# Modul 12 : Proxy & Firewall



## Pendahuluan

Proxy server adalah sebuah server yang bertindak sebagai penghubung untuk menangani permintaan dari klien untk mencari resource dari server lainnya. Sebuah klien terhubung dengan proxy server , meminta service seperti file , koneksi , halaman web, atau lainnya dari server yang berbeda. Proxy server kan mengevaluasi permintaan dari klien untuk mempermudah dan mengontrol kompleksitas.

### Instalasi Proxy Server (Squid)

Pada praktikum kali ini, kita akan menggunakan repository yang sudah disediakan untuk menginstall Squid. Berikut ini adalah langkah-langkah instalasinya:

* + - 1. Buka terminal dan ketikan :

|  |
| --- |
| #apt-get install squid |

* + - 1. Squid tidak dapat berjalan sebagai root, sebab itu terlebih dahulu kita buat sebuah user khusus untuk menangani service proxy ini

|  |
| --- |
| #useradd squid –d /cache/ -U –r –s /dev/null 2>&1  #mkdir /cache  #chown –R squid /cache/ |

### Konfigurasi Squid

Setelah proses instalasi berhasil dilakukan, langkah selanjutnya adalah menkonfigurasi squid. File konfigurasi ini terletak pada direktori /usr/local/squid/etc.

Berikut akan dibahas bagian-bagian minimal yang perlu di edit pada file squid.conf agar squid proxy server dapat berjalan

|  |  |
| --- | --- |
| **Parameter** | **Keterangan** |
| http\_port 8080 | parameter ini menyatakan port yang akan digunakan squid untuk menerima request HTTP client. |
| icp\_port 0 | parameter ini menentukan port tempat squid mengirim dan menerima request ICP dari proxy lainnya. Untuk mendisable kita set 0. |
| cache\_mem 16 MB | parameter ini menyatakan jumlah memori (RAM) yang digunakan untuk caching. Squid menggunakan memori lebih besar dari nilai yang tertera. Sebaiknya jumlah ini sepertiga dari jumlah memori total. |
| cache\_dir ufs /cache/ 200 16 256 | parameter ini menyatakan jenis sistem storage yang digunakan, direktori cache, besar space yang digunakan untuk caching, jumlah subdirektori tingkat pertama dan jumlah subdirektori tingkat kedua. |
| cache\_mgr admin | parameter ini menyatakan administrator yang bertanggung jawab atas proxy ini. |
| cache\_effective\_user squid | parameter ini menyatakan user yang akan menjalankan service squid |
| cache\_effective\_group squid | parameter ini menyatakan group yang akan menjalankan service squid |
| log\_icp\_queries off | Jika tidak menggunakan ICP sebaiknya diset off |

**Tabel 10.1 Daftar parameter dasar**

Untuk menjalankan squid untuk **pertama kalinya**, lakukan perintah berikut :

|  |
| --- |
| #/usr/local/squid/sbin/squid –z |

Perintah ini akan membuat struktur direktori cache sebanyak dan sedalam yang telah dideklarasikan pada file squid.conf. Kemudian jalankan service, dengan cara :

|  |
| --- |
| #/usr/local/squid/sbin/squid –sY& |

Untuk selanjutnya jika dilakukan perubahan pada squid.conf, untuk merestart service, cukup memakai perintah :

|  |
| --- |
| #/usr/local/squid/sbin/squid –k reconfigure |

### Konfigurasi ACL pada Squid

ACL adalah sebuah list aturan yang mengontrol request yang melewati proxy server. Apakah sebuah request diterima atau ditolak oleh proxy server. Konfigurasi ACL menjadi satu dengan konfigurasi yang lainnya di squid.conf. ACL mengontrol akses berdasarkan IP address asal, IP address tujuan, domain asal, domain tujuan, port, URLs dll. Secara umum konfigurasi ACL dapat dilihat sebagai berikut

