1. Pendahuluan

1.1 Latar Belakang

Virtualisasi memberikan keuntungan dari segi keamanan dengan adanya isolasi yang diberikan oleh *host* pada *guest*. Pada teknologi ini, aplikasi-aplikasi *server* dipisahkan ke sejumlah sistem operasi guest yang berjalan di atas *virtual machine*-nya masing-masing. Ide dasar dipisahkannya aplikasi ke virtual machine masing-masing adalah untuk mencegah adanya saling intervensi antar aplikasi ketika hal yang tidak diinginkan terjadi [7]. Untuk merealisasikan isolasi dalam teknologi virtualisasi, pemisahan aplikasi saja tidaklah cukup. Mode jaringan virtual yang tepat dan mekanisme *firewall* yang tepat diperlukan untuk memperkuat isolasi ini. Firewall bukan solusi lengkap dalam mengisolasi guest-guest pada virtual machine, tetapi merupakan komponen utama dalam isolasi jaringan.

Ketika yang guest divirtualisasikan adalah sebuah server yang melayani *request* dari luar, maka guest harus dapat diakses dari jaringan luar. Mode jaringan virtual yang paling umum dijumpai untuk menangani kasus virtualisasi server adalah *bridged networking* [3, 4, 11]. Pada bridged networking, virtual machine menggunakan *network interface* fisik pada komputer *host* untuk mengirim dan menerima data (dalam bentuk *frame* pada *data link* layer).

Permasalahan yang dihadapi dalam briged networking adalah tidak-adanya mekanisme isolasi jaringan. Virtual machine pada mode briged networking seolah-olah berdiri sendiri pada jaringan fisik. Pada mode ini isolasi yang dimungkinkan adalah dengan melakukan *filter MAC address* virtual machine saja. Karena keterbatasannya, mode bridged networking tidak tepat digunakan untuk mengisolasi guest dari sisi jaringan. Sedangkan untuk mencapai keamanan teknologi virtualisasi perlu diterapkan isolasi seperti firewall pada host [12]

Mode jaringan virtual lain yang bisa digunakan adalah *host-only networking* [3, 4]. Host-only networking adalah mode jaringan virtual dimana sistem operasi host dan sistem operasi guest dapat berkomunikasi secara langsung. Mekanisme akses ke jaringan luar pada mode ini dilakukan dengan memposisikan host sebagai router (opsional). Host-only networking dengan host yang berfungsi router sering disebut dengan *routed mode* [11].

Karena host bisa diposisikan sebagai router, maka host juga bisa berperan sebagai firewall. Pada sistem operasi berbasis kernel linux, firewall dapat dibangun dengan menggunakan framework Netfilter dan antarmuka IPTables.

Firewall yang dibangun harus mampu sejumlah fungsi filtering. Fungsi paling dasar adalah filtering berdasarkan *port* dan protokol. Selain itu firewall harus mampu memfilter paket yang bukan merupakan bagian koneksi legal walaupun menggunakan port dan protocol yang benar. Permasalahan yang lain adalah, aplikasi seperti VOIP/SIP menggunakan sejumlah port UDP secara dinamis.

Dampak positif yang biasa didapatkan dari adanya firewall ini adalah berkurangnya *vulneralbilities* yang terlihat dari jaringan luar. Selain itu informasi tentang guest yang berada dalam host yang mungkin didapatkan juga berkurang. Berkurangnya informasi ini dapat dilihat melalui OS fingerprinting

Selain dampak positif, firewall pada host juga berpotensi menimbulkan penurunan performansi akibat delay yang ditimbulkan oleh penanganan paket. Oleh karena itu perlu ada pengukuran terhadap penurunan performansi.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah :

1. Bagaimanakah cara mengimplementasikan netfilter sebagai firewall pada host untuk guest pada virtual machine?
2. Apakah sajakah yang difilter dari komunikasi antar guest? (parameter: permulaan koneksi, port yang dipakai, kecocokan antara port dan *state* dari aplikasi).
3. Seberapa besar keberhasilan pengambilan informasi melalui Port Scanning dan OS fingerprinting? (parameter: perubahan fingerprint).
4. Vulneralbilities apa sajakah yang mampu ditutupi oleh netfilter? (parameter: penyebab dan asal vulneralbilities)
5. Seberapa besar penurunan performansi yang dialami setelah diimplementasikannya firewall? (parameter: ratio performansi setelah diterapkan firewall dengan sebelum firewall).

1.3 Batasan Masalah

Permasalahan tersebut penulis teliti dengan batasan masalah sebagai berikut :

1. Framework *packet*-filtering yang digunakan adalah Netfilter/Iptables.
2. Host menggunakan kernel linux versi 2.4 keatas.
3. Guest system yang dibangun hanya menggunakan sebuah network interface.
4. Guest system yang akan diuji adalah HTTP server, database server, dan VoIP/SIP Server.

1.4 Tujuan

Tujuan yang ingin penulis capai dalam penelitian ini:

1. mengimplementasikan netfilter sebagai firewall pada host untuk guest pada virtual machine
2. Membatasi guest yang boleh memulai koneksi, port yang digunakan, dan state protocol tersebut.
3. Mengacaukan hasil Port Scanning dan mengacaukan hasil OS Fingerprinting sehingga hasilnya ambigu.
4. Menutup vulneralbilities yang bukan merupakan bagian dari aplikasi yang dizinkan firewall.
5. Mengetahui besarnya penurunan performansi setelah diimplementasikannya firewall.

1.5 Metodologi Penyelesaian Masalah

Metodologi yang digunakan untuk menyelesaikan masalah tersebut adalah dengan melakukan langkah-langkah sebagai berikut:

1. Studi Literatur dan Eksplorasi

Pada tahap ini dikumpulkan literatur mengenai:

a. Firewall, untuk mengetahui prinsip kerja firewall yang bekerja dengan cara memfilter paket-paket (packet filtering).
b. Netfilter dan IPtables, untuk mengetahui *connection tracking* pada netfilter dan pembuatan rule pada iptables.
c. Mode Jaringan pada virtualisasi, untuk mengetahui mode jaringan yang tepat dalam penerapan firewall pada host.

2. Implementasi Sistem

Sistem host direalisasikan dengan sistem operasi Slackware64 13.37. Kernel bawaan sistem operasi tidak digunakan pada penelitian. Kernel yang digunakan adalah Linux 3.0.3 yang dikonfigurasi manual.

Hypervisor virtual mesin yang digunakan adalah KVM. Sedangkan virtual interface digunakan adalah TAP device.

Penulisan rule pada sistem ini menggunakan iptables bawaan Slackware64 13.37.

3. Pengujian Sistem

Pada tahap ini dilakukan pengujian fungsionalitas firewall dengan port scanning, OS fingerprinting, dan vulneralbilities scanning.

Penurunan performansi dilakukan dengan cara mengukur waktu yang dibutuhkan sistem untuk melakukan sejumlah tugas secara paralel. Tugas tersebut antara lain: panggilan VOIP/SIP dan request HTTP.

4. Pengambilan Kesimpulan dan Pembuatan Laporan

Pada tahap ini dilakukan penarikan kesimpulan terhadap hasil pengujian sistem dan mendokumentasikan proses dan hasil penelitian dalam buku TA.