IPTables part II

@danangnurfauzi

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 Pemakai	Keterangan

`-v`
`--verbose`
	`--list`
	`--append`
	`--insert`
	`--delete`
	`--replace`
		Memberikan output yang lebih detail, utamanya digunakan dengan `--list`. Jika digunakan dengan
		`--list`, akan menampilkam 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 ni akan mempermudah bagi kita untuk melakukan modifikasi aturan, jika kita mau meyisipkan 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 protokol yang umum adalah 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. Penggunannya sama dengan `--in-interface`. Berlaku untuk chain OUTPUT, FORWARD dan POSTROUTING

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 mecocokkan paket berdasarkan port asal. Dalam hal ini kia bisa mendefinisikan nomor port atau nama *service*-nya. Daftar nama service dan nomor port yang bersesuaian dapat dilihat di `/etc/services`. `--sport` juga bisa dituliskan untuk range port tertentu. Misalkan kita ingin mendefinisikan range antara port 22 sampai dengan 80, maka kita bisa menuliskan `--sport 22:80.` Jika bagian salah satu bagian pada range tersebut kita hilangkan maka hal itu bisa kita artikan dari port 0, jika bagian kiri yang kita hilangkan, atau 65535 jika bagian kanan yang kita hilangkan. Contohnya `--sport :80` artinya paket dengan port asal nol sampai dengan 80, atau `--sport 1024:` artinya paket dengan port asal 1024 sampai dengan 65535.Match ini juga mengenal inversi.

`--dport`
 	
`--destination-port`
	Penggunaan *match* ini sama dengan match `--source-port.`

`--tcp-flags`	Digunakan untuk mencocokkan paket berdasarkan TCP *flags* yang ada pada paket tersebut. Pertama, pengecekan akan mengambil daftar *flag* yang akan diperbandingkan, dan kedua, akan memeriksa paket yang di-*set* 1, atau *on*. Pada kedua *list*, masing-masing entry-nya 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 inversi.
`--syn`	*Match* ini akan memeriksa apakah flag SYN di-*set* dan ACK dan FIN tidak di-*set*. Perintah ini sama artinya jika kita menggunakan *match* `--tcp-flags SYN,ACK,FIN SYN` Paket dengan *match* di atas digunakan untuk melakukan *request* koneksi 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
```

Explicit Matches

a. MAC Address

b. Multiport Matches

c. Owner Matches

d. State Matches

MAC Address

- Match jenis ini berguna untuk melakukan pencocokan paket berdasarkan MAC source address.
- Perlu diingat bahwa MAC hanya berfungsi untuk jaringan yang menggunakan teknologi ethernet.

```
iptables -A INPUT -m mac -mac-source 00:00:00:00:00:01
```

Multiport Matches

- Ekstensi Multiport Matches digunakan untuk mendefinisikan port atau port range lebih dari satu, yang berfungsi jika ingin didefinisikan aturan yang sama untuk beberapa port.
- Tapi hal yang perlu diingat bahwa kita tidak bisa menggunakan port matching standard dan multiport matching dalam waktu yang bersamaan.

```
iptables -A INPUT -p tcp -m multiport --source-port 22,53,80,110
```

Owner Matches

- Penggunaan match ini untuk mencocokkan paket berdasarkan pembuat atau pemilik/owner paket tersebut.
- Match ini bekerja dalam chain OUTPUT, akan tetapi penggunaan match ini tidak terlalu luas, sebab ada beberapa proses tidak memiliki owner (??).

```
iptables -A OUTPUT -m owner --uid-owner 500
```

State Matches

- Match ini mendefinisikan state apa saja yang cocok.
- Ada 4 state yang berlaku, yaitu NEW, ESTABLISHED, RELATED dan INVALID.

State Matches

- NEW digunakan untuk paket yang akan memulai koneksi baru.
- ESTABLISHED digunakan jika koneksi telah tersambung dan paket-paketnya merupakan bagian dari koneki tersebut.
- RELATED digunakan untuk paket-paket yang bukan bagian dari koneksi tetapi masih berhubungan dengan koneksi tersebut, contohnya adalah FTP data transfer yang menyertai sebuah koneksi TCP atau UDP.
- INVALID adalah paket yang tidak bisa diidentifikasi, bukan merupakan bagian dari koneksi yang ada

State Matches

```
iptables -A INPUT -m state --state RELATED,ESTABLISHED
```

Target/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_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 kompuuter 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.

Target/Jump

- Beberapa target yang lain biasanya memerlukan parameter tambahan:

a. LOG Target

b. REJECT Target

c. SNAT Target

d. DNAT Target

e. MASQUERADE Target

f. REDIRECT Target