MODUL PRAKTIKUM JARINGAN KOMPUTER

WWW.AJK.IF.ITS.AC.ID

MODUL 05

Laboratorium Arsitektur dan Jaringan Komputer
Jurusan Teknik Informatika Ruang IF 307
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember Surabaya

Apa Itu Firewall?

Firewall adalah seperangkat software atau hardware yang menyaring data yang keluar dan masuk dalam sebuah jaringan komputer. Jika ada data yang dicurigai membawa hal berbahaya dan dapat merusak, seperti virus, spyware, dan trojan, firewall akan melarangnya untuk keluar atau masuk ke dalam jaringan. Dulu, firewall bertugas seperti keamanan hotel atau bandara yang hanya menyaring data masuk, dan mengabaikan data yang keluar dari jaringan. Tapi, seiring dengan meningkatnya ancaman keamanan di dunia maya, kini firewall pun dirancang untuk mengecek data yang keluar atau masuk dari sebuah jaringan

Mengapa Perlu Firewall?

Tanpa firewall, semua komputer yang berada dalam jaringan tersebut dapat diakses oleh siapa saja melalui Internet. Seseorang yang mengetahui apa yang dilakukan oleh para karyawan kantor dengan komputernya dapat memeriksa port apa saja yang terbuka di komputer tersebut, kemudian mencoba melakukan koneksi FTP, telnet, dan sebagainya. Selain itu, firewall dapat membuat kebijakan serupa untuk web server, e-mail server, telnet server, dan lain-lain. Lebih jauh lagi, firewall dapat menyaring website apa saja yang boleh dan yang tidak boleh diakses oleh komputer karyawan, apakah file-file boleh dikirimkan ke luar melalui jaringan, dan lain sebagainya.

Metode Pada Firewall

Dalam proses kerjanya, firewall menggunakan metode untuk dapat melakukan proses penyeleksian data dan juga konten yang akan diizinkan untuk diakses atau tidak. Berikut ini adalah beberapa metode yang sering digunakan firewall pada umumnya.

1. **Packet filtering**
   Paket data diperiksa menggunakan seperangkat aturan penyaringan, informasi dari paket yang diperiksa seperti alamat asal, alamat tujuan, protokol dan port alamat tujuan . Paket yang lulus proses penyaringan diizinkan untuk meneruskan perjalanannya, sedangkan yang tidak lulus penyaringan akan dihapus.

2. **Proxy service**
   Metode firewall ini bekerja pada layer aplikasi, bertindak sebagai perantara antara request dari salah satu network ke network lainnya. Data yang berasal dari Internet diterima oleh firewall kemudian diteruskan kepada komputer yang merequestnya, begitu juga sebaliknya.

