Projek ini dihantar kepada Fakulti Sains Komputer dan Teknologi Maklumat, Universiti Malaya
Dalam memenuhi keperluan Ijazah Sarjana Muda Sains Komputer

Packet Filtering Enhancement - Iptables

Oleh :
Noor Maizatulshima Bt Muhammad Sabri
(WEK000160)

Di bawah seliaan
Puan Fazidah Bt Othman
Fakulti Sains Komputer dan Teknologi Maklumat
Universiti Malaya

Moderator
Dr. Mazliza
Fakulti Sains Komputer dan Teknologi Maklumat
Universiti Malaya

Abstrak

Walaupun kita mempercayai bahawa adalah penting bagi pengurus sistem untuk berhati – hati dan peka dengan isu – isu berkenaan keselamatan agar menjadi lebih teliti, namun kita juga perlu menyedari bahawa usaha ini masih tidak mampu untuk melindungi *site* daripada pendedahan terhadap protokol yang belum lengkap dalam memastikan keselamatan sistem. Tidak kira juga bagaimana teknologi Internet berkembang pada masa akan datang, perkara yang perlu diberi penekanan utama terhadap capaian dan kelajuan ialah keselamatan. Kadangkala semua orang tidak menyedari bahawa sekali mereka memasuki dunia Internet bermakna mereka telah memasuki kawasan pengalamatan IP yang mempunyai baik dan buruknya masing – masing. Apabila kita berada di Net, sebenarnya kita telah terdedah kepada banyak jenis risiko, sesetengahnya tanpa kita sedari.

Selaras dengan ledakan maklumat dan komputer yang semakin canggih, isu keselamatan komputer – komputer peribadi dan rangkaian menjadi sangat penting dalam sesebuah organisasi. Tidak ketinggalan juga para penggodam telah berusaha mempertingkatkan serta menyusun strategi –strategi baru untuk menyerang sistem dan rangkaian sesebuah organisasi, samaada bertujuan untuk membalas dendam, mencuri maklumat, mengaut keuntungan ataupun menguji kekebalan rangkaian organisasi tersebut.

Teknologi *firewall* menyediakan peralatan paling efektif kepada organisasi dalam mengurangkan risiko pengurusan rangkaian dengan menyediakan mekanisme kawalan

yang mampu melaksanakan polisi keselamatan yang kompleks. *Firewall* yang mempunyai beberapa teknik perlaksanaan adalah dibangunkan untuk tujuan melindungi sistem dalaman dari serangan luaran. Tanpa *firewall* sesebuah organisasi tidak mampu untuk menghalang trafik – trafik yang tidak sah masuk ke dalam sistem komputer dan melakukan pencerobohan terhadap aset – aset maklumat. Bentuk – bentuk capaian ini akan menyebabkan kegagalan kepada kebolehsediaadaan perkhidmatan, menyebabkan pendedahan terhadap infrastruktur sistem dan vandalisme terhadap perkhidmatan maklumat seperti laman web organisasi.

Teknik penapisan paket adalah teknik yang digunakan memandangkan penggunaannya yang agak meluas dipasaran. Teknik ini melindungi rangakaian melalui pengujian paket – paket yang melalui *firewall* kemudian menentukan samaada trafik ini adalah dibenarkan, dihalang atau di*drop*kan daripada melalui *firewall* dengan berpandukan kepada polisi keselamatan organisasi yang terlibat.

Penghargaan

Dengan nama Allah Yang Maha Pemurah Lagi Maha Mengasihani. Selawat dan salam buat junjungan mulia Baginda Nabi S.A.W, kaum keluarga serta sahabat - sahabat baginda.

Setinggi – tinggi kesyukuran kehadrat Allah S.W.T kerana dengan limpahan rahmat dan izin – Nya dapat saya menyempurnakan Projek Ilmiah Tahap Akhir ini. Semoga dengan segala penat jerih dan pengorbanan dalam menyiapkan projek ini akan membuahkan kejayaan yang cemerlang, Insyaallah.

Pertama sekali ingin saya menghadiahkan sekalung penghargaan buat mama, abah dan adik – adik tercinta di atas kiriman doa serta sokongan material dan spiritual yang tidak pernah putus. Semoga kita sekeluarga akan kekal berada dalam rahmat Allah dunia akhirat, Insyaallah.

Seterusnya yang dihormati Puan Fazidah binti Othman, selaku penyelia saya yang banyak membantu memberikan tunjuk ajar, khidmat nasihat dan sokongan beliau. Tanpa sokongan ini sudah pasti adalah agak sukar bagi saya untuk menyiapkan Projek Ilmiah Tahap Akhir ini. Serta yang dihormati Dr. Mazliza, selaku moderator saya untuk Projek Ilmiah Tahap Akhir ini. Didoakan agar keduanya sentiasa diredhai Allah hendaknya. Terima kasih atas ilmu yang dicurahkan.

Seterusnya buat pensyarah – pensyarah yang sentiasa sedia membantu ; Encik Zaidi Razak, Encik Badrul dan Mr Bukhori dari MMU. Jutaan terima kasih di atas segala tunjuk ajar dan pendapat.

Buat rakan – rakan seperjuangan yang tak pernah meninggalkan saya diwaktu susah dan senang. Terima kasih yang tidak terhingga kerana sokongan kalian banyak membantu saya menyempurnakan Projek ilmiah Tahap Akhir ini.

Kepada Allah didoakan agar kita semua dilimpahkan rahmat dan sejahtera, bahagia dunia akhirat, Insyaallah

Terima kasih.

ISI KANDUNGAN

SENARAI RAJAH

SENARAI JADUAL

BAB 1: PENGENALAN

BAB 1 : PENGENALAN

1.1 Latarbelakang Projek

Para pekerja CERT (*Computer Emergency Reason Team*) telah menggalakkan semua pengurus – pengurus sistem, pengurus – pengurus rangkaian bahagian dan pembekal – pembekal yang berkaitan dengan rangkaian meluangkan sedikit masa untuk memahami isu – isu yang berkaitan dengan penapisan paket (*packet filtering*).

Hal ini adalah disebabkan oleh kekurangan dalam beberapa perkhidmatan TCP / IP, maka bahagian ini mestilah berkemampuan untuk menghalang sebarang capaian dari luar kepada perkhidmatan sistem. Bahagian ini juga perlu mempertimbangkan pembelian penghala yang telah diprogramkan. Pembekal – pembekal seharusnya menawarkan penapisan paket sebagai pilihan kepada perkhidmatan yang sedia ada.

Projek latihan ilmiah ini adalah dibangunkan untuk mengatasi kelemahan yang wujud dalam persekitaran penapisan paket semasa. Pembangunan dan peningkatan bagi teknik penapisan paket *firewall* ini akan dilakukan terhadap projek pelajar sebelum ini yang lebih menekankan kepada analisis konsep, rekabentuk dan perlaksanaan sistem *firewall*. Teknik penapisan paket ini seharusnya mampu melindungi sesebuah organisasi daripada sebarang ancaman penggodam yang boleh membawa kepada kemusnahan sistem. Beberapa kelemahan dalam teknik penapisan paket ini telah dikenalpasti. Di antara kelemahan – kelemahan yang wujud ialah seperti mudah dipengaruhi oleh penyamaran IP (*IP spoofing*), kurang selamat dan tidak mampu melindungi sistem dari ancaman dalaman.

Dalam mengatasi kelemahan – kelemahan di atas, cadangan bagi mengatasinya telah diperolehi hasil maklumat yang dilakukan melalui kaedah temubual, carian melalui Internet, buku – buku rujukan, majalah –majalah komputer dan ilmiah serta hasil perbincangan dengan pensyarah. Di antara cadangan yang telah dikenalpasti ialah melalui peningkatan penggunaan dari *ipchains* kepada *iptables*, melalui penggunaan penapisan paket secara dinamik, penguatkuasaan polisi serta melakukan penapisan pada soket atau port tertentu.

Jadi berdasarkan kepada tajuk Projek Ilmiah Tahap Akhir 2 ini, saya telah memilih cadangan untuk meningkatkan penggunaan dari *ipchains* kepada *iptables* yang lebih mantap dan terkini. Matlamat projek ini adalah untuk memberikan perkhidmatan keselamatan alternatif bagi sesebuah organisasi yang sederhana dan kecil terutamanya tetapi masih mampu dihubungkan kepada dunia luar Internet dalam persekitaran yang lebih selamat.

Melalui projek ini juga penganalisaan terhadap objektif, rekabentuk, sasaran dan pelaksanaan bagi peningkatan ini telah dilakukan. Perlindungan sistem menggunakan *iptables* hanya melibatkan kos yang rendah dan mudah untuk dikonfiguasikan oleh pentadbir sistem dan rangkaian khususnya.

Maka peningkatan ini seharusnya mampu mengurangkan risiko serangan yang akan berlaku melalui pengesanan awal oleh pentadbir sistem kerana ia meliputi

pengesanan awal terhadap pencerobohan, kawalan kepada keutuhan dan mengesan sebarang percubaan yang mencurigakan.

1.2 Motivasi Projek

Projek ini memfokuskan kepada peningkatan kepada *iptables* yang dilaksanakan di dalam teknik penapisan paket bagi *firewall*. Mekanisme ini dilaksanakan dalam Red Hat Linux 7.3 dengan versi *kernel* 2.4.

Selain itu juga penggunaan *iptables* ini dapat membantu menguatkuasakan polisi keselamatan yang telah ditetapkan dalam organisasi. Polisi ini merangkumi kesemua maklumat yang berkenaan dengan kawalan kepada capaian dan pengesahan kepada penggunaan perkhidmatan yang bertujuan untuk melindungi data, fail dan aturcara daripada pengguna yang tidak sah.

Kumpulan Tindakbalas Kecemasan Komputer Malaysia (MyCERT) baru – baru ini telah mengeluarkan satu rumusan yang menerangkan jenis – jenis serangan yang telah dilaporkan, juga sebagai satu isu yang patut diberi perhatian dan kekebalan terhadap sesuatu maklumat.

Tambahan pula ancaman dari penggodam telah berkembang dengan pesat. Hal ini adalah merupakan hasil kajian daripada Bill Hancock, iaitu *Vice President of Cable and Wireless* di mana pertambahan yang berlaku adalah dari sejumlah 2000 pada tahun 2001 dan kemudiannya bertambah kepada 86,000 pada tahun berikut.

MyCERT juga telah menyarankan kepada organisasi – organisasi serta pemilik komputer persendirian yang merupakan sasaran utama melakukan penganalisaan terhadap port dengan menambahkan *firewall* persendirian ke dalam sistem masing – masing. Hal ini bertujuan untuk menjadikan pemilik komputer dan sistem agar lebih peka terhadap mana – mana penganalisaan yang tidak sah kepada mesin – mesin komputer, serta menghalang alamat IP yang mencurigakan.

Terdapat beberapa faktor yang mendorong kepada peningkatan *ipchains* kepada *iptables* :

1. Setiap organisasi kecil mahupun besar berisiko tinggi untuk diserang oleh penggodam sama ada secara serius atau tidak
2. Peningkatan ini juga adalah sesuai untuk organisasi bersaiz sederhana dan kecil kerana kos pembelian dan konfigurasinya yang rendah
3. Memberikan tambahan pilihan *(alternative options)* kepada perkhidmatan keselamatan rangkaian di pasaran
4. Merupakan pilihan yang baik kerana mudah dikonfigurasikan dan diselenggarakan

Namun demikian teknik ini masih tidak mampu melindungi sistem dari ancaman penggodam kerana masih terdapat banyak lagi teknik – teknik canggih penggodam yang mampu berkompromi dengan *firewall* itu sendiri. Tetapi sekurang –kurangnya mekanisme ini boleh membantu pentadbir sistem mengurangkan risiko serangan.

1.3 Objektif projek

Objektif bagi projek ini adalah seperti yang berikut :

1. Mengenalpasti punca – punca kelemahan yang wujud dalam teknik penapisan paket bagi smartfw yang merupakan laluan utama penggodam untuk membuat serangan
2. Mendapatkan langkah – langkah bagi mangatasi kelemahan – kelemahan yang wujud. Analisis berkenaan teknik *firewall* yang baru telah diperkenalkan iaitu peningkatan dari *ipchains* kepada *iptables*. Dengan pengubahsuaian ini maka keselamatan sesebuah rangkaian dapat dipertingkatkan
3. Memberikan cadangan dan pendapat – pendapat baru dalam meningkatkan perkhidmatan yang sedia ada. Cadangan pengubahsuaian ini merangkumi rekabentuk, analisis, cara diimplementasi dan penerangan konsep
4. Membuat analisa perbandingan di antara persekitaran penapisan paket yang sedia ada dengan pengubahsuaian yang dilakukan
5. Menunjukkan kaedah pembangunan dan penyediaan kepada peningkatan yang dilakukan ini
6. Meningkatkan keselamatan dalam sesebuah organisasi supaya lebih efektif, efisyen dan dipercayai. Selain itu mekanisme ini merupakan satu pilihan alternatif kepada mana – mana organisasi kerana melibatkan penggunaan kos yang rendah dan mudah dikonfigurasikan
7. Meningkatkan kepekaan kepada pentadbir sistem supaya melakukan pemeriksaan terhadap *log file* untuk mengelakkan serangan dari peringkat awal berlaku

1.4 Skop Projek

Skop yang terlibat dalam projek peningkatan dari versi ipchains kepada iptables adalah seperti yang berikut :

1. Menunjukkan aktiviti – aktiviti log fail seperti bilangan paket, *bytes*, destinasi, sumber, port dan protokol
2. Tidak mampu menghalang ancaman dari sistem dalaman
3. *Rules* dimasukkan oleh pentadbir mengikut kesesuaian dan rujukan dari aktiviti sebelum ini
4. Pentadbir boleh memeriksa pada mana – mana port yang mencurigakan sebagai contoh pada polisi *DROP* akan dirujuk kepada bilangan paket dan bytes samaada banyak atau sedikit yang melalui smartfw tersebut

1.5 Definisi Projek

1.5.1 Peningkatan Yang Dilakukan Terhadap Teknik Penapisan Paket Bagi *Firewall* : Menukar Dari Penggunaan Ipchains Kepada Iptables

Beberapa peningkatan dan pengubahsuaian dari projek pelajar terdahulu telah dilakukan bagi meningkatkan keselamatan organisasi. Beberapa kelemahan yang wujud pada persekitaran teknik penapisan paket yang sedia ada juga telah dikenalpasti.

Iptables adalah digunakan sebagai satu peningkatan dalam smartfw kerana *iptables* mempunyai beberapa ciri –ciri tambahan yang lebih baik berbanding *ipchains*.

1.6 Hasil yang dijangkakan

Hasil yang dijangkakan daripada peningkatan versi *ipchains* kepada *iptables* adalah seperti yang dinyatakan di bawah.

