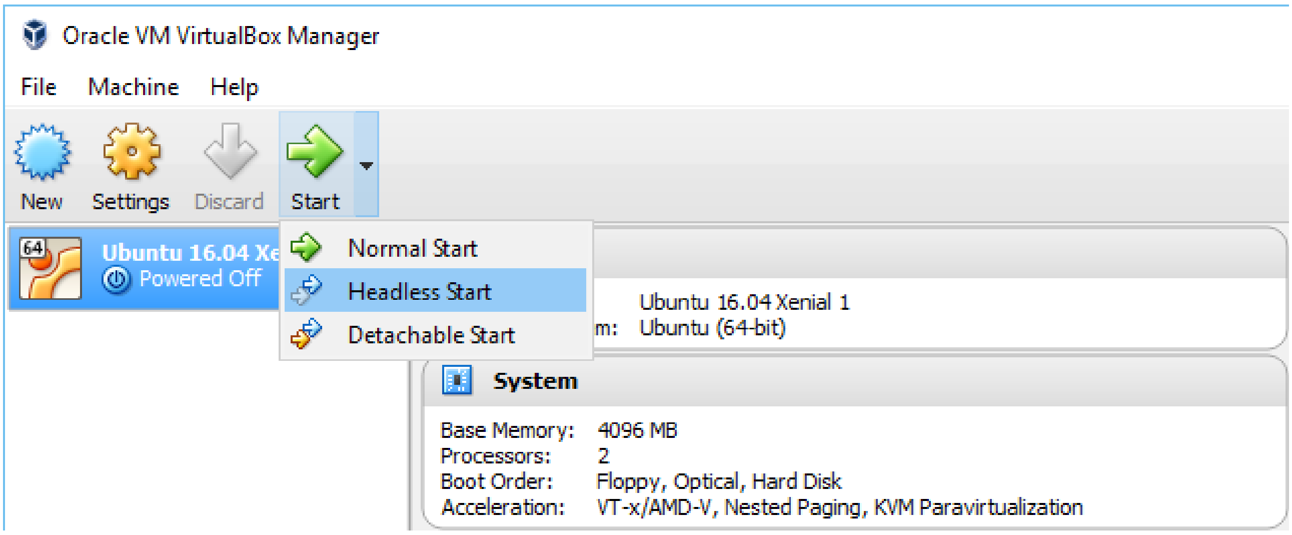
How to use iptables on Ubuntu

Revised: 16-August-2016 by David Walling

This "How To" document describes using the iptables program to define firewall rules for our Ubuntu server. We will also explore using the iptables-persistent package for Ubuntu. Also, we will demonstrate running our VM "headless" and accessing Ubuntu for our configuration work using SSH instead of the native console.

First we will restart our Ubuntu server VM using the "headless start" VirtualBox feature and connect to Ubuntu over SSH using PuTTY. The "Headless Start" option will start our VM without a GUI.

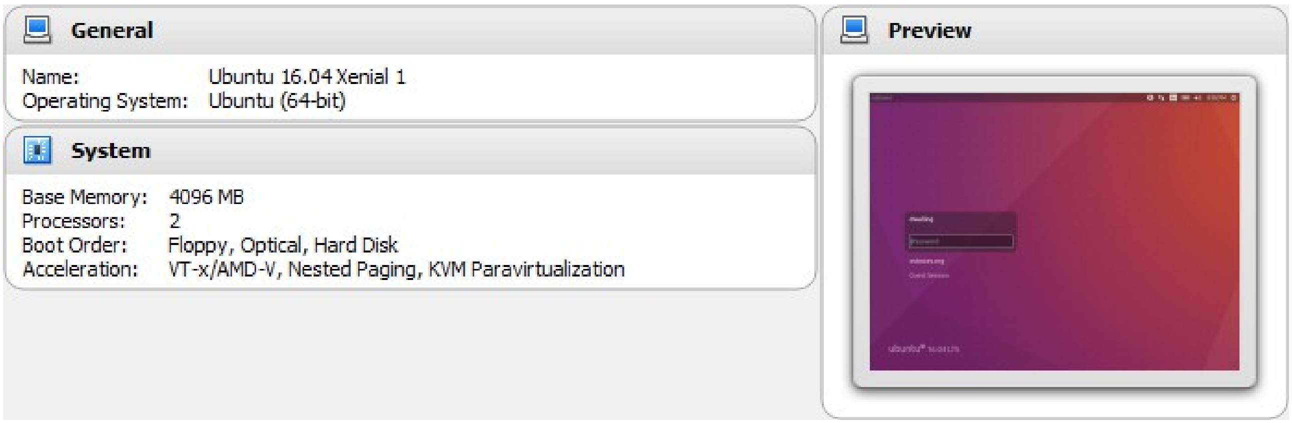


Ubuntu 16.04 Xenial 1

m: Ubuntu (64-bit)

Base Memory: 4096 MB
Processors: 2
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

During startup, we can see what the VM would display in the "Preview" pane.

