BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan dari penelitian ini adalah sebagai berikut:

1. Dengan memanfaatkan *ruleset signature* Snort, kemudian menjalankan dan mengkonfigurasinya dengan benar, dapat dihasilkan sistem pendeteksi terhadap aktifitas *malicious*. Untuk aktifitas normal yang tidak terdeteksi *malicious* akan diteruskan seperti biasa menuju server asli. Di samping untuk mendeteksi intrusi, pemanfaatan Snort berbasis *signature* dapat menghasilkan sebuah IDS yang minim kesalahan, karena *signature* Snort memang benar-benar *ruleset* yang mendefinisikan *malicious* secara tepat, bukan dari statistik sebuah anomali (kejanggalan) pada jaringan.
2. Mengkombinasikan Snort (IDS) dan IPTables (*Firewall*) adalah dengan memanfaatkan program BlockIt, *parser* untuk *log* Snort. Secara bertahap proses-proses yang terjadi adalah sebagai berikut: semua hasil *alert* Snort yang di *generate* ke file *log* Snort di *parsing*. Nilai-nilai yang di *parsing* tersebut selanjutnya dimasukkan ke dalam perintah-perintah IPTables pada sistem Linux. Perintah-perintah yang telah di buat selanjutnya akan segera mengkonfigurasi *firewall* sistem untuk mengaktifkan *rule firewall* baru untuk nilai-nilai tertentu (*matches*).

3. Dengan memanfaatkan *database* sebagai tempat penyimpanan alamat *host* yang pernah melakukan serangan, dapat dijadikan bahan untuk membuat statistik serangan sehingga toleransi serangan dapat dibuat. Selain sebagai tempat memperoleh data untuk keperluan patokan batas toleransi, *data host* yang tercatat pada *database* bisa dimanfaatkan sebagai sumber informasi untuk mengetahui total aktifias *malicious* yang dilakukan seluruh *host* terhadap server, mengetahui jumlah *host* yang termasuk *blacklist*, mengetahui banyaknya serangan dan jenis serangan yang dilakukan *host* tertentu, dan sebagainya.
4. *Network Address Translation* (NAT) dimanfaatkan untuk me-*redirect* paket. *Destination* NAT (DNAT) digunakan pada sistem ketika paket yang datang melewati IPTables cocok/*match* dengan aturan pada tabel NAT. Sehingga *field destination host* pada *header* paket diganti oleh kernel menjadi sesuai yang diinginkan. Dalam hal ini, *field destination* pada tabel NAT akan diubah menjadi menuju ke IP *address host* Honeyweb.
5. *Source* NAT (SNAT) dipakai saat ingin merubah *field source* IP *address* pada *header* paket. Dalam hal ini, seluruh *header* paket yang berasal dari *host* Honeyweb akan diubah *field* sumber asalnya dari alamat IP *host* Honeyweb menjadi alamat IP server. Sehingga bagi *client*, *response* server terlihat seakan-akan berasal dari *webserver* asli.
6. Honeyweb digunakan sebagai tempat interaksi bagi *client webserver* yang terdeteksi IDS melakukan tindakan *malicious*. Cara kerjanya adalah

dengan melakukan *port listen* pada *port* 80 (HTTP). Sehingga seluruh HTTP *request* menuju *port* 80 dari *client* akan di *response* oleh Honeyweb. Bedanya dengan *webserver* asli adalah Honeyweb Server sudah di *setting* menjadi *vulnerable*. Honeyweb yang *vulnerable* akan terlihat dari HTTP *response* yang dihasilkannya benar-benar seperti sebuah sistem yang lemah dan tidak diproteksi ketika *client* melakukan HTTP *request* yang tidak sewajarnya. Pada Honeyweb, seluruh aktifitas HTTP *request* dari *client* akan dicatat.