1. Seharusnya pendedahan terhadap serangan dapat diminimumkan di dalam sesebuah organisasi yang menggunakan mekanisme ini
2. Meningkatkan kebolehpercayaan pengguna terhadap sistem dari segi kekebalan maklumat sensitif yang tidak boleh dicapai oleh pengguna yang tidak sah, juga pemeriksaan yang kerap dilakukan oleh pentadbir sistem atau rangkaian ke atas sistem *firewall* apabila capaian yang mencurigakan dikesan
3. Mutu kerja di dalam organisasi dapat dipertingkatkan kerana kurang berlakunya risiko serangan
4. Dapat memenuhi keperluan pengguna dalam meningkatkan keselamatan dalam organisasi. Projek ini merangkumi maklumat dan prosedur bagaimana *iptables* ini berfungsi sebagai salah satu cara *firewall* melindungi rangkaian persendirian
5. Mampu beroperasi 24 jam sehari dan mudah dikonfigurasikan. Projek ini juga termasuklah dari segi pemilihan *tools* dan keperluan dalam membangunkan projek ini
6. Penganalisaan terhadap peningkatan ini di bawah teknik penapisan paket merangkumi konsep, rekabentuk, perlaksanaan, senibina dan maklumat – maklumat lain yang berkaitan dengan sistem *firewall*

1.7 Jadual Perjalanan Projek

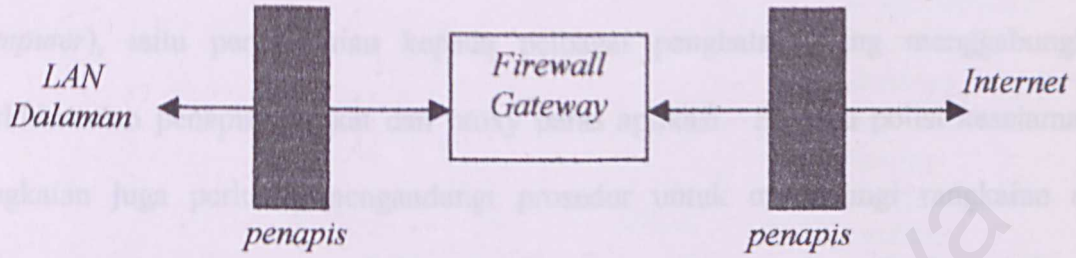
Jadual 1.1 : Carta Gantt Bagi Projek Mekanisme Pra – Amaran

Bil	Aktiviti	Bulan Mei	Bulan Jun	Bulan Julai	Bulan Ogos	Bulan September
1.	Memahami Keperluan dan Mula Mengumpul bahan					
2.	Bab 4– Rekabentuk Sistem					
3.	Bab 5- Analisis Sistem					
4.	Kajian Khusus Terhadap Projek					
5.	Bab 6- Pelaksanaan Sistem					
6.	Bab 7- Pengujian Sistem					
7.	Bab 5 – Penilaian Sistem					

BAB 2 : KAJIAN LITERASI

BAB 2 : KAJIAN LITERASI

2.1 Definisi *Firewall*



Rajah 2.1 : Fungsi Asas *firewall*

Firewall adalah satu sistem yang direkabentuk untuk menghalang capaian yang tidak sah ke atau dari rangkaian persendirian serta menyediakan perlindungan asas daripada ancaman luaran terhadap sistem rangkaian komputer bagi sesebuah organisasi, contohnya seperti intranet.

Biasanya *firewall* terdiri daripada kombinasi perkakasan dan perisian yang digunakan untuk melaksanakan polisi keselamatan yang telah ditetapkan melalui trafik rangkaian. Trafik ini berlaku di antara dua atau lebih rangkaian, yang mana salah satunya di bawah kawalan organisasi.

Firewall berfungsi dengan menilai setiap paket data di antara komputer dan Internet serta membuat keputusan berdasarkan kriteria keselamatan yang telah ditetapkan oleh organisasi sama ada untuk halang, biar atau lepas (*permit, block, ignore*). Sesetengah

firewall juga menyimpan log untuk pentadbir sistem atau rangkaian memantau dan melihat aktiviti yang telah berlaku.

Firewall boleh menjadi seringkas – ringkasnya, seperti penghala yang akan menapis paket – paket ataupun menjadi lebih kompleks seperti pelbagai komputer (*multi computer*), iaitu penyelesaian kepada pelbagai penghala yang menggabungkan perkhidmatan penapisan paket dan proxy paras aplikasi. Sesuatu polisi keselamatan rangkaian juga perlulah mengandungi prosedur untuk melindungi rangkaian dari sebarang serangan yang menyebabkan kehilangan atau kemusnahan. Terdapat beberapa jenis teknik *firewall* :

i. Penapis paket – Melihat kepada setiap paket yang memasuki atau meninggalkan rangkaian, dan menerima atau menolaknya berdasarkan kepada peraturan pengguna yang tertakrif.

ii. Aplikasi *gateway* – Menggunakan mekanisma keselamatan terhadap aplikasi yang spesifik, contohnya seperti pelayan FTP dan TELNET.

iii. *Circuit level gateway* – Menggunakan mekanisma keselamatan apabila sambungan TCP atau UDP diperkukuhkan.

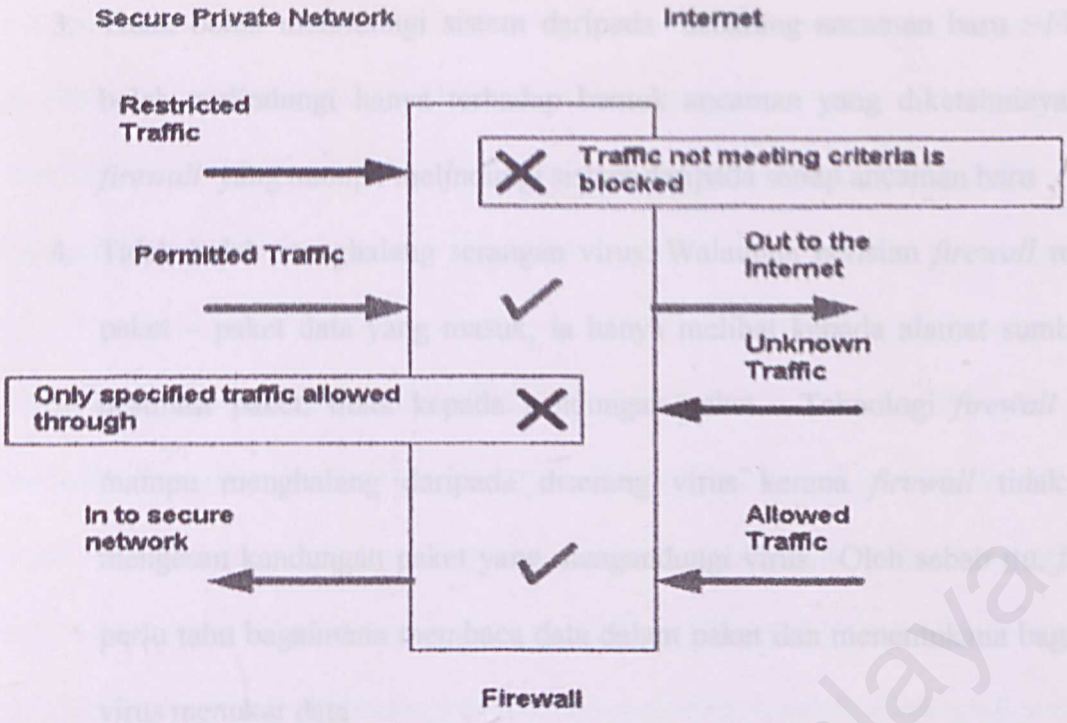
iv. Pelayan proxy – Membuat pintasan terhadap mesej yang memasuki atau meninggalkan rangkaian

Jadual berikut menunjukkan jenis *firewall* yang wujud dan penerangan yang berkenaan dengannya :

Jenis firewall	Definisi	Kebaikan	Keburukan
Perisian	Dilarikan pada sistem komputer . Proses ini	- Murah - Mudah	- Setelah dilarikan pada sistem

	akan memintas dan menghalang setiap permohonan rangkaian serta menentukan samaada permohonan ini adalah sah atau tidak	dikonfigurasikan	komputer, *firewall* ini akan menuntut sumber – sumber (ruang CPU, ingatan, cakera) daripada sistem - Menunjukkan ketidaksesuaian terhadap sistem pengoperasian - Memerlukan versinya yang tepat dan betul untuk sistem pengoperasian syarikat - Perlu membeli satu salinan untuk setiap sistem dalam rangkaian
Perkakasan	Secara umumnya adalah merupakan kotak kecil yang ditempatkan di antara komputer dan modem	- Menyediakan perlindungan yang lebih lengkap - Mampu melindungi lebih dari satu sistem pada satu masa - Tidak memberi kesan kepada perlaksanaan sistem - Tidak bergantung kepada sistem pengoperasian dan aplikasi	- Mahal, namun jika kita mempunyai bilangan mesin yang agak banyak untuk dilindungi, ia boleh mengurangkan kos pembelian pada satu *firewall* perkakasan berbanding bilangan salinan yang diperlukan bagi perisian - Selagi ia tidak dilarikan di dalam komputer, maka adalah sukar untuk dikonfigurasikan

Jadual 2.1 : Jenis – jenis *Firewall*



Rajah 2.2 : Bagaimana *Firewall* Berfungsi

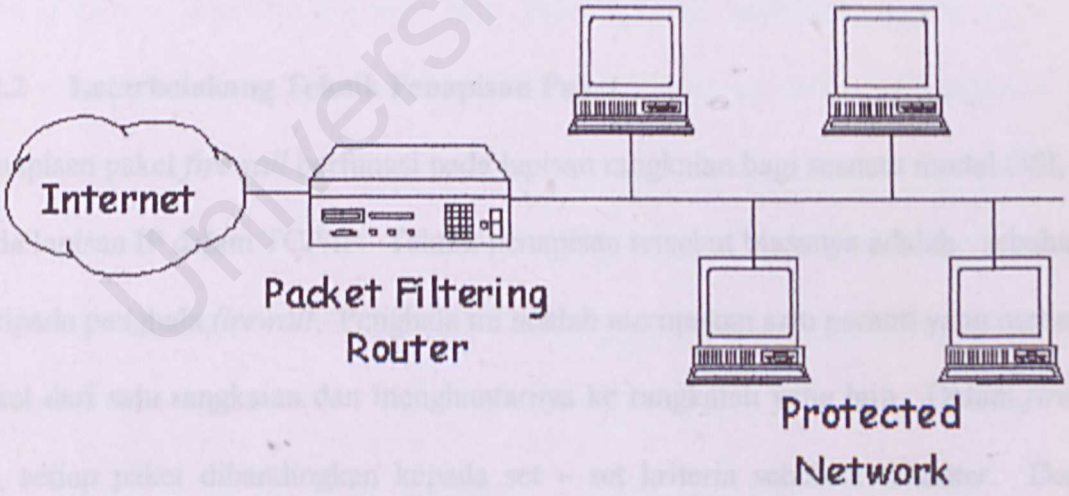
2.1.1 Had – had bagi *firewall*

1. Tidak mampu melindungi sistem daripada serangan dalaman : Pengguna sah boleh mencuri data, membuat serangan ke atas perkakasan atau perisisan, dan menjana serangan tanpa perlu berinteraksi dengan *firewall*. Untuk mengatasinya, organisasi perlu membangunkan dan menguatkuasakan polisi keselamatan dalaman
2. Tidak boleh menghalang trafik yang tidak dikawal : *Firewall* tidak boleh mengawal trafik rangkaian yang melaluinya. *Firewall* adalah direkabentuk untuk mengawal pengguna berbanding sistem

3. Tidak boleh melindungi sistem daripada sebarang ancaman baru : *Firewall* boleh melindungi hanya terhadap bentuk ancaman yang diketahuinya, tiada *firewall* yang mampu melindungi sistem daripada setiap ancaman baru
4. Tidak boleh menghalang serangan virus: Walaupun perisian *firewall* meneliti paket – paket data yang masuk, ia hanya melihat kepada alamat sumber dan destinasi paket, tidak kepada kandungan paket. Teknologi *firewall* tidak mampu menghalang daripada diserang virus kerana *firewall* tidak boleh mengesan kandungan paket yang mengandungi virus. Oleh sebab itu, *firewall* perlu tahu bagaimana membaca data dalam paket dan menentukana bagaimana virus menukar data

2.2 Teknik Penapisan Paket

2.2.1 Definisi Penapisan Paket



Rajah 2.3 : Fungsi Penapisan Paket *Firewall*

Penapis paket adalah merupakan teknologi *firewall* yang asas, yang digunakan untuk melindungi rangkaian persendirian daripada gangguan luar. Penapisan paket menyediakan keselamatan lapisan rangkaian (model OSI) untuk mengawal jenis – jenis maklumat yang dihantar di antara rangkaian dan host.

Penapisan paket juga adalah teknik yang wujud di antara rangkaian dalaman dengan dunia luar Internet. Pelayan dan pelanggan dihubungkan secara terus, namun demikian paket – paket yang dihantar akan melalui penapisan paket yang menghubungkan rangkaian dalaman dengan dunia luar. Apabila paket – paket mula dihantar, penapis paket akan membuat perbandingan berdasarkan peraturan paket. Jika konfigurasi yang dibangunkan adalah membenarkan laluan paket, maka paket boleh meneruskan penghantaran melalui rangkaian kepada hop yang berkenaan. Manakala jika konfigurasi tidak membenarkan paket melaluinya, ia akan dibuang. Kadangkala terdapat beberapa *firewall* akan menghantar notis kepada sumber.

2.2.2 Latarbelakang Teknik Penapisan Paket

Penapisan paket *firewall* berfungsi pada lapisan rangkaian bagi sesuatu model OSI, atau pada lapisan IP dalam TCP/IP. Teknik penapisan tersebut biasanya adalah sebahagian daripada penghala *firewall*. Penghala ini adalah merupakan satu peranti yang menerima paket dari satu rangkaian dan menghantarnya ke rangkaian yang lain. Dalam *firewall* ini, setiap paket dibandingkan kepada set – set kriteria sebelum dihantar. Dengan merujuk kepada paket – paket dan kriteria – kriteria yang ditetapkan, *firewall* boleh

mengabaikan paket, menghantarnya ke destinasi atau menghantar mesej balik kepada penghantar.

Boleh diandaikan aktiviti penapisan paket yang keluar atau masuk akan mengatur aliran data berdasarkan kepada beberapa kriteria seperti jenis perkhidmatan, bilangan port, bilangan antaramuka, alamat sumber dan alamat destinasi. Sebagai contoh kita boleh menganggap penapis paket adalah untuk menghalang trafik rangkaian yang tidak diperlukan berdasarkan kepada alamat asal (atau julat bagi alamat) atau jenis paket (seperti email atau FTP – *file transfer protocol*). Keadaan ini terutamanya untuk menghalang perkhidmatan atau tawaran oleh SPAM daripada masuk kedalam sistem rangkaian dan memperlahankan larian atau melumpuhkan perkhidmatan rangkaian.

Kebiasaannya, perbezaan yang ditunjukkan oleh penapis paket termasuklah alamat sumber, port sumber, alamat destinasi dan port destinasi. Penapisan yang dilakukan pada alamat sumber dan destinasi akan membenarkan kawalan terhadap siapa yang akan berkomunikasi dengan rangkaian dalaman. Kesemua trafik dari rangkaian yang tidak dijangkakan akan disenaraikan. Port dengan kata lainnya, adalah digunakan untuk membezakan perkhidmatan dalam rangkaian. Dengan melakukan penapisan pada port, adalah tidak mustahil untuk menafikan sebarang capaian dari dunia luar terhadap perkhidmatan yang ditawarkan oleh rangkaian dalaman.

