استفاده از آن‌ها می‌توان بسته‌های متعلق به آن‌ها را تصفیه کرد. حالت‌های مختلف عبارتند از:

- New: مشخصه ارتباط‌هایی است که تنها یک بسته در یک جهت ارسال کرده‌اند. در واقع پس از دیدن اولین بسته از هر ارتباط (بسته‌ای که پرچم SYN از سرآیند TCP آن‌ها روشن باشد.)، وضعیت آن به New تغییر می‌کند.
- Established: مشخصه ارتباط‌هایی است که بسته در هر دو جهت از طریق آن‌ها ارسال شده است. پس از دیدن اولین پاسخ به بسته ارسال شده (بسته‌ای که پرچم ACK از سرآیند TCP آن‌ها روشن باشد)، مشخصه ارتباط ایجاد شده از New به Established تغییر می‌کند.
- Related: مشخصه ارتباط‌هایی است که بسته جدیدی ملاقات می‌کنند، با این تفاوت که بسته ایجاد شده به: ارتباط برقرار شده قبلی مرتبط است. یک مثال شناخته شده از این نوع ارتباطات، ارتباط داده‌ای در قرارداد FTP می‌باشد که به ارتباط کنترلی از آن مربوط می‌شود.

در لینک زیر مثال‌های بسیار خوبی برای Stateful Packet Filtering آمده است.

https://wiki.archlinux.org/index.php/simple_stateful_firewall

برای اینکه بتوانید از Stateful Packet Filtering استفاده نمایید باید ماژول‌های آن در کرنل بارگذاری شده باشد برای این کار دستورات زیر را اجرا نمایید:

```
vim /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
```

redirect کردن

REDIRECT برای برگرداندن بسته‌ها به ماشین محلی (که حفاظ روی آن نصب شده) مورد استفاده قرار می‌گیرد. از این امکان IPtables زمانی استفاده می‌شود که بخواهیم سرویس‌های نصب شده در ماشین حفاظ از دید کاربران شفاف باشد. فرض کنید می‌خواهیم در کنار http server از یک http proxy مثل squid استفاده کنیم. با استفاده از این امکان به راحتی می‌توان، همه بسته‌هایی که می‌خواهند به درگاه ۸۰ از

ماشین حفاظ متصل شوند را به راحتی به سمت squid هدایت کرد. دستور زیر این هدف را برآورده می‌کند.

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 3128
```

مثال‌هایی از iptables

در این قسمت نمونه کاربردهایی از iptable گفته خواهد شد. توجه کنید در تمامی مثال‌های زیر: سوئیچ A- به معنی Append کردن Rule به انتهای فایل پیکربندی iptables، سوئیچ s- برای مشخص کردن Source Address، سوئیچ d- برای مشخص کردن Destination Address، سوئیچ p- برای مشخص کردن نوع پروتکل از میان udp، tcp و icmp، سوئیچ sport- برای مشخص کردن شماره پورت مبدأ، سوئیچ dport- برای مشخص کردن شماره پورت مقصد و J- برای تعیین نوع عمل بر روی بسته از میان ACCEPT، DROP و یا REJECT و در نهایت سوئیچ D- یک خط Rule خاص را پاک می‌کند.

مثال زیر تمامی کلاینت‌های شبکه ۲۴/۱۹۲,۱۶۸,۱,۰ را قادر به دسترسی به پورت ۵۳ (یعنی DNS) می‌کند. (بسته‌ها یا درخواست‌های آن‌ها را قبول می‌کند)

```
iptables -A INPUT -s 192.168.1.0/24 -p udp –dport 53 -j ACCEPT
```

دو خط کد زیر بسته‌های icmp (دستور ping) را REJECT می‌کنند. تفاوت REJECT و DROP این است که اگر بسته‌ای REJECT شود یک پیغام به درخواست‌کننده (مبدأ) فرستاده می‌شود.

```
iptables -A OUTPUT -p icmp -j REJECT
```

```
iptables -A INPUT -p icmp -j REJECT
```

دو خط زیر، دو خط کد بالا را پاک می‌کنند.

