Giới thiệu về Iptables

Tài liệu này được dịch từ
http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO_:_Ch14_:_Linux_Firewalls_Using_iptables

Vẫn còn rất thiếu sót trong tài liệu này . Mong mọi người ủng hộ và đóng góp ý kiến để tài liệu này hoàn thiện hơn . Mọi ý kiến đóng góp xin gởi về trannhathuy@gmail.com .

Tp Hồ Chí Minh , 12/2006

Nhóm biên dịch : Trần Nhật Huy
Hoàng Hải Nguyên
Ngô Trí Hùng Nam

I. GIỚI THIỆU VỀ IPTABLES:

Bảo mật mạng là một vấn đề quan trọng hàng đầu đối việc lập một website , cũng như nhiều dịch vụ khác trên mạng . Một trong những cách bảo vệ là sử dụng firewall . bài viết này sẽ cho thấy làm sao để chuyển một Linux server thành :

- Một firewall đồng thời cho mail server , web server , DNS server.
- Một thiết bị dẫn đường (router) sẽ dùng NAT và chuyển tiếp cổng (port forwarding) để vừa bảo vệ hệ thống mạng của bạn , vừa cho phép một web server công khai chia sẻ địa chỉ IP firewall .

Một trong những firewall thông dụng nhất chạy trên Linux là iptables . Ta sẽ xem qua một số chức năng của iptables :

- Tích hợp tốt với Linux kernel , để cải thiện sự tin cậy và tốc độ chạy iptables .
- Quan sát kỹ tất cả các gói dữ liệu . Điều này cho phép firewall theo dõi mỗi một kết nối thông qua nó , và dĩ nhiên là xem xét nội dung của từng luồng dữ liệu để từ đó tiên liệu hành động kế tiếp của các giao thức . Điều này rất quan trọng trong việc hỗ trợ các giao thức FTP , DNS ….
- Lọc gói dựa trên địa chỉ MAC và các cờ trong TCP header. Điều này giúp ngăn chặn việc tấn công bằng cách sử dụng các gói dị dạng (malformed packets) và ngăn chặn việc truy cập từ nội bộ đến một mạng khác bất chấp IP của nó.
- Ghi chép hệ thống (System logging) cho phép việc điều chỉnh mức độ của báo cáo
- Hỗ trợ việc tính hợp các chương trình Web proxy chẳng như Squid .
- Ngặn chặn các kiểu tấn công từ chối dịch vụ.

II. SỬ DỤNG IPTABLES

1. Khởi động iptables :

Câu lệnh start, stop, và restart iptables .

```
[root@bigboy tmp]# service iptables start
[root@bigboy tmp]# service iptables stop
[root@bigboy tmp]# service iptables restart
```

Để khởi động iptables mỗi khi khởi động máy .

```
[root@bigboy tmp]# chkconfig iptables on
```

Để xem tình trạng của iptables

```
[root@bigboy tmp]# service iptables status
```

2. Xử lý gói trong iptables:

Tất cả mọi gói dữ liệu đều được kiểm tra bởi iptables bằng cách dùng các bảng tuần tự xây dựng sẵn (queues) . Có 3 loại bảng này gồm :

_ ***Mangle :*** chịu trách nhiệm thay đổi các bits chất lượng dịch vụ trong TCP header như TOS (type of service), TTL (time to live), và MARK.

_ ***Filter :*** chịu trách nhiệm lọc gói dữ liệu . Nó gồm có 3 quy tắc nhỏ (chain) để giúp bạn thiết lập các nguyên tắc lọc gói , gồm :

- Forward chain: lọc gói khi đi đến đến các server khác .
- Input chain: lọc gói khi đi vào trong server .
- Output chain: lọc gói khi ra khỏi server .

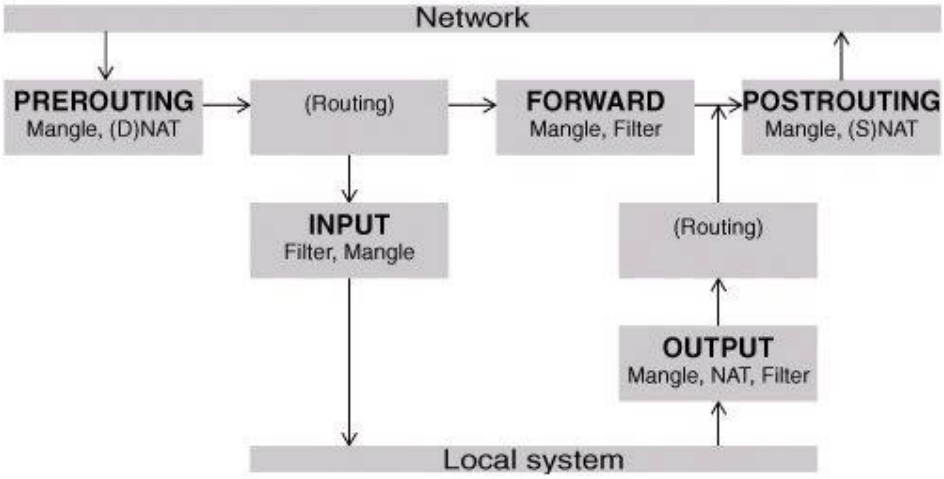
_ ***NAT :*** gồm có 2 loại :

- Pre-routing chain: thay đổi địa chỉ đến của gói dữ liệu khi cần thiết.
- Post-routing chain: thay đổi địa chỉ nguồn của gói dữ liệu khi cần thiết .