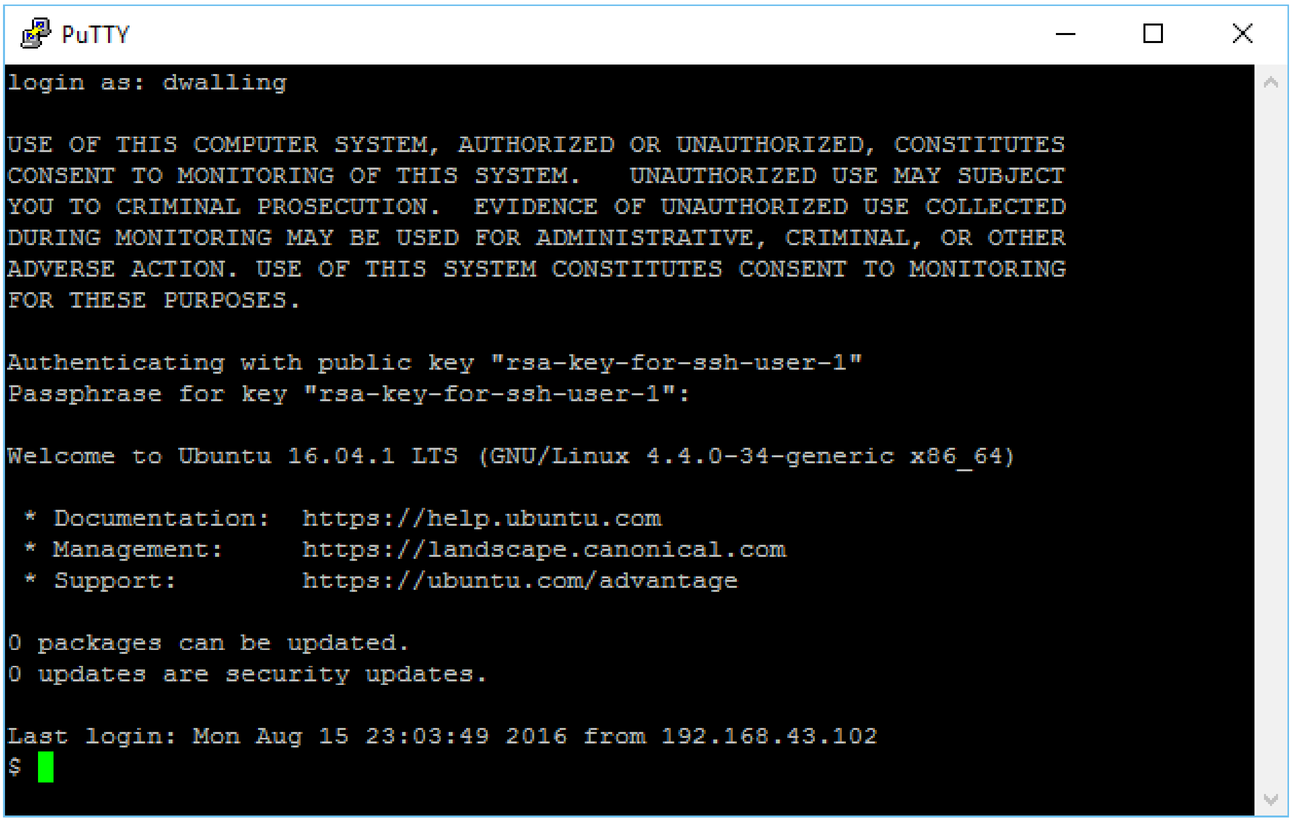


Name: Ubuntu 16.04 Xenial 1
Operating System: Ubuntu (64-bit)

Base Memory: 4096 MB
Processors: 2
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

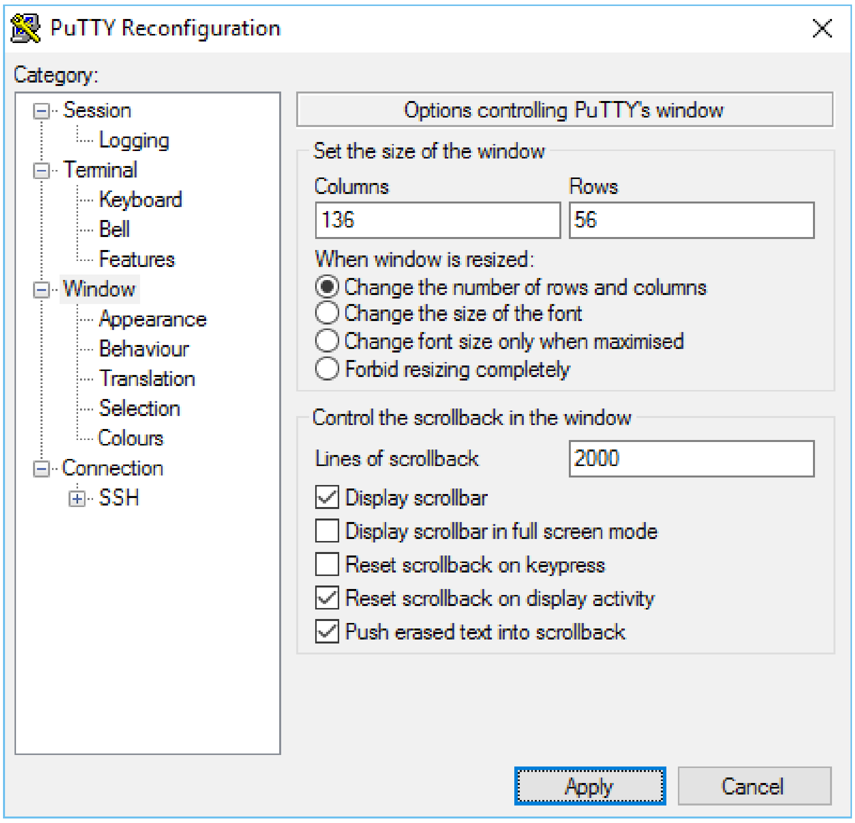
Once the VM is started, the VirtualBox OS Manager window can be closed. But our Ubuntu VM is still running.

Now, connect using PuTTY to Ubuntu using SSH-2 and public-key authentication.



```
login as: dwalling

USE OF THIS COMPUTER SYSTEM, AUTHORIZED OR UNAUTHORIZED, CONSTITUTES
CONSENT TO MONITORING OF THIS SYSTEM.   UNAUTHORIZED USE MAY SUBJECT
YOU TO CRIMINAL PROSECUTION.  EVIDENCE OF UNAUTHORIZED USE COLLECTED
DURING MONITORING MAY BE USED FOR ADMINISTRATIVE, CRIMINAL, OR OTHER
ADVERSE ACTION. USE OF THIS SYSTEM CONSTITUTES CONSENT TO MONITORING
FOR THESE PURPOSES.

Authenticating with public key "rsa-key-for-ssh-user-1"
Passphrase for key "rsa-key-for-ssh-user-1":

Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-34-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

Last login: Mon Aug 15 23:03:49 2016 from 192.168.43.102
$ 
```

Since we have some work to do in this session, we can change our SSH console settings to make our screen dimensions a little bigger and maybe change the color options. Here, I am setting columns to 136, rows to 56 and scrollback lines to 2000.



Options controlling PuTTY's window

Set the size of the window

Columns: 136

Rows: 56

When window is resized:

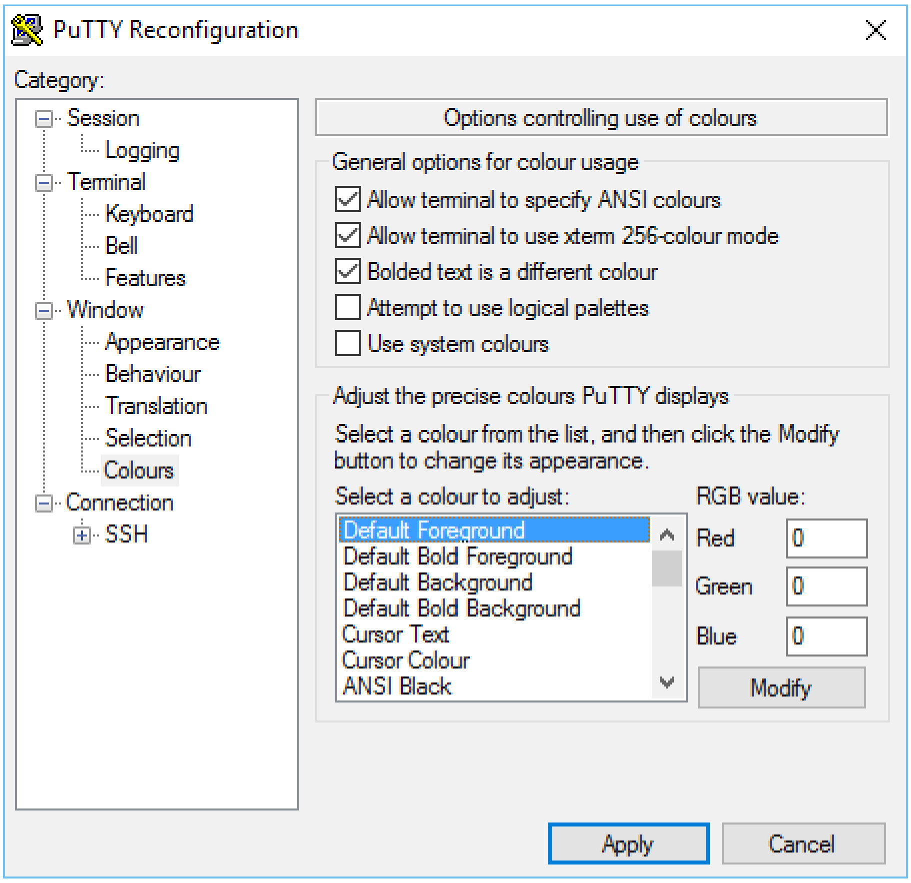
● Change the number of rows and columns
○ Change the size of the font
○ Change font size only when maximised
○ Forbid resizing completely

