PEMANFAATAN *INTRUSION DETECTION SYSTEM* (IDS) SEBAGAI OTOMATISASI KONFIGURASI *FIREWALL* BERBASIS *WEB SERVICE* MENGGUNAKAN ARSITEKTUR *REPRESENTATIONAL STATE TRANSFER* (REST)

Demmy Nanda Awangga, Haruno Sajati, Yenni Astuti
Teknik Informatika STTA Yogyakarta
Informatika@stta.ac.id

ABSTRACT

Many things can destabilize a computer network connections, both with regard to hardware and software. Therefore, we need a technique for network security, one of them is firewall. The problems that arise in this final project is to build a linux based firewall automation application via web service by using REST (Representational State Transfer) architecture and IDS (Intrusion Detection System). The system buid firewall rules using linux operating system with the help of 2 pieces of IDS to detect the activities of traffic data between the intruder and the server that will be recorded in the IDS database. The system will compare the server with IDS on the router to get the IP address of the actual intruders, so it will be blocked by the firewall. The applications is used to prevents the ping of death attack using web service and REST protocol so that firewall rules will run automatically.

Keywords: Intrusion Detection System, Representational State Transfer, Web Service

1. PENDAHULUAN

Pada kasus pengaplikasian *firewall*, seorang *admin* tidak mungkin melakukan monitoring jaringannya selama 24 jam, hal inilah yang membuat munculnya ide untuk membangun otomatisasi *firewall*, yakni *firewall* bekerja secara otomatis pada sistem, sehingga tidak memerlukan admin untuk mengelola atau bahkan memantau sistem. Pada sistem yang dibangun ini, jika terdapat sebuah serangan maka *firewall* akan menutup secara otomatis dengan memanfaatkan *web service*.

Aplikasi *firewall* yang akan dibangun yaitu berbasis *linux* dengan menggunakan *web service* dan IDS (*Intrusion Detection System*). Dengan adanya data-data yang dikirimkan oleh IDS (*Intrusion Detection System) maka* sistem akan melakukan analisis suatu aksi aturan *firewal* (Imam Cartealy, 2013).

Intrusion Detection System merupakan sebuah aplikasi perangkat lunak yang dapat mendeteksi aktivitas yang mencurigakan dalam sebuah sistem atau jaringan. IDS dapat melakukan inspeksi terhadap lalu lintas *inbound* dan *outbound* dalam sebuah sistem atau jaringan, melakukan analisis dan mencari bukti dari percobaan intrusi (penyusupan) (Deny Rusnanto, 2007).

Web service akan mengecek jaringan-jaringan mana saja yang sering diserang pada IDS IDS secara berkala dan setelah mendapatkan informasi dari IDS, *web service* akan mengirim informasi ke *server* dengan mengunakan REST (*Representational State Transfer*) setelah itu informasi yang dikirim melalui REST (*Representational State Transfer*) akan diterjemahkan ke aturan *firewal server* pada linux.

2. LANDASAN TEORI

2.1 Tinjauan Pustaka

a. *Instrusion Detection System* untuk Membangun Keamanan Jaringan Komputer dengan menggunakan *Snort.*

Materi IDS ini pernah dibahas sebelumnya pada skripsi yang berjudul "*Instrusion Detection System* untuk Membangun Keamanan Jaringan Komputer dengan menggunakan *Snort*" karya Deny Rusnanto pada tahun 2007 dari jurusan Teknik Informatika Sekolah Tinggi Teknologi Adisutjipto Yogyakarta. Skripsi tersebut membahas mengenai sistem arsitektur pada *Intrusion Detections System* dengan sistem operasi *FreeBSD*, *Apache*, *PHP*, *MySQL*, dan *Acid*, juga dijelaskan tentang fungsi *Snort* pada IDS.

b. Penerapan XML *Web Service* pada Sistem Terdistribusi Barang.

Pada penelitian sebelumnya, *web service* pernah dibahas pada sebuah Jurnal karya Hartati Deviana tahun 2011 yang berjudul, "Penerapan XML *Web Service* pada Sistem Terdistribusi Barang" Tulisan tersebut membahas tentang sebuah sistem informasi dengan menggunakan teknologi *Web service* menggunakan PHP dan *NuSOAP* yang diimplementasikan pada sistem pengelolaan distribusi barang.

2.2 *Web Service*

Web Service adalah aplikasi yang *modular*, *self-describing* (deskripsi diri), dan *self-contained* (mengandung informasi yang utuh) yang bisa di-*publish*, ditempatkan pada semua web. Salah satu klasifikasi pada web service yaitu arsitektur REST (*Representational State Transfer*), dimana REST merupakan REST adalah suatu pendekatan untuk mendapatkan informasi isi dari sebuah website dengan membaca halaman web yang ditunjuk yang berisi sebuah *file* XML yang mendeskripsikan dan memasukkan isi yang diinginkan.

