Transparent Firewall adalah firewall yang "tidak tampak" baik oleh user di dalam zona yang kita amankan atau dari luar zona kita. Transparent Firewall pada dasarnya adalah firewall biasa hanya saja implementasinya dilakukan pada *bridge*, sehingga tidak ada konfigurasi yang harus dilakukan pada jaringan yang sudah ada.

Tutorial ini akan (mencoba) untuk menjelaskan secara singkat bagaimana dan apa saja yang diperlukan dalam pembuatan bridge firewall pada Debian GNU/Linux , dan pengaturan umum kerja firewall yang kita buat.

Topologi yang dipakai diasumsikan seperti gambar di bawah ini.

```
+----------+     +--------+     +---------------+
| INTERNET |-----| BRIDGE |-----| JARINGAN KITA |
+----------+     +--------+     +---------------+
```

1. Kebutuhan Dasar

Implementasi ini membutuhkan beberapa hal yaitu:

1. Komputer dengan 2 NIC
2. iptables
3. bridge-utils
4. Kernel Linux 2.6 atau 2.4(dengan patch) dengan opsi bridge firewall diaktifkan

Bila anda ingin menggunakan kernel versi 2.4 silakan cari patchnya di http://ebtables.sourceforge.net/ . Kernel 2.6 sudah menyertakan fasilitas ini, sehingga tidak perlu dipatch lagi.

Di sini diasumsikan bahwa kernel sudah beres, dan tinggal menginstall program lain yang dibutuhkan untuk menjalankan bridge. Komputer yang dipakai mempunyai 2 NIC yaitu eth0

dan eth1.

2. Instalasi & Konfigurasi

Seperti biasa, untuk menginstall paket pada debian kita menggunakan apt-get.

root:~# apt-get install bridge-utils iptables

Bila anda tidak menggunakan distro lain anda dapat mendownload source code untuk kedua program itu pada http://bridge.sourceforge.net/ dan http://www.iptables.org/files/ . Panduan proses instalasi dapat mengikuti file README/INSTALL yang disertakan pada tarball yang anda download.

Program yang kita dapatkan dari bridge-utils adalah brctl. Program inilah yang mengatur segala macam bagian bridge mulai pembuatan, penghapusan, penambahan anggota bridge, dan sebagainya. Buat interface bridge (br0), dan tambahkan kedua interface ke dalam interface bridge yang baru dibuat.

```
root:~# brctl addbr br0
root:~# brctl addif br0 eth0
root:~# brctl addif br0 eth1
root:~# ifconfig eth0 0
root:~# ifconfig eth1 0
```

Hapus alamat IP pada eth0 dan eth1, dan bila bridge ini akan diberi alamat IP maka yang perlu diberi alamat adalah br0. Interface lain harus tetap menyala tanpa mempunyai alamat IP sendiri. Nantinya, kedua ethernet yang ada akan merespon setiap request yang masuk ke alamat IP bridge.

Cobalah untuk memping jaringan di luar jaringan anda. Bila lancar, berarti bridge ini sudah berjalan dengan baik. Agar setiap booting kita tidak mengulangi langkah-langkah di atas maka edit file /etc/network/interfaces dan tambahkan konfigurasi seperlunya. Contoh file saya ada di

bawah ini.

```
auto br0
iface br0 inet static
    address 10.11.12.3
    netmask 255.255.255.0
    network 10.11.12.0
    broadcast 10.11.12.255
    gateway 10.11.12.1
    bridge_ports eth0 eth1
```

Catatan: perhatikan item konfigurasi yang dicetak tebal.

Pastikan juga anda mengaktifkan IP Forwarding dengan mengeksekusi perintah di bawah ini tiap kali komputer booting.

```
root:~# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Selain cara manual itu anda dapat pula mengedit file /etc/network/options.

```
ip_forward=yes
spoofprotect=yes
syncookies=no
```

Sekarang anda dapat mengkonfigurasi iptables untuk melakukan penyaringan terhadap paket-paket data yang melewati firewall ini. Iptables tidak akan dibahas mendalam di sini. Sekedar info, penyaringan ini dilakukan pada tabel **Filter** chain **FORWARD**. Contoh:

```
root:~# iptables -t Filter -A FORWARD -s 0.0.0.0 -d 10.11.12.0/24 -p tcp --dport 23 -j DROP
root:~# iptables -t Filter -A FORWARD -s 0.0.0.0 -d 10.11.12.0/24 -p tcp --dport 25 -j DROP
root:~# iptables -t Filter -A FORWARD -s 0.0.0.0 -d 10.11.12.0/24 -p udp --dport 161 -j DROP
```

Potongan instruksi iptables di atas memfilter paket-paket dari luar network kita yang akan mengakses port telnet, smtp, dan snmp. Tambahkan filter lain sesuai dengan yang anda butuhkan.

Simpanlah perintah-perintah yang anda jalankan pada sebuah script .sh dan ubahlah permission file tersebut agar bisa dieksekusi(*executable*). Aturlah agar file itu diekseuksi setiap kali boot. Ada beberapa cara melakukannya, yang termudah adalah menambahkan entri pada /etc/network/interfaces. Bila script iptables itu disimpan di /etc/init.d/aturanfirewall.sh anda dapat menambahkan baris berikut di bawah entri br0.

```
up command /etc/init.d/aturanfirewall.sh
```

Dengan demikian isi file /etc/network/interfaces menjadi seperti di bawah ini.

```
auto br0
iface br0 inet static
    address 10.11.12.3
    netmask 255.255.255.0
    network 10.11.12.0
    broadcast 10.11.12.255
    gateway 10.11.12.1
    bridge_ports eth0 eth1
    up command /etc/init.d/aturanfirewall.sh
```

Dengan menggunakan bash script kecil untuk memparsing output command "iptables -nvL FORWARD" anda dapat memantau aktifitas filtering seperti ini :)

```
 +-------------------------------------------------------+
| I/O Total   |   95M Packets    62G Bytes                |
 +-------------+--------------------+--------------------+
 |             |     Traffic In     |     Traffic Out    |
 |   Filter    +----------+---------+----------+---------+
 |             |   Byte   |  Packet |   Byte   |  Packet |
 +-------------+----------+---------+----------+---------+
 | Ping Blaster|       0 |       0 |    23184 |     252 |
 | udp 69      |       0 |       0 |        0 |       0 |
 | udp 135     |       0 |       0 |        0 |       0 |
 | udp 137     |    38298 |     491 |     9828 |     126 |
 | udp 138     |     534 |       2 |     1343 |       5 |
 | udp 445     |       0 |       0 |        0 |       0 |
 | udp 161     |    3672 |      54 |        0 |       0 |
 | tcp 23      |     912 |      19 |        0 |       0 |
 | tcp 135     |    47520 |     990 |     1584 |      33 |
 | tcp 445     |    1027K |   21402 |    15180 |     316 |
 | tcp 593     |       0 |       0 |        0 |       0 |
 | tcp 4444    |     528 |      12 |      864 |      18 |
 +-------------+----------+---------+----------+---------+
```

Contoh script bisa didownload di sini , selamat mencoba, semoga sukses :)
(Kamas Muhammad, http://www.sokam.or.id).

3. Catatan Pinggir

Dari beberapa percobaan yang saya lakukan, ethernet yang bermutu tinggi sangat membantu kelancaran kerja firewall. Segala macam ethernet yang menggunakan driver *8139too* menyebabkan firewall sering macet, dan secara rutin harus direboot sekali dalam seminggu. Setelah ethernet diganti dengan
Intel EtherExpress 100
dan
3Com 3c905B
semuanya berjalan lancar tanpa gangguan.
AMD Lance PCnet32
juga menunjukkan hasil yang baik. Bila anda mempunyai pengalaman yang lain saya sangat senang untuk mencantumkannya di sini.

Ressa Restullah(30 Mei 2005) - bridge yang dibuat Mas Ressa ternyata tidak berjalan lancar, karena salah satu interface *eth*nya terhubung dengan Cisco

1700. Menurut kabar dari ISPnya, masalahnya timbul karena ada fasilitas "auto negotiation" yang dimiliki router tersebut. Masalah ini hilang setelah Mas Ressa mematikan fasilitas *auto-negotiation* pada router & bridge, dan menggunakan kabel cross untuk menghubungkan router dengan bridgenya. Selain itu, usahakan agar interface pada bridge dan router sejenis. Kalau router menggunakan FastEthernet, maka gunakan juga FastEthernet untuk bridgenya :)

4. Referensi

1. http://ebtables.sourceforge.net
2. Securing Debian Manual
3. Milis tanya-jawab@linux.or.id
4. http://www.google.com :D
5. Ressa Restullah

Kamas Muhammad adalah seorang Linuxer di Institute Teknologi Sepuluh November. Wajahnya yang boros usia sering membuat orang mengira ia adalah seorang dosen :) Terima Oom Sokam atas artikelnya yang menarik dan bermanfaat ini.