|  |  |
| --- | --- |
| **Tujuan** | **ACL** |
| Akses berdasarkan IP address asal | acl [nama\_acl] src [IP/NETMASK]  acl localhost src 127.0.0.1/255.255.255.255 |
| Akses berdasarkan IP address tujuan | acl [nama\_acl] dst [IP/NETMASK]  acl intranet dst 10.14.0.0/255.255.0.0 |
| Akses berdasarkan domain asal | acl [nama\_acl] srcdomain [nama\_domain]  acl intranet srcdomain .ittelkom.ac.id |
| Akses berdasarkan domain tujuan | acl [nama\_acl] dstdomain [nama\_domain]  acl forum dstdomain .kaskus.us |
| Akses berdasarkan port | acl [nama\_acl] port [nomor\_port]  acl SSL\_PORT port 443 |
| Akses berdasarkan method http | acl [nama\_acl] methode [nama\_method]  acl CONNECT method CONNECT  acl GET method GET |
| Akses berdasarkan URL regex | acl [nama\_acl] url\_regex “[nama\_file]”  acl PORN url\_regex “/usr/local/squid/etc/porn.acl”  **catatan** :isi dari file /usr/local/squid/etc/porn.acl adalah list dari kata-kata yang berbau pornografi. |

**Tabel 10.2 Daftar konfigurasi ACL**

Setelah semua acl didefinisikan, sekarang tinggal mengatur aksesnya. Format umumnya adalah sebagai berikut

|  |  |
| --- | --- |
| http\_access deny [nama\_acl] | Untuk menolak request |
| http\_access allow [nama\_acl] | Untuk menerima request |

**Tabel 10.3 Format pengaksesan http**

### Manajemen Bandwidth pada Squid

squid mendukung fitur ini, saat konfigurasi kompilasi harus ditambah dengan parameter –-enable-delay-pools, seperti pada contoh berikut

|  |
| --- |
| #./configure --prefix=/usr/local/squid –enable-delay-pools |

Parameter-parameter yang harus dikonfigurasi pada squid.conf

* delay\_pools [jumlah]

menyatakan berapa banyak pool yang akan dibuat

* delay\_class [pool] [kelas]

menentukan kelas pembagian bandwidth dari setiap pool. Satu pool hanya boleh memiliki satu kelas. Secara umum kelas terbagi menjadi beberapa behaviour.

| Kelas | keterangan |
| --- | --- |
| 1 | Semua bandwidth yang ada akan dibagi rata untuk semua user  Contoh: bandwidth 128Kbps semuanya dipakai untuk browsing |
| 2 | Membatasi pemakaian bandwidth dari total yang ada berdasarkan keperluannya.  Contoh: bandwidth 128Kbps, 28Kbps dipakai untuk email, sisanya 100Kbps dipakai untuk browsing |
| 3 | Membatasi pemakaian bandwidth dari total yang ada berdasarkan network class  Contoh: bandwidth 384Kbps, dipakai untuk 3 subnet yang berbeda, misal lab (10.14.10.0/24), admin (10.14.11.0/24), akses point (10.14.16.0/24). Misalnya sang admin membagi bandwidth sama rata per network, maka masing-masing network akan mendapatkan jatah bandwidth sebesar 128Kbps |

**Tabel 10.4 Kelas pada Pembagian Bandwidth Mikrotik**

* delay\_access[acl]

member batasan kepada siapa saja yang berhak menggunakan delay pools ini

|  |
| --- |
| delay\_parameters 1 32000/32000 8000/8000 1000/64000 |

* delay\_parameters

parameter ini adalah bagian terpenting dari delay pools. Parameter ini akan memberikan aturan main setiap delay pools yang dibuat. Delay parameter mempunyai format yang disesuaikan dengan kelas yang dipakai. Tetapi di setiap kelas yang dipakai ada satu format baku, yaitu restore/max. Restore menunjukkan maksimum kecepatan data yang dapat dilewatkan bila harga max sudah terlampaui, dalam satuan bytes/second. Max menunjukkan besarnya file yang dapat dilewatkan tanpa melalui proses delay, dalam satuan bytes.