2.3 *Firewall Linux*

Firewall merupakan sebuah sistem atau perangkat yang mengijinkan lalu lintas jaringan yang dianggap aman untuk melaluinya dan mencegah lalu lintas jaringan yang tidak aman. Umumnya, sebuah *firewall* diimplementasikan dalam sebuah mesin terdedikasi, yang berjalan pada pintu gerbang (*gateway*) antara jaringan lokal dan jaringan lainnya. *Firewall* umumnya juga digunakan untuk mengontrol akses pengguna pada jaringan pribadi dari pihak luar.

2.4 *Intrusion Detection System* (IDS)

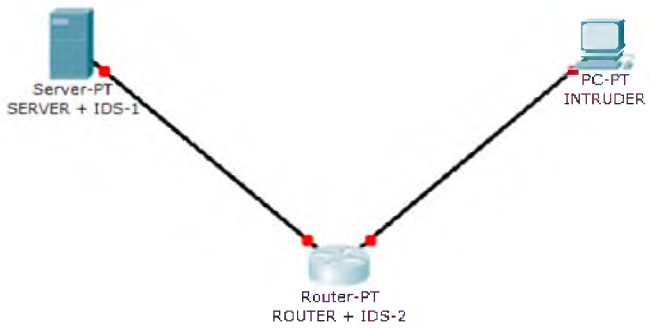
IDS (*Intrusion Detection System*) adalah sebuah sistem yang melakukan pengawasan terhadap trafik jaringan dan pengawasan terhadap kegiatan-kegiatan yang mencurigakan didalam sebuah sistem jaringan. Jika ditemukan kegiatan-kegiatan yang mencurigakan berhubungan dengan trafik jaringan maka *IDS* akan memberikan peringatan kepada sistem atau administrator jaringan. Dalam banyak kasus,IDS juga merespon terhadap trafik yang tidak normal atau anomali melalui aksi pemblokiran seorang pengguna atau alamat IP (*Internet Protocol*).

3. PERANCANGAN ALUR SISTEM

Seorang admin tidak mungkin mengamati jaringan selama 24 jam padahal serangan seorang *intruder* dari jaringan ke server tidak diketahui waktunya. Oleh karena itu, dalam tugas akhir ini dibangun sebuah web *firewall* yang berjalan secara otomatis, sehingga

diharapkan dapat lebih memudahkan pengaturan *firewall* untuk mendeteksi dan menutup serangan-serangan yang diambil oleh IDS.

Konsep yang diterapkan pada sistem ini yaitu jika *intruder* mengirimkan ICMP (Internet Control Message Protocol) atau ping ke *server* lebih dari 4 kali maka *firewall* akan menutup IP *intruder* untuk berhubungan dengan *server* dan IDS akan merekam aktivitas antara *intruder* dan *server*. Aktivitas yang direkam oleh IDS akan dikirm ke *firewall* dengan menggunakan web *service* dan setelah itu diterima oleh protokol REST untuk mengambil keputusan IP mana saja yang akan ditutup agar tidak bisa terhubung dengan *server* atau jaringan yang lainnya. Penutupan IP akan berlaku sampai dengan pukul 01.00 WIB, setelah itu otomatis aturan-aturan *firewall* akan dihapus. Sistem ini akan berjalan otomatis dengan sendirinya tanpa ikut campur *admin*. Konsep sistem ini diberikan pada Gambar 1.



