PENERAPAN HIDS (*HOST INTRUSION DETECTION SYSTEM*) DALAM MEMBANGUN KONFIGURASI *FIREWALL* SECARA DINAMIK

Alfian Firdaus, Haruno Sajati, Yuliani Indrianingsih
Teknik Informatika STTA Yogyakarta
Informatika@stta.ac.id

ABSTRACT

Development of information technology and computer networks in particular its services on the one hand facilitate the work of human beings, but as the very widespread use of the internet, the security level of digital data has become more vulnerable to exploitation. The problem arises when the attacks happened on a computer network in a relatively fast, so that an administrator must always keep an eye on the computer network security. Limitations of an administrator is exactly what underlies the creation of a system capable of detecting and prevention of these attacks, so the time efficiency can be actually implemented. The system is built to prevent attacks on computer networks such as port scanning, ftp, ssh and telnet brute force. This system will analyze the number of errors that occur on login these services, and makes a decision system if the login error exceeds the tolerance of an administrator. If the number of errors exceeds the tolerance, it will automatically create the firewall rules in a very short duration is one minute, so the system is able to prevent the attacks which happened in a very quick time.

Keyword : IDS, Port Scanning, ftp, ssh and telnet brute force.

1. LATAR BELAKANG MASALAH

Mengingat beban pekerjan seorang *network administrator* yang besar dan luas, sangat tidak mungkin seorang *network administrator* untuk selalu *update* setiap waktu terhadap serangan-serangan baru dan dengan singkat membuat *signature* untuk serangan baru tersebut. Maka muncullah ide bagaimana membuat suatu sistem deteksi intrusi yang dapat mengenali pola serangan baru dari serangan-serangan lama yang sudah ada pada aturan pada *firewall* dan secara otomatis membuat *signaturee* untuk serangan tersebut dan menambahkannya ke dalam *rule* yang ada pada *router firewall.*

2. LANDASAN TEORI

HIDS (*Intrusion Detection System*)

Host Intrusion Detection System dapat didefinisikan sebagai suatu sistem yang mampu mendeteksi aktifitas yang mencurigakan dalam sebuah jaringan yang menuju ke perangkat komputer tersebut, HIDS mampu melakukan pedeteksian dengan cara melakukan pemantauan terhadap lalu lintas (*traffic*) yang keluar maupun masuk dalam sebuah sistem atau jaringan ataupun mendeteksi berdasarkam perbandingan pola lalu lintas jaringan normal yang ada dan kemudian membandingkannya dengan lalu lintas yang ada pada jaringan komputer tersebut.

Komponen-komponen HIDS

a. IDS *Rule*. Merupakan database yang berisi pola-pola serangan berupa signature jenis-jenis serangan. *Rule* IDS ini, harus di *update* secara rutin sehingga IDS mampu mendeteksi jenis serangan baru.
b. IDS *Engine*. Merupakan program yang berjalan sebagai proses yang selalu bekerja untuk membaca paket data dan kemudian membandingkan dengan *rule* IDS.
c. IDS *Alert*. Merupakan catatan serangan pada deteksi penyusupan, jika IDS *engine* mengukumi paket daya yang lewat sebagai serangan, maka IDS *engine* akan mengirimkan *alert* berupa *log file*. Untuk kebutuhan analisa, *alert* dapat disimpan di dalam *database*, sebagai contoh BASE (*Basic Analysis and Securtiy Engine*) yang berfungsi untuk mencari dan mengolah *database* dari *alert network security* yang dibangkitkan oleh perangkat lunak pendeteksi intrusi (IDS).

Jenis Serangan

Jenis Serangan yang dibahas dalam pengujian ini adalah sebagai berikut :

a. *Port Scan*
b. FTP *Brute Force*
c. SSH *Brute Force*
d. TELNET *Brute Force*

Filter

Router akan membuat aturan *firewall* untuk menentukan apakah hendak meneruskan paket yang masuk atau menghentikannya. Ada 3 macam *chain* yang digunakan dalam tabel *filter* ini yaitu :

a. *Input Chain*, yaitu penyaringan *traffic* yang menuju ke sebuah mesin *firewall* atau *router*.
b. *Output Chain*, yaitu penyaringan *traffic* dari mesin *firewall* menuju ke perangkat luar
c. *Forward Chain*, yaitu penyaringan *traffic* untuk meneruskan *traffic* ataupun *request* dari dan ke perangkat lain yang melewati *firewall* atau *router*.

