Final Project

Nicholas Somerville

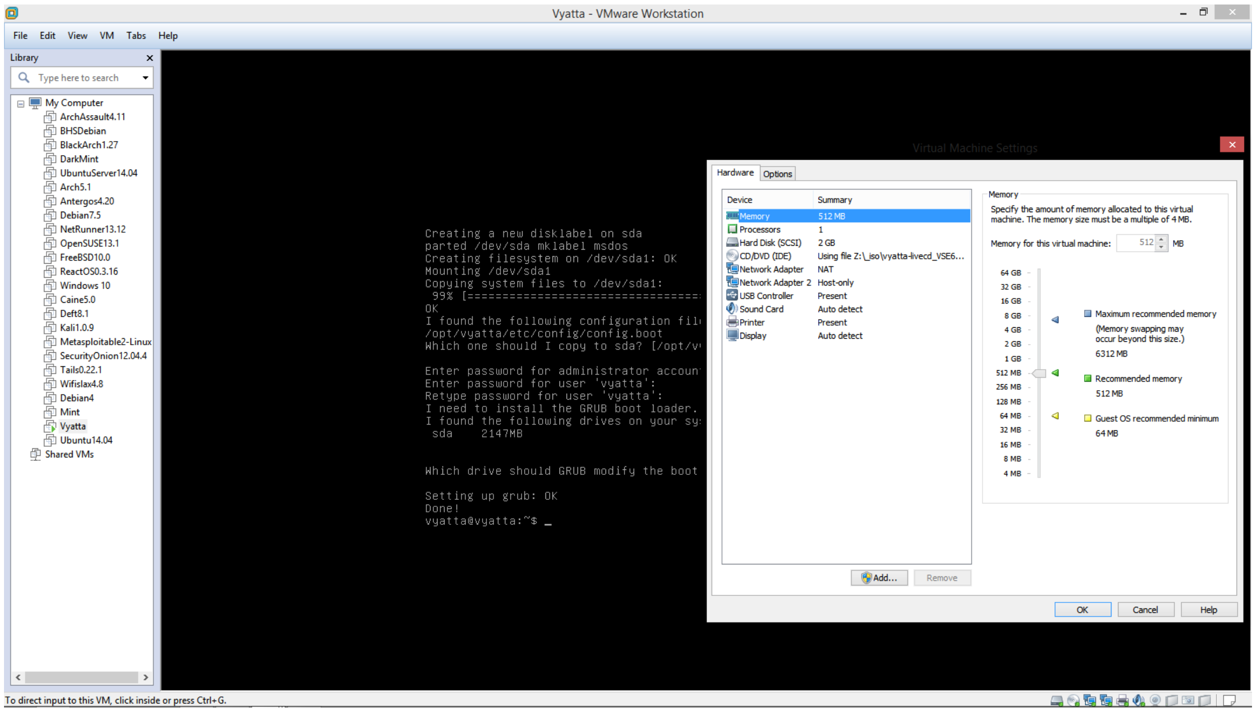
University of Advancing Technology

TABLE OF CONTENTS

Lab Memorandum

Professor:	**Barrett**
Course:	**NTW415, Network Defense & Countermeasures**
Student:	**Nick Somerville**
Date:	**22-Jan-15**
Lab / Activity:	**Assignment 1.1 – Creating a Virtual Network Appliance**

1) Set up of Vyatta VM with 512MB RAM, 2GB storage, and 2 network adapters: NAT (first) and host-only (second). Then, I installed the Vyatta system, cut power to system with command "sudo halt," removed the disc from settings by switching to physical drive, and booted.



```
Creating a new disklabel on sda
parted /dev/sda mklabel msdos
Creating filesystem on /dev/sda1: OK
Mounting /dev/sda1
Copying system files to /dev/sda1:
 99% [==================================
OK
I found the following configuration fil
/opt/vyatta/etc/config/config.boot
Which one should I copy to sda? [/opt/v

Enter password for administrator accoun
Enter password for user 'vyatta':
Retype password for user 'vyatta':
I need to install the GRUB boot loader.
I found the following drives on your sy
 sda    2147MB


Which drive should GRUB modify the boot

Setting up grub: OK
Done!
vyatta@vyatta:~$ _
```

Device	Summary
Memory	512 MB
Processors	1
Hard Disk (SCSI)	2 GB
CD/DVD (IDE)	Using file Z:_iso\vyatta-livecd_VSE6...
Network Adapter	NAT
Network Adapter 2	Host-only
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Memory

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

Memory for this virtual machine: 512 MB

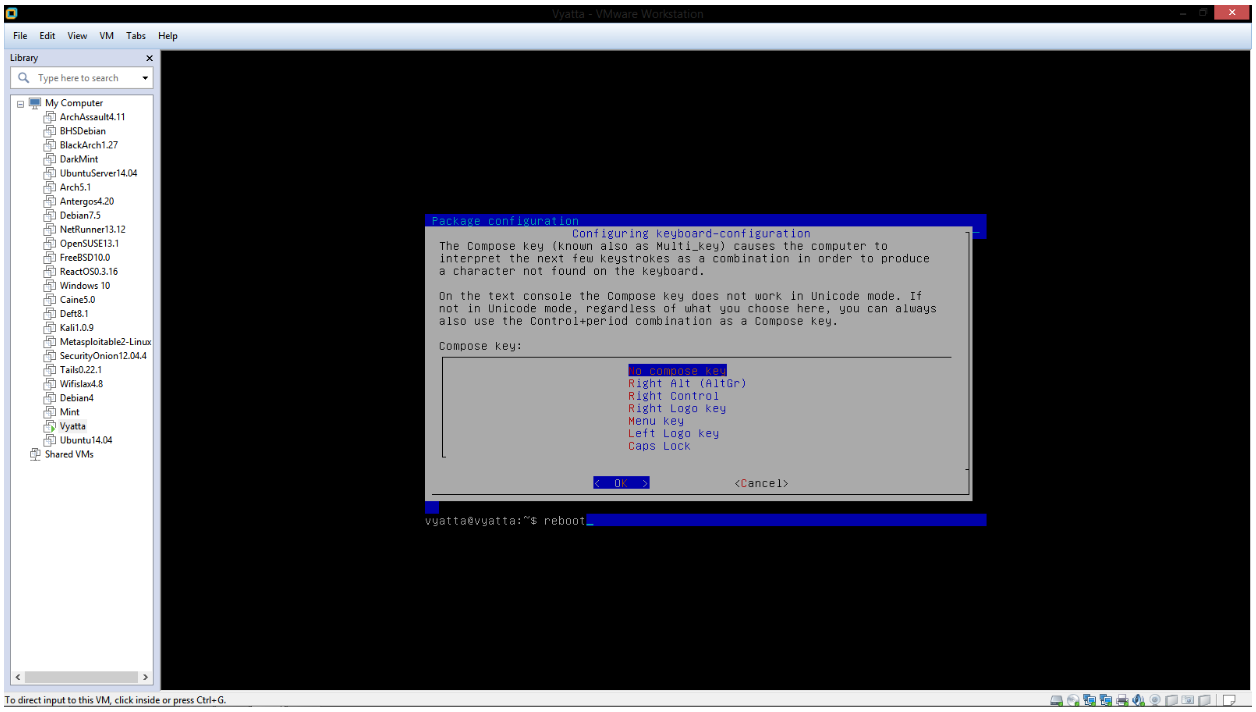
64 GB
32 GB
16 GB
8 GB
4 GB
2 GB
1 GB
512 MB
256 MB
128 MB
64 MB
32 MB
16 MB
8 MB
4 MB

🟦 Maximum recommended memory
(Memory swapping may occur beyond this size.)
6312 MB

🟩 Recommended memory
512 MB

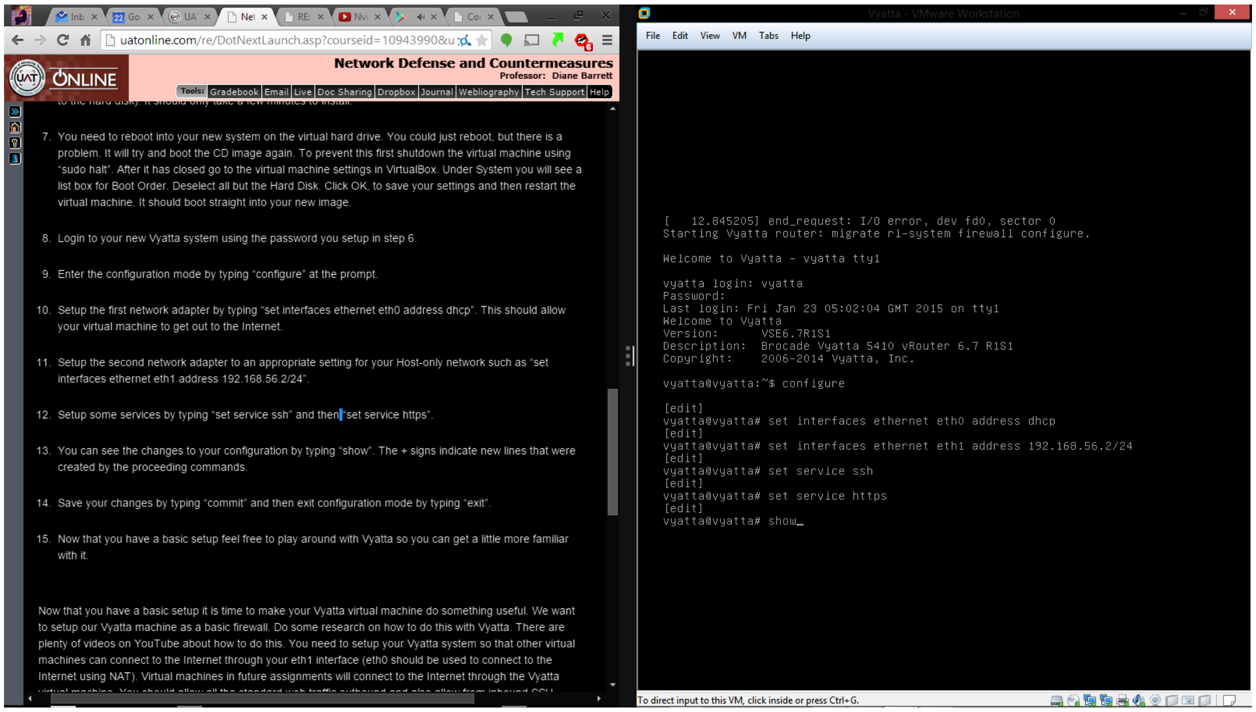
🟨 Guest OS recommended minimum
64 MB

2) Detour step: I'm a dvorak user, not qwerty. I entered the command "sudo dpkg-reconfigure keyboard-configuration" to switch to dvorak through the keyboard configuration GUI. In order for changes to apply, I reboot.



```
Package configuration
                 Configuring keyboard-configuration
  The Compose key (known also as Multi_key) causes the computer to
  interpret the next few keystrokes as a combination in order to produce
  a character not found on the keyboard.

  On the text console the Compose key does not work in Unicode mode. If
  not in Unicode mode, regardless of what you choose here, you can always
  also use the Control+period combination as a Compose key.

  Compose key:

                          No compose key
                          Right Alt (AltGr)
                          Right Control
                          Right Logo key
                          Menu key
                          Left Logo key
                          Caps Lock

                        <  Ok  >           <Cancel>

vyatta@vyatta:~$ reboot_
```

3) I enter the following commands as per instructions:
configure
set interfaces ethernet eth0 address dhcp
set interfaces ethernet eth1 address 192.168.56.2/24
set service ssh
set service https
show


uatonline.com/re/DotNextLaunch.asp?courseid=10943990&u
UAT ONLINE
Network Defense and Countermeasures
Professor: Diane Barrett
Tools: Gradebook Email Live Doc Sharing Dropbox Journal Webliography Tech Support Help
Vyatta - VMware Workstation
File Edit View VM Tabs Help
To direct input to this VM, click inside or press Ctrl+G.