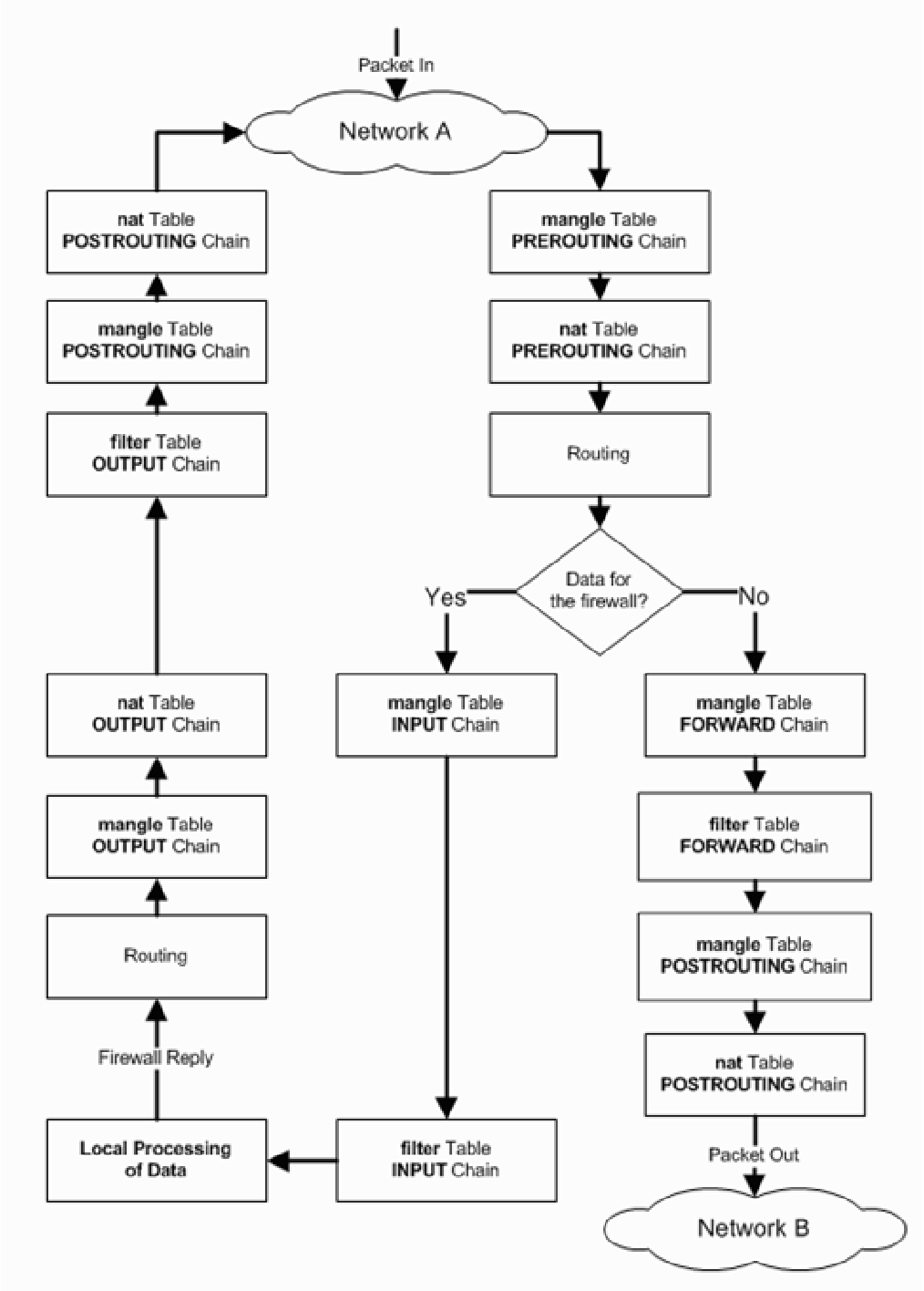
Bảng 1 : Các loại queues và chain cùng chức năng của nó.

Loại queues	Chức năng queues	Quy tắc xử lý gói (chain)	Chức năng của chain
Filter	Lọc gói	FORWARD	Lọc gói dữ liệu đi đến các server khác kết nối trên các NIC khác của firewall
		INPUT	Lọc gói đi đến firewall
		OUTPUT	Lọc gói đi ra khỏi firewall
NAT	Network Address Translation (Biên dịch địa chỉ mạng)	PREROUTING	Việc thay đổi địa chỉ diễn ra trước khi dẫn đường. Thay đổi địa chỉ đích sẽ giúp gói dữ liệu phù hợp với bảng chỉ đường của firewall. Sử dụng **destination NAT** or **DNAT**.
		POSTROUTING	Việc thay đổi địa chỉ diễn ra sau khi dẫn đường . Sử dụng **source NAT**, or **SNAT**.
		OUTPUT	NAT sử dụng cho các gói dữ liệu xuất phát từ firewall . Hiếm khi dùng trong môi trường SOHO (small office - home office) .
Mangle	Chỉnh sửa TCP header .	PREROUTING POSTROUTING OUTPUT INPUT FORWARD	Điều chỉnh các bit quy địch chất lượng dịch vụ trước khi dẫn đường . Hiếm khi dùng trong môi trường SOHO (small office - home office) .

Để cái nhìn tổng quát đối với việc lọc và xử lý gói trong iptables , ta xem hình sau :



Ta cùng xem qua 1 ví dụ mô tả đường đi của gói dữ liệu .



Đầu tiên, gói dữ liệu đến mạng A , tiếp đó nó được kiểm tra bởi mangle table PREROUTING chain (nếu cần).Tiếp theo là kiểm tra gói dữ liệu bởi nat table's PREROUTING chain để kiểm tra xem gói dữ liệu có cần DNAT hay không? DNAT sẽ thay đổi địa chỉ đích của gói dữ liệu . Rồi gói dữ liệu được dẫn đi .

Nếu gói dữ liệu đi vào một mạng được bảo vệ, thì nó sẽ được lọc bởi FORWARD chain của filter table, và nếu cần gói dữ liệu sẽ được SNAT trong POSTROUTING chain để thay đổi IP nguồn trước khi vào mạng B.

Nếu gói dữ liệu được định hướng đi vào trong bên trong firewall , nó sẽ được kiểm tra bởi INPUT chain trong mangle table, và nếu gói dữ liệu qua được các kiểm tra của INPUT chain trong filter table, nó sẽ vào trong các chương trình của server bên trong firewall .

Khi firewall cần gởi dữ liệu ra ngoài . Gói dữ liệu sẽ được dẫn và đi qua sự kiểm tra của OUTPUT chain trong mangle table(nếu cần), tiếp đó là kiểm tra trong OUTPUT chain của nat table để xem DNAT (DNAT sẽ thay đổi địa chỉ đến) có cần hay không và OUTPUT chain của filter table sẽ kiểm tra gói dữ liệu nhằm phát hiện các gói dữ liệu không được phép gởi đi. Cuối cùng trước khi gói dữ liệu được đưa ra lại Internet, SNAT and QoS sẽ được kiểm tra trong POSTROUTING chain .