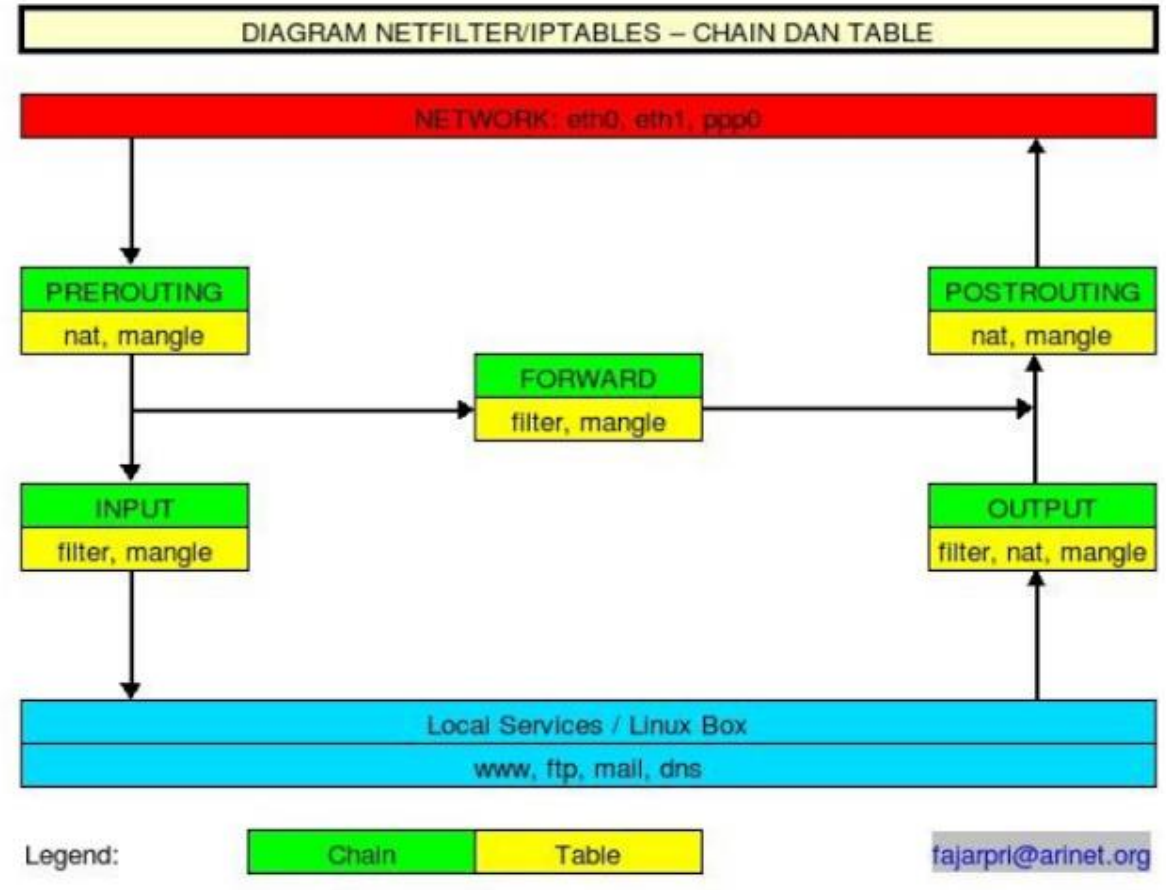
3. **Stateful inspection**
   Metode membandingkan bagian kunci dari paket data tersebut dengan database data-data terpercaya. Data yang keluar dari firewall ditandai dengan ciri-ciri khusus, selanjutnya data yang masuk akan dibandingkan dengan ciri-ciri khusus tersebut. Jika dalam proses perbandingan terdapat kecocokan, data diizinkan masuk. Jika terdapat ketidakcocokan, firewall akan menghapus data tersebut.

Lebih Jauh Tentang Packet Filtering Firewall

Packet filter adalah sebuah software yang memeriksa header dari paket ketika paket tersebut lewat, dan memutuskan tindakan apa yang dilakukan terhadap paket tersebut. Apakah paket tersebut di-DROP (misal dengan menghapus paket tersebut), ACCEPT (misal, paket tersebut diteruskan ke tujuannya), atau hal lain yang lebih kompleks.

Pada Linux, packet filtering ditanamkan pada kernel (sebagai modul kernel, atau digabungkan ke dalam kernel). Penerapan packet filtering sudah cukup lama sejak kernel 1.1. Versi pertamanya, masih banyak mencontoh cara kerja ipfw milik BSD (Sistem Operasi buatan University California atBerkeley), dibuat oleh Alan Cox pada akhir 1994. Berkembang menjadi ipfwadm pada kernel 2.0, ipchains pada kernel 2.2 dan terakhir iptables sejak kernel 2.4.

IPTABLES



IPTABLES adalah paket aplikasi (program berbasis Linux) yang saat ini sudah menjadi platform untuk membuat (mensetup) firewall hampir di kebanyakan distro Linux. Dengan menggunakan IPtables seorang pengguna / admin jaringan bisa mengatur lalu lintas paket data yang keluar masuk pada router atau server yang menjadi gateway (pintu gerbang) antara jaringan perusahaan/lokal (LAN) dengan jaringan publik (WAN/internet).

1. Penulisan IPTABLES

 # iptables [–t table] command [match][target/jump]

Contohnya :

 # iptables –P FORWARD ACCEPT

Keterangan :

- *Table :* IPTables memiliki 3 buah tabel, yaitu NAT, MANGLE dan FILTER. Penggunannya disesuaikan dengan sifat dan karakteristik masing-masing.