Control the scrollback in the window

Lines of scrollback: 2000

☑ Display scrollbar
□ Display scrollbar in full screen mode
□ Reset scrollback on keypress
☑ Reset scrollback on display activity
☑ Push erased text into scrollback

On the "Colours" dialog, we set the "Default Background" to RGB values 255, 255, 255 and the "Default Foreground" values to 0, 0, 0.



General options for colour usage

☑ Allow terminal to specify ANSI colours
☑ Allow terminal to use xterm 256-colour mode
☑ Bolded text is a different colour
□ Attempt to use logical palettes
□ Use system colours

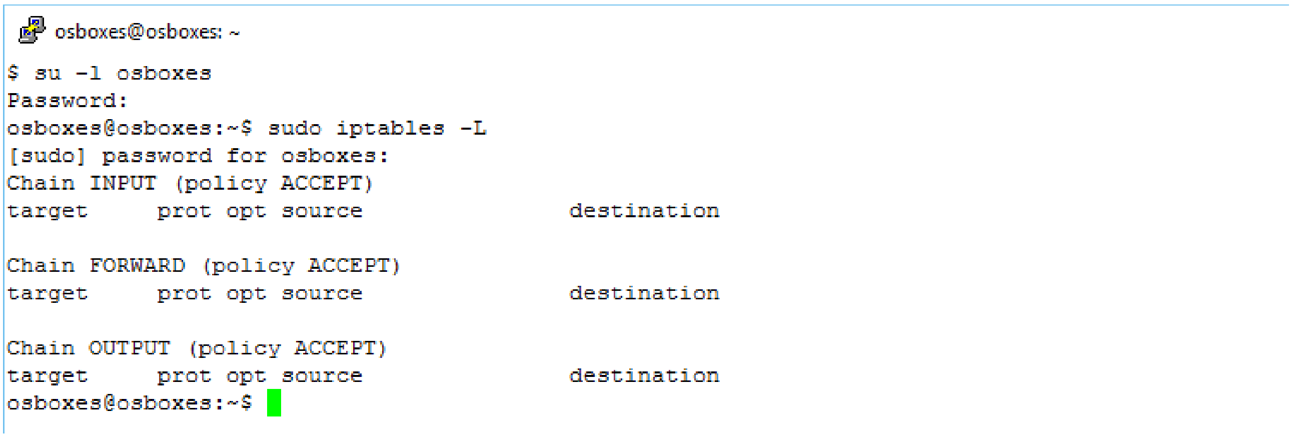
Adjust the precise colours PuTTY displays

Select a colour from the list, and then click the Modify button to change its appearance.

Select a colour to adjust: Default Foreground, Default Bold Foreground, Default Background, Default Bold Background, Cursor Text, Cursor Colour, ANSI Black

RGB value: Red 0, Green 0, Blue 0

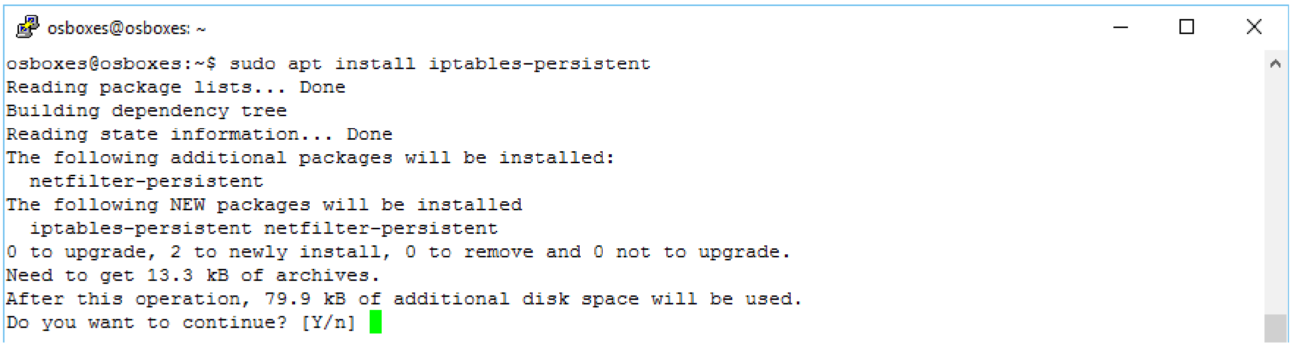
Working with iptables requires supervisor privileges. So, we will "su" to the osboxes logon. To display the current iptables configuration, use the command "iptables -L". As you can see, by default, Ubuntu sets the three default iptables "chains" to a wide-open ACCEPT policy. Nothing is restricted.



osboxes@osboxes: ~

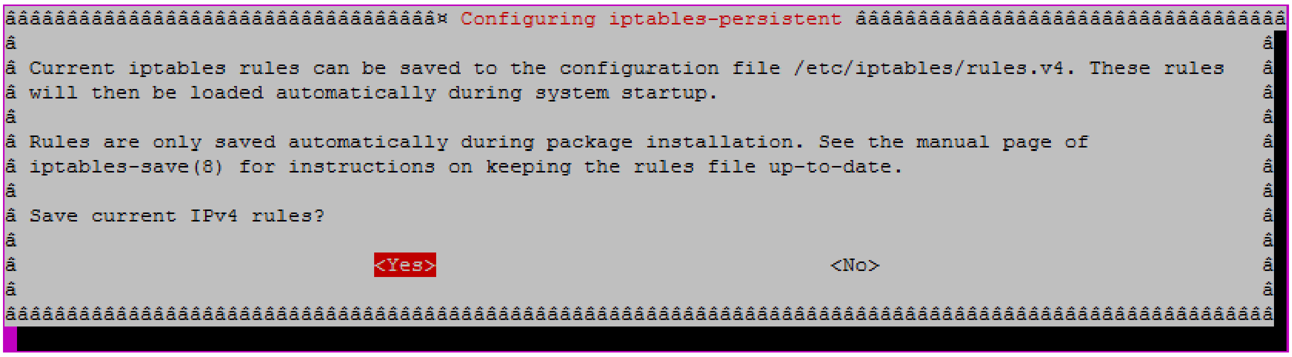
```
$ su -l osboxes
Password:
osboxes@osboxes:~$ sudo iptables -L
[sudo] password for osboxes:
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
osboxes@osboxes:~$ █
```

Before we define firewall rules with iptables, we want to make sure we can persist the changes. There are several methods to do this. Analyzing all of them is beyond the scope of this "How To". The method we will use here is to leverage the iptables-persistent package. Use apt to install this.

