,3] Fakultas Ilmu Terapan - Telkom University
[1]dwi.suryadinata@gmail.com [2]jul@tass.telkomuniversity.ac.id [3]mfrizal@tass.telkomuniversity.ac.id

Abstrak

Penyalahgunaan internet kerap ditemui dalam berbagai bentuk seperti melakukan akses ke situs pornografi bahkan melakukan penyerangan ke situs lainnya yang menyebabkan situs yang dituju mengalami masalah. Penyerangan ini merupakan hal yang sangat merugikan bagi perusahaan karena data yang bersifat penting dapat diambil begitu saja oleh pihak yang tidak bertanggung jawab. Smoothwall Express merupakan suatu sistem operasi berbasis Linux, yang dapat melakukan fungsi *firewall* yang berguna untuk mengamankan jaringan internal. *Filtering web* dan membatasi waktu akses merupakan fitur yang dimiliki sistem operasi ini. Sistem operasi ini dapat diintegrasikan dengan Snort yaitu salah satu *software* yang dapat melakukan pencegahan terhadap serangan (*Intrusion Detection System*) untuk pengamanan terhadap *server*. Smoothwall Express ini akan diimplementasikan pada jaringan yang terhubung dengan satu DMZ dan jaringan internal sehingga DMZ memiliki fitur keamanan dan jaringan internal sendiri juga memiliki fitur yang dapat melakukan pemblokiran terhadap situs tertentu dan pembatasan waktu akses.

Kata kunci: ***Firewall, Intrusion Detection System*, Smoothwall Express**

Abstract

Internet abuse often encountered in various forms such as access to pornography site, attack to other sites, etc. This attack is very harmful for the company because some data can be taken by those who are not responsible. Smoothwall Express is a Linux based operating system, which can perform the firewall function that useful to secure internal network. Web filtering and access control are the features of this operating system. This operating system can be integrated with Snort, a software that can detect against attack for the security on the server. Smoothwall Express will be implemented on a network with one DMZ and internal network. DMZ has security feature and internal network also has feature that able to block the specific site and restrictions on access time.

Keywords: ***Firewall, Intrusion Detection System*, Smoothwall Express**

1. Latar Belakang

Internet merupakan hal yang vital bagi masyarakat saat ini, terutama masyarakat yang bergelut dengan teknologi informasi. Dengan adanya internet, komunikasi baik yang jarak jauh maupun dekat dapat dilakukan hanya dengan suatu perangkat yang mendukung jaringan internet. Selain melakukan komunikasi, informasi juga mudah didapat dengan mengandalkan teknologi ini. Dengan kemudahan ini, tidak jarang beberapa pengguna menyalahgunakan teknologi internet dengan berbagai hal. Seperti mengakses situs - situs yang berbau pornografi, melakukan kejahatan di dunia maya (*cybercrime*), bahkan melakukan serangan terhadap informasi – informasi yang sudah ada seperti penyerangan *database*, *web*, dan data – data penting lainnya.

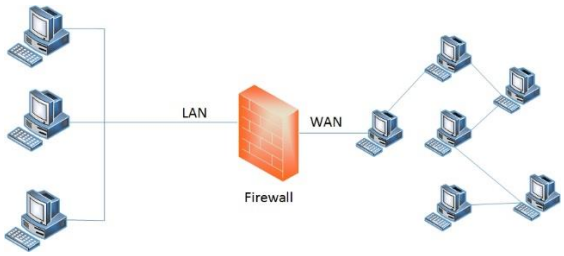
Untuk menghindari dampak negatif tersebut, diperlukan perlindungan terhadap jaringan komputer agar pengguna tidak menyalahgunakan dan merasa aman pada jaringan yang dimilikinya. *Firewall* merupakan fitur keamanan pada jaringan komputer yang dapat mengatur masuk keluarnya paket data. Pada jaringan yang sederhana, *firewall* biasanya hanya diimplementasikan pada komputer yang bersangkutan. Padahal belum tentu *firewall* pada komputer mampu memproteksi keamanan paket data yang diakses oleh penggunanya.

Oleh karena itu, sebaiknya dipersiapkan suatu *firewall* yang mampu memproteksi pengguna dari akses internal maupun eksternal. Smoothwall Express merupakan sistem operasi berbasis Linux yang memiliki fitur *firewall* yang mampu membatasi akses yang masuk maupun keluar pada suatu jaringan. Smoothwall Express memiliki fitur *web filtering* dan memiliki IDS yang terintegrasi langsung pada sistem operasinya. Dengan demikian penulis akan mengangkat judul Implementasi *Firewall* dan IDS pada Smoothwall Express. Implementasi ini rencananya akan digunakan pada jaringan yang memiliki topologi satu *firewall* dan satu DMZ, dengan pengujian *URL filter*, pembatasan waktu akses dan menampilkan hasil *log* dari serangan DDoS, *sniffing*, *port scanning*.