- *Command :* Command pada baris perintah IPTables akan memberitahu apa yang harus dilakukan terhadap lanjutan sintaks perintah. Umumnya dilakukan penambahan atau penghapusan sesuatu dari tabel atau yang lain.

Command	Keterangan
-A **--append**	Perintah ini menambahkan aturan pada akhir chain. Aturan akan ditambahkan di akhir baris pada chain yang bersangkutan, sehingga akan dieksekusi terakhir
-D **--delete**	Perintah ini menghapus suatu 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**	Memasukkan aturan pada suatu baris di chain. Aturan akan dimasukkan pada baris yang disebutkan, dan aturan awal yang menempati baris tersebut akan digeser ke bawah. Demikian pula baris-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. *Command* ini bisa 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 akan membuat chain baru.
-X **--delete- chain**	Perintah ini akan menghapus chain yang disebutkan. Agar perintah di atas berhasil, tidak boleh ada aturan lain yang mengacu kepada 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 ini.
-E **--rename- chain**	Perintah ini akan merubah nama suatu chain.

- *Option :* Option digunakan dikombinasikan dengan command tertentu yang akan menghasilkan suatu variasi perintah.

Option	Command	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 ataupun nama aplikasi/servis.
--line-number	**--list**	Akan menampilkan nomor dari daftar aturan. Hal ini akan mempermudah bagi kita untuk melakukan modifikasi aturan, jika kita mau menyisipkan atau menghapus aturan dengan nomor tertentu.
--modprobe	**All**	Memerintahkan IPTables untuk memanggil modul tertentu. Bisa digunakan bersamaan dengan semua *command*.

- *Generic Matches :* Generic Matches artinya pendefinisian kriteria yang berlaku secara umum. Dengan kata lain, sintaks generic matches akan sama untuk semua protokol. Setelah protokol didefinisikan, maka baru didefinisikan aturan yang lebih spesifik yang dimiliki oleh protokol tersebut. Hal ini dilakukan karena tiap-tiap protokol memiliki karakteristik yang berbeda, sehingga memerlukan perlakuan khusus.

Match	Keterangan
-p **--protocol**	Digunakan untuk mengecek tipe protokol tertentu. Contoh: TCP, UDP, ICMP dan ALL. Daftar protokol bisa dilihat pada **/etc/protocols**. Tanda inversi juga bisa diberlakukan di sini, misal kita menghendaki semua protokol kecuali icmp, maka kita bisa menuliskan **--protokol ! icmp** yang berarti semua kecuali icmp.
-s **--src** **--source**	Kriteria ini digunakan untuk mencocokkan paket berdasarkan alamat IP asal. Alamat di sini bisa berberntuk alamat tunggal seperti 192.168.1.1, atau suatu alamat network menggunakan netmask misal 192.168.1.0/255.255.255.0, atau bisa juga ditulis 192.168.1.0/24 yang artinya semua alamat 192.168.1.x. Kita juga bisa menggunakan inversi.

-d **--dst** **--destination**	Digunakan untuk mecocokkan paket berdasarkan alamat tujuan. Penggunaannya sama dengan *match* **–src**
-i **--in-interface**	*Match* ini berguna untuk mencocokkan paket berdasarkan interface di mana paket datang. *Match* ini hanya berlaku pada chain INPUT, FORWARD dan PREROUTING
-o **--out-interface**	Berfungsi untuk mencocokkan paket berdasarkan interface di mana paket keluar. Penggunaannya sama dengan **--in-interface**. Berlaku untuk chain OUTPUT, FORWARD dan POSTROUTING

- *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.

- *Target/Jump: T*arget 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_packets

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 tersebut atau chain yang lain yang mereferensi chain tersebut. Akan tetapi paket masih akan memasuki chain-chain pada tabel yang lain seperti biasa.
-j DROP --jump DROP	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 *client* dan *server*. Paket yang menerima target DROP benar-benar mati dan target tidak akan mengirim informasi tambahan dalam bentuk apapun kepada client atau server.

