Protect Linux computers with Iptables

Linux is one of the operating systems that runs faster and has a reputation for more security functions than other operating systems, but that doesn't mean it can be completely secure. So the best way is to do some other security methods. Our suggestion is to use a firewall. There are several firewall options in Linux, but here we want to mention Iptables.

Linux is one of the operating systems that runs faster and has a reputation for more security functions than other operating systems, but that doesn't mean it can be completely secure. So the best way is to do some other security methods. Our suggestion is to use a firewall. There are several firewall options in Linux, but here we want to mention the Iptables firewall.

1. Lock an IP address in the IPTable
2. What is a firewall? General knowledge about Firewall

So what is Iptables?

Iptables is a Linux firewall that is configured and operates on a very small and handy Console platform. It comes with all Linux distributions and is the most direct way to control incoming and outgoing traffic from your computer.

Iptables is notoriously complicated, however, you don't need to know everything about iptables to use it effectively on your computer. You only need to know some basic knowledge about how it works and its structure rules.

Command structure

All iptables rules have the same basic structure. Each rule is a single command for iptables - showing how to handle traffic on a specific port. See the example below:

-A INPUT -i eth0 -p tc -m state --state ESTABLISHED, RELATED --sport 80 -j ACCEPT

It looks complicated but is actually very simple when you split it up. First, this rule starts with **-A** because it will connect to your iptables firewall rules.

Next, **-i** determine the interface of the rules. In this case, that's **eth0** . When you write your own rules, make sure you know the interface you are connecting to.

Next, **-p** name the protocol. This rule is for tcp - this is web traffic.

-m is a little different. It is used to confirm that a condition needs to be met so that traffic is not rejected. The condition in this rule is **state.**

Next is **--state** . You need to provide --state for a list of accepted states, all in uppercase and separated by commas. This rule accepts both new connections and established connections.

--sport is an acronym for 'source port', and it tells iptables the origin of the **traffic.In** addition, **--dport** stands for 'destination port'. It is used for OUTPUT rules to handle incoming port traffic.

Finally, **-j** . It lets iptables know the action to "jump" to. In this case, ACCEPT traffic needs to meet the previous conditions.

Use File

You can enter the rules in turn into iptables. However, this is easy to lose track of your position and what you are doing. It is better to create a rule file that you can import into iptables at the same time.

Create your file. This tutorial will use **/ tmp / iptables-ip4** . In the file, add the following two lines. All your rules will lie between them.

*** filter**

 # Your Rules Here

COMMIT

Create your rules

You can start setting up your rules. These are just suggestions. Obviously, if you are running other services or need to open other ports, you can certainly customize some things or add some of your own rules.

Loopback

Loopback interface is an internal interface that Linux uses.

-A INPUT -i lo -j ACCEPT
-A OUTPUT -o lo -j ACCEPT

Ping

Many people do not want to allow pings on their computers. However, it is quite useful to check connections. If you want to allow pings, add the rules below.

-A INPUT -i eth0 -p icmp -m state --state NEW --icmp-type 8 -j ACCEPT
-A INPUT -i eth0 -p icmp -m state --state ESTABLISHED, RELATED -j ACCEPT
-A OUTPUT -o eth0 -p icmp -j ACCEPT

Web

You can connect to the web. On the other hand, you don't want to allow connections originating from the Internet.

-A INPUT -i eth0 -p tc -m state --state ESTABLISHED, RELATED --sport 80 -j ACCEPT
-A INPUT -i eth0 -p tc -m state --state ESTABLISHED, RELATED --sport 443 -j ACCEPT

-A OUTPUT -o eth0 -p tcp -m tcp --dport 80 -j ACCEPT
-A OUTPUT -o eth0 -p tcp -m tcp --dport 443 -j ACCEPT

You will need to allow DNS connectivity so that the computer can use the URL instead of the IP address because it is not very convenient. Replace the router's IP address for the IP address used below.

-A INPUT -i ens3 -s 192.168.1.1 -p udp --sport 53 -m state --state ESTABLISHED, RELATED -j ACCEPT
-A OUTPUT -o ens3 -d 192.168.1.1 -p udp --dport 53 -m udp -j ACCEPT

Time

Most Linux desktops use NTP to set up and maintain system time from the Internet. You need to allow the computer to connect to the NTP server to set the time.

-A INPUT -i eth0 -p udp -m state --state ESTABLISHED, RELATED --dport 123 -j ACCEPT
-A OUTPUT -o eth0 -p udp -m udp --sport 123 -j ACCEPT

Print

Unless you are using a USB printer or an external print server, then you need to enable connectivity with CUPS.

-A INPUT -p udp -m udp --dport 631 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 631 -j ACCEPT
-A OUTPUT -p udp -m udp --sport 631 -j ACCEPT
-A OUTPUT -p tcp -m tcp --sport 631 -j ACCEPT

Email

You can also send and receive emails. The ports allowed here are SSL email ports. If you need to use unsafe email, replace those ports.