7. You need to reboot into your new system on the virtual hard drive. You could just reboot, but there is a problem. It will try and boot the CD image again. To prevent this first shutdown the virtual machine using "sudo halt". After it has closed go to the virtual machine settings in VirtualBox. Under System you will see a list box for Boot Order. Deselect all but the Hard Disk. Click OK, to save your settings and then restart the virtual machine. It should boot straight into your new image.

8. Login to your new Vyatta system using the password you setup in step 6.

9. Enter the configuration mode by typing "configure" at the prompt.

10. Setup the first network adapter by typing "set interfaces ethernet eth0 address dhcp". This should allow your virtual machine to get out to the Internet.

11. Setup the second network adapter to an appropriate setting for your Host-only network such as "set interfaces ethernet eth1 address 192.168.56.2/24".

12. Setup some services by typing "set service ssh" and then "set service https".

13. You can see the changes to your configuration by typing "show". The + signs indicate new lines that were created by the proceeding commands.

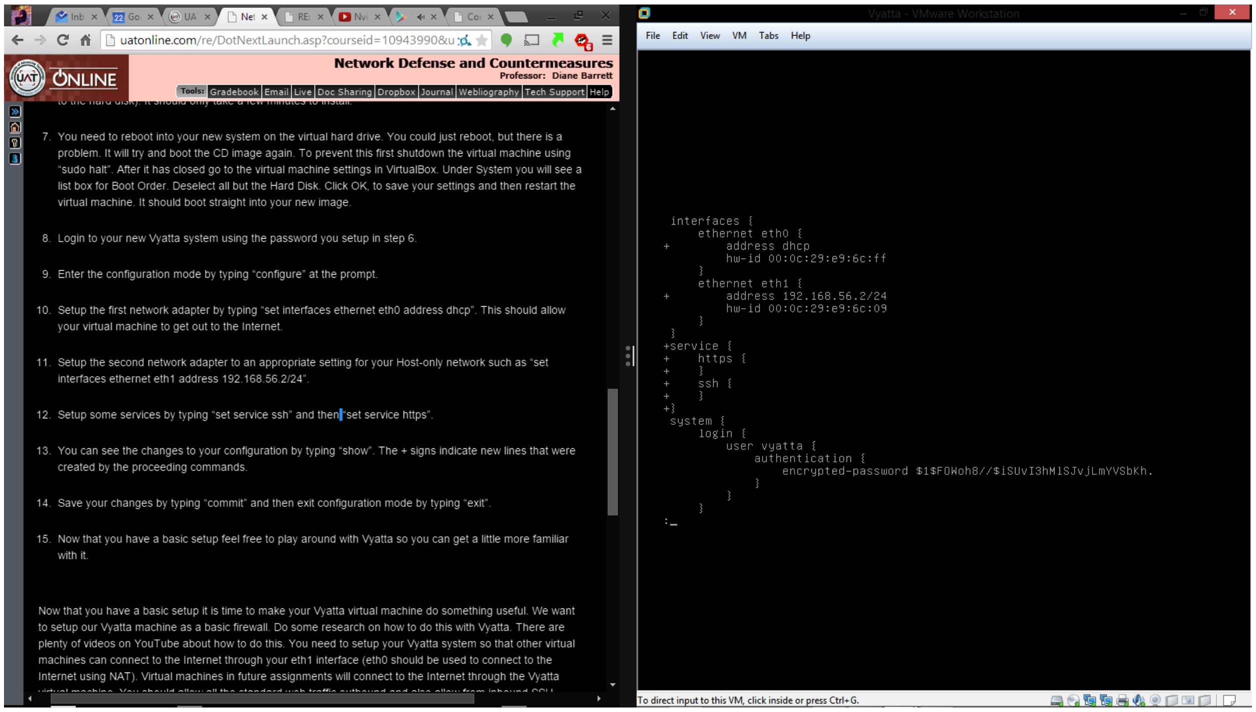
14. Save your changes by typing "commit" and then exit configuration mode by typing "exit".

15. Now that you have a basic setup feel free to play around with Vyatta so you can get a little more familiar with it.

Now that you have a basic setup it is time to make your Vyatta virtual machine do something useful. We want to setup our Vyatta machine as a basic firewall. Do some research on how to do this with Vyatta. There are plenty of videos on YouTube about how to do this. You need to setup your Vyatta system so that other virtual machines can connect to the Internet through your eth1 interface (eth0 should be used to connect to the Internet using NAT). Virtual machines in future assignments will connect to the Internet through the Vyatta

```
[   12.845205] end_request: I/O error, dev fd0, sector 0
Starting Vyatta router: migrate rl-system firewall configure.

Welcome to Vyatta - vyatta tty1

vyatta login: vyatta
Password:
Last login: Fri Jan 23 05:02:04 GMT 2015 on tty1
Welcome to Vyatta
Version:      VSE6.7R1S1
Description:  Brocade Vyatta 5410 vRouter 6.7 R1S1
Copyright:    2006-2014 Vyatta, Inc.

vyatta@vyatta:~$ configure

[edit]
vyatta@vyatta# set interfaces ethernet eth0 address dhcp
[edit]
vyatta@vyatta# set interfaces ethernet eth1 address 192.168.56.2/24
[edit]
vyatta@vyatta# set service ssh
[edit]
vyatta@vyatta# set service https
[edit]
vyatta@vyatta# show_
```

4) The results from the command “show”



Network Defense and Countermeasures
Professor: Diane Barrett

7. You need to reboot into your new system on the virtual hard drive. You could just reboot, but there is a problem. It will try and boot the CD image again. To prevent this first shutdown the virtual machine using “sudo halt”. After it has closed go to the virtual machine settings in VirtualBox. Under System you will see a list box for Boot Order. Deselect all but the Hard Disk. Click OK, to save your settings and then restart the virtual machine. It should boot straight into your new image.

8. Login to your new Vyatta system using the password you setup in step 6.

9. Enter the configuration mode by typing “configure” at the prompt.

10. Setup the first network adapter by typing “set interfaces ethernet eth0 address dhcp”. This should allow your virtual machine to get out to the Internet.

11. Setup the second network adapter to an appropriate setting for your Host-only network such as “set interfaces ethernet eth1 address 192.168.56.2/24”.

12. Setup some services by typing “set service ssh” and then “set service https”.

13. You can see the changes to your configuration by typing “show”. The + signs indicate new lines that were created by the proceeding commands.

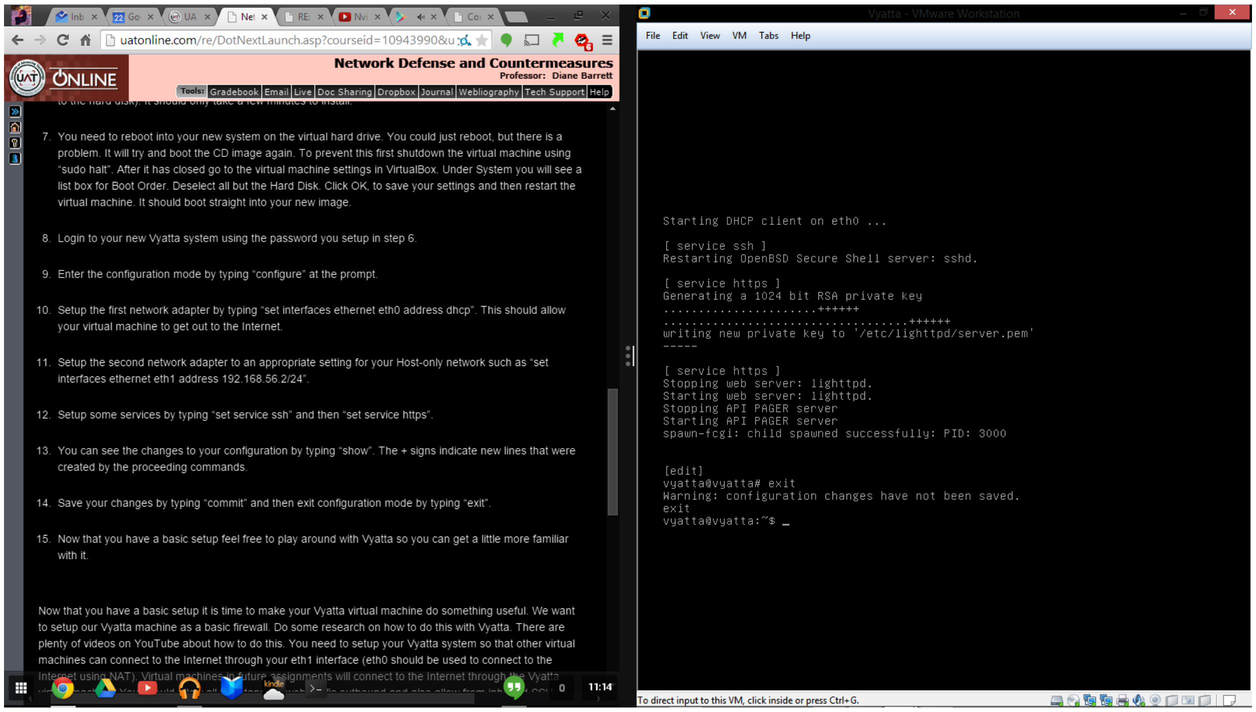
14. Save your changes by typing “commit” and then exit configuration mode by typing “exit”.

