IPTables

iptables firewall is used to manage packet filtering and NAT rules. IPTables comes with all Linux distributions. By configuring the IPTables, we can manage the firewall rules.

# Structure of IPTables:

The structure is: iptables -> Tables -> Chains -> Rules.

|  |  |  |  |
| --- | --- | --- | --- |
| Table | Table Function | Chain | Chain Function |
| Filter | Packet filtering(Default) | INPUT | Incoming to firewall. For packets coming to the local server. |
|  |  | OUTPUT | Filters packets originating from the firewall |
|  |  | FORWARD | Packet for another NIC on the local server. For packets routed through the local server. |
| Nat | Network Address Translation | PREROUTING | Packets will enter this chain before a routing decision is made. |
|  |  | POSTROUTING | Routing decision has been made. Packets enter this chain just before handing them off to the hardware. |
|  |  | OUTPUT | NAT for locally generated packets on the firewall. |
| Mangle | TCP header modification | PREROUTING, POSTROUTING, OUTPUT, INPUT, FORWARD | Modification of the TCP packet quality of service bits before routing occurs. |

## 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
* 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.

## IPTables commands

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

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

# 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

# IPTables Rules understanding:

## Rule-38444

IPV6 must implement DENY-ALL, ALLOW-BY-EXCEPTION Policy for inbound packets

Steps:

Check if TPV6 is enabled or not

and

Check deny-all policy for inbound packets

Fixrule:

Set the DENY-ALL POLICY

## Rule-38513

Same as above but for IPV4

## Rule-38546

Must not be bound to any network stack

Steps:

Check if IPV6 is enabled or not

Check the server status and/or modprobe to verify if its bound to any network stack

## Rule-38513

Check the IPV4 firewell status

## Rule-38549

Same as above but for IPV6 firewall status

## Rule-38686

Local firwall must implement the DENY-ALL, ALLOW-BY-EXCEPTION Policy

## Rule-38512

The operating system must prevent public IPv4 access into an organizations internal networks, except as appropriately mediated by managed interfaces employing boundary protection devices.

Rule for Incomming Public IPV4 access:

1) First verify the firewall is active,

2) Verify the IPV4 is enabled and deny all policy is set by default for Public IPV4

3) Configure a set of interfaces(SNAT the public IP to private IP) to allow incomming traffic,

3.1) Check if IP forwarding is enabled or not

Enable IP forwarding

The first is easy, check the contents of /proc/sys/net/ipv4/ip\_forward

cat /proc/sys/net/ipv4/ip\_forward

An output of 0 indicates IP forwarding is disabled, a value of 1 indicates IP forwarding is enabled.

If it is disabled, edit /etc/sysctl.conf with a text editor and change the value of net.ipv4.ip\_forward to 1. After making the change, do a

/etc/init.d/network restart

iptables -t nat -A POSTROUTING -s 192.168.1.101/32 -j SNAT –to-source 10.10.10.99

service iptables save

4) Verify the traffic is accepted from the whitelisted Public IP'S and blocked on other

## Rule-38553

Same as above, but for IPV6

## Rule-38551

The operating system must connect to external networks or information systems only through managed IPv6 interfaces consisting of boundary protection devices arranged in accordance with an organizational security architecture

This is rule for Outgoing Public IPV6 access:

1) First verify the firewall is active,

2) Configure a set of IP's for out going traffic,

sudo iptables -A OUTPUT -d google.com -j ACCEPT

sudo iptables -A OUTPUT -d facebook.com -j DROP

3) Verify the traffic is sent only on the configured IP's and blocked on other interfaces, 5) Verfiy that the rule is applied according to the order mentioned ((i.e if first its set to DROP and then ACCEPT, it should DROP by default)

ping google.com

ping facebook.com

## Rule-38560

Same as above but for IPV4

## Steps to run the IPTables test in local:

Goto the raxak code directory.

>> cd ~/RAXAK/Server1/raxak1

Start the localhost cloudraxak server

$ ./gstart.sh -f

rm: cannot remove ‘/tmp/gunicorn.pid’: No such file or directory

[2016-07-20 09:35:50 +0000] [5017] [INFO] Starting gunicorn 19.3.0

[2016-07-20 09:35:50 +0000] [5017] [INFO] Listening at: http://0.0.0.0:8080 (5017)