2.2.3 Aplikasi Bagi Teknik Penapisan Paket

- Apabila suatu penghala penapisan paket menerima paket yang mempunyai nombor port destinasinya 23 (iaitu TELNET) ia akan menghalang paket bergantung kepada sumber bagi paket atau sistem mana paket akan dihantar
- Satu blok diagram yang masuk atau keluar dengan medan protokol IPnya = 17 dan sumber lain atau port destinasinya = 23 bermaksud kesemua aliran UDP yang masuk dan keluar serta sambungan TELNET adalah dihalang
- Menghalang segmen TCP mengikut had dengan ACK = 0 atau dengan sistem bit SYN dan bukan set bit ACK. Penapis paket akan menghalang pengguna dari luar daripada membuat sambungan TCP dengan pengguna di dalam, tetapi membenarkan pengguna dalaman dihubungkan keluar

2.2.4 Fungsi – fungsi Teknik Penapisan Paket

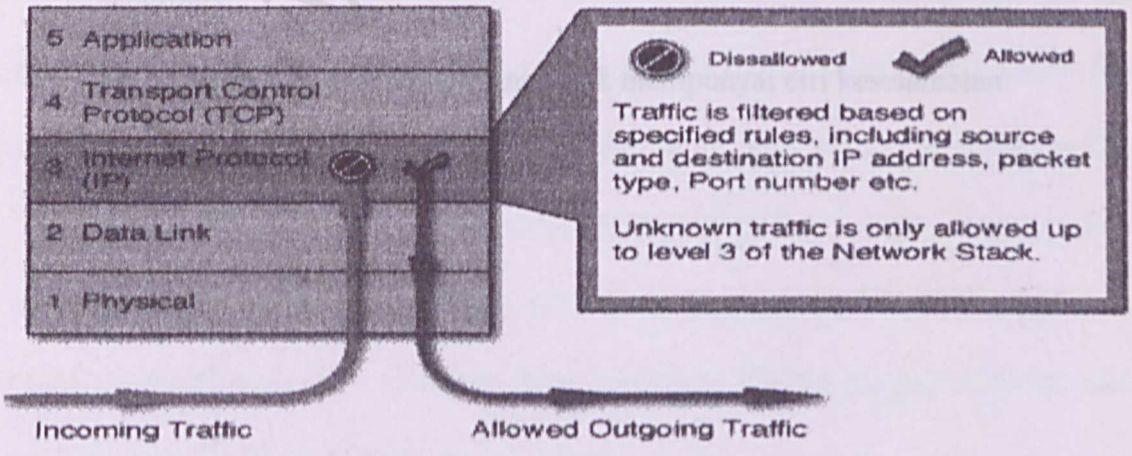
- Menghantar paket – paket ke destinasi yang ditentukan
- Menolak mana – mana paket dan memaklumkan kepada penghantar
- Atau *drop* paket tanpa memaklumkan kepada penghantar
- Menerima log atau menafikan maklumat paket
- Mempunyai NAT (*network address translation*) untuk menterjemahkan alamat IP awam kepada alamat IP persendirian pada LAN persendirian

2.2.5 Senibina Teknik Penapisan Paket

Senibina penapisan paket menunjukkan analisis untuk satu atau lebih protokol rangkaian dengan menggunakan set peraturan yang terhad. Paket – paket akan masuk

ke dalam rangkaian yang dipercayai dan dibuat perbandingan dengan peraturan yang telah ditentukan daripada set peraturan bagi satu atau lebih protokol seperti IP, TCP atau ICMP. Paket – paket adalah samaada diterima dan dihantar ke timbunan rangkaian (*network stack*) untuk penghantaran atau dinafikan capaiannya. Jika paket – paket adalah mematuhi kesemua peraturan penapisan paket, paket samaada akan dihantar ke timbunan rangkaian untuk pemprosesan berikutnya atau dihantar ke host rangkaian. Set peraturan telah dikekalkan di dalam TCP/IP kernel.

Set peraturan akan digunakan dari mula penapis paket tidak memahami keselamatan protokol lapisan aplikasi yang digunakan di dalam paket komunikasi. Set peraturan ini mengandungi tindakan – tindakan yang akan dilaksanakan pada mana – mana paket yang berpadanan dengan kriteria dalam set peraturan. Set peraturan ini juga mengandungi senarai penafian dan senarai yang dibenarkan yang akan dikekalkan di dalam kernel. Paket rangkaian akan melalui kedua – dua senarai penafian dan yang dibenarkan jika ia perlu dihalakan ke destinasi yang betul. Paket – paket sepatutnya tidak boleh dinafikan atau dibenarkan terlalu cepat.



Rajah 2.4 : Set peraturan teknik penapisan paket dalam TCP/IP

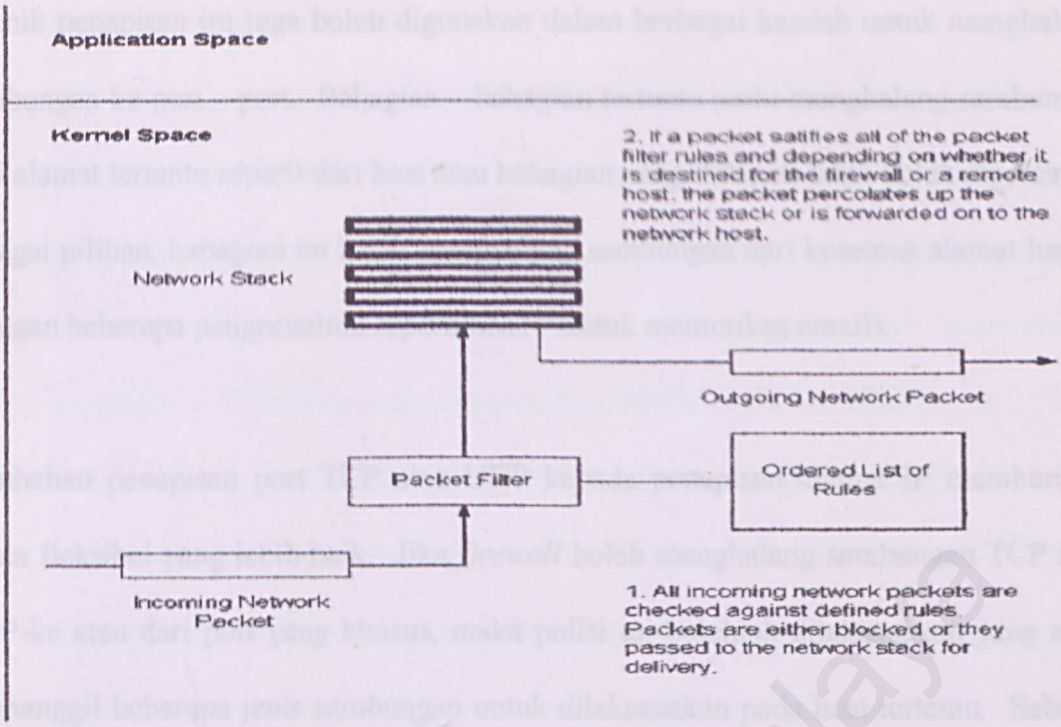
2.2.6 Perlaksanaan Teknik Penapisan Paket

Penapisan paket adalah merupakan teknik yang terbaik untuk membuat pilihan dalam membenarkan atau menghalang capaian ke pelbagai perkhidmatan yang berbeza.

Teknik ini juga adalah merupakan teknik termudah *firewall*. Caranya ialah pengguna akan menghalakan secara terus paket - paket yang datang dari Internet ke port – port yang selamat. Teknik penapisan paket biasanya dilaksanakan dengan menggunakan penapisan yang dibina di dalam penghala Internet. Ini bermakna tiada capaian yang dibenarkan untuk port di bawah 1024 kecuali untuk sambungan khusus yang tertentu kepada perkhidmatan yang selamat seperti SMTP, NNTP, DNS, FTP dan HTTP.

Kebaikan perlaksanaannya ialah capaian akan dinafikan untuk perkhidmatan yang merbahaya dan yang disyaki seperti berikut :

- *Finger* – memberikan senarai pengguna yang log – in, dan dalam proses memberitahu *'Bad Guy'* apa yang perlu mereka log – in sendiri
- *Exec* – membenarkan *'Bad Guy'* untuk melarikan aturcara jauh (*remote program*)
- TFTP – protokol pindah fail yang tidak mempunyai ciri keselamatan



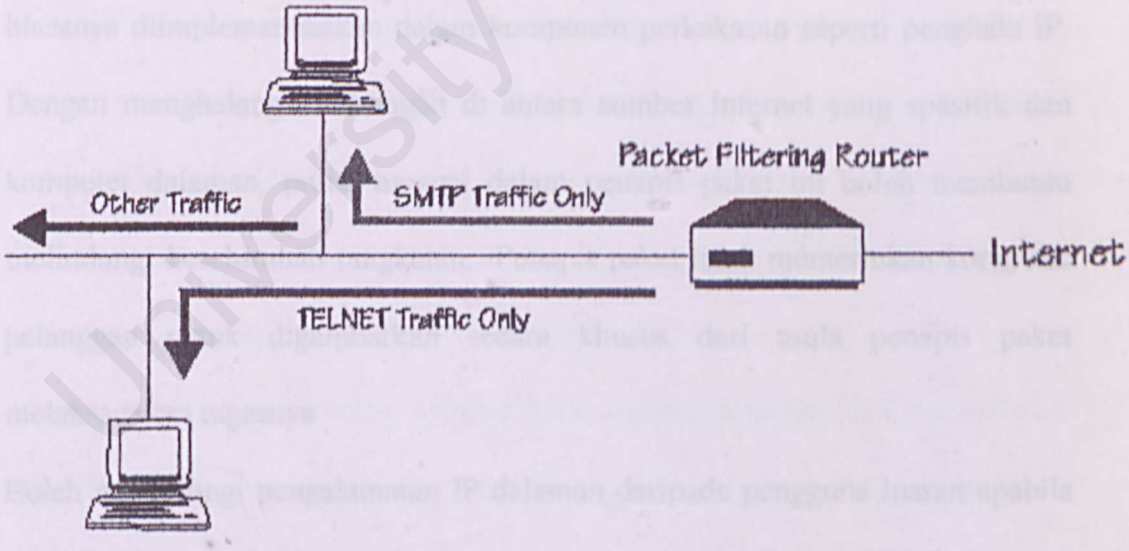
Rajah 2.5 : Perlaksanaan Teknik Penapisan Paket

2.2.7 Pertimbangan Terhadap Isu Keselamatan Teknik Penapisan Paket

Penapisan paket adalah kaedah asas bagi keselamatan parameter. Paket akan di di*rop*kan atau ditapis oleh peranti, biasanya penghala atau *firewall* apabila kedua – dua ini tidak menemui sebarang penentuan peraturan atau polisi yang telah disetkan oleh pentadbir rangkaian. Penapisan paket mempertimbangkan alamat sumber dan destinasi bagi paket – paket dan jenis protokol yang terdapat di dalam paket. Jadi penapis paket perlu *drop* atau buang mana – mana paket yang datang dari sumber Internet atau yang tidak dipercayai oleh penghala yang mengandungi protokol Telnet. Protokol Telnet boleh digunakan untuk menyediakan kawalan jauh bagi *workstation*, pelayan atau peranti rangkaian yang lain. Capaian Telnet yang tidak disekat daripada Internet adalah merupakan isu keselamatan yang paling utama.

Teknik penapisan ini juga boleh digunakan dalam berbagai kaedah untuk menghalang sambungan ke port – port. Bahagian – bahagian tertentu perlu menghalang sambungan dari alamat tertentu seperti dari host atau bahagian yang didapati tidak boleh dipercayai; sebagai pilihan, bahagian ini boleh menghalang sambungan dari kesemua alamat luaran (dengan beberapa pengecualian seperti SMTP untuk memeriksa email).

Tambahan penapisan port TCP atau UDP kepada penapisan alamat IP memberikan kesan fleksibel yang lebih baik. Jika *firewall* boleh menghalang sambungan TCP atau UDP ke atau dari port yang khusus, maka polisi ini bolehlah dilaksanakan yang akan memanggil beberapa jenis sambungan untuk dilaksanakan pada host tertentu. Sebagai contoh pada satu bahagian mungkin perlu menghalang kesemua sambungan yang masuk ke host kecuali untuk beberapa sistem yang dihubungkan secara terus kepada *firewall*.



Rajah 2.6 : Contoh Aplikasi Yang Selamat Dalam Telnet dan SMTP

2.2.8 Kelebihan Penapisan Paket

Terdapat beberapa kelebihan yang wujud bagi teknik penapisan paket :

1. Melibatkan kos yang rendah terhadap perlaksanaannya kerana kebanyakan penghala adalah menyokong teknik penapisan paket. Kebanyakannya juga semua sambungan Internet yang berkelajuan tinggi adalah memerlukan penghala. Oleh sebab itu organisasi yang mempunyai sambungan kepada Internet yang berkelajuan tinggi telahpun mempunyai kemampuan untuk melaksanakan teknik penapisan paket asas pada penghala tanpa perlu membeli perkakasan dan perisian baru. NAT (*network address translation*) penghala juga telah menawarkan kemampuannya untuk melindungi alamat IP bagi komputer di belakang *firewall*.
2. Merupakan teknologi *firewall* yang terpantas semenjak ia mula diperkenalkan dan pemprosesannya yang tidak kompleks berbanding teknologi lain. Ia biasanya diimplementasikan dalam komponen perkakasan seperti penghala IP. Dengan menghalang sambungan di antara sumber Internet yang spesifik dan komputer dalaman, polisi tunggal dalam penapis paket ini boleh membantu melindungi keseluruhan rangkaian. Penapis paket tidak memerlukan komputer pelanggan untuk digambarkan secara khusus dari mula penapis paket melaksanakan tugasnya
3. Boleh melindungi pengalamatan IP dalaman daripada pengguna luaran apabila penapis paket digunakan untuk dihubungkan dengan NAT
4. Pelaksanaannya yang baik, iaitu mempunyai NAT dan PAT (*port address translation*) yang memberikan keselamatan yang lebih baik. Ia tidak

memerlukan pengubahsuaian kepada pelanggan, tetapi meningkatkan keselamatan dengan menggunakan penapisan paket dinamik dan penyemakan yang baik (*stateful inspection*)

2.2.9 Kelemahan – Kelemahan Teknik Penapisan Paket

Terdapat juga beberapa kelemahan yang wujud dalam teknik ini :

1. Penapis paket tidak mampu untuk membezakan nama pengguna (*user name*)
2. Mudah dipengaruhi atau berkompromi dengan penyamaran IP (*IP spoofing*)
3. Secara umumnya penapis paket tidak memahami bagaimana untuk memproses maklumat pada protokol paras tinggi seperti FTP (*file transfer protocol*) kerana penapis paket diimplementasi pada lapisan rangkaian. Pentadbir – pentadbir hanya membenarkan jenis – jenis sambungan tertentu disambungkan kepada komputer yang dikhaskan untuk mencegah sambungan berjenis lain kepada komputer dengan menggunakan penapis paket yang mempunyai kemampuan untuk menapis port TCP/UDP. Dalam kes agoritma umum iaitu bagi pemeriksaan paket rangkaian yang lengkap, jika tiada polisi yang berpadanan ditemui, maka paket rangkaian akan di*drop*. Manakala jika sebaliknya paket akan dibenarkan berkomunikasi
4. Penapis paket adalah kurang selamat kerana ia tidak melakukan pemeriksaan terhadap paket – paket rangkaian pada data lapisan aplikasi dan tidak menjejaki paras bagi sambungan. Penapisan paket membenarkan capaian terus melalui *firewall* dengan jumlah pemeriksaan rapi yang rendah. Jika pemeriksaan polisi

adalah berjaya, paket rangkaian akan dihalakan melalui *firewall* seperti yang ditakrifkan dalam jadual penghalaan.

5. Tidak mempunyai beberapa ciri utama seperti penyembunyian HTTP (*hypertext transfer protocol*), penapisan URL atau pengesahan dan mekanisme kepekaan

2.3 Kajian Terhadap Projek Yang Dibangunkan Sebelum Ini

Satu kajian terhadap sistem yang dibangunkan oleh pelajar sebelum ini – smartfw, telah dilakukan untuk mengenalpasti kelebihan, kekurangan, definisi dan aplikasinya. Saya telah memilih untuk melakukan peningkatan dari penggunaan *ipchains* dalam smartfw kepada *iptables* yang lebih terkini namun masih lagi sesuai dengan kernel 2.4 yang digunakan.