2. Dasar Teori

2.1 Firewall

Firewall [1] adalah suatu aturan yang diterapkan baik terhadap *hardware*, *software* ataupun sistem itu sendiri dengan tujuan untuk melindungi, baik dengan melakukan filterisasi, membatasi ataupun menolak suatu koneksi pada jaringan yang dilindunginya dengan jaringan luar lainnya seperti internet. *Firewall* bekerja di *layer network* pada OSI *Layer*.

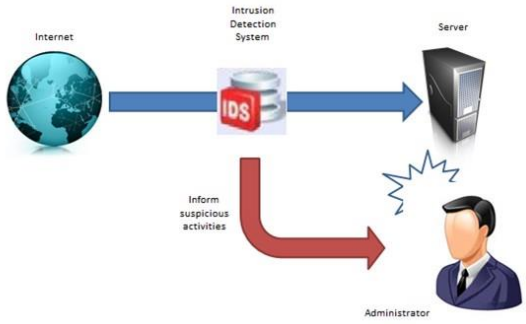


Gambar 1 Skema *Firewall*

Fitur yang terdapat pada *firewall* [4] :

1. Memblokir lalu lintas data yang masuk baik dari sumber maupun tujuan.
2. Memblokir lalu lintas data yang keluar baik darisumber maupun tujuan.
3. Memblokir lalu lintas data dari konten yang diakses.
4. Membolehkan adanya komunikasi data ke jaringan internal.
5. Melaporkan lalu lintas data dan aktifitas *firewall*.

2.2 IDS (Intrusion Detection System)



Gambar 2 Skema IDS Secara Umum [5]

Intrusion detection system (IDS) [2] adalah suatu teknik atau metode yang digunakan untuk mendeteksi ancaman yang mencurigakan pada jaringan. IDS memiliki dua kategori dasar yaitu *signature-based* dan *anomaly detection system*.

Pada *signature-based*, deteksi dilakukan dengan cara melakukan perbandingan dengan *rules* yang sudah ada. Pada *anomaly detection*, deteksi dilakukan berdasarkan kondisi yang tidak normal dari biasanya. Contohnya keadaan *traffic* normal 2 mbps, tiba - tiba menjadi 5 mbps, maka IDS berdasarkan *anomaly detection* akan menganggap adanya serangan.

2.3 Smoothwall

Smoothwall [3] merupakan suatu distro Linux yang dirancang untuk digunakan sebagai *firewall*. Smoothwall dikonfigurasi melalui GUI berbasis *web*. Sistem operasi ini fokus pada konten *web filtering*. *Web filtering* pada sistem operasi ini mencakup blok berdasarkan kategori, URL, dan *string*. Selain keamanan berupa *firewall*, terdapat fitur IDS yang terintegrasi dengan Snort yaitu aplikasi untuk mendeteksi ancaman pada jaringan.

2.4 Web Filtering

Web filtering merupakan suatu perangkat lunak yang dapat membatasi konten pada *website* yang dikunjungi oleh penggunanya. Fitur ini bekerja dengan cara membandingkan aturan dengan permintaan dari pengguna. Jika pengguna melakukan permintaan berupa membuka situs yang yang sebelumnya telah didaftarkan sebagai *blacklist* oleh *admin*, maka pengguna tidak dapat mengakses situs tersebut dan mendapatkan peringatan dari *admin*. *Web filtering* pada sistem operasi Smoothwall Express dapat di*update* secara otomatis pada situs Shala Secure Service.

2.5 DDoS (Distribute Denial of Services)

DDoS (*Distribute Denial of Services*) merupakan serangan untuk melumpuhkan sebuah layanan dengan cara menghabiskan sumber daya yang diperlukan sistem komputer untuk melakukan kegiatan normalnya. Serangan ini awalnya melakukan *scanning* terhadap celah yang ada. Setelah mendapatkan celah keamanan, maka serangan akan dilakukan. Beberapa cara melakukan penyerangan diantaranya *traffic flooding* yaitu membanjiri lalu lintas data, *request flooding* yaitu membanjiri jaringan dengan banyak *request*, dan melakukan perubahan sistem terhadap konfigurasi yang sudah disediakan. Jenis serangan DoS diantaranya [6] :

1. *Ping of death*. Serangan ini dilakukan dengan *tools* ping yang biasanya digunakan untuk memeriksa keberadaan sebuah *network*. Data yang dikirimkan pada ping secara *default* adalah 32 *bytes*, sedangkan pada *ping of death* mampu mengirimkan data hingga 65 kb.
2. SYN Flood. Dilakukan dengan cara memanfaatkan kelemahan protokol pada saat terjadinya proses *handshake*.
3. *Remote controlled attack*. Mengendalikan beberapa jaringan untuk menyerang target.

