Packet Flow control using IP tables

On Flow Rate

Using iptables, we could control the incoming packets. The following command (Rule 1) is to limit the echo request in a period of time. The limit in the command is one echo request per second (--*limit 1/s*, we can also write *1/m* for one per minute).

The concept of credit is there. The credit means maintaining the number of violation. If the violation is more than the threshold then action will be taken. The threshold here is the burst (--*limit-burst*).

If the rule 1 satisfies for the incoming packets then it is accepted otherwise what is the action to be taken. Hence, we have to write the second rule that is drop the packets.

Rule 1: *sudo iptables -I INPUT -p icmp --icmp-type echo-request -m limit --limit 1/s --limit-burst 2 -j ACCEPT*

Rule 2: *sudo iptables -I INPUT -p icmp --icmp-type echo-request -j DROP [At first this rule has to be inserted]*

Important Condition 1: The rule order should be maintained that is first is the accept rule and the second is the drop rule.

The iptables entry can be validated using the *iptables -L* command.

```
sudo iptables -L
Chain INPUT (policy ACCEPT)
target      prot   opt    source          destination
ACCEPT     icmp    --     anywhere        anywhere        icmp echo-request limit: avg 1/min burst 5
DROP       icmp    --     anywhere        anywhere        icmp echo-request

Chain FORWARD (policy ACCEPT)
target    prot opt source           destination

Chain OUTPUT (policy ACCEPT)
target    prot opt source           destination
```

Important Condition 2: These rules should be part of INPUT chain otherwise the rules will not work for the incoming packets. If required, we can create the new chain but that should be linked through INPUT chain.

After creating the rule, try to ping the respective machine with the following.

ping -i 0.3 172.26.23.224

This pings the machine with the interval of 0.3 seconds that is approximately 3 ping request per second but the rule says (*--limit 1/s)* one request per second. --> This is the violation however, the violation is permitted twice (*--limit-burst 2)*. Afterwards, packets will be dropped without any notice to the client.

To see the packet flow, *tcpdump* with the command *tcpdump -n icmp* or wireshark can be used.

If client has to be informed then instead of DROP put REJECT in second rule. You can update the rule using replace (*R*) as follows. The value 2 indicates the rule number in the chain.

```
sudo iptables -R INPUT 2 -p icmp --icmp-type echo-request -j REJECT
```

You can notice the change in iptables as follows.

```
sudo iptables -L
Chain INPUT (policy ACCEPT)
target      prot   opt   source          destination
ACCEPT     icmp   --    anywhere        anywhere       icmp echo-request limit: avg 1/min burst 5
DROP       icmp   --    anywhere        anywhere       icmp echo-request reject-with icmp-port-
unreachable

Chain FORWARD (policy ACCEPT)
target    prot opt source              destination

Chain OUTPUT (policy ACCEPT)
target    prot opt source              destination
```

The rate control can be done for specific IP(s) and protocols. This can be done on the same machine and the wireshark or tcpdump (*tcpdump -i lo -n icmp)* can be run on the loopback interface.

On Packet Count

Using iptables, we can block the request based on number of requests. For example, in the following we block the IPs, if it send the ICMP request more than the limit that is 2 in a second. Here, there are four rules given and it can be done with three rules also. Use sudo while running this command.

```
sudo iptables -N ICMPSCAN
sudo iptables -I INPUT -p icmp -m icmp --icmp-type echo-request -j ICMPSCAN
sudo iptables -A ICMPSCAN -m recent --set --name badicmp --rsource
sudo iptables -A ICMPSCAN -m recent --update --seconds 1 --hitcount 2 --name badicmp --rsource
-j DROP
```

Rule 1: sudo *iptables -N ICMPSCAN*

It is the creation of the new chain ICMPSCAN. We may use the INPUT chain instead of creating the new chain however if the number of rules are more in the chain then it helps for segregation and better interpretation.

Rule 2: *sudo iptables -I INPUT -p icmp -m icmp --icmp-type echo-request -j ICMPSCAN*

It says that if the request is of icmp request then jump to the ICMPSCAN chain.

Rule 3: *sudo iptables -A ICMPSCAN -m recent --set --name badicmp --rsource*

It creates a name set that is badicmp for the remote source.

Rule 4: *sudo iptables -A ICMPSCAN -m recent --update --seconds 1 --hitcount 2 --name badicmp --rsource -j DROP*

It updates and take the action as drop if any violation. The *--hitcount* checks the value of an internal counter which is incremented by one for each hit.

After running the above commands, the iptables will look like the following.

```
sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ICMPSCAN   icmp --  anywhere             anywhere             icmp echo-request

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain ICMPSCAN (1 references)
target   prot opt source      destination
         all  --  anywhere    anywhere    recent: SET name: badicmp side: source mask:
255.255.255.255
REJECT   all  --  anywhere    anywhere    recent: UPDATE seconds: 1 hit_count: 2 name: badicmp
side: source mask: 255.255.255.255 reject-with icmp-port-unreachable
```

The mask in the above defines any IP address. The above rules block the IP address if it violates the rules.

To delete the iptables rules, use the following command with the options: *D* for delete, chain name and rule number in the table.

```
iptables -D INPUT 1
```