Intrusion Preventing System Pada Jaringan Wireless STIKOM Surabaya Menggunakan SNORT dan IPTables

Slamet
Program Studi S1 Sistem Informasi, Institut Bisnis dan Informatika Stikom Surabaya
Email:slamet@stikom.edu

Abstrak

Perkembangan IT yang demikian pesat sangat membantu pekerjaan manusia. Di satu sisi, pekerjaan manusia sangat terbantu, tetapi di sisi lain jumlah insiden keamanan sistem informasi meningkat tajam, sehingga pada hakekatnya sisi-sisi kehidupan manusia dalam posisi terancam. Teknik-teknik pencegahan terhadap serangan pada sistem informasi harus terus dikembangkan sehingga integritas, availibilitas dan confidentialitas pada sebuah sistem informasi menjadi lebih terjamin. Salah satu caranya adalah dengan mengembangkan *Intrusion Preventing System* (Sistem Pencegahan Penyusupan). Dalam tulisan ini, penulis membangun sebuah *Intrusion Preventing System* dengan menggunakan *Snort IDS* dan *IPTables Firewall*. Sistem ini bekerja dengan cara membuat *alert* yang dibangun dari sebuah *engine* yang membaca parameter berupa IP *address* penyerang. Kemudian *alert* memerintahkan *firewall* untuk memblokir akses dari IP *Address* tersebut. Untuk mempermudah manajemen *rule* digunakan *Webmin.* Sedangkan untuk menganalisa *log* (*history* serangan) digunakan ACID (*Analysis Console for Intrusion Databases*). Pengujian dilakukan pada jaringan *wireless* Institut Bisnis dan Informatika Stikom Surabaya. Implementasi sistem menggunakan PC Router dengan sistem operasi Linux Ubuntu 14.04. Dengan implementasi aplikasi ini, sistem mampu memblokir (menutup) akses terhadap usaha-usaha penyerangan di jaringan *wireless* Stikom Surabaya. Dengan demikian koneksi internet lebih lancar dan pemakaian *bandwidth* lebih efisien.

Kata kunci: *intrusion detection, intrusion prevention*, snort, iptables, *bandwidth*

Berkembangnya teknologi informasi khususnya jaringan komputer dan layanan-layanannya di satu sisi mempermudah pekerjaan manusia sehari-hari. Akan tetapi di sisi lain timbul masalah yang sangat serius yakni faktor keamanan. Di satu sisi, manusia sudah sangat tergantung dengan sistem informasi, akan tetapi statistik insiden keamanan meningkat tajam. Hal ini secara umum terjadi karena kepedulian terhadap keamanan sistem informasi masih sangat kurang.

Pada awalnya jaringan komputer hanya menggunakan media kabel untuk saling berhubungan. Namun dalam perkembangannya, pemakaian jaringan *wireless* semakin meningkat. Aplikasi jaringan *wireless* ini memberikan dampak perubahan yang cukup signifikan yang memungkinkan orang–orang bisa memperluas ruang kerja mereka karena tidak terikat pada penggunaan kabel.

Keamanan jaringan sebagai sebuah bagian dari sistem menjadi sangat penting guna menjaga validitas dan integritas data, serta menjamin ketersediaan layanan bagi pemakainya. Terutama pada jaringan *wireless* karena data–data

dibawa oleh sinyal komunikasi yang merambat melalui udara dan lebih rentan terhadap serangan. Tidak seperti jaringan kabel, dimana serangan yang datang harus mengakses secara fisik melalui kabel yang terhubung dan dilengkapi dengan pertahanan, serangan yang terjadi pada jaringan *wireless* dapat datang dari mana saja, dan serangan dapat ditujukan ke semua *node*.

Sistem pertahanan terhadap aktivitas gangguan saat ini umumnya dilakukan secara manual oleh Administrator. Hal ini berarti bahwa keamanan pada sebuah jaringan bergantung kepada ketersediaan dan kecepatan respon Administrator. Apabila serangan membuat sistem menjadi *down*, maka Administrator tidak dapat lagi mengakses sistem dan melakukan pemulihan. Oleh karena itu dibutuhkan suatu sistem yang dapat membantu Administrator untuk memantau jaringan sehingga dapat menanggulangi ancaman yang mungkin akan terjadi secara optimal dalam waktu yang cepat.