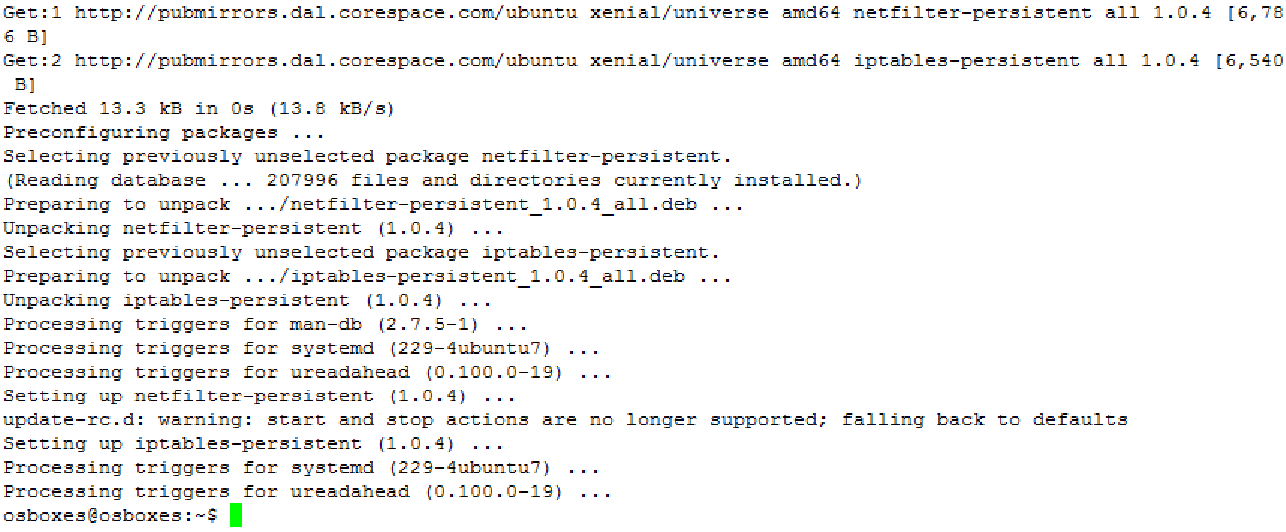


```
osboxes@osboxes:~$ sudo apt install iptables-persistent
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  netfilter-persistent
The following NEW packages will be installed
  iptables-persistent netfilter-persistent
0 to upgrade, 2 to newly install, 0 to remove and 0 not to upgrade.
Need to get 13.3 kB of archives.
After this operation, 79.9 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Once installed, iptables-persistent will save IPv4 and IPv6 rules in separate files in the /etc/iptables folder. During installation of this package, you will be prompted whether to save the current IPv4 rules into the /etc/iptables/rules.v4 file. We answer "<Yes>" to this question.

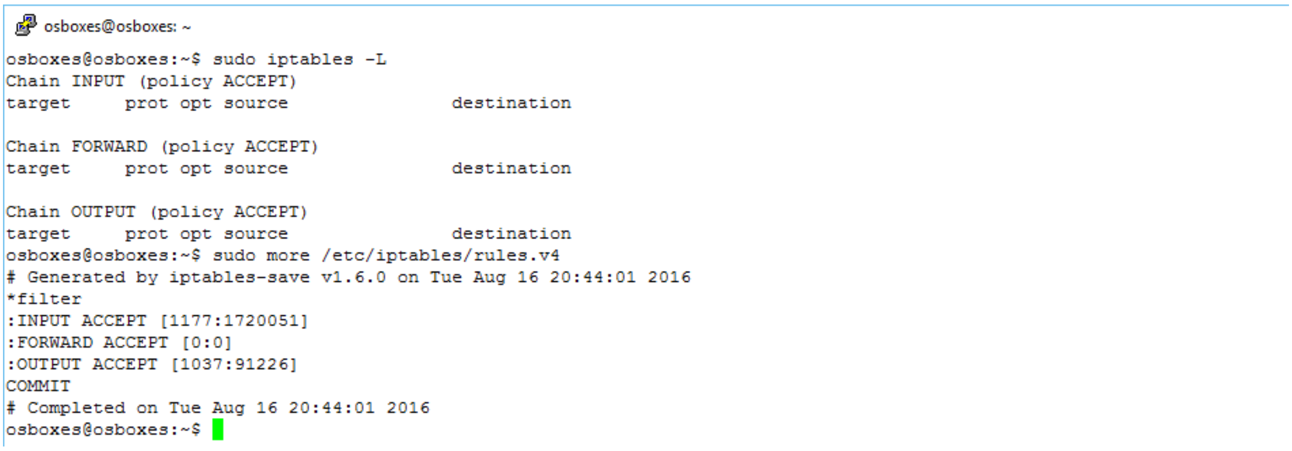


```
ââââââââââââââââââââââââââââââââââââ¤ Configuring iptables-persistent âââââââââââââââââââââââââââââââââââââ
â                                                                                                        â
â Current iptables rules can be saved to the configuration file /etc/iptables/rules.v4. These rules     â
â will then be loaded automatically during system startup.                                               â
â                                                                                                        â
â Rules are only saved automatically during package installation. See the manual page of                 â
â iptables-save(8) for instructions on keeping the rules file up-to-date.                                â
â                                                                                                        â
â Save current IPv4 rules?                                                                               â
â                                                                                                        â
â                              <Yes>                                 <No>                                â
â                                                                                                        â
ââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââ
```



```
Get:1 http://pubmirrors.dal.corespace.com/ubuntu xenial/universe amd64 netfilter-persistent all 1.0.4 [6,78
6 B]
Get:2 http://pubmirrors.dal.corespace.com/ubuntu xenial/universe amd64 iptables-persistent all 1.0.4 [6,540
 B]
Fetched 13.3 kB in 0s (13.8 kB/s)
Preconfiguring packages ...
Selecting previously unselected package netfilter-persistent.
(Reading database ... 207996 files and directories currently installed.)
Preparing to unpack .../netfilter-persistent_1.0.4_all.deb ...
Unpacking netfilter-persistent (1.0.4) ...
Selecting previously unselected package iptables-persistent.
Preparing to unpack .../iptables-persistent_1.0.4_all.deb ...
Unpacking iptables-persistent (1.0.4) ...
Processing triggers for man-db (2.7.5-1) ...
Processing triggers for systemd (229-4ubuntu7) ...
Processing triggers for ureadahead (0.100.0-19) ...
Setting up netfilter-persistent (1.0.4) ...
update-rc.d: warning: start and stop actions are no longer supported; falling back to defaults
Setting up iptables-persistent (1.0.4) ...
Processing triggers for systemd (229-4ubuntu7) ...
Processing triggers for ureadahead (0.100.0-19) ...
osboxes@osboxes:~$
```

With iptables-persistent installed, the current firewall rules can still be displayed using the command "iptables -L". In addition, displaying the file /etc/iptables/rules.v4 will display the last saved configuration, which will take effect when the system is restarted.

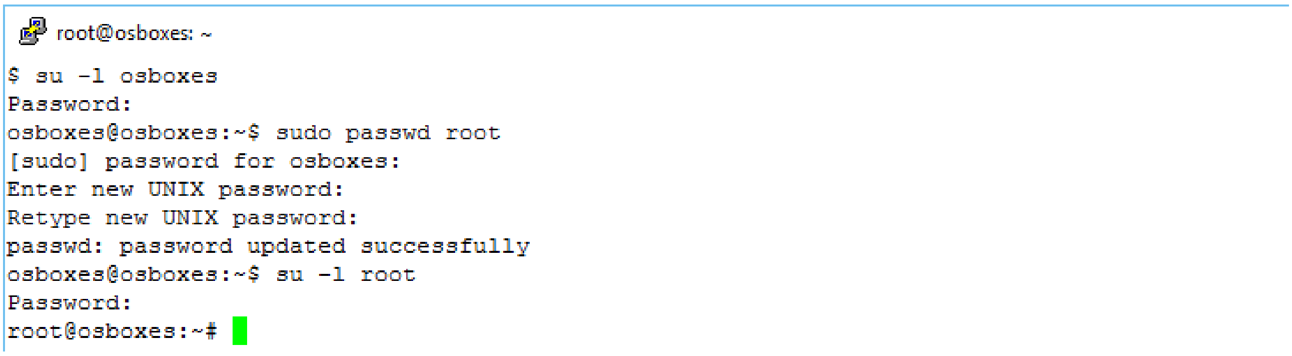