```
iptables -D OUTPUT -p icmp -j REJECT
iptables -D INPUT -p icmp -j REJECT
```

دستور زیر درخواست یک کلاینت خاص (یک آدرس خاص) را DROP می‌کند.

```
iptables -A INPUT -s 192.168.1.200 -j DROP
```

دستور زیر درخواست‌های یک کلاینت خاص مبنی بر دسترسی به پورت ۸۰ را DROP می‌کند.

```
iptables -A INPUT -s 192.168.1.200  -p tcp -dpost -j
DROP
```

دستور زیر بسته‌های ssh (پورت ۲۲ و پروتکل tcp) را از یک مبدأ (کلاینت/آدرس خاص) DROP می‌کند.

```
iptables -A INPUT -s 192.168.1.200 -p tcp –dport 22
-j DROP
```

دستور زیر Rule بالایی را پاک می‌کند.

```
iptables -D INPUT -s 192.168.1.200 -p tcp –dport 22
-j DROP
```

دستورهای زیر به ترتیب LOG هایی را در فایل /var/log/message برای دسترسی‌های خروجی (Outgoing Access) و دسترسی‌های ورودی (Incoming Access) به پورت ۲۱ و پروتکل tcp (پروتکل ftp) از هر مبدائی (نشان داده شده توسط 0/0 -s) ثبت می‌کنند. log–prefix یعنی یک رشته خاص که در اینجا FTP:> است را به ابتدای پیام اضافه می‌کند.

iptables -A OUTPUT -p tcp -s 0/0 –dport 21 -j :> ”
LOG –log-prefix “FTP

iptables -A INPUT -p tcp -s 0/0 –dport 21 -j LOG –:> ”
log-prefix “FTP

دستورهای زیر همان کار LOG بالا را برای پروتکل icmp یا دستور ping در فایل /var/log/message ثبت می‌کنند.

iptables -A OUTPUT -p icmp -j LOG –log-prefix :> ”
“PING

iptables -A INPUT -p icmp -j LOG –log-prefix :> ”
“PING

دستور زیر پورت ۲۲ را باز می‌کند:

iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT

باز کردن پورت SSH برای تمامی ارتباطات ورودی

iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT

باز کردن پورت ssh برای یک ip یا رنج ip خاص

iptables -A INPUT -i eth0 -p tcp -s xxx.xxx.xxx.xxx/24 --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A INPUT -i eth0 -p tcp -s xxx.xxx.xxx.xxx --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT

```
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

باز کردن پورت ۸۰ برای http

```
iptables -A INPUT -i eth0 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A OUTPUT -o eth0 -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
```

باز کردن پورت ۴۴۳ برای https

```
iptables -A INPUT -i eth0 -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A OUTPUT -o eth0 -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT
```

باز کردن چند پورت به‌صورت یک جا

```
iptables -A INPUT -i eth0 -p tcp -m multiport --dports 22,80,443 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A OUTPUT -o eth0 -p tcp -m multiport --sports 22,80,443 -m state --state ESTABLISHED -j ACCEPT
```

باز کردن پورت برای ارتباط خروجی ssh

```
iptables -A OUTPUT -o eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A INPUT -i eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

باز کردن پورت خروجی ssh تنها برای یک شبکه خاص

```
iptables -A OUTPUT -o eth0 -p tcp -d
192.168.101.0/24 --dport 22 -m state --state
NEW,ESTABLISHED -j ACCEPT

iptables -A INPUT -i eth0 -p tcp --sport 22 -m state --
state ESTABLISHED -j ACCEPT
```

باز کردن پورت https برای ارتباطات خروجی

```
iptables -A OUTPUT -o eth0 -p tcp --dport 443 -m state
--state NEW,ESTABLISHED -j ACCEPT

iptables -A INPUT -i eth0 -p tcp --sport 443 -m state --
state ESTABLISHED -j ACCEPT
```

ایجاد امکان ping از داخل به خارج

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

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

ایجاد امکان ping از خارج به داخل

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

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

ایجاد امکان دسترسی loopback

