Linux Firewalls (Ubuntu IPTables)

Introduction

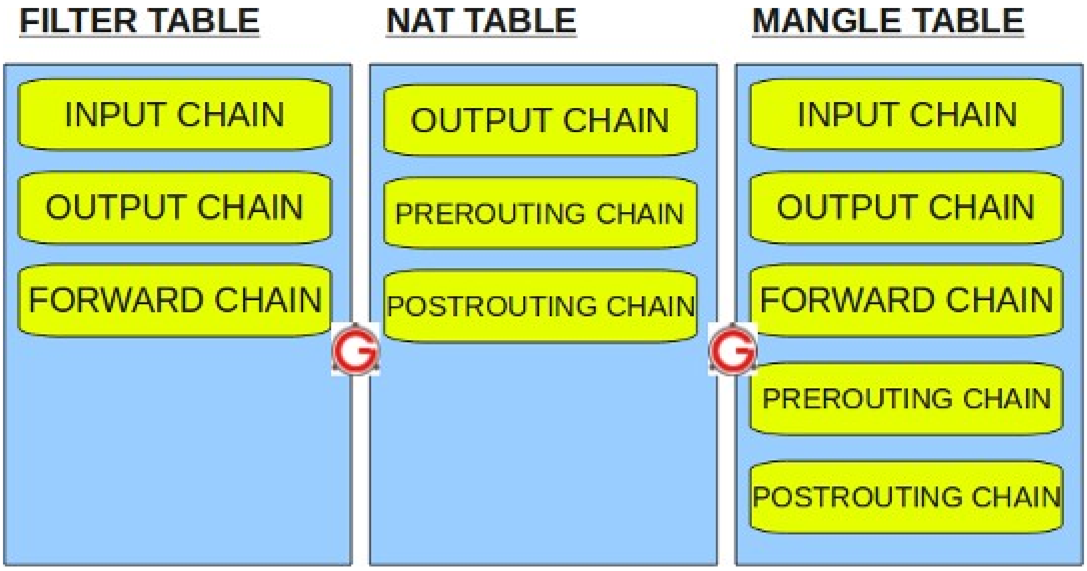
Linux Firewalls is the way to make our Linux OS more secure and safe because it enables you to control your connection ports and your inbound and outbound traffic. To control this inbound and outbound traffic, Linux OS uses a software called iptables. IPTables is actually a net filter software which is integrated with the kernel implementation, it also provides filtering features that can filter the inbound and outbound in routed traffics to our computer system.

IPTables Organization

The iptables organizes its information in tables, each tables consists of number of chains and each chain consists of many rules, these rules are responsible for controlling the system traffic. Each Chain has a default taken action called chain default policy and it used when there is no rule created for a specified service or traffic.

Main IPTabels

1. **filter table** is the default table.
2. **nat table** is used to tell the kernel what connections to change and how to change them.
3. **mangle table** is mainly used for mangling packets.



FILTER TABLE

INPUT CHAIN

OUTPUT CHAIN

FORWARD CHAIN

NAT TABLE

OUTPUT CHAIN

PREROUTING CHAIN

POSTROUTING CHAIN

MANGLE TABLE

INPUT CHAIN

OUTPUT CHAIN

FORWARD CHAIN

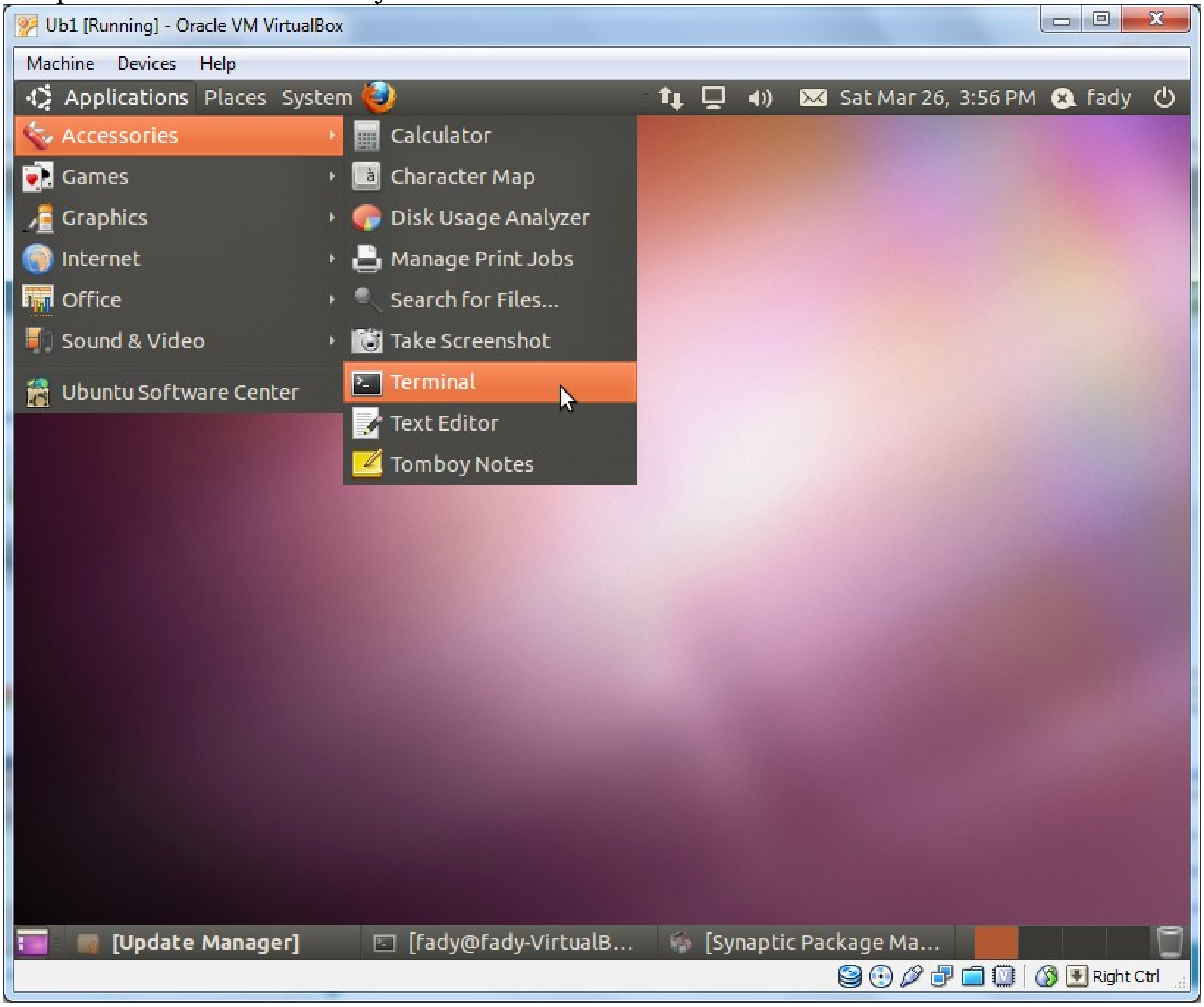
PREROUTING CHAIN

POSTROUTING CHAIN

IPTables Useful Commands

$dpkg-query -l iptables	used for **l**isting the packages that matches iptables.
$dpkg-query -s iptables	used for recognizing the **s**tatus of the iptable packages.
$dpkg-query -L iptables	used for **l**isting the files and libraries that related to the iptables.
$iptables -h	used for getting the iptables **h**elp.
$iptables -L	used for **l**isting the table chains and the rules of each chain, to specify a certain table use -t followed by the name of the table.
$iptable -A	used to **A**ppend a rule to a certain chain.
$iptables -I	used for **I**nserting a rule in a location not only adding it to the tail of the chain of rules.
$iptable -D	used for **d**eleting a rule from a chain in a table.
$iptable -F	used for **f**lushing all rules of a chain.
$iptable-save	used for **sav**ing all current rules and chains of the system in a file.
$iptable-restore	used for **restor**ing saved rules from a file.

To open the console in Ubuntu just click as shown



http://en.wikipedia.org/wiki/Dpkg

dpkg is the software at the base of the Debian package management system. dpkg is used to install, remove, and provide information about .deb packages. dpkg itself is a low level tool; higher level tools, such as APT, are used to fetch packages from remote locations or deal with complex package relations. The Debian package "dpkg" provides the dpkg program, as well as several other programs necessary for run-time functioning of the packaging system.

http://man.he.net/man1/dpkg-query

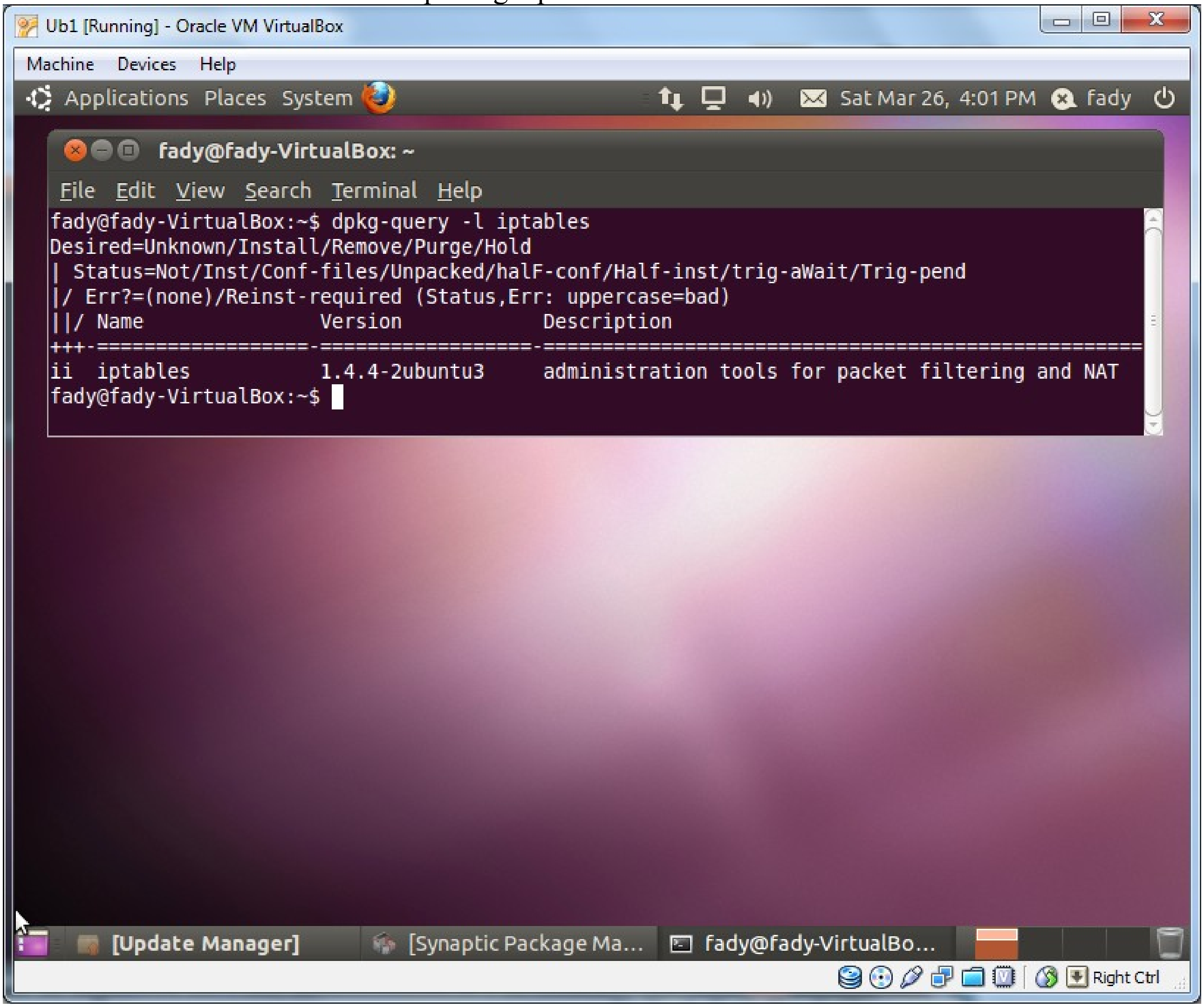
dpkg-query is a tool to show information about packages listed in the dpkg database.

