**<http://venkataraoss.blogspot.in/2011/02/linux-firewall-tutorial-iptables-tables.html>**

**Linux Firewall Tutorial: IPTables Tables, Chains, Rules Fundamentals**

iptables firewall is used to manage packet filtering and NAT rules. IPTables comes with all Linux distributions. Understanding how to setup and configure iptables will help you manage your Linux firewall effectively.

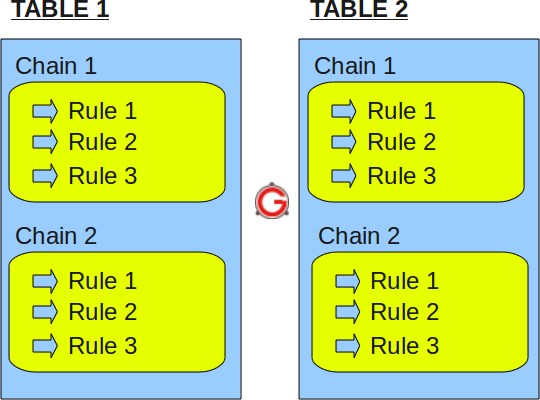
iptables tool is used to manage the Linux firewall rules. At a first look, iptables might look complex (or even confusing). But, once you understand the basics of how iptables work and how it is structured, reading and writing iptables firewall rules will be easy.

This article is part of an ongoing iptables tutorial series. This is the 1st article in that series.

This article explains how iptables is structured, and explains the fundamentals about iptables tables, chains and rules.

On a high-level iptables might contain multiple tables. Tables might contain multiple chains. Chains can be built-in or user-defined. Chains might contain multiple rules. Rules are defined for the packets.

So, the structure is: iptables -> Tables -> Chains -> Rules. This is defined in the following diagram.

  
**Fig**: IPTables Table, Chain, and Rule Structure

Just to re-iterate, tables are bunch of chains, and chains are bunch of firewall rules.

**I. IPTABLES TABLES and CHAINS**

IPTables has the following 4 built-in tables.

1. Filter Table

Filter is default table for iptables. So, if you don’t define you own table, you’ll be using filter table. Iptables’s filter table has the following built-in chains.

* INPUT chain – Incoming to firewall. For packets coming to the local server.
* OUTPUT chain – Outgoing from firewall. For packets generated locally and going out of the local server.
* FORWARD chain – Packet for another NIC on the local server. For packets routed through the local server.

2. NAT table

Iptable’s NAT table has the following built-in chains.

* PREROUTING chain – Alters packets before routing. i.e Packet translation happens immediately after the packet comes to the system (and before routing). This helps to translate the destination ip address of the packets to something that matches the routing on the local server. This is used for DNAT (destination NAT).
* POSTROUTING chain – Alters packets after routing. i.e Packet translation happens when the packets are leaving the system. This helps to translate the source ip address of the packets to something that might match the routing on the desintation server. This is used for SNAT (source NAT).
* OUTPUT chain – NAT for locally generated packets on the firewall.

3. Mangle table

Iptables’s Mangle table is for specialized packet alteration. This alters QOS bits in the TCP header. Mangle table has the following built-in chains.

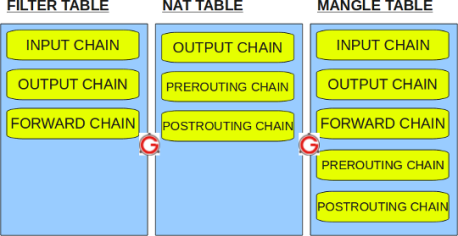
* PREROUTING chain
* OUTPUT chain
* FORWARD chain
* INPUT chain
* POSTROUTING chain

4. Raw table

Iptable’s Raw table is for configuration excemptions. Raw table has the following built-in chains.

* PREROUTING chain
* OUTPUT chain

The following diagram shows the three important tables in iptables.



**Fig**: IPTables built-in tables

**II. IPTABLES RULES**

Following are the key points to remember for the iptables rules.