```
osboxes@osboxes:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
osboxes@osboxes:~$ sudo more /etc/iptables/rules.v4
# Generated by iptables-save v1.6.0 on Tue Aug 16 20:44:01 2016
*filter
:INPUT ACCEPT [1177:1720051]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1037:91226]
COMMIT
# Completed on Tue Aug 16 20:44:01 2016
osboxes@osboxes:~$
```

The format of the /etc/iptables/rules.v4 file is different than the ouput of the "iptables -L" command. But, provided that the active firewall state matches the rules.v4 file, both outputs indicate the same state. Just remember that updating rules.v4 doesn't make new rules active. We will see below how to activate new rules with "iptables-save" and update the rules.v4 file in a single command.

Now, we're going to create a local rules.v4 file to issue a series of iptables commands to define our firewall rules. This file will be only a working copy. We are going to create this file as the "root" user. On this installation of Ubuntu, the osboxes user was not able write to /etc/iptables.

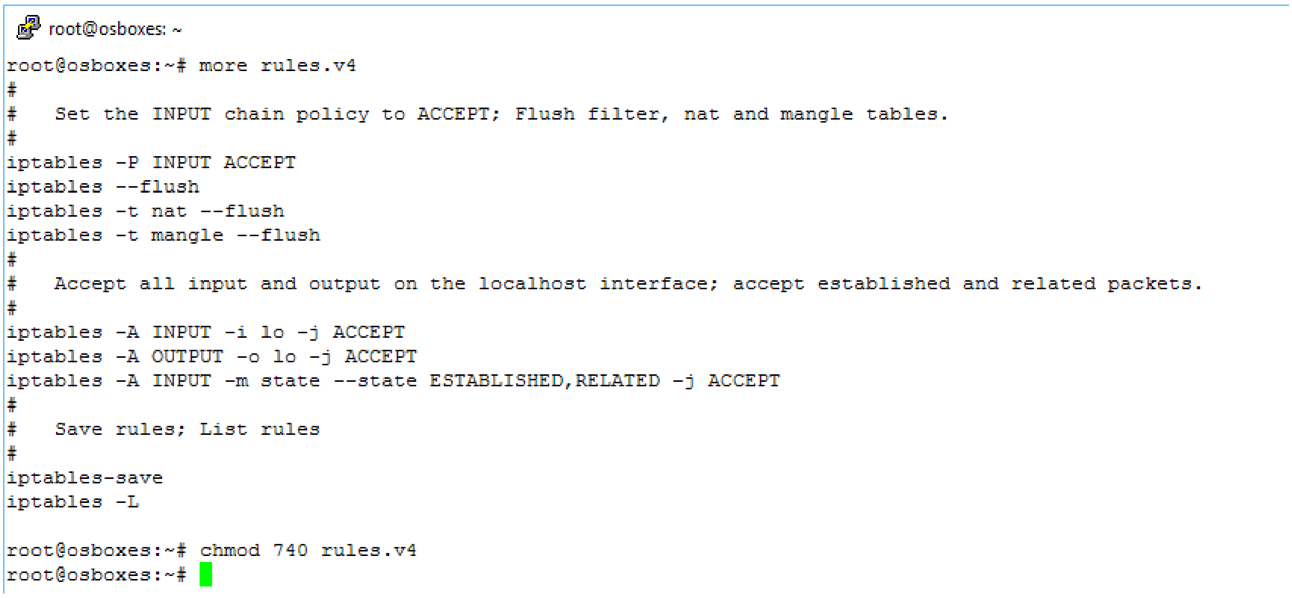
Change the root user password. Use appropriate password policies. Then switch to the root user.