15. Now that you have a basic setup feel free to play around with Vyatta so you can get a little more familiar with it.

Now that you have a basic setup it is time to make your Vyatta virtual machine do something useful. We want to setup our Vyatta machine as a basic firewall. Do some research on how to do this with Vyatta. There are plenty of videos on YouTube about how to do this. You need to setup your Vyatta system so that other virtual machines can connect to the Internet through your eth1 interface (eth0 should be used to connect to the Internet using NAT). Virtual machines in future assignments will connect to the Internet through the Vyatta

```
 interfaces {
     ethernet eth0 {
+        address dhcp
         hw-id 00:0c:29:e9:6c:ff
     }
     ethernet eth1 {
+        address 192.168.56.2/24
         hw-id 00:0c:29:e9:6c:09
     }
 }
+service {
+    https {
+    }
+    ssh {
+    }
+}
 system {
     login {
         user vyatta {
             authentication {
                 encrypted-password $1$FOWoh8//$iSUvI3hMlSJvjLmYVSbKh.
             }
         }
     }
:_
```

5) Entered commands:
commit
exit



Network Defense and Countermeasures
Professor: Diane Barrett

7. You need to reboot into your new system on the virtual hard drive. You could just reboot, but there is a problem. It will try and boot the CD image again. To prevent this first shutdown the virtual machine using “sudo halt”. After it has closed go to the virtual machine settings in VirtualBox. Under System you will see a list box for Boot Order. Deselect all but the Hard Disk. Click OK, to save your settings and then restart the virtual machine. It should boot straight into your new image.

8. Login to your new Vyatta system using the password you setup in step 6.

9. Enter the configuration mode by typing “configure” at the prompt.

10. Setup the first network adapter by typing “set interfaces ethernet eth0 address dhcp”. This should allow your virtual machine to get out to the Internet.

11. Setup the second network adapter to an appropriate setting for your Host-only network such as “set interfaces ethernet eth1 address 192.168.56.2/24”.

12. Setup some services by typing “set service ssh” and then “set service https”.

13. You can see the changes to your configuration by typing “show”. The + signs indicate new lines that were created by the proceeding commands.

14. Save your changes by typing “commit” and then exit configuration mode by typing “exit”.

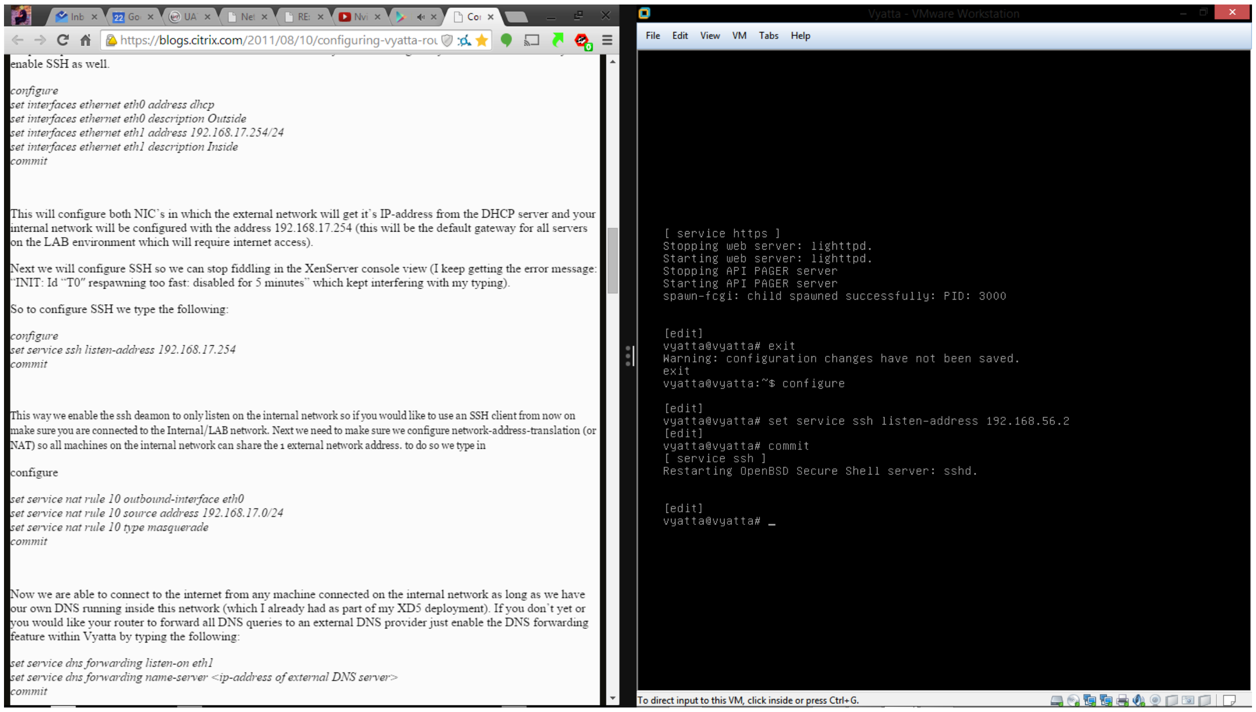
15. Now that you have a basic setup feel free to play around with Vyatta so you can get a little more familiar with it.

Now that you have a basic setup it is time to make your Vyatta virtual machine do something useful. We want to setup our Vyatta machine as a basic firewall. Do some research on how to do this with Vyatta. There are plenty of videos on YouTube about how to do this. You need to setup your Vyatta system so that other virtual machines can connect to the Internet through your eth1 interface (eth0 should be used to connect to the Internet

```
Starting DHCP client on eth0 ...

[ service ssh ]
Restarting OpenBSD Secure Shell server: sshd.

[ service https ]
Generating a 1024 bit RSA private key
.......................++++++
..................................++++++
writing new private key to '/etc/lighttpd/server.pem'
-----

[ service https ]
Stopping web server: lighttpd.
Starting web server: lighttpd.
Stopping API PAGER server
Starting API PAGER server
spawn-fcgi: child spawned successfully: PID: 3000


[edit]
vyatta@vyatta# exit
Warning: configuration changes have not been saved.
exit
vyatta@vyatta:~$ _
```

6) Using the source provided by Kofi (I was using 4 with different directions in combination and were lacking), I start with SSH:
configure

set service ssh listen-address 192.168.56.2
commit


https://blogs.citrix.com/2011/08/10/configuring-vyatta-rou
Vyatta - VMware Workstation
File Edit View VM Tabs Help
To direct input to this VM, click inside or press Ctrl+G.

enable SSH as well.

configure
set interfaces ethernet eth0 address dhcp
set interfaces ethernet eth0 description Outside
set interfaces ethernet eth1 address 192.168.17.254/24
set interfaces ethernet eth1 description Inside
commit

This will configure both NIC's in which the external network will get it's IP-address from the DHCP server and your internal network will be configured with the address 192.168.17.254 (this will be the default gateway for all servers on the LAB environment which will require internet access).

Next we will configure SSH so we can stop fiddling in the XenServer console view (I keep getting the error message: "INIT: Id "T0" respawning too fast: disabled for 5 minutes" which kept interfering with my typing).

So to configure SSH we type the following:

configure
set service ssh listen-address 192.168.17.254
commit

This way we enable the ssh deamon to only listen on the internal network so if you would like to use an SSH client from now on make sure you are connected to the Internal/LAB network. Next we need to make sure we configure network-address-translation (or NAT) so all machines on the internal network can share the 1 external network address. to do so we type in

configure

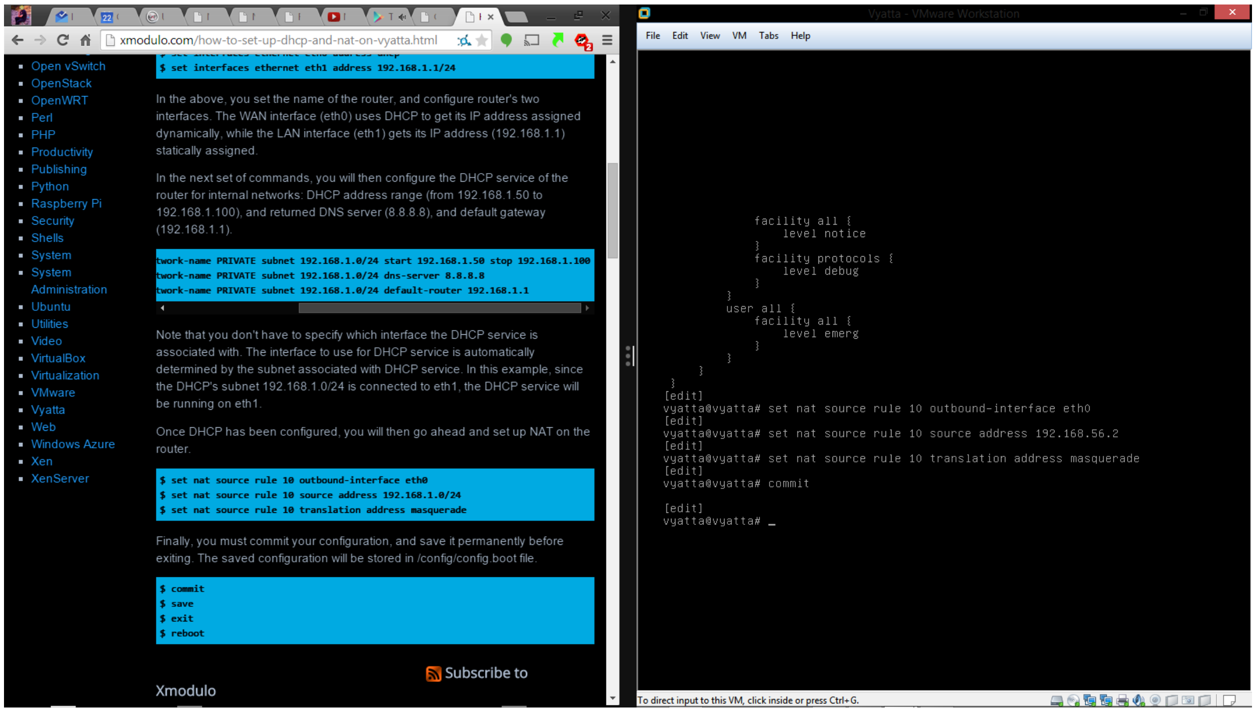
set service nat rule 10 outbound-interface eth0
set service nat rule 10 source address 192.168.17.0/24
set service nat rule 10 type masquerade
commit

Now we are able to connect to the internet from any machine connected on the internal network as long as we have our own DNS running inside this network (which I already had as part of my XD5 deployment). If you don't yet or you would like your router to forward all DNS queries to an external DNS provider just enable the DNS forwarding feature within Vyatta by typing the following:

set service dns forwarding listen-on eth1
set service dns forwarding name-server <ip-address of external DNS server>
commit

```
[ service https ]
Stopping web server: lighttpd.
Starting web server: lighttpd.
Stopping API PAGER server
Starting API PAGER server
spawn-fcgi: child spawned successfully: PID: 3000


[edit]
vyatta@vyatta# exit
Warning: configuration changes have not been saved.
exit
vyatta@vyatta:~$ configure

[edit]
vyatta@vyatta# set service ssh listen-address 192.168.56.2
[edit]
vyatta@vyatta# commit
[ service ssh ]
Restarting OpenBSD Secure Shell server: sshd.


[edit]
vyatta@vyatta# _
```

7) Then I set up the NAT rules
configure
set nat source rule 10 outbound-interface eth0
set nat source rule 10 source address 192.168.56.0/24
set nat source rule 10 type masquerade
commit



xmodulo.com/how-to-set-up-dhcp-and-nat-on-vyatta.html

- Open vSwitch
- OpenStack
- OpenWRT
- Perl
- PHP
- Productivity
- Publishing
- Python
- Raspberry Pi
- Security
- Shells
- System
- System Administration
- Ubuntu
- Utilities
- Video
- VirtualBox
- Virtualization
- VMware
- Vyatta
- Web
- Windows Azure
- Xen
- XenServer

```
$ set interfaces ethernet eth1 address 192.168.1.1/24
```

In the above, you set the name of the router, and configure router's two interfaces. The WAN interface (eth0) uses DHCP to get its IP address assigned dynamically, while the LAN interface (eth1) gets its IP address (192.168.1.1) statically assigned.