* Rules contain a criteria and a target.
* If the criteria is matched, it goes to the rules specified in the target (or) executes the special values mentioned in the target.
* If the criteria is not matached, it moves on to the next rule.

Target Values

Following are the possible special values that you can specify in the target.

* ACCEPT – Firewall will accept the packet.
* DROP – Firewall will drop the packet.
* QUEUE – Firewall will pass the packet to the userspace.
* RETURN – Firewall will stop executing the next set of rules in the current chain for this packet. The control will be returned to the calling chain.

If you do iptables –list (or) service iptables status, you’ll see all the available firewall rules on your system. The following iptable example shows that there are no firewall rules defined on this system. As you see, it displays the default input table, with the default input chain, forward chain, and output chain.

# iptables -t filter --list

Chain INPUT (policy ACCEPT)

target prot opt source destination

Chain FORWARD (policy ACCEPT)

target prot opt source destination

Chain OUTPUT (policy ACCEPT)

target prot opt source destination

Do the following to view the mangle table.

# iptables -t mangle --list

Do the following to view the nat table.

# iptables -t nat --list

Do the following to view the raw table.

# iptables -t raw --list

Note: If you don’t specify the -t option, it will display the default filter table. So, both of the following commands are the same.

# iptables -t filter --list

(or)

# iptables --list

The following iptable example shows that there are some rules defined in the input, forward, and output chain of the filter table.

# iptables --list

Chain INPUT (policy ACCEPT)

num target prot opt source destination

1 RH-Firewall-1-INPUT all -- 0.0.0.0/0 0.0.0.0/0

Chain FORWARD (policy ACCEPT)

num target prot opt source destination

1 RH-Firewall-1-INPUT all -- 0.0.0.0/0 0.0.0.0/0

Chain OUTPUT (policy ACCEPT)

num target prot opt source destination

Chain RH-Firewall-1-INPUT (2 references)

num target prot opt source destination

1 ACCEPT all -- 0.0.0.0/0 0.0.0.0/0

2 ACCEPT icmp -- 0.0.0.0/0 0.0.0.0/0 icmp type 255

3 ACCEPT esp -- 0.0.0.0/0 0.0.0.0/0

4 ACCEPT ah -- 0.0.0.0/0 0.0.0.0/0

5 ACCEPT udp -- 0.0.0.0/0 224.0.0.251 udp dpt:5353

6 ACCEPT udp -- 0.0.0.0/0 0.0.0.0/0 udp dpt:631

7 ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:631

8 ACCEPT all -- 0.0.0.0/0 0.0.0.0/0 state RELATED,ESTABLISHED

9 ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 state NEW tcp dpt:22

10 REJECT all -- 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-prohibited

The rules in the iptables –list command output contains the following fields:

* num – Rule number within the particular chain
* target – Special target variable that we discussed above
* prot – Protocols. tcp, udp, icmp, etc.,
* opt – Special options for that specific rule.
* source – Source ip-address of the packet
* destination – Destination ip-address for the packet
* 5)What is the difference between IPchains and IPtables?
* **IPchains :** Ipchains is a utility for Linux that System Administrators can use to create and modify the ruleset that is used for their host based firewall. These rules are used by a system to decide whether or not it is going to allow a specific remote connection.
* **IP tables :** Iptables is a generic table structure that defines rules and commands as part of the [netfilter](http://searchenterpriselinux.techtarget.com/sDefinition/0,,sid39_gci1099982,00.html) framework that facilitates Network Address Translation ([NAT](http://searchenterprisewan.techtarget.com/sDefinition/0,,sid200_gci214107,00.html)), [packet filtering](http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci1101066,00.html), and [packet mangling](http://searchenterpriselinux.techtarget.com/sDefinition/0,,sid39_gci1101542,00.html) in the [Linux](http://searchenterpriselinux.techtarget.com/sDefinition/0,,sid39_gci212482,00.html) 2.4 and later operating systems.
* **How to disable IPTABLES??**  
  ==================  
    