Untuk mencegah insiden keamanan perlu dilakukan langkah-langkah preventif baik teknis maupun non teknis. Pencegahan dengan non teknis dapat dilakukan dengan membuat *security policy* yang baik dan terkendali. Sedangkan pencegahan secara teknis dapat dilakukan dengan langkah-langkah *hardening* di sistem operasi, aplikasi, infrastruktur jaringan, dan implementasi *Intrusion Prevention System* (Sistem Pencegahan Penyusupan).

TUJUAN DAN RUANG LINGKUP PEMBAHASAN

Berdasarkan latar belakang di atas, tulisan ini membahas pencegahan penyusupan secara teknis yang secara khusus berupa *hardening* sistem dengan IPS *(Intrusion Prevention System)*. IPS berbasis Snort IDS dan IPTables *Firewall*, yakni sebuah sistem yang mampu melakukan deteksi dan kemudian melakukan pencegahan terhadap usaha penyusupan dengan memblok (menutup) akses paket data yang berasal dari penyusup tersebut.

Sistem Pencegahan Penyusupan ini diimplementasikan dalam sebuah PCRouter yang terkoneksi ke konsentrator berupa Switch dengan *wireless* sebagai media koneksinya.

INTRUSION PREVENTING SYSTEM (SISTEM PENCEGAHAN PENYUSUPAN)

Sistem Pencegahan Penyusupan (*Intrusion Preventing System* atau IPS) adalah suatu *tools* yang digunakan untuk mencegah adanya penyusupan.

Ada 2 fungsi dalam IPS yakni kemampuan mendeteksi penyusupan dan kemampuan untuk mencegah akses penyusupan. Kemampuan mendeteksi penyusupan secara umum disebut IDS *(Instrusion Detection System)* dan kemampuan untuk mencegah akses dikenal dengan *Firewall.*

Komponen Sistem Pencegahan Penyusupan

Sistem pencegahan penyusupan harus dapat mendeteksi dan merespon terhadap penyusupan yakni dengan mengkonfigurasi ulang *rule firewall* yang ada. Untuk itu komponen-komponen yang harus ada pada sistem pencegahan penyusupan meliputi: *IDS (Intrusion Detection System, Packet Filtering dan IDS-Firewall.*

IDS (Intrusion Detection System)

Dilihat dari cara kerja dalam menganalisa apakah paket data dianggap sebagai penyusupan atau bukan, IDS dibagi menjadi 2: *knowledge based* atau *misuse detection* dan *behavior based* atau *anomaly based*.

Knowledge-based IDS dapat mengenali adanya penyusupan dengan cara menyadap paket data kemudian membandingkannya dengan *data base rule* IDS (berisi *signature-signature* paket serangan). Jika paket data mempunyai pola yang sama dengan (setidaknya) salah satu pola di *database rule* IDS, maka paket tersebut dianggap sebagai serangan, dan demikian juga sebaliknya, jika paket data tersebut sama sekali tidak mempunyai pola yang sama dengan pola di database *rule* IDS, maka paket data tersebut dianggap bukan serangan.

Sedangkan *behavior based (anomaly)* dapat mendeteksi adanya penyusupan dengan mengamati adanya kejanggalan-kejanggalan pada sistem, atau adanya penyimpangan-penyimpangan dari kondisi normal, sebagai contoh ada penggunaan memori yang melonjak secara terus menerus atau ada koneksi paralel dari 1 buah IP dalam jumlah banyak dan dalam waktu yang bersamaan.

Kondisi-kondisi diatas dianggap kejanggalan yang kemudian oleh IDS jenis *anomaly based* dianggap sebagai serangan.

Sedangkan dilihat dari kemampuan mendeteksi penyusupan pada jaringan, IDS dibagi menjadi 2 yakni: *host based* dan *network based*. *Host based* mampu mendeteksi hanya pada *host* tempat implementasi IDS, sedangkan *network based IDS* mampu mendeteksi seluruh *host* yang berada satu jaringan dengan *host* implementasi IDS tersebut. Tulisan ini secara khusus menggunakan IDS jenis *knowledge based* dan *network based*.