7. Untuk memproteksi akses layanan SSH dari *host* atau subnet tertentu dimanfaatkan *Generic Matches* "--source" pada IPTables. Sehingga jika paket yang datang dari *source* tertentu cocok dengan aturan pada *generic matches --source*, maka paket tersebut akan mendapatkan perlakuan tertentu dari *firewall* yang telah dikonfigurasi oleh *Network Administrator* (diizinkan/ditolak/*redirect*).
8. Untuk pengamanan dari aktifitas *bruteforce*, dimanfaatkan sintaks IPTables *explicit matches* "--state", "--hitcount", dan "--seconds". Jadi sebagai parameter *bruteforce* adalah *state* koneksi dengan *hitcount* yang terjadi dalam waktu tertentu. Misalnya, telah ditetapkan bahwa aktifitas *bruteforce* adalah jika terjadi koneksi pada server dengan *hitcount* 8 kali dalam *interval* waktu 30 detik. Maka, jika ada koneksi ke-9 dari suatu *host* pada server tetapi masih pada *interval* 30 detik pertama, *firewall* akan menganggap hal tersebut sebagai tindakan *bruteforce*.

9. Kebutuhan pemanfaatan *Intrusion Prevention System* (IPS) terhadap suatu *node* yang ingin dilindungi kembali lagi kepada kesiapan *Network Administrator* untuk menghadapi berbagai kemungkinan yang dilakukan oleh IPS yang proaktif tersebut. Di satu sisi IPS bisa menolong *Network Administrator* untuk mencegah terjadinya serangan lebih lanjut secara mandiri, tetapi di sisi lain, IPS malah bisa menambah pekerjaan *Network Administrator* jika ternyata yang diblokir merupakan hasil *false positive* IPS. Maka dari itu diperlukan analisis kebutuhan dan parameter yang jelas oleh *Network Administrator*. Yang mana yang akan dianggap intrusi bagi IPS, dan yang mana yang tidak. Analisis kebutuhan dan parameter tiap jaringan bisa sangat berbeda-beda atau dengan kata lain bersifat kondisional.
10. *More security doesn't make you more secure. Better management does.* Ini adalah kalimat bermakna sangat dalam bagi para praktisi keamanan jaringan. Se-aman-amannya proteksi yang diterapkan pada suatu *node*, tetapi ketika *node* tidak diatur dengan baik, maka tetap bisa terancam keamanannya. Bisa dilakukan bahkan hanya dengan cara-cara non-teknis.

5.2 Saran

Saran dari penulis terhadap pengembangan dari penelitian ini adalah sebagai berikut:

1. Mengembangkan Sistem Pakar yang bisa menentukan suatu *request* termasuk kegiatan yang bisa merusak dan membahayakan sistem asli atau

tidak. Sistem Pakar ini dikonfigurasi untuk memantau *log* aktifitas pada Honeypot Server. Jika termasuk kegiatan yang bisa merusak, maka Sistem Pakar akan menulis *ruleset* Snort baru. Dengan begini, sistem menjadi lebih mandiri bahkan dapat dikatakan tidak diperlukan lagi jasa *Network Administrator* sama sekali.

2. Mengembangkan IPS pada IDS yang bertipe *anomaly-based* (berbasis kejanggalan pada jaringan). Dengan catatan terlebih dahulu bisa meminimalisir *false positive* yang dihasilkan pada IDS tipe tersebut.
3. Mengembangkan ruang lingkup yang di-*monitor* tidak hanya HTTP dengan Honeyweb-nya. Tetapi juga dapat mengemulasi *services* protokol-protokol lain yang sering dieksploitasi (SMTP, Telnet, POP, dll). Sehingga Honeypot bisa dimanfaatkan sebagai *Knowledge Base* jenis-jenis *threat* yang mengancam keamanan komputer di jaringan dan internet.
4. Membuat IPS *ruleset based* yang bisa mendeteksi pola serangan di luar *ruleset signature* dengan cara *heuristik.*