3. Targets

Targets là hành động sẽ diễn ra khi một gói dữ liệu được kiểm tra và phù hợp với một yêu cầu nào đó. Khi một target đã được nhận dạng , gói dữ liệu cần nhảy (jump) để thực hiện các xử lý tiếp theo . Bảng sau liệt kê các targets mà iptables sử dụng .

Bảng 2 : Miêu tả các target mà iptables thường dùng nhất .

Targets	Ý nghĩa	Tùy chọn
ACCEPT	`iptables` ngừng xử lý gói dữ liệu đó và chuyển tiếp nó vào một ứng dụng cuối hoặc hệ điều hành để xử lý .	
DROP	`iptables` ngừng xử lý gói dữ liệu đó và gói dữ liệu bị chặn, loại bỏ.	
LOG	Thông tin của gói sẽ được đưa vào syslog để kiểm tra . Iptables tiếp tục xử lý gói với quy luật kế tiếp .	--log-prefix "string" `iptables` sẽ thêm vào log message một chuỗi do người dùng định sẵn . Thông thường là để thông báo lý do vì sao gói bị bỏ .

REJECT	Tương tự như DROP , nhưng nó sẽ gởi trả lại cho phía người gởi một thông báo lỗi rằng gói đã bị chặn và loại bỏ .	--reject-with *qualifier* Tham số qualifier sẽ cho biết loại thông báo gởi trả lại phía gởi . Qualifier gồm các loại sau : icmp-port-unreachable (default) icmp-net-unreachable icmp-host-unreachable icmp-proto-unreachable icmp-net-prohibited icmp-host-prohibited tcp-reset echo-reply
DNAT	Dùng để thực hiện **Destination network address translation ,** địa chỉ đích của gói dữ liệu sẽ được viết lại .	--to-destination *ipaddress* Iptables sẽ viết lại địa chỉ ipaddress vào địa chỉ đích của gói dữ liệu .
SNAT	Dùng để thực hiện **Source network address translation** , viết lại địa chỉ nguồn của gói dữ liệu .	--to-source *<address>[-<address>][:<port>-<port>]* Miêu tả IP và port sẽ được viết lại bởi iptables .
MASQUERADE	Dùng để thực hiện **Source Networkaddress Translation.**Mặc định thì địa chỉ IP nguồn sẽ giống như IP nguồn của firewall .	*[--to-ports <port>[-<port>]]* Ghi rõ tầm các port nguồn mà port nguồn gốc có thể ánh xạ được.

4. Các tham số chuyển mạch quan trọng của Iptables:

Các tham số sau sẽ cho phép Iptables thực hiện các hành động sao cho phù hợp với biểu đồ xử lý gói do người sử dụng hoạch định sẵn .

Bảng 3 : Các tham số chuyển mạch (switching) quan trọng của Iptables .

Lệnh switching quan trọng	Ý nghĩa
-t <table>	Nếu bạn không chỉ định rõ là tables nào , thì filter table sẽ được áp dụng. Có ba loại table là filter, nat, mangle.
-j <target>	Nhảy đến một chuỗi target nào đó khi gói dữ liệu phù hợp quy luật hiện tại .
-A	Nối thêm một quy luật nào đó vào cuối chuỗi (chain).
-F	Xóa hết tất cả mọi quy luật trong bảng đã chọn .
-p <protocol-type>	Phù hợp với giao thức (protocols) , thông thường là `icmp`, `tcp`, `udp`, và `all` .
-s <ip-address>	Phù hợp IP nguồn
-d <ip-address>	Phù hợp IP đích
-i <interface-name>	Phù hợp điều kiện INPUT khi gói dữ liệu đi vào firewall
-o <interface-name>	Phù hợp điều kiện OUTPUT khi gói dữ liệu đi ra khỏi firewall .

Để hiểu rõ hơn về các lệnh ta , ta cùng xem một ví dụ sau :

```
iptables -A INPUT -s 0/0 -i eth0 -d 192.168.1.1  -p TCP \
         -j ACCEPT
```

Iptables được cấu hình cho phép "firewall" chấp nhận các gói dữ liệu có giao tiếp (protocols) là TCP , đến từ giao tiếp card mạng eth0 , có bất kỳ địa chỉ IP nguồn là bất kỳ đi đến địa chỉ 192.168.1.1, là địa chỉ IP của firewall. 0/0 nghĩa là bất kỳ địa chỉ IP nào .

Bảng 4 : Các điều kiện TCP và UDP thông dụng .

Lệnh switching	Miêu tả
-p tcp --sport <port>	Điều kiện TCP port nguồn (source port) . Có thể là một giá trị hoặc một chuỗi có dạng : **start-port-number:end-port-number**
-p tcp --dport <port>	Điều kiện TCP port đích (destination port) Có thể là một giá trị hoặc một chuỗi có dạng : **starting-port:ending-port**

-p tcp –syn	Dùng để nhận dạng một yêu cầu kết nối TCP mới . **! --syn** , nghĩa là không có yêu cần kết nối mới .
-p udp --sport <port>	Điều kiện UDP port nguồn (source port) . Có thể là một giá trị hoặc một chuỗi có dạng : **start-port-number:end-port-number**
-p udp --dport <port>	Điều kiện TCP port đích (destination port) Có thể là một giá trị hoặc một chuỗi có dạng : **starting-port:ending-port**

Ta cùng xem ví dụ sau :

```
iptables –A FORWARD –s 0/0 –i eth0 –d 192.168.1.58 –o eth1 –p TCP \
         --sport 1024:65535 --dport 80 –j ACCEPT
```

Iptables được cấu hình cho phép firewall chấp nhận các gói dữ liệu có giao tiếp (protocols) là TCP , đến từ card mạng eth0 , có bất kỳ địa chỉ IP nguồn là bất kỳ , đi đến địa chỉ 192.168.1.58 qua card mạng eth1. Số port nguồn là từ 1024 đến 65535 và port đích là 80 (www/http).

Bảng 5 : Điều kiện ICMP

Lệnh	Miêu tả
--icmp-type <type>	Thường dùng nhất là echo-reply và echo-request

Ta cùng xem mộ ví dụ sau về ICMP .

```
iptables –A OUTPUT –p icmp --icmp-type echo-request –j ACCEPT
iptables –A INPUT  –p icmp --icmp-type echo-reply   –j ACCEPT
```

Iptables được cấu hình cho phép firewall chấp nhận gởi ICMP echo-requests (pings) và gởi trả các ICMP echo-replies.