Gambar 1 Konsep Sistem Aplikasi *Web Firewall*

5. IMPLEMENTASI

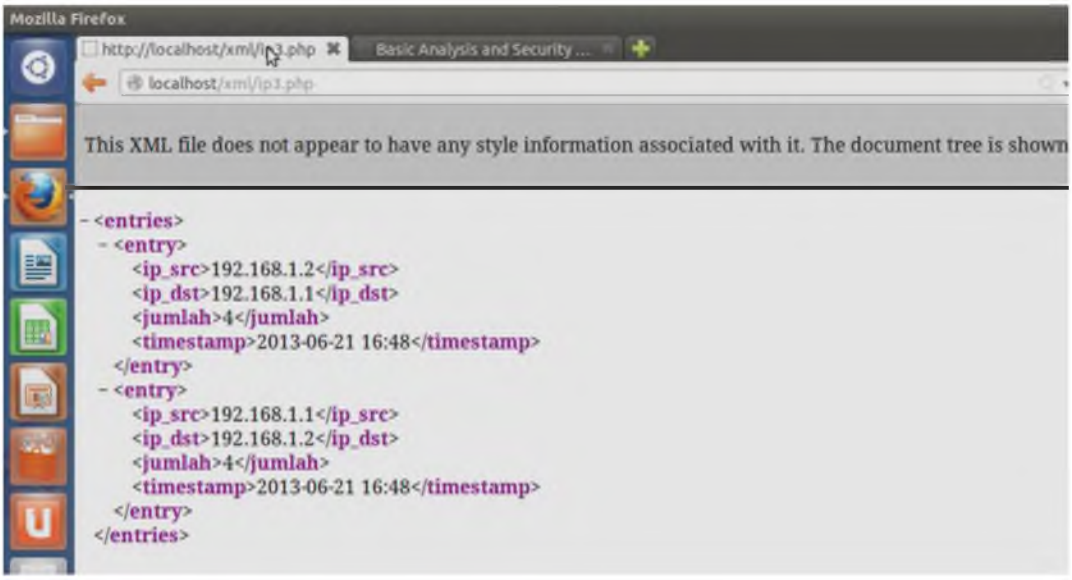
5.1 Skema Pengujian Jaringan

Pengujian jaringan dilakukan dengan menggunakan 3 buah komputer/ laptop dan 1 buah *Switch*, untuk lebih jelasnya akan dijelaskan pada Sub bab 4.8.1. Skema jaringan yang diterapkan pada saat pengujian sistem dapat dilihat seperti pada Gambar 4.23. Pengujian dilakukan dengan satu komputer dan dua buah laptop. Komputer tersebut berfungsi sebagai *intruder* (penyerang). Laptop yang berada ditengah berfungsi sebagai *routing firewall* dan IDS-2, dan laptop yang berada disebelah kanan berfungsi sebagai *server* dan IDS-1.

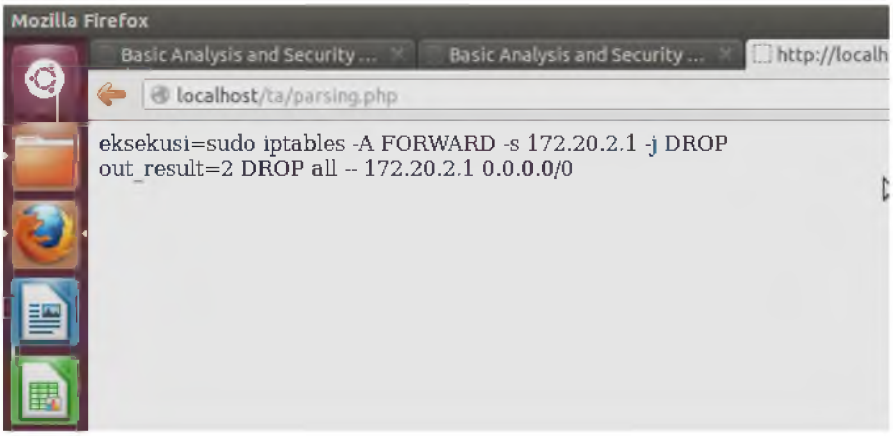
Gambar 2 Skema Pengujian Jaringan

4.2 Komunikasi Protokol

Komunikasi antara server dengan firewall menggunakan xml dan REST.xml mengirimkan alamat-alamat IP ke firewall jika syarat terpenuhi seperti pada gambar 3 setelah itu xml akan di kirim ke firewall di firewall akan di Rest untuk mengabil tindakan IP mana saja yang akan di tutup oleh firewall gambar 4 menujukan IP tersebut di tutup oleh firewall.