-j RETURN --jump RETURN	Target ini akan membuat paket berhenti melintasi 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* di atasnya 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 komputer A menjalankan target seperti contoh di atas, kemudian komputer B melakukan koneksi http ke komputer A, maka yang akan muncul pada browser adalah website komputer B itu sendiri. Karena fungsi utama target ini adalah membalik *source address* dan *destination address.* Target ini bekerja pada chain INPUT, FORWARD dan PREROUTING atau chain buatan yang dipanggil melalui chain tersebut.

2. IPTables memiliki beberapa buah tabel yaitu NAT, MANGLE, dan FILTER. Penjelasannya adalah:
   a. **Table Mangle**: tabel yang bertanggung jawab untuk melakukan penghalusan (mangle) paket seperti merubah quality of service (QOS), TTL, dan MARK di header TCP. Biasanya tabel ini jarang digunakan di lingkungan SOHO.
   b. **Table Filter**: yaitu tabel yang bertanggung jawab untuk pemfilteran paket. Tabel ini mempunyai 3 rantai (chain) yaitu:
      1. **Rantai Forward** yaitu rantai yang memfilter paket-paket yang akan ke server yang dilindungi oleh firewall. Rantai ini digunakan ketika paket-paket datang dari IP Publik dan bukan dari IP lokal.
      2. **Rantai Input:** yaitu rantai yang memfilter paket-paket yang ditujukan ke firewall.
      3. **Rantai Output**: yaitu rantai yang memfilter paket-paket yang berasal dari firewall.
   c. **Tabel NAT**: yaitu rantai yang bertanggung jawab untuk melakukan *Network Address Translation* (NAT). NAT yaitu mengganti field asal atau alamat tujuan dari sebuah paket. Pada tabel ini terdapat 2 rantai, yaitu:
      1. **Rantai Pre-Routing:** Merubah paket-paket NAT di mana alamat tujuan dari paket-paket tersebut terjadi perubahan. Biasanya dikenal dengan destination NAT atau DNAT.
      2. **Rantai Post-Routing**: Mengubah paket-paket NAT di mana alamat sumber dari paket-paket tersebut terjadi perubahan. Biasanya dikenal dengan source NAT atau SNAT.

3. **Forward chain** **:** meneruskan paket dari atau ke luar interface.
   Input chain **:** paket yang masuk ke dalam interface.
   Output chain **:** paket yang keluar dari interface.

4. Jenis NAT:
    A. *Full-cone:* adalah bentuk yang paling membatasi perilaku NAT, dimana binding dari local address dan port ke public-side dan port, ketika didirikan, dapat digunakan oleh semua remote host pada setiap remote port address.
    B. *Restricted-cone:* adalah salah satu dimana NAT hanya dapat diakses oleh host tujuan, meskipun dalam kasus ini host tujuan dapat mengirim paket dari port address setelah binding dibuat.
    C. *Port-restricted-cone:* adalah salah satu di mana NAT dapat diakses oleh remote host, meskipun dalam kasus ini remote host harus menggunakan source port address yang sama sebagai original port address yang memicu NAT.
    D. *Symmetric:* adalah NAT di mana pemetaan NAT mengacu khusus untuk koneksi antara local host address dan port number dan destination address dan port number dan local address dan port ke public side address dan port.

5. Ada 2 tipe NAT:
    A. *Source NAT* adalah ketika anda mengubah alamat asal dari paket pertama dengan kata lain anda merubah dari mana koneksi terjadi. *Source NAT* selalu dilakukan setelah routing, sebelum paket keluar ke jaringan. Masquerading adalah contoh dari SNAT. Untuk melakukan *Source NAT* anda harus merubah asal dari koneksi. Hal ini dilakukan di chain POSTROUTING, saat sebelum keluar. Hal ini sangat penting, dikarenakan berarti tools lain yang di dalam router itu (routing, packet filtering) akan melihat paket itu tidak berubah. Hal ini juga berarti opsi '-o' (outgoing interface) juga bisa digunakan. Source dispesifikasikan dengan menggunakan '-j SNAT', dan juga opsi '--to-source' untuk menspesifikasikan sebuah alamat IP, range alamat IP dan port atau range port (hanya untuk protokol UDP dan TCP) yang sifatnya optional. Contoh dari *Source NAT:*
        - Mengubah alamat asal ke 192.168.0.1

```
# iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 192.168.0.1
```