In the next set of commands, you will then configure the DHCP service of the router for internal networks: DHCP address range (from 192.168.1.50 to 192.168.1.100), and returned DNS server (8.8.8.8), and default gateway (192.168.1.1).

```
twork-name PRIVATE subnet 192.168.1.0/24 start 192.168.1.50 stop 192.168.1.100
twork-name PRIVATE subnet 192.168.1.0/24 dns-server 8.8.8.8
twork-name PRIVATE subnet 192.168.1.0/24 default-router 192.168.1.1
```

Note that you don't have to specify which interface the DHCP service is associated with. The interface to use for DHCP service is automatically determined by the subnet associated with DHCP service. In this example, since the DHCP's subnet 192.168.1.0/24 is connected to eth1, the DHCP service will be running on eth1.

Once DHCP has been configured, you will then go ahead and set up NAT on the router.

```
$ set nat source rule 10 outbound-interface eth0
$ set nat source rule 10 source address 192.168.1.0/24
$ set nat source rule 10 translation address masquerade
```

Finally, you must commit your configuration, and save it permanently before exiting. The saved configuration will be stored in /config/config.boot file.

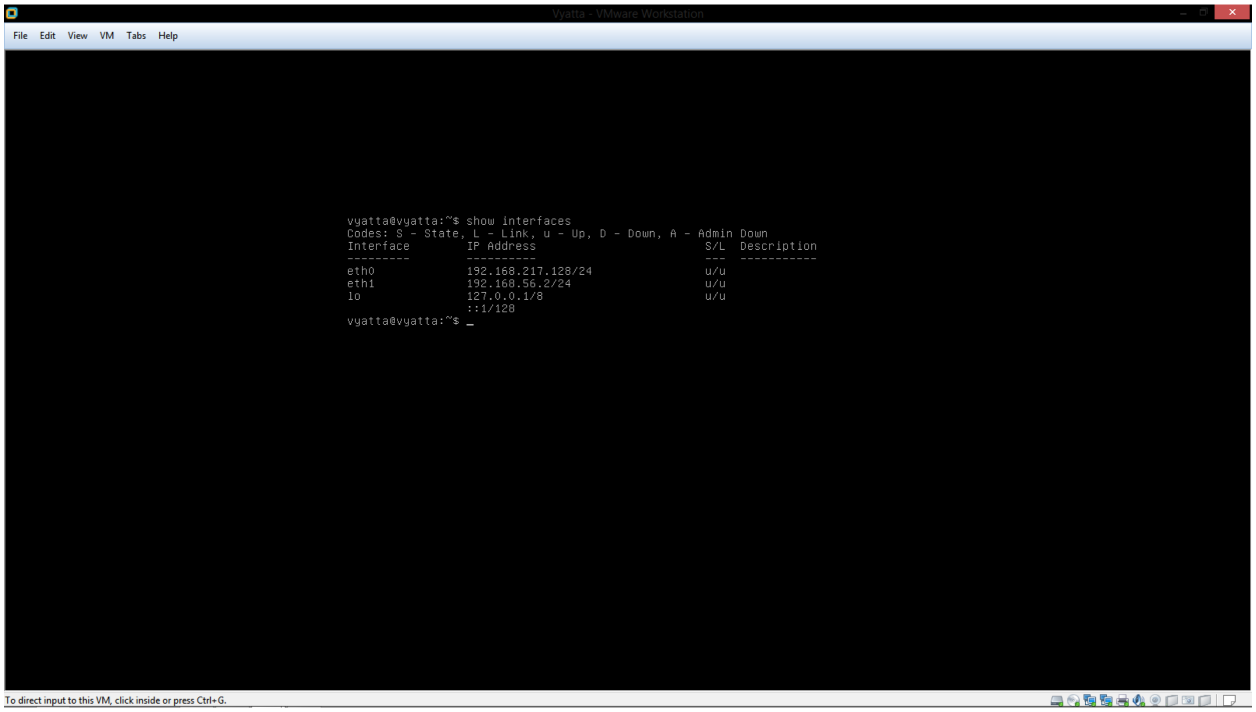
```
$ commit
$ save
$ exit
$ reboot
```

Subscribe to Xmodulo

File Edit View VM Tabs Help
To direct input to this VM, click inside or press Ctrl+G.

```
                facility all {
                    level notice
                }
                facility protocols {
                    level debug
                }
            }
            user all {
                facility all {
                    level emerg
                }
            }
        }
    }
[edit]
vyatta@vyatta# set nat source rule 10 outbound-interface eth0
[edit]
vyatta@vyatta# set nat source rule 10 source address 192.168.56.2
[edit]
vyatta@vyatta# set nat source rule 10 translation address masquerade
[edit]
vyatta@vyatta# commit

[edit]
vyatta@vyatta# _
```

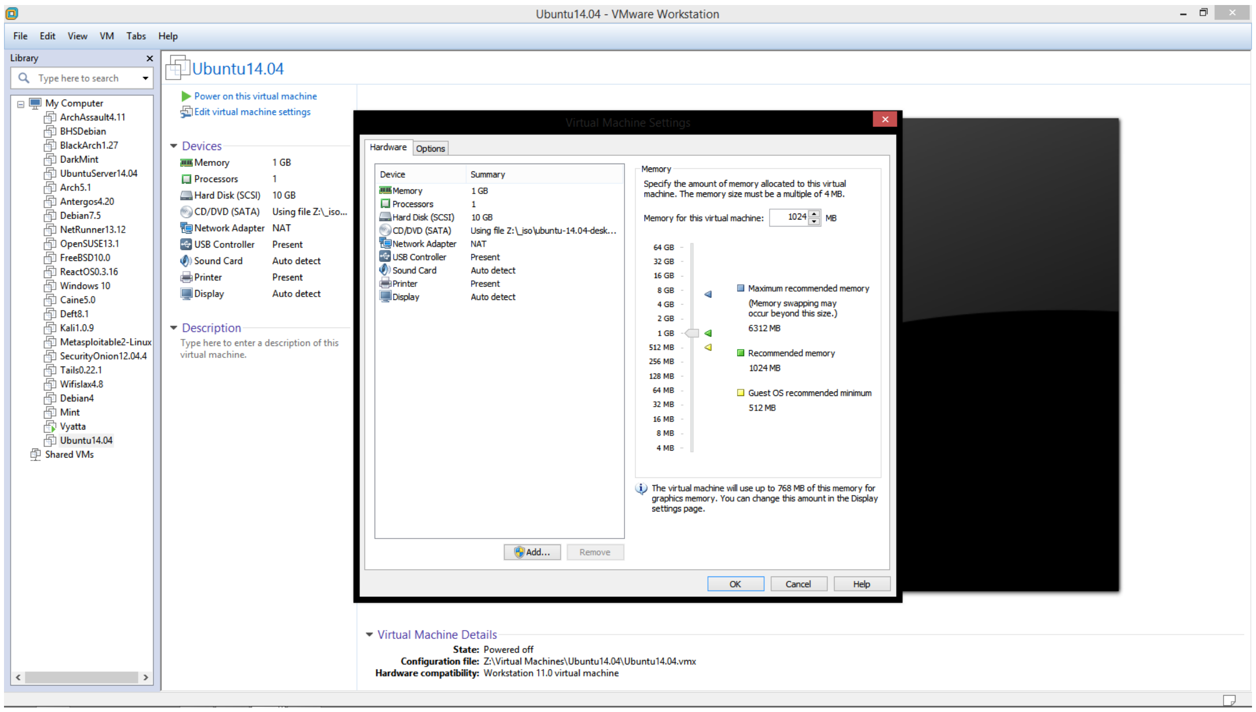
8) “show interfaces”



Vyatta - VMware Workstation
File Edit View VM Tabs Help
To direct input to this VM, click inside or press Ctrl+G.

```
vyatta@vyatta:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                           S/L  Description
---------        ----------                           ---  -----------
eth0             192.168.217.128/24                   u/u
eth1             192.168.56.2/24                      u/u
lo               127.0.0.1/8                          u/u
                 ::1/128
vyatta@vyatta:~$ _
```

9) Now it’s time to set up Ubuntu with 512MB RAM, 10GB storage just in case, and one network adapter of host only. That failed. I assumed since I set up NAT only that it would fail again. Reset to NAT.