Merujuk kepada antaramuka bagi sistem smartfw (rujuk apendix B), peraturan yang digunakan serta chainsnya adalah menggunakan *ipchains*. Di mana *ipchains* adalah merupakan satu peraturan yang menentukan status paket dengan merujuk kepada kernel yang bertindak sebagai pusat bagi penapisan paket firewall

Peningkatan *iptables* yang akan saya lakukan juga mempunyai peranan yang hampir sama dengan *ipchains* namun terdapat beberapa ciri – ciri tambahan yang akan dibincangkan nanti.

2.4 Cadangan dalam projek ilmiah Akhir 2

Meningkatkan penggunaan *ipchains* bagi smartfw kepada *iptables* yang lebih pantas, tepat dan terkini. *Iptables* juga adalah merupakan versi terbaru yang menggantikan *ipchains* dan juga di gunakan dalam kernel 2.4.

2.4.1 Cadangan Terhadap Polisi Keselamatan Organisasi

Terdapat beberapa cadangan dan penambahan polisi yang ingin dikemukakan dalam meningkatkan kualiti keselamatan organisasi :

1. Penapisan paket mampu melaksanakan polisi keselamatan organisasi dengan tepat. Maka pentadbir perlu lebih peka dalam menentukan paras atau bahagian mana yang memerlukan pemerhatian dan ketelitian dalam melaksanakan polisi keselamatan ini. Jadi polisi keselamatan sistem rangkaian bagi sesebuah organisasi perlu menekankan kepada beberapa perkara berikut :
    i) Kesemua trafik rangkaian yang mencurigakan (atau tidak jelas) akan dinafikan perkhidmatannya
    ii) Konfigurasi penapisan paket untuk sistem *firewall* ditunjukkan pada persekitaran yang berasingan daripada rangkaian pengoperasian
2. Boleh diandaikan juga bahawa peraturan terakhir di dalam setiap set peraturan bagi sistem *firewall* adalah untuk menafikan kesemua paket. Walaubagaimanapun dicadangkan juga pentadbir menerangkan peraturan ini dengan lebih jelas dan tepat untuk mengingatkan kita berkenaan polisi yang ingin dilaksanakan agar menepati dengan set peraturan yang lebih lengkap.

3. Mendapatkan sokongan dan pendapat dari pengguna berhubung dengan saling sambungan (*interconnection*) untuk mendapatkan apakah yang mereka jangka untuk dilakukan . Maklumat dan senarai ini akan dikumpulkan dalam satu jadual protokol, port serta alamat sumber dan destinasi. Kemudian barulah pentadbir sistem atau rangkaian memilih mana – mana bahagian untuk dilaksanakan sebagai polisi keselamatan organisasi.
4. Sentiasa melakukan pemerhatian dan mengawal peraturan penapisan paket yang hadir bersama – sama dengan perisian *firewall*. Kadangkala ia adalah tidak jelas dan tidak mematuhi polisi keselamatan. Maka pentadbir disarankan agar berhati – hati semasa membuat pertimbangan terhadap pendokumentasian perisian *firewall*.
5. Jika *firewall* mempunyai set – set peraturan yang berbeza untuk penerimaan dan penghantaran pada setiap antaramuka, pentadbir perlu mengulang peraturan pada set peraturan penerimaan bagi setiap antaramuka. Keadaan ini mampu mengurangkan kesalahan yang tidak disengajakan.
6. Membuat pemeriksaan jika pentadbir mendapati sistem *firewall* berkebolehan untuk menjana penyamaran peraturan IP secara automatik dari jadual penghalaan. Kebaikan pendekatan ini terhadap peraturan yang direkabentuk secara manual ialah peraturan akan dapat menyesuaikan dirinya secara automatik kepada perubahan penghalaan. Jika kita tidak menjana peraturan anti – penyamaran secara automatik, maka perubahan penghalaan akan menyebabkan kesan terhadap kemampuan kita untuk sampai pada destinasi – destinasi tertentu dari mula penghalaan dan penapisan berkonflik.

7. ICMP (*Internet control message protocol*) adalah protokol yang tidak bersambungan. Oleh sebab itu ia adalah tidak elok bagi had – had penapisan dan ancaman yang sama jenis seperti UDP. Walaubagaimanapun hanya 13 jenis paket ICMP yang ada. Bagi setiap jenis, organisasi perlu menentukan dengan jelas yang mana adalah dibenarkan.
8. Polisi perlu menghadkan penggunaan rangkaian dalaman dan perkhidmatan organisasi serta berhati – hati terutama terhadap pengguna yang bukan terdiri daripada kakitangan organisasi. Jika kita menyediakan perkhidmatan yang spesifik untuk digunakan oleh bukan ahli (seperti capaian kepada laman web syarikat), maka polisi ini perlulah diasingkan daripada sistem dalaman. Keadaan ini akan mengurangkan risiko walapun jika terdapat mana – mana serangan yang 'kuat' kerana capaian melalui kaedah ini sering diperkukuhkan dan diselenggarakan
9. Polisi perlu mengalamatkan keperluan capaian bagi ahli – ahli organisasi yang berada pada rangkaian yang tidak dipercayai (seperti *mobile user* pada Internet, pekerja yang berada di *business partner site*). Kita boleh melaksanakan mekanisma untuk membenarkan capaian yang sesuai melalui sistem *firewall* ke rangkaian dalaman atau sistem oleh pihak tersebut dengan menggunakan kaedah enkripsi dan pengesahan

BAB 3 : METODOLOGI

BAB 3 : METODOLOGI

3.1 Pengenalan

Apabila kita hendak membangunkan sesuatu produk atau perkhidmatan kita memerlukan satu set arahan tugas yang melibatkan langkah – langkah aktiviti, sekatan dan had – had tertentu serta sumber – sumber dalam menghasilkan output bagi produk tersebut.

Keadah pembangunan sistem adalah satu kaedah untuk mencipta sistem dengan beberapa siri langkah atau operasi atau boleh juga ditakrifkan sebagai model kitar hayat. Kita menggunakan kaedah ini untuk membantu merekabentuk dan membangunkan sesuatu sistem yang kompleks dengan menggunakan pendekatan berfasa. Pendekatan ini adalah untuk memastikan kita tidak beralih dari satu fasa ke fasa yang lain sebelum keputusan dicapai dan disahkan pada fasa semasa. Berdasarkan keadaan ini, maka kita telahpun mempunyai maklumat yang diperlukan untuk bergerak ke fasa lain jika fasa semasa tidak menghasilkan output yang diingini.

Sesuatu proses adalah penting kerana ia menekankan kepada kekonsistenan dan struktur pada set – set aktiviti. Fungsi ini adalah berguna apabila kita mengetahui bagaimana untuk melakukan sesuatu dengan lebih baik dan untuk memastikan pihak lain juga tidak mengalami kesilapan. Ini kerana proses juga adalah untuk membolehkan pentadbir sistem atau rangkaian mendapat pengalaman dalam meningkatkan pengetahuan agar boleh disampaikan kepada pihak yang lain.

Setiap model proses pembangunan perisian merangkumi keperluan – keperluan sistem yang bertindak sebagai input dan menghasilkan keluaran iaitu output. Terdapat beberapa model yang dicadangkan untuk membantu memudahkan perjalanan projek mengikut persekitaran projek yang ingin dibangunkan :

i) Model Air Terjun

ii) Model Air Terjun Dengan Prototaip

iii) Model V

iv) Model Prototaip

v) Model Pembangunan Berfasa : Penokokan dan Iterasian

vi) Model Spesifikasi Operasian

vii) Model 'Transformasi'

viii) Model Spiral

3.2 Model

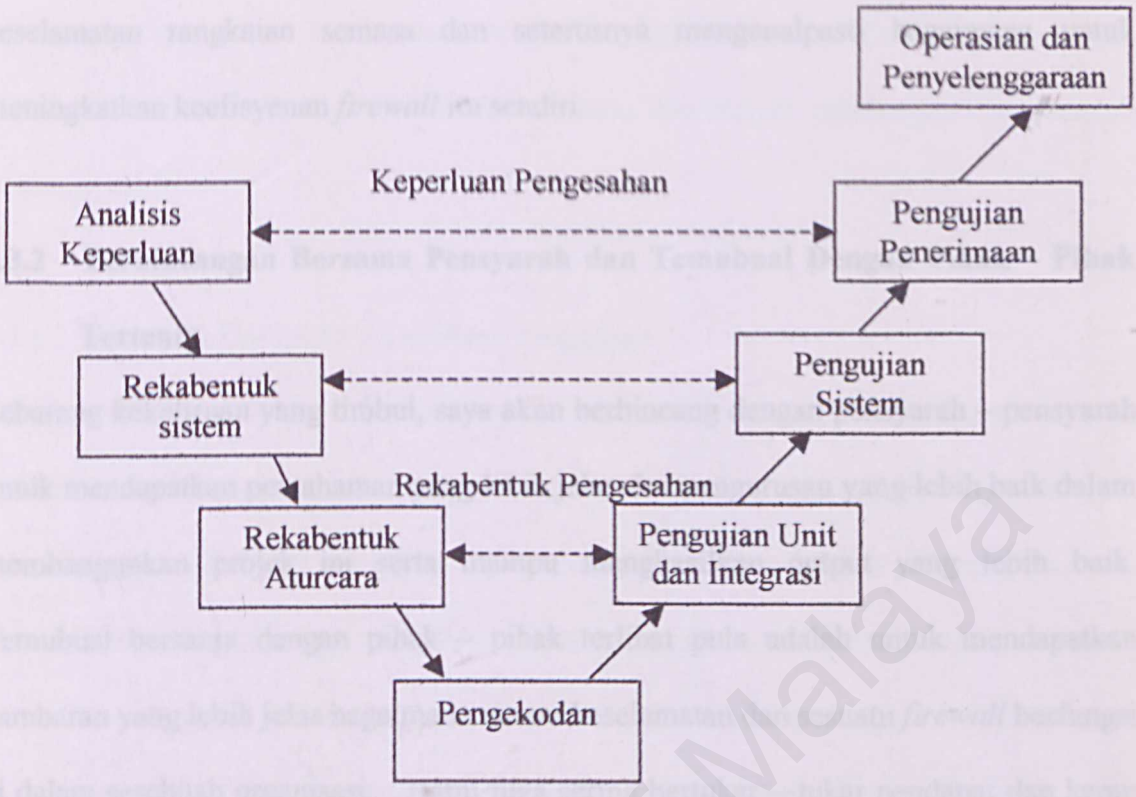
3.2.1 Model V

Model proses yang akan digunakan untuk projek peningkatan dari *ipchains* kepada *iptables* ini yang menggunakan teknik penapisan paket ialah model V.

Model V adalah berbeza berbanding model air terjun yang menunjukkan bagaimana aktiviti – aktiviti pengujian dihubungkan dengan analisis dan rekabentuk. Berdasarkan kepada rajah 3.1, pengekodan akan membentuk titik utama bagi model ini, dimana analisis dan rekabentuk berada di kiri model manakala pengujian dan penyelenggaraan berada di sebelah kanan. Pengujian unit dan integrasi akan menentukan ketepatan

aturcara. Model V mencadangkan bahawa pengujian unit dan integrasi juga akan digunakan untuk mengesahkan rekabentuk aturcara. Oleh sebab itu semasa pengujian unit dan integrasi, ahli bagi pengujian dan pengekod perlu memastikan kesemua aspek rekabentuk aturcara telah dilaksanakan dengan tepat pada kod. Begitu juga dengan pengujian sistem yang perlu mengesahkan rekabentuk sistem, memastikan kesemua aspek rekabentuk sistem telah dilaksanakan dengan tepat. Pengujian penerimaan yang telah dilakukan oleh pelanggan adalah lebih baik berbanding pengujian yang telah dilakukan oleh pembangun, mengesahkan segala keperluan dengan mengumpulkan langkah – langkah pengujian bagi setiap elemen yang tertentu. Pengujian berjenis ini akan memeriksa untuk melihat bahawa kesemua keperluan telah dilaksanakan dengan sepenuhnya sebelum sistem diterima atau dapat digunakan.

Sambungan di antara bahagian sebelah kiri dengan bahagian di sebelah kanan model V menunjukkan akan terdapat masalah yang dikenalpasti sepanjang proses pengesahan. Kemudian bahagian kiri model V boleh dilaksanakan semula untuk menetapkan dan meningkatkan keperluan – keperluan , rekabentuk dan kod – kod sebelum langkah pengujian di sebelah kanan di lakukan pengubahsuaian semula. Dengan kata lainnya model V menjadi sistem yang lebih jelas bagi beberapa siri pengulangan semula yang terselindung di sebalik gambaran air terjun. Keadaan ini menunjukkan ciri bagi air terjun adalah lebih kepada pendokumentasian dan artifak, manakala bagi model V pula menekankan kepada aktiviti – aktiviti dan proses pembetulan.



Rajah 3.1 : Model V

3.3 Teknik Mentakrif Keperluan

3.3.1 Carian Sumber Dari Bahan Bercetak dan Buku Ilmiah

Untuk mendapatkan lebih banyak maklumat – maklumat tambahan dan sumber, saya telah pergi ke Perpustakaan Negara Malaysia dan Bilik Dokumen, Fakulti Sains Komputer Dan Teknologi Maklumat, Universiti Malaya untuk mendapatkan sebarang rujukan yang berkaitan dengan keselamatan rangkaian dan sistem *firewall*. Kemudian saya juga turut membuat rujukan dan kajian menggunakan majalah – majalah komputer, buku - buku teks dan buku – buku ilmiah. Keadaan ini adalah penting untuk saya lebih memahami dengan jelas berkenaan kedua – dua cabang rangkaian. Seterusnya melalui

pemahaman ini saya dapat mengenalpasti kelemahan – kelemahan dalam sistem keselamatan rangkaian semasa dan seterusnya mengenalpasti bagaimana untuk meningkatkan keefisyenan *firewall* itu sendiri.

3.3.2 Perbincangan Bersama Pensyarah dan Temubual Dengan Pihak – Pihak Tertentu

Sebarang kekeliruan yang timbul, saya akan berbincang dengan pensyarah – pensyarah untuk mendapatkan pemahaman yang lebih jelas dan pengurusan yang lebih baik dalam membangunkan projek ini serta mampu menghasilkan output yang lebih baik. Temubual bersama dengan pihak – pihak terlibat pula adalah untuk mendapatkan gambaran yang lebih jelas bagaimana sistem keselamatan dan sesuatu *firewall* berfungsi di dalam sesebuah organisasi. Kami juga sering bertukar – tukar pendapat dan kerap melakukan perbincangan untuk memastikan perlindungan keselamatan ini berjalan dengan baik dan melibatkan pengendalian yang teliti terhadap data – data sensitif dan sumbernya.

3.3.3 Carian Sumber Maklumat Melalui Internet

Carian melalui Internet adalah merupakan teknik terpenting dan terpantas untuk membantu saya mendapatkan maklumat yang tepat dan terkini. Keperluan - keperluan yang sesuai telah dipilih berdasarkan maklumat dan perbandingan yang dilakukan melalui Internet untuk membangunkan projek dan yang paling utama ia mudah dikonfigurasikan berbanding yang lain. Adalah ampir 80% daripada sumber maklumat yang diperolehi adalah rujukan dari Internet. Secara tidak langsung dari aktiviti –

aktivit pencarian ini telah mendedahkan saya kepada sejumlah besar maklumat berkenaan persekitaran keselamatan rangkaian dan komputer. Seterusnya ia dapat membantu meluaskan pengetahuan berkenaannya dan banyak pandangan – pandangan baru diperolehi.