4. UDP *Flood*. Memanfaatkan protokol UDP yang bersifat *connectionless* untuk menyerang target.
5. SYN *Attack*. Penyerangan dengan memanfaatkan ICMP *echo request* yang sering digunakan pada saat melakukan *broadcast* identitas kepada *broadcast address* dalam sebuah jaringan

2.6 Sniffing

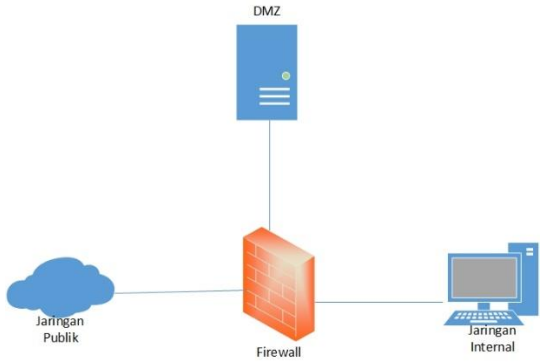
Snifing adalah kegiatan penyadapan pada lalu lintas data di jaringan komputer. Serangan ini dilakukan dengan cara melakukan *scanning* berupa IP dan MAC *address* pada jaringan yang tersambung, kemudian melakukan ARP *poisoning* dimana komputer target akan mengirimkan informasi yang diakses ke penyerang. *Sniffing* dibagi menjadi dua bagian yaitu :

1. *Passive sniffing* adalah kegiatan penyadapan tanpa merubah data atau paket apapun di jaringan.
2. *Active sniffing* adalah kegiatan *sniffing* yang dapat melakukan perubahan paket data dalam jaringan agar bisa melakukan *sniffing*.

2.7 Port Scanning

Port scanning [4] merupakan proses mencari tahu *port* yang terbuka pada *host* tertentu atau semua *host* pada jaringan. Langkah pertama pada penyerangan ini yaitu untuk mengetahui layanan apa saja yang sedang berjalan. Setelah mengetahui layanan yang berjalan, proses penyerangan lain akan dilakukan pada celah keamanan tersebut. Serangan ini melibatkan protokol TCP untuk mengetahui apakah ada respon dari layanan yang tersedia.

2.8 DMZ (Demilitarized Zone)



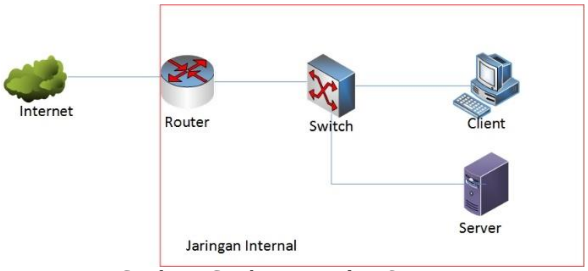
Gambar 3 Topologi DMZ Sederhana

DMZ [7] merupakan mekanisme pengamanan untuk jaringan internal dimana *server* dipisahkan dari jaringan internal. *Firewall* diimplementasikan pada topologi ini untuk memberikan pembatasan baik pada jaringan internal dan *server* itu sendiri, sehingga tidak sembarang pengguna dapat melakukan modifikasi *server* dan pengguna hanya dapat mengakses layanan yang dibuka oleh *server*. DMZ dapat digunakan pada jaringan yang menggunakan *router*.

3. Analisis dan Perancangan

3.1 Gambaran Sistem Saat Ini

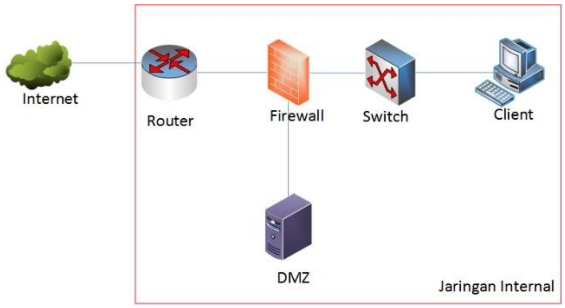
Pada jaringan sederhana, *firewall* hanya bergantung pada komputer yang bersangkutan, selain itu *server* dan juga jaringan internal dimuat dalam satu jaringan yang sama. Topologi ini terdapat beberapa kelemahan, diantaranya *client* yang ada pada suatu jaringan dengan *server* bisa saja menyerang secara langsung pada layanan *server*, pengguna jaringan publik juga dapat mengakses *server* dengan mudah sehingga tidak menutup kemungkinan untuk pengguna dapat menyerang *server*. Tanpa adanya *firewall client* dapat mengakses semua konten yang ada di internet tanpa proses *filter* terlebih dahulu, sehingga bisa saja pengguna mendapatkan *malware* atau *virus* pada komputernya.