  # service iptables save  
  # service iptables stop  
  # chkconfig iptables off  
    
    
  **To clear IP rules use below commands:**  
  =====================================  
    
  # iptables -F  
  # iptables -X  
  # iptables -t nat -F  
  # iptables -t nat -X  
  # iptables -t mangle -F  
  # iptables -t mangle -X  
  # iptables -P INPUT ACCEPT  
  # iptables -P OUTPUT ACCEPT  
    
  **How to Enable IPTABLES??**  
  ===================  
    
  #/etc/init.d/iptables start  
    
  #chkconfig iptables on  
    
  #iptables-save > /root/working.fw  
    
    
  **To restore Rules:**  
  =============  
    
  #iptables-restore < /root/firewall.rules  
    
  #iptables-save > /root/firewall.rules
* **To List the iptable Rules:**  
    
  # iptables --list  
    
  #iptables -L  
    
  **To delete iptable rules**  
    
  # iptables --flush  
    
  # iptables --flush OUTPUT //To delete particular CHAIN  
    
    
  **some basic Rules:  
  =============**  
    
  **Interface level:**  
    
  Allow incomming packets at interface level  
    
  iptables -A INPUT -i lo -j ACCEPT  
  iptables -A INPUT -i eth0 -j ACCEPT  
    
  # Accept packets from trusted IP addresses  
    
  iptables -A INPUT -s 192.168.0.4 -j ACCEPT # change the IP address as appropriate  
    
  # Accept packets from trusted IP addresses  
    
  iptables -A INPUT -s 192.168.0.0/24 -j ACCEPT //using standard slash notation  
  iptables -A INPUT -s 192.168.0.0/255.255.255.0 -j ACCEPT // using a subnet mask  
    
    
  # Accept tcp packets on destination port 6881 (bittorrent)  
    
  iptables -A INPUT -p tcp --dport 6881 -j ACCEPT  
    
    
  # Accept tcp packets on destination ports 6881-6890  
    
  iptables -A INPUT -p tcp --dport 6881:6890 -j ACCEPT  
    
    
  **Rules for SSH:**  
  ===========  
    
  # Accept tcp packets on destination port 22 (SSH)  
    
  iptables -A INPUT -p tcp --dport 22 -j ACCEPT  
    
    
  # Accept tcp packets on destination port 22 (SSH) from private LAN  
  iptables -A INPUT -p tcp -s 192.168.0.0/24 --dport 22 -j ACCEPT
* **What command can you use to review boot messages?**
* The **dmesg** command can be used to display the system messages during boot time.
* **What are the fields in the /etc/passwd file. Explain them?**
* There are 7 fields in /etc/passwd:
* **username:x:UID:GID:comment:home directory:shell**  
    
  1)Username.  
  2)Password. Dummy value x denotes that its using shadow passwords
* 3)UID.
* 4)Primary group ID.  
  5)Comment /description.  
  6)Home directory path.  
  7)Shell assigned to the user.
* **Whats the journaling data contains in ext3?**
* A journaled file system records information in a log area on a disk. It logs the "metadata" i.e ownership, date stamp information etc..Once the log is updated the system then writes the actual data to the appropriate areas of the filesystem and marks an entry in the log to say the data is committed.  
    
  After a crash the filesystem can very quickly be brought back on-line using the journal logs
* using fsck there is considerably less chance of data loss or corruption.
* **How do u extend the LV in Linux?**
* First check whether is there free space available in the VG where the LV resides.
* # vgdisplay or
* # vgs
* Now, extend the LV using below command: "L" option to specify the size to be increased.
* # lvextend -L +5G /dev/vg00/lv01
* Finally, extend the File system space:
* # resize2fs /dev/vg00/lv01
* That's all..! Look at the other posts as well for interview questions. I am posting 4-5 questions in every post whenever i get time.

# How to Log Linux IPTables Firewall Dropped Packets to a Log File

This article is part of our ongoing Linux IPTables series of articles. When things are not working as expected with your IPTables rules, you might want to log the IPTables dropped packets for troubleshooting purpose. This article explains how to log both incoming and outgoing dropped firewal packets.  
  