3. PERANCANGAN

Untuk keperluan analisa serangan yang terjadi, maka diperlukan tabel pendukung yang ditambahkan pada *database snort*, masing-masing tabel memiliki kegunaan tersendiri. Adapun tabel yang akan dibuat antara lain :

Pembuatan Tabel Pengamatan

a. Tabel xsshtel

Tabel *xsshtel* adalan tabel yang ditambahkan untuk keperluan penampungan *record* kesalahan otentikasi terhadap layanan FTP, SSH dan TELNET dengan menggunakan serangan *brute force*. Data yang terdapat dalam tabel *xsshtel* merupakan data yang berisi keseluruhan hasil dari pengambilan *record* kesalahan otentikasi menggunakan fungsi *grep* pada file *auth.log* untuk SSH dan TELNET dan file *vsftpd.log* untuk FTP, sehingga *record* yang ada pada tabel *xsshtel* sangat banyak dan masih terdapat banyak *record* data yang sama atau *duplicate record*

b. Tabel xfilter

Tabel *xfilter* berfungsi untuk menampung *record* yang telah diolah berdasarkan *record* yang ada pada tabel *xsshtel*, pengolahan yang dilakukan yaitu mengelompokkan *record* berdasarkan *sig_name (signature name)* dan *ip_src*, sehingga *record* yang dihasilkan yaitu

jumlah total dari masing-masing *record* berdasarkan *sig_name* dan *ip_src* tersebut. Jumlah total ini yang nantinya akan digunakan untuk penentuan apakah kesalahan otentikasi tersebut akan dirubah menjadi *firewall* atau tidak.

c. Tabel temp_ip

Tabel *temp_ip* merupakan tabel yang dibuat secara manual pada *database snort* untuk keperluan menampung *record* yang telah siap untuk dieksekusi menjadi aturan *firewall* pada *router* mikrotik. Pada tabel ini terdapat suatu *field* yang menjadi patokan untuk pengeksekusian *record* menjadi aturan pada *firewall,* yaitu *field executed. field executed* tersebut berisi dua nilai *string* yaitu "*No*" dan "*Yes*", hal tersebut berfungsi sebagai penanda bahwa baris *record* tersebut telah dieksekusi atau belum.

Mekanisme Penyerangan

a. *Port Scan*

Mekanisme penyerangan dengan terhadap *port scanning* akan dilakukan menggunakan perangkat lunak *nmap*. Yaitu dengan memberikan perintah *nmap –A ip_address* pada *nmap* tersebut, penggunakan fitur "*-A*" tersebut dimaksudkan untuk melakukan pemindaian (*scanning*) secara agresif terhadap perangkat komputer *server.*

b. FTP *Brute Force*

Serangan *brute force* terhadap layanan FTP dapat dilakukan menggunakan perangkat lunak *hydra.* Pada perancangan ini *port* yang digunakan untuk layanan FTP masih secara *default* yaitu pada *port* 21. Sehingga perintah pada *hydra* yang akan digunakan yaitu # *hydra –L user.txt –P pass.txt ip_server ftp.* File *user.txt* dan *pass.txt* adalah file yang berisi kumpulan kata atau kamus kata yang akan digunakan penyerang untuk melakukan *brute force.*

c. SSH *Brute Force*

Serangan terhadap layanan SSH menggunakan perangkat lunak *hydra* dengan perintah *#hydra –L user.txt –P pass.txt ip_server ssh.* Untuk menampilkan pencocokan kemungkinan kombinasi *username* dan atau *password* pada layanan SSH, *hydra* memiliki fitur "*-V*", sehingga penulisan perintah menjadi *#hydra -V –L user.txt –P pass.txt ip_server ssh.*

d. TELNET *Brute Force*

Mekanisme serangan yang akan dirancang untuk serangan terhadap TELNET menggunakan perangat lunak *hydra* yaitu dengan memberikan perintah *##hydra –L user.txt –P pass.txt ip_server telnet*, yang secara *default* akan tertuju pada *port* 23. Apabila *administrator* mengganti *port* pada layanan tertentu, maka dengan *hydra* dapat menuliskan perintah *#hydra –L user.txt –P pass.txt –s port ip_server ssh.*