## Langkah Praktikum

* di pool nomor 1 semua network mendapat bandwidth yang sama sebesar 1KBps (8Kbps) dengan maksimum file 64KB

|  |
| --- |
| *delay\_parameters 1 1000/64000* |

* di pool nomor 1 squid memakai bandwidth maksimum 32KBps (256Kbps) dari total bandwidth, tiap user mendapat bandwidth maksimum 1KBps (8Kbps) dengan maksimum file (64KB)

|  |
| --- |
| *delay\_parameter 1 32000/32000 1000/64000* |

* di pool nomor 1 squid memakai bandwidth maksimum 32KBps (256Kbps) dari total bandwidth, tiap network maksimum mendapat jatah 8KBps (64Kbps) dantiap user pada satu network mendapat maksimum 1KBps (8Kbps) dengan maksimum file 64KB

|  |
| --- |
| *delay\_parameters 1 -1/-* |

* di pool nomor 1 tidak ada batasan bandwidth alias unlimited.

- Port yang digunakan adalah Port 1999  
- Hanya boleh diakses dari network 192.168.1.0 dan 192.168.1.16 dengan netmask 255.255.255.240  
- Masing-masing network mendapat bandwidth sebesar 256 Kbps (32KB) dengan maksimum file 20MB  
- Blok Port SSH, Telnet, dan FTP  
- Allow HTTP, SMTP, dan POP3  
- Blok URL yang mengandung Galih, Satrio, dan 613080046

### Konfigurasi Squid

Konfigurasi Squid adalah :

* 1. Lakukan penginstalan Squid dengan mengetikan perintah pada terminal :

|  |
| --- |
| *sudo apt-get install squid* |

* 1. Setelah terinstall, untukmelakukan konfigurasi bukalah squid.conf dengan mengetikan : *sudo gedit*

|  |
| --- |
| */etc/squid/squid.conf* |

* 1. Jika sudah terbuka, Buatlah list ACLnya :

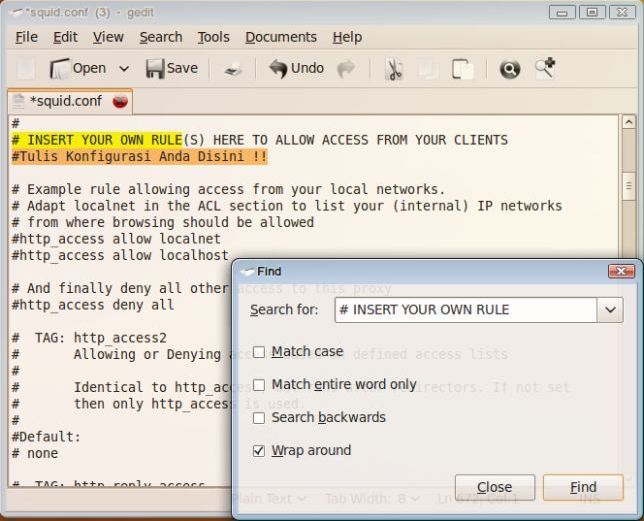
|  |
| --- |
| *#Port http\_port 1999*  *#List ACL acl one src 192.168.1.0/28 acl two src 192.168.1.16/28*  *#Port terlarang aclport\_terlarang port 21 #ftp aclport\_terlarang port 22 #telnet aclport\_terlarang port 23 #ssh*  *#Block URL aclblock\_urlurl\_regex -i (galih+)|(satrio+)|(hernadi+)|(613080046+) http\_access deny block\_url http\_access allow one !port\_terlarang http\_access allow two !port\_terlarang http\_access allow manager localhost http\_access deny manager*  *http\_access allow purge localhost http\_access deny purge http\_access allow localhost http\_access deny all*  *#Membatasiukuran file download 20MB reply\_body\_max\_size 20971520 deny one reply\_body\_max\_size 20971520 deny two* |

* 1. Lalu buatlah konfigurasi Delay Poolsnya :

|  |
| --- |
| *#Konfigurasi Pools delay\_pools 2 delay\_class 1 1 delay class 2 1 delay\_access 1 allow one delay access 1 deny all delay\_access 2 allow two delay\_access 2 deny all delay\_parameters 1 32000/32000 delay\_parameters 2 32000/32000* |