3.3.4 Rumusan Daripada Keperluan Pengguna

Merujuk kepada aktiviti – aktiviti carian sumber di atas, saya mendapati bahawa keperluan dalam keselamatan rangkaian adalah semakin meningkat sejajar dengan perkembangan teknologi komputer. Selaras dengan perkembangan ini yang semakin pesat, tidak ketinggalan juga dengan penglibatan penggodam dalam menceroboh masuk ke dalam rangkaian atau sistem persendirian juga semakin canggih. Berdasarkan kepada keperluan keselamatan inilah telah memberi inisiatif kepada saya untuk mempelajari, memahami dan membangunkan salah satu daripada teknik keselamatan dalam rangkaian. Tambahan pula pada setiap komputer persendirian atau milik organisasi adalah memerlukan kepada teknik *firewall* dalam melindungi rangkaian dari ancaman penggodam.

BAB 4 : REKABENTUK SISTEM

BAB 4 : REKABENTUK SISTEM

4.1 Konsep *iptables*

Iptables adalah merupakan satu tools peraturan yang memasukkan dan membuang mana – mana peraturan daripada jadual kernel bagi penapisan paket *firewall*. Ini bermakna apa - apa sahaja yang telah disetkan oleh pentadbir, akan hilang semasa *reboot*. Jadi untuk memastikan *setting* ini masih boleh dicapai, maka saya perlu memastikan segala bentuk *setting* ini adalah disimpan semula di dalam *rules permanent* agar ia masih boleh dicapai kelak.

Iptables adalah merupakan peningkatan daripada *ipchains* dan *ipfwadm* serta merupakan sebahagian daripada pakej "*Netfilter*"

4.2 Rekabentuk *iptables*

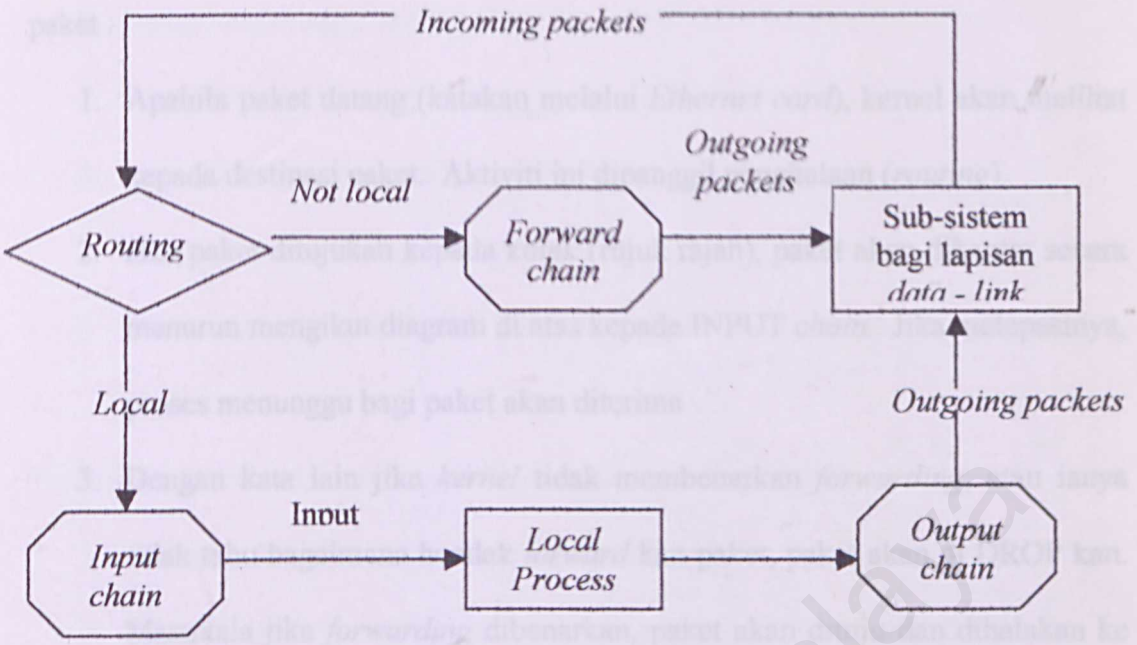
Iptables menggunakan ruang pengguna (*user-space*) yang boleh kita gunakan untuk membuat sebarang pengubahsuaian dalam 3 *built – in* bagi lapisan rangkaian *kernel – filtering – chains* (INPUT, OUTPUT, FORWARD). Perubahan demi perubahan terhadap penggunaan *iptables* dilakukan di mana *iptables* hanya boleh dilarikan pada kernel 2.3 dan 2.4 bagi Linux dan ianya berbeza dari *ipchains* dalam beberapa keadaan :

Iptables	Ipchains
• Dicipta bagi menggantikan ipchains • Parameter *built – in* dalam huruf besar (INPUT,OUTPUT,FORWARD) kerana INPUT dan OUTPUT chains adalah *locally-destined* dan *locally - generated packets*. Iptables akan melihat kepada semua paket yang masuk dan keluar dengan teliti • -i flag untuk *incoming interface*, berfungsi dalam INPUT dan FORWARD chains. –o flags untuk outgoing interface, berfungsi dalam OUTPUT dan FORWARD chain sahaja • Polisi DENY ditukar kepada DROP • Nama chian maksimum adalah 31 karektor • -y flags ditukar kepada --syn dan ditempatkan selepas –p tcp • Dilarikan pada kernel 2.4 • Pantas dan efisyen	• Dicipta bagi menggantikan ipfwadm • Parameter built – in adalah dalam huruf kecil • Menggunakan –i flags bagi kedua – dua incoming dan outgoing interface • Menggunakan istilah DENY bagi menafikan paket • -y flag untuk memadankan paket – paket IP dengan set bit SYN • Dilarikan pada kernel 2.3 dan 2.4 • Tiada kemudahan untuk *pass* paket ke ruang pengguna • *Transparent proxying* yang sukar • Peraturan penapis paket bergantung kepada alamat antaramuka

Rajah 4.1 : Ciri – ciri pada *iptables* dan *ipchains*

4.3 Bagaimana Paket Melalui Penapis Dengan Menggunakan *Iptables*

Kernel bagi *iptables* ini akan dimulakan dengan 3 jenis senarai peraturan yang dimuatkan dalam jadual penapis. Senarai ini dipanggil *firewall chains* atau *chains* sahaja. Ketiga – tiga *chains* ini ialah INPUT, OUTPUT dan FORWARD.



Rajah 4.1 : *Linux kernel packet – filtering built in chains*

Rajah di atas menunjukkan aliran paket – paket yang bergerak mengelilingi kernel dan menunjukkan di mana 3 *chains* ini beroperasi dalam keseluruhan proses. Tiga bentuk ⬡ mewakili 3 jenis *chain* yang telah dinyatakan. Apabila paket tiba pada kotak ini, *chain* akan memeriksa untuk menentukan nasib bagi paket – paket tersebut. Jika *chain* mendapati DROP bagi sesuatu paket, maka paket akan dibuang. Tetapi jika *chain* mengatakan ACCEPT, paket akan meneruskan perjalanannya melalui diagram di atas. *Chain* juga bertindak sebagai *check – rules* di mana setiap peraturan mengatakan 'Jika kepala menunjukkan seperti ini, maka inilah yang akan dilakukan kepada paket'. Jika peraturan adalah tidak berpadanan dengan paket, peraturan yang berikutnya dalam *chain* akan dipertimbangkan. Akhirnya jika tiada peraturan yang boleh dipertimbangkan, maka *kernel* akan

melihat pula kepada polisi yang biasanya akan memberitahu kernel unutk DROP paket :

1. Apabila paket datang (katakan melalui *Ethernet card*), kernel akan melihat kepada destinasi paket. Aktiviti ini dipanggil penghalaan (*routing*)
2. Jika paket ditujukan kepada kotak (rujuk rajah), paket akan dihantar secara menurun mengikut diagram di atas kepada INPUT *chain*. Jika melepasinya, proses menunggu bagi paket akan diterima
3. Dengan kata lain jika *kernel* tidak membenarkan *forwarding*, atau ianya tidak tahu bagaimana hendak *forward* kan paket, paket akan di DROP kan. Manakala jika *forwarding* dibenarkan, paket akan dituju dan dihalakan ke antaramuka rangkaian yang lain (jika ada), kemudian paket akan menghala ke kiri diagram iaitu ke FORWARD *chain*. Jika diterima maka paket akan dihantar keluar
4. Akhir sekali, aturcara yang dilarikan dalam kotak boleh menghantar paket – paket rangkaian. Paket – paket ini akan dihantar melalui OUTPUT *chain* dengan segera. Jika ia mengatakan ACCEPT, paket akan terus keluar ke antaramuka mengikut arah tuju yang telah ditetapkan

4.4 Menggunakan *Iptables*

Iptables mempunyai laman manual (*man page*) yang terperinci. Terdapat beberapa perbezaan yang wujud dalam *iptables*. Kita akan mulakannya dengan 3 *built – in chains* ; INPUT, OUTPUT, FORWARD di mana kita boleh padam. Terdapat beberapa operasi yang menguruskan *chains* ini :

1. Mencipta *chain* baru (-N)
2. Padam *chain* yang kosong (-X)
3. Mengubah polisi untuk *built – in chain* (-P)
4. Senaraikan peraturan dalam *chain* (-L)
5. Buang peraturan keluar dari *chain* (-F)
6. Mengosongkan paket dan pembilang *byte* pada semua peraturan dalam *chain* (-z)

Terdapat juga beberapa cara untuk laksanakan peraturan dalam *chain*

1. Menambahkan peraturan baru kepada *chain* (-A)
2. Masukkan peraturan baru pada posisi tertentu dalam *chain* (-I)
3. Manggantikan peraturan pada beberapa polisi dalam *chain* (-R)
4. Padam peraturan pada beberapa posisi dalam *chain* (-D)
5. Padam peraturan pertama yang berpadanan dalam *chain* (-D)

BAB 5: ANALISIS SISTEM

BAB 5 : ANALISIS SISTEM

5.1 Perisian

Menggunakan *kernel* 2.4 mengandungi pelbagai ciri – ciri yang khusus yang telah disediakan oleh Linux. Tambahan pula Linux menawarkan banyak bahasa pengaturcaraan, pengkompil dan *tools* pembangunan yang berpadanan dengannya.

5.1.1 Platform Sistem Pengoperasian - Linux

Linux adalah sistem pengoperasian yang direkabentuk untuk menyediakan perbandingan sistem pengoperasian yang melibatkan kos yang rendah atau percuma bagi sistem lama dan pastinya lebih mahal untuk sistem Unix. Linux mempunyai reputasi yang baik dari segi efisyen dan perlaksanaannya yang pantas. Linux *kernel* (iaitu bahagian pusat bagi sistem pengoperasian) sebenarnya telah dibangunkan oleh *Linus Torvald* di *University Of Iielsinki di Finland.*

Linux adalah sistem pengoperasian yang lengkap, yang mengandungi antaramuka pengguna grafik, sistem tetingkap X, TCP / IP, pengarang Emacs dan komponen – komponen lain yang biasanya terdapat pada sistem Unix yang komprehensif. Walaubagaimanapun, hakcipta terpeliharanya telah dipegang oleh pelbagai pencipta komponen – komponen Linux.

Perlaksanaan *open – source* bagi Unix telah melarikan banyak platform perkakasan yang berbeza termasuklah Intel, Sparc, PowerPC dan pemproses Alpha. Terdapat

beratus – ratus aturcara aplikasi telah ditulis untuk Linux, sesetengah daripadanya adalah daripada projek GNU. Linux dan *Linux tools* boleh dimuaturun secara percuma melalui Internet atau BBS atau melalui pembelian, iaitu yang merupakan sebahagian daripada CD – ROM

5.1.2 Netfilter

Terdapat beberapa ciri dan peningkatan yang dilakukan dalam linux *kernel* 2.4 yang menjadikannya satu platform yang utuh untuk kegunaan *firewall*. Salah satu peningkatan yang paling penting terhadap versi *kernel* ini ialah sub sistem bagi penapisan paket yang dinamakan *netfilter*. Pembangunan bagi *netfilter* telah dilakukan secara besar – besaran oleh *Watchguards Technologies*, iaitu syarikat US yang membangunkan serta memasarkan aplikasi *firewall* yang komersial yang berasaskan platform Linux, juga bertindak sebagai perkhidmatan keselamatan pada platform tersebut.

Dalam projek ini, *iptables* adalah digunakan sebagai peningkatan kepada *ipchains* yang sedia ada pada smartfw. *Iptables* adalah sebahagian daripada pakej *netfilter* yang digunakan oleh ruang pengguna untuk membuat apa – apa perubahan dalam 3 *built – in*.

5.1.3 Text Editor

Text Editor adalah aturcara untuk mencipta fail – fail atau membuat sebarang perubahan terhadap fail yang sedia ada. Editor adalah biasanya kurang berkuasa

berbanding pemproses *word*. Tambahan pula ia tidak mempunyai ciri – ciri kemampuan terkini untuk pemformatan teks, contohnya seperti kegunaan *italics*. Teks atau pengarang muka penuh (*full screen editors*) membenarkan pengguna untuk bergerak sepanjang dokumen dengan menggunakan anak panah penunjuk arah. Secara perbandingannya baris bagi editor memerlukan pengguna untuk memulakan nombor baris pada teks yang hendak diubahsuai. Terdapat beberapa jenis teks editor dalam Linux seperti Vi dan emacs. Platform yang dipilih untuk melaksanakan peningkatan ini yang bertindak sebagai pengarang teks ialah Vi.

5.2 Perkakasan

Berikut adalah disenaraikan keperluan perkakasan minima untuk membangunkan dan menggunakan iptables dalam Linux *firewall*

- 486 – DX66 dengan 32MB RAM (*random access memory*)
- 250MB cakera keras (*hard disk*)
- Sambungan rangkaian (*LAN Cards, serial ports, wireless*)
- Monitor dan papan kekunci

Senarai perkakasan di atas adalah khusus bagi seorang pengguna. Jika anggaran terdapat lebih kurang 50 orang pengguna terdapat sedikit pengubahsuaian terhadap perkakasan ini :

- Pentium II dengan 64meg *memory*
- 2 gig cakera keras untuk penyimpanan log
- 2 sambungan rangkaian
- *monitor* dan papan kekunci

Sambungan – sambungan rangkaian boleh menjadi apa – apa sahaja jenis (*NIC cards*, ISDN, apa jua jenis modem)

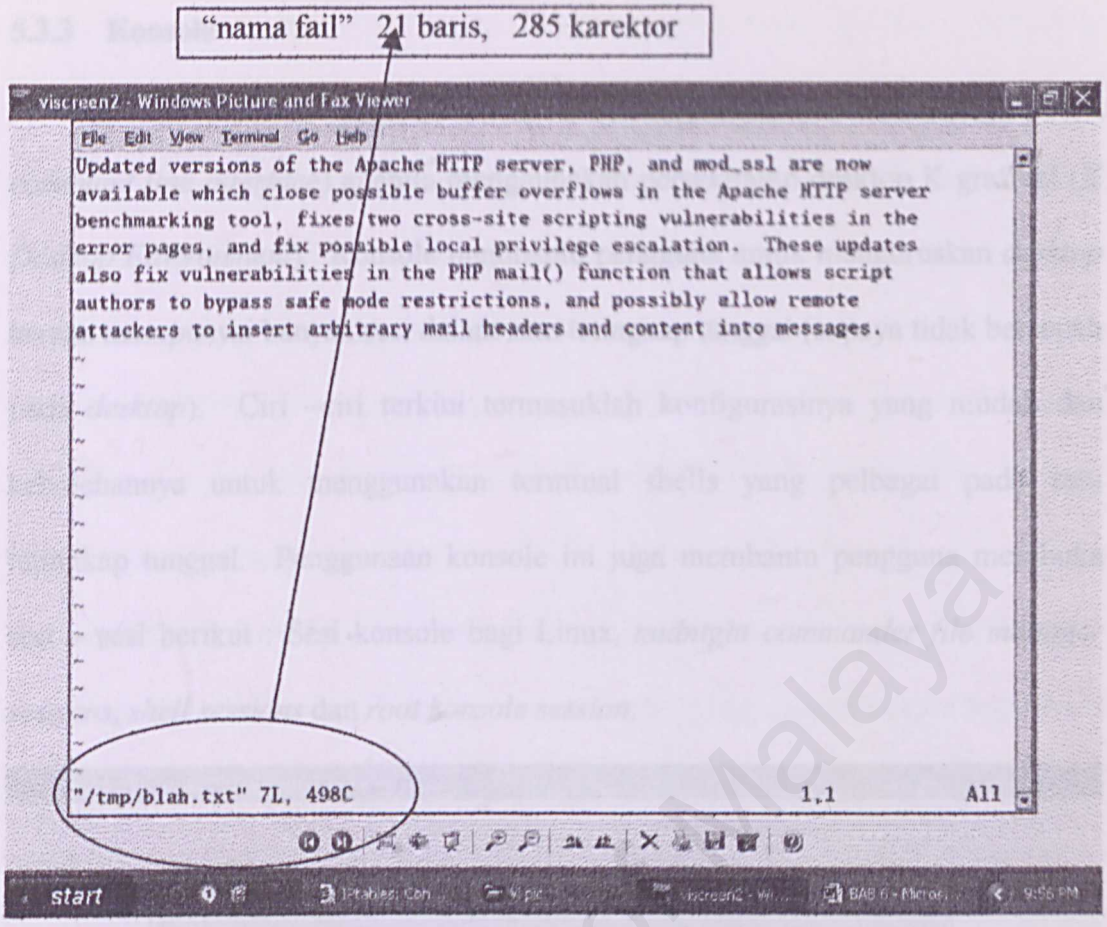
5.3 TOOL YANG DIGUNAKAN

5.3.1 Vi

Vi editor ditakrifkan sebagai satu editor berasaskan skrin yang digunakan oleh kebanyakan pengguna Linux. *Vi editor* mempunyai ciri – ciri yang baik untuk membantu pengaturcara, namun ramai pengaturcara pada mulanya mengelak dari menggunakan Vi kerana ciri – ciri yang wujud padanya.