Options 1

-l, --list package-name-pattern

List packages matching given pattern.

Here we choose the package iptables.



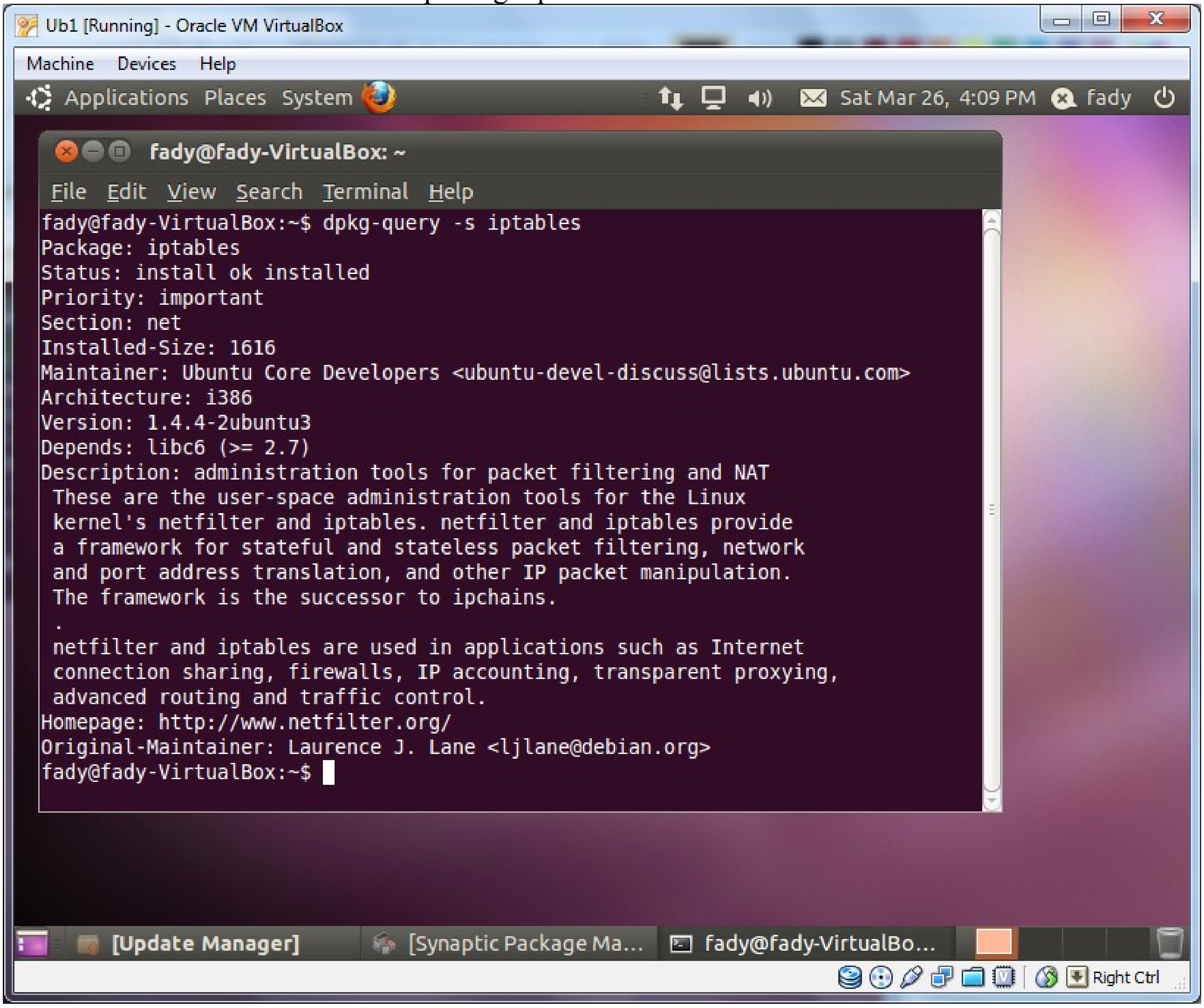
```
fady@fady-VirtualBox:~$ dpkg-query -l iptables
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name                Version             Description
+++-===================-===================-=================================================
ii  iptables            1.4.4-2ubuntu3      administration tools for packet filtering and NAT
fady@fady-VirtualBox:~$
```

Options 2

-s, --status package-name

Report status of specified package.

Here we choose the package iptables.



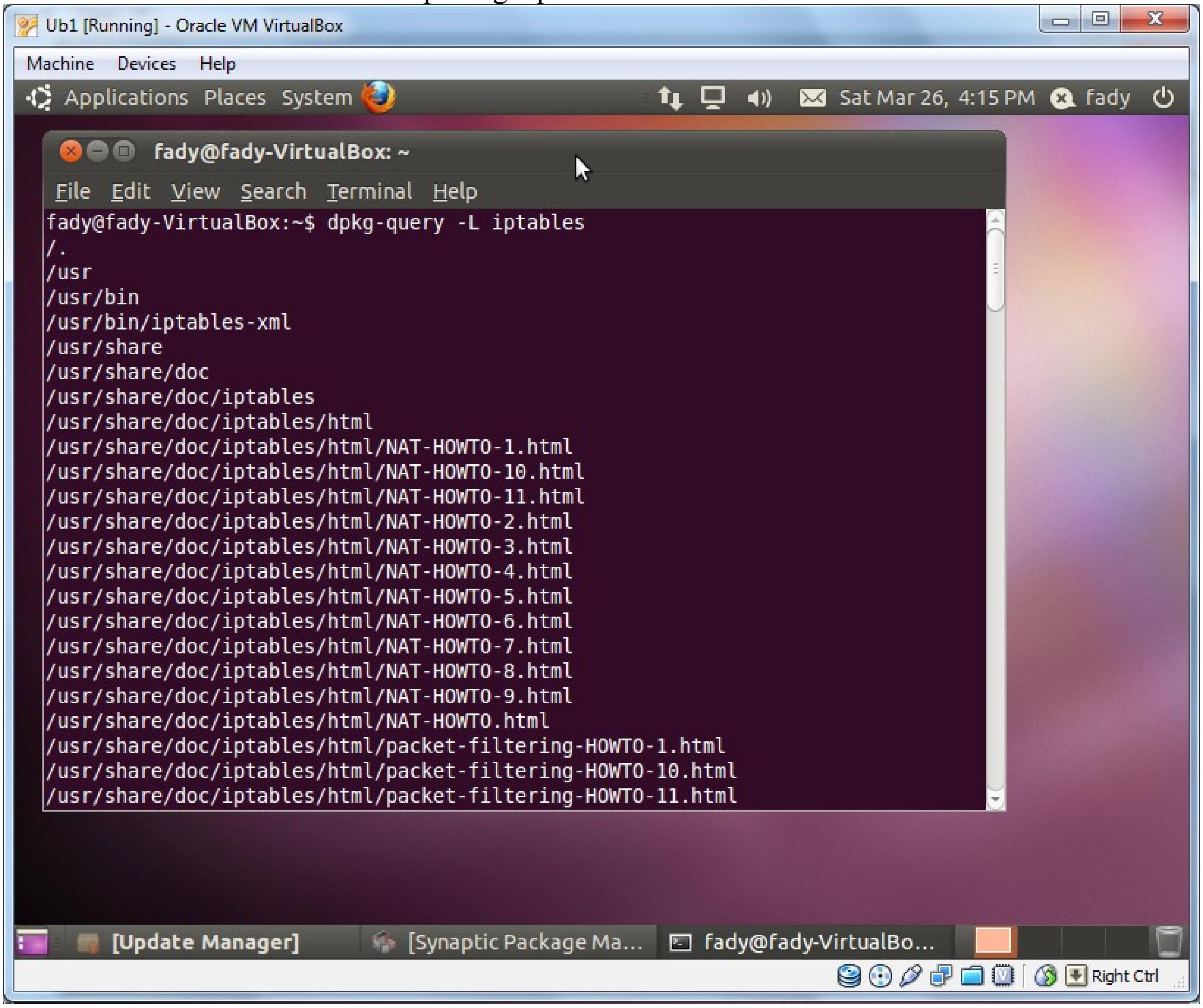
```
fady@fady-VirtualBox:~$ dpkg-query -s iptables
Package: iptables
Status: install ok installed
Priority: important
Section: net
Installed-Size: 1616
Maintainer: Ubuntu Core Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Architecture: i386
Version: 1.4.4-2ubuntu3
Depends: libc6 (>= 2.7)
Description: administration tools for packet filtering and NAT
 These are the user-space administration tools for the Linux
 kernel's netfilter and iptables. netfilter and iptables provide
 a framework for stateful and stateless packet filtering, network
 and port address translation, and other IP packet manipulation.
 The framework is the successor to ipchains.
 .
 netfilter and iptables are used in applications such as Internet
 connection sharing, firewalls, IP accounting, transparent proxying,
 advanced routing and traffic control.
Homepage: http://www.netfilter.org/
Original-Maintainer: Laurence J. Lane <ljlane@debian.org>
fady@fady-VirtualBox:~$
```

Options 3

-L, --listfiles package-name

List files installed to your system from package-name.

Here we choose the package iptables.



```
fady@fady-VirtualBox:~$ dpkg-query -L iptables
/.
/usr
/usr/bin
/usr/bin/iptables-xml
/usr/share
/usr/share/doc
/usr/share/doc/iptables
/usr/share/doc/iptables/html
/usr/share/doc/iptables/html/NAT-HOWTO-1.html
/usr/share/doc/iptables/html/NAT-HOWTO-10.html
/usr/share/doc/iptables/html/NAT-HOWTO-11.html
/usr/share/doc/iptables/html/NAT-HOWTO-2.html
/usr/share/doc/iptables/html/NAT-HOWTO-3.html
/usr/share/doc/iptables/html/NAT-HOWTO-4.html
/usr/share/doc/iptables/html/NAT-HOWTO-5.html
/usr/share/doc/iptables/html/NAT-HOWTO-6.html
/usr/share/doc/iptables/html/NAT-HOWTO-7.html
/usr/share/doc/iptables/html/NAT-HOWTO-8.html
/usr/share/doc/iptables/html/NAT-HOWTO-9.html
/usr/share/doc/iptables/html/NAT-HOWTO.html
/usr/share/doc/iptables/html/packet-filtering-HOWTO-1.html
/usr/share/doc/iptables/html/packet-filtering-HOWTO-10.html
/usr/share/doc/iptables/html/packet-filtering-HOWTO-11.html
```