Untuk memasukan konfigurasi di atas, yaitu :

* 1. **memasukan konfigurasi di atas pada tempat yang sudah disediakan (squid.conf) dengan menekan ctrl+F dan ketikan # INSERT YOUR OWN RULE :**



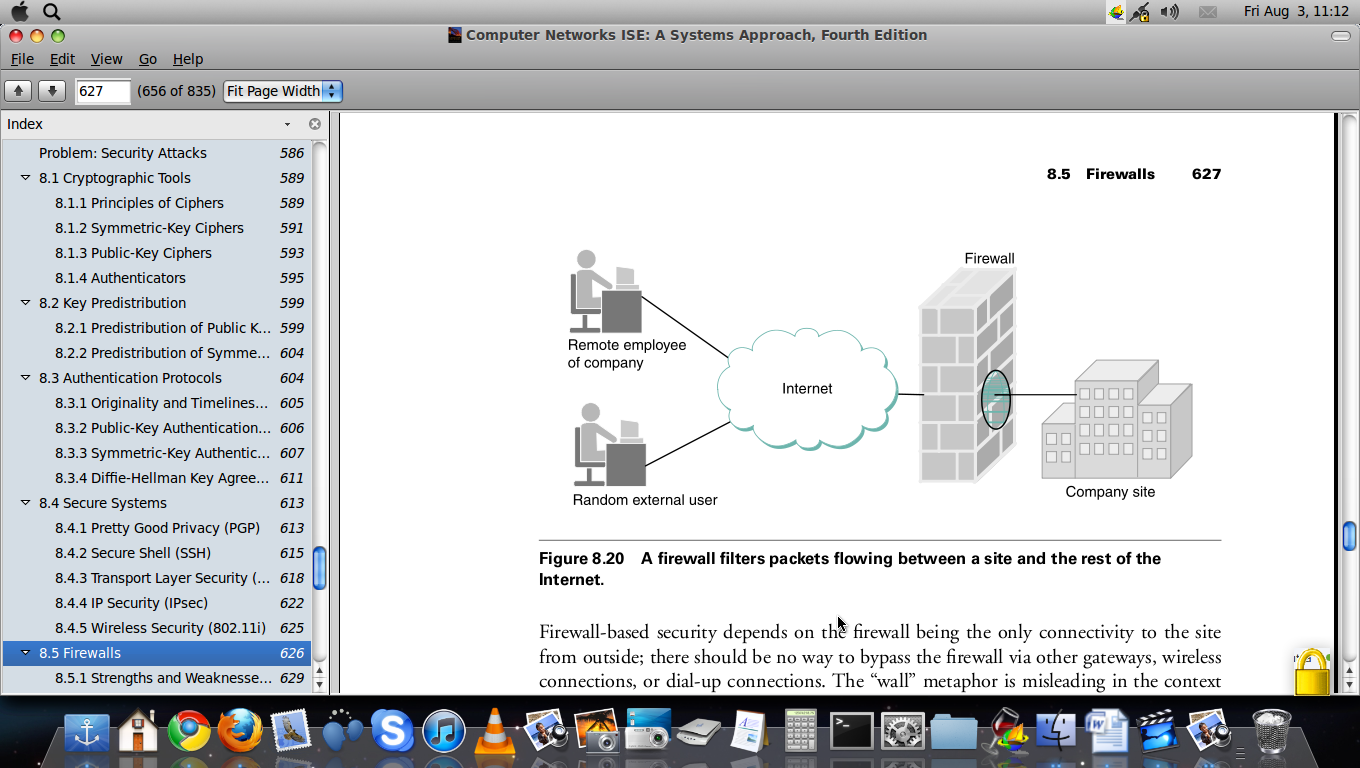
Gambar 9 - 1 Memasukan Konfigurasi ke squid.conf

* 1. Restart Squid setelah memasukan konfigurasi di atas: *sudo /etc/init.d/squid restart*
  2. pasang proxy di browser, dan cek.