Virtual Machine Settings

Device	Summary
Memory	1 GB
Processors	1
Hard Disk (SCSI)	10 GB
CD/DVD (SATA)	Using file Z:_iso\ubuntu-14.04-desk...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Memory

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

Memory for this virtual machine: 1024 MB

Maximum recommended memory (Memory swapping may occur beyond this size.) 6312 MB

Recommended memory 1024 MB

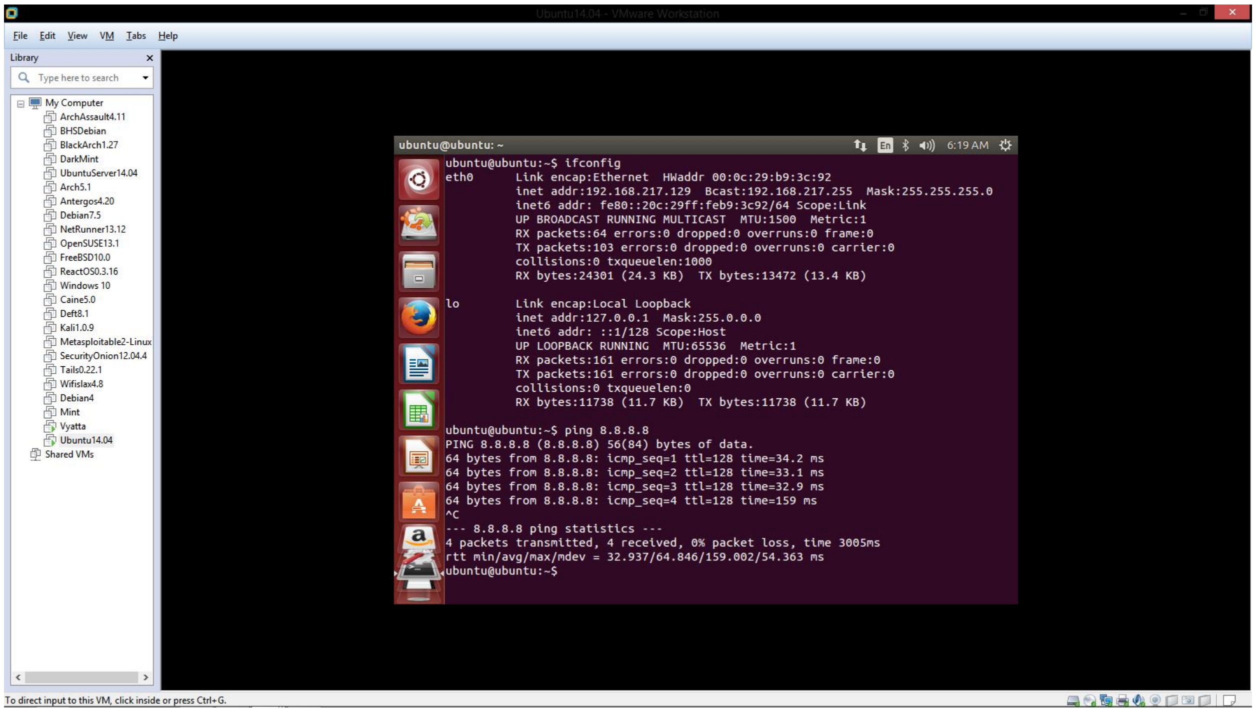
Guest OS recommended minimum 512 MB

The virtual machine will use up to 768 MB of this memory for graphics memory. You can change this amount in the Display settings page.

Virtual Machine Details

State: Powered off
Configuration file: Z:\Virtual Machines\Ubuntu14.04\Ubuntu14.04.vmx
Hardware compatibility: Workstation 11.0 virtual machine

10) Changed keyboard to Dvorak and opened terminal. Entered “ifconfig” and “ping 8.8.8.8”



```
ubuntu@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:b9:3c:92
          inet addr:192.168.217.129  Bcast:192.168.217.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:feb9:3c92/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:64 errors:0 dropped:0 overruns:0 frame:0
          TX packets:103 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:24301 (24.3 KB)  TX bytes:13472 (13.4 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:65536  Metric:1
          RX packets:161 errors:0 dropped:0 overruns:0 frame:0
          TX packets:161 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:11738 (11.7 KB)  TX bytes:11738 (11.7 KB)

ubuntu@ubuntu:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=34.2 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=33.1 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=32.9 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=128 time=159 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 32.937/64.846/159.002/54.363 ms
ubuntu@ubuntu:~$
```

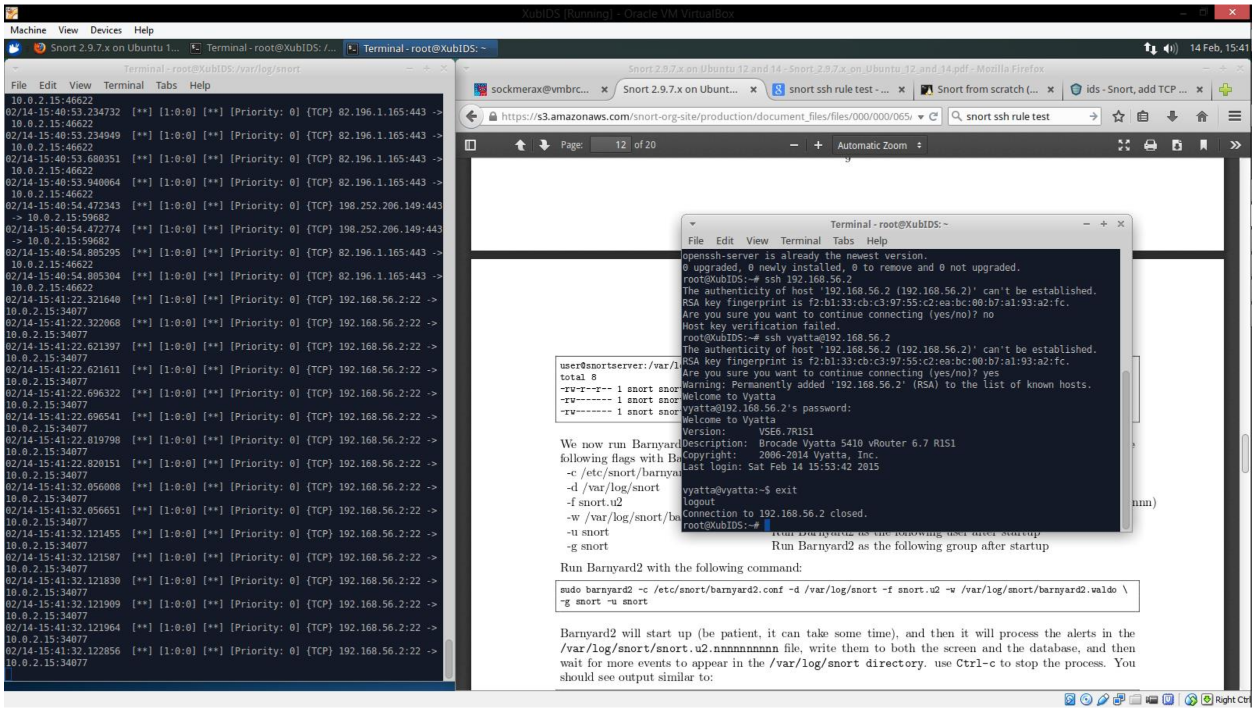
Lab/Activity Summary:

For 7, I had to take a big detour since the commands “set service nat” didn’t want to work. I assume this aspect may be outdated, but the used “set nat source” actually did work. This was a huge issue the first try that took me down a more convoluted path than this attempt. I also learned what my issue was last time. First, I had Ubuntu powered on right when I was installing Vyatta vs waiting until after setting up NAT rules. Second, because of the first, I added a default gateway not realizing it would be done automatically. My issues the first two times I tried this, this attempt being the third, was a matter of impatience and rushing that allowed me to confuse myself and skip a rule or two, i.e. that the NAT rules changed.

Lab Memorandum

Professor: Barrett
Course: NTW415, Network Defense & Countermeasures
Student: Nick Somerville
Date: 14-Feb-15
Lab / Activity: Assignment 3.1 – Intrusion Detection System

Rules to follow



```
 10.0.2.15:46622
02/14-15:40:53.234732  [**] [1:0:0] [**] [Priority: 0] {TCP} 82.196.1.165:443 ->
 10.0.2.15:46622
02/14-15:40:53.234949  [**] [1:0:0] [**] [Priority: 0] {TCP} 82.196.1.165:443 ->
 10.0.2.15:46622
02/14-15:40:53.680351  [**] [1:0:0] [**] [Priority: 0] {TCP} 82.196.1.165:443 ->
 10.0.2.15:46622
02/14-15:40:53.940064  [**] [1:0:0] [**] [Priority: 0] {TCP} 82.196.1.165:443 ->
 10.0.2.15:46622
02/14-15:40:54.472343  [**] [1:0:0] [**] [Priority: 0] {TCP} 198.252.206.149:443
 -> 10.0.2.15:59682
02/14-15:40:54.472774  [**] [1:0:0] [**] [Priority: 0] {TCP} 198.252.206.149:443
 -> 10.0.2.15:59682
02/14-15:40:54.805295  [**] [1:0:0] [**] [Priority: 0] {TCP} 82.196.1.165:443 ->
 10.0.2.15:46622
02/14-15:40:54.805304  [**] [1:0:0] [**] [Priority: 0] {TCP} 82.196.1.165:443 ->
 10.0.2.15:46622
02/14-15:41:22.321640  [**] [1:0:0] [**] [Priority: 0] {TCP} 192.168.56.2:22 ->
10.0.2.15:34077
02/14-15:41:22.322068  [**] [1:0:0] [**] [Priority: 0] {TCP} 192.168.56.2:22 ->
10.0.2.15:34077
02/14-15:41:22.621397  [**] [1:0:0] [**] [Priority: 0] {TCP} 192.168.56.2:22 ->
10.0.2.15:34077
02/14-15:41:22.621611  [**] [1:0:0] [**] [Priority: 0] {TCP} 192.168.56.2:22 ->
10.0.2.15:34077
02/14-15:41:22.696322  [**] [1:0:0] [**] [Priority: 0] {TCP} 192.168.56.2:22 ->
10.0.2.15:34077
02/14-15:41:22.696541  [**] [1:0:0] [**] [Priority: 0] {TCP} 192.168.56.2:22 ->
10.0.2.15:34077
02/14-15:41:22.819798  [**] [1:0:0] [**] [Priority: 0] {TCP} 192.168.56.2:22 ->
10.0.2.15:34077
02/14-15:41:22.820151  [**] [1:0:0] [**] [Priority: 0] {TCP} 192.168.56.2:22 ->
10.0.2.15:34077
02/14-15:41:32.056008  [**] [1:0:0] [**] [Priority: 0] {TCP} 192.168.56.2:22 ->
10.0.2.15:34077
02/14-15:41:32.056651  [**] [1:0:0] [**] [Priority: 0] {TCP} 192.168.56.2:22 ->
10.0.2.15:34077
02/14-15:41:32.121455  [**] [1:0:0] [**] [Priority: 0] {TCP} 192.168.56.2:22 ->
10.0.2.15:34077
02/14-15:41:32.121587  [**] [1:0:0] [**] [Priority: 0] {TCP} 192.168.56.2:22 ->
10.0.2.15:34077
02/14-15:41:32.121830  [**] [1:0:0] [**] [Priority: 0] {TCP} 192.168.56.2:22 ->
10.0.2.15:34077
02/14-15:41:32.121909  [**] [1:0:0] [**] [Priority: 0] {TCP} 192.168.56.2:22 ->
10.0.2.15:34077
02/14-15:41:32.121964  [**] [1:0:0] [**] [Priority: 0] {TCP} 192.168.56.2:22 ->
10.0.2.15:34077
02/14-15:41:32.122856  [**] [1:0:0] [**] [Priority: 0] {TCP} 192.168.56.2:22 ->
10.0.2.15:34077
```

```
openssh-server is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@XubIDS:~# ssh 192.168.56.2
The authenticity of host '192.168.56.2 (192.168.56.2)' can't be established.
RSA key fingerprint is f2:b1:33:cb:c3:97:55:c2:ea:bc:00:b7:a1:93:a2:fc.
Are you sure you want to continue connecting (yes/no)? no
Host key verification failed.
root@XubIDS:~# ssh vyatta@192.168.56.2
The authenticity of host '192.168.56.2 (192.168.56.2)' can't be established.
RSA key fingerprint is f2:b1:33:cb:c3:97:55:c2:ea:bc:00:b7:a1:93:a2:fc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.2' (RSA) to the list of known hosts.
Welcome to Vyatta
vyatta@192.168.56.2's password:
Welcome to Vyatta
Version:      VSE6.7R1S1
Description:  Brocade Vyatta 5410 vRouter 6.7 R1S1
Copyright:    2006-2014 Vyatta, Inc.
Last login: Sat Feb 14 15:53:42 2015

vyatta@vyatta:~$ exit
logout
Connection to 192.168.56.2 closed.
root@XubIDS:~#
```