        - Mengubah alamat asal ke 192.168.0.1, 192.168.0.2, 192.168.0.3

```
# iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 192.168.0.1-192.168.0.3
```

        - Mengubah alamat asal ke 192.168.0.1, port 1-1023

```
# iptables -t nat -A POSTROUTING -p tcp -o eth0 -j SNAT --to 192.168.0.1:1-1023
```

    B. *Destination NAT* adalah ketika anda mengubah alamat tujuan dari paket pertama dengan kata lain anda merubah ke mana komunikasi terjadi. *Destination NAT* selalu dilakukan sebelum routing, ketika paket masuk dari jaringan. Port forwarding, load sharing dan transparent proxy semuanya adalah bentuk dari DNAT. Destination NAT dilakukan pada chain PREROUTING, pas ketika paket masuk, hal ini berarti semua tools di dalam router akan melihat paket akn pergi ke tujuan yang sebenarnya . Hal ini juga berarti bahwa opsi '-i' (incoming interface) bisa digunakan. Destination NAT dispesifikasikan dengan menggunakan '-j DNAT' dan opsi '--to-destination' menspesifikasikan sebuah

alamat IP, range alamat IP dan range dari port (hanya untuk protokol UDP dan TCP) yang sifatnya optional.

- Merubah alamat tujuan ke 10.151.36.1

```
# iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 10.151.36.1
```

- Merubah alamat tujuan ke 10.151.36.1, 10.151.36.2, 10.151.36.3

```
# iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 10.151.36.1-10.151.36.3
```

- Merubah alamat tujuan dari lalu lintas web ke 10.151.36.1 port 8080

```
# iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth0 -j DNAT --to 10.151.36.1:8080
```

Contoh kasus penggunaan IPTABLES

1. **Blok semua koneksi yang MASUK dari alamat IP tertentu**

```
# iptables -A INPUT -s [alamat_ip] -j DROP
```

Keterangan:

-A INPUT

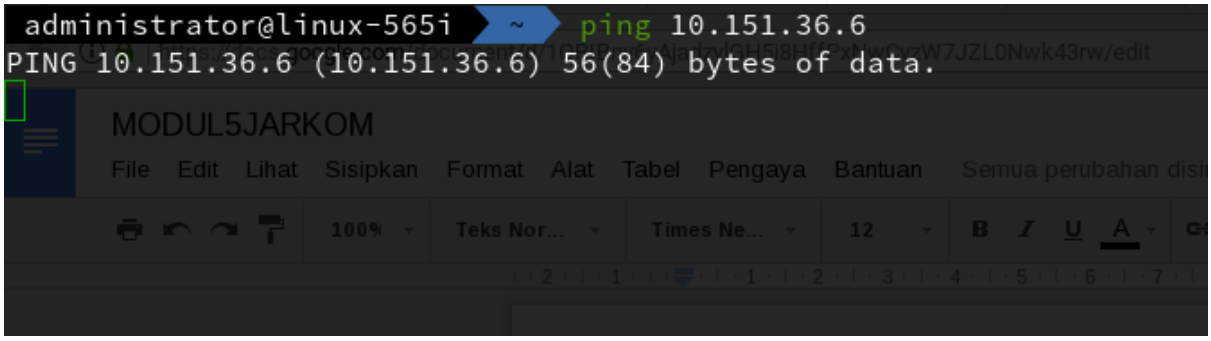
Menambahkan (-A) rules pada chain input (INPUT)

-s [alamat_ip]

Alamat ip yg akan diblok (source IP Address)

-j DROP

Paket yang memenuhi kriteria tersebut akan di drop



Quiz: Coba cari perbedaan saat menggunakan perintah -j DROP dengan -j REJECT

2. **Blok koneksi pada port tertentu**

```
# iptables -A INPUT -p [protokol] --dport [port_number/general_name] -j DROP
```