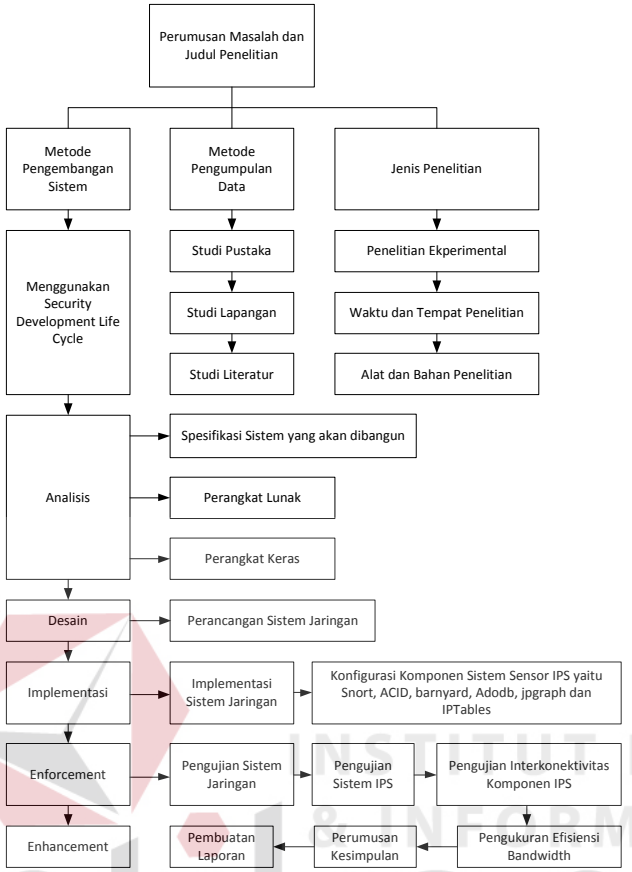
Packet Filtering Firewall

Packet Filtering Firewall dapat membatasi akses koneksi berdasarkan parameter-parameter: protokol, IP asal, IP tujuan, port asal, port tujuan, *chain* (aliran data) dan *code bit* sehingga dapat diatur hanya akses yang sesuai dengan *policy* saja yang dapat mengakses sistem.

Packet filtering firewall ini bersifat statik sehingga fungsi untuk membatasi aksespun statik, misalnya akses ke *web server* (port 80) diijinkan oleh *policy*, maka dari manapun dan apapun aktifitas terhadap web server diijinkan walaupun merupakan usaha penetrasi oleh *cracker*. Untuk itulah *packet filtering firewall* tidak dapat mengatasi gangguan yang bersifat dinamik sehingga harus dikombinasikan dengan IDS untuk membentuk sistem *hardening* yang maksimal.

Engine* Sistem Pencegahan Penyusupan *(IDS-Firewall)

Engine ini bertugas untuk membaca *alert* dari IDS (antara lain berupa jenis serangan dan *IP Address* penyusup) untuk kemudian memerintahkan *firewall* untuk memblok akses koneksi ke sistem dari penyusup tersebut.



Gambar 1: Diagram Alir Penelitian

METODE PENELITIAN

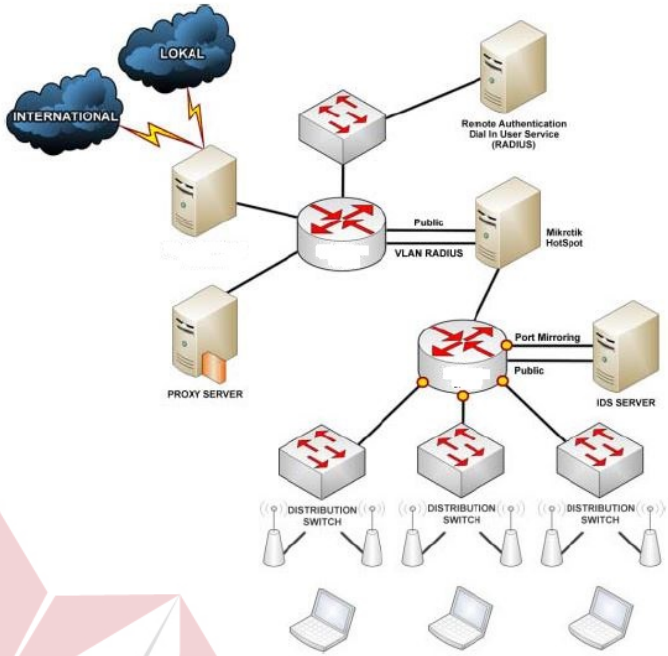
Langkah-langkah yang digunakan dalam menyelesaikan penelitian ini adalah seperti pada Gambar 1.