If you are new to IPTables, first get yourself comfortable with the [IPTables fundamental concepts](http://www.thegeekstuff.com/2011/01/iptables-fundamentals/).

### Log All Dropped Input Packets

First we need to understand how to log all the dropped input packets of iptables to syslog.

If you already have whole bunch of iptables firewall rules, add these at the bottom, which will log all the dropped input packets (incoming) to the /var/log/messages

iptables -N LOGGING

iptables -A INPUT -j LOGGING

iptables -A LOGGING -m limit --limit 2/min -j LOG --log-prefix "IPTables-Dropped: " --log-level 4

iptables -A LOGGING -j DROP

In the above example, it does the following:

* iptables -N LOGGING: Create a new chain called LOGGING
* iptables -A INPUT -j LOGGING: All the remaining incoming packets will jump to the LOGGING chain
* line#3: Log the incoming packets to syslog (/var/log/messages). This line is explained below in detail.
* iptables -A LOGGING -j DROP: Finally, drop all the packets that came to the LOGGING chain. i.e now it really drops the incoming packets.

In the line#3 above, it has the following options for logging the dropped packets:

* -m limit: This uses the limit matching module. Using this you can limit the logging using –limit option.
* –limit 2/min: This indicates the maximum average matching rate for logging. In this example, for the similar packets it will limit logging to 2 per minute. You can also specify 2/second, 2/minute, 2/hour, 2/day. This is helpful when you don’t want to clutter your log messages with repeated messages of the same dropped packets.
* -j LOG: This indicates that the target for this packet is LOG. i.e write to the log file.
* –log-prefix “IPTables-Dropped: ” You can specify any log prefix, which will be appended to the log messages that will be written to the /var/log/messages file
* –log-level 4 This is the standard syslog levels. 4 is warning. You can use number from the range 0 through 7. 0 is emergency and 7 is debug.

### Log All Dropped Outgoing Packets

This is same as above, but the 2nd line below has OUTPUT instead of INPUT.

iptables -N LOGGING

iptables -A OUTPUT -j LOGGING

iptables -A LOGGING -m limit --limit 2/min -j LOG --log-prefix "IPTables-Dropped: " --log-level 4

iptables -A LOGGING -j DROP

### Log All Dropped Packets (both Incoming and Outgoing)

This is same as before, but we’ll be taking the line number 2 from the previous two examples, and adding it here. i.e We’ll have a separate line for INPUT and OUTPUT which will jump to LOGGING chain.

To log both the incoming and outgoing dropped packets, add the following lines at the bottom of your existing iptables firewall rules.

iptables -N LOGGING

iptables -A INPUT -j LOGGING

iptables -A OUTPUT -j LOGGING

iptables -A LOGGING -m limit --limit 2/min -j LOG --log-prefix "IPTables-Dropped: " --log-level 4

iptables -A LOGGING -j DROP

Also, as we explained earlier, by default, the iptables will use /var/log/messages to log all the message. If you want to change this to your own custom log file add the following line to /etc/syslog.conf

kern.warning /var/log/custom.log

How to read the IPTables Log

The following is a sample of the lines that was logged in the /var/log/messages when an incoming and outgoing packets was dropped.

Aug 4 13:22:40 centos kernel: IPTables-Dropped: IN= OUT=em1 SRC=192.168.1.23 DST=192.168.1.20 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=0 DF PROTO=ICMP TYPE=8 CODE=0 ID=59228 SEQ=2

Aug 4 13:23:00 centos kernel: IPTables-Dropped: IN=em1 OUT= MAC=a2:be:d2:ab:11:af:e2:f2:00:00 SRC=192.168.2.115 DST=192.168.1.23 LEN=52 TOS=0x00 PREC=0x00 TTL=127 ID=9434 DF PROTO=TCP SPT=58428 DPT=443 WINDOW=8192 RES=0x00 SYN URGP=0

In the above output:

* IPTables-Dropped: This is the prefix that we used in our logging by specifying –log-prefix option
* IN=em1 This indicates the interface that was used for this incoming packets. This will be empty for outgoing packets
* OUT=em1 This indicates the interface that was used for outgoing packets. This will be empty for incoming packets.
* SRC= The source ip-address from where the packet originated
* DST= The destination ip-address where the packets was sent to
* LEN= Length of the packet
* PROTO= Indicates the protocol (as you see above, the 1st line is for outgoing ICMP protocol, the 2nd line is for incoming TCP protocol)
* SPT= Indicates the source port
* DPT= Indicates the destination port. In the 2nd line above, the destination port is 443. This indicates that the incoming HTTPS packets was dropped

### [setup NAT using iptables in Linux : How to ??](http://www.sysadminshare.com/2011/09/setup-nat-using-iptables-in-linux-how.html)

11:22 PM Posted byMadeswaran N

**Step #1. Add 2 Network cards to the Linux server**  
  
**Step #2. Verify the Network cards, Whether they installed properly or not**  
  
ls /etc/sysconfig/network-scripts/ifcfg-eth\* | wc -l  
  
( The output should be "2")  
  
**Step #3. Configure eth0 for Internet with a Public ( IP External network or Internet)**  
  
cat /etc/sysconfig/network-scripts/ifcfg-eth0  
  
DEVICE=eth0  
BOOTPROTO=none  
BROADCAST=xx.xx.xx.255 # Optional Entry  
HWADDR=00:50:BA:88:72:D4 # Optional Entry  
IPADDR=xx.xx.xx.xx  
NETMASK=255.255.255.0 # Provided by the ISP  
NETWORK=xx.xx.xx.0 # Optional  
ONBOOT=yes  
TYPE=Ethernet  
USERCTL=no  
IPV6INIT=no  
PEERDNS=yes  
GATEWAY=xx.xx.xx.1 # Provided by the ISP  
  
**Step #4. Configure eth1 for LAN with a Private IP (Internal private network)**  
  
cat /etc/sysconfig/network-scripts/ifcfg-eth1  
  
BOOTPROTO=none  
PEERDNS=yes  
HWADDR=00:50:8B:CF:9C:05 # Optional  
TYPE=Ethernet  
IPV6INIT=no  
DEVICE=eth1  
NETMASK=255.255.0.0 # Specify based on your requirement  
BROADCAST=""  
IPADDR=192.168.2.1 # Gateway of the LAN  
NETWORK=192.168.0.0 # Optional  
USERCTL=no  
ONBOOT=yes  
  
**Step #5. Host Configuration** (Optional)  
  
cat /etc/hosts  
  
127.0.0.1 nat localhost.localdomain localhost  
  
Step #6. Gateway Configuration  
  
cat /etc/sysconfig/network  
  
NETWORKING=yes  
HOSTNAME=nat  
GATEWAY=xx.xx.xx.1  
  
 **Step #7. DNS Configuration**  
  
cat /etc/resolv.conf  
  
nameserver 203.145.184.13 # Primary DNS Server provided by the ISP  
nameserver 202.56.250.5 # Secondary DNS Server provided by the ISP  
  
**Step #8. NAT configuration with IP Tables**  
  
# Delete and flush. Default table is "filter". Others like "nat" must be explicitly stated.  
  
iptables --flush # Flush all the rules in filter and nat tables  
  
iptables --table nat --flush  
  
iptables --delete-chain  
  
# Delete all chains that are not in default filter and nat table  
  
iptables --table nat --delete-chain  
  
# Set up IP FORWARDing and Masquerading  
  
iptables --table nat --append POSTROUTING --out-interface eth0 -j MASQUERADE  
  
iptables --append FORWARD --in-interface eth1 -j ACCEPT  
  
# Enables packet forwarding by kernel   
  
echo 1 > /proc/sys/net/ipv4/ip\_forward  
  
#Apply the configuration  
  
service iptables restart  
  
**Step #9. Test the config**

# Ping the Gateway of the network from client system  
  
ping 192.168.2.1  
  
Try it on your client systems  
  
ping yahoo.com