Ta cùng xem ví dụ khác như sau :

```
iptables –A INPUT –p icmp --icmp-type echo-request –m limit
         \   –limit 1/s –i eth0 –j ACCEPT
```

Iptables cho phép giới hạn giá trị lớn nhất số lượng các gói phù hợp trong một giây . Bạn có chỉ định thời gian theo định dạng /second, /minute, /hour, hoặc /day . Hoặc sử dụng dạng viết tắt 3/s thay vì 3/second . Trong ví dụ này ICMP echo requests bị giới hạn không nhiều hơn một yêu cần trong một giây . Đặc điểm này của iptables giúp ta lọc bớt các lưu lượng lớn , đây chính là đặc tính của tấn công từ chối dịch vụ (DOS) và sâu Internet.

```
iptables –A INPUT –p tcp --syn –m limit --limit 5/s –i \
```

```
eth0 -j ACCEPT
```

Bạn có thể mở rộng khả năng giới hạn của iptables để giảm thiểu khả năng bị tấn công bởi các loại tấn công từ chối dịch vụ. Đây là cách phòng vệ chống lại kiểu tấn công SYN flood bằng cách hạn chế sự chấp nhận các phân đoạn TCP có bit SYS không nhiều hơn 5 phân đoạn trong 1 giây.

Bảng 6 : Các điều kiện mở rộng thông dụng

Lệnh	Ý nghĩa
-m multiport --sport <port, port>	Nhiều port nguồn khác nhau của TCP/UDP được phân cách bởi dấu phẩy (,) . Đây là liệt kê của các port chứ không phải là một chuỗi các port.
-m multiport --dport <port, port>	Nhiều port đích khác nhau của TCP/UDP được phân cách bởi dấu phẩy (,) . Đây là liệt kê của các port chứ không phải là một chuỗi các port.
-m multiport --ports <port, port>	Nhiều port khác nhau của TCP/UDP được phân cách bởi dấu phẩy (,) . Đây là liệt kê của các port chứ không phải là một chuỗi các port. Không phân biệt port đích hay port nguồn .
-m --state <state>	Các trạng thái thông dụng nhất được dùng là : **ESTABLISHED**:Gói dữ liệu là một phần của kết nối đã được thiết lập bởi cả 2 hướng . **NEW**:Gói dữ liệu là bắt đầu của một kết nối mới . **RELATED**: Gói dữ liệu bắt đầu một kết nối phụ . Thông thường đây là đặt điểm của các giao thức như FTP hoặc lỗi ICMP . **INVALID**: Gói dữ liệu không thể nhận dạng được . Điều này có thể do việc thiếu tài nguyên hệ thống hoặc lỗi ICMP không trùng với một luồng dữ liệu đã có sẵn .

Đây là phần mở rộng tiếp theo của ví dụ trước :

```
iptables -A FORWARD -s 0/0 -i eth0 -d 192.168.1.58 -o eth1 -p TCP \
         --sport 1024:65535 -m multiport --dport 80,443 -j ACCEPT

iptables -A FORWARD -d 0/0 -o eth0 -s 192.168.1.58 -i eth1 -p TCP \
         -m state --state ESTABLISHED -j ACCEPT
```

Iptables được cấu hình cho phép firewall chấp nhận các gói dữ liệu có giao tiếp (protocols) là TCP , đến từ card mạng eth0 , có bất kỳ địa chỉ IP nguồn là bất kỳ , đi

đến địa chỉ 192.168.1.58 qua card mạng eth1. Số port nguồn là từ 1024 đến 65535 và port đích là 80 (www/http) và 443 (https). Đến khi các gói dữ liệu nhận trở lại từ 192.168.1.58, thay vì mở các port nguồn và đích , bạn chỉ việc cho phép dùng kết nối cũ đã thiết lập bằng cách dùng tham số -m state và --state ESTABLISHED.

5_ Sử dụng user defined chains:

Chuỗi User Defined Chains nằm trong bảng iptables. Nó giúp cho quá trình sử lý gói tốt hơn.

Ví dụ: Thay vì sử dụng gói đơn được xây dựng trong chain cho tất cả giao thức, ta có thể sử dụng chain này để quyết định loại giao thức cho gói và sau đó kiểm soát việc xử lý user-defined, protocol-specific chain trong bảng filter table.

Mặt khác, ta có thể thay thế một chuỗi "long chain" với chuỗi chính "stubby main chain" bởi nhiều chuỗi "stubby chain", bằng cách chia ngắn đó tổng chiều dài của tất cả chain gói phải thông qua.

⇨ *Sáu lệnh sau giúp việc cải tiến tốc độ xử lý:*

```
iptables -A INPUT -i eth0 -d 206.229.110.2 -j \
         fast-input-queue
iptables -A OUTPUT -o eth0 -s 206.229.110.2 -j \
         fast-output-queue
iptables -A fast-input-queue -p icmp -j icmp-queue-in
iptables -A fast-output-queue -p icmp -j icmp-queue-out
iptables -A icmp-queue-out -p icmp --icmp-type \
         echo-request -m state --state NEW -j ACCEPT
iptables -A icmp-queue-in -p icmp --icmp-type echo-reply\
         -j ACCEPT
```

DANH SÁCH CÁC LỆNH (QUEUE)

Chain	Description
INPUT	Được xây dựng trong INPUT chain trong bảng iptables
OUTPUT	Được xây dựng trong ONPUT chain trong bảng iptables
Fast-input-queue	Input chain tách riêng biệt để hỗ trợ cho những giao thức đặc biệt và chuyển các gói đến những protocol specific chains.
fast-output-queue	Output chain tách riêng biệt để hỗ trợ cho những giao thức đặc biệt và chuyển các gói đến những protocol specific chains.
icmp-queue-out	lệnh output tách riêng cho giao thức ICMP

icmp-queue-in	Lệnh input tách riêng cho giao thức ICMP

6_ Lưu lại những đoạn mã iptables:

Đoạn mã iptables được lưu tạm thời ở file **"/etc/sysconfig/iptables"**

Định dạng mẫu trong file iptables cho phép giao thức ICMP, IPSec (những gói ESP và AH), thiết lập liên kết, và quay lại SSH.

```
[root@bigboy tmp]# cat /etc/sysconfig/iptables

# Generated by iptables-save v1.2.9 on Mon Nov 8 11:00:07 2004 *filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [144:12748]
:RH-Firewall-1-INPUT -  [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp -m icmp --icmp-type 255 -j ACCEPT
-A RH-Firewall-1-INPUT -p esp -j ACCEPT
-A RH-Firewall-1-INPUT -p ah -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state RELATED,ESTABLISHED -j
ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j
ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Mon Nov 8 11:00:07 2004

[root@bigboy tmp]#
```

7_ Thiết lập những Rule cho Fedora's iptable:

Trong Fedora có chương trình gọi lokkit, chương trình này có thể thiết lập một rule firewall đơn giản, giúp **tăng cường bảo mật**. Chương trình lokkit lưu những rule firewall trong file mới "/etc/sysconfig/iptables".