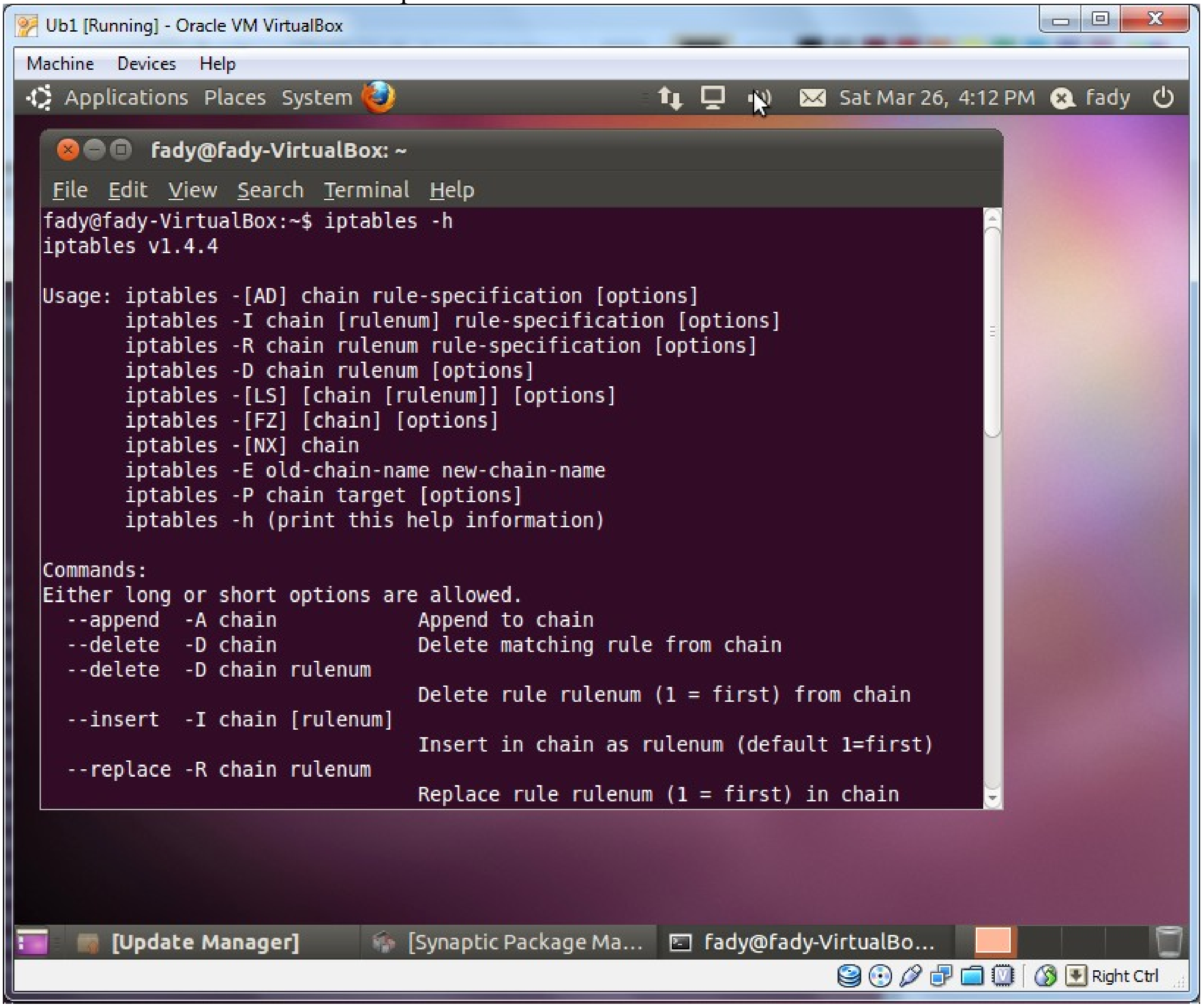
http://man.he.net/?topic=iptables&section=all

iptables is used to set up, maintain, and inspect the tables of IP packet filter rules in the Linux kernel. Each rule can match a set of packets and specify what to do with a packet that matches.

Options 1

-h, --help

To receive command help.

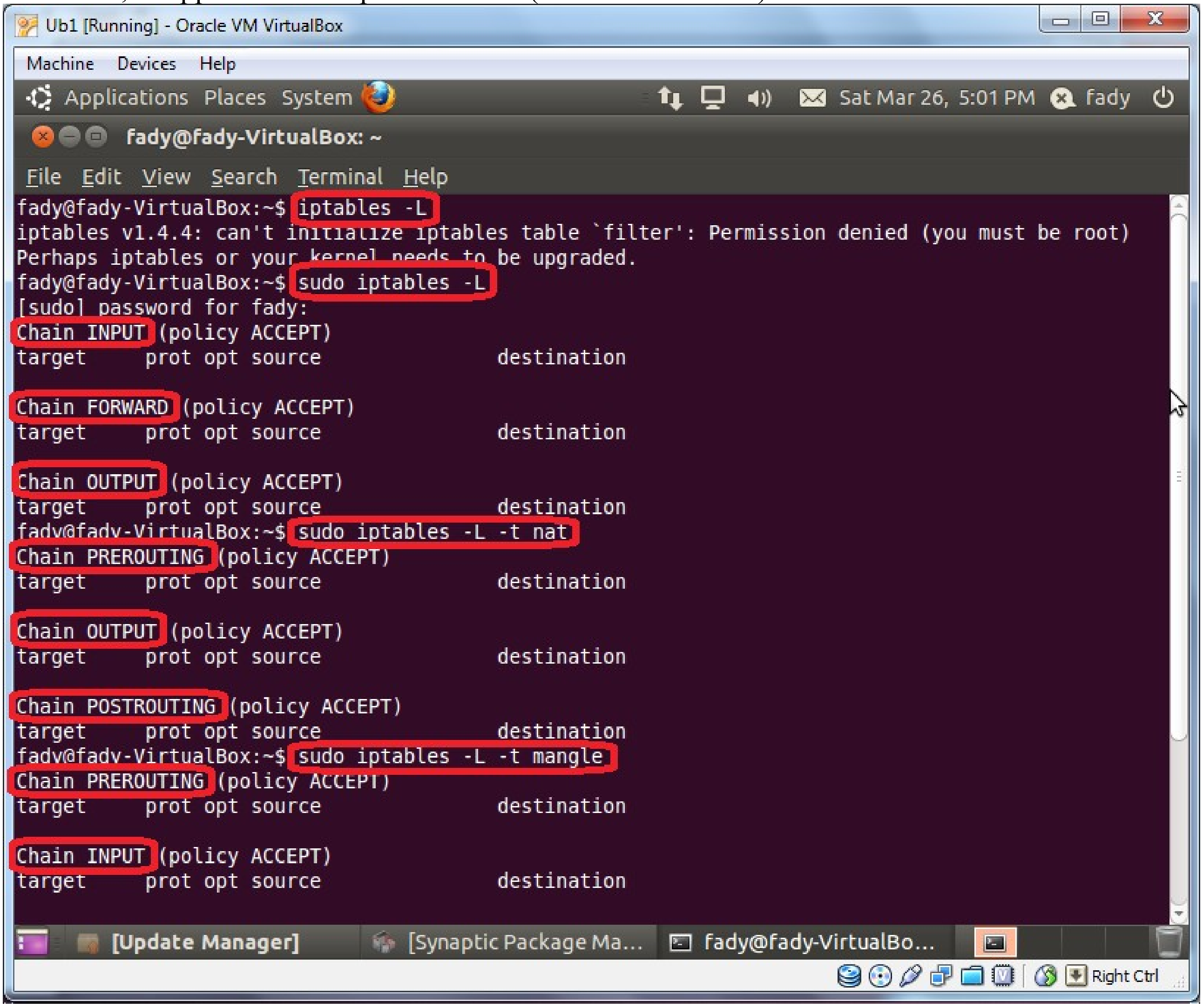


```
fady@fady-VirtualBox:~$ iptables -h
iptables v1.4.4

Usage: iptables -[AD] chain rule-specification [options]
       iptables -I chain [rulenum] rule-specification [options]
       iptables -R chain rulenum rule-specification [options]
       iptables -D chain rulenum [options]
       iptables -[LS] [chain [rulenum]] [options]
       iptables -[FZ] [chain] [options]
       iptables -[NX] chain
       iptables -E old-chain-name new-chain-name
       iptables -P chain target [options]
       iptables -h (print this help information)

Commands:
Either long or short options are allowed.
  --append  -A chain            Append to chain
  --delete  -D chain            Delete matching rule from chain
  --delete  -D chain rulenum
                                Delete rule rulenum (1 = first) from chain
  --insert  -I chain [rulenum]
                                Insert in chain as rulenum (default 1=first)
  --replace -R chain rulenum
                                Replace rule rulenum (1 = first) in chain
```