```
iptables -A INPUT -i lo -j ACCEPT

iptables -A OUTPUT -o lo -j ACCEPT
```

ایجاد امکان دسترسی به شبکه خارجی eth1 از شبکه داخلی eth0

```
iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
```

باز کردن دسترسی خروجی پورت dns

```
iptables -A OUTPUT -p udp -o eth0 --dport 53 -j ACCEPT
iptables -A INPUT -p udp -i eth0 --sport 53 -j ACCEPT
```

redirect کردن تمامی درخواست‌های SMTP از اینترفیس eth0 به یک کامپیوتر دیگر داخل شبکه (LAN)

```
# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 25 -j DNAT --to 192.168.0.88:25
# iptables -A FORWARD -p tcp -d 192.168.0.88 --dport 25 -j ACCEPT
```

redirect کردن گروهی از پورت‌ها

```
# iptables -t nat -A PREROUTING -p tcp -d 203.145.184.246 --dport 20:23 -j DNAT --to 192.168.0.88:20-23
# iptables -A FORWARD -p tcp -d 192.168.0.88 --dport 20:23 -j ACCEPT
# iptables -t nat -A PREROUTING -p udp -d 203.145.184.246 --dport 20:23 -j DNAT --to 192.168.0.88:20-23
# iptables -A FORWARD -p udp -d 192.168.0.88 --dport 20:23 -j ACCEPT
```

بستن پروتکل ICMP

ICMP (مخفف Internet Control Message Protocol) یکی از اصلی‌ترین پروتکل‌های TCP/IP برای ارسال و دریافت پیام‌ها می‌باشد. ICMP تنها جهت ارسال پیام‌هایی (بسته‌های IP) از یک ماشین

(سیستم PC یا Router یا Mobile Device و هر آنچه که درون شبکه LAN یا اینترنت باشد و از TCP/IP استفاده کند) به ماشین دیگر استفاده می‌کنیم. دستور ping اصلی‌ترین دستور (برنامه) ای است که از این پروتکل استفاده می‌کند. دو پیام echo request و echo reply در این پروتکل وجود دارند. ping توسط پیام‌های echo request پیامی به ماشین دیگر فرستاده و سمت دیگر با echo reply پاسخ می‌دهد. شکل زیر این مسئله را نشان می‌دهد.

Type	Code	Description
0 – Echo Reply	0	Echo reply (used to ping)
8 – Echo Request	0	Echo request (used to ping)

ICMP تنها پروتکلی است که فاقد شماره پورت است و به‌منظور رفع مشکلات درون شبکه استفاده می‌شود. سه قانون زیر در iptables معرف این هستند که همه چیز DROP شود؛ یعنی هر بسته که وارد می‌شود (INPUT Chain)، هر بسته‌ای که خارج می‌شود (OUTPUT Chain) و هر بسته‌ای که می‌خواهد Forward شود (FORWARD Chain) باید DROP شوند به‌جز آن‌هایی که ما تعیین می‌کنیم.

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

مشکل اصلی این است که با این سه قانون همه چیز فیلتر می‌شود ولی می‌خواهیم تنها بسته‌های ICMP را فیلتر کنیم. باید توجه داشت که فیلتر کردن ICMP به این صورت است که پیام‌های echo request خروجی، echo reply ورودی و همچنین بسته‌های echo request ورودی و echo reply خروجی را باید فیلتر کنیم.

دستور زیر پیام‌های echo request ورودی و پیـام‌هـای echo reply خروجی را فیلتر می‌کند.

```
iptables -A INPUT -p icmp –icmp-type 8 -s 0/0 -d
192.168.1.4 -m state –state NEW,ESTABLISHED -j
DROP

iptables -A OUTPUT -p icmp –icmp-type 0 –s
192.168.1.3 -d 0/0 -m state –state ESTABLISHED -j
DROP
```

سوئیچ p- برای تعیین پروتکل در iptables استفاده می‌شود که یکی از مقادیر TCP، UDP و ICMP را می‌تواند بگیرد. همچنـین در جلـوی سوئیچ icmp-type– از دو عدد ۰ و ۸ استفاده شده است کـه در جـدول بالا نشان داده شده‌اند و می‌توانیـد بجـای عـدد از معـادل حروفـی آن‌هـا استفاده کنید. ۰/۰ یعنی از هر آدرسی درون شبکه. دستور زیـر در مـورد پیام‌های echo request و echo reply عکس دستور بالا می‌باشد.

