How to make NAT enable hosts in a local network to access the internet, create port forwarding to local IPs behind the router using iptables

Author : admin

I'm bulding new iptables firewall on one Linux server. The Debian GNU/Linux is required to act as firewall do Network Adress Translation for a small network of office PCs as well as forward some of the inbound ports to hosts from the local network located behind the router.

The local network besides the router had an IP addressing in the class C network e.g. (192.168.1.1-255)

First I procceded and enabled the **Network Address Translation** via the Linux kernel variable:

```
linux:~# sysctl -w net.ipv4.ip_forward=1
linux:~# echo 'net.ipv4.ip_forward=1' >> /etc/sysctl.conf
```

Initially I even forgot to switch on the **net.ipv4.ip_forward to 1** (by default this value is set to 0) - GNU/Linux's default network behaviour is not predetermined to act as network router.
However, since I haven't configured Network Address Translation for quite some time it completely slipped my mind!

Anyways next the actual iptables rule which makes *NAT* possible I used is:

```
linux:~# /sbin/iptables -t nat -A POSTROUTING -s 192.168.1.0/24 ! -d 192.168.1.0/24 -j SNAT --to-source xxx.xxx.xxx.xxx
```

Whether **xxx.xxx.xxx.xxx** is the External IP address assigned to the router on *eth0*

With this very simple rules now Network the local network is capable of accessing the Internet withotu problem.

It's a good time to say that still many system administrators, still erroneously use **MASQUERADE** rules instead of **SNAT** .
IP MASQUERADING is an ancestry from ipchains and these days should be completely abandonded, especially where no often change of primary IP address to access the internet is made.
For dial-ups or other kind of networking, where the IP addresses are often changed still IP MASQUERADING might be a good idea though.

My next goal was to make the Linux router to do port forwarding of Traffic which arrives on **port 80** to a IIS server assigned with a local IP address of **192.168.1.5**
I did the webserver (port 80), port forwarding from IP **xxx.xxx.xxx.xxx** to **192.168.1.5** with the iptables

rule:

```
linux:~# /sbin/iptables -t nat -A PREROUTING -d xxx.xxx.xxx.xxx/32 -p tcp -m tcp --dport 80 -j DNAT
--to-destination 192.168.1.5:80
```

There was a requirement to do port forwarding for a Windows remote Desktop running on standard port 3389 from the router to the internal Windows IP address running the IIS webserver, however the company required me to only allow access to the *rdesktop* 3389 port to certain real IP addresses.
Initially I thought about using the above **PREROUTING** rule which makes the port redirection to the IIS server and only change *port 80* to *port 3389* , and then use **filter** table INPUT chain rules like:

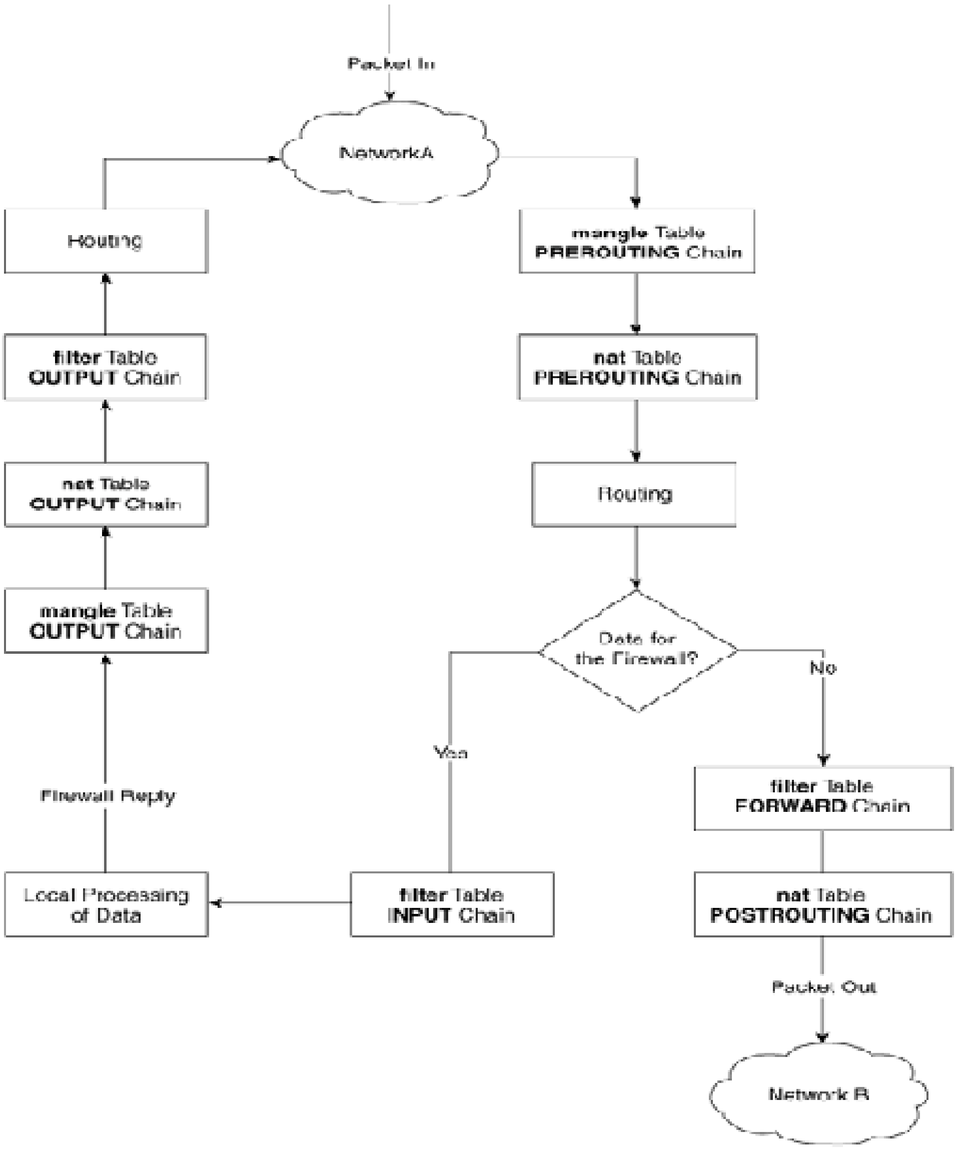
```
/sbin/iptables -A INPUT -s xx1.xx2.xx3.xx4,1xx,2xx,3xx,4xx,xxx.xxx.xxx.xxx -p tcp -m tcp --dport 3389
-j ACCEPT/sbin/iptables -A INPUT -p tcp -m tcp --dport 3389 -j REJECT --reject-with icmp-port-
unreachable
32
```