Options 2

-L, --list -t table-name

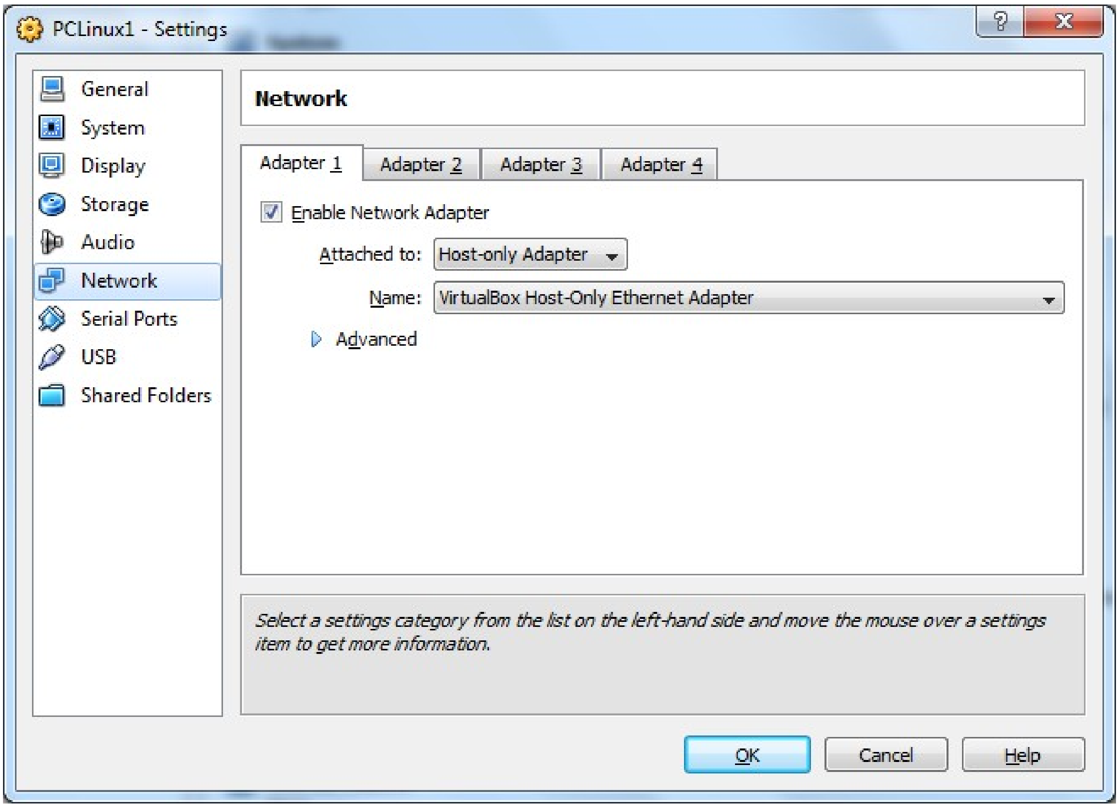
List all chains in the selected table. If no table is selected, like every other iptables command, it applies to the specified table (filter is the default).



```
fady@fady-VirtualBox:~$ iptables -L
iptables v1.4.4: can't initialize iptables table `filter': Permission denied (you must be root)
Perhaps iptables or your kernel needs to be upgraded.
fady@fady-VirtualBox:~$ sudo iptables -L
[sudo] password for fady:
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
fady@fady-VirtualBox:~$ sudo iptables -L -t nat
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
fady@fady-VirtualBox:~$ sudo iptables -L -t mangle
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

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

Scenario

We will use the VirtualBox as our virtual machine to create two guest machines (Ubuntu and PCLinux) both of them with network card configured with the virtual LAN that installed by the VirtualBox.



PCLinux1 - Settings

Network

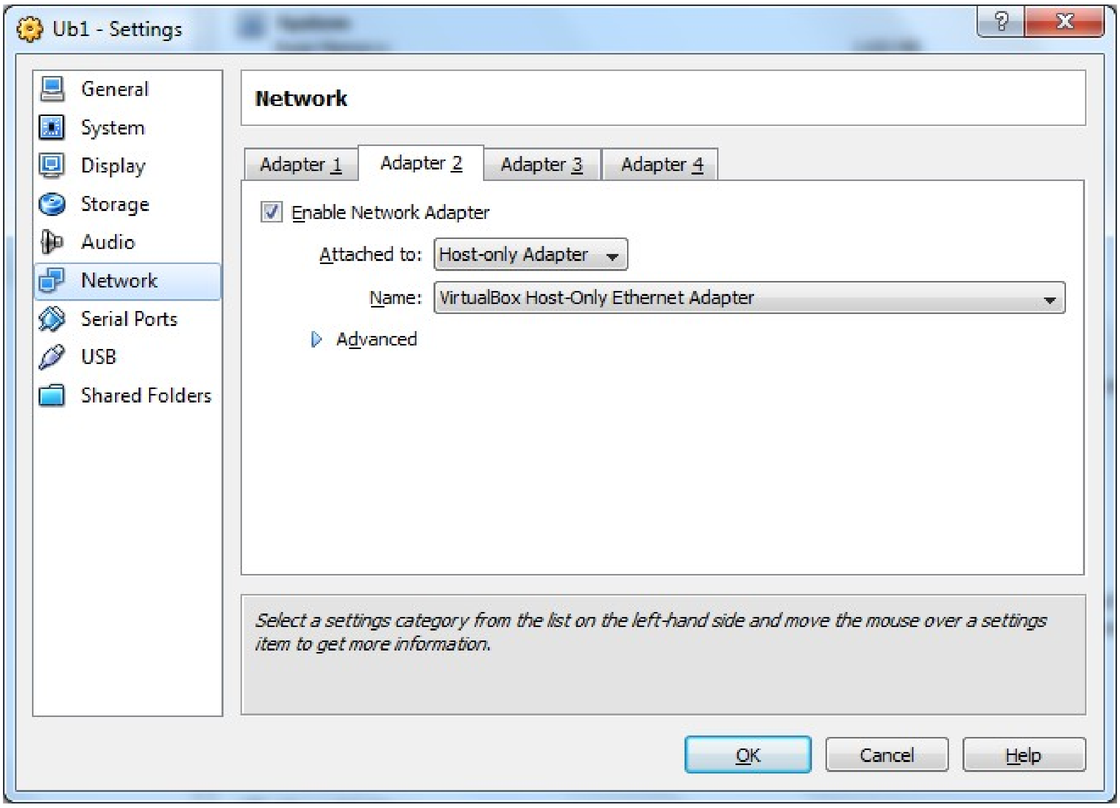
☑ Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter

Advanced

Select a settings category from the list on the left-hand side and move the mouse over a settings item to get more information.



Ub1 - Settings

Network

☑ Enable Network Adapter

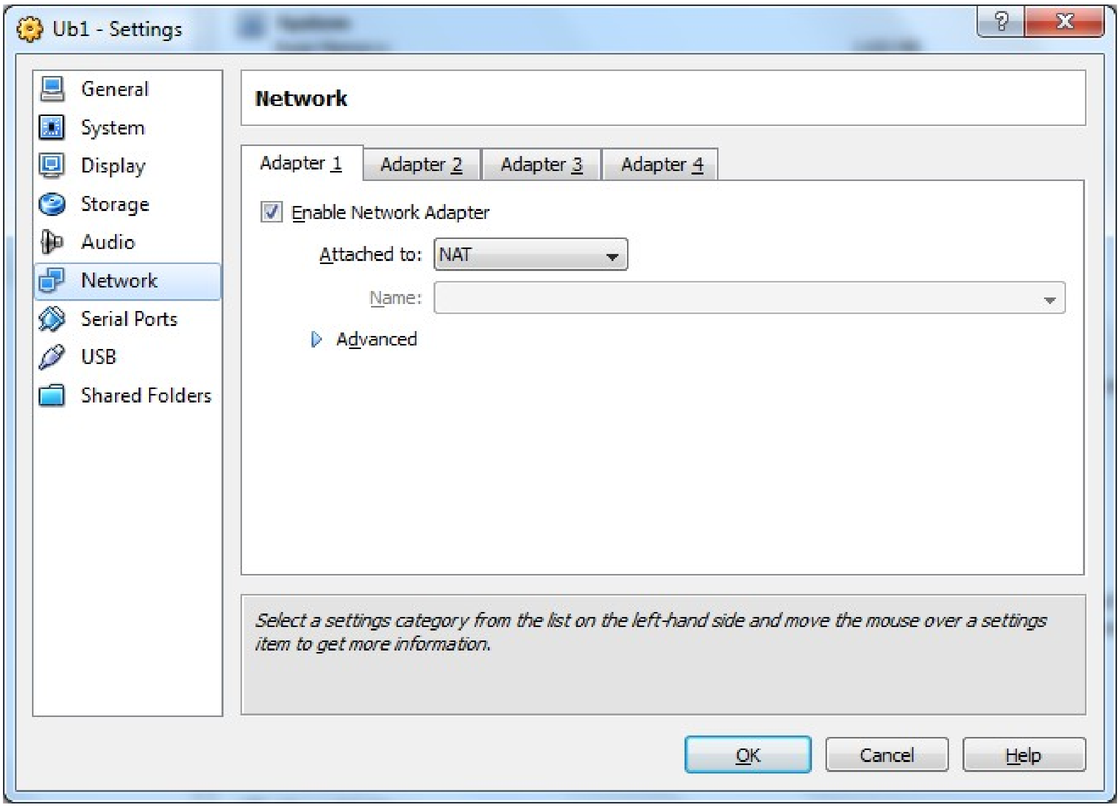
Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter

Advanced

Select a settings category from the list on the left-hand side and move the mouse over a settings item to get more information.

while Ubuntu machine has another network card configured with NATing to be certain that it will have an Internet connection from the host machine.



☑ Enable Network Adapter

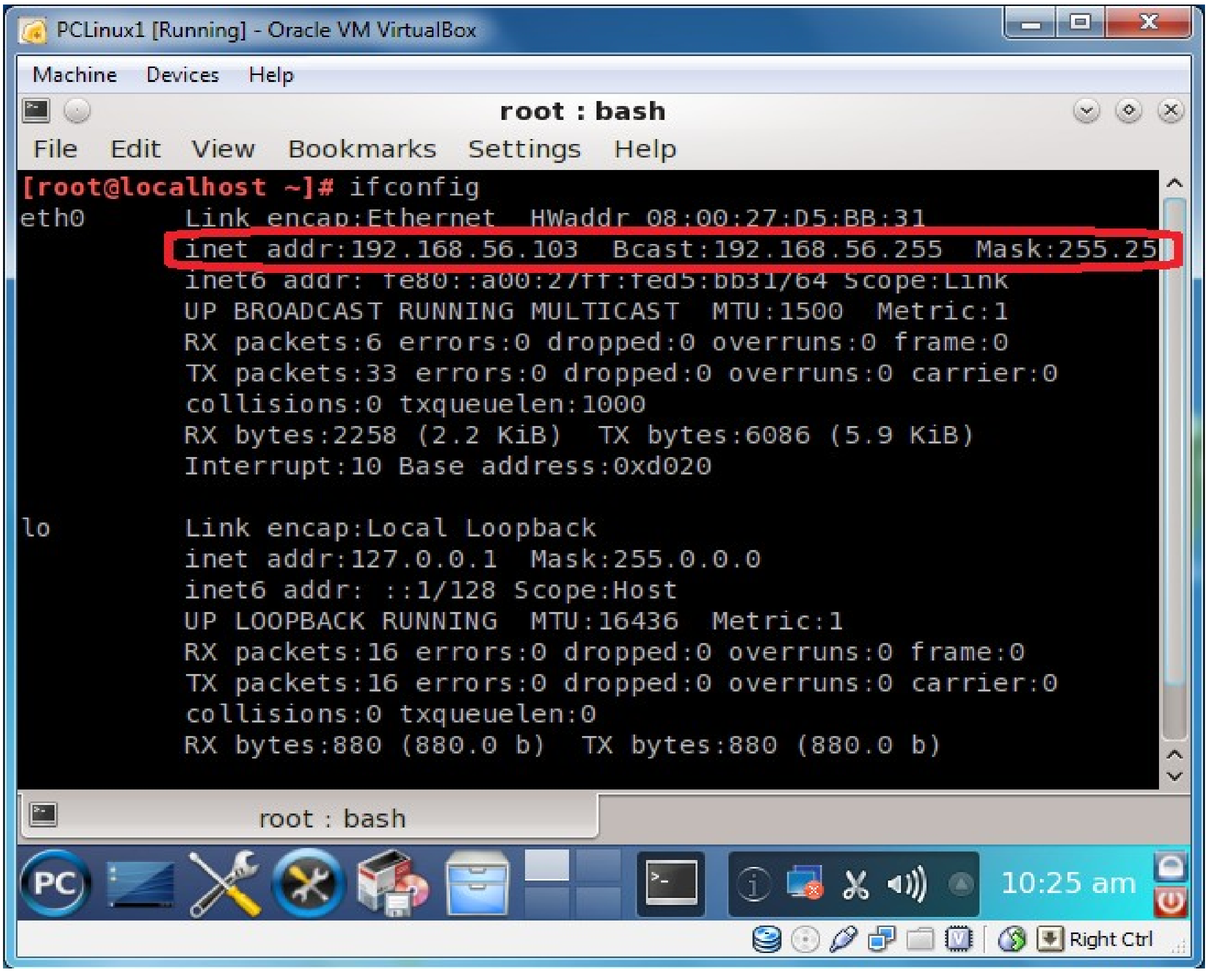
Attached to: NAT

Name:

Select a settings category from the list on the left-hand side and move the mouse over a settings item to get more information.

We have to be certain all the three machines are in the same LAN

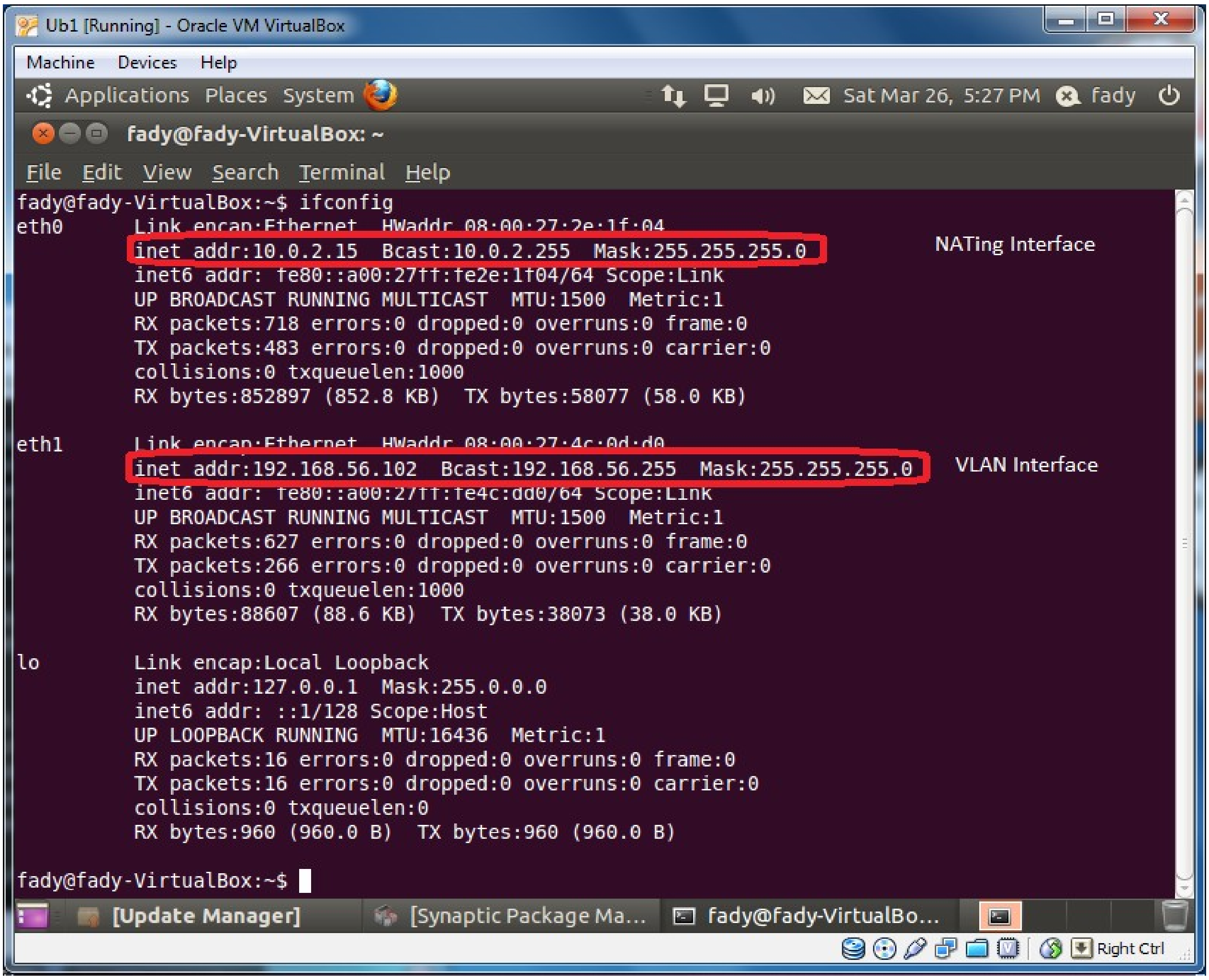
PCLinux Machine with one Network Card Named eth0 for Virtual LAN



PCLinux1 [Running] - Oracle VM VirtualBox
Machine Devices Help
root : bash
File Edit View Bookmarks Settings Help
root : bash
10:25 am
Right Ctrl

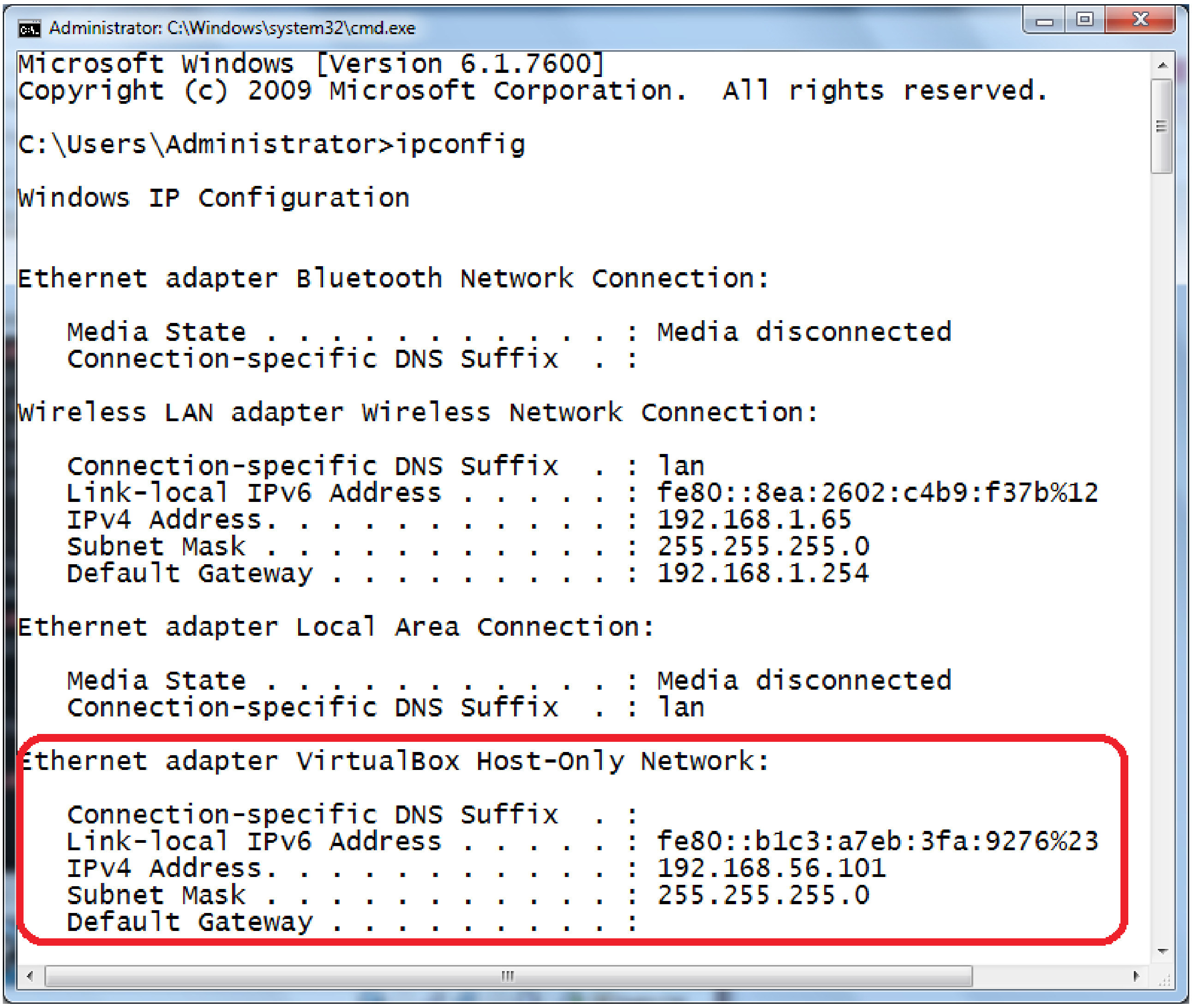
```
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:D5:BB:31
          inet addr:192.168.56.103  Bcast:192.168.56.255  Mask:255.25
          inet6 addr: fe80::a00:27ff:fed5:bb31/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:33 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2258 (2.2 KiB)  TX bytes:6086 (5.9 KiB)
          Interrupt:10 Base address:0xd020