Gambar 4 Gambaran Topologi Saat Ini

3.2 Sistem Usulan

Usulan sistem yang dapat dirancang dari gambaran topologi diatas adalah dengan menambahkan *firewall* pada jaringan sehingga pengguna jaringan publik maupun internal tidak dapat mengakses *website* yang telah diblokir oleh *admin* yang mengatur *firewall*. Selain itu *server* dan jaringan internal dipisahkan sehingga *server* dapat terhindar dari ancaman yang dilakukan oleh penyerang pada jaringan publik.



Gambar 5 Rancangan Usulan Topologi

3.3 Kebutuhan Perangkat Keras

Dalam pengerjaan penelitian ini, digunakan perangkat keras dengan spesifikasi sebagai berikut :

Tabel 1 Kebutuhan Perangkat Keras

Jenis	Jumlah	Keterangan
Server	1	Intel Corei3; 512 MB DDR3; 8GB HDD; *Broadcom* 802.11n *Network Adapter*
Client	1	Intel Corei3; 1 GB DDR3; 12GB HDD; *Broadcom* 802.11n *Network Adapter*
Firewall	1	Intel Corei3; 1 GB DDR3; 8GB HDD; *Broadcom* 802.11n *Network Adapter*
Penyerang	1	Intel Corei3; 1 GB DDR3; 16GB HDD; *Broadcom* 802.11n *Network Adapter*

Keempat perangkat yang berada pada Tabel 1 akan dibuat dalam bentuk *virtual* dengan bantuan aplikasi VM VirtualBox.

3.4 Kebutuhan Perangkat Lunak

Berikut merupakan spesifikasi perangkat lunak yang digunakan :

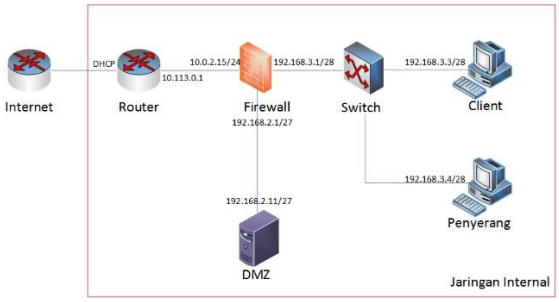
Tabel 2 Kebutuhan Perangkat Lunak

Jenis	Versi	Keterangan
Oracle VM Virtualbox	4.3	Aplikasi Virtualisasi sistem operasi
Smoothwall Express	3.1	*Firewall* yang digunakan
Ubuntu Server	14.04	Sistem Operasi DMZ
Windows	7	Sistem Operasi *Client*
Kali Linux	1.0	Sistem Operasi Penyerang
Cain & Abel	4.9	Aplikasi *Sniffing*
Nmap	6.4	Aplikasi *port scanning*
Hping3	1.0	Aplikasi DDoS *attack*

3.5 Langkah Pengerjaan

Adapun tahap pengerjaan penelitian ini diantaranya :

1. Melakukan konfigurasi jaringan sesuai dengan topologi pada Gambar 6.
2. Melakukan instalasi *firewall* dengan Smoothwall Express.
3. Mengakses jaringan baik internal maupun eksternal dari *client*.
4. Mengintegrasikan IDS pada Smoothwall Express.
5. Melakukan konfigurasi *web filtering* dan pembatasan waktu akses situs pada *firewall*.
6. Melakukan simulasi serangan DDoS, *sniffing* dan *port scanning* pada *firewall*.
7. Melakukan dokumentasi terhadap konfigurasi *web filtering*, pembatasan waktu akses situs dan serangan DDoS, *sniffing* dan *port scanning*.