8_ Tìm lại Đoạn mã bị mất:

Đoạn mã iptables được lưu trữ trong file **"/etc/sysconfig/iptables"**. Ta có thể chỉnh sửa những đoạn mã và tạo lại những thành những rule mới.

Ví dụ: xuất những lệnh trong iptables đã lưu trữ ra file văn bản với tên firewall-config:

```
[root@bigboy tmp]# iptables-save > firewall-config
[root@bigboy tmp]# cat firewall-config
#  Generated by  iptables-save v1.2.9  on Mon Nov 8  11:00:07  2004 *filter
```

```
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [144:12748]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp -m icmp --icmp-type 255 -j ACCEPT
-A RH-Firewall-1-INPUT -p esp -j ACCEPT
-A RH-Firewall-1-INPUT -p ah -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state RELATED,ESTABLISHED \
-j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --dport 22 \
-j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Mon Nov 8 11:00:07 2004
```
[root@bigboy tmp]#

Sau khi chỉnh sửa file firewall-config, ta có thể tải nó lại trong rule firewall với lệnh:

[root@bigboy tmp]# iptables-restore < firewall-config

Ta có thể lưu tạm thời:

[root@bigboy tmp]# service iptables save

9_ Những modun Kernel cần thiết :

Modun Kernel cần thiết để hoạt động một vài chương trình của ứng dụng iptables

Một số modun: **iptables_nat module, ip_conntrack_ftp module,**

+ **iptables_nat module** cần cho một số loại NAT.
+ **ip_conntrack_ftp module** cần cho việc thêm vào giao thức FTP.
+ **ip_conntrack module** giữ trạng thái liên kết với giao thức TCP.
+ **ip_nat_ftp module** cần được tải cho những máy chủ FTP sau một firewall NAT

*CHÚ Ý: file **/etc/sysconfig/iptables** không cập nhật những mô dun tải về, vì vậy chúng ta phải thêm vào những trạng thái đó vào file /etc/rc.local và chạy nó tại cuối mỗi lần boot lại.

Những mẫu đoạn mã trong phần này bao gồm những trạng thái được lưu trong file **/etc/rc.local**:

```
# File: /etc/rc.local
# Module to track the state of connections modprobe ip_conntrack
# Load the iptables active FTP module, requires ip_conntrack modprobe
# ip_conntrack_ftp
# Load iptables NAT module when required modprobe iptable_nat
# Module required for active an FTP server using NAT modprobe ip_nat_ftp
```

10_Những đoạn mã iptables mẫu:

10.1_ Cơ bản về hoạt động của hệ thống bảo vệ:

Hệ Điều Hành Linux có cơ chế bảo vệ là các thông số kernel hệ thống trong file hệ thống **/proc** qua file **/etc/sysctl.conf**. Dùng file /etc/systl.conf cho các thông số kernel hỗ trợ.

Đây là một cấu hình mẫu:

```
# File:  /etc/sysctl.conf
#--------------------------------------------------------
# Disable routing triangulation. Respond to queries out
# the same interface, not another. Helps to maintain
state
# Also protects against IP spoofing
#--------------------------------------------------------
net/ipv4/conf/all/rp_filter = 1
#--------------------------------------------------------
_
# Enable logging of packets with malformed IP addresses
#--------------------------------------------------------
_
net/ipv4/conf/all/log_martians = 1
# Disable redirects
#--------------------------------------------------------
_
net/ipv4/conf/all/send_redirects = 0
#--------------------------------------------------------
_
# Disable source routed packets
#--------------------------------------------------------
_
net/ipv4/conf/all/accept_source_route = 0
#--------------------------------------------------------
_
# Disable acceptance of ICMP redirects
#--------------------------------------------------------
_
net/ipv4/conf/all/accept_redirects = 0
#--------------------------------------------------------
_
# Turn on protection from Denial of Service (DOS) attacks
#--------------------------------------------------------
_
net/ipv4/tcp_syncookies = 1
#--------------------------------------------------------
_
# Disable responding to ping broadcasts
#--------------------------------------------------------
_
net/ipv4/icmp_echo_ignore_broadcasts = 1
#--------------------------------------------------------
_
# Enable IP routing. Required if your firewall is
#   protecting
# network, NAT included
```

```
#-------------------------------------------------------
-
net/ipv4/ip_forward = 1
```

10.2_ Ưu điểm của sự khởi tạo iptables:

Ta có thể thêm vào nhiều cái ứng dụng khởi tạo cho đoạn mã, bao gồm việc kiểm tra đường truyền internet từ những địa chỉ riêng RFC1918. Nhiều hơn nhứng khởi tạo phức tạpbao gồm kiểm tra lỗi bởi sự tấn công sử dụng cờ TCP không có giá trị.

Đoạn mã cũng sử dụng nhiều "user-defined chain" để tạo đoạn mã ngắn hơn và nhanh hơn như những chain có thể bị truy cập lặp lại. Điều này loại bỏ việc cần thiết lặp lại những trạng thái tương tự.

