

FIREWALL DI UBUNTU

Prolog :

Pada Sistem Operasi Linux modul firewall menggunakan Netfilter (tetapi kebanyakan orang mengenalnya dengan nama iptables), Netfilter sudah build-in kedalam kernel, sehingga apabila dilakukan pembaruan kernel maka Netfilter otomatis akan juga terbaharui.

Untuk menggunakan Netfilter maka secara standar(default) dapat menggunakan perintah iptables.

Iptables

Iptables ada perintah untuk menggunakan fungsi netfilter yang ada dalam kernel linux.

Contoh kode sederhana untuk pengaturan firewall dengan perintah iptables:

```
#!/bin/bash
# bersihkan seluruh rantai aturan yang ada
iptables -F
# tentukan aturan standar untuk setiap rantai yang telah ada didefinisinya
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
# ijinakan terjadinya inisialisasi hubungan oleh paket yang dikirim keluar
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
# jatuhkan yang lainnya
iptables -A INPUT -i eth+ -p udp -j DROP
iptables -A INPUT -i eth+ -p tcp -m tcp --syn -j DROP
# terima setiap dari localhost
iptables -A INPUT -i lo -j ACCEPT
```

Kondisi diatas digunakan pada komputer yang memiliki dua antarmuka jaringan yaitu eth0 dan eth1, jika kita memiliki antarmuka jaringan yang berbeda maka perlu dilakukan pengaturan ulang. Konfigurasi di atas akan menjatuhkan setiap paket yang datang, kecuali paket yang berasal dari inisialisasi paket keluar dari komputer. Jadi skrip diatas tidak bagus untuk komputer server.

Simpan skrip diatas pada berkas /opt/scripts/iptables.script dan berikan hak untuk bisa dijalankan. Jika Anda telah menjalankan script diatas, Anda dapat memeriksa apakah aturan pada skrip berjalan dengan baik menggunakan perintah :

```
sudo iptables -L -v
```

Untuk lebih gampang untuk menjalankan dan memberhentikan firewall tersebut kita dapat membuat skrip inisialiasi sbb :

```
#!/bin/bash
if [[ $1 == start ]] ; then
    sudo /opt/scripts/iptables.script
else
    sudo iptables -F
fi
```

Skrip diatas kita simpan pada /etc/init.d/firewall

Kemudian buat simbol tautan (symlink/shortcut) kedalam direktori /etc/rc.* menggunakan alat updates-rc.d.

Firewall harus berjalan sebelum koneksi ke jaringan terjadi :

```
update-rc.d firewall start 20 2 3 4 5 . stop 99 0 1 6 .
```