Gambar 6 Topologi Rencana Implementasi

3.6 Rencana Pengujian

Pengujian yang dilakukan meliputi :

1. Dilakukannya ping antar jaringan internal sehingga terhubung satu dengan yang lain.
2. Penyerang dapat melakukan ping ke *server*.
3. Akses internet secara bebas digunakan oleh *client* pada jaringan internal.
4. Dilakukannya pembatasan pada beberapa situs serta pembatasan waktu akses pada situs - situs tertentu.
5. Dilakukannya ancaman terhadap *server* berupa DDoS yang berupa ICMP *Flooding*, *Sniffing* untuk mendapatkan *password* pada saat pengguna melakukan autentikasi pada *web* dengan menggunakan aplikasi cain and abel, dan melakukan *port scanning* oleh penyerang.
6. Hasil berupa *report* dari snort dapat terdeteksi pada *firewall* Smoothwall Express.

4. Pengujian

Pada tahap ini, pengujian dilakukan menjadi tiga tahap, yaitu pengujian konektifitas antar jaringan, pengujian *firewall* (URL *filter* dan pembatasan waktu akses) dan pengujian terhadap ancaman yang diatasi oleh snort [8].

4.1 Pengujian Firewall

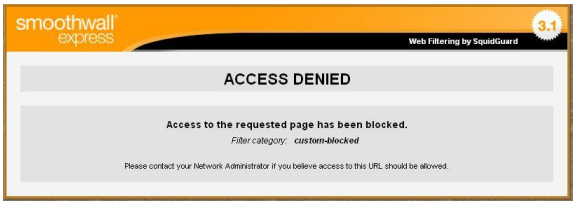
Pengujian *firewall* dibagi menjadi dua tahap yaitu pengujian pada URL *filter* dan pengujian pada pembatasan waktu akses.

4.1.1 URL Filter

Pada pengujian ini, URL *filter* diinput berdasarkan nama *domain*. Dimana admin harus menambahkan daftar *domain* yang akan diblok oleh *firewall*. Berikut merupakan hasil pengujiannya :
Sebagai contoh, pada penelitian ini URL detik.com akan diblokir.

Gambar 7 Daftar URL yang di Blok oleh URL *Filter*

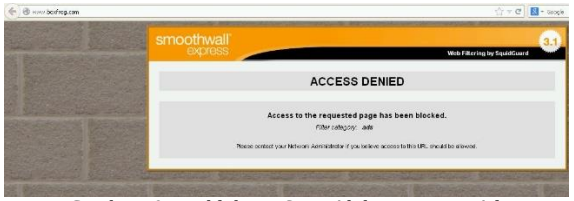
Gambar 8 merupakan tampilan dari URL *filter* tehadap situs yang diblokir.



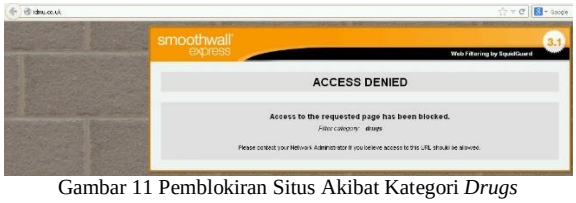
Gambar 8 Tampilan Domain yang Diblokir

Selain filtering berdasarkan alamat yang diinput oleh *admin*, terdapat juga *filter* URL berdasarkan kategori *blacklist* yang telah disediakan Smoothwall Express. Kategori *blacklist* tersebut diambil dari *website* Shalla Secure Service. Gambar 9 merupakan kategori yang dapat digunakan.
File yang berisikan domain dari *web* yang diblokir terdapat pada /var/smoothwall/urlfilter/blacklist/ . *File* konfigurasi bernama *domains*. Gambar 10 dan 11 merupakan hasil dari pemblokiran situs yang dilakukan Smoothwall Express sesuai dengan kategorinya.

Gambar 7 Kategori yang Dapat Biblokir.