Gambar 3 Komunikasi *Server* ke *Firewall*

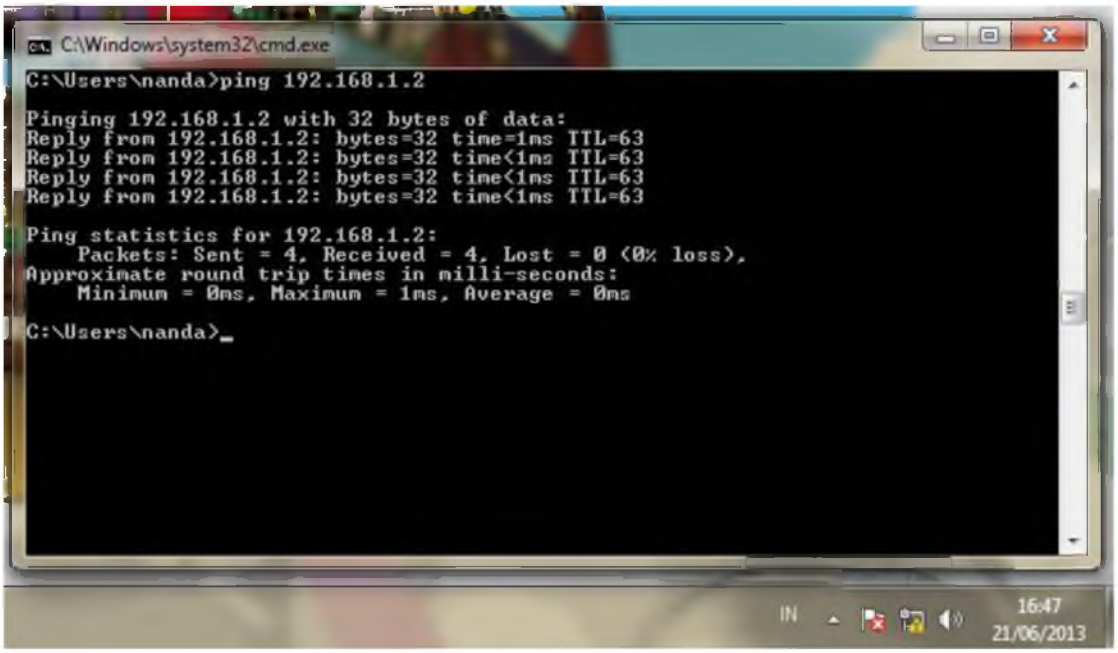


Gambar 4 Eksekusi *Iptables Firewall*

4.3 Pengujian Serangan

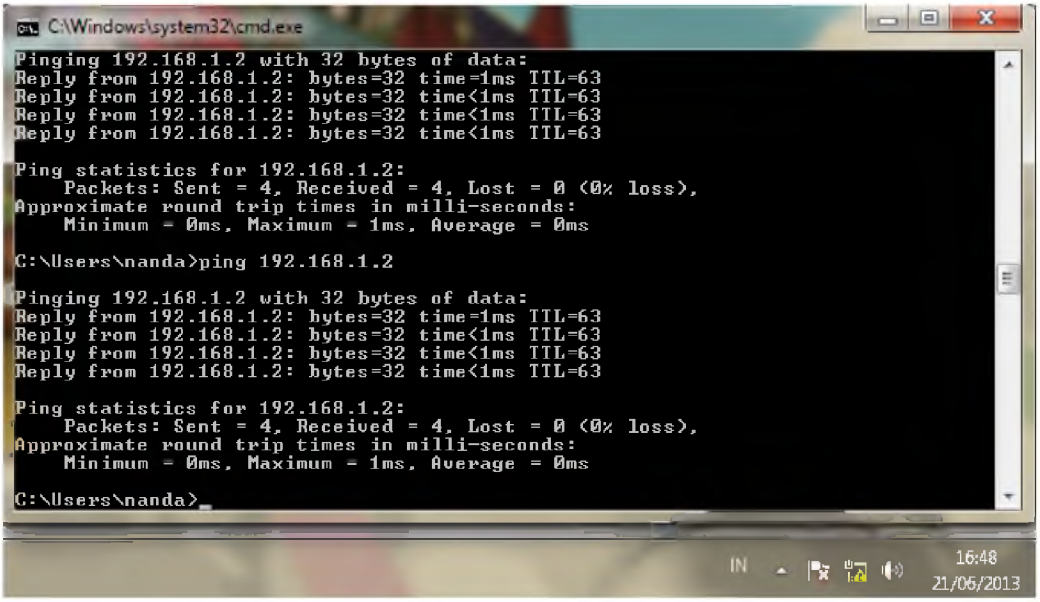
Pengujian sistem ini dilakukan melalui dua tahap, pertama dilakukan pengujian dengan melakukan serangan *ping* sebanyak 4 kali ke *server*, dan untuk pengujian kedua akan dilakukan serangan *ping* sebanyak lebih dari 4 kali ke *server*.

Pengujian pertama, dilakukan serangan *ping* oleh komputer *intruder* sebanyak 4 kali dalam 1 menit yang sama. Hasil serangan *ping* yang tampak pada komputer *intruder* dapat dilihat seperti pada gambar 5. Hasil tersebut menandakan bahwa komputer *intruder* berhasil melakukan serangan *ping* sebanyak 4 kali.