Run Barnyard2 with the following command:

```
sudo barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.u2 -w /var/log/snort/barnyard2.waldo \
-g snort -u snort
```

Barnyard2 will start up (be patient, it can take some time), and then it will process the alerts in the /var/log/snort/snort.u2.nnnnnnnnnn file, write them to both the screen and the database, and then wait for more events to appear in the /var/log/snort directory. use Ctrl-c to stop the process. You should see output similar to:

# IP address range used by Vyatta eth0 may vary
# Some packages and sections not essential, but better to be safe than sorry

sudo -s
apt-get update
apt-get upgrade -y
apt-get install -y openssh-server
reboot

sudo -s
apt-get install -y nmap nbtscan apache2 php5 php5-mysql php5-gd g++ make autoconf libtool
apt-get install -y ethtool
ethtool -K eth0 gro off

```
ethtool -K eth0 lro off

apt-get install -y build-essential

apt-get install libpcap0.8-dev libpcap-dev libpcre3-dev libdumbnet-dev

apt-get install -y mysql-server
apt-get install -y libmysqlclient-dev mysql-client

apt-get update && apt-get upgrade

mkdir ~/snort_src
cd ~/snort_src

apt-get install -y bison flex

wget http://hem.bredband.net/jpgraph/jpgraph-1.27.1.tar.gz
mkdir /var/www/jpgraph
tar zxvf jpgraph-1.27.1.tar.gz
cp -r jpgraph-1.27.1/src /var/www/jpgraph/

wget http://symmetrixtech.com/wp/wp-content/uploads/2014/09/snortreport-1.3.4.tar.gz
tar zxvf snortreport-1.3.4.tar.gz -C /var/www/
nano /var/www/snortreport-1.3.3/srconf.php

# change password

wget https://www.snort.org/downloads/snort/daq-2.0.4.tar.gz
tar -xvzf daq-2.0.4.tar.gz
cd daq-2.0.4
./configure
make
make install

apt-get install -y zlib1g-dev

cd ..
wget http://libdnet.googlecode.com/files/libdnet-1.12.tgz
tar zxvf libdnet-1.12.tgz
cd libdnet-1.12/
./configure
make
make install
```

```
ln -s /usr/local/lib/libdnet.1.0.1 /usr/lib/libdnet.1

cd ..
wget https://www.snort.org/downloads/snort/snort-2.9.7.0.tar.gz
tar -xvzf snort-2.9.7.0.tar.gz
cd snort-2.9.7.0
./configure --enable-sourcefire
make
make install

ldconfig

ln -s /usr/local/bin/snort /usr/sbin/snort

groupadd snort
useradd snort -r -s /sbin/nologin -c SNORT_IDS -g snort
mkdir /etc/snort
mkdir /etc/snort/rules
mkdir /etc/snort/preproc_rules
touch /etc/snort/rules/white_list.rules /etc/snort/rules/black_list.rules /etc/snort/rules/local.rules
mkdir /var/log/snort
mkdir /usr/local/lib/snort_dynamicrules
chmod -R 5775 /etc/snort
chmod -R 5775 /var/log/snort
chmod -R 5775 /usr/local/lib/snort_dynamicrules
chown -R snort:snort /etc/snort
chown -R snort:snort /var/log/snort
chown -R snort:snort /usr/local/lib/snort_dynamicrules

cp ~/snort_src/snort-2.9.7.0/etc/*.conf* /etc/snort
cp ~/snort_src/snort-2.9.7.0/etc/*.map /etc/snort

sed -i 's/include \$RULE\_PATH/#include \$RULE\_PATH/' /etc/snort/snort.conf

nano /etc/snort/snort.conf

        ipvar HOME_NET 10.0.2.15/24
        ipvar EXTERNAL_NET !$HOME_NET


        var RULE_PATH /etc/snort/rules
        var SO_RULE_PATH /etc/snort/so_rules
        var PREPROC_RULE_PATH /etc/snort/preproc_rules
        var WHITE_LIST_PATH /etc/snort/rules
        var BLACK_LIST_PATH /etc/snort/rules

        include $RULE_PATH/local.rules

nano /etc/snort/rules/local.rules

        alert tcp any any -> 10.0.2.15/24 1:65535

# HOME_NET is the same as entering home networks' IP range

nano /etc/snort/snort.conf

        output unified2: filename snort.u2, limit 128

cd ..
wget https://github.com/firnsy/barnyard2/archive/master.tar.gz -O barnyard2-2-1.13.tar.gz
tar zxvf barnyard2-2-1.13.tar.gz
cd barnyard2-master
autoreconf -fvi -I ./m4
./autogen.sh
./configure --with-mysql --with-mysql-libraries=/usr/lib/x86_64-linux-gnu
make
make install

cp etc/barnyard2.conf /etc/snort
mkdir /var/log/barnyard2
chown snort.snort /var/log/barnyard2
touch /var/log/snort/barnyard2.waldo
chown snort.snort /var/log/snort/barnyard2.waldo
touch /etc/snort/sid-msg.map

echo "create database snort;" | mysql -u root -p
mysql -u root -p -D snort < ~/snort_src/barnyard2-master/schemas/create_mysql
echo "grant create, insert, select, delete, update on snort.* to snort@localhost identified by
'MYSQLSNORTPASSWORD'" | mysql -u root -p

nano /etc/snort/barnyard2.conf

        output database: log, mysql, user=snort password=MYSQLSNORTPASSWORD
dbname=snort host=localhost

chmod o-r /etc/snort/barnyard2.conf
```

```
/etc/init.d/networking restart

apt-get install -y libcrypt-ssleay-perl liblwp-useragent-determined-perl

wget https://pulledpork.googlecode.com/files/pulledpork-0.7.0.tar.gz
tar xvfvz pulledpork-0.7.0.tar.gz
cd pulledpork-0.7.0/
cp pulledpork.pl /usr/local/bin
chmod +x /usr/local/bin/pulledpork.pl
cp etc/*.conf /etc/snort

mkdir /etc/snort/rules/iplists
touch /etc/snort/rules/iplists/default.blacklist

/usr/local/bin/pulledpork.pl -V

nano /etc/snort/pulledpork.conf

        # enter oinkcode for 19 / 26

        rule_path=/etc/snort/rules/snort.rules

        local_rules=/etc/snort/rules/local.rules

        sid_msg=/etc/snort/sid-msg.map

        config_path=/etc/snort/snort.conf

        distro=Ubuntu-10-4

        black_list=/etc/snort/rules/iplists/default.blacklist

        IPRVersion=/etc/snort/rules/iplists

        enablesid=/etc/snort/enablesid.conf
        dropsid=/etc/snort/dropsid.conf
        disablesid=/etc/snort/disablesid.conf
        modifysid=/etc/snort/modifysid.conf

/usr/local/bin/pulledpork.pl -c /etc/snort/pulledpork.conf -l

nano /etc/snort/snort.conf
```

```
        include $RULE_PATH/snort.rules

snort -T -c /etc/snort/snort.conf

crontab -e

        01 04 * * * /usr/local/bin/pulledpork.pl -c /etc/snort/pulledpork.conf -l

nano /etc/init/snort.conf

        description "Snort NIDS Service"
        stop on runlevel [!2345]
        start on runlevel [2345]
        script
                exec /usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i eth0 -D end
        script

chmod +x /etc/init/snort.conf
initctl list | grep snort

nano /etc/init/barnyard2.conf

        description "Barnyard2 service"
        stop on runlevel [!2345]
        start on runlevel [2345]
        script
                exec /usr/local/bin/barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f
                snort.u2 -w /var/log/snort /barnyard2.waldo -g snort -u snort -D
        end script

chmod +x /etc/init/barnyard2.conf
initctl list | grep barnyard

service snort status
service barnyard2 status

apt-get install -y apache2 libapache2-mod-php5 php5 php5-mysql php5-common \ php5-gd
php5-cli php-pear

pear install -f Image_Graph

cd ..
```

```
wget http://sourceforge.net/projects/adodb/files/adodb-php5-only/adodb-518-for-
php5/adodb518a.tgz/download -O adodb518.tgz
tar -xvzf adodb518.tgz
mv adodb5 /var/adodb

wget http://sourceforge.net/projects/secureideas/files/BASE/base-1.4.5/base-1.4.5.tar.gz
tar -zxvf base-1.4.5.tar.gz

mv base-1.4.5 /var/www/base/
cd /var/www/base
cp base_conf.php.dist base_conf.php

mv base-1.4.5 /var/www/html/base/
cd /var/www/html/base
cp base_conf.php.dist base_conf.php

nano /var/www/html/base/base_conf.php

        $BASE_urlpath = '/base';
        $DBlib_path = '/var/adodb/';
        $alert_dbname = 'snort';
        $alert_host = 'localhost';
        $alert_port = '';
        $alert_user = 'snort';
        $alert_password = 'MYSQLSNORTPASSWORD';

chown -R www-data:www-data /var/www/html/base

chmod o-r /var/www/html/base/base_conf.php

service apache2 restart
```