Contoh kode yang lebih rumit

Code:

```
#!/bin/bash
# Flush active rules and custom tables
IPTABLES --flush
IPTABLES --delete-chain

# Set default-deny policies for all three default chains
IPTABLES -P INPUT DROP
IPTABLES -P FORWARD DROP
IPTABLES -P OUTPUT DROP

# Give free reign to the loopback interfaces, i.e. local processes may connect
# to other processes' listening-ports.
IPTABLES -A INPUT -i lo -j ACCEPT
IPTABLES -A OUTPUT -o lo -j ACCEPT

# Do some rudimentary anti-IP-spoofing drops. The rule of thumb is "drop
# any source IP address which is impossible" (per RFC 1918)
#
# NOTE: If you use RFC 1918 address-space, comment out or edit the appropriate
# lines below!
#
IPTABLES -A INPUT -s 255.0.0.0/8 -j LOG --log-prefix "Spoofed source IP"
IPTABLES -A INPUT -s 255.0.0.0/8 -j DROP
IPTABLES -A INPUT -s 0.0.0.0/8 -j LOG --log-prefix "Spoofed source IP"
IPTABLES -A INPUT -s 0.0.0.0/8 -j DROP
IPTABLES -A INPUT -s 127.0.0.0/8 -j LOG --log-prefix "Spoofed source IP"
IPTABLES -A INPUT -s 127.0.0.0/8 -j DROP
# IPTABLES -A INPUT -s 192.168.0.0/16 -j LOG --log-prefix "Spoofed source IP"
# IPTABLES -A INPUT -s 192.168.0.0/16 -j DROP
IPTABLES -A INPUT -s 172.16.0.0/12 -j LOG --log-prefix "Spoofed source IP"
IPTABLES -A INPUT -s 172.16.0.0/12 -j DROP
IPTABLES -A INPUT -s 10.0.0.0/8 -j LOG --log-prefix " Spoofed source IP"
IPTABLES -A INPUT -s 10.0.0.0/8 -j DROP

# NOTE: If you use RFC 1918 address-space in your internal or DMZ networks,
# comment out or edit the appropriate lines below!
# The following will NOT interfere with local inter-process traffic, whose
# packets have the source IP of the local loopback interface, e.g. 127.0.0.1

# IPTABLES -A INPUT -s $IP_LOCAL -j LOG --log-prefix "Spoofed source IP"
# IPTABLES -A INPUT -s $IP_LOCAL -j DROP

# Tell netfilter that all TCP sessions do indeed begin with SYN
# (There may be some RFC-non-compliant application somewhere which
# begins its transactions otherwise, but if so I've never heard of it)

IPTABLES -A INPUT -p tcp ! --syn -m state --state NEW -j LOG --log-prefix
"Stealth scan attempt?"
```

```
IPTABLES -A INPUT -p tcp ! --syn -m state --state NEW -j DROP

# INBOUND POLICY:

# Accept inbound packets that are part of previously-OK'ed sessions
IPTABLES -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

# Accept inbound packets which initiate SSH sessions
IPTABLES -A INPUT -p tcp --dport 22 -m state --state NEW -j ACCEPT

# Accept inbound packets which initiate HTTP sessions
IPTABLES -A INPUT -p tcp --dport 80 -m state --state NEW -j ACCEPT

# Log and drop anything not accepted above
# (Obviously we want to log any packet that doesn't match any ACCEPT rule, for
# both security and troubleshooting. Note that the final "DROP" rule is
# redundant if the default policy is already DROP, but redundant security is
# usually a good thing.)

IPTABLES -A INPUT -j LOG --log-prefix "Dropped by default (INPUT):"
IPTABLES -A INPUT -j DROP

# OUTBOUND POLICY:

# (Applies to packets sent to the network interface (NOT loopback)
# from local processes)

# If it's part of an approved connection, let it out
IPTABLES -I OUTPUT 1 -m state --state RELATED,ESTABLISHED -j ACCEPT

# Allow outbound ping
# (For testing only! If someone compromises your system they may attempt
# to use ping to identify other active IP addresses on the DMZ. Comment
# this rule out when you don't need to use it yourself!)

# IPTABLES -A OUTPUT -p icmp -j ACCEPT --icmp-type echo-request

# Allow outbound DNS queries, e.g. to resolve IPs in logs
# (Many network applications break or radically slow down if they
# can't use DNS. Although DNS queries usually use UDP 53, they may also use TCP
# 53. Although TCP 53 is normally used for zone-transfers, DNS queries with
# replies greater than 512 bytes also use TCP 53, so we'll allow both TCP and
# UDP
# 53 here

IPTABLES -A OUTPUT -p udp --dport 53 -m state --state NEW -j ACCEPT
IPTABLES -A OUTPUT -p tcp --dport 53 -m state --state NEW -j ACCEPT

# Log & drop anything not accepted above; if for no other reason, for
troubleshooting
```

```
# NOTE: you might consider setting your log-checker (e.g. Swatch) to
# sound an alarm whenever this rule fires; unexpected outbound trans-
# actions are often a sign of intruders!
```

```
iptables -A OUTPUT -j LOG --log-prefix "Dropped by default (OUTPUT):"
iptables -A OUTPUT -j DROP
```

```
# Log & drop ALL incoming packets destined anywhere but here.
# (We already set the default FORWARD policy to DROP. But this is
# yet another free, reassuring redundancy, so why not throw it in?)
```

```
iptables -A FORWARD -j LOG --log-prefix "Attempted FORWARD? Dropped by default:"
iptables -A FORWARD -j DROP
```

CONTOH MEMBUAT SHARING KONEKSI INTERNET DENGAN MENGGUNAKAN IPTABLES.

Disain Koneksi :

Internet/ISP<=====>Server<=====> Switch |<=====> Klien

Keterangan:

1. ISP dengan IP 192.168.1.1 netmask 255.255.255.0
2. Komputer server dengan OS Ubuntu sebagai Gerbang(Gateway) dengan 2 Ethernet Card
Ethernet Card:
 - eth0 tersambung ke ISP
IP 192.168.1.2 netmask 255.255.255.0
IP Gerbang 192.168.1.1
DNS : 202.134.2.5 dan 202.134.0.155 (DNS Speedy)eth1 tersambung ke swicth / LAN
 - IP 192.168.0.1 netmask 255.255.255.0
Gerbang : --
DNS : 202.134.2.5 dan 202.134.0.155
3. Pengaturan IP komputer klien mulai 192.168.0.2 s/d 192.168.0.254
 - netmask 255.255.255.0
 - gateway 192.168.0.1
 - DNS 192.168.0.1

Konfigurasi untuk Internet Sharing nya dengan Ubuntu sebagai server, isikan nilai seperti ketentuan di atas dimana