 # IMAP
-A INPUT -i eth0 -p tc -m state --state ESTABLISHED, RELATED --sport 993 -j ACCEPT
-A OUTPUT -o eth0 -p tcp -m tcp --dport 993 -j ACCEPT

 # POP3
-A INPUT -i eth0 -p tc -m state --state ESTABLISHED, RELATED --sport 995 -j ACCEPT
-A OUTPUT -o eth0 -p tcp -m tcp --dport 995 -j ACCEPT

 # SMTP
-A INPUT -i eth0 -p tc -m state --state ESTABLISHED, RELATED --sport 465 -j ACCEPT
-A OUTPUT -o eth0 -p tcp -m tcp --dport 465 -j ACCEPT

SSH

To make full use of SSH connections, you need to enable both output and input via SSH.

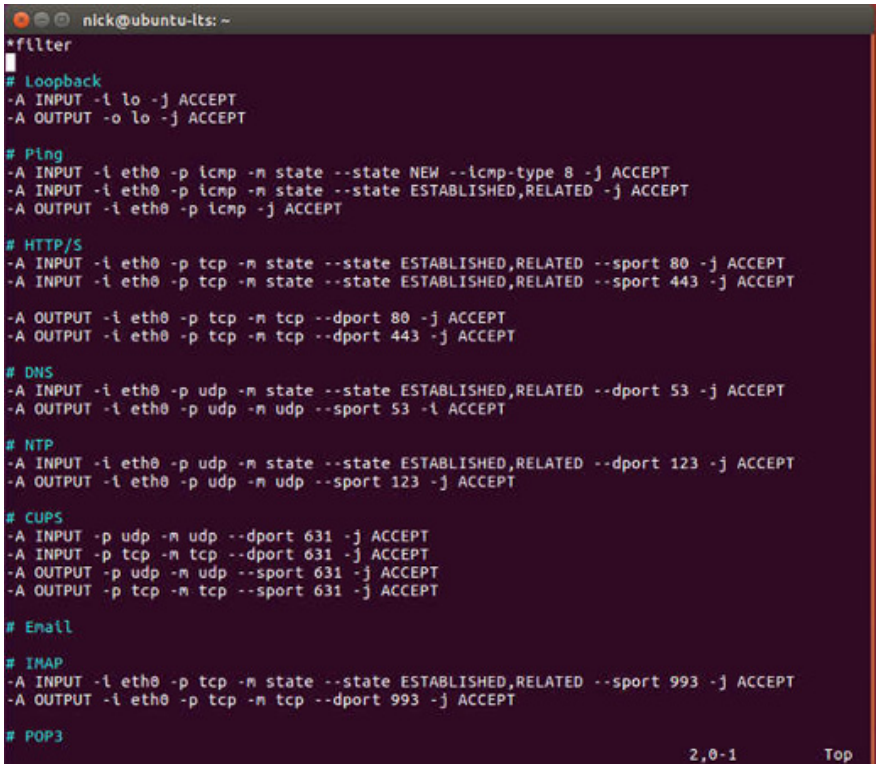
 # Input
-A INPUT -i ens3 -p tcp -m state --state NEW, ESTABLISHED --dport 22 -j ACCEPT
-A OUTPUT -o ens3 -p tcp -m state --state ESTABLISHED --sport 22 -j ACCEPT
 # Output
-A OUTPUT -o ens3 -p tcp -m state --state NEW, ESTABLISHED --dport 22 -j ACCEPT
-A INPUT -i ens3 -p tcp -m state --state ESTABLISHED --sport 22 -j ACCEPT

DHCP

Most Linux computers use DHCP to automatically receive IP addresses from a router. DHCP uses private ports, so they need to be accessed. If you are using static IP, you will not need these rules.

-A INPUT -i eth0 -p udp -m state -state ESTABLISHED, RELATED --sport 67:68 -j ACCEPT
-A OUTPUT -o eth0 -p udp -m udp --dport 67:68 -j ACCEPT

After all, your rules will look like this:



Enter the rules

Now, you already have a list of fully functional iptables rules. You just need to include it in iptables to use. In case some rules have been added over time, delete them. After the commands below, you will see the default settings that allow everything.

sudo iptables -F && sudo iptables -X

Your computer is now using new iptables rules. You can check by entering the command.

sudo iptables -S

However, these rules are only temporary. When you restart the computer, they will disappear.

How to create permanent rules

There are several ways to make these rules permanent. This tutorial will focus on systems based on Debian and Ubuntu because they are the most popular.

There is an available package, iptables-persistant - that handles the storage and restoration of iptables. All you need to do is install it.

During the installation process, this package will ask if you want to save your configuration, select **Yes.**

If you later want to add rules, you can save them by running the following commands:

sudo service netfilter-persistent save

You are controlling traffic through your computer. You can do more with iptables, but try these basic operations!