Mekanisme Pendeteksian

a. *Port Scan*

Mekanisme pendeteksian serangan menggunakan *port scanning* dapat dideteksi dengan menganalisa *database snort* pada tabel *acid_event,* untuk *port scanning* akan memiliki *signature name "(port scan) Open Port". Signature name* atau tanda peringatan tersebut dapat dimanfaatkan untuk dijadikan sebuah aturan pemblokiran pada *firewall.*

b. SSH dan TELNET *Brute Force*

Untuk serangan terhadap layanan SSH dan TELNET dapat memanfaatkan file pencatat kejadian otentikasi pada sistem operasi *linux,* yaitu pada file *auth.log* yang berada pada direktori */var/log/auth.log.* File *auth.log* berisi seluruh kejadian *login* baik sukses ataupun gagal, *record* yang dimanfaatkan yaitu *record* kesalahan *login* yang berulang-ulang dan terjadi dari

satu *ip address* yang sama. Pengambilan *record* pada file tersebut dapat menggunakan fungsi *grep* yang terdapat pada sistem operasi *linux* dengan menambahkan kata kunci untuk menyaring *record* yang diperlukan untuk penambahan aturan *firewall* tersebut.

Untuk layanan SSH, kata kunci pengambilan *record* kesalahan pada file *auth.log* yaitu "*Failed*" dan "*ssh*". Pengambilan *record* menggunakan fungsi *grep* dilakukan dengan menuliskan perintah pada terminal (*console*) seperti #*grep Failed /var/log/auth.log | grep ssh.* Sedangkan untuk pengambilan *record* TELNET menggunakan kata kunci "*failure*" dan "*login*" yaitu dengan menuliskan perintah #*grep failure /var/log/auth.log | grep login.*

c. FTP *Brute Force*

Untuk serangan terhadap layanan FTP dapat memanfaatkan file *vsftpd.log* yang berada pada direktori */var/log/vsftpd.log* dengan kata kunci "*FAIL*". Sehingga perintah *grep* yang digunakan yaitu #*grep FAIL /var/log/vsftpd.log.*

Mekanisme Pertahanan

a. *Port Scan*

Pertahanan yang dilakukan terhadap serangan *port scanning* yaitu dengan mengambil *record* pada *database snort* yaitu pada tabel *acid_event* dengan kata kunci "(*port scan*) *Port Open*". *Record* yang didapat tersebut selanjutnya akan dimasukkan (*insert*) kedalam tabel *temp_ip* untuk menunggu dieksekusi menjadi sebuah aturan pada *firewall* pada saat program dijalankan secara terjadwal.

b. FTP, SSH dan TELNET *Brute Force*

Pencegahan terhadap serangan dilakukan dengan menganlisa tabel *xsshtel,* yaitu mengelompokkan kumpulan *record-record* tersebut berdasarkan jenis serangan (*signature name*) dan *ip address* untuk selanjutnya akan dimasukkan (*insert*) kedalam tabel *xfilter.* Berdasarkan tabel *xfilter* akan ditentukan apakah *record* dari jenis serangan (*signature name*) dan *ip address* yang sama melebihi batas toleransi kesalahan, dalam pengujian ini diberikan toleransi sebanyak tiga kali kesalahan *login,* jika *record* kesalahan tersebut melebihi tiga, maka *record* tersebut akan dimasukkan (*insert*) kedalam tabel *temp_ip* untuk menunggu dieksekusi menjadi sebuah aturan pada *firewall* pada saat program berjalan.

4. IMPLEMENTASI

Uji Serangan FTP *Brute Force*

Serangan terhadap layanan FTP pada sebuah perangkat tidak jauh berbeda dengan serangan terhadap layanan SSH, perbedaan hanya terdapat pada *default port* untuk FTP yang berbeda dengan SSH yaitu FTP memiliki *port* 21, penggunaan *port* pada layanan SSH, FTP maupun TELNET dapat disesuaikan atau diganti oleh *administrator* jaringan tersebut, sehingga menyulitkan penyerangan untuk mengakses *port* pada layanan-layanan tersebut. Adapun perintah untuk melakukan serangan terhadap layanan FTP menggunakan terminal (*console*) adalah sebagai berikut #*hydra –L user.txt –P pass.txt ip_tujuan* FTP.