```
iptables -A OUTPUT -p icmp –icmp-type 8 -s
192.168.1.3 -d 0/0 -m state –state NEW,ESTABLISHED -
j DROP

iptables -A INPUT -p icmp –icmp-type 0 -s 0/0 -d
192.168.1.4 -m state –state ESTABLISHED -j DROP
```

۴ خط بالا در ماشینی که به‌عنوان فایروال در نظر گرفتـه شـده اسـت اجرا می‌شود (ماشین‌های مبدأ و مقصد مشخص هستند.) فرض کنیـد در شـبکه‌ای مـی‌خواهیـد ماشـین ۱۹۲,۱۶۸,۱,۴ فقـط و فقـط از ماشـین ۱۹۲,۱۶۸,۱,۳ ping شـود یعنـی وقتـی روی ماشـینی بـه آدرس ۱۹۲,۱۶۸,۱,۵ دستور 192.168.1.4 ping را اجـرا کنـیم جلـوی ping کردن گرفته شود ولـی از ماشـین ۱۹۲,۱۶۸,۱,۳ بتـوانیم دسـتور ping

192.168.1.4 را انجام داده و پاسخ را دریافت کنیم. برای این کار باید ۴ خط زیر را در ماشین با آدرس ۱۹۲,۱۶۸,۱,۴ بنویسیم.

```
iptables -A INPUT -p icmp –icmp-type 8 -s
192.168.1.3  -m state –state NEW,ESTABLISHED -j
ACCESS

iptables -A OUTPUT -p icmp –icmp-type 0 –s
192.168.1.3   -m state –state ESTABLISHED -j ACCESS

iptables -A OUTPUT -p icmp –icmp-type 8 -s
192.168.1.3   -m state –state NEW,ESTABLISHED -j
ACCESS

iptables -A INPUT -p icmp –icmp-type 0 -s
192.168.1.3  -m state –state ESTABLISHED -j ACCESS
```

اجازه دادن به WWW و SSH برای دسترسی به فایروال

کدی که در لینک زیر آمده است این کار را انجـام مـی‌دهـد و مثـالی برای حالت Stateful نیز می‌باشد.

http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO_:_Ch14_:_Linux_Firewalls_Using_iptables#Sample_iptables_Scripts

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

اجازه دادن به فایروال برای دسترسی به اینترنت

کدی که در لینک زیر آمده است این کار را انجـام مـی‌دهـد و مثـالی برای حالت Stateful نیز می‌باشد.

http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO_:_Ch14_:_Linux_Firewalls_Using_iptables#Sample_iptables_Scripts

```
#--------------------------------------------------------------
# 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 --dports 80,443 --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
```

If you want all TCP traffic originating from the firewall to be accepted, then remove the line:

```
-m multiport --dports 80,443 --sport 1024:65535
```

بررسی LOG های فایروال

برای LOG گیری از کد زیر استفاده نمایید:

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

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

نمونه‌ای از LOG های اخذ شده و شرح آن‌ها در زیر آمده است:

1) Firewall denies replies to DNS queries (UDP port 53) destined to server 192.168.1.102 on the home network.

Feb 23 20:33:50 raja kernel: IN=wlan0 OUT= MAC=00:06:25:09:69:80:00:a0:c5:e1:3e:88:08:00 SRC=192.42.93.30 DST=192.168.1.102 LEN=220 TOS=0x00 PREC=0x00 TTL=54 ID=30485 PROTO=UDP SPT=53 DPT=32820 LEN=200

2) Firewall denies Windows NetBIOS traffic (UDP port 138)

Feb 23 20:43:08 raja kernel: IN=wlan0 OUT= MAC=ff:ff:ff:ff:ff:ff:00:06:25:09:6a:b5:08:00 SRC=192.168.1.100 DST=192.168.1.255 LEN=241 TOS=0x00 PREC=0x00 TTL=64 ID=0 DF PROTO=UDP SPT=138 DPT=138 LEN=221

3) Firewall denies Network Time Protocol (NTP UDP port 123)

Feb 23 20:58:48 raja kernel: IN= OUT=wlan0 SRC=192.168.1.102 DST=207.200.81.113 LEN=76 TOS=0x10 PREC=0x00 TTL=64 ID=0 DF PROTO=UDP SPT=123 DPT=123 LEN=56

Intrusion Detection and Log Analysis with iptables که psad می‌باشد نرم‌افزاری برای تحلیل LOG های iptables، شناسایی اسکن پورت‌ها و ترافیک‌های مشکوک است. در لینک زیر علاوه بر خود نرم‌افزار، نحوه کار با آن به‌صورت کامل شرح داده شده است.

http://cipherdyne.org/psad/

در لینک زیر نیز نحوه نصب و کار ابتدایی با آن در اوبونتو شرح داده شده است.

https://www.digitalocean.com/community/tutorials/how-to-use-psad-to-detect-network-intrusion-attempts-on-an-ubuntu-vps

inactive بودن فایروال

اسکریپت startup فایروال در مسیر /etc/sysconfig/iptables وجود دارد. در صورتی که این فایل وجود نداشته باشد status همیشه inactive نشان داده می‌شود. برای رفع این اشکال کد زیر را اجرا نمایید:

```
[root@raja tmp]# touch /etc/sysconfig/iptables
[root@raja tmp]# chmod 600 /etc/sysconfig/iptables
[root@raja tmp]# systemctl start iptables.service
[root@raja tmp]# systemctl status iptables.service
iptables.service - IPv4 firewall with iptables
  Loaded: loaded (/usr/lib/systemd/system/iptables.service; enabled)
  Active: active (exited) since Thu, 09 Aug 2012 22:25:50 -0700; 4s ago
  rocess: 19177 ExecStop=/usr/libexec/iptables.init stop (code=exited, status=0/SUCCESS)
  rocess: 19231 ExecStart=/usr/libexec/iptables.init start (code=exited, status=0/SUCCESS)
  CGroup: name=systemd:/system/iptables.service

Aug 09 22:25:50 web-003 iptables.init[19231]: iptables: Applying firewall rules: [  OK  ]
```

مثالی جامع برای حالت stateful firewall

مثال زیر که در لینک زیر آمده است مثالی جامع برای حالت Stateful می‌باشد که بررسی آن خالی از لطف نیست.

https://gist.github.com/TylerJFisher/8961054#file-gistfile1-sh

```
#!/bin/bash
sudo apt-get install iptables-persistent


# Remove all iptables rules
echo 'Clearing all iptables rules'
iptables=/sbin/iptables
iptables -F
iptables -X
iptables -t raw -F
iptables -t raw -X
iptables -t filter -F
iptables -t filter -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X
iptables -t security -F
```