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:880 (880.0 b)  TX bytes:880 (880.0 b)
```

Ubuntu Machine with Network Card called eth0 for NATing
and another Network Card called eth1 for Virtual LAN



```
fady@fady-VirtualBox:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:2e:1f:04
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe2e:1f04/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:718 errors:0 dropped:0 overruns:0 frame:0
          TX packets:483 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:852897 (852.8 KB)  TX bytes:58077 (58.0 KB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:4c:0d:d0
          inet addr:192.168.56.102  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe4c:dd0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:627 errors:0 dropped:0 overruns:0 frame:0
          TX packets:266 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:88607 (88.6 KB)  TX bytes:38073 (38.0 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:960 (960.0 B)  TX bytes:960 (960.0 B)

fady@fady-VirtualBox:~$ 
```

Host Machine with a lot of Network Cards
One is called Virtual Box Host-Only Adapter for Virtual LAN



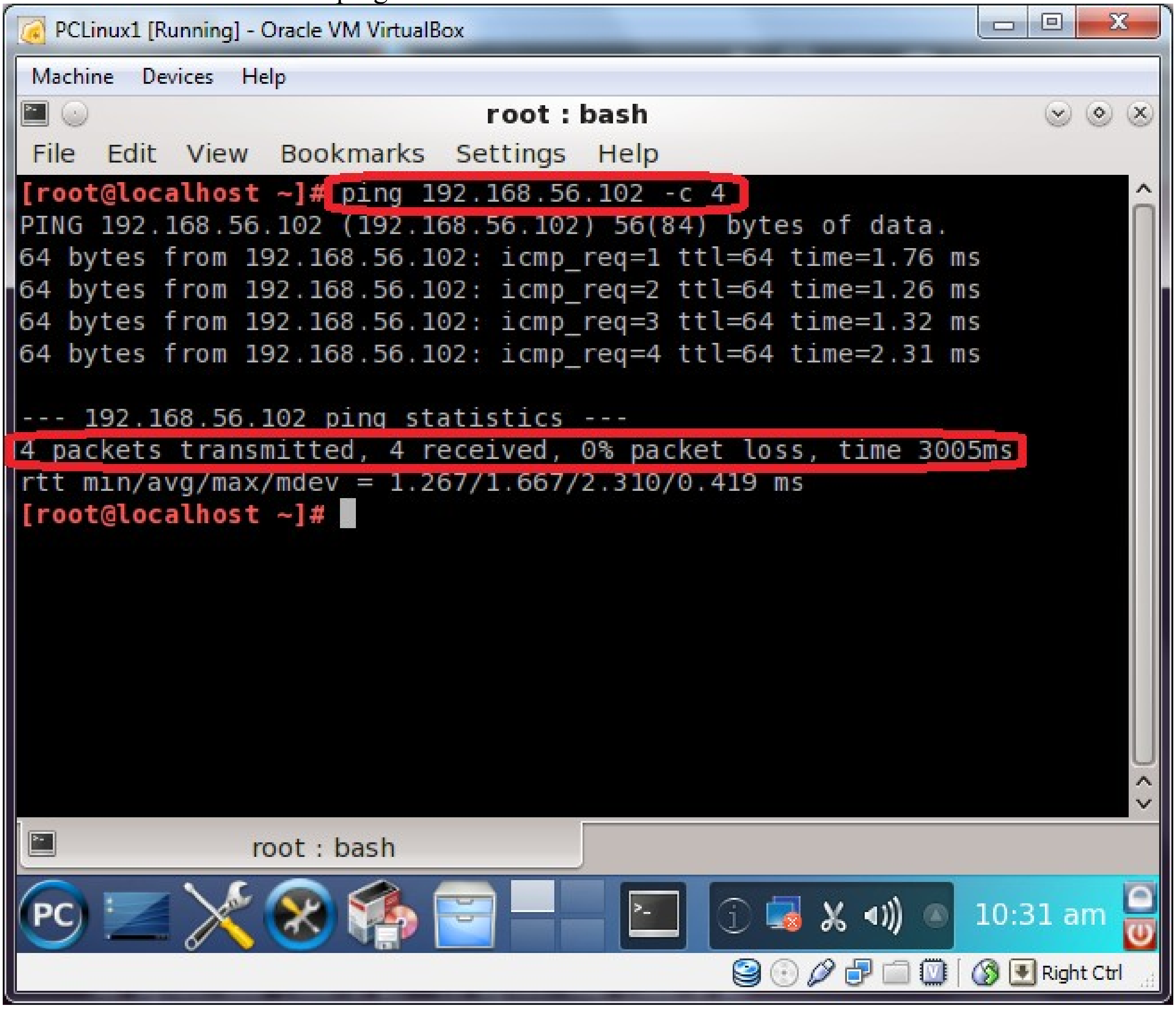
```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Administrator>ipconfig

Windows IP Configuration


Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wireless Network Connection:

   Connection-specific DNS Suffix  . : lan
   Link-local IPv6 Address . . . . . : fe80::8ea:2602:c4b9:f37b%12
   IPv4 Address. . . . . . . . . . . : 192.168.1.65
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.254

Ethernet adapter Local Area Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : lan

Ethernet adapter VirtualBox Host-Only Network:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::b1c3:a7eb:3fa:9276%23
   IPv4 Address. . . . . . . . . . . : 192.168.56.101
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
```

Filter Table (INPUT & OUTPUT Chains)

Example#1

Write a firewall rule to prevent pinging the Ubuntu machine from the PCLinux machine.

Now PCLinux Machine can ping Ubuntu Machine

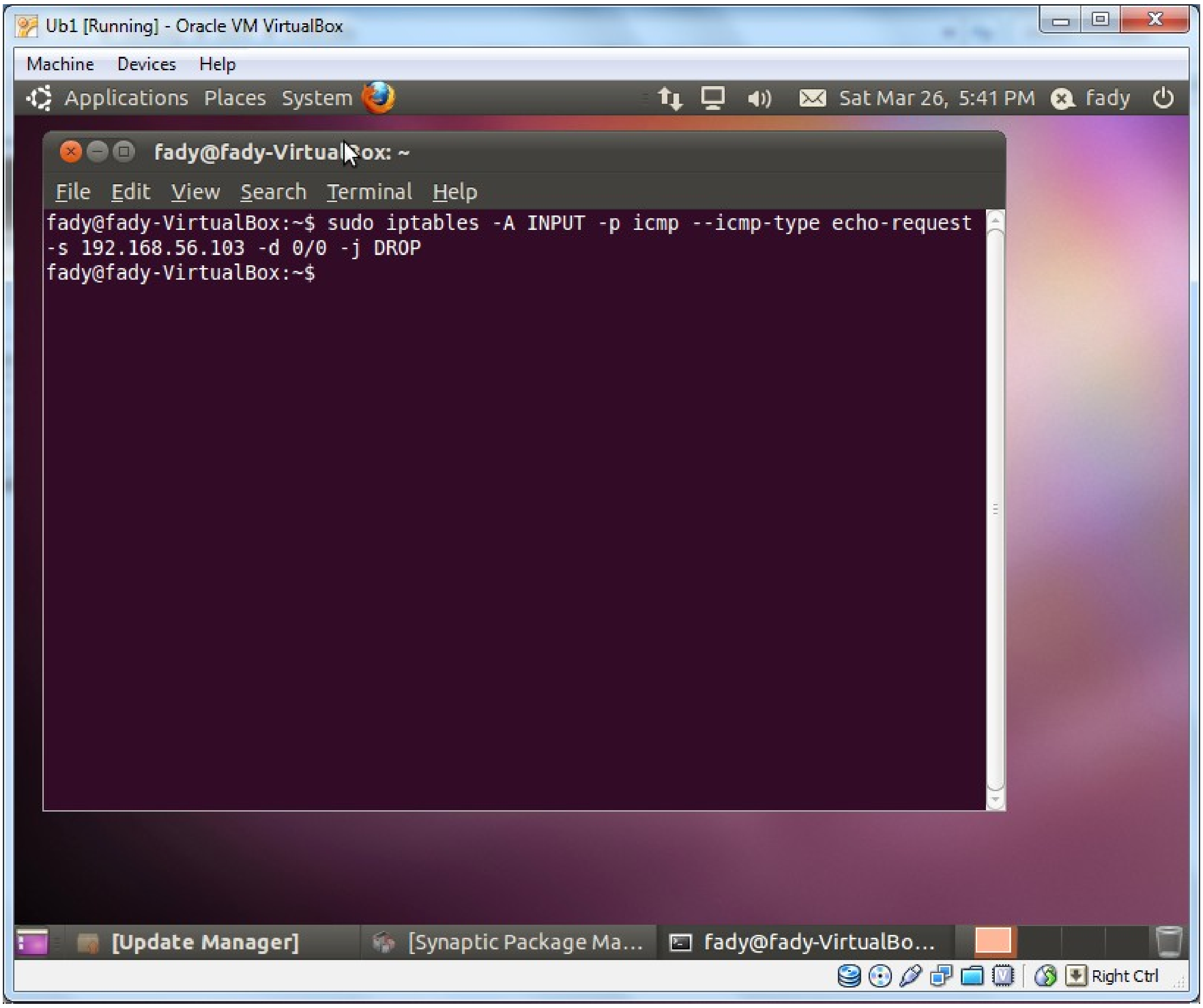


PCLinux1 [Running] - Oracle VM VirtualBox
Machine Devices Help
root : bash
File Edit View Bookmarks Settings Help
root : bash
10:31 am
Right Ctrl

```
[root@localhost ~]# ping 192.168.56.102 -c 4
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_req=1 ttl=64 time=1.76 ms
64 bytes from 192.168.56.102: icmp_req=2 ttl=64 time=1.26 ms
64 bytes from 192.168.56.102: icmp_req=3 ttl=64 time=1.32 ms
64 bytes from 192.168.56.102: icmp_req=4 ttl=64 time=2.31 ms

--- 192.168.56.102 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.267/1.667/2.310/0.419 ms
[root@localhost ~]#
```

Now if we just write the shown command, it will prevent the PCLinux Machine from pinging the Ubuntu Machine

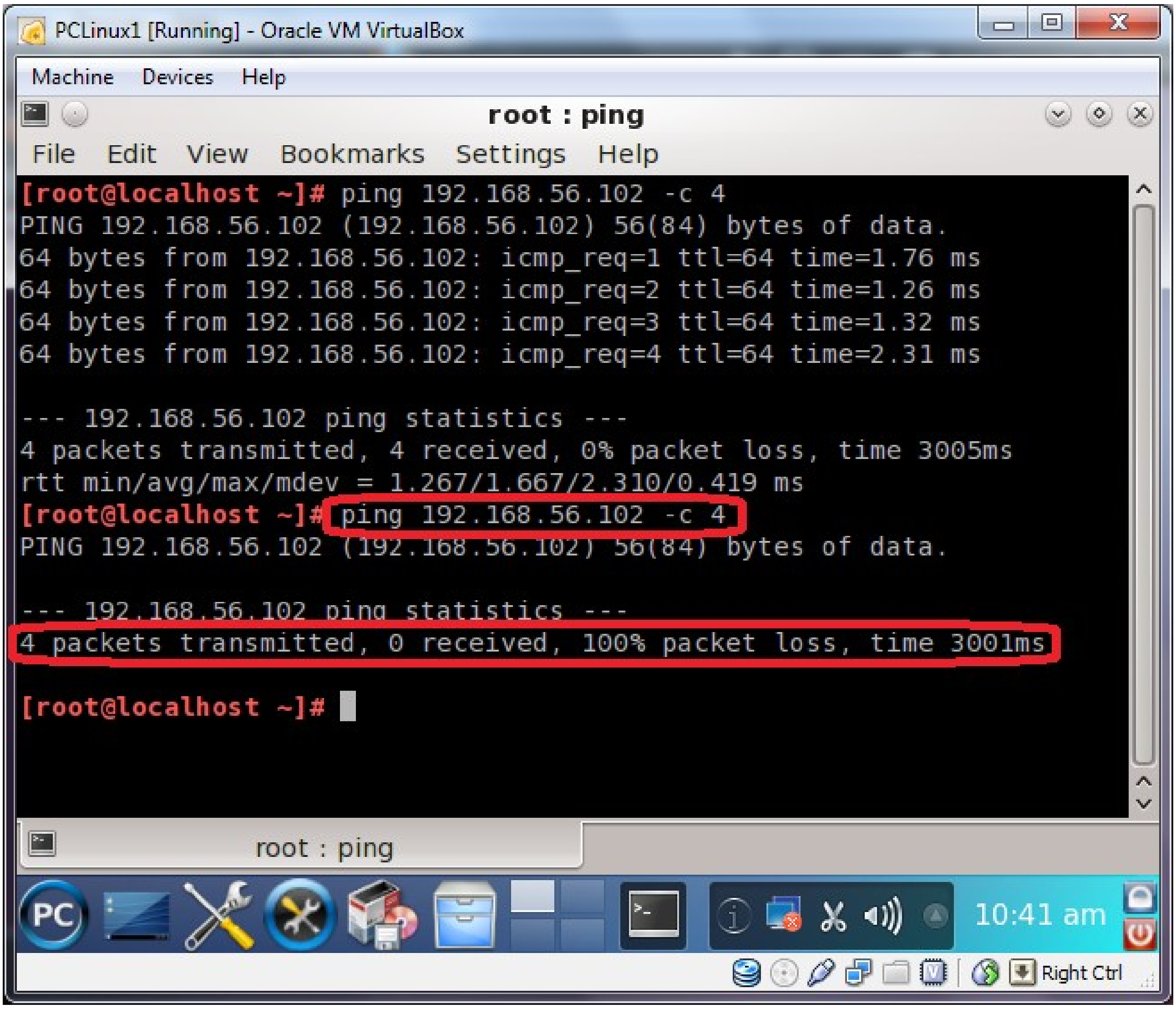
$sudo iptables -A INPUT -p icmp --icmp-type echo-request -s 192.168.56.103 -d 0/0 -j DROP