HASIL DAN PEMBAHASAN

Metode penelitian yang penulis gunakan adalah metode *Security Policy Development Life Cycle (SPDLC)*. Dengan SPDLC, siklus hidup pengembangan sistem jaringan didefinisikan pada sejumlah fase, antara lain: *analysis*, *design*, *implementation*, *enforcement* dan *enhancement.*

TAHAP *ANALYSIS* (ANALISIS)

Model SPDLC memulai siklus pengembangan sistem jaringannya pada tahap analisis. Pada tahap ini, dianalisa spesifikasi sistem yang akan dibangun, perangkat yang dibutuhkan seperti perangkat lunak *(software)* dan perangkat keras *(hardware)* yang dibutuhkan untuk sistem IDS.



Gambar 2: Rancangan Topologi Jaringan

TAHAP *DESIGN* (PERANCANGAN)

Perancangan ini berdasarkan konsep dan gambaran yang menjelaskan perangkat yang sebenarnya. Sistem pendeteksi intrusi yang dikembangkan berjenis NIDS *(Network Intrusion Detection System).* Hal ini karena IDS jenis ini ditempatkan di sebuah tempat/titik yang strategis atau sebuah titik di dalam sebuah jaringan untuk melakukan pengawasan terhadap *traffic* yang menuju dan berasal dari semua alat-alat *(devices)* dalam jaringan. Idealnya semua *traffic* yang berasal dari luar dan dalam jaringan dilakukan proses *scanning*. Rancangan topologi saat diterapkan IPS dapat dilihat pada gambar 2.

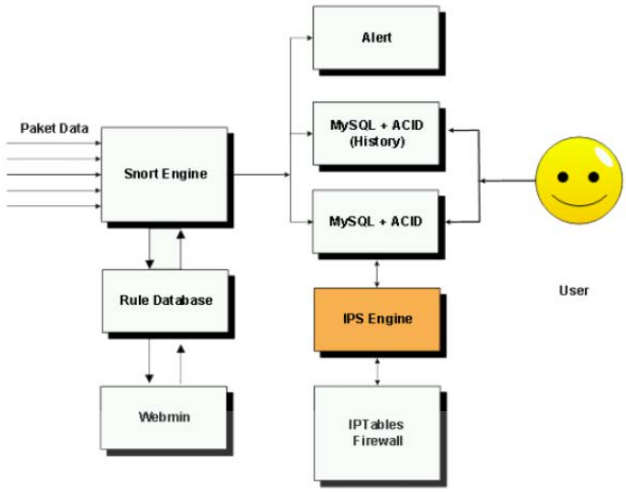
TAHAP *IMPLEMENTATION* (IMPLEMENTASI)

Fase selanjutnya adalah implementasi atau penerapan detail rancangan topologi dan rancangan sistem. Detail rancangan yang digunakan sebagai instruksi atau panduan tahap implementasi agar sistem yang dibangun menjadi relevan dengan sistem yang sudah dirancang.

Implementasi Sistem Pencegahan Penyusupan

Untuk memenuhi kebutuhan fungsional sistem pencegahan penyusupan dibutuhkan modul-modul utama dan modul pendukung. Modul utama berupa: *snort engine, rule snort, engine* IPS dan *firewall*. Sedangkan modul pendukung berupa: ACID (manajemen *event*) dan Webmin (manajemen *rule*).