## Firewall

Firewall adalah sistem yang merupakan gerbang pengubung antara situs yang akan dilindungi dengan dunia luar atau jaringan secara keseluruhan. Firewall bisanya diimplementasikan pada router namun terdapat juga firewall yang diimplementasikan pada PC.



Gambar 10 - 1 Ilustrasi Firewall

### Iptables

Iptables adalah sebuah service di linux yang menangani tentang aturan aturan firewall. Perintah iptables mengimplementasikan aturan-aturan firewall yang di buat dan mengatur tingkah laku firewall itu sendiri. Iptables sendiri mempunyai 3 individual table, yaitu : filter, NAT, mangle. Diantara table-tabel tersebut, iptable membuat suatu chain untuk menggabungkan/ menyambungkan tabel-tabel tersebut menjadi satu perintah iptables.

Tabel filter default adalah chain untuk input, atau data yang masuk ke firewall, chain untuk output atau data keluar firewall, dan untuk chain forwarding atau data yang di kirim langsung ke firewall, dan untuk chain-chain yang lain. Tabel NAT digunakan untuk melayani NAT dan fungsi-fungsi yang berhubungan. Dan tabel mangle digunakan ketika paket akan di ubah oleh firewall.

Iptables mempunyai sintax yang sangat spesifik. Basic syntax iptables sendiri di diawali dengan perintah iptables sendiri dan diikuti dengan satu atau lebih options, chain, dan matching criteria dan target. Secara umum, sintaxnya adalah sebagai berikut :

|  |
| --- |
| iptables <option> <chain> <matching criteria> <target> |

Dibawah ini adalah sedikit gambaran tentang bagaimana paket tersebut diperlakukan oleh iptables :

* Perjalanan paket yang diforward ke host yang lain:

1. Paket berada pada jaringan fisik, contoh internet
2. Paket masuk ke interface jaringan,contoh eth0
3. Paket masuk ke chain PREROUTING pada tabel Mangle. Chain ini berfungsi untuk me-mangle (menghaluskan) paket, seperti merubah TOS, TTL dan lain-lain.
4. Paket masuk ke chain PREROUTING pada tabel nat. Chain ini fungsi utamanya untuk melakukan DNAT (Destination Network Address Translation)
5. Paket masuk dalam routing, apakah akan diproses oleh host local atau diteruskan ke host lain
6. Paket masuk ke chain FORWARD pada tabel filter. Disini proses pemfilteran utama terjadi
7. Paket masuk ke chain POSTROUTING pada tabel nat. Chain ini fungsi utamanya untuk melakukan SNAT (Source Network Address Translation)
8. Paket keluar menuju interface jaringan, contoh eth1
9. Paket paket kembali berada pada jaringan fisik, contoh LAN

* Perjalanan paket yang ditujukan bagi host local

1. Paket berada pada jaringan fisik, contoh internet
2. Paket masuk ke interface jaringan,contoh eth0
3. Paket masuk ke chain PREROUTING pada tabel Mangle. Chain ini berfungsi untuk me-mangle (menghaluskan) paket, seperti mengubah TOS, TTL dan lain-lain.
4. Paket masuk ke chain PREROUTING pada tabel nat. Chain ini fungsi utamanya untuk melakukan DNAT (Destination Network Address Translation)
5. Paket masuk dalam routing, apakah akan diproses oleh host local atau diteruskan ke host lain
6. Paket masuk ke chain INPUT pada tabel filter untuk mengalami proses penyaringan
7. Paket akan diterima oleh aplikasi local

* Perjalanan paket yang berasal dari host local

1. Aplikasi local menghasilkan paket data yang akan dikirimkan melalui jaringan
2. Paket memasuki chain OUTPUT pada tabel mangle
3. Paket memasuki chain OUTPUT pada tabel nat
4. Paket memasuki chain OUTPUT pada tabel filter
5. Paket mengalami keputusan routing
6. Paket masuk ke chain POSTROUTING pada tabel NAT
7. Paket masuk ke interface jaringan, contoh eth0
8. Paket berada pada jaringan fisik, contoh internet

### Sintax Iptables

* **Tabel**

Seperti yang sudah disebutkan diatas, bahwa syntax iptable sendiri terdiri dari beberapa options, command, actions dan lain-lain. Hal-hal tersebut akan dibahas pada bagian ini.