```
fady@fady-VirtualBox:~$ sudo iptables -A INPUT -p icmp --icmp-type echo-request
-s 192.168.56.103 -d 0/0 -j DROP
fady@fady-VirtualBox:~$
```

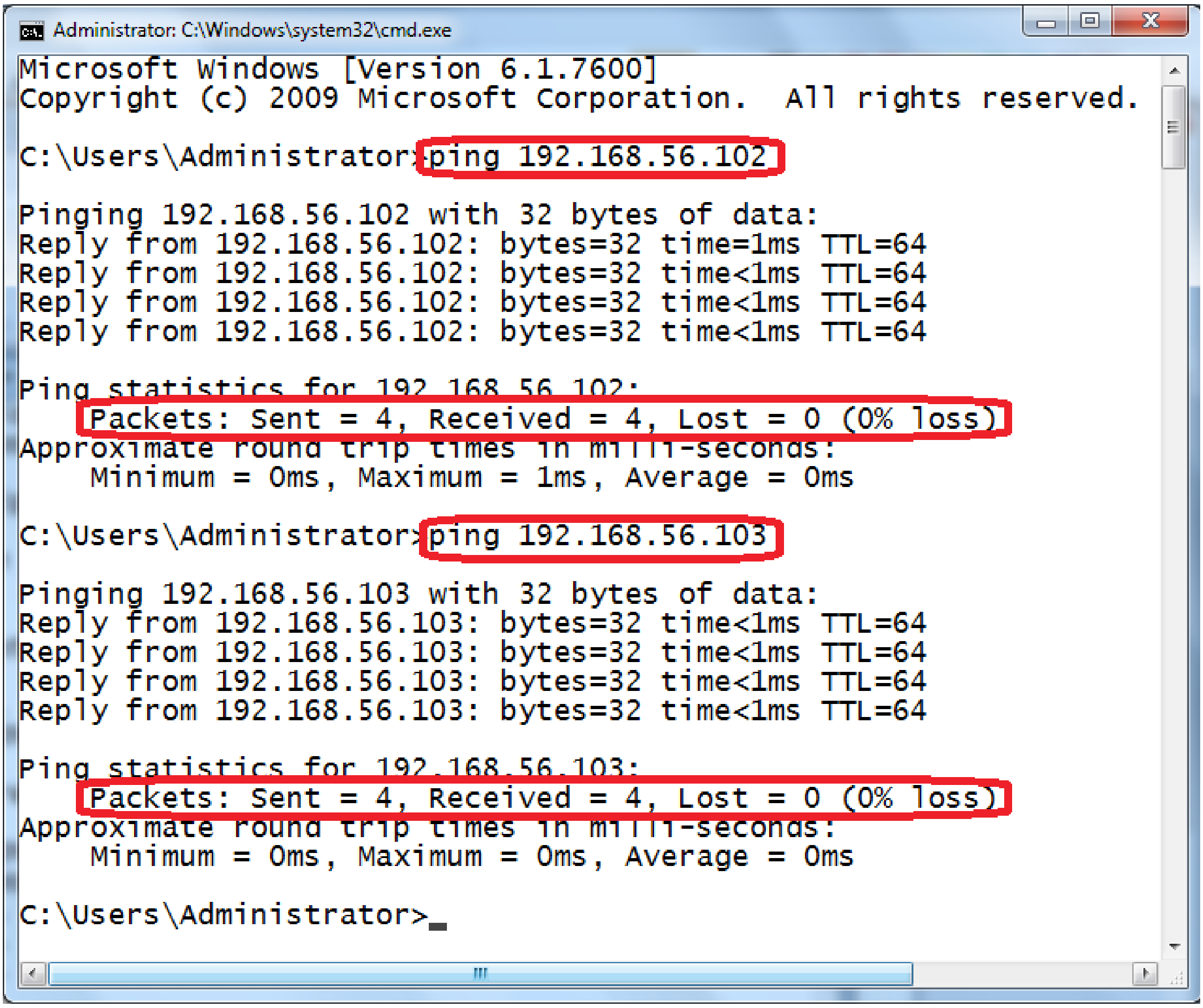
sudo	To take the root privilege.
iptables	To start using the iptables.
-A INPUT	To add a rule to a specified chain, here we specify the INPUT chain. (as no table specified we deal with the default table which is the filter table).
-p icmp	To specify the used protocol, here we specify the ICMP Protocol
--icmp-type echo-request	To specify the ICMP type, here we choose the ping (echo-request). This is option can be used only if we choose icmp as our protocol.
-s 192.168.56.103	To specify the source, here we specify the PCLinux IP
-d 0/0	To specify the destination, here we choose 0/0 which mean all hosts IPs.
-j DROP	To specify the Target Action, here we choose DROP Action.

Now as shown the PCLinux Machine ping but no answer


PCLinux1 [Running] - Oracle VM VirtualBox
Machine Devices Help
root : ping
File Edit View Bookmarks Settings Help
root : ping
10:41 am
Right Ctrl

```
[root@localhost ~]# ping 192.168.56.102 -c 4
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_req=1 ttl=64 time=1.76 ms
64 bytes from 192.168.56.102: icmp_req=2 ttl=64 time=1.26 ms
64 bytes from 192.168.56.102: icmp_req=3 ttl=64 time=1.32 ms
64 bytes from 192.168.56.102: icmp_req=4 ttl=64 time=2.31 ms

--- 192.168.56.102 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.267/1.667/2.310/0.419 ms
[root@localhost ~]# ping 192.168.56.102 -c 4
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.

--- 192.168.56.102 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3001ms

[root@localhost ~]# 
```

While the host machine can ping both no problem

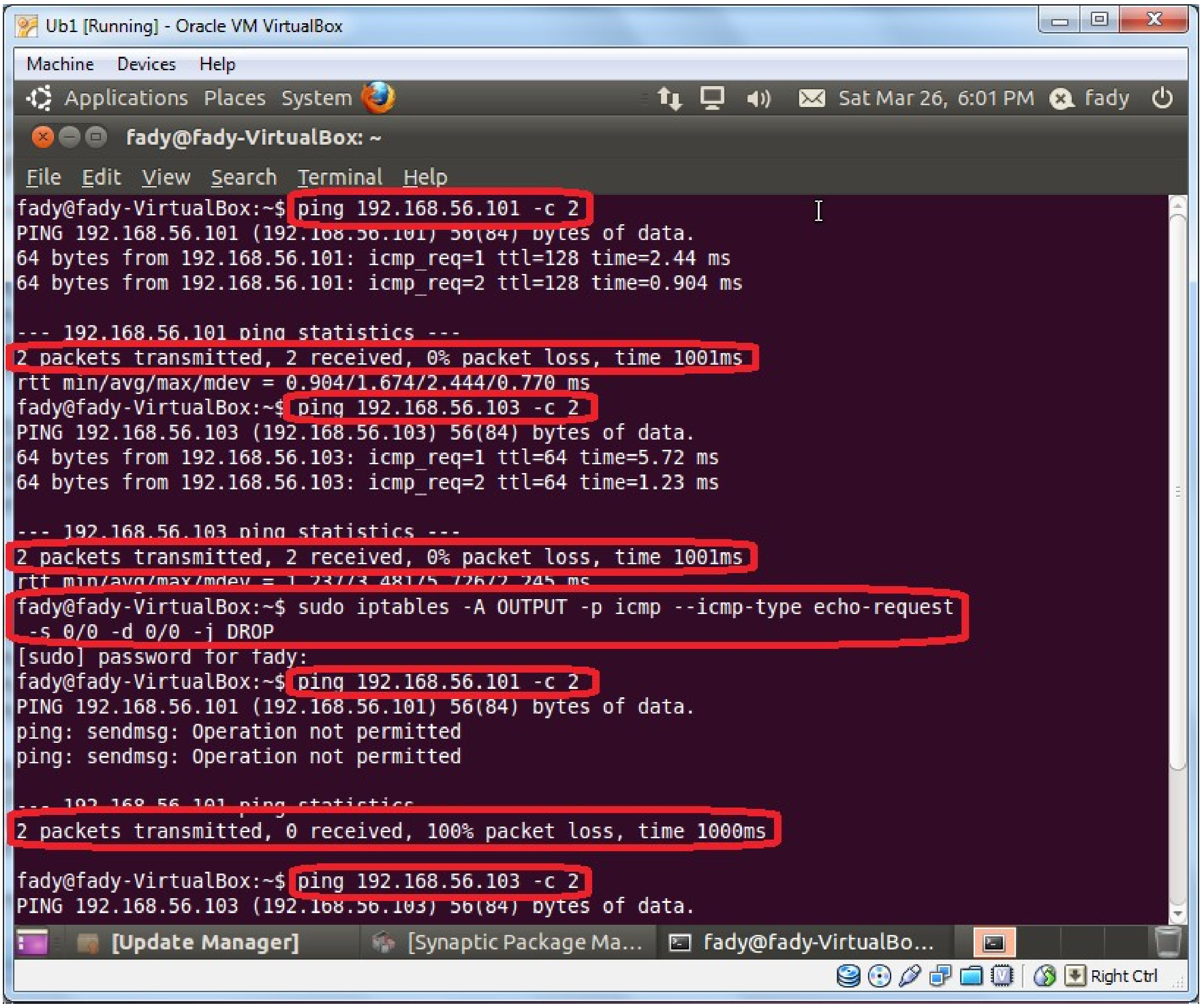


```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Administrator>ping 192.168.56.102

Pinging 192.168.56.102 with 32 bytes of data:
Reply from 192.168.56.102: bytes=32 time=1ms TTL=64
Reply from 192.168.56.102: bytes=32 time<1ms TTL=64
Reply from 192.168.56.102: bytes=32 time<1ms TTL=64
Reply from 192.168.56.102: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.56.102:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\Administrator>ping 192.168.56.103

Pinging 192.168.56.103 with 32 bytes of data:
Reply from 192.168.56.103: bytes=32 time<1ms TTL=64
Reply from 192.168.56.103: bytes=32 time<1ms TTL=64
Reply from 192.168.56.103: bytes=32 time<1ms TTL=64
Reply from 192.168.56.103: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.56.103:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Administrator>_
```

Example#2

Write a firewall rule to prevent the Ubuntu Machine from pinging any other computer on the network.

```
$sudo iptables -A OUTPUT -p icmp --icmp-type echo-request -s 0/0 -d 0/0 -j DROP
```

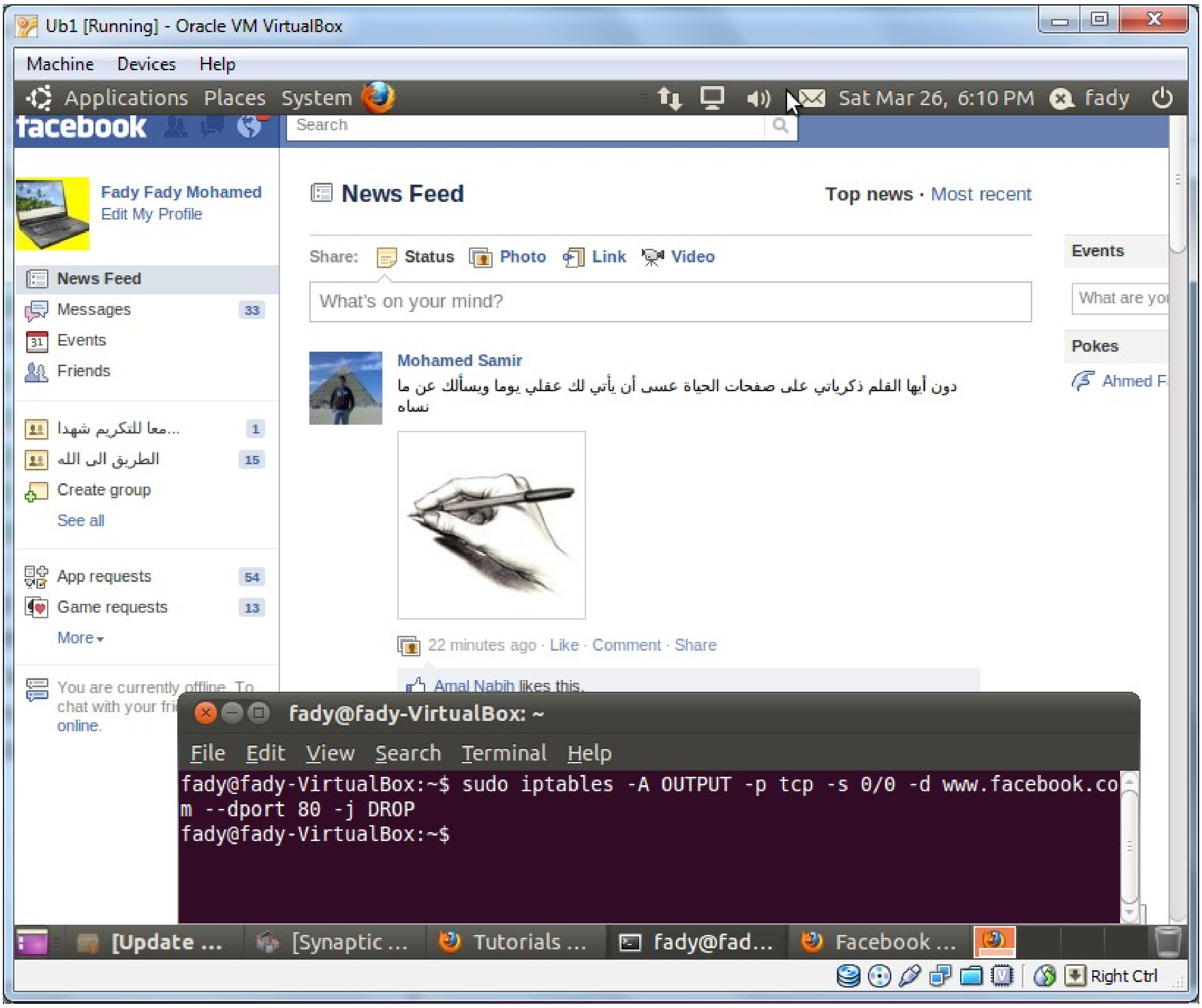