Đoạn mã firewall hoàn tất :

```
#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#
#
# Define networks: NOTE!! You may want to put these
# "EXTERNAL"
# definitions at the top of your script.
#
#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#

EXTERNAL_INT="eth0"            # External Internet
interface
EXTERNAL_IP="97.158.253.25"   # Internet Interface IP
address
#-------------------------------------------------------
-
# Initialize our user-defined chains
#-------------------------------------------------------
-
iptables -N valid-src iptables -N valid-dst
#-------------------------------------------------------
-
# Verify valid source and destination addresses for all
packets
#-------------------------------------------------------
-

iptables -A INPUT   -i $EXTERNAL_INT -j valid-src
iptables -A FORWARD -i $EXTERNAL_INT -j valid-src
iptables -A OUTPUT  -o $EXTERNAL_INT -j valid-dst
iptables -A FORWARD -o $EXTERNAL_INT -j valid-dst

#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#
#
# Source and Destination Address Sanity Checks
# Drop packets from networks covered in RFC 1918
# (private nets)
# Drop packets from external interface IP
#
#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#
```

```
iptables -A valid-src -s $10.0.0.0/8      -j DROP
iptables -A valid-src -s $172.16.0.0/12   -j DROP
iptables -A valid-src -s $192.168.0.0/16  -j DROP
iptables -A valid-src -s $224.0.0.0/4     -j DROP
iptables -A valid-src -s $240.0.0.0/5     -j DROP
iptables -A valid-src -s $127.0.0.0/8     -j DROP
iptables -A valid-src -s 0.0.0.0/8        -j DROP
iptables -A valid-src -d 255.255.255.255 -j DROP
iptables -A valid-src -s 169.254.0.0/16   -j DROP
iptables -A valid-src -s $EXTERNAL_IP     -j DROP
iptables -A valid-dst -d $224.0.0.0/4     -j DROP
```

10.3_ Sự cho phép máy chủ DNS truy cập đến Firewall:

Firewall không thể tạo yêu cầu DNS queries đến Internet bởi vì Internet được yêu cầu cho hàm cơ bản của firewall, nhưng bởi vì Fedora Linux's yum RPM sẽ giúp giữ máy chủ cập nhật với trạng thái bảo vệ mới nhất. Những trạng thái theo sau sẽ cập nhật không chỉ cho firewall hoạt động như nhưng DNS client nhưng cũng cho những firewall làm việc trong một bộ đệm hoặc có vai trò như DNS server.

```
#-------------------------------------------------------------------------------------------------
-
# Allow outbound DNS queries from the FW and the replies
too #
# - Interface eth0 is the internet interface #
# Zone transfers use TCP and not UDP. Most home networks
# / websites using a single DNS server won't require TCP
# statements
#-------------------------------------------------------------------------------------------------
-

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
```

10.4 Cho phép WWW và SSH truy cập vào firewall:

Đoạn mã ngắn này là cho một firewall và gấp đôi như một web server được quản lý bởi người quản trị hệ thống web server "web server system adminitrator" qua những lớp vỏ bảo mật (SSH_secure shell). Những gói quay lại đã được dự định trước cho port 80 (WWW) và 22 (SSH) được phép. Vì vậy tạo những bước đầu tiên để thiết lập liên kết.. Ngược lại, những port trên (80 và 22) sẽ không được thiết lập chế độ bảo mật tại ngõ ra cho những gói chỉ được chuyển đi không quay về cho tất cả liên kết thiết lập được phép.

```
#-------------------------------------------------------
-# 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
```

10.5_Cho phép Firewall truy cập internet:

Đoạn mã iptables này có thể cho phép một user tren firewall sử dụng Web browser đến giao tiếp Internet. Đường truyền giao thức HTTP sử dụng TCP port 80, HTTPs (HTTP secure) port 443

```
#-------------------------------------------------------
-
# Allow port 80 (www) and 443 (https) connections from
the
# firewall
#-------------------------------------------------------
-
iptables -A OUTPUT -j ACCEPT -m state -state \
         NEW,ESTABLISHED,RELATED -o eth0 -p tcp -m \
         multiport --dport 80,443 -m multiport --sport \
         1024:65535
#-------------------------------------------------------
-
# Allow previously established connections
# - Interface eth0 is the internet interface
#-------------------------------------------------------
-

iptables -A INPUT -j ACCEPT -m state --state \
         ESTABLISHED,RELATED -i eth0 -p tcp
```

Nếu muốn tất cả đường truyền từ firewall được chấp nhận, sau đó xoá:

```
-m multiport --dport 80,443 -m multiport --sport \
1024:65535
```

10.6_ Cho phép mạng ở nhà truy cập vào firewall:

Ví dụ: eth1 được liên kết với mạng ở nhà dùng địa chỉ IP từ mạng 192.168.1.0. Tất cả đường truyền này và firewall được giả sử là liên kết được:

Những rule được cần cho liên kết giao tiếp đến Internet để cho phép chỉ những cổng đặc trưng, những loại liên kết và có thể điều chỉnh những server có truy cập đến firewall và mạng ở nhà.

```
#-----------------------------------------------------------
-# Allow all bidirectional traffic from your firewall to
#the
# protected network
# - Interface eth1 is the private network interface
#-----------------------------------------------------------
-

iptables -A INPUT -j ACCEPT -p all -s 192.168.1.0/24 -i
eth1 iptables -A OUTPUT -j ACCEPT -p all -d
192.168.1.0/24 -o eth1
```

10.7_ Mặt nạ (Masquerade_many to one NAT):

Đường truyền từ tất cả thiết bị trên một hoặc nhiều mạng được bảo vệ sẽ xuất hiện như là nó bắt đầu từ địa chỉ IP đơn trên vị trí Internet của firewall.

Địa chỉ IP mặt nạ (masquerade) luôn luôn mặc định đến địa chỉ IP của giao tiếp chính của firewall. Ưu điểm của địa chỉ IP mặt nạ (masquerade) là ta không phải chỉ rõ địa chỉ IP NAT. Điều này tạo cho việc cấu hình bảng iptables NAT với giao thức DHCP.

Ta có thể cấu hình nhiều đến một NAT cho một tên IP bằng cách sử dụng POSTROUTING và không dùng trạng thái MASQUERADE.