```
iptables -t security -X


# Create two user-defined chains: TCP, UDP (to handle port forwarding)
iptables -N TCP
iptables -N UDP


# Since the server is not acting as a router, do not forward packets
iptables -P FORWARD DROP


# Since outgoing traffic will not be filtered, set OUTPUT policy to ACCEPT
iptables -P OUTPUT ACCEPT


# ACCEPT ICMP traffic
iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT


# ACCEPT incoming traffic via loopback adapter
iptables -A INPUT -i lo -j ACCEPT


# ACCEPT ICMPv6 Neighbour Discovery packets (classified as invalid)
```

iptables -A INPUT -p 41 -j ACCEPT

ACCEPT ICMP echo requests

iptables -A INPUT -p icmp --icmp-type 8 -m conntrack --ctstate NEW -j ACCEPT

Allow inbound TCP traffic on specified destination ports

iptables -A TCP -p tcp --dport 22 -j ACCEPT

iptables -A TCP -p tcp --dport 2222 -j ACCEPT

iptables -A TCP -p tcp --dport 80 -j ACCEPT

iptables -A TCP -p tcp --dport 443 -j ACCEPT

Allow inbound UDP traffic on specified destination ports

iptables -A UDP -p udp --dport 53 -j ACCEPT

ACCEPT TCP, UDP traffic with state NEW

iptables -A INPUT -p tcp --syn -m conntrack --ctstate NEW -j TCP

iptables -A INPUT -p udp -m conntrack --ctstate NEW -j UDP

DROP TCP packets with state NEW if connection is not initiated with a SYN packet

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

DROP packets with an invalid state

iptables -A INPUT -m conntrack --ctstate INVALID -j DROP

REJECT TCP connections with RST packets, UDP packets with ICMP port unreachable messages

iptables -A INPUT -p tcp -m recent --set --name TCP-PORTSCAN -j REJECT --reject-with tcp-rst

iptables -A INPUT -p udp -m recent --set --name UDP-PORTSCAN -j REJECT --reject-with icmp-port-unreachable

DROP inbound XMAS packets

iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP

DROP inbound FIN packets

iptables -A INPUT -p tcp --tcp-flags ALL FIN -j DROP

DROP inbound malformed NULL packets

iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP

```
# DROP traffic that is not explicitly allowed through the firewall
iptables -P INPUT DROP

# REJECT other connections with ICMP protocol unreachable messages
iptables -A INPUT -j REJECT --reject-with icmp-proto-unreachable

# Show tables
iptables -L INPUT -n --line-numbers –v
```

همچنین در لینک‌های زیر iptables به‌صورت مشروح توضیح داده شده است.

http://rlworkman.net/howtos/iptables/iptables-tutorial.html

http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO_:_Ch14_:_Linux_Firewalls_Using_iptables#.VqpKmmGmnJA

دفاع ابتدایی از سیستم‌عامل توسط sysctl.conf

علاوه بر Rule هایی که می‌توان درون iptable برای جلوگیری از حمله به سیستم‌عامل نوشت، توسط فایل sysctl.conf می‌توان جلوی یک سری از فعالیت‌های مشکوک سیستم را گرفت. به پیکربندی نمونه که در آدرس زیر وجود دارد توجه نمایید:

http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO_:_Ch14_:_Linux_Firewalls_Using_iptables#.VqpKmmGmnJA

```
# 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 a

network, NAT included

#--

net/ipv4/ip_forward = 1

در نهایت برای اعمال تغییرات دستور زیر را اجرا نمایید:

```
sysctl –p
```

نصب و پیکربندی OpenSSH

ssh یک پروتکل امن برای ارتباط و مدیریت یک ماشین راه دور است. برخلاف مشابه خود یعنی telnet که ارتباط میان دو سیستم ناامن یا به اصطلاح ارتباطی واضح است، ارتباط میان دو سیستم رمز شده خواهد بود و امکان دزدی اطلاعات توسط شخص یا سیستم سومی حتی با گوش دادن به ارتباط به دلیل رمزنگاری وجود ندارد. به دلیل استفاده از کلید عمومی و کلید خصوصی حتی اگر شخص سوم به کلید عمومی دسترسی پیدا کند به دلیل اینکه کلید خصوصی تنها در خود همان ماشین صاحب کلید خصوصی وجود دارد شخص سوم نمی‌تواند اطلاعات را از حالت رمز شده خارج کند. در توزیع‌های مبتنی بر RedHat دو بسته openssh-client و openssh-server به همراه openssl برای رمزنگاری باید نصب شده باشد. از دستور زیر برای اطلاع از نصب بودن بسته‌ها استفاده کنید.

```
rpm  -qa | grep openssh*
```

خروجی:

```
openssh-5.3p1-70.el6.i686
openssh-askpass-5.3p1-70.el6.i686
openssl-1.0.0-20.el6.i686
openssh-clients-5.3p1-70.el6.i686
openssh-server-5.3p1-70.el6.i686
```

اگر OpenSSH روی توزیع نصب نباشد دستور زیر را اجرا کنید.

```
yum -y install openssh-server openssh-clients
```