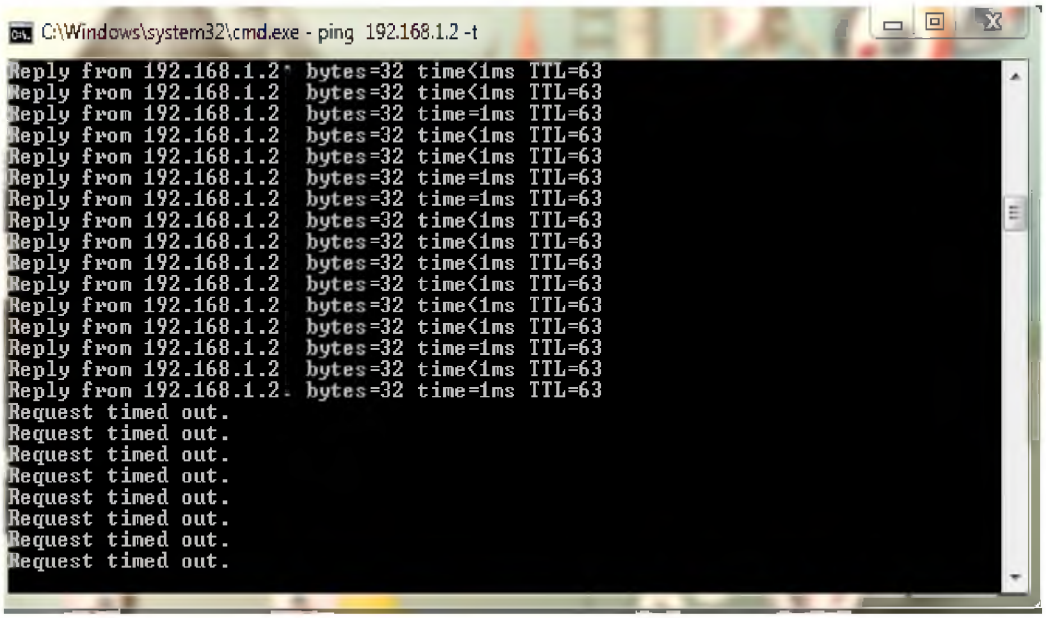
Gambar 5 *Intruder Ping Server*

Selanjutnya, *intruder* kembali melakukan serangan *ping* pada menit berikutnya sebanyak 4 kali seperti terdapat pada Gambar 6. Gambar 6 membuktikan bahwa *intruder* berhasil melakukan *ping* tanpa terkena penutupan oleh *firewall*.



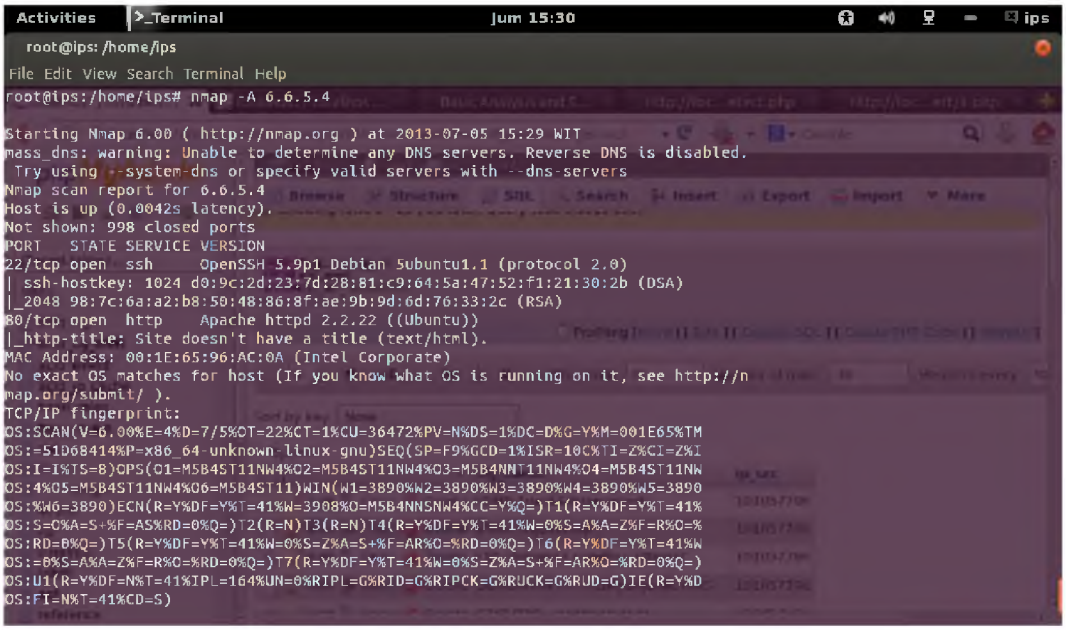
Gambar 6 *Intruder Ping* Menit Berbeda

Pengujian kedua, *intruder* melakukan *ping* terus menerus pada menit yang sama atau melebihi 4 kali dalam 1 menit. Gambar 7 menunjukkan bahwa *intruder* yang melakukan *ping* secara terus menerus terkena penutupan dari *firewall* yang berjalan secara otomatis.