Việc che đậy (Masquerading) phụ thuộc vào Hệ Điều Hành Linux được cấu hình để cập nhật định tuyến giữa internet và giao tiếp mạng riêng của firewall. Điều này được thực hiện bởi IP enabling bằng cách cho file /proc/sys/net/ipv4/ip_forward giá trị 1 như là đối với giá trị mặc định 0

Một masquerading được thiết lập sử dụng POSTROUTING chain của bảng nat table, ta sẽ phải định dạng iptables để cho phép nhiều gói đi qua giữa 2 bề mặt. Để làm được điều này, sử dụng FORWARD chain của filter table. Nhiều hơn, nhiều gói liên quan những liên kết NEW và ESTABLISHED sẽ được cho phép outbound đến Internet, nhưng chỉ những gói liên quan đến liên kết ESTABLISHES sẽ được phép inbound. Điều này sẽ giúp bảo vệ mạng ở nhà từ bất cứ một người nào cố gắng kết nối với mạng nhà từ Internet.

```
#-----------------------------------------------------------
-
# Load the NAT module
# Note: It is best to use the /etc/rc.local example in
# this
#       chapter. This value will not be retained in the
#       /etc/sysconfig/iptables file. Included only as a
# reminder.
#-----------------------------------------------------------
-
```

```
modprobe iptable_nat

#--------------------------------------------------------
-
# Enable routing by modifying the ip_forward /proc
#  filesystem
#  file
#
# Note: It is best to use the /etc/sysctl.conf example in
# this
#       chapter. This value will not be retained in the
#       /etc/sysconfig/iptables file. Included only as a
#       reminder.
#--------------------------------------------------------
-

echo 1 > /proc/sys/net/ipv4/ip_forward

#--------------------------------------------------------
-
# Allow masquerading
# - Interface eth0 is the internet interface
# - Interface eth1 is the private network interface
#--------------------------------------------------------
-

iptables -A POSTROUTING -t nat -o eth0 -s 192.168.1.0/24
\
         -d 0/0 -j MASQUERADE

#--------------------------------------------------------
-
# Prior to masquerading, the packets are routed via the
# filter
# table's FORWARD chain.
# Allowed outbound: New, established and related
# connections
# Allowed inbound : Established and related connections
#--------------------------------------------------------
-

iptables -A FORWARD -t filter -o eth0 -m state -state \
         NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -t filter -i eth0 -m state --state \
         ESTABLISHED,RELATED -j ACCEPT
```

10.8. Port forwarding theo loại NAT (giao thức DHCP DSL):

Một số trường hợp, nhiều home user có thể nhận địa chỉ IP công cộng DHCP đơn từ những nhà cung cấp dịch vụ ISP. Nếu một Linux firewall cũng là giao tiếp với Internet và ta muốn dẫn một trang Web trên một trong những home server được bảo vệ

NAT, sau đó ta phải sử dụng port forwarding. Ở đây việc kết hợp địa chỉ IP đơn của firewall, địa chỉ IP của server, và port nguồn/đích của đường truyền có thể được sử dụng bổ sung đường truyền.

Port forwarding được điều chỉnh bởi PREROUTING chain của bảng nat table.

Giống như Masquerading, modun iptables_nat phải được tải và định tuyến phải được hiển thị cho port forwarding để làm việc. Định tuyến cũng phải được phép trong bảng iptables với FORWARD chain, điều này bao gồm tất cả liên kết NEW inbound từ Internet làm phù hợp port forwarding và tất cả gói liên kết với kết nối ESTABLISHED trong những sự điều khiển:

```
     #--------------------------------------------------------
-
     # Load the NAT module
     # Note: It is best to use the /etc/rc.local example in
     # this
     #       chapter. This value will not be retained in the
     #       /etc/sysconfig/iptables file. Included only as a
     #       reminder.
     #--------------------------------------------------------
-

     modprobe iptable_nat

     #--------------------------------------------------------
-
     # Get the IP address of the Internet interface eth0
(linux
     #       only)
     #
     # You'll have to use a different expression to get the IP
     #       address
     # for other operating systems which have a different
     ifconfig
     #       output
     # or enter the IP address manually in the PREROUTING
     #       Statement
     #
     # This is best when your firewall gets its IP address
     using
     #       DHCP.
     # The external IP address could just be hard coded
("typed
     # in
     # normally")
     #--------------------------------------------------------
-

     external_int="eth0"
     external_ip=""ifconfig $external_int | grep 'inet addr'
|\
                 awk '{print $2}' | sed -e 's/. *://'""
```

```
#------------------------------------------------------------
-
# Enable routing by modifying the ip_forward /proc
# filesystem
#       File
#
# Note: It is best to use the /etc/sysctl.conf example in
#       this chapter. This value will not be retained in
# the
#       /etc/sysconfig/iptables file. Included only as a
#       reminder.
#------------------------------------------------------------
-
echo 1 > /proc/sys/net/ipv4/ip_forward
#------------------------------------------------------------
-
# Allow port forwarding for traffic destined to port 80
of
#       the
# firewall's IP address to be forwarded to port 8080 on
#       server
# 192.168.1.200
#
# - Interface eth0 is the internet interface
# - Interface eth1 is the private network interface
#------------------------------------------------------------
-

iptables -t nat -A PREROUTING -p tcp -i eth0 -d \
$external_ip --dport 80 --sport 1024:65535 -j DNAT -to \
             192.168.1.200:8080

#------------------------------------------------------------
-
# After DNAT, the packets are routed via the filter
# table's
# FORWARD chain.
# Connections on port 80 to the target machine on the
# private
# network must be allowed.
#------------------------------------------------------------
-

iptables -A FORWARD -p tcp -i eth0 -o eth1 -d \
         192.168.1.200 --dport 8080 --sport 1024:65535 \
         -m state --state NEW -j ACCEPT
iptables -A FORWARD -t filter -o eth0 -m state --state \
         NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -t filter -i eth0 -m state --state \
         ESTABLISHED,RELATED -j ACCEPT
```

10.9_ NAT tĩnh (SNAT):

Ví dụ: tất cả đường truyền đến một địa chỉ IP công cộng riêng biệt, được chuyển đổi đến một server đơn trên Subnet được bảo vệ. Bởi vig firewall có nhiều hơn một địa chỉ IP, ta không thể thực hiện MASQUERADE; nó sẽ bắt buộc khởi tạo như địa chỉ IP của giao tiếp chính và không nhưng bất cứ những địa chỉ IP trùng lặp mà firewall có thể có. Thay vì vậy, sử dụng SNAT để chỉ rõ địa chỉ IP bị trùng lặp được sử dụng cho việc liên kết ban đầu bởi những server khác trong mạng được bảo vệ.

Ghi chú: Mặc dù những NAT của bảng nat table, tất cả đường truyền đến server đích (192.168.1.100 đến 102), chỉ liên kết với port 80, 443 và 22 là được phép thông qua bởi FORWARD chain. Ta phải chỉ rõ lựa chọn riêng biệt –m multiport khi ta cần làm cho thích hợp những cổng không tuần tự (multiple non-sequential) cho cả nguồn và đích.