Tabel iptable memiliki 3 buah tabel yaitu NAT, Mangle, dan filter. Penggunaannya disesuaikan dengan sifat dan karakteristik masing-masing. Fungsi dari masing-masing tabel tersebut adalah sebagai berikut:

* NAT : secara umum digunakan untuk melakukan Network Address translation. NAT adalah penggantian field alamat asal atau alamat tujuan pada sebuah paket
* Mangle : digunakan untuk melakukan penghalusan paket
* Filter : secara umum inilah pemfilteran yang sesungguhnya. Pada bagian ini ditentukan apakah paket akan di DROP, LOG, ACCEPT, atau REJECT
* **Command**

Commad pada baris perntah iptables akan memberitahu apa yang harus dilakukan terhadap lanjutan sintax perintah. Pada umumnya dilakukan penambahan atau penghapusan sesuatu dari tabel atau yang lain.

Berikut adalah sintax command dan keterangannya :

| **Command** | **Keterangan** |
| --- | --- |
| -A , --append | Perintah ini menambahkan aturan pasa akhir chain . Aturan akan ditambahkan di akhir baris pada chain yang bersangkutan ,sehingga akan dieksekusi terakhir. |
| -D, --delete | Perintah ini menghapus aturan pada chain.. Dilakukan dengan cara menyebutkan secara lengkap perintah yang ingin dihapus atau dengan menyebutkan nomor baris dimana perintah akan dihapus. |
| -R, --replace | Penggunaannya sama seperti –delete , tetapi command ini menggantinya dengan entry yang baru. |
| -I , --insert | Memsukkan aturan pada suatu baris di chain. Aturan akan dimasukan pada baris yang disebutkan , dan aturan awal yang menempati baris tersebut akan digeser ke bawah . Demikian baris selanjutnya |
| -L , --list | Perintah ini menampilkan semua aturan pada sebuah tabel. Apabila tabel tidak disebutkan maka seluruh aturan pada semua tabel akan ditampilkan, walaupun tidak ada aturan sama sekali pada sebuah tabel. Perintah ini bias dikombinasikan dengan option –v (verbose), -n (numeric) dan –x (exact). |
| -F, --flush | Perintah ini mengosongkan aturan pada sebuah chain . Apabila chain tidak disebutkan maka semua chain akan di *flush* |
| -N , --new-chain | Perintah tersebut untuk membuat chain baru |
| -X , --delete-chain | Perintah ini kuntuk menghapus chain yang disebutkan , Agar perintah ini berhasil tidak boleh ada aturan lain yang mengacu pada chain tersebut. |
| -P , --policy | Perintah ini membuat kebijakan default pada sebuah chain. Sehingga jika ada sebuah paket yang tidak memenuhi aturan pada baris-baris yang telah didefinisikan, maka paket akan diperlakukan sesuai dengan kebijakan default |
| -E , --rename-chain | Perintah untuk merubah nama suatu chain |

* Options

Options digunakan untuk dikombinasikan dengan command tertentu yang akan menghasilkan suatu variasi perintah.

|  |  |  |
| --- | --- | --- |
| **Option** | **Command Pemakai** | **Keterangan** |
| - v  -- verbose | --list  --append  --insert  --delete  --replace | Memberikan output yang lebih detail, utamanya digunakan dengan – list. Jika digunakan dengan – list akan menampilkan K (x1.000), M (1.000.000) dan G (1.000.000.000). |
| -x  --exact | --list | Memberikan output yang lebih tepat |
| -n  --numeric | --list | Memberikan output yang berbentuk angka. Alamat IP dan nomor port akan ditampilkan dalam bentuk angka dan bukan hostname atau nama service |
| --line-number | --list | Akan menampilkan nomor dari daftar aturan. Hal in akan mempermudah bagi kita untuk melakukan modifikasi aturan. Jika kita mau menysipkan atau mengkapus aturan dengan nomor tertentu. |
| --modprobe | All | Memerintahkan iptables untuk memanggil modul tertentu. Bias digunakan bersamaan dengan semua command |