However this did not work out, so I decided to give a try to do the same within the filter table using the FORWARD chain, like so:

```
FORWARD/sbin/iptables -A FORWARD -p tcp -m tcp -s
xx1.xx2.xx3.xx4,1xx,2xx,3xx,4xx,xxx.xxx.xxx.xxx -p tcp -m tcp --dport 3389 -j ACCEPT
/sbin/iptables -A FORWARD -p tcp -m tcp --dport 3389 -j REJECT --reject-with icmp-port-unreachable
```

Adding this rules did not added any filtering to the forwarded remote desktop port. I suspected that somehow probably my above PREROUTING nat rules are read before any other rules and therefore automatically allows any IP address to port forward traffic.
I've checked the iptables documentation and it seems my guess was partially right.

When some kind of network traffic enters the iptables firewall it first goes through the PREROUTING channel and then the traffic flows in a certain order.



The iptables network packets flow is clearly seen in above's diagram a thorough looks gives a very good idea on how packet is being processed by iptables

Finally as I couldn't think about a good solution on how to only filter the port redirected traffic, which always firstly entered in the POSTROUTING chain, I've consulted with the guys in irc.freenode.net in #Netfilter.

I'm quite thanksful as a guy nicknamed Olipro has given me a pretty good picture on the port forwarding POSTROUTING problem and has provided me with a very logical easy and great fix.
He suggested that I only do port forwarding for certain IP addresses instead of allowing all IP addresses and then lookup for a way to allow only some of them and filter the rest.

The iptables rule to restrict the incoming traffic to the remote desktop forwarded port 3389 to few only allowed IP addresses looks like so:

```
linux:~# /sbin/iptables -t nat -A PREROUTING -d xxx.xxx.xxx.xxx/32 -s
xx1.xx2.xx3.xx4,1xx,2xx,3xx,4xx,xxx.xxx.xxx.xxx -p tcp -m tcp --dport 3389 -j DNAT --to-destination
192.168.1.5:3389
```

Now the three sample IPs passed *xx1.xx2.xx3.xx4,1xx,2xx,3xx,4xx,xxx.xxx.xxx.xxx* has added to port forward traffic on 3389 to *192.168.1.5*

By the way I did not know that newer versions of iptables support passing by multiple IP addresses to the --source or --destination IP. This is really great feature I've learned from the good guys from #Netfilter. However one should be careful when using the multiple IPs with **-s** or **-d**, it's really important that the passed consequent IPs has no space between the , delimiter.

Now that's all my task is completed. All computerse inside the Network 192.168.1.1-255 on the Linux router freely can access the Internet, all IPs are also capable to access the IIS server located behind the NAT as well as only certain IPs are capable of accessing to the IIS remote desktop.
Hope the article helps somebody ;)