Gambar 10 Pemblokiran Situs Akibat Kategori *Ads*

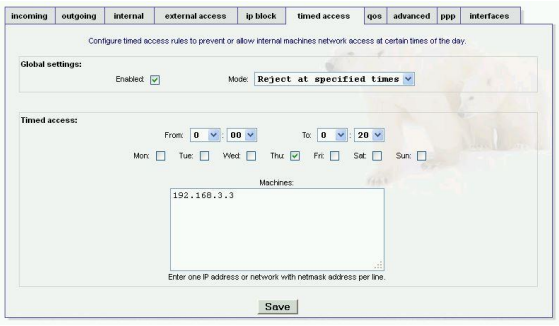


Gambar 11 Pemblokiran Situs Akibat Kategori *Drugs*

4.1.2 Pembatasan Waktu Akses

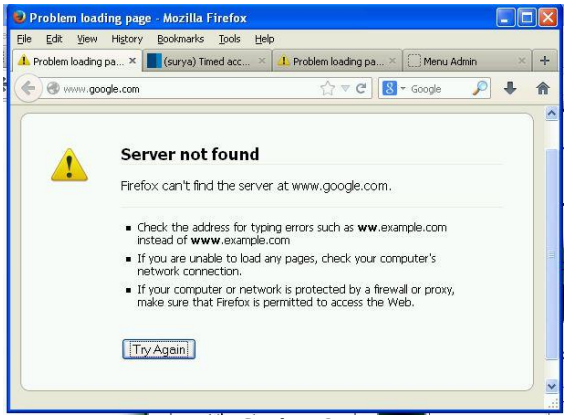
Pembatasan waktu akses memungkinkan *admin* untuk membatasi secara menyeluruh aktivitas internet. Pembatasan ini dapat dilakukan berdasarkan IP maupun *subnet*.

Sebagai contoh, pada penelitian ini, IP 192.168.3.3 akan diblok aksesnya pada jam 0.00 hingga 0.20 pada hari Kamis.

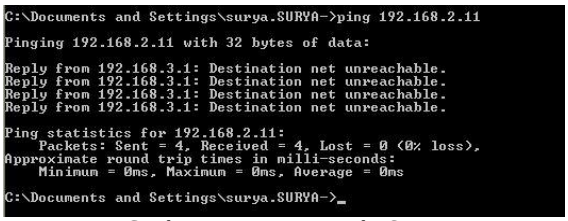


Gambar 12 Konfigurasi Pembatasan Waktu Akses

Jika konfigurasi telah dilakukan oleh admin, maka IP yang bersangkutan tidak dapat mengakses internet maupun berhubungan dengan jaringan internal, seperti yang terlihat pada Gambar 13 dan Gambar 14.



Gambar 13
Tampilan Pembatasan Waktu Akses pada Website.



Gambar 14 Pengujian Ping ke *Server*.

4.2 Pengujian Terhadap Ancaman

Pengujian ini dilakukan menjadi tiga tahap. Yaitu pengujian terhadap serangan ICMP *flooding*, *Scanning*, dan *Sniffing*. Pendeteksian serangan dilakukan dengan memanfaatkan fitur snort pada sistem operasi ini.

4.2.1 Pengujian ICMP Flooding

Penyerangan ICMP *Flooding* dilakukan dengan bantuan aplikasi hping3. Penyerangan dilakukan dengan perintah berikut :

```
root@surya:~#hping3 --icmp -c 10000 -i u100
192.168.2.11
```

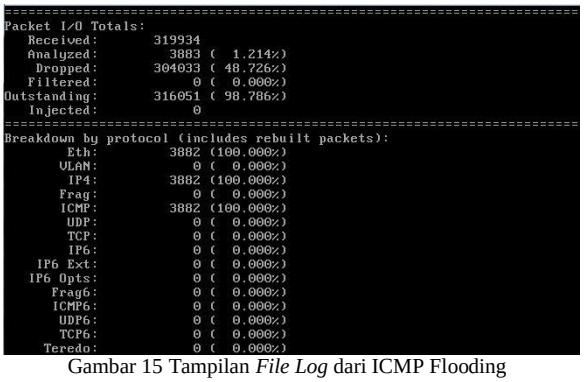
Pada Smoothwall Express mode CLI, lakukan proses *logging* terhadap paket yang berjalan dengan perintah :

```
[root@surya ~]#snort -I eth2 -dev -l
```

Setelah proses *flooding* yang berlangsung dalam beberapa saat, *file log* dari penyerangan ini disimpan dalam *file* snort.log.1433061973. Untuk membuka *file* ini lakukan dengan perintah :

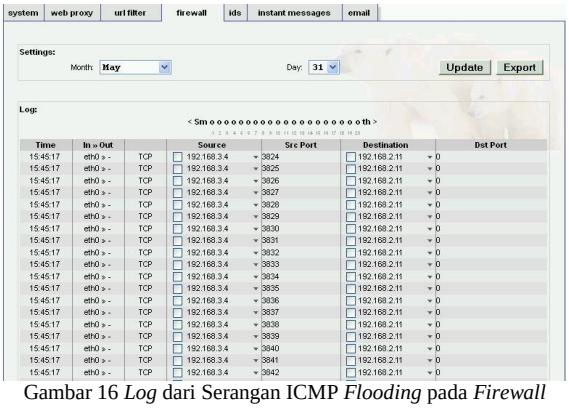
```
[root@surya ~]#snort -dev -r
snort.log.1433061973
```

Pada akhir *log*, terdapat sebuah *report* yang menandakan bahwa komunikasi yang berlangsung kebanyakan adalah ICMP.