* Generic Matches

Generic matches artinya pendefinisian criteria yang berlaku secara umum. Dengan kata lain sintax tersebut akan sama untuk semua protocol. Setelah protocol didefinisikan maka baru didefinisikan aturan yang lebih spesifik yang dimiliki oleh protocol tersebut. Hal ini dilakukan karena tiap protocol memiliki karakteristik yang berbeda. Sehingga memerlukan perlakuan khusus.

|  |  |
| --- | --- |
| **Match** | **Keterangan** |
| -p , --protocol | Digunakan untuk mengecek tipe protocol tertentu. Contoh protocol yang umum adalah TCp, UDP, ICMP, dan ALL. |
| -s, --src, --source | Kriteria ini digunakan untuk mencocokan paket berdasarkan alamat IP. Alamat disini bisa berbentuk tunggal seperti 192.168.1.0/255.255.255.0 atau 192.168.1/24 yang artinya semua alamat 192.168.1.x . |
| -d, --dst, --destination | Digunakan untuk mencocokan paket berdasarkan alamat tujuan. Penggunaanya sama dengan match --src |
| -I, --in-interface | Match ini bertugas untuk mencocokan paket berdasarkan interface dimana paket datang. Match ini hanya berlaku pada INPUT , FORWARD dan PREROUTING. |
| -o, --out-intrface | Match ini bertugas untuk mencocokan paket berdasarkan interface dimana paket keluar. Match ini hanya berlaku pada INPUT,FORWARD dan PREROUTING. |

* Implicit Matches

Implicit Matches adalah match yang spesifik untuk tipe protokol tertentu. Implicit Match merupakan sekumpulan rule yang akan diload setelah tipe protokol disebutkan. Ada 3 Implicit Match berlaku untuk tiga jenis protokol, yaitu TCP matches, UDP matches dan ICMP matches

* TCP Matches

| **Match** | **Keterangan** |
| --- | --- |
| --sport, --source-port | Match ini berguna untuk mencocokan paket berdasarkan port asal. Dalam hal ini kita bisa mendefinisikan nomor port atau nama *service* nya. Daftar nama *service* dana nomor port yang bersesuaian dapat dilihat di /etc/services  --sport juga bisa dituliskan untuk range port tertentu , misalkan ingin mendefinisikan antara port 20 sampai 80. Maka bisa menuliskan –sport 22:80. Jika bagian dari port tersebut dihilangkan seperti :80 berarti dari port 0 sampai port 80, sedangkan jika 20: , berarti dari port 20 sampai port 65535. |
| --dport, --destination-port | Penggunaannya sama dengan --sport |
| --tcp-flags | Digunakan untuk mencocokan paket berdasarkan UTP flag yang ada pada paket tersebut. Pertama , pengecekan akan mengambil daftar flag yang akan diperbandingkan . kedua akan memeriksa paket yang diset 1, atau on. Pada kedua list, masing-msaing entrynya harus dipisahkan oleh koma dan tidak boleh ada spasi antar entry, kecuali spasi antar kedua list. Match ini mengenali SYN, ACK, FIN,RST,URG,PSH .Selain itu , kita juga menuliskan ALL dan NONE . Match ini juga bisa menggunakan inverse. |
| --syn | Match ini akan memeriksa apakah flag SYN diset dan ACK dan tidak diset . Perintah ini sama artinya jika kita menggunakan match --tcp-flags SYN,ACK, FIN SYN. Paket dengan match diatas digunakan untuk melakukan request TCP yang baru terhadap server. |