Target implementasi IPS di sistem Linux Ubuntu 14.04. Diagram blok sistem pencegahan penyusupan yang dirancang sebagai berikut.



Gambar 3: Blok diagram IPS

Rule Snort

Modul ini menyediakan *rule-rule* berupa *pattern* jenis serangan. *Rule* ini berupa file text yang disusun dengan aturan tertentu.

Snort Engine

Modul ini berfungsi untuk membaca paket data dan membandingkannya dengan *rule database*, jika paket data dihukumi sebagai penyusupan/serangan, maka *Snort engine* akan menuliskannya ke *alert* (berbentuk *file log*) dan ke *database* (yang digunakan dalam eksperimen ini adalah database MySQL).

Alert

Bagian ini merupakan catatan serangan pada sebuah *file log*.

Webmin

Webmin (http://www.webmin.com/) yang telah ditambahkan *module snort rule* (http://msbnetworks.com/snort/) digunakan untuk mengelola *rule*. *Rule* mana saja yang akan di-*enable* dan *disable* dapat diatur melalui Webmin, bahkan dapat digunakan untuk menambahkan rule-rule secara manual dengan editor berbasis web.

ACID (Analysis Console for Intrusion Databases)

ACID (http://www.cert.org/kb/acid) digunakan untuk mengelola *data-data security event*, keuntungan menggunakan ACID diantaranya: *log-log* yang tadinya susah dibaca menjadi mudah di baca, data-data dapat dicari dan difilter sesuai dengan kriteria tertentu, *Managing Large Alert Databases (Deleting and*

Archiving), dan untuk kasus-kasus tertentu dapat merujuk *alert* pada situs *database security* seperti *Securityfocus, CVE, arachNIDS*.

ACID (Analysis Console for Intrusion Databases) History

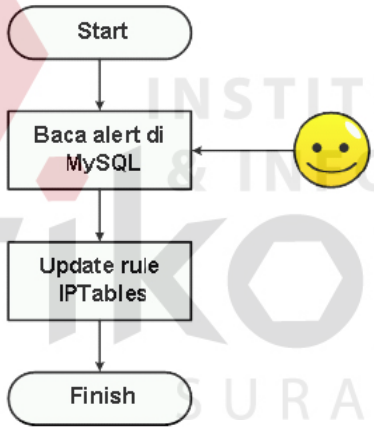
ACID history ini digunakan untuk menganalisa catatan-catatan IDS. *Database* dalam *history* ini tidak dihapus karena bersifat seperti arsip-berbeda dengan database pada ACID yang akan dihapus catatan serangannya jika pengelola IPS ini berkehendak untuk membuka akses bagi IP address yang pernah ditutupnya.

Firewall

Firewall digunakan untuk membuka dan menutup akses sesuai dengan rule yang dibuat, dalam hal ini rule akan dinamis sesuai dengan kondisi yang dideteksi oleh IDS. Firewall yang digunakan dalam eksperimen ini adalah Iptables yang merupakan firewall bawaan Linux.

IPS Engine

IPS engine merupakan sistem yang akan membaca alert kemudian memerintahkan firewall untuk menutup akses paket data dari penyerang. Cara kerja IPS engine digambarkan dalam *flowchart* berikut ini:



Gambar 4: *Flowchart IPS engine*

IPS akan menutup akses bagi penyerang ketika aktivitas tersebut terdeteksi oleh IDS. Dalam eksperimen ini, proses pembacaan *alert* dan *update rule* pada *firewall* dilakukan secara periodik dengan meletakkan program IPS *engine* (yang ditulis dalam bahasa PHP) dilakukan *crontab (scheduling task)*. Jadi, ketika terjadi usaha penyusupan dan terdeteksi oleh IDS, maka IPS akan memerintahkan *firewall* untuk menutup akses dari IP *address* penyerang, adapun jika pada waktu yang lain pengelola IPS akan membuka IP address - IP address yang telah melakukan penyerangan, hal ini dapat dilakukan dengan menghapus isi *alert* pada serangan dari IP yang dimaksud pada *database* ACID.