- eth0 tersambung ke ISP, dengan IP

```
IP 192.168.1.2
netmask 255.255.255.0
gateway 192.168.1.1
DNS : 202.134.2.5 dan 202.134.0.155
```

- eth1 tersambung ke swicth/LAN dengan IP

```
IP 192.168.0.1
netmask 255.255.255.0
gateway
DNS : 202.134.2.5 dan 202.134.0.155
```

Edit file /etc/resolv.conf dan isikan
nameserver 202.134.2.5
nameserver 202.134.0.155

Lanjutkan dengan mengedit file /etc/network/interfaces, diisi seperti ini

```
auto eth0
iface eth0 inet static
address 192.168.1.2
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.20
gateway 192.168.1.1
```

```
auto eth1
iface eth1 inet static
address 192.168.0.1
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
```

Kemudian aktifkan ip_forward, dengan mengedit file /etc/sysctl.conf
net.ipv4.ip_forward = 1

atau dengan cara
echo 1 > /proc/sys/net/ipv4/ip_forward

Aktifkan NAT dengan iptables

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Restart network
/etc/init.d/networking restart

Pasang IPMASQ dan DNSMASQ untuk caching NS
apt-get install dnsmasq ipmasq

Untuk mengkonfigurasi IPMASQ
dpkg-reconfigure ipmasq

dan DNSMASQ
vim /etc/dnsmasq.conf

Jalankan DNSMASQ
/etc/init.d/dnsmasq start

Selesai... 😊

NYANG LEBIH GAMPANG LAGI

Seperti yang telah dibahas sebelumnya, semua firewall di linux menggunakan modul netfilter pada kernel dan iptables adalah perintah yang digunakan untuk mengkonfigurasi fungsi dari firewall tersebut. Nah kalau kita Pusiing dengan perintah-perintah diatas maka kita dapat menggunakan ujung depan (front end) yang bersifat antar muka grafis.

Salah satunya Adalah FIRESTARTER

Instalasi :

Dari terminal atau dari Synaptic Package Manager

Contoh dari Terminal :

```
syafiudin@syafiudin-laptop:~$ sudo apt-get install firestarter
```

Hasil Instalasi dapat dilihat pada Applications->Internet->Firestarter

Saat pertama dijalankan firestarter akan meminta kita untuk melakukan pengaturan sbb:



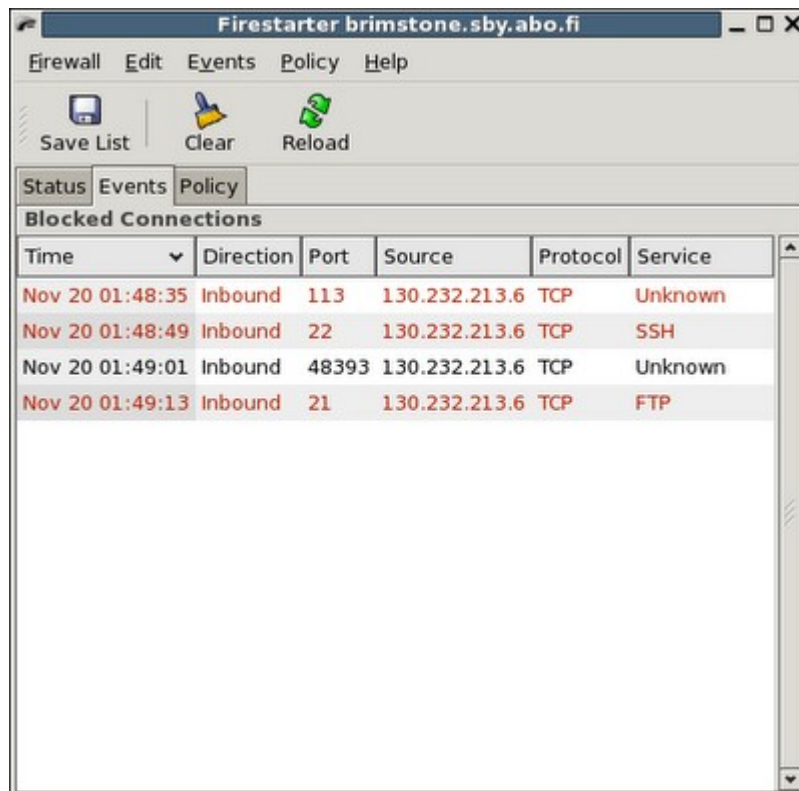
Setelah pengaturan maka firewall otomatis akan dijalankan, secara standar maka firestarter akan melarang semua koneksi yang masuk, Anda harus membuat aturan agar koneksi yang masuk dapat diterima pada Bagian Policy. Pengaturan tersebut dapat disesuaikan dengan kebutuhan kita, apakah berdasarkan alamat IP, mau berdasarkan layanan jaringan yang tersedia.