* UDP Matches

Karena bahwa protokol UDP bersifat connectionless, maka tidak ada flags yang mendeskripsikan status paket untuk untuk membuka atau menutup koneksi. Paket UDP juga tidak memerlukan acknowledgement. Sehingga Implicit Match untuk protokol UDP lebih sedikit daripada TCP. Ada dua macam match untuk UDP:

|  |
| --- |
| --sport atau --source-port  --dport atau --destination-port |

* ICMP Matches

Paket ICMP digunakan untuk mengirimkan pesan-pesan kesalahan dan kondisi-kondisi jaringan yang lain. Hanya ada satu implicit match untuk tipe protokol ICMP, yaitu

|  |
| --- |
| --icmp-type |

* Target atau Jump

Target atau jump adalah perlakuan yang diberikan terhadap paket-paket yang memenuhi kriteria atau match. Jump memerlukan sebuah chain yang lain dalam tabel yang sama. Chain tersebut nantinya akan dimasuki oleh paket yang memenuhi kriteria. Analoginya ialah chain baru nanti berlaku sebagai prosedur/fungsi dari program utama. Sebagai contoh dibuat sebuah chain yang bernama tcp\_packets. Setelah ditambahkan aturan-aturan ke dalam chain tersebut, kemudian chain tersebut akan direferensi dari chain input.

Iptables -A INPUT -p tcp -j tcp packet

| **Target** | **Keterangan** |
| --- | --- |
| -j ACCEPT  --jump ACCEPT | Ketika paket cocok dengan daftar match dan target ini diberlakukan , maka paket tidak akan melalui baris-baris aturan yang lain dalam chain yang lain yang mereferensi chain tersebut . akan tetapi masih akan memasuki chain-chain pada table yang lain seperti biasa. |
| -j RETURN  --jump RETURN | Target ini men-drop paket dan menolak untuk memproses lebih jauh. Dalam beberapa kasus mungkin hal ini kurang baik, karena akan meninggalkan *dead socket* antara klien dan server . Paket yang menerima target DROP benar-benar mati dan target tidak akan mengirim informasi tambahan dalam bentuk apapun kepada klien atau server. |
| -j DROP  --jump DROP | Target ini akan membuat packet berhenti menlintasi aturan-aturan pada chain dimana paket tersebut menemui target RETURN . jika chain merupakan subchain dari chain yang lain , maka paket akan kembali ke superset chain diatasnya dan masuk ke baris aturan berikutnya. Apabila chain adalah chain utama misalnya INPUT , maka paket akan dikembalikan kepada kebijakan default dari chain tersebut. |
| -j MIRROR | Apabila computer A menjalankan target seperti contoh diatas, kemudian computer B melakukan koneksi http ke computer A. Maka yang akan muncul pada browser adalah website computer B itu sendiri. Karena fungsi utama target ini bekerja pada chain INPUT, FORWARD, dan PREROUTING atau chain buatan yang dipanggil melalui chain tersebut. |

## Langkah Praktikum

### Langkah dengan Menggunakan Iptables

* Allowing DNS Access To Your Firewall

|  |
| --- |
| *iptables -A OUTPUT -p udp -o eth0 --dport 53 --sport 1024:65535 -j ACCEPT*  *iptables -A INPUT -p udp -i eth0 --sport 53 --dport 1024:65535 -j ACCEPT*  #Allowing WWW And SSH Access To Your Firewall  -------------------------------------------------------  # Allow previously established connections  # - Interface eth0 is the internet interface  #---------------------------------------------------------------  *iptables -A OUTPUT -o eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT*  #---------------------------------------------------------------  # Allow port 80 (www) and 22 (SSH) connections to the firewall  #---------------------------------------------------------------  *iptables -A INPUT -p tcp -i eth0 --dport 22 --sport 1024:65535 -m state --state NEW -j ACCEPT*  *iptables -A INPUT -p tcp -i eth0 --dport 80 --sport 1024:65535 -m state --state NEW -j ACCEPT* |

* DROP ALL SYN from pc89184

|  |
| --- |
| *iptables – A INPUT –p TCP –s 137.189.89.184 –syn-j DROP* |

**Melakukan por t forwarding dari eth0 (internet) dari port tcp 80 ke mail server kita di 10.14.1.1**

|  |
| --- |
| *iptables –t nat -A PREROUTING –i eth0 –p tcp --dport 80 –j DNAT –to 10.14.1.1.1:80* |