Gambar 7 *Intruder Ping Server* 2

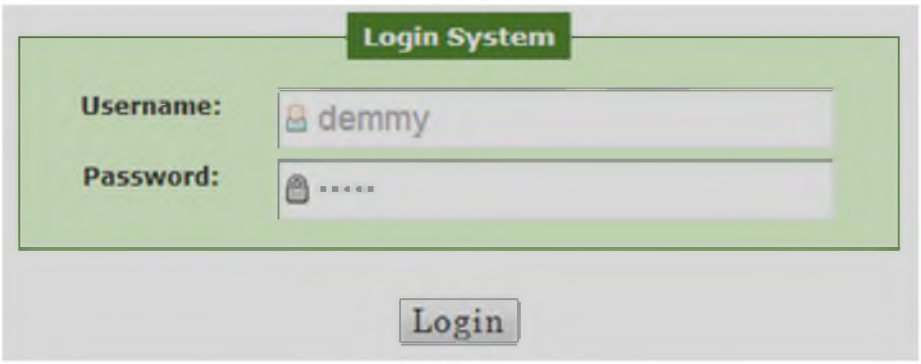
Pengujian ketiga, *intruder* akan melakukan serangan dengan cara *port scanning* yang terbuka di *server* dan secara otomastis IP *intruder* akan ditutup oleh *firewall*. Gambar 8 menunjukan serangan *port scanning* ditutup oleh *firewall*.



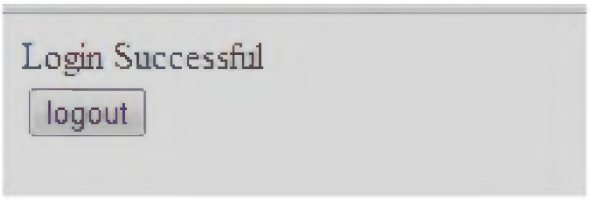
Gambar 8 *Port Scanning*

4.4 Tampilan *Web Login*

Tampilan *web login* ini digunakan untuk pengujian pada saat *login*. Jika terdapat kesalahan dalam menginputkan *user* atau *password* sebanyak 3 kali maka *user* dan *password* akan diblokir selama 24 jam. Gambar 9 menunjukan tampilan halaman web *login*. Selanjutnya, apabila berhasil melakukan *login* maka akan tampil seperti pada Gambar 10.



Gambar 9 *Web Login*

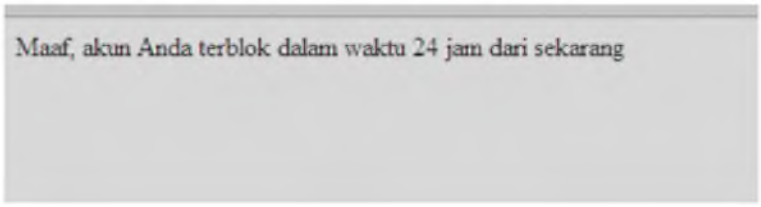


Gambar 10 Sukses *Login*

Gambar 11 menunjukkan bahwa *user* atau *password* yang dimasukkan adalah salah sehingga sistem akan meminta ulang untuk memasukkan *user* atau *password* dengan benar, dimana kesempatan untuk memasukkan *user* dan *password* yaitu sebanyak 3 kali.

Gambar 11 Salah *User* atau *Password*

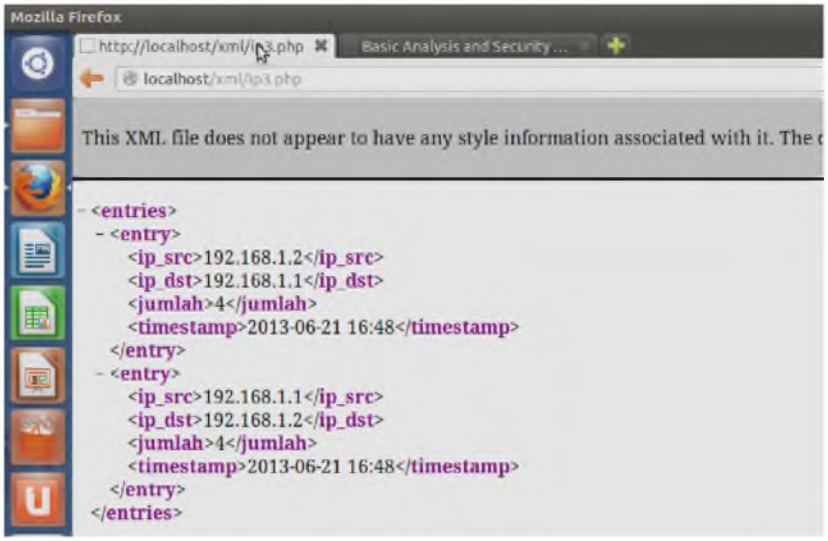
Apabila saat memasukan *user* dan *password* terjadi kesalahan sebanyak 3 kali maka akan muncul tampilan seperti pada gambar 12 yang menunjukkan pemblokiran akun selama 24 jam, jika sudah lebih dari 24 jam maka akun dapat diakses kembali.