```
$ su -l osboxes
Password:
osboxes@osboxes:~$ sudo passwd root
[sudo] password for osboxes:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
osboxes@osboxes:~$ su -l root
Password:
root@osboxes:~#
```

Now we can edit the local rules.v4 file

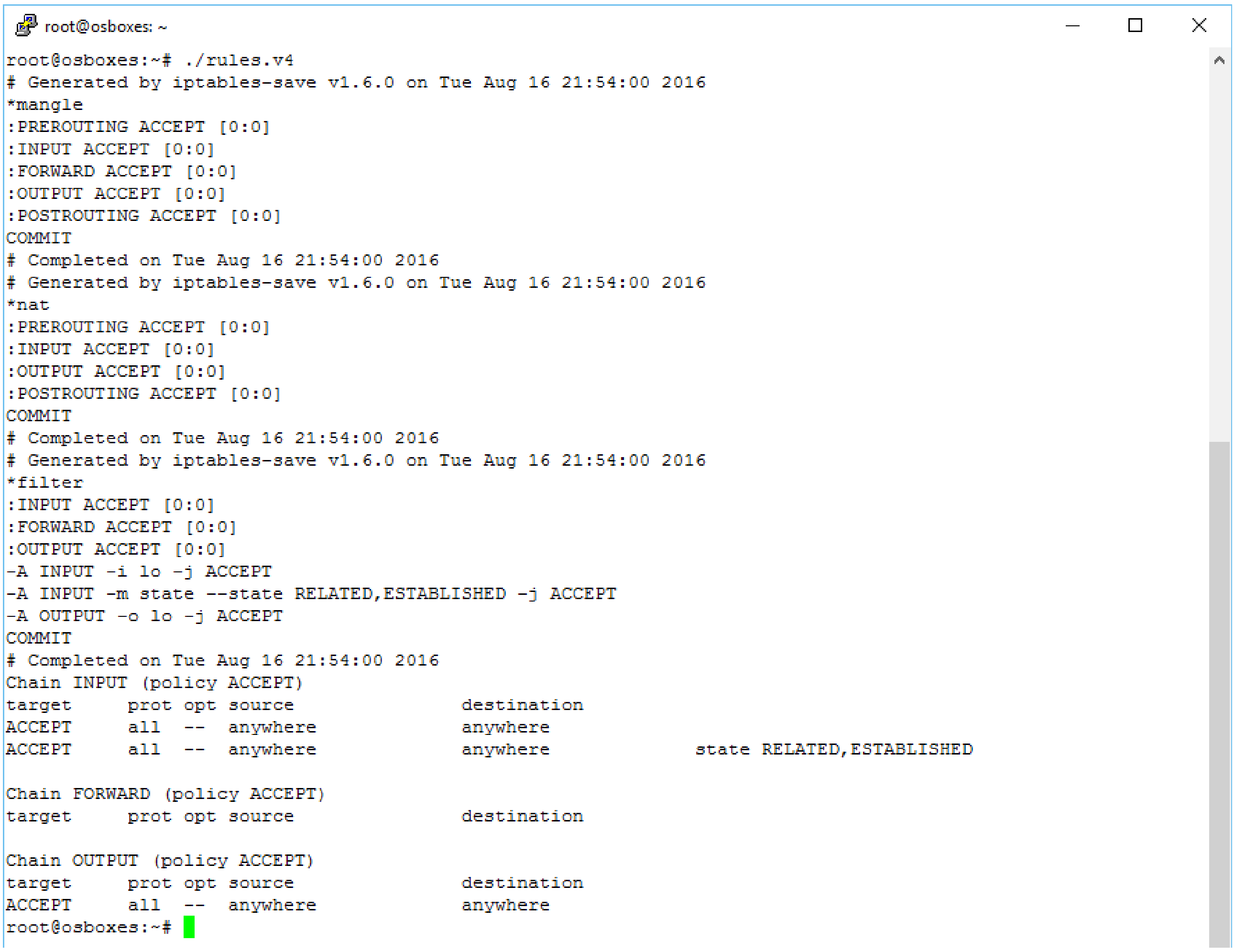
The first commands we will include will set the INPUT policy to ACCEPT and clear, or "flush", the current set of rules. We want to set the INPUT policy to ACCEPT at first so that, in case we make an error in our configuration, we can still connect to our server to make edits. Once we are satisfied that our rules are working, we will set the INPUT policy to DROP so that any incoming connection not covered by our rules is not allowed.



```
root@osboxes:~# more rules.v4
#
#   Set the INPUT chain policy to ACCEPT; Flush filter, nat and mangle tables.
#
iptables -P INPUT ACCEPT
iptables --flush
iptables -t nat --flush
iptables -t mangle --flush
#
#   Accept all input and output on the localhost interface; accept established and related packets.
#
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
#
#   Save rules; List rules
#
iptables-save
iptables -L

root@osboxes:~# chmod 740 rules.v4
root@osboxes:~#
```

We update the permissions on our rules.v4 file so that the osboxes user can read, write and execute it. Other "users" group members can read it.

Now when we run the rules.v4 as a script, we will define firewall rules, save them and list them. Note that the iptables-save command will output rules for each table. The "iptables -L" command lists one consolidated output.



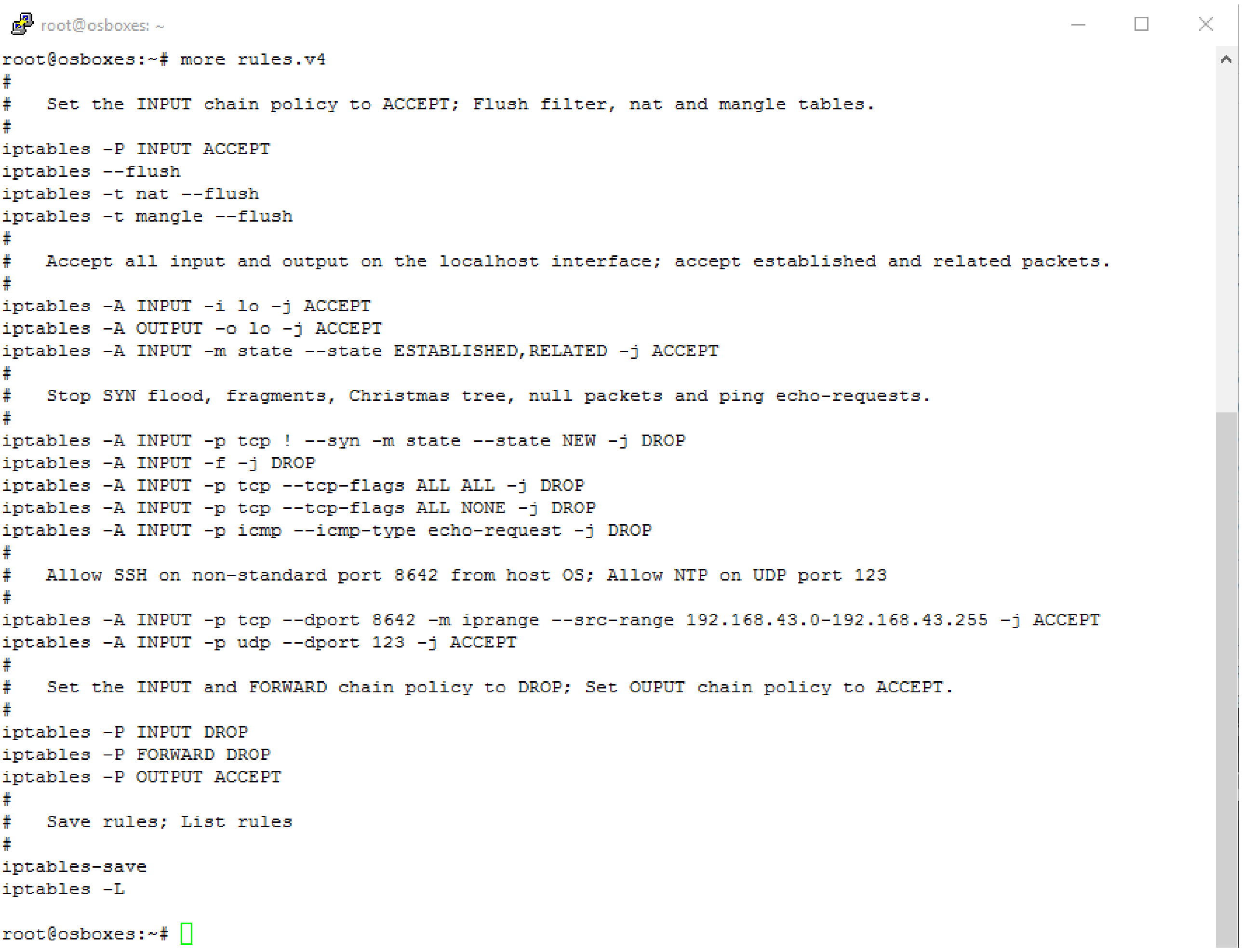
```
root@osboxes:~# ./rules.v4
# Generated by iptables-save v1.6.0 on Tue Aug 16 21:54:00 2016
*mangle
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
COMMIT
# Completed on Tue Aug 16 21:54:00 2016
# Generated by iptables-save v1.6.0 on Tue Aug 16 21:54:00 2016
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
COMMIT
# Completed on Tue Aug 16 21:54:00 2016
# Generated by iptables-save v1.6.0 on Tue Aug 16 21:54:00 2016
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -o lo -j ACCEPT
COMMIT
# Completed on Tue Aug 16 21:54:00 2016
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
root@osboxes:~#
```

Next we will add rules to protect against common attacks. We will stop SYN-flood attacks, packet fragments, Christmas-tree packets, null packets and we will prevent ICMP echo requests generated by the "ping" command.