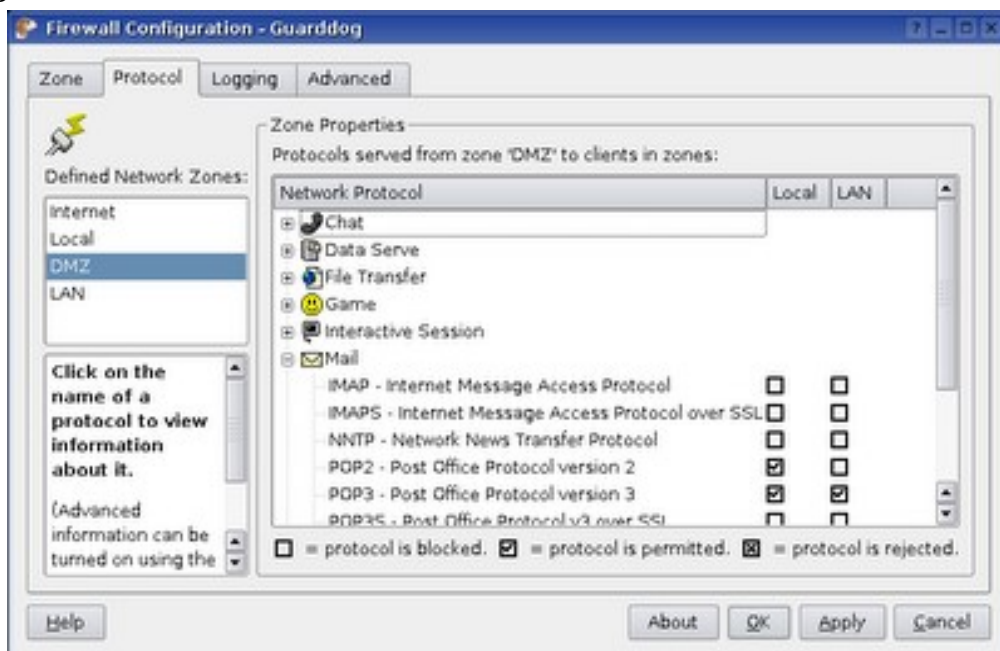
Berikut contoh tampilan saat firewall aktif dan antar muka yang ada



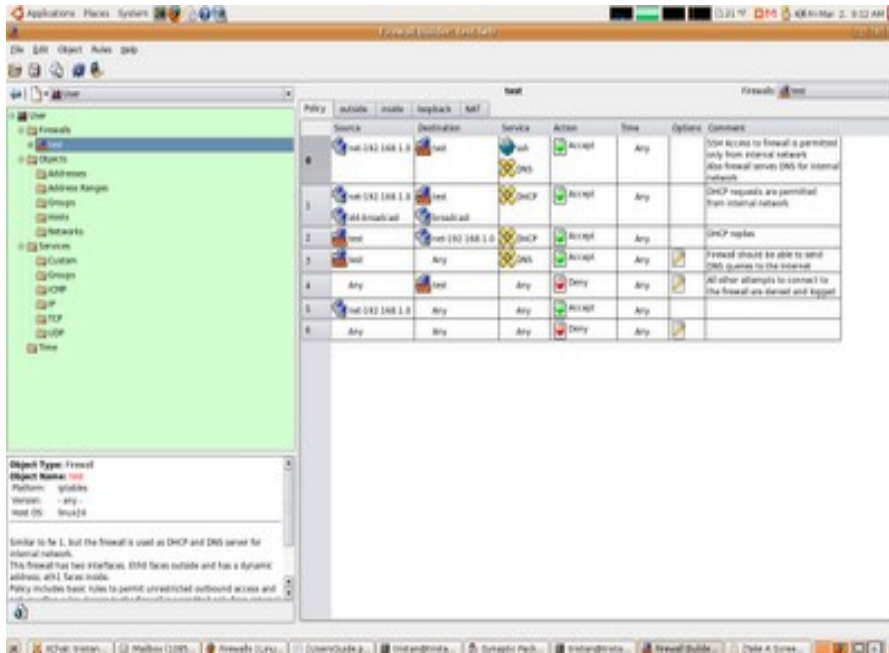
Salah satu keistimewaan firestarter adalah adanya log dari koneksi yang diblokir sehingga kita dapat melihat dari alamat ip mana saja yang mencoba akses ke komputer dan jaringan yang kita kelola.



Selain itu juga banyak paket pengaturan firewall lainnya seperti **Guarddog Firewall**



Fwbuilder Firewall



Atau paket berbasis web seperti e-box platform

