Basic configuration for iptables (suitable for Raspberry Pi)

dmcostello.co.uk/2016/09/25/basic-configuration-for-iptables-suitable-for-raspberry-pi/

@davidcostello42 25th September 2016

This is just a quick post for my own personal benefit to remind me what commands to use when setting up iptables rules.

If you want to configure a firewall on your Raspbian instance of your Raspberry Pi, iptables is a good starting point.

```
~# sudo apt-get install iptables nano
```

This will install iptables and nano if it isn't available already (on Raspbian Nano should already be installed).

Create a basic rule set

Once installed, do the following:

```
~# sudo nano /etc/iptables.rules
```

This will start the Nano editor with a new blank file. Copy the following text into the editor:

```
*filter
# Allow all loopback (lo0) traffic and drop all traffic to 127/8 that doesn't
use lo0
-A INPUT -i lo -j ACCEPT
-A INPUT -d 127.0.0.0/8 -j REJECT

# Accept all established inbound connections
-A INPUT -m state –state ESTABLISHED,RELATED -j ACCEPT

# Allow all outbound traffic – you can modify this to only allow certain
traffic
-A OUTPUT -j ACCEPT

# === PLACE ADDITIONAL RULE BLOCKS HERE === #

# ========= #

# Allow SSH connections from ANYWHERE (even the outside world!)
# IMPORTANT: check the –dport number matches what you use. on Rasbian this
should be 22.
# If you get this wrong, you won't be able to connect via SSH!
-A INPUT -p tcp -m state –state NEW –dport 22 -j ACCEPT
```

```
# Allow SSH connections only on LAN (outside world connections refused)
# Note: To use this, delete/comment the '-A' line above and uncomment this
line.
# Make sure to set the local network mask of '192.168.1.0' to your local
network mask
# (check ifconfig if not sure or google for how to do it).
#-A INPUT -p tcp -m state –state NEW -s 192.168.1.0/24 –dport 22 -j ACCEPT

# Allow response to remote ping  (delete/comment this line if you don't want
to respond to ping commands)
-A INPUT -p icmp –icmp-type echo-request -j ACCEPT

# Log iptables denied calls – these will go in your kernal log file. Google
for kernal log info.
-A INPUT -m limit –limit 5/min -j LOG –log-prefix "iptables denied: " –log-
level 7

# Drop all other inbound – default deny unless explicitly allowed policy
-A INPUT -j DROP
-A FORWARD -j DROP

COMMIT
```

The above block will create a set of Firewall rules which will block all access to your machine except for SSH.

To save your work, press CTRL+X to exit Nano. You will then be prompted to confirm saving your changes. Press 'y' and then the Enter key to confirm saving the file to the /etc/iptables.rules file you specified when opening Nano. If you get errors relating to file permissions, you may not have started Nano with the 'Sudo' action or as a super user, in which case you will need to start again!

Additional rule blocks for other services

In the above code I included a marker for where to drop additional rule blocks

```
# === PLACE ADDITIONAL RULE BLOCKS HERE === #
# ========= #
```

Between these two lines, drop as many rule blocks as you need for your Raspberry Pi.

Here are some good examples:

```
# Allow HTTP and HTTPS connections from anywhere (the normal ports for
websites and SSL).
-A INPUT -p tcp –dport 80 -j ACCEPT
-A INPUT -p tcp –dport 443 -j ACCEPT
# Alternate HTTP/HTTPS (you can leave these out if you only use 80/443)
```

```
-A INPUT -p tcp –dport 8080 -j ACCEPT
-A INPUT -p tcp –dport 8443 -j ACCEPT
```

If you use SVN for source control and want external access to the whole web:

```
# Allow SVN (Source Control) Services
-A INPUT -p tcp –dport 3690 -j ACCEPT
-A INPUT -p udp –dport 3690 -j ACCEPT
```

This block permits Local network only access to Samba shares

```
# Allow Samba access (aka Windows File Sharing) on LAN
-A INPUT -s 192.168.1.0/24 -p udp -m udp –dport 137 -j ACCEPT
-A INPUT -s 192.168.1.0/24 -p udp -m udp –dport 138 -j ACCEPT
-A INPUT -m state –state NEW -m tcp -p tcp -s 192.168.1.0/24 –dport 139 -j
ACCEPT
-A INPUT -m state –state NEW -m tcp -p tcp -s 192.168.1.0/24 –dport 445 -j
ACCEPT
```

I use a printer with FTP support to send scans to an FTP folder on my Pi, so I use local FTP only:

```
# Allow FTP access only on LAN
-A INPUT -m state –state NEW -m tcp -p tcp -s 192.168.1.0/24 –dport 21 -j
ACCEPT
-A INPUT -m state –state NEW -m tcp -p tcp -s 192.168.1.0/24 –dport 20 -j
ACCEPT
```

If you run a MySQL server on your PI, it's nice to use tools like MySQL Workbench to connect to it:

```
# Allow MySQL remote access only on LAN
# NOTE: You also need to disable the bind_address field in your my.cnf file if
you want outside access
-A INPUT -m state –state NEW -m tcp -p tcp -s 192.168.1.0/24 –dport 3306 -j
ACCEPT
```

Installing your rules file

So you have created your rules file but it isn't being used yet. To enable your rules in iptables run the following command:

```
~# sudo iptables-restore < /etc/iptables.rules
```

This will have iptables read the rules file you made and try and load in the new rules. If there are any errors in your file IPTables will fail to load the file and tell you. If any errors occur, use Nano to edit the file again and correct any errors.

IMPORTANT NOTE: Like many Linux programs iptables is sensitive to the line endings of files. If you create your iptables.rules file on a Windows PC, make sure you save your rules file with Linux-style end of line characters. This is not something that Microsoft Notepad

typically does. You should use a better text editor like Notepad2, Notepad++, Textpad etc. Those tools offer abilities to either save as a Linux-style text file or give you the option of setting or converting the line endings.

Have your rules file read automatically

I don't use this myself, but if you want to make your iptables reload the rules from periodically or at system start up, open crontab:

```
~# sudo EDITOR=nano crontab -e
```

Then enter the following lines at the end of the crontab file:

```
# IPTables rules reloading
* */1 * * * sudo iptables-restore < /etc/iptables.rules >/dev/null 2>&1
@reboot sudo iptables-restore < /etc/iptables.rules >/dev/null 2>&1
```

The first crontab line should reload the rules every hour. The second crontab line will trigger a rules load at system reboot. The user you are doing this as needs access to sudo – either a sudoer user (like 'pi' on the Raspbian distro), or the root account itself.

Use the same CTRL+X, y, Enter key presses to save and exit Nano.