Dengan menggunakan *Vi editor*, ianya membenarkan pengguna untuk mereka satu fail baru atau mengubahsuai fail – fail yang sedia ada. Arahan yang digunakan untuk memulakan *Vi editor* ialah *vi* diikuti dengan nama fail. Sebagai contoh, jika kita hendak mengubahsuai fail bernama *temporary*, kita akan taip *vi temporary* dan nilai akan dipulangkan di mana fail tersebut akan dipaparkan pada skrin *Vi*. Kita juga boleh memulakan *Vi* tanpa mengkhususkan kepada nama fail, tetapi apabila hendak menyimpan hasil kerja kita, maka kita perlu memberitahu nama fail kita kepada *Vi*.

Bentuk *tildes* (~) akan kelihatan pada skrin *Vi* semasa *Vi* ingin digunakan pada sebelah kiri skrin. Pada bahagian tengah skrin pula akan ditunjukkan nama fail dan saiz bagi fail tersebut. Sebagai contoh :



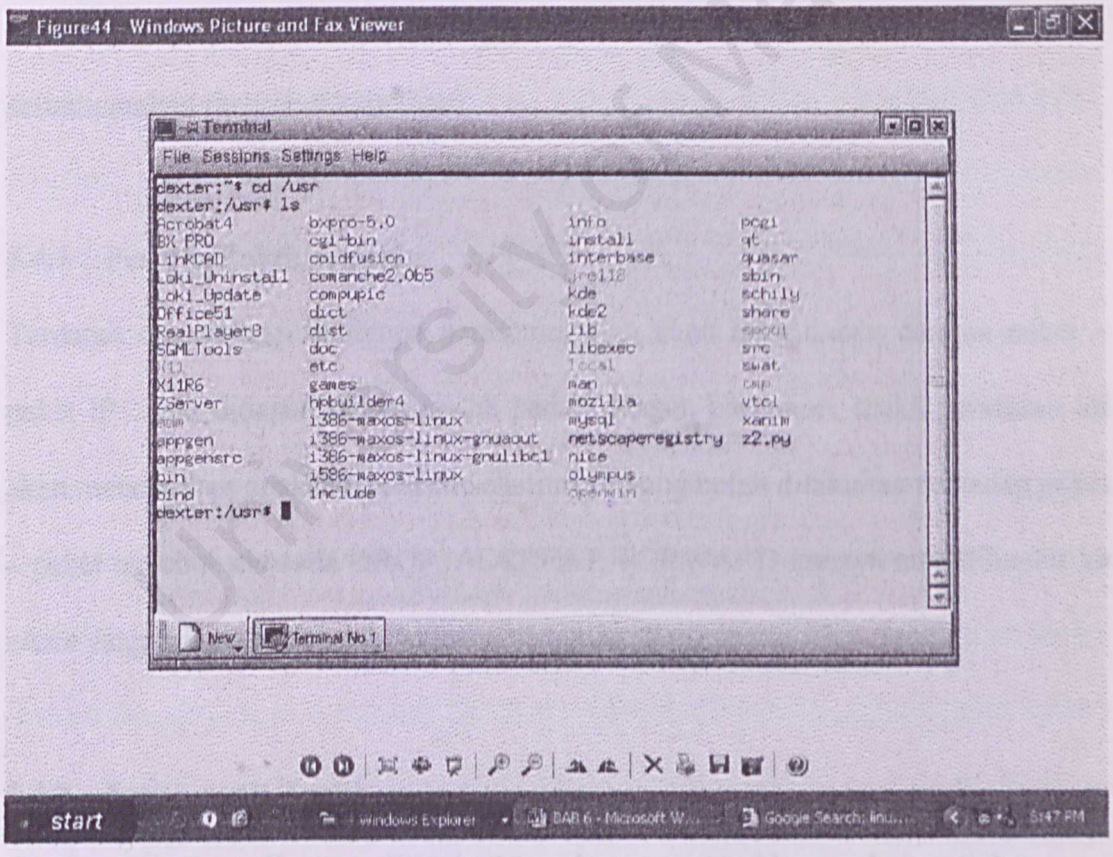
Rajah 5.1 : Contoh paparan pada editor Vi

5.3.2 Netfilter / Iptables

Terdapat beberapa ciri dan peningkatan yang dilakukan dalam Linux kernel 2.4 yang menjadikannya satu platform yang utuh untuk kegunaan *firewall*. Salah satu peningkatan yang paling penting terhadap versi kernel ini ialah sub sistem bagi penapisan paket yang dinamakan *Netfilter*. Pembangunan bagi *Netfilter* telah dilakukan secara besar – besaran oleh Watcguard Technologies iaitu syarikat US yang membangunkan serta memasarkan aplikasi *firewall* yang komersial yang berasaskan platform Linux, juga bertindak sebagai perkhidmatan keselamatan pada platform firewall tersebut

5.3.3 Konsole

Konsole ialah contoh terminal X yang menyediakan antaramuka baris arahan (*command line interface*) apabila menggunakan persekitaran desktop K grafikal (*K Desktop Environment*). Konsole membantu pengguna untuk menguruskan *desktop* kerana mempunyai banyak sesi dalam satu tetingkap tunggal (supaya tidak bersepah pada *desktop*). Ciri –ciri terkini termasuklah konfigurasinya yang mudah dan kebolehannya untuk menggunakan terminal shells yang pelbagai pada satu tetingkap tunggal. Penggunaan konsole ini juga membantu pengguna membuka sesi – sesi berikut : Sesi konsole bagi Linux, *midnight commander file manager sessions, shell sessions* dan *root konsole session.*



Rajah 5.2 : Contoh paparan bagi konsole

5.3.4 Smartfw Shell Script

Smartfw shell script adalah digunakan untuk melaksanakan *firewalling*. Skrip ini dilaksanakan dengan menggunakan penterjemah arahan (*shell*) bagi sistem pengoperasian ini. Istilah ini secara amnya merujuk kepada skrip yang akan dilaksanakan dengan menggunakan shell seperti Bourne, C dan Korne pada platform Linux ini.

5.4 Keperluan Fungsian

Keperluan fungsian merupakan satu pernyataan tentang perkhidmatan atau fungsi – fungsi yang perlu disediakan oleh sesuatu sistem yang menunjukkan bagaimana sistem bertindak balas terhadap input dan bagaimana sistem perlu laksanakan sesuatu arahan mengikut keperluan

5.4.1 Penapis Paket Kernel

Terdapat *chains* bagi beberapa peraturan yang akan berpadanan dengan paket – paket IP. Jika didapati paket adalah padan dengan peraturan, maka peraturan ini akan menentukan apakah tindakan selanjutnya yang boleh dilakukan terhadap paket – paket tersebut samaada DROP, ACCEPAT, FORWARD kannya atau dihantar ke *chain* yang lain.

5.4.2 Jenis – jenis Trafik

Konfigurasi *firewall* yang ditingkatkan akan mengarahkan paket – paket yang datang dari *host en – route* luaran ke *host* yang lain perlulah melalui ketiga – tiga

chains: bagi paket – paket yang destinasinya ialah *local machine* akan terus dihalakan ke INPUT chain, manakala paket yang akan keluar dari *local machine* pula akan melalui OUTPUT chain. Skema ini membenarkan pelaksanaan yang fleksibel terhadap peraturan – peraturan untuk berbagai jenis trafik.

Pentadbir perlu mencipta peraturan mereka di mana peraturan yang mempunyai *standard chains* boleh diarahkan untuk menghalakan paket ke *user – defined chains.*

5.5 Keperluan Bukan Fungsian

Keperluan bukan fungsian ditakrifkan sebagai kekangan terhadap mana – mana sistem yang perlu beroperasi dan piawai yang perlu dimiliki oleh sistem penerima. Keadaan ini menyebabkan pilihan yang diingini dalam membangunkan penyelesaian terhadap masalah dihadkan. Berikut adalah beberapa keperluan fungsian yang akan ditakrifkan :

5.5.1 Ketepatan

Sistem ini harus beroperasi dengan tepat kerana akan memberi kesan walaupun sedikit kepada pengguna – pentadbir jika digunakan sebaliknya. Ketepatan adalah merupakan darjah pencapaian log bagi aplikasi *firewall.* Untuk memastikan ketepatan, beberapa kajian dan pencarian telah dilakukan

5.5.2 Kebolehpercayaan

Kebolehpercayaan ialah apabila sistem adalah berjaya melaksanakan fungsinya dengan tepat dan teliti. Keadaan ini merangkumi pemprosesan paket dengan betul, penerimaan maklumat adalah tepat pada masanya, ada pemulihan kegagalan dan pemulihan rangkaian. Ciri ini juga menunjukkan sejauh mana pengguna yakin dengan perlaksanaan mekanisme ini dalam melindungi rangkaian dalaman daripada berkompromi.

5.5.3 Efisyen

Ialah kemampuan sesuatu sistem itu daripada memanggil prosedur – prosedur proses dengan mudah dan cepat. Capaian kepada prosedur ini sepatutnya tidak boleh dinafikan oleh sistem. Tahap efisyen ini dinilai berdasarkan kepada perlaksanaan rangkaian, transparensi dan perkhidmatan terjemahan alamat rangkaian (n*etwork address translation*)

5.5.4 Penyelenggaraan

Penyelenggaraan adalah diperlukan apabila sistem tidak mampu lagi untuk melaksanakan tugas dan proses dengan baik. Penyelenggaraan ini membantu mengesan aturcara mana yang perlu diperbetulkan jika berlaku ralat, disesuaikan mengikut keperluan persekitaran atau dilakukan peningkatan sejajar dengan kehendak pengguna. Jadi seharusnya pengesanan kepada percubaan yang gagal dapat dikesan dalam jangka masa yang singkat.

5.5.5 Kebolehsediaadaan

Sistem ini sentiasa beroperasi sebagai organisasi pembuka laluan capaian dari dunia luar kepada sistem dalaman rangkaian. Mekanisme ini mampu beroperasi setiap kali pentadbir sistem atau rangkaian memerlukan maklumat capaian yang gagal.

5.6 Keperluan Antaramuka

Rekabentuk antaramuka pengguna sebenarnya merujuk kepada sebuah aplikasi yang berkomunikasi dengan pengguna dan seterusnya antara pengguna dengan aplikasi. Dalam projek ini, saya banyak menggunakan pengarang teks Vi dalam berkomunikasi sebagai platform untuk perlaksanaan dan paparan.

BAB 6 : PERLAKSANAAN SISTEM

BAB 6 : PERLAKSANAAN SISTEM

Apabila fasa rekabentuk telah siap dilaksanakan, fasa berikutnya bagi projek ini dikenali sebagai fasa perlaksanan sistem. Pada fasa ini pengekodan akan disertakan disamping sedikit huraian berkenaannya. Pengekodan di sini ditakrifkan sebagai satu proses untuk menukar rupabentuk yang tertentu kepada beberapa set aturcara atau unit aturcara. Proses ini dimulakan dengan carian di Internet bagaimana untuk *upgrade* penggunaan *ipchains* dalam smartfw kepada *iptables*, di ikuti dengan pembangunan modul dalam aturcara. Pada bahagian terakhir fasa ini, modul akan digabungkan untuk membentuk satu sistem yang lengkap.

Pendokumentasian kod sumber yang berkualiti tinggi adalah penting agar dapat membantu untuk mengurangkan ralat pada pengekodan semasa aturcara mula ditulis serta memberi kemudahan untuk tujuan penyelenggaraan nanti. Saya telah cuba untuk mengatasi beberapa masalah yang timbul dengan beberapa kaedah berikut :

1. Penggunaan kod yang mudah difahami, iaitu mudah dibaca dan faham. Perkara ini dilakukan melalui penggunaan pembolehubah – pembolehubah, pemalar – pemalar, jenis – jenis, objek – objek serta modul yang mudah dan bermakna untuk membantu pengaturcara menggunakan kod sumber aturcara
2. Pengendalian yang baik di mana fungsi – fungsi boleh berinteraksi dengan kesemua input termasuklah bagi input yang salah. Sebagai contoh sistem

tidak akan *hang* walaupun pengguna memasukkan data yang tidak bersesuaian

3. Baris komen adalah digunakan dalam aturcara bagi menerangkan situasi aturcara, kepelbagaian komponen dan aliran logik. Baris komen boleh dibahagikan kepada beberapa komen : komen kepala (*header comments*) dan komen modul
4. Salinan paparan kod sumber adalah mudah dibaca. Sebagai contoh setiap ayat baru akan dimulakan dengan baris baru, manakala ruang kosong pula adalah untuk memisahkan blok – blok kod
5. Jenis – jenis kod akan dihimpunkan sekali. Kesemua pembolehubah dan pemalar berhubung dengna modul – modul tertentu atau fungsi – fungsinya dihimpunkan bersama

6.1 Persekitaran Pembangunan

Persekitaran kepada pembangunan mempunyai beberapa kesan ke atas pembangunan sistem. Dengan menggunakan persekitaran, perkakasan dan perisian yang sesuai tidak hanya membantu meningkatkan kelajuan pembangunan sistem tetapi juga menentukan kejayaan sesuatu projek

6.1.1 Keperluan Perkakasan

Penambahan modul *iptables* menggantikan *ipchains* tidak memerlukan banyak perkakasan. Keperluan yang paling minimum adalah seperti berikut :

1. 486 – DX66 dengan 32MB RAM

2. 250MB cakera keras
3. Sambungan rangkaian (LAN *cards*, *serial ports*)
4. Monitor dan papan kekunci

6.1.2 Keperluan Perisian

Untuk penambahan modul baru ini tiada perisian khusus yang diperlukan melainkan Red Hat Linux dengan *kernel* 2.4. Sebelum penambahan modul ini dilakukan, fail smartfw perlu dimuat turun ke dalam Red Hat Linux seperti yang dinyatakan tadi. Versi *kernel* 2.4 adalah digunakan kerana bersesuaian dengan penggunaan *iptables*. Namun jika versi kernelnya ialah 2.2, untuk menukar *ipchains* kepada *iptables* maka versi tersebut perlu di *upgrade* terlebih dahulu.