Gambar 12 Akun Diblokir

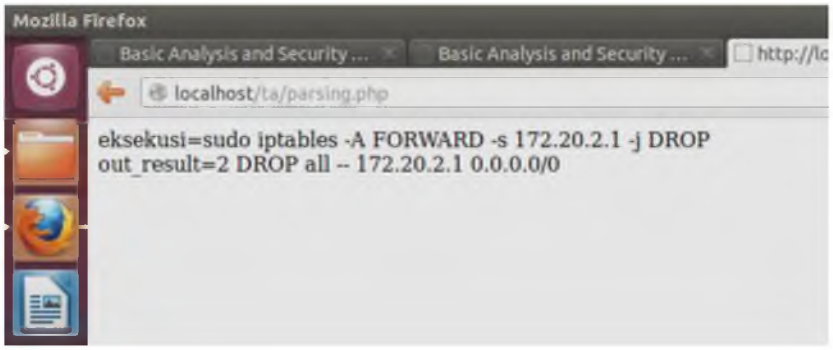
5. ANALISA PENGUJIAN SISTEM

Sistem ini membuktikan bahwa *web service* dengan protokol REST dapat membangun sebuah *firewall* otomatis dan sistem ini dapat menangani lebih dari 1 *server* sehingga setiap *server* dapat dilakukan pengaturan dalam hal jumlah penyerangan untuk dimasukkan ke aturan *firewall*. Aplikasi ini bisa mengeblok IP *intruder* dengan benar karena yang dikirim ke *firewall* dari *server* adalah IP dan *gateway* server bukan IP *intruder* seperti Gambar 13.



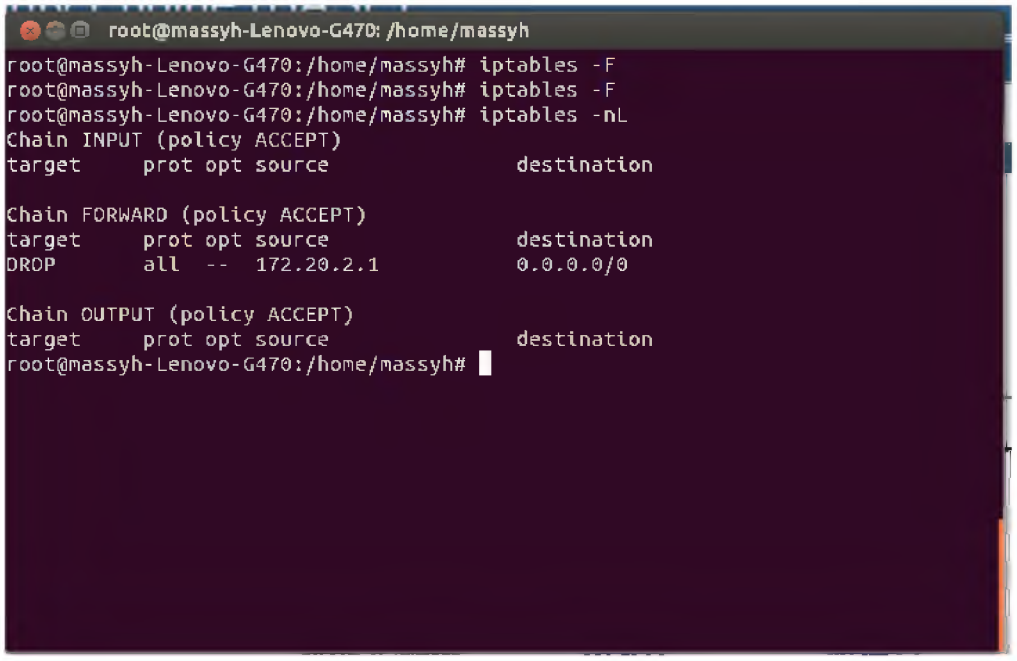
Gambar 13 XML IP dan *Gateway*

Proses sistem tersebut melalui proses *routing* sehingga untuk mengetahui alamat IP *intruder* dengan benar maka *firewall* akan membandingkan data yang dikirim melalui XML dari *server* dengan cara mengambil waktu kejadian yang sama dan dibandingkan lagi dengan IP dan *gateway server* sehingga ditemukan IP *intruder* seperti ditunjukan pada Gambar 14 dan langsung ditutup oleh *firewall*.