TAHAP *ENFORCEMENT* (PENGUJIAN)

Model pengembangan sistem jaringan komputer SPDLC mengkategorikan *enforcement* pada tahap pengujian. Hal ini dikarenakan pengawasan sistem yang sudah dibangun hanya dapat dilakukan jika sistem sudah dapat bekerja sesuai dengan kebutuhan. Proses pengujian *(testing)* dibutuhkan untuk menjamin dan memastikan bahwa sistem yang dibangun sudah sesuai memenuhi spesifikasi rancangan dan memenuhi kebutuhan permasalahan yang ada di Institut Bisnis dan Informatika Stikom Surabaya.

Kasus 1: Ping Attack (ICMP Attack)

Untuk menguji rancangan sistem pencegahan penyusupan dengan cara melancarkan paket serangan ke sistem yang dilindungi oleh IPS. IPS diimplementasikan pada jaringan router yang menghubungkan jaringan intranet dan DMZ. Pada pengujian ini dikirimkan paket ICMP dalam ukuran besar sehingga dikategorikan oleh IDS sebagai DOS attack *(denial of service).*

Berikut pengujian yang dilakukan melalui *client* di jaringan internal.

ping 173.194.126.116 -l 10000 -t
Pinging 173.194.126.116 with 10000 bytes of data:
Reply from 173.194.126.116: bytes=10000 time=10ms TTL=63
Reply from 173.194.126.116: bytes=10000 time=10ms TTL=63
Reply from 173.194.126.116: bytes=10000 time=10ms TTL=63
Ping statistics for 173.194.126.116:
Packets: Sent = 3, Received = 3, Lost = 0 (0Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 10ms, Average = 3ms

DOS *attack* ini akan segera terdeteksi oleh *snort engine* yang kemudian *snort engine* akan mengirimkan *alert* ke *alert log*, MySQL ACID dan MySQL ACID *history*. *IPS engine* membaca *alert* pada ACID MySQL dan kemudian memerintahkan *firewall* untuk meng-*update rule*-nya dengan menambahkan *rule* untuk memblok akses dari IP penyerang yang terdeteksi. Pengamatan eksperimen ini dilakukan pada 2 tempat: di *client* tempat melancarkan serangan dan di sistem IPS.

Pengamatan di *client* penyerang

Pengamatan dilakukan dengan cara melakukan pengiriman paket ICMP dengan perintah ping ke komputer target seperti berikut ini ping 173.194.126.116 -t
Pinging 173.194.126.116 with 32 bytes of data:
Reply from 173.194.126.116:bytes=32 time<10ms TTL=63
Reply from 173.194.126.116:bytes=32 time<10ms TTL=63
Reply from 173.194.126.116:bytes=32 time<10ms TTL=63
Request timed out.
Request timed out.

Request timed out.
Ping statistics for 173.194.126.116:
Packets: Sent = 6, Received =3, Lost =35 (50Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

Dari pengamatan di atas terlihat bahwasanya IPS telah bekerja dengan baik, hal ini ditunjukkan dengan tertutupnya akses ke komputer target serangan dengan munculnya pesan *"Request Time Out"* yang sebelumnya *"Reply"*

Pengamatan di Sistem Pencegahan Penyusupan

Adapun pengamatan di sistem pencegahan penyusupan dilakukan dengan mengamati *rule firewall* yang telah berubah, yakni IPS *engine* memasukkan IP address penyerang sebagai sebuah *rule* dimana akses dari komputer tersebut harus diblok (tidak diijinkan). Berikut pengamatan di sistem:

Chain INPUT (policy ACCEPT)
target prot opt source destination

Chain FORWARD (policy ACCEPT)
target prot opt source destination
DROP all -- 10.1.1.11 anywhere

Chain OUTPUT (policy ACCEPT)
target prot opt source destination

Dari pengamatan di dua sisi, sisi penyerang dan sisi sistem pencegahan penyusupan, dapat disimpulkan bahwa fungsional sistem ini telah berjalan dengan yang diharapkan. Serangan-serangan jenis yang lain juga akan bernasib sama yakni diblok, hal ini tentunya tergantung dari ketelitian dan kelengkapan *rule snort*. *Rule snort* begitu lengkap sehingga mampu mendeteksi banyak jenis serangan. Berikut contoh *rule Snort*.