```
fady@fady-VirtualBox:~$ ping 192.168.56.101 -c 2
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 bytes from 192.168.56.101: icmp_req=1 ttl=128 time=2.44 ms
64 bytes from 192.168.56.101: icmp_req=2 ttl=128 time=0.904 ms

--- 192.168.56.101 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.904/1.674/2.444/0.770 ms
fady@fady-VirtualBox:~$ ping 192.168.56.103 -c 2
PING 192.168.56.103 (192.168.56.103) 56(84) bytes of data.
64 bytes from 192.168.56.103: icmp_req=1 ttl=64 time=5.72 ms
64 bytes from 192.168.56.103: icmp_req=2 ttl=64 time=1.23 ms

--- 192.168.56.103 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.237/3.481/5.726/2.245 ms
fady@fady-VirtualBox:~$ sudo iptables -A OUTPUT -p icmp --icmp-type echo-request
 -s 0/0 -d 0/0 -j DROP
[sudo] password for fady:
fady@fady-VirtualBox:~$ ping 192.168.56.101 -c 2
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted

--- 192.168.56.101 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1000ms

fady@fady-VirtualBox:~$ ping 192.168.56.103 -c 2
PING 192.168.56.103 (192.168.56.103) 56(84) bytes of data.
```

Example#3

Write a firewall rule to prevent the Ubuntu Machine from opening the facebook

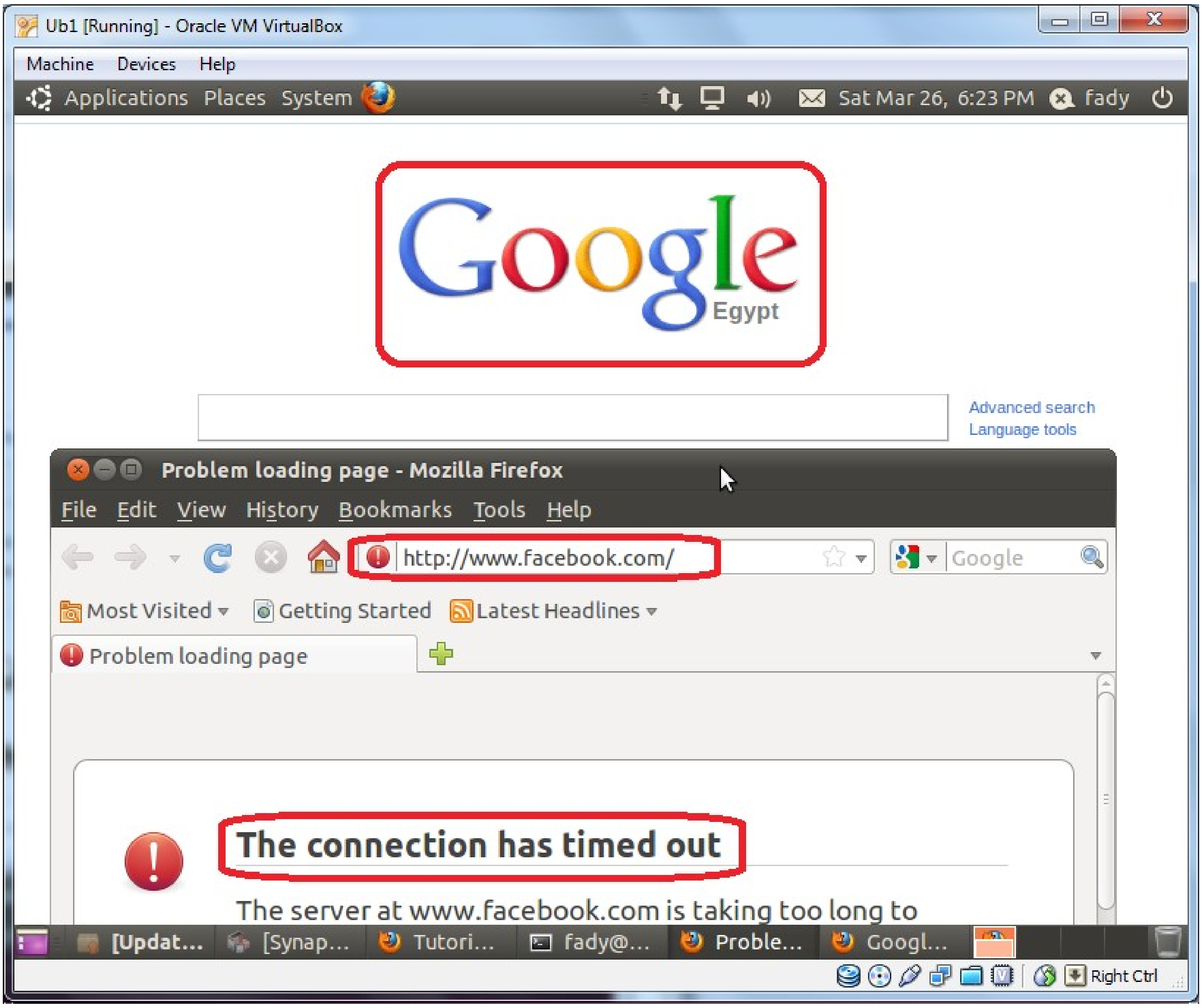
$sude iptables -A OUTPUT -p tcp -s 0/0 -d www.facebook.com --dport 80 -j DROP



دون أيها القلم ذكرياتي على صفحات الحياة عسى أن يأتي لك عقلي يوما ويسألك عن ما
نساه

```
fady@fady-VirtualBox:~$ sudo iptables -A OUTPUT -p tcp -s 0/0 -d www.facebook.co
m --dport 80 -j DROP
fady@fady-VirtualBox:~$
```

After adding the rule, the Ubuntu Machine can't access the facebook website while other websites are aavailable like google.com.



Ub1 [Running] - Oracle VM VirtualBox
Machine Devices Help
Applications Places System
Sat Mar 26, 6:23 PM
fady
Google
Egypt
Advanced search
Language tools
Problem loading page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.facebook.com/
Google
Most Visited
Getting Started
Latest Headlines
Problem loading page
[Updat… [Synap… Tutori… fady@… Proble… Googl…
Right Ctrl

The connection has timed out

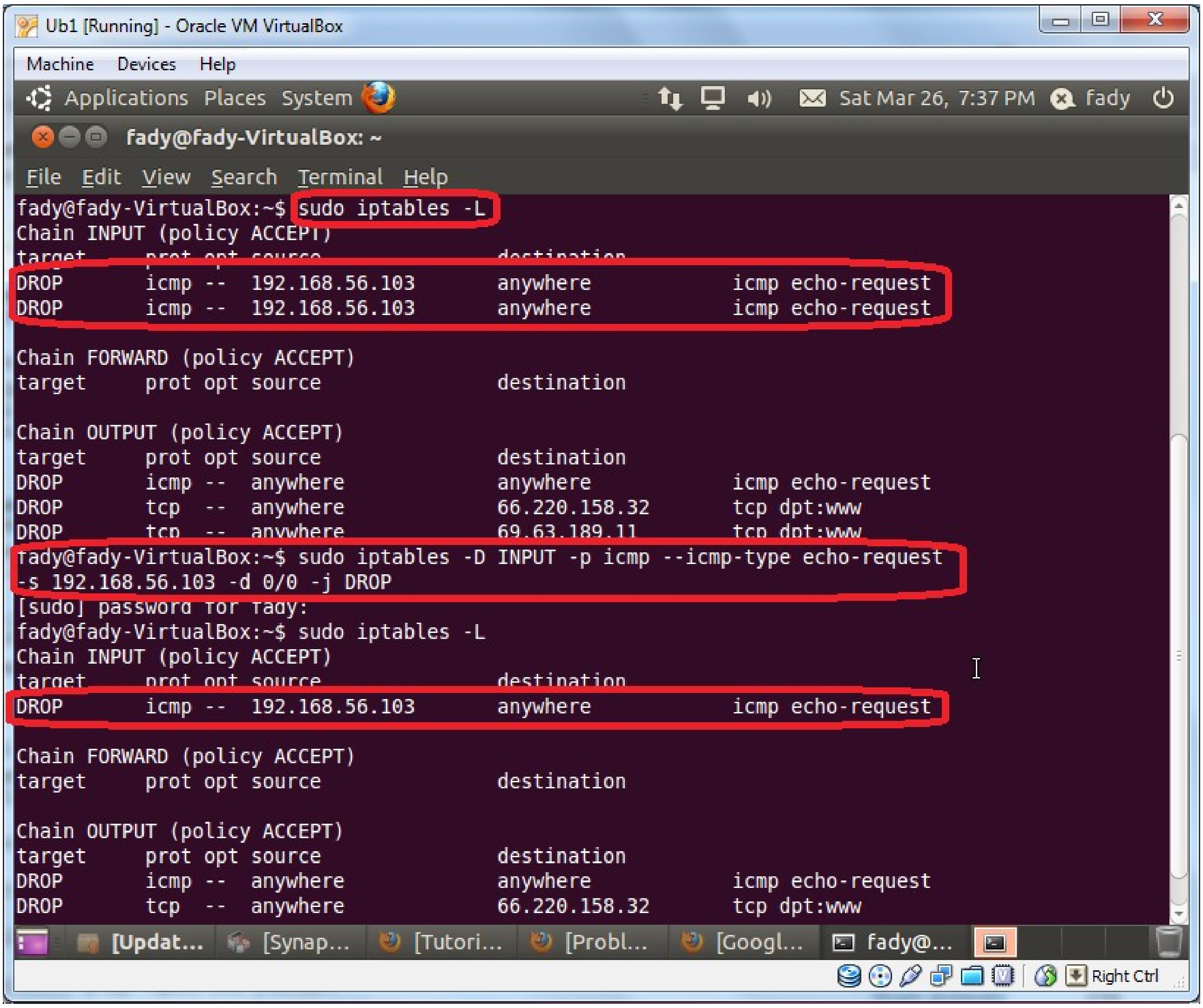
The server at www.facebook.com is taking too long to

Example#4

Delete the firewall rule that we used in Example#1

$sudo iptables -D INPUT -p icmp --icmp-type echo-request -s 192.168.56.103 -d 0/0 -j DROP

It is the same like adding except using -D instead of -A



Ub1 [Running] - Oracle VM VirtualBox
Machine Devices Help
Applications Places System
Sat Mar 26, 7:37 PM
fady
fady@fady-VirtualBox: ~
File Edit View Search Terminal Help
[Updat... [Synap... [Tutori... [Probl... [Googl... fady@...
Right Ctrl

```
fady@fady-VirtualBox:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  192.168.56.103       anywhere            icmp echo-request
DROP       icmp --  192.168.56.103       anywhere            icmp echo-request

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  anywhere             anywhere            icmp echo-request
DROP       tcp  --  anywhere             66.220.158.32       tcp dpt:www
DROP       tcp  --  anywhere             69.63.189.11        tcp dpt:www
fady@fady-VirtualBox:~$ sudo iptables -D INPUT -p icmp --icmp-type echo-request
-s 192.168.56.103 -d 0/0 -j DROP
[sudo] password for fady:
fady@fady-VirtualBox:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  192.168.56.103       anywhere            icmp echo-request

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  anywhere             anywhere            icmp echo-request
DROP       tcp  --  anywhere             66.220.158.32       tcp dpt:www
```

Exercise#1

Write a firewall rule to prevent the Ubuntu Machine from opening any website.

Exercise#2

Write firewall rules to prevent any client except the PCLinux Machine from accessing the web server on the Ubuntu Machine.