Keterangan:

-A INPUT

Menambahkan (-A) rules pada cain input (INPUT)

-p [protokol]

Protokol yang akan di-filter oleh iptables

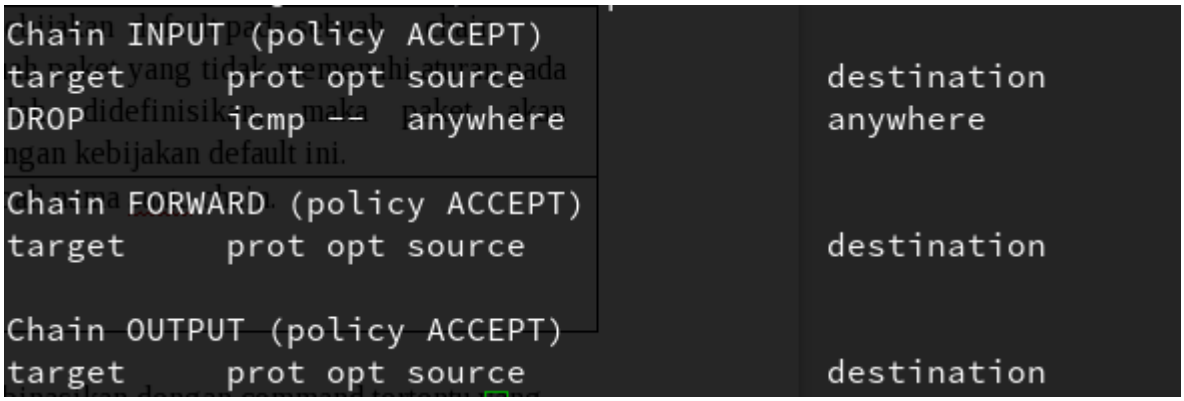
--dport [port_number/general_name]

Nomor port atau nama port yang sudah dikenal, misal: ssh

3. **Melihat list**

Untuk melihat semua rule yang ada pada IPTables digunakan perintah:

```
# iptables -L
```



4. **NAT (Network Address Translation)**

Untuk melakukan overloading NAT (akan dijelaskan asisten apa itu overloading NAT) maka perintah yang digunakan adalah sebagai berikut:

```
# iptables -t nat -A POSTROUTING -o [interface] -j MASQUERADE -s [alamat_ip]
```

Keterangan:

-t nat

Memilih tabel NAT untuk dimasukkan ke rule.

-A INPUT

Menambahkan (-A) rules pada cain POSTROUTING

-o [interface]

Interface yang digunakan untuk jalur keluar

-j MASQUERADE

Pilihan agar paket yang cocok di'masquerade', dengan mengganti source address menjadi IP router

5. **Flushing**

Menghapus semua rules atau rule tertentu pada iptables.

```
# iptables -F
```

Keterangan:

Menghapus semua rule yang ada

NOTE !!!

Role diurutkan dari yang spesifik dahulu kemudian yang lebih umum. Contoh:

-> Allow IP address tertentu

-> Allow subnet tertentu

-> Block semua koneksi

Karena soal praktikum tidak semudah yang kalian bayangkan maka silahkan baca-baca referensi berikut ini:

- **Manual resmi iptables ($man iptables)**
- **http://ipset.netfilter.org/iptables.man.html**
- **https://www.howtogeek.com/177621/the-beginners-guide-to-iptables-the-linux-firewall/**
- **https://help.ubuntu.com/community/IptablesHowTo**
- **https://www.netfilter.org/documentation/HOWTO/NAT-HOWTO-6.html**

Latihan (Gunakan topologi pada soal shift “Routing dan Subnetting”)

1. Komputer di subnet NDUS tidak diizinkan mengakses server ZAID
2. Komputer di subnet POLI tidak dapat mengakses komputer di subnet KIYEP pada pukul 10.00 - 19.00
3. Buatlah agar tiap subnet dapat mengakses internet menggunakan SNAT
4. Block port 80 agar KOPET tidak bisa mengakses HTTP