attack-responses.rules, dos.rules, local.rules, oracle.rules, scan.rules, web-cgi.rules, backdoor.rules, experimental.rules, Makefile other-ids.rules, shellcode.rules, web-client.rules, bad-traffic.rules, exploit.rules, Makefile.am p2p.rules, smtp.rules, web-coldfusion.rules, chat.rules, finger.rules, Makefile.in policy.rules, snmp.rules, web-frontpage.rules, db.config ftp.rules, misc.rules, pop2.rules, sql.rules, web-iis.rules, db.timestamp icmp-info.rules, multimedia.rules, pop3.rules, telnet.rules, web-misc.rules, ddos.rules, icmp.rules, mysql.rules, porn.rules, tftp.rules, web-php.rules, deleted.rules, imap.rules, netbios.rules, rpc.rules, virus.rules, x11.rules, dns.rules, info.rules, nntp.rules, rservices.rules, web-attacks.rules

Kasus 2: Nmap Port Scanning Attack

Pada kasus ini, penulis akan mensimulasikan dan menganalisis jenis aktivitas *port scanning* dengan menggunakan nmap, yang dilakukan dari kedua mesin penyerang, internal *(Client)* dan penyerang eksternal.

Langkah pertama adalah membuat *rules/signatures* untuk mendefinisikan jenis aktivitas ini. Berdasarkan hasil analisis *traffic*, penulis mendefinisikan nmap ping sebagai berikut:

alert icmp any any -> any any (msg:"ICMP PING NMAP attack"; dsize:0;
itype:8; rev:1; sid: 100003;)

Signature atau *rules* di atas akan meng-*generate alert* Snort jika mendeteksi akses protokol ICMP yang berasal dari segmen jaringan eksternal maupun internal, melalui *port* berapapun ke 172.16.1.1 (mesin server) *port* berapapun: keterangan *rules:* "ICMP PING NMAP attack"; berukuran paket 0 byte; menggunakan tipe icmp 8; revisi *rules* pertama: nomor identitas *rules* 100003.

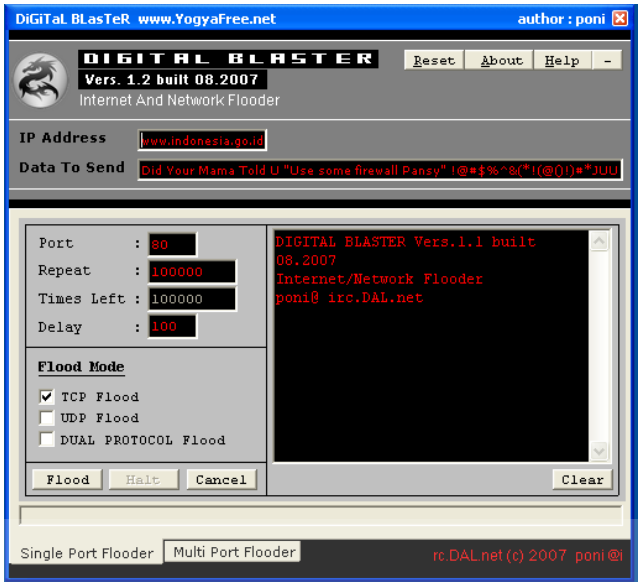
Langkah kedua adalah menerapkan *rules/signature* baru ini dengan menempatkannya pada direktori *rules* Snort (/etc/snort/rules). Pada penelitian ini, penulis menyimpan *signature* ini dengan nama localrules. Setelah itu, proses Snort harus di-*restart*, agar Snort dapat mendeteksi, membaca, dan menerapkan *rules* baru tersebut pada kode intinya. Proses untuk *merestart* snort adalah (/etc/init.d/snort restart).

Langkah ketiga adalah melancarkan serangan dengan menggunakan nmap.

Kasus 3: Digital Blaster

Pada kasus ini, penulis mendefinisikan akan menggunakan dan menganalisa jenis aktivitas *port scanning* dengan menggunakan program digital blaster, yang dilakukan dari *client* atau mesin penyerang.