```
#
#   Stop SYN flood, fragments, Christmas tree, null packets and ping echo-requests.
#
iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
iptables -A INPUT -f -j DROP
iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP
iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP
iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
```

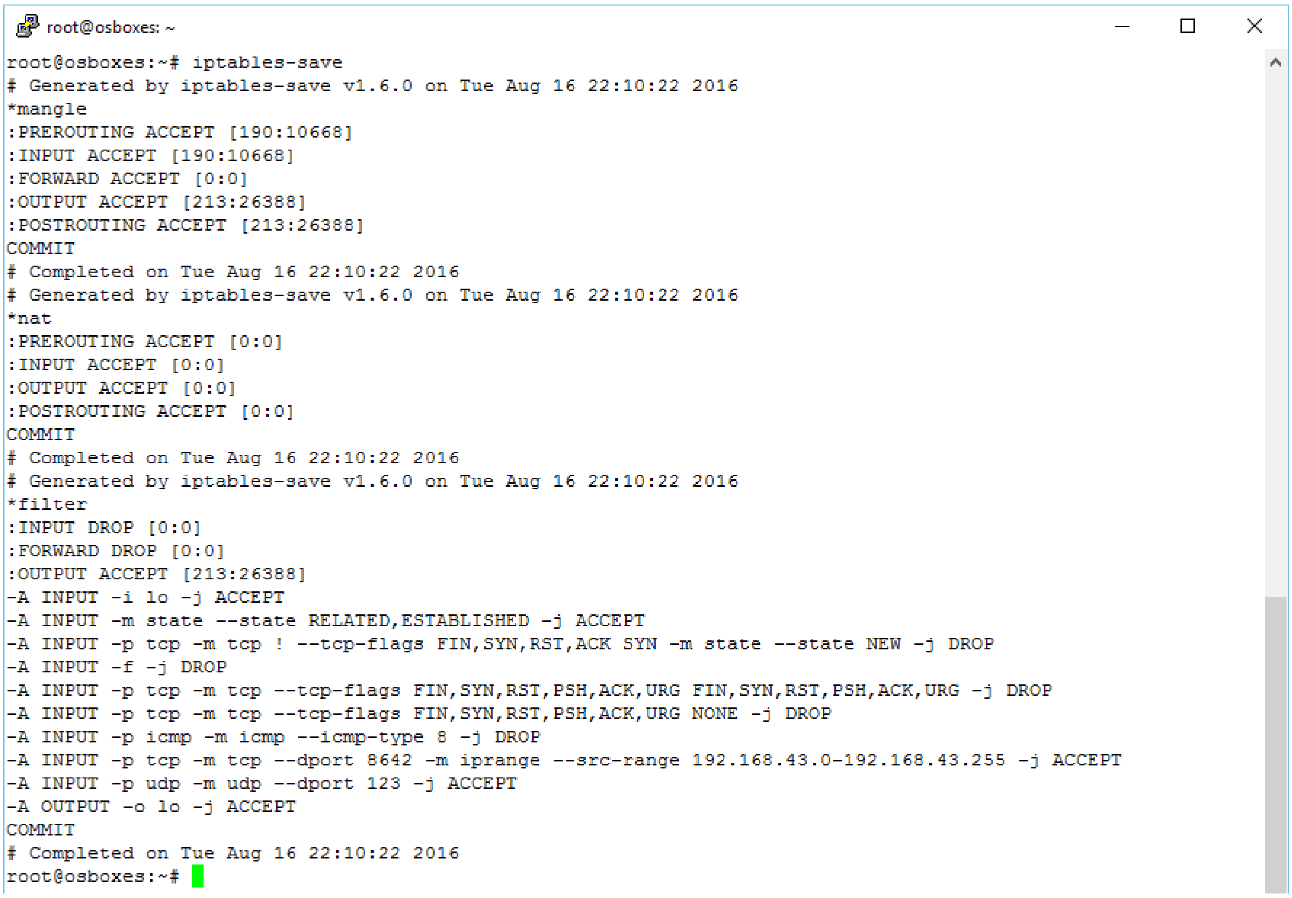
Next, we want to explicitly allow incoming SSH connections on our non-standard port 8642, but only from addresses on our host interface. We have already allowed established and related packets, so incoming and outgoing data flow on SSH connections will be allowed. We will also allow NTP (network time protocol) packets on port 123 using the UDP protocol. After allowing this traffic, we will change the INPUT and FORWARD chain policies to DROP, but leave the OUTPUT chain policy as ACCEPT.

Here is our completed rules.v4 file:



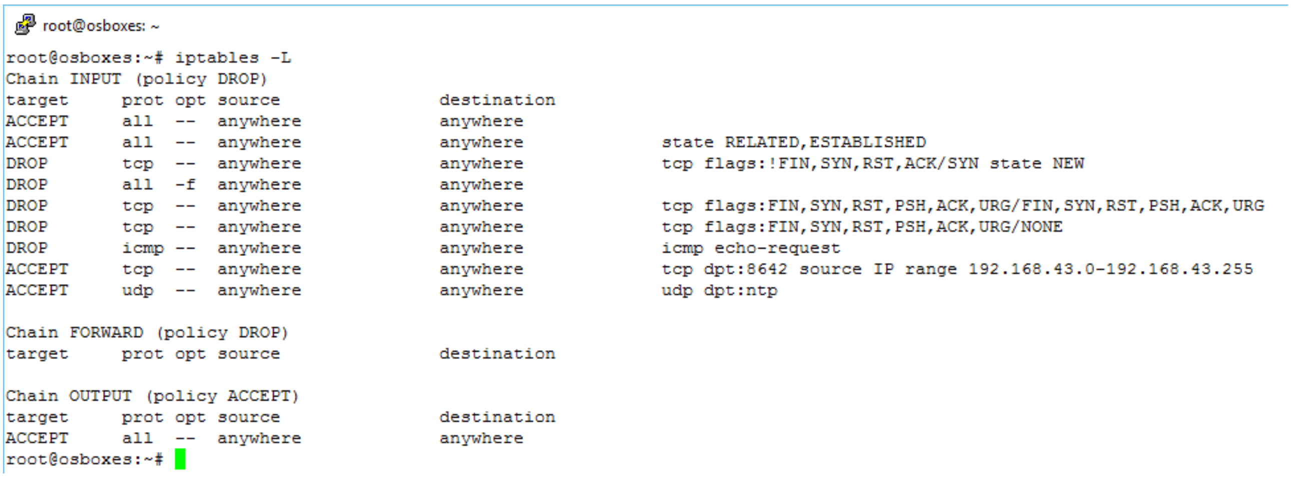
```
root@osboxes:~# more rules.v4
#
#   Set the INPUT chain policy to ACCEPT; Flush filter, nat and mangle tables.
#
iptables -P INPUT ACCEPT
iptables --flush
iptables -t nat --flush
iptables -t mangle --flush
#
#   Accept all input and output on the localhost interface; accept established and related packets.
#
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
#
#   Stop SYN flood, fragments, Christmas tree, null packets and ping echo-requests.
#
iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
iptables -A INPUT -f -j DROP
iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP
iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP
iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
#
#   Allow SSH on non-standard port 8642 from host OS; Allow NTP on UDP port 123
#
iptables -A INPUT -p tcp --dport 8642 -m iprange --src-range 192.168.43.0-192.168.43.255 -j ACCEPT
iptables -A INPUT -p udp --dport 123 -j ACCEPT
#
#   Set the INPUT and FORWARD chain policy to DROP; Set OUPUT chain policy to ACCEPT.
#
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT
#
#   Save rules; List rules
#
iptables-save
iptables -L