Trong ví dụ này, firewall có:

Sử dụng one to one NAT tạo server 192.168.1.100 trên home network xuất hiện trên Internet như những địa chỉ IP (97.158.253.26).

+ Tạo một many to one NAT cho địa chỉ IP 192.168.1.100 ở home network, tất cả những server như những địa chỉ IP (97.158.253.26). Điều này khác từ khởi tạo.

Ta tạo những địa chỉ IP trùng lặp cho mỗi nhóm IP Internet cho one to one NAT

```
#------------------------------------------------------------
-
# Load the NAT module
# Note: It is best to use the /etc/rc.local example in this chapter. This value will
# not
# be retained in the /etc/sysconfig/iptables file. Included only as a reminder.
#------------------------------------------------------------
-

modprobe iptable_nat

#------------------------------------------------------------
-
# Enable routing by modifying the ip_forward /proc filesystem file
# Note: It is best to use the /etc/sysctl.conf example in this chapter. This value
will
# not be retained in the /etc/sysconfig/iptables file. Included only as a
reminder.
#------------------------------------------------------------
-

echo 1 > /proc/sys/net/ipv4/ip_forward

# NAT ALL traffic:
###########
# REMEMBER to create aliases for all the internet IP addresses below
###########
#
```

```
# TO:                FROM:              MAP TO SERVER:
# 97.158.253.26      Anywhere           192.168.1.100(1:1 NAT-Inbound)
# Anywhere           2.168.1.100        97.158.253.26(1:1 NAT–Outbound)
# Anywhere           192.168.1.0/24     97.158.253.29(FW  IP)
#
# SNAT is used to NAT all other outbound connections initiated
# from the protected network to appear to come from
# IP address 97.158.253.29
#
# POSTROUTING:
# NATs source IP addresses. Frequently used to NAT connections
# from your home network to the Internet
#
# PREROUTING:
# NATs destination IP addresses. Frequently used to NAT
# connections from the Internet to your home network
#
# - Interface eth0 is the internet interface
# - Interface eth1 is the private network interface
#-----------------------------------------------------------
_

# PREROUTING statements for 1:1 NAT
# (Connections originating from the Internet)
#-----------------------------------------------------------
_

iptables -t nat -A PREROUTING -d 97.158.253.26 -i eth0 \
         -j DNAT --to-destination 192.168.1.100
#-----------------------------------------------------------
_

# POSTROUTING statements for 1:1 NAT
# (Connections originating from the home network servers)
#-----------------------------------------------------------
_

iptables -t nat -A POSTROUTING -s 192.168.1.100 -o eth0 \
         -j SNAT --to-source 97.158.253.26
#-----------------------------------------------------------
_

# POSTROUTING statements for Many:1 NAT
#  (Connections originating from the entire home network)
#-----------------------------------------------------------
_

iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -j SNAT
\
         -o eth0 --to-source 97.158.253.29
#-----------------------------------------------------------
_

# Allow forwarding to each of the servers configured for 1:1 NAT
#  (For connections originating from the Internet. Notice how you use the
```

```
# real
# IP addresses here)
#--------------------------------------------------------
_

iptables -A FORWARD -p tcp -i eth0 -o eth1 -d \
          192.168.1.100 -m multiport --dport 80,443,22 \
          -m state --state NEW -j ACCEPT
#--------------------------------------------------------
_
# Allow forwarding for all New and Established SNAT connections originating
# on the # home network AND already established DNAT connections
#--------------------------------------------------------
_

iptables -A FORWARD -t filter -o eth0 -m state --state \
         NEW,ESTABLISHED,RELATED -j ACCEPT



#--------------------------------------------------------
_
# Allow forwarding for all 1:1 NAT connections originating on the Internet that
have  # already passed through the NEW forwarding statements above
#--------------------------------------------------------
_

iptables -A FORWARD -t filter -i eth0 -m state --state \
         ESTABLISHED,RELATED -j ACCEPT
#--------------------------------------------------------
_
# Allow forwarding to each of the servers configured for 1:1 NAT
#  (For connections originating from the Internet. Notice how you use the real
IP
#  addresses here)
#--------------------------------------------------------
_

iptables -A FORWARD -p tcp -i eth0 -o eth1 -d \
         192.168.1.100 -m multiport --dport 80,443,22 -m
     \
         state --state NEW -j ACCEPT
#--------------------------------------------------------
_
# Allow forwarding for all New and Established SNAT connections originating
# on the # home network AND already established DNAT connections
#--------------------------------------------------------
_

iptables -A FORWARD -t filter -o eth0 -m state --state \
         NEW,ESTABLISHED,RELATED -j ACCEPT
```

```
#-------------------------------------------------------
-
# Allow forwarding for all 1:1 NAT connections originating on the Internet that
# have # already passed through the NEW forwarding statements above
#-------------------------------------------------------
-

iptables -A FORWARD -t filter -i eth0 -m state --state \
         ESTABLISHED,RELATED -j ACCEPT
```

10.10_ Sửa lỗi bảng iptables:

Một số công cụ cho phép sửa lỗi đoạn mã firewall iptables. Một trong những phương pháp tốt nhất là loại bỏ tất cả những gói bị khoá.

* Kiểm tra the firewall log:

Ta theo dõi những gói đi qua firewall có trong danh sách bảng iptables của những rule sử dụng LOG target.

LOG target sẽ:

+ Tạm dừng tất cả đường truyền để chỉnh sửa rule trong iptables trong nơi nó được chứa.

+ Tự động viết vào file **/var/log/messages** và sau đó thực thi rule kế tiếp

Để tạm dừng đường truyền không mong muốn, ta phải thêm vào rule phù hợp với một DROP target sau LOG rule.

Tạm dừng một nhóm gói bị lỗi vào file **/var/log/messages**.

```
#-------------------------------------------------------
#  Log and drop all other packets to file /var/log/messages
#  Without this we could be crawling around in the dark
#-------------------------------------------------------

iptables -A OUTPUT -j LOG
iptables -A INPUT -j LOG
iptables -A FORWARD -j LOG

iptables -A OUTPUT -j DROP
iptables -A INPUT -j DROP
iptables -A FORWARD -j DROP
```