[2016-07-20 09:35:50 +0000] [5017] [INFO] Using worker: sync

[2016-07-20 09:35:50 +0000] [5022] [INFO] Booting worker with pid: 5022

Rules loaded from an xml file = U\_Ubuntu\_6\_V1R2\_Manual-xccdf.xml

Profile: DISA - Mission Critical Classified

Profile: PCI-DSS Compliance

Profile: Demonstration Profile

Profile: Undo Demonstration Profile

Profile: Boot-TXT for OpenStack Controller

Profile: PCI for OpenStack Controller

Rules loaded from an xml file = U\_RedHat\_6\_V1R2\_Manual-xccdf.xml

Profile: DISA - Mission Critical Classified

Profile: PCI-DSS Compliance

Profile: Demonstration Profile

Profile: Undo Demonstration Profile

Profile: Boot-TXT for OpenStack Controller

Profile: PCI for OpenStack Controller

Rules loaded from an xml file = U\_Apache\_2.2\_Unix\_V1R2\_Manual-xccdf.xml

Profile: I - Mission Critical Classified

Profile: Demonstration Profile

Profile: Reset to Factory Default

Profile: I - Mission Critical Public

Profile: I - Mission Critical Sensitive

Profile: II - Mission Support Classified

Profile: II - Mission Support Public

Profile: II - Mission Support Sensitive

Profile: III - Administrative Classified

Profile: III - Administrative Public

Profile: III - Administrative Sensitive

Merging lastruninfo from old to new key structure

Authorized users of HP Demo['seshmurthy@cloudraxak.com', 'prasanna@cloudraxak.com']

Initializing flask, setting key

Cleaning up residual tunnels if any:

Valid servers ['cloud-raxak-server']

MWPYT106 is not a known server: No user validation

cloudraxak: enroll: GTestUser:{'ip': '0.0.0.0', 'login': 'Local Auth', 'user': 'GTestUser', 'firstname': 'GTestUser', 'email': 'GTestUser@cloudraxak.com'}

cloudraxak: enroll: Overwriting GTestUser with {"ip": "0.0.0.0", "login": "Local Auth", "user": "GTestUser", "firstname": "GTestUser", "email": "GTestUser@cloudraxak.com"}

cloudraxak: enroll: Returning OK

cloudraxak: checking user existance

Email ID - raxak@cloudraxak.com already registered.

Running version: debian/b47fb5d

In another terminal, run the test

./testRule.sh -c sneha@192.168.0.131 38444

Following is the sample Output for V-38444 check rule and fix rule.

Console : Remote: executing checkRule38444

Console : Checking if IPv6 module is enabled.

Console : # sysctl -a | grep 'disable\_ipv6 = 0'

Console : > net.ipv6.conf.all.disable\_ipv6 = 0

Console : > net.ipv6.conf.br0.disable\_ipv6 = 0

Console : > net.ipv6.conf.default.disable\_ipv6 = 0

Console : > net.ipv6.conf.lo.disable\_ipv6 = 0

Console : > net.ipv6.conf.vnet0.disable\_ipv6 = 0

Console : IPV6 module is enabled, need to enable firewall.

Console : Checking the deny-all policy for inbound packets.

Console : # ip6tables -L INPUT | head -1

Console : > Chain INPUT (policy ACCEPT)

Console : IPV6 firewall is not configured to deny-all policy for inbounds packets,

hence it is recommended to fix it manually.

Console : Remote: finished executing

Console : Rule: V-38444 needs manual intervention

Console : =================================================

closeLink: sneha@192.168.0.131 needs a tunnel. Closing any open routes

cloudraxak: [2, None, None]

wsgi: 2

V-38444 FIXRULE

Console : Remote: executing fixRule38444

Console : Please fix it manually.

Console : Remote: finished executing

Console : Re-verifying remediation status of Rule:

Console : Remote: executing checkRule38444

Console : Checking if IPv6 module is enabled.

Console : # sysctl -a | grep 'disable\_ipv6 = 0'

Console : > net.ipv6.conf.all.disable\_ipv6 = 0

Console : > net.ipv6.conf.br0.disable\_ipv6 = 0

Console : > net.ipv6.conf.default.disable\_ipv6 = 0

Console : > net.ipv6.conf.lo.disable\_ipv6 = 0

Console : > net.ipv6.conf.vnet0.disable\_ipv6 = 0

Console : IPV6 module is enabled, need to enable firewall.

Console : Checking the deny-all policy for inbound packets.

Console : # ip6tables -L INPUT | head -1

Console : > Chain INPUT (policy ACCEPT)

Console : IPV6 firewall is not configured to deny-all policy for inbounds packets,

hence it is recommended to fix it manually.

Console : Remote: finished executing

Console : Rule: checkRule38444 needs manual intervention

Console : =================================================

closeLink: sneha@192.168.0.131 needs a tunnel. Closing any open routes

cloudraxak: [2, None, None]