Uji Serangan SSH *Brute Force*

Serangan terhadap layanan SSH yang secara *default* pada *port* 22 yaitu dengan menggunakan perangkat lunak pencocokan kata (*Dictionary Attack*) THC-Hydra, perangkat lunak hydra ini akan melakukan pencocokan *username* dan atau *password* berdasarkan kamus kata yang telah dipersiapkan terlebih dahulu. Proses ini berjalan selama *port* SSH tetap terbuka dan diberikan izin untuk mengakses alamat ip yang dituju. Melakukan serangan

dengan menggunakan *hydra* melalui terminal cukup mudah, yaitu dengan menuliskan perintah *hydra -L user.txt –P pass.txt ip_tujuan* SSH, namun apabila penyerang terlebih dulu mengetahui *username* dari layanan yang akan diserang, maka dapat menuliskan perintah pada terminal seperti # *hydra –l admin –P pass.txt ip_tujuan* SSH.

Uji Serangan TELNET *Brute Force*

TELNET merupakan layanan yang digunakan untuk melakukan pertukaran atau pengaksesan file dari jarak jauh (*remote*) pada suatu perangkat. Untuk *login* secara normal pada layanan TELNET dapat menggunakan perintah pada *console* yaitu # *sudo* TELNET *ip_address*. Serangan pada layanan TELNET tidak jauh berbeda dengan serangan terhadap layanan SSH dan FTP, melakukan serangan dapat dilakukan dengan mengetikkan perintah diterminal *hydra –L user.txt –P pass.txt ip_tujuan* TELNET.

Uji Serangan *Port Scanning* (*NMAP*)

Penggunaan fungsi –A pada *nmap* berguna untuk melakukan *scanning port* secara *aggressive,* penggunaan fungsi ini akan menghasilkan informasi yang terperinci mengenai alamat ip yang dituju, informasi yang dihasilkan seperti daftar *port-port* yang aktif pada perangkat tersebut, *Operating System, Computer-Name, OS finger print,* dan lain-lain. *Snort* akan mencatat aktifitas tersebut ke dalam *database snort* untuk selanjutnya diperoses. *Snort* akan mencatat beberapa entitas dari data tersebut seperti *sig_name, timestamp, ip_src, ip_dst*. Untuk aktifitas seperti *port scanning* akan memiliki *signature name* yang unik yaitu "*port scan*", yang tidak sama dengan aktifitas lainnya yang dicatat oleh *snort*. *Signature name* tersebut dapat dimanfaatkan sebagai kata kunci untuk memilih *record port scanning* pada *database snort*.

5. KESIMPULAN DAN SARAN

Kesimpulan

a. Penerapan aturan *firewall* dapat digunakan secara otomatis berdasarkan jenis dan jumlah serangan yang terjadi dengan menggunakan *nmap* dan *hydra* sebanyak lebih dari tiga kali.
b. Memanfaatkan IDS *Snort*, *auth.log* dan *vsftpd.log* sebagai pendeteksian serangan yang menuju ke *server*.
c. Sistem otomatisasi penambahan aturan *firewall* ini cukup handal dari sisi keamanan jaringan, dikarenakan sistem ini dieksekusi dalam waktu yang sesingkat-singkatnya yaitu dalam interval waktu 1 menit, sehingga *intruder* akan sulit untuk melewati sistem ini

Saran

a. Sistem dapat dikembangkan menjadi sebuah sistem yang *user friendly* dari sisi *administrator,* sehingga hasil penambahan aturan *firewall* yang telah ditambahkan dapat disajikan dalam bentuk informasi dan sistem bisa dikendalikan dari *interface* yang tersedia dengan *web base*.
b. Sistem dapat dikembangkan dengan menambah jumlah serangan yang mampu di deteksi dan di cegah oleh sistem, karena saat ini sistem hanya membahas tentang empat jenis serangan terhadap jaringan komputer.

DAFTAR PUSTAKA

Cartealy, Imam. 2013. *Linux Networking (Ubuntu, Kubuntu, Debian, dll)*. Jasakom.

Raharjo, Budi. 2011. *Belajar otodidak PEMROGRAMAN WEB dengan PHP + ORACLE*. Bandung : Informatika Bandung.

Riyanto. 2010. *Sistem Informasi Penjualan dengan PHP dan MySQL*. Yogyakarta : GAVA MEDIA

Sulistiyani, Sri. *Administrasi jaringan dengan Linux ubuntu*. 2011. Yogyakarta : ANDI.

Tuxkeren, Athailah. 2013 . *Ubuntu Server Panduan Singkat & Cepat*. Jasakom.

Zam, Efvy. 2012. *Wireless Hacking*. Jakarta : PT Elex Media Komputindo