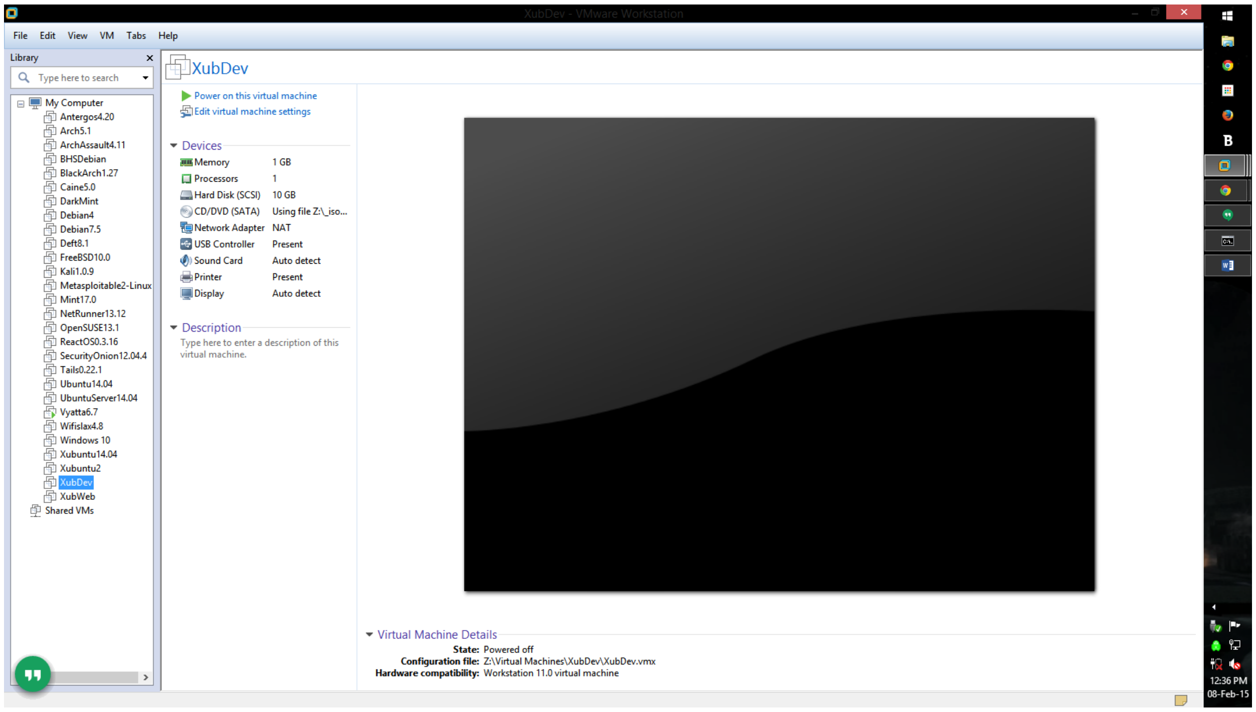
Lab/Activity Summary:

It took me a few hours during one day to sort out the issues. First, my installation steps were out of place with the instructions that followed. I merged the two installation guides for a flawless installation. Second, my last two attempts were using rules found online. The most common that directly mentioned SSH was against a brute force attack with 5 attempts over 60 seconds. This rule was way too specific. For the point of copping out to test the IDS, I set the rule to be so generic that every bit of traffic was logged. That’s a terrible idea in the real world when trying to find malicious / suspicious traffic, but it did its function for the assignment.

Lab Memorandum

Professor:	**Barrett**
Course:	**NTW415, Network Defense & Countermeasures**
Student:	**Nick Somerville**
Date:	**08-Feb-15**
Lab / Activity:	**Assignment 4.1 – Linux Firewalls Using IPtables**

1) I created the Xubuntu VMs for the web server and developer console with the same configurations. I didn't need to make any additional configurations to the network adapter since, by running Vyatta first, all of my VMs automatically connect to Vyatta.



Devices

Memory	1 GB
Processors	1
Hard Disk (SCSI)	10 GB
CD/DVD (SATA)	Using file Z:_iso...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Description

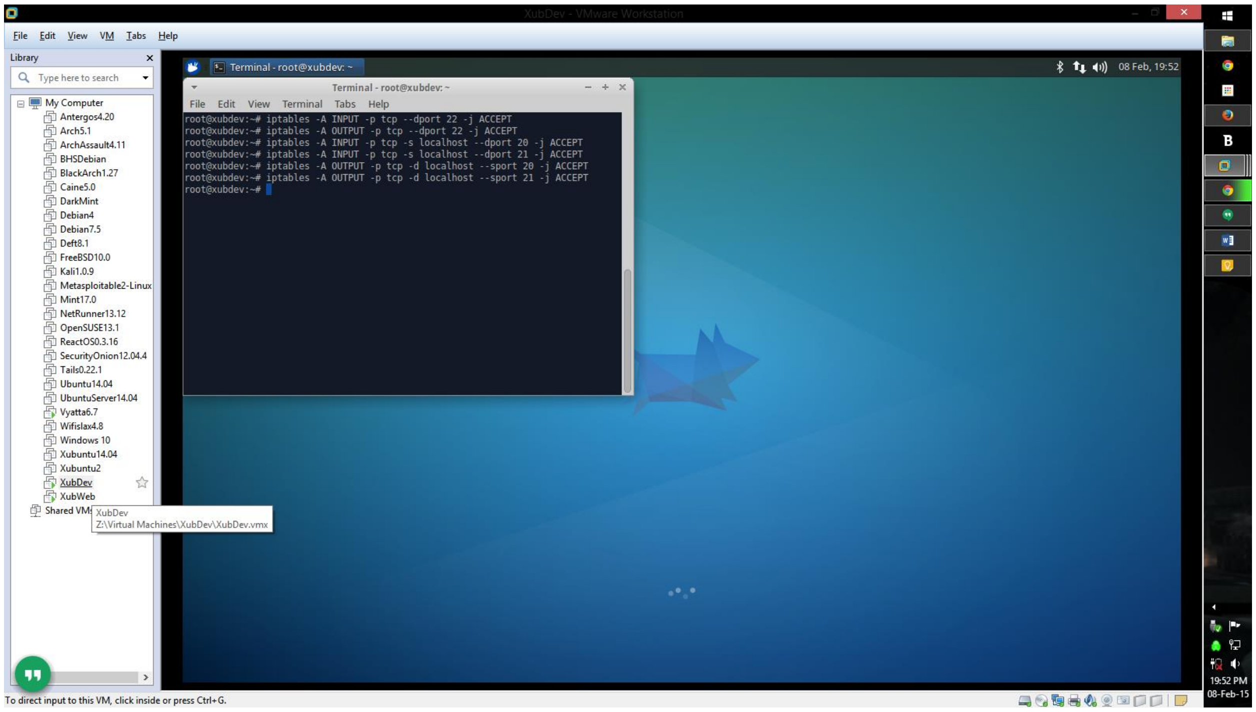
Type here to enter a description of this virtual machine.

Virtual Machine Details

State:	Powered off
Configuration file:	Z:\Virtual Machines\XubDev\XubDev.vmx
Hardware compatibility:	Workstation 11.0 virtual machine

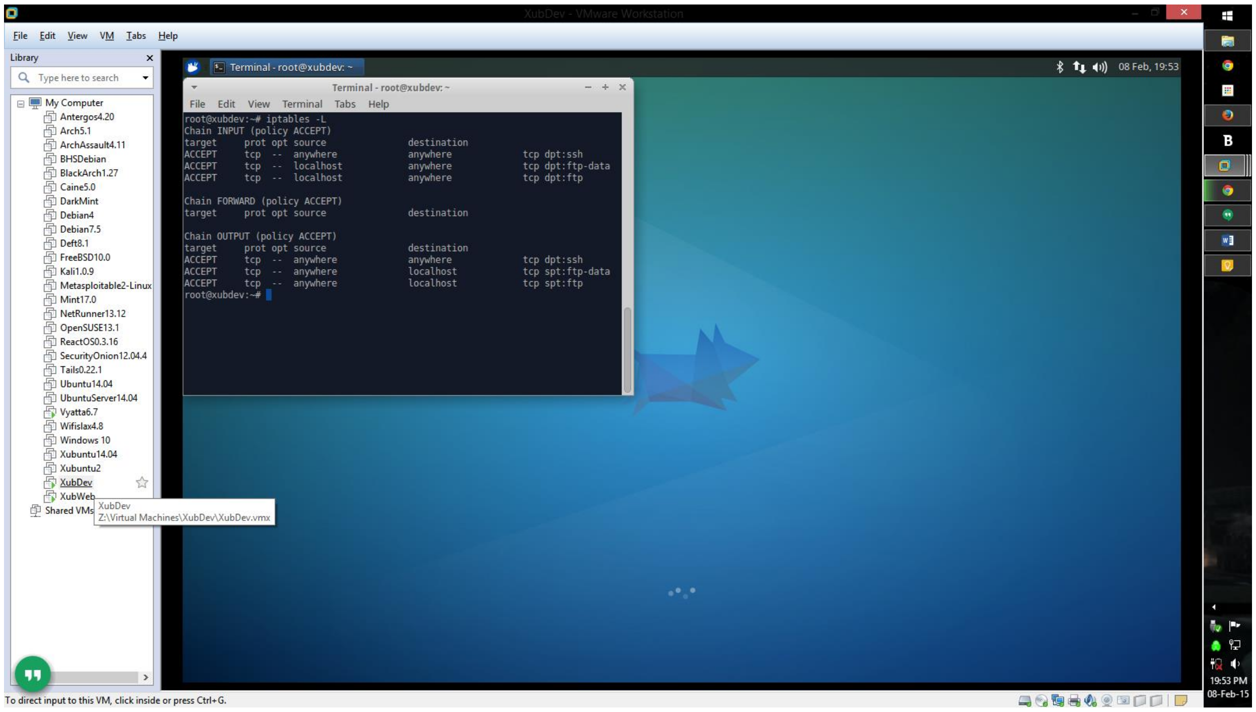
2) Then, after installing both Xubuntu VMs and updating, I added the following IPtable rules for the Xubuntu dev VM to allow SSH from all and FTP from local only.