root@osboxes:~# 
```

Here is the output of the iptables-save command.



```
root@osboxes:~# iptables-save
# Generated by iptables-save v1.6.0 on Tue Aug 16 22:10:22 2016
*mangle
:PREROUTING ACCEPT [190:10668]
:INPUT ACCEPT [190:10668]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [213:26388]
:POSTROUTING ACCEPT [213:26388]
COMMIT
# Completed on Tue Aug 16 22:10:22 2016
# Generated by iptables-save v1.6.0 on Tue Aug 16 22:10:22 2016
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
COMMIT
# Completed on Tue Aug 16 22:10:22 2016
# Generated by iptables-save v1.6.0 on Tue Aug 16 22:10:22 2016
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [213:26388]
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m tcp ! --tcp-flags FIN,SYN,RST,ACK SYN -m state --state NEW -j DROP
-A INPUT -f -j DROP
-A INPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG FIN,SYN,RST,PSH,ACK,URG -j DROP
-A INPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG NONE -j DROP
-A INPUT -p icmp -m icmp --icmp-type 8 -j DROP
-A INPUT -p tcp -m tcp --dport 8642 -m iprange --src-range 192.168.43.0-192.168.43.255 -j ACCEPT
-A INPUT -p udp -m udp --dport 123 -j ACCEPT
-A OUTPUT -o lo -j ACCEPT
COMMIT
# Completed on Tue Aug 16 22:10:22 2016
root@osboxes:~#
```

Here is the output of the "iptables -L" command.



```
root@osboxes:~# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED
DROP       tcp  --  anywhere             anywhere             tcp flags:!FIN,SYN,RST,ACK/SYN state NEW
DROP       all  -f  anywhere             anywhere
DROP       tcp  --  anywhere             anywhere             tcp flags:FIN,SYN,RST,PSH,ACK,URG/FIN,SYN,RST,PSH,ACK,URG
DROP       tcp  --  anywhere             anywhere             tcp flags:FIN,SYN,RST,PSH,ACK,URG/NONE
DROP       icmp --  anywhere             anywhere             icmp echo-request
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:8642 source IP range 192.168.43.0-192.168.43.255
ACCEPT     udp  --  anywhere             anywhere             udp dpt:ntp

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
root@osboxes:~#
```

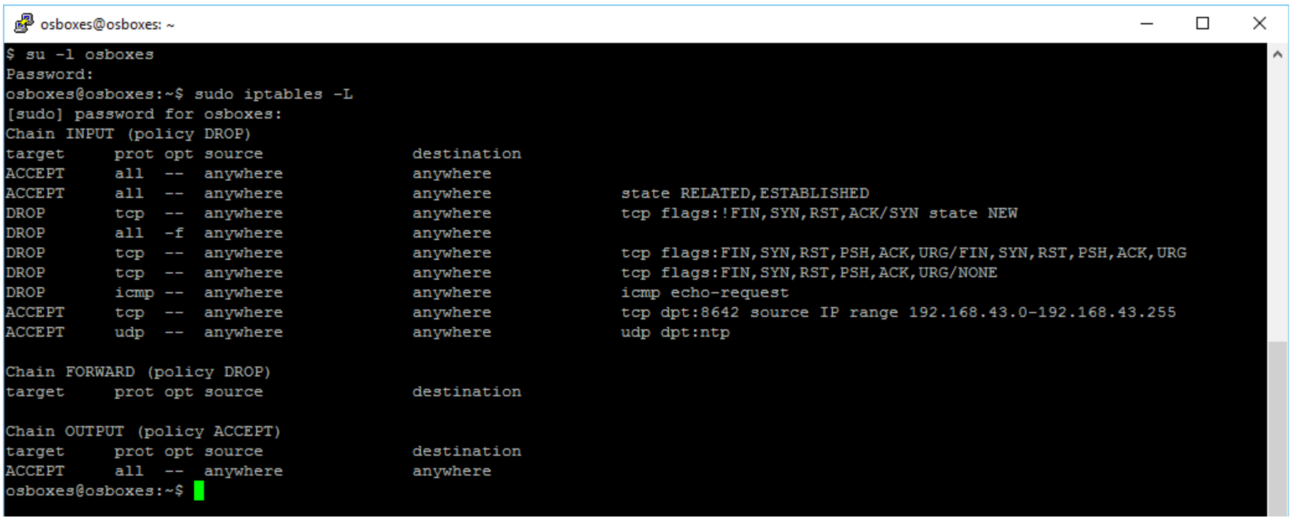
Now, with our rules saved locally and made active using the iptables-save command. We can overwrite the copy of rules.v4 in /etc/iptables. Use this command for this purpose:

iptables-save >> /etc/iptables/rules.v4

Now we can restart Ubuntu and verify that our firewall rules are in place on startup.

Issue the "reboot" command from our root user prompt. This will close our SSH connection immediately and restart the Ubuntu VM in headless mode.

Open a new SSH session using PuTTY. Logon as our SSH user. use the "su" command to logon as osboxes and issue the "sudo iptables -L" command to display active firewall rules.



```
$ su -l osboxes
Password:
osboxes@osboxes:~$ sudo iptables -L
[sudo] password for osboxes:
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED
DROP       tcp  --  anywhere             anywhere             tcp flags:!FIN,SYN,RST,ACK/SYN state NEW
DROP       all  -f  anywhere             anywhere
DROP       tcp  --  anywhere             anywhere             tcp flags:FIN,SYN,RST,PSH,ACK,URG/FIN,SYN,RST,PSH,ACK,URG
DROP       tcp  --  anywhere             anywhere             tcp flags:FIN,SYN,RST,PSH,ACK,URG/NONE
DROP       icmp --  anywhere             anywhere             icmp echo-request
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:8642 source IP range 192.168.43.0-192.168.43.255
ACCEPT     udp  --  anywhere             anywhere             udp dpt:ntp

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
osboxes@osboxes:~$
```

That concludes this "How To". We have installed the iptables-persistent package and defined our initial firewall configuration.