Manakala penggunaan *tools* dan bahasa pengaturcaraan telah diterangkan di dalam bab 4.

6.2 Pelaksanaan Peningkatan *Ipchains* kepada *Iptables*

6.2.1 Muat Turun smartfw

Pada Linux *kernel* 2.4 ini, smartfw akan dimuat turun dari Internet. Berikut ditunjukkan langkah – langkah semasa muat turun dilakukan :

1. Laman web smartfw dibuka : http://www31.brinkster.com/linuxfirewall
2. Pada perkataan *download*, akan klik kanan dan pilih *save link as*. Seterusnya muat turun akan dilaksanakan

3. Di dapati fail yang disimpan dalam *directory root* adalah belum dipecah – pecahkan kepada sub fail yang lebih khusus. Fail perlu dipecahkan adalah bertujuan untuk memudahkan pemahaman dan tugas yang ingin dilakukan.

6.2.2 Nyahmampatan Fail

Berikut ditunjukkan langkah – langkah pemecahan fail :

1. *tar –zxvf smartfw*** .tar gz* akan ditaip pada terminal
2. Satu *folder* baru akan dicipta di mana kesemua fail tersebut boleh disimpan
3. Output yang akan diperolehi ialah */ smartfw****
4. Seterusnya kita bolehlah memulakan aktiviti –aktiviti yang terlibat iaitu penggantian *ipchains* kepada *iptables* dengan penggunaan *change directory* (*cd*)

6.2.3 Menghasilkan Peraturan Tetap

Firewall semasa yang telah di*set – up*kan akan disimpan di dalam *kernel* dan akan hilang apabila sistem di*reboot*. Untuk memastikan hal ini tidak berlaku saya telah menggunakan skrip bagi *iptables – save* dan *iptables – restore* untuk penyimpanan *firewall* semasa dan mendapatkannya semula dari fail. Selain itu terdapat satu lagi kaedah lain iaitu menempatkan arahan yang diperlukan untuk *set – up*kan peraturan – peraturan dalam skrip yang tertentu. Tetapi dalam projek ini saya akan menggunakan kaedah pertama kerana ianya lebih mudah dan jelas bagi saya untuk melaksanakannya.

6.2.4 Menghentikan Ipchains

Untuk menghentikan sebarang aktiviti *firewall* dalam *ipchains*, maka saya perlu mendapatkan satu skrip bagi *iptables – save*. Tujuannya adalah untuk menyimpan set – set peraturan semasa ke dalam fail. Set – set peraturan ini akan dapat digunakan semula dengan menggunakan *iptables – restore*. Skrip bagi *iptables – save* akan disimpan di dalam parameter direktori seperti berikut :

/ etc / rc.d / init.d / ipatbles – save

Setelah disimpan di dalam parameter direktori ini barulah *ipchains* akan dihentikan aktivitinya dengan menaip . */ ipchains stop* pada konsole. (Sila rujuk appendix A untuk skrip)

6.2.5 Memulakan *Iptables*

Skrip *Iptables – restore* akan digunakan untuk memanggil semula fail – fail semasa yang telah disimpan menggunakan *iptables – save* tadi yang akan digunakan selepas sistem di*reboot*. Skrip *iptables – restore* juga akan disimpan di dalam direktori yang sama dengan *iptables – save* cuma nama fail pada hujung parameter ditukar kepada *iptables – restore*. Setelah skrip yang ditaip pada teks editor Vi siap dan disimpan, maka barulah aktiviti *iptables* boleh dimulakan dengan menggunakan arahan berikut : *. / iptables start* (Sila rujuk appendix A untuk skrip)

6.3 Paparan Konfigurasi dan Aktiviti Fail

Untuk melihat paparan konfigurasi, pada tetingkap terminal (konsole) saya akan pergi ke direktori di mana smartfw disimpan mengikut parameter berikut :

/ usr / local / smartfw.

Pada direktori ini apabila *ipchains –L* di taip, output yang akan dipaparkan ialah pernyataan berikut :

```
[root@loucheryuen smartfw]# ipchains -L
ipchains: Incompatible with this kernel
```

Rajah 5.1 : Paparan *ipchains –L* yang dihentikan

Hal ini bermaksud walaupun *ipchain* adalah sesuai dengan kernel yang digunakan tetapi tiada paparan output di mana aktivitinya telah terhenti.

Seterusnya apabila *iptables – L* pula ditaip, hasil yang akan keluar adalah seperti berikut :

```
[root@loucheryuen smartfw]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  200.200.200.1        anywhere
tcp dpt:telnet

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere
tcp spt:nfs
DROP       all  --  10.0.0.0/8           anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Rajah 5.2 : Paparan aktiviti *iptables –L*

Manakala untuk melihat kepada *events log* pula saya akan menaip arahan *iptables –nvL*. Fungsi arahan ini adalah untuk memaparkan segala aktiviti semasa yang berlaku berdasarkan kepada peraturan yang ditetapkan. Berikut ditunjukkan rangka output bagi arahan jenis ini :

```
[root@loucheryuen smartfw]# iptables -nvL
Chain INPUT (policy ACCEPT 65 packets, 9397 bytes)
 pkts bytes target     prot opt in     out     source
destination
    0     0 DROP       tcp  --  *      *       200.200.200.1
0.0.0.0/0           tcp dpt:23

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source
destination
    0     0 DROP       tcp  --  *      *       0.0.0.0/0
0.0.0.0/0           tcp spt:2049
    0     0 DROP       all  --  eth0   *       10.0.0.0/8
0.0.0.0/0
    0     0            tcp  --  *      *       0.0.0.0/0
0.0.0.0/0           tcp spt:2049

Chain OUTPUT (policy ACCEPT 93 packets, 11373 bytes)
```

```
 pkts bytes target     prot opt in     out     source
destination
```

Rajah 5.3 : Paparan aktiviti *iptables -nvL*

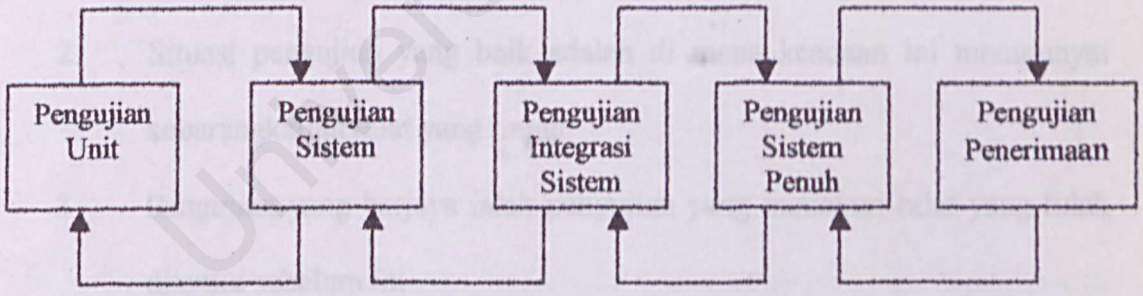
BAB 7 : PENGUJIAN SISTEM

BAB 7 : PENGUJIAN SISTEM

Pengujian sistem seringkali di kait rapat dengan pengesahan dan sejauh mana sesuatu sistem itu dapat berfungsi. Pengesahan adalah merujuk kepada set –set aktiviti untuk memastikan perisian melaksanakan sesuatu fungsi itu dengan betul. Manakala sejauh mana fungsi itu berjalan menunjukkan set – set aktiviti yang akan memastikan bahawa sistem telah memenuhi keperluan pengguna. Objektif utama bagi pengesahan aktiviti ini adalah untuk mencapai dan meningkatkan kualiti akhir bagi pengeluaran yang diingini.

7.1 Proses Pengujian

Proses pengujian bagi penambahan modul ini mempunyai 5 paras seperti yang ditunjukkan dalam rajah di bawah :



Rajah 7.1 : Proses Pengujian Sistem

Jujukan bagi aktiviti – aktiviti pengujian ialah pengujian komponen, pengujian integrasi dan pengujian oleh pengguna. Jika berlaku sebarang ralat pada mana –

mana paras, pengubahsuaian aturcara adalah diperlukan untuk memperbetulkan paras tersebut dan ini menunjukkan ada paras – paras yang perlu melakukan proses pengujian sekali lagi. Oleh sebab itu pengulangan semula proses dilakukan di mana maklumat yang menjadi *feed – back* oleh paras sebelum itu untuk bahagian terawal proses.

Merujuk kepada rajah 6.1, arah anak panah dari atas kotak menunjukkan jujukan perjalanan yang normal bagi pengujian. Manakala anak panah yang menunjukkan arah yang bertentangan di bahagian bawah kotak pula menunjukkan paras - paras pengujian sebelum ini yang perlu diulang.

Beberapa peraturan akan ditetapkan kemudian semasa fasa pengujian untuk memenuhi objektif proses :

1. Pengujian ialah proses perlaksanaan aturcara dengan matlamat untuk mendapatkan ralat
2. Situasi pengujian yang baik adalah di mana keadaan ini mempunyai kebarangkalian ralat yang tinggi
3. Pengujian yang berjaya ialah pengujian yang menemui ralat yang tidak ditemui sebelum ini

7.2 Strategi Pengujian

Strategi pengujian adalah pendekatan secara umum terhadap proses pengujian berbanding kaedah memikirkan sistem – sistem tertentu atau melibatkan pengujian komponen. Strategi pengujian ini termasuklah :

i) Membuat perbandingan kelebihan antara *ipchains* –L dan *iptables* –L

ii) Membuat pelbagai peraturan untuk memastikan ianya berpadanan dengan *event log* dan konfigurasi fail

iii) Pengujian *top – down* : Pengujian dimulakan dengan komponen yang paling abstrak hingga kepada tugasannya

iv) *Black – box testing* : Ciri – ciri input boleh dilaksanakan menggunakan pengujian *black – box*. Ciri – ciri ini dilakukan mengikut keperluan fungsian. Tujuannya adalah untuk mendapatkan ralat bagi beberapa kategori di bawah :

a) Ralat antaramuka

b) Fungsi yang salah atau tertinggal

c) Ralat kepada persembahan

d) Ralat kepada umpukan

v) Pengujian *back – to – back* :Digunakan untuk perbandingan di antara versi sedia ada dengan versi baru. Sistem akan dilaksanakan serentak dan output bagi kedua – duanya akan diperbandingkan

7.3 Pengujian Iptables

7.3.1 Pengujian Unit

Pengujian unit adalah memfokuskan kepada penilaian modul di dalam aturcara. Pengujian jenis ini adalah untuk memastikan bahawa komponen – komponen seperti *iptables – save script*, *iptables – restore script*, *smartfw kernel modules* berfungsi dengan baik.

Langkah pertama pengujian ini adalah dengan menilai kod aturcara melalui pemahaman dan pembacaan secara sepintas lalu, *try and error* pada algoritma – algoritma, data dan kesalahan sintaks. *Desk checking* adalah digunakan untuk menguji secara terus kod sumber yang digunakan. Sepanjang proses ini, segala ralat yang timbul seperti kesalahan sintaks dan logik diperbetulkan dalam kod – kod modul.

Seterusnya aturcara akan dilarikan untuk memastikan kod – kod sumber ini tiada ralat. Akhir sekali kes uji ini akan dibangunkan dan digabungkan dengan input untuk menghasilkan output yang dikehendaki.

7.3.2 Pengujian Integrasi

Apabila komponen individu telah berfungsi, komponen – komponen yang ada akan digabungkan menjadi satu sistem. Dengan kata lainnya, pengujian integrasi adalah merupakan satu proses yang memastikan komponen – komponen sistem mampu bekerja bersama – sama.

Dalam projek peningkatan ini, pendekatan *top – down* adalah digunakan untuk pengujian integrasi. Aturcara ini mewakili komponen – komponen abstrak yang tunggal : *ipchains stop* dan *iptables start* dengan sub – komponen – sub – komponen (*ipatbles – save, iptables – restore, smartfw kernel modules*)

Selepas komponen paras tinggi telah diuji, sub – komponen akan dilaksanakan dan diuji. Proses – proses ini akan berterusan sehingga komponen paras bawah selesai dilaksanakan.

Pengujian *top – down* adalah digunakan bersama – sama dengan pembangunan aturcara *top – down* maka komponen sistem secara tidak langsung akan diuji semasa ianya dikodkan. Pengekodan dan pengujian adalah aktiviti – aktiviti tunggal tanpa komponen yang dipisahkan atau dikenali sebagai fasa pengujian modul. Kebaikan menggunakan pendekatan *top – down* adalah seperti berikut :

i) Ralat rekabentuk yang tidak diberitahu akan dapat dikesan pada paras awal dalam proses pengujian. Biasanya ralat yang timbul adalah ralat struktur. Pengesanan awal bermaksud tambahan rekabentuk dan perlaksanaan semula yang boleh dielakkan

ii) Sistem dapat berfungsi pada paras mula pembangunan. Sistem akan terus dapat ditunjukkan kepada pengguna

7.3.3 Pengujian Antaramuka

Terdapat beberapa jenis kesalahan antaramuka yang berlainan jenis yang boleh berlaku :

Antaramuka Parameter : Antaramuka pada ketika ini adalah di mana rujukan data atau kadangkala fungsi dihantar dari satu komponen ke komponen yang lain.

Antaramuka Prosedur : Ini adalah antaramuka di mana satu sub – system meringkaskan set – set prosedur yang boleh dipanggil oleh sub –sistem yang lain

Antaramuka penghantaran pesanan : Antaramuka ini adalah apabila satu sub – sistem meminta perkhidmatan dari sub – sistem yang lain melalui penghantaran pesanan kepadanya. Pesanan yang dikembalikan adalah merupakan keputusan bagi perkhidmatan yang dilaksanakan itu.

7.4 Pengujian Peraturan – peraturan Firewall

Sesi ini menunjukkan bagaimana *iptables* boleh digunakan sebagai *diagnostic tool* untuk pengesahan dan pengujian peraturan *firewall*. Berikut disenaraikan beberapa cara yang boleh digunakan sebagai pembangunan *firewall* secara umum :

i. Skrip pengujian perlu di lakukan di dalam *konsole*. Elakkan daripada menguji dari *remote machine*

ii. Laksanakan satu tugas pada satu masa. Tambah peraturan pada satu masa sebagai input atau output dan ujinya. Ciri ini menjadikan lebih mudah untuk menentukan sebarang masalah yang timbul

iii. *Double check* pada sintaks *iptables*. Jika tidak ia akan mengelirukan arah perjalanan peraturan tersebut. Lakukan pemeriksaan pada alamat sumber dan destinasi, *port – port* atau huruf kecil dan huruf besar

iv. Jika berlaku ralat sintaks, peraturan bagi *firewall* akan keluar tanpa sebarang penambahan terhadap peraturan yang berikutnya

BAB 8 : PENILAIAN SISTEM

BAB 8 : PENILAIAN SISTEM

Selepas pengujian sistem dilaksanakan, seterusnya ialah penilaian terhadap sistem yang telah dibangunkan. Terdapat beberapa kaedah bagi saya untuk menilai sistem pada peringkat akhir ini. Tujuan penilaian ini adalah untuk mengenalpasti had kemampuan, keupayaan dan peningkatan yang boleh dilakukan pada masa akan datang.

8.1 Masalah dan Penyelesaian

Saya telah mengenalpasti beberapa masalah yang timbul sepanjang sistem ini dibangunkan. Masalah ini merangkumi tentang pengetahuan dan penggunaan dalam *ipchians*, *iptables*, memahami polisi dan peraturan serta takrifan yang jelas berkaitan dengan penapisan paket *firewall*. Tambahan pula mempunyai asas pengetahuan berkaitan dengna bahasa pengaturcaraan yang kurang serta beberapa proses lain sedikit sebanyak memerlukan saya mengambil masa untuk mempelajarinya. Berikut adalah beberapa masalah dan beberapa penyelesaian yang terlibat :

8.1.1 Memahami *Firewall*

Walaupun definisi *firewall* dari sudut bahasa sudah difahami, namun bagi menunjukkan satu kitar lengkap penggunaannya saya perlu membuat banyak pencarian dan rujukan. Tidak hanya merangkumi pemahaman tentang kitarnya

sahaja namun saya perlu juga untuk mengenalpasti sebarang perkembangan semasa dan kelemahan yang wujud dalam penapisan paket ini.

8.1.2 Mentakrifkan Skop Projek

Setelah memilih tajuk dan memahami sedikit sebanyak maklumat berkenaan *iptables*, saya perlu mengenalpasti skop serta hasil yang dijangkakan dari projek ini agar ianya mampu dilaksanakan dalam tempoh masa yang telah ditetapkan. Keupayaan sistem yang dihasilkan ini juga perlu ditentukan samaada ia mampu berfungsi mengikut cadangan atau tidak. Jika tidak mampu maka skop projek perlu dikenalpasti semula.

8.1.3 Tiada Pengetahuan Tentang Linux

Tidak dinafikan bahawa ini adalah kali pertama saya berkomunikasi dengan Linux. Secara tidak langsung melalui projek ini telah memberi saya peluang bagaimana untuk bekerja denga Linux. Namun demikian terlalu banyak yang perlu saya pelajari memandangkan saya langsung tiada pendedahan dan pengetahuan tentangnya seperti penggunaan *tools* dan bahasa pengaturcaraan dalam Linux untuk memastikan kefungsian sistem yang berjaya.

8.1.4 Pembelajaran Yang Meluas

Setiap hari saya perlu melayari Internet bagi mendapatkan sebarang maklumat terkini. Tambahan pula terlalu banyak laman web – laman web yang menyediakan perkhidmatan berkenaan dengan Linux dan saya memerlukan masa untuk

memahami, mempelajari dan membuat perbandingan tentangnya. Tidak hanya terbatas kepada rujukan melalui Internet, saya juga turut menggunakan buku – buku rujukan, majalah – majalah komputer serta perbincangan dengan pensyarah juga dilakukan.

8.1.5 Penerangan Secara Lisan Yang Terhad

Walaupun saya menggunakan pelbagai kaedah carian maklumat, namun pada pendapat saya juga penerangan secara lisan amat membantu kerana di situ wujud komunikasi dua hala yang saling berinteraksi dan menjadikan pemahaman itu lebih jelas.

8.2 Kelebihan sistem

Pada asalnya smartfw yang telah dihasilkan hanya memaparkan konfigurasi fail yang melaluinya. Tetapi dalam smartfw yang dihasilkan menggunakan peraturan *iptables* ini, telah menunjukkan output adalah melibatkan paparan konfigurasi fail dan juga dalam bentuk *events log*. Dalam *events log*, terdapat beberapa pertambahan maklumat seperti bilangan paket, kilobytes, maklumat *in* dan *out*. Berdasarkan pada ciri – ciri ini pentadbir boleh memeriksa aktiviti – aktiviti fail yang berlaku dengan jelas berdasarkan ciri – ciri fail yang ditunjukkan.

Selain itu penggunaan *iptables* juga adalah pantas dan melibatkan peraturan yang jelas. Sebagai contoh ditunjukkan peraturan berikut :