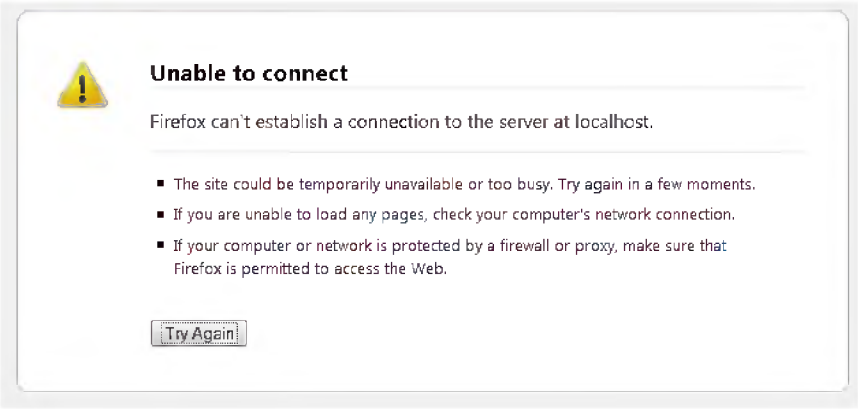
Gambar 14 Eksekusi *Iptables Firewall 2*

Sementara untuk melihat di *firewall* IP mana saja yang sudah ditutup dilakukan dengan cara mengetikkan perintah #*iptables –nL* di cmd linux seperti pada Gambar 15. IP yang ditutup adalah 172.20.2.1 dengan target *drop all* sehingga IP tersebut tidak bisa mengakses ke semua jaringan. Aturan-aturan *firewall* yang sudah ditutup aksesnya akan dihapus pada pukul 01.00 wib sehingga IP-IP yang sudah masuk *list firewall* akan dapat mengakses kembali.



Gambar 15 *List Firewall*

Web login ini menggunakan *security* penutupan IP sehingga jika melakukan kesalahan pada saat memasukan *user* atau *password* sebanyak 3 kali maka IP yang mengakses *web* tersebut akan ditutup selama 24 jam karena dianggap melakukan *brute force password* sehingga IP yang melakukan *brute force* tidak bisa mengakses *web login* seperti pada gambar 16.



Gambar 16 Tampilan Gagal *Login*

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

1. Metode REST dapat digunakan untuk komunikasi antara *firewall* dan *server* dalam bentuk *parsing* XML.
2. Parsing XML dapat digunakan untuk konfigurasi *firewall* yang otomatis.
3. *Firewall* otomatis melakukan proses pemblokiran alamat IP hingga pukul 01.00 WIB untuk jenis serangan *ping of death*, *ilegal login* dan *scan port*.

6.2 Saran

1. Sistem dapat dikembangkan untuk memperoleh sistem yang dapat bekerja secara optimal, khususnya untuk menghadapi dua atau lebih *intruder* dengan penyerangan waktu yang sama.
2. Sistem dapat dikembangkan untuk diterapkan pada sistem operasi selain Ubuntu, misalnya mikrotik, windows dan sebagainya.

DAFTAR PUSTAKA

Anonim, 2010. *Router*. https://help.ubuntu.com/community/Router, diakses pada tanggal 21 April 2013.

Cartealy, Imam. 2013. *Linux Networking*. Jakarta : Jasakom.

Deviana, Hartati. 2011. *Jurnal Generic : Penerapan XML Web Service Pada Sistem Distribusi Barang*. Palembang : Politeknik Negeri Sriwijaya.

Hakim, Lukman. 2008. *Membongkar Trik Rahasia Para Master PHP*. Yogyakarta: Lokomedia

Kristanto, Andri. 2004. *Rekayasa Perangkat Lunak (Konsep Dasar)*. Yogyakarta : Gaya Media.

Rusnanto, Deny. 2007. *Skripsi : Intrusion Detection System Untuk Membangun Keamanan Jaringan Komputer Dengan Menggunakan Snort*. Yogyakarta : STTA.

Sajati, Haruno. 2013. *Instalasi snort snort-mysql dan acidbase di ubuntu 12.04*. http://jati.stta.ac.id/2013/05/instalasi-snort-snort-mysql-dan.html, diakses pada tanggal 21 Mei 2013.

Santoso, Budi 2008. *Analisa perancangan web service untuk sistem informasi universitas*. Yogyakarta: UPN.