Pada percobaan ini, *client* mencoba *port scanning* dengan menggunakan digital blaster yaitu ke mesin sensor IDS 172.16.1.4. Dengan IP DigiBlast atau Digital Blaster dapat mengirimkan paket ke alamat IP target ke sebuah *port* yang ditentukan oleh *user ("Single Port Flooder")* maupun ke banyak *port* yang terbuka *("Multi Port Flooder")*. Yang perlu dilakukan untuk mengirimkan paket ke alamat IP target adalah memastikan bahwa alamat IP target aktif.



Gambar 5.8: Uji Coba DigiBlast Client ke Server

TAHAP *ENHANCEMENT*

Pada fase ini meliputi aktivitas perbaikan terhadap sistem yang telah dibangun. Fase *enhancement* melalui serangkaian proses perbaikan dilakukan untuk sejumlah tujuan:

1. Memperbaiki sejumlah kesalahan yang terdapat pada penerapan sistem sebelumnya (sistem yang sudah ada).
2. Menambahkan fungsionalitas atas komponen spesifik atau fitur tambahan terbaru untuk melengkapi kekurangan pada sistem sebelumnya.
3. Mengadaptasi sistem yang sudah dibangun terhadap *platform* dan teknologi baru dalam mengatasi sejumlah perkembangan permasalahan baru yang muncul.

Dengan demikian, fase perbaikan dapat secara efektif menjamin kehandalan kinerja dari IDS.

KESIMPULAN DAN SARAN

Sebuah sistem pencegahan penyusupan haruslah mempunyai fungsi: deteksi (IDS) dan memberikan respon berupa update *rule firewall*, untuk mempermudah pengelolaan IPS dibutuhkan *module-module* tambahan selain IDS dan Firewall.

Kesimpulan

Secara keseluruhan eksperimen ini, dapat disimpulkan bahwa:

1. Serangan/penyusupan dapat dicegah dengan implementasi Sistem Pencegahan Penyusupan
2. Serangan dapat terdeteksi atau tidak tergantung pola serangan tersebut ada di dalam *rule* IDS atau tidak. Oleh karena itu, pengelola IDS harus secara rutin meng-*update rule* terbaru.

3. *Plugin snort rule* berupa webmin dapat digunakan untuk mempermudah pengolahan rule dan ACID dapat dimanfaatkan guna mempermudah analisa terhadap catatan *IDS (security event).*

Saran

Sistem yang dibangun masih terdapat banyak kekurangan, karena keterbatasan waktu dan sarana. Saran-saran yang dapat diberikan bagi yang ingin mendalami lebih lanjut sistem ini antara lain:

1. *Update rule* pada *firewall* seharusnya dalam bentuk *daemon* proses sehingga proses bekerja secara *real time.*
2. Manajemen *rule* dapat dibuat sendiri dengan pemrograman aplikasi berbasis web dan tidak menggunakan Webmin. Webmin merupakan administrasi secara keseluruhan sistem Linux dan hanya dilindungi dengan form *autentification.*

DAFTAR PUSTAKA

Ariyus, Dony. 2007. *Intrusion Detection System*. Yogyakarta : Penerbit Andi

Geier, Jim. 2005. *Wireless Network First-Step*. Yogyakarta : Penerbit Andi

IDS FAQ. 2014. *http://www.sans.org/resources/idfaq/*. Sans.

IPTables Manual. 2014. *http://ipset.netfilter.org/iptables.man.html.* Jozsef Kadlecsik

Pradana, Yoga. 2013. *Desain dan Implementasi Keamanan Jaringan Firewall pada Embedded System*. Program Teknologi Informasi dan Ilmu Komputer : Universitas Indonesia

Rahardjo, Budi. 2013. *Security Tools untuk pengamanan, Firewall dan Intrusion Detection System (IDS)*. IndoCisc

SnortTM Users Manual. 2014. http://www.snort.org/. The Snort Project

Snort FAQ. 2014. http://www.snort.org/. The Snort Project

Wahsheh, Luay A dan Foss, Jim Alves. 2008. *Security Policy Development: Towards a Life-Cycle and Logic-Based Verification Model*. USA