```
Iptables -A INPUT -p tcp --dport 22 -s 192.168.1.0/24 -j
ACCEPT
```

Bermaksud bagi input yang mepunyai protokol tcp dengan alamat destinasinya 22 dan alamat sumbernya ialah 192.168.1.0/24 jika ianya didapati melalui smartfw maka statusnya adalah ACCEPT.

8.3 Had Bagi Sistem Ini

Walaupun saya telah cuba untuk memberikan yang terbaik dalam projek ini, namun banyak lagi modul – modul mudah yang tidak dapat saya sertakan. Hal ini adalah disebabkan kurangnya pengetahuan dan ini menuntut saya meningkatkan lagi pencarian. Tambahan pula kemahiran yang tinggi adalah diperlukan untuk menghasilkan penambahan modul – modul lain dan sudah pastinya faktor masa perlu dipertimbangkan.

8.3.1 Peraturan Dimasukkan Secara Manual

Untuk menentukan samaada sesuatu paket adalah DROP, ACCEPT atau FORWARD, mengikut protokol dan *port* yang diperlukan, maka pentadbir perlu memahami bagaimana untuk membentuk dan menggunakan peraturan *iptables*. Ini kerana pentadbir perlu memasukkan peraturan yang mereka hendak pertimbangkan secara manual.

8.3.2 Tiada modul CONNTRACK

Pada asalnya saya ingin menambahkan modul CONNTRACK (*connection tracking*) kepada smartfw tetapi kerana kesuntukan masa maka saya terpaksa

membatalkannya. Namun sedikit kajian telah saya lakukan terhadap modul ini di mana penggunaan CONNTRACK adalah untuk menjamin keselamatan yang lebih baik kerana mampu mempastikan paras maklumat sambungan berdasarkan kepada jadual ingatan seperti alamat IP, sumber dan destinasi, pasangan nombor *port*, jenis protokol dan *timeouts*.

8.4 Peningkatan Pada Masa Akan Datang

Saya berharap pada masa akan datang beberapa ciri tambahan kepada sistem sedia ada dapat dipraktikkan di dalam smartfw kerana dengan pertambahan ciri ini, akan membantu meningkatkan keselamatan dari ancaman penggodam dalam sesebuah organisasi. Selain itu sebarang peningkatan juga boleh dilakukan berdasarkan kelemahan yang dinyatakan di atas.

Seterusnya saya juga berharap suatu masa nanti pemaparan aktiviti – aktiviti ini dapat dinyatakan dalam bentuk antaramuka grafik yang lebih menarik. Dalam pencarian saya penghasilan antaramuka yang ada yang berfungsi seperti klon VB adalah seperti perisian TCL / TK dan Phoenix namun pembelajaran yang lebih terperinci dan meluas perlu dilakukan lagi.

Bibliografi

1. Eric Maiwald, *Network Security A Begginer's Guide*, Osborne

2. Kieron Conway, *Software Project Management (From Concept To Deployment)*, 2001, CORIOLIS

3. Marcus Goncalves, *Firewalls Complete*, 1997, McGraw Hill

4. PC Worl Malaysia Magazine (Technology Advice You Can Trust), Vol 9 no. 8, August 2002

5. Tiffany Tailor, *Security Complete 2nd Edition*, 2002, SYBEX

6. Neil Matthew & Richard Stones, *Beginning Linux Programming*, 1996, WROX PRESS

7. Takrifan *Firewall*

   http://www.cert.org/security-improvement/practice/p058.html

8. Teknik Penapisan Paket

   http://www.cert.org/tech_tips/packet_filtering.html

9. Kebaikan dan Kelemahan Penapisan Paket

http://www. grouplifw.com/fw_tech.html

10. Firewall HOW TO

http://www.linuxfaq.com/LDP/HOWTO/firewall-HOWTO.html

11. Iptables – save script

http://www.linuxquestions.org/questions/archieve

12. Netfilter / iptables

http://www.netfilter.org/

13. http://www.planet-source-code/

APPENDIX A :

```
#
# iptables  - save script
#

#!/bin/sh
#
#
IPTABLES="/sbin/iptables"
OUTSIDE=ppp0
# multiple network intefaces on router so:
INSIDE=eth+
#INSIDE=eth1
#
$IPTABLES -F
$IPTABLES -F INPUT
$IPTABLES -F OUTPUT
$IPTABLES -F FORWARD
$IPTABLES -F -t mangle
$IPTABLES -F -t nat
$IPTABLES -X
$IPTABLES -P INPUT DROP
$IPTABLES -P OUTPUT ACCEPT
$IPTABLES -P FORWARD ACCEPT
#
#
$IPTABLES -N silent
$IPTABLES -A silent -j DROP

$IPTABLES -N tcpflags
$IPTABLES -A tcpflags -m limit --limit 5/minute -j LOG --
log-prefix TCPflags:
$IPTABLES -A tcpflags -j DROP

$IPTABLES -N internal
$IPTABLES -A internal -m limit --limit 5/minute -j LOG --
log-prefix Internal:
$IPTABLES -A internal -j DROP

$IPTABLES -N firewalled
$IPTABLES -A firewalled -m limit --limit 5/minute -j LOG --
log-prefix Firewalled:
$IPTABLES -A firewalled -j DROP
#
#
$IPTABLES -t nat -A POSTROUTING -o $OUTSIDE -j MASQUERADE
#
#
```

```
$IPTABLES -t nat -A PREROUTING -i $OUTSIDE -p tcp --dport
443 -j DNAT --to 192.168.42.1
$IPTABLES -t nat -A PREROUTING -i $OUTSIDE -p udp --dport
5190 -j ACCEPT
#
$IPTABLES -A INPUT -p tcp --tcp-flags ALL FIN,URG,PSH -j
tcpflags
$IPTABLES -A INPUT -p tcp --tcp-flags ALL ALL -j tcpflags
$IPTABLES -A INPUT -p tcp --tcp-flags ALL
SYN,RST,ACK,FIN,URG -j tcpflags
$IPTABLES -A INPUT -p tcp --tcp-flags ALL NONE -j tcpflags
$IPTABLES -A INPUT -p tcp --tcp-flags SYN,RST SYN,RST -j
tcpflags
$IPTABLES -A INPUT -p tcp --tcp-flags SYN,FIN SYN,FIN -j
tcpflags
#
#
$IPTABLES -A INPUT -p icmp --icmp-type 0 -j ACCEPT
$IPTABLES -A INPUT -p icmp --icmp-type 3 -j ACCEPT
$IPTABLES -A INPUT -p icmp --icmp-type 11 -j ACCEPT
$IPTABLES -A INPUT -p icmp --icmp-type 8 -m limit --limit
1/second -j ACCEPT
$IPTABLES -A INPUT -p icmp -j firewalled
#
#
$IPTABLES -A INPUT -i lo -j ACCEPT
#
#
# $IPTABLES -A INPUT -i $INSIDE -d 192.168.42.1 -j ACCEPT
#
# NEW CODE HERE!
##############################################
#
$IPTABLES -A INPUT -i $INSIDE -s 192.168.42.8 -d
192.168.42.1 -j ACCEPT
$IPTABLES -A INPUT -i $INSIDE -s 192.168.43.7 -d
192.168.42.1 -j ACCEPT
$IPTABLES -A INPUT -i $INSIDE -s 192.168.43.7 -d
192.168.43.1 -j ACCEPT
$IPTABLES -A INPUT -i $INSIDE -d 192.168.42.1 -j internal
#
$IPTABLES -A INPUT -m state --state RELATED,ESTABLISHED -j
ACCEPT
#
$IPTABLES -A INPUT -p udp --sport 137 --dport 137 -j silent
#
$IPTABLES -A INPUT -i $OUTSIDE -d 0/0 -p udp --dport 5190 -
```

```
j ACCEPT
${IPTABLES -A INPUT -i $OUTSIDE -d 0/0 -p tcp --dport 443 -j
ACCEPT
#
${IPTABLES -A INPUT -j firewalled
```

```
#
# iptables - restore script
#

#!/bin/sh

# chain policies
# set default policies
/usr/sbin/iptables -P INPUT DROP
/usr/sbin/iptables -P OUTPUT ACCEPT
/usr/sbin/iptables -P FORWARD DROP

# flush tables
/usr/sbin/iptables -F
/usr/sbin/iptables -F INPUT
/usr/sbin/iptables -F OUTPUT
/usr/sbin/iptables -F FORWARD
/usr/sbin/iptables -F -t mangle
/usr/sbin/iptables -X
/usr/sbin/iptables -F -t nat

# create DUMP table
/usr/sbin/iptables -N DUMP > /dev/null
/usr/sbin/iptables -F DUMP
/usr/sbin/iptables -A DUMP -p tcp -j LOG
/usr/sbin/iptables -A DUMP -p udp -j LOG
/usr/sbin/iptables -A DUMP -p tcp -j REJECT --reject-with
tcp-reset
/usr/sbin/iptables -A DUMP -p udp -j REJECT --reject-with
icmp-port-unreachable
/usr/sbin/iptables -A DUMP -j DROP

# Stateful table
/usr/sbin/iptables -N STATEFUL > /dev/null
/usr/sbin/iptables -F STATEFUL
/usr/sbin/iptables -I STATEFUL -m state --state
ESTABLISHED,RELATED -j ACCEPT
/usr/sbin/iptables -A STATEFUL -m state --state NEW -i !
eth0 -j ACCEPT
/usr/sbin/iptables -A STATEFUL -j DUMP

# loopback rules
/usr/sbin/iptables -A INPUT -i lo -j ACCEPT
/usr/sbin/iptables -A OUTPUT -o lo -j ACCEPT

# drop reserved addresses incoming
/usr/sbin/iptables -A INPUT -i eth0 -s 127.0.0.0/8 -j DUMP
```

```
/usr/sbin/iptables -A INPUT -i eth0 -s 192.168.0.0/16 -j
DUMP
/usr/sbin/iptables -A INPUT -i eth0 -s 172.16.0.0/12 -j
DUMP
/usr/sbin/iptables -A INPUT -i eth0 -s 10.0.0.0/8 -j DUMP

# allow certain inbound ICMP types
/usr/sbin/iptables -A INPUT -i eth0 -p icmp --icmp-type
destination-unreachable -j ACCEPT
/usr/sbin/iptables -A INPUT -i eth0 -p icmp --icmp-type
time-exceeded -j ACCEPT
/usr/sbin/iptables -A INPUT -i eth0 -p icmp --icmp-type
echo-reply -j ACCEPT

# opened ports (adjust to suit)
/usr/sbin/iptables -A INPUT -p tcp -i eth0 --dport 22 -j
ACCEPT
/usr/sbin/iptables -A INPUT -p tcp -i eth0 --dport 80 -j
ACCEPT

# push everything else to state table
/usr/sbin/iptables -A INPUT -j STATEFUL
```

APPENDIX B :

```
#!/bin/sh
# smartfw-1.0-1 - smartfw.kernel.modules.sh
# Copyright (C) 2002/2003 <WEK000081 LOU CHER YUEN>
# WXES3182 LATIHAN ILMIAH TAHAP AKHIR II
#### Start Firewall ####

## Allow loopback interface
$IPCHAINS -A input -i lo -s 0/0 -d 0/0 -j ACCEPT
$IPCHAINS -A output -i lo -s 0/0 -d 0/0 -j ACCEPT

# Allow packets with ack bit set, they are from an established connection.
$IPCHAINS -A input ! -y -p tcp -s $REMOTENET -d $OUTERNET -j ACCEPT


# Setting up IP Spoofing protection

# Turn on Source Address Verification

if [ -e /proc/sys/net/ipv4/conf/all/rp_filter ]
 then
  for f in /proc/sys/net/ipv4/conf/*/rp_filter
  do
   echo 1 > $f
  done
fi

#Turn on SYN COOKIES PROTECTION
if [ -e /proc/sys/net/ipv4/tcp_syncookies ]
 then
  echo 1 > /proc/sys/net/ipv4/tcp_syncookies
fi

# Now read smartfw.localrules.sh
```