Gambar 15 Tampilan *File Log* dari ICMP Flooding

Gambar 16 dan 17 merupakan *log* yang terdapat pada *firewall* dan IDS di sistem operasi Smoothwall Express yang diakses melalui GUI.



Gambar 16 *Log* dari Serangan ICMP *Flooding* pada *Firewall*

Gambar 17 *Log* dari Serangan ICMP *Flooding* pada IDS

Serangan ICMP *flooding* telah berhasil dideteksi oleh *firewall* dan IDS. IP penyerang terdeteksi sebagai 192.168.3.4 dengan *range port* 3824 - 3842 dengan tujuan 192.168.2.11. Hanya saja IDS belum berhasil mengkategorikan serangan secara detail dikarenakan *rules* pada snort sejak tahun 2015 tidak bersifat *open source* lagi.

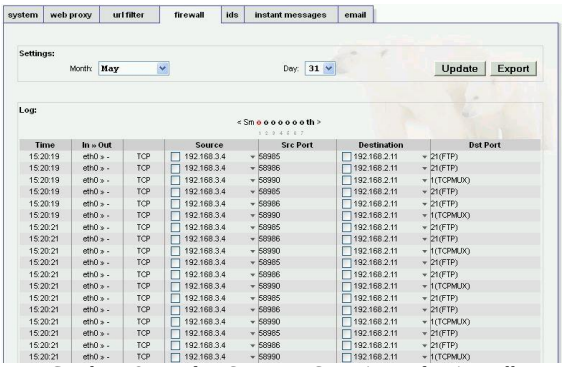
Pada pengujian sebelumnya, serangan ICMP *flooding* tidak berhasil dideteksi. Setelah penambahan *rule* icmp dan info-icmp yang terletak pada /var/smoothwall/snort/rules, maka Smoothwall dapat mendeteksi serangan. Akan tetapi serangan yang dideteksi tidak akurat baik dalam waktu penyerang maupun alamat IP penyerang. Hal ini dapat terjadi dikarenakan fitur snort yang memiliki *rules* yang tidak sesuai.

4.2.2 Pengujian Scanning

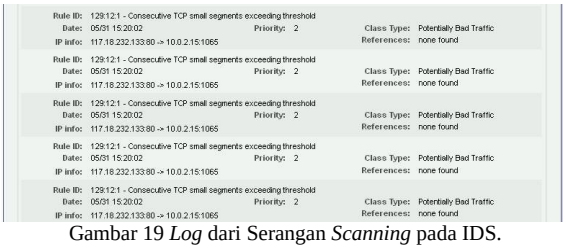
Pengujian *scanning* dilakukan dengan bantuan aplikasi nmap. Pengujian dilakukan dengan perintah berikut pada penyerang :

```
root@surya:~#nmap -v -A 192.168.2.11
```

Gambar 18 dan 19 merupakan hasil *log* dari Smoothwall Express.



Gambar 18 *Log* dari Serangan *Scanning* pada *Firewall*.



Gambar 19 *Log* dari Serangan *Scanning* pada IDS.

Serangan *Scanning* telah berhasil dideteksi oleh *firewall* dan IDS. IP penyerang terdeteksi sebagai 192.168.3.4 dengan tujuan 192.168.2.11. Penyerangan ini melibatkan *port* 1 dan 21 pada komputer target. Hanya saja IDS belum berhasil mengkategorikan serangan secara detail dikarenakan *rules* pada snort sejak tahun 2015 tidak bersifat *open source* lagi.

Pada pengujian sebelumnya, serangan *Port Scanning* tidak berhasil dideteksi. Setelah penambahan *rule scan* yang terletak pada /var/smoothwall/snort/rules, maka Smoothwall dapat mendeteksi serangan. Akan tetapi serangan yang dideteksi tidak akurat baik dalam waktu penyerang maupun alamat IP penyerang. Hal ini dapat terjadi dikarenakan fitur snort yang memiliki *rules* yang tidak sesuai.

4.2.3 Pengujian Sniffing

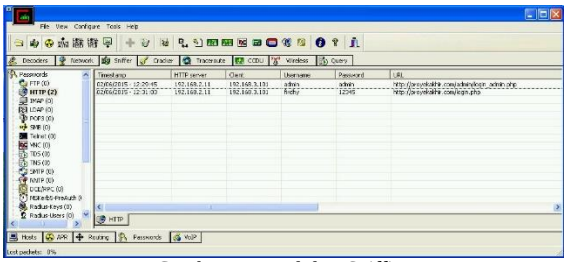
Pengujian *sniffing* dilakukan dengan bantuan aplikasi Cain and Abel. Penyerang pada pengujian *sniffing* ini berada pada satu jaringan pada satu subnet.

1. Buka aplikasi Cain and Abel, kemudian klik *start*/*stop sniffer*.
2. Lakukan *mac scanning* untuk mengetahui komputer yang terhubung dalam jaringan pada satu subnet.
3. Pada menu ARP, tambahkan IP yang akan di *sniffing*.
4. Klik *start*/*stop* ARP untuk memulai proses *sniffing*.
5. Masuk ke *web* proyekakhir.com.
6. Lakukan *login* pada halaman awal seperti pada Gambar 20

Gambar 20 Tampilan *Login Web*.

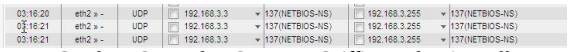
Gambar 21 Tampilan *Member* yang Berhasil *Login*.

7. Pada aplikasi cain and abel, akan terlihat *username* dan *password* dari aktivitas yang dilakukan pada satu jaringan yang mengakses proyekakhir.com. Gambar 22 merupakan hasil dari *sniffing*.



Gambar 22 Hasil dari *Sniffing*.

Serangan *sniffing* berhasil dilakukan. Pada *log firewall*, aktivitas proses IP dan MAC *scanning* dari aplikasi cain and abel terdefinisikan bahwa penyerang melakukan *broadcast* untuk mengetahui pengguna yang tersambung pada jaringan seperti Gambar 23. Namun pada *log* IDS tidak terlihat adanya ancaman yang masuk dikarenakan *rules* pada snort sejak tahun 2015 tidak bersifat *open source* lagi.



Gambar 23 *Log* dari Serangan *Sniffing* pada *Firewall*

5. Kesimpulan dan Saran

5.1 Kesimpulan

Dari pengujian, dapat ditarik kesimpulan :

1. Fungsi *firewall* yang diujikan berupa URL *filter* dan pembatasan waktu akses berhasil diimplementasikan.
2. Snort berhasil diintegrasikan pada Smoothwall Express dengan cara memasukkan *oinkcode* dan men*download rule* dari snort.
3. Ancaman berupa DDOS, *Scanning* dan *Sniffing* berhasil dilakukan. *Log* pada *firewall* berjalan dengan baik. Namun pada *log* IDS, tidak menampilkan peringatan (*alert*) dikarenakan snort pada Smoothwall

Express tidak dapat mengkategorikan jenis serangan yang masuk.

4. Smoothwall Express hanya dapat mendeteksi pada aktivitas *firewall*, sedangkan pada IDS sistem operasi ini tidak dapat mendeteksi serangan yang masuk dikarenakan kita harus membeli lisensi "*rules* snort" dari snort.

5.2 Saran

Saran yang dapat penulis berikan pada penelitian ini diantaranya :

1. Untuk *firewall* terutama pada pembatasan waktu akses, disarankan untuk mengunduh *file* modifikasi dari Smoothwall Express yaitu FFC (*Full Firewall Control*). Pada FFC pembatasan waktu akses dapat diatur dalam jangka waktu per hari, perminggu maupun perbulan.
2. Disarankan menggunakan fitur IDS dari aplikasi pihak ke-tiga pada DMZ, karena IDS pada sistem operasi ini memiliki *rules* yang tidak lengkap sehingga tidak ada *log* IDS yang menyebabkan tidak adanya peringatan jika ada ancaman yang masuk.

DAFTAR PUSTAKA

[1] R. Farunuddin, Membangun Firewall dengan IPTables di LINUX, Jarkarta: PT. Elex Media Komputindo, 2005.

[2] Brian Komar, Ronald Beekelaar, Joern Wettern PhD, Firewalls for Dummies, New York: Wiley Publishing, 2003.

[3] Wikipedia, "Wikipedia," 25 January 2015. [Online]. Available: http://en.wikipedia.org/wiki/SmoothWall. [Accessed 6 February 2015].

[4] R. U. Rehman, Intrusion Detection Systems with Snort, New Jersey: Prentice Hall PTR, 2003.

[5] C. Ng, "IDS: A Technical Understanding," 20 December 2012. [Online]. Available: http://www.pkfavantedge.com/technology/ids-a-technical-understanding/. [Accessed 13 Februari 2015].

[6] W. Komputer, Tutorial 5 Hari : Belajar Hacking dari Nol, Yogyakarta: Penerbit Andi, 2010.

[7] W. Stallings, Network Security Essentials:Applications and Standards 4th Edition, Prentice Hall, 2010.

[8] I. M. D. Suryadinata, Implementasi Firewall dan IDS pada Smoothwall Express, Bandung: Telkom University, 2015.