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p tcp -s localhost --dport 20 -j ACCEPT
iptables -A INPUT -p tcp -s localhost --dport 21 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 22 -j ACCEPT
iptables -A OUTPUT -p tcp -d localhost --sport 20 -j ACCEPT
iptables -A OUTPUT -p tcp -d localhost --sport 21 -j ACCEPT
```



```
root@xubdev:~# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
root@xubdev:~# iptables -A OUTPUT -p tcp --dport 22 -j ACCEPT
root@xubdev:~# iptables -A INPUT -p tcp -s localhost --dport 20 -j ACCEPT
root@xubdev:~# iptables -A INPUT -p tcp -s localhost --dport 21 -j ACCEPT
root@xubdev:~# iptables -A OUTPUT -p tcp -d localhost --sport 20 -j ACCEPT
root@xubdev:~# iptables -A OUTPUT -p tcp -d localhost --sport 21 -j ACCEPT
root@xubdev:~#
```

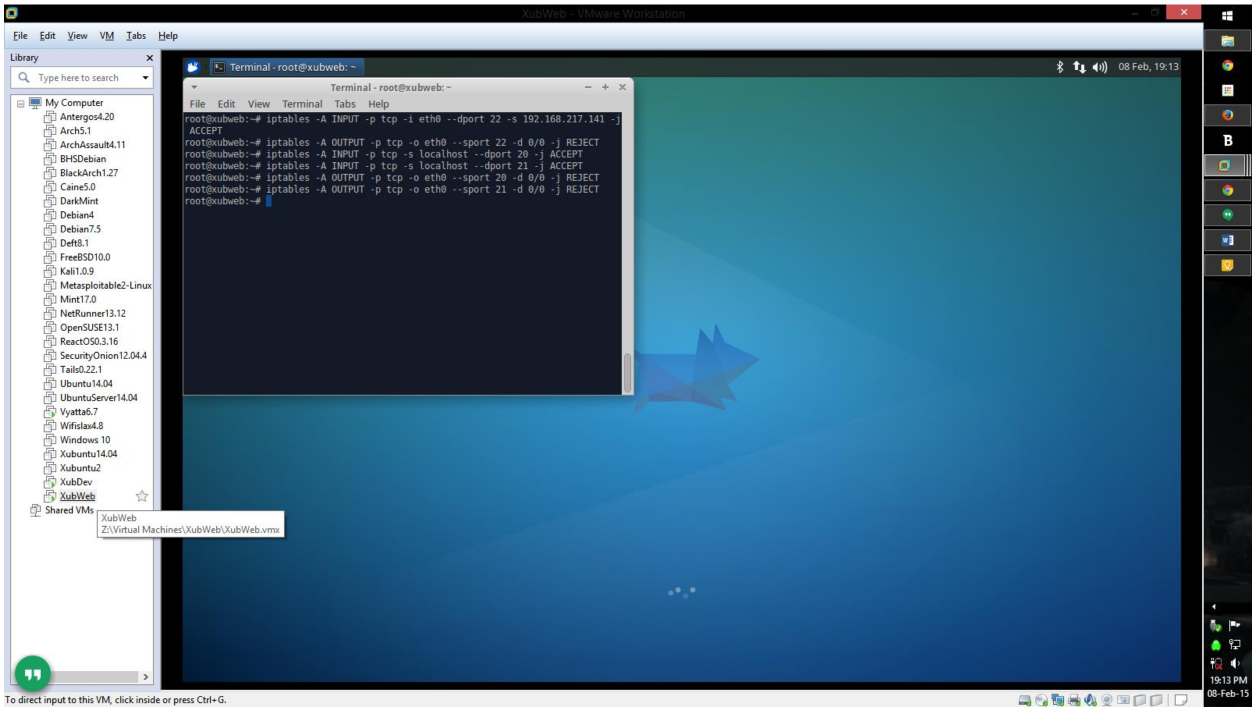
3) I made the directory and saved the IPtables. So the IPtables should look like the following:



```
root@xubdev:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:ftp-data
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:ftp

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             localhost            tcp spt:ftp-data
ACCEPT     tcp  --  anywhere             localhost            tcp spt:ftp
root@xubdev:~#
```

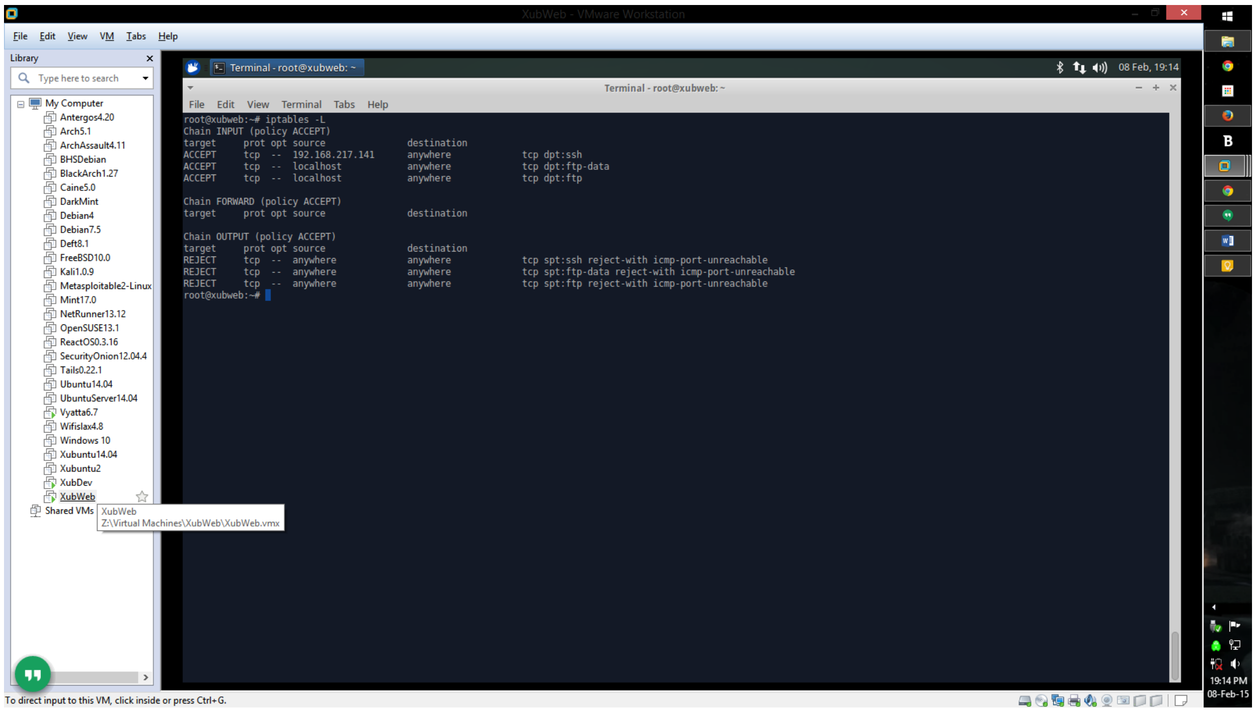
4) Next, I added the following IPtable rules for the Xubuntu web server VM to allow FTP from local only, to allow SSH from dev only, and to block all outgoing FTP / SSH connections.

```
iptables -A INPUT -p tcp -i eth0 --dport 22 -s 192.168.217.143 -j ACCEPT
iptables -A OUTPUT -p tcp -o eth0 --sport 22 -d 0/0 -j REJECT
iptables -A INPUT -p tcp -s localhost --dport 20 -j ACCEPT
iptables -A INPUT -p tcp -s localhost --dport 21 -j ACCEPT
iptables -A OUTPUT -p tcp -o eth0 --sport 20 -d 0/0 -j REJECT
iptables -A OUTPUT -p tcp -o eth0 --sport 21 -d 0/0 -j REJECT
```



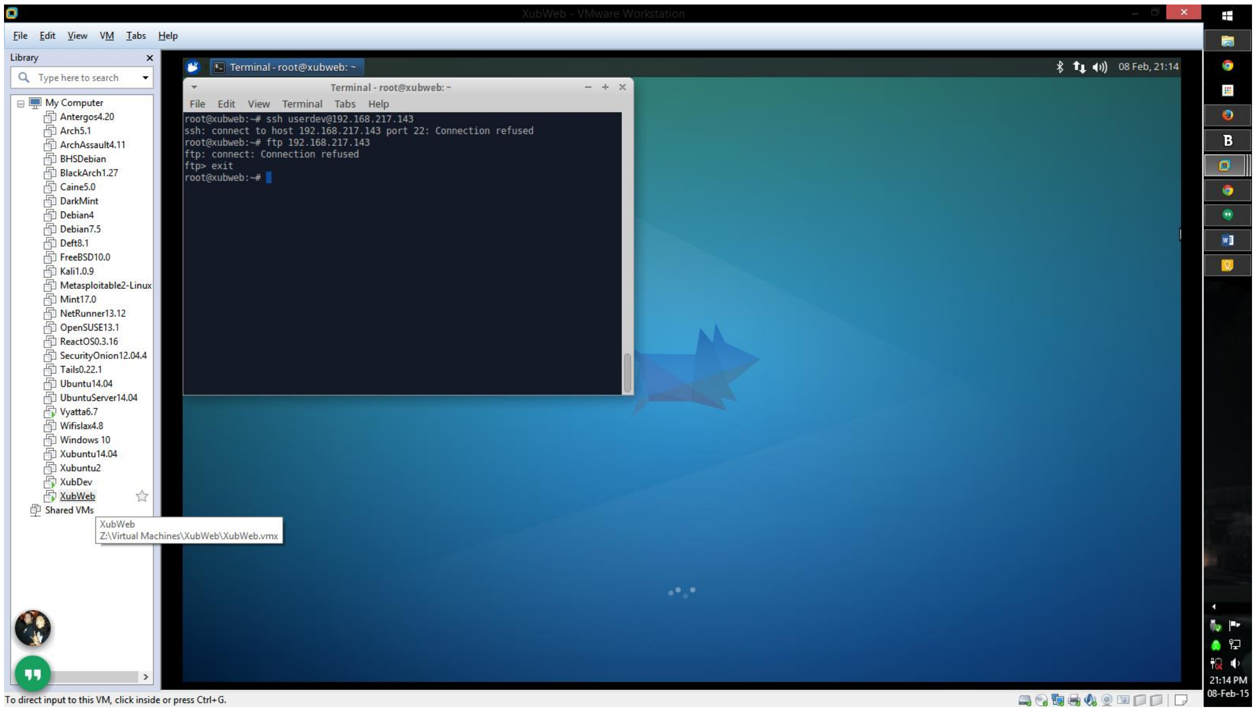
```
root@xubweb:~# iptables -A INPUT -p tcp -i eth0 --dport 22 -s 192.168.217.141 -j
 ACCEPT
root@xubweb:~# iptables -A OUTPUT -p tcp -o eth0 --sport 22 -d 0/0 -j REJECT
root@xubweb:~# iptables -A INPUT -p tcp -s localhost --dport 20 -j ACCEPT
root@xubweb:~# iptables -A INPUT -p tcp -s localhost --dport 21 -j ACCEPT
root@xubweb:~# iptables -A OUTPUT -p tcp -o eth0 --sport 20 -d 0/0 -j REJECT
root@xubweb:~# iptables -A OUTPUT -p tcp -o eth0 --sport 21 -d 0/0 -j REJECT
root@xubweb:~# 
```

6) I made the directory and saved the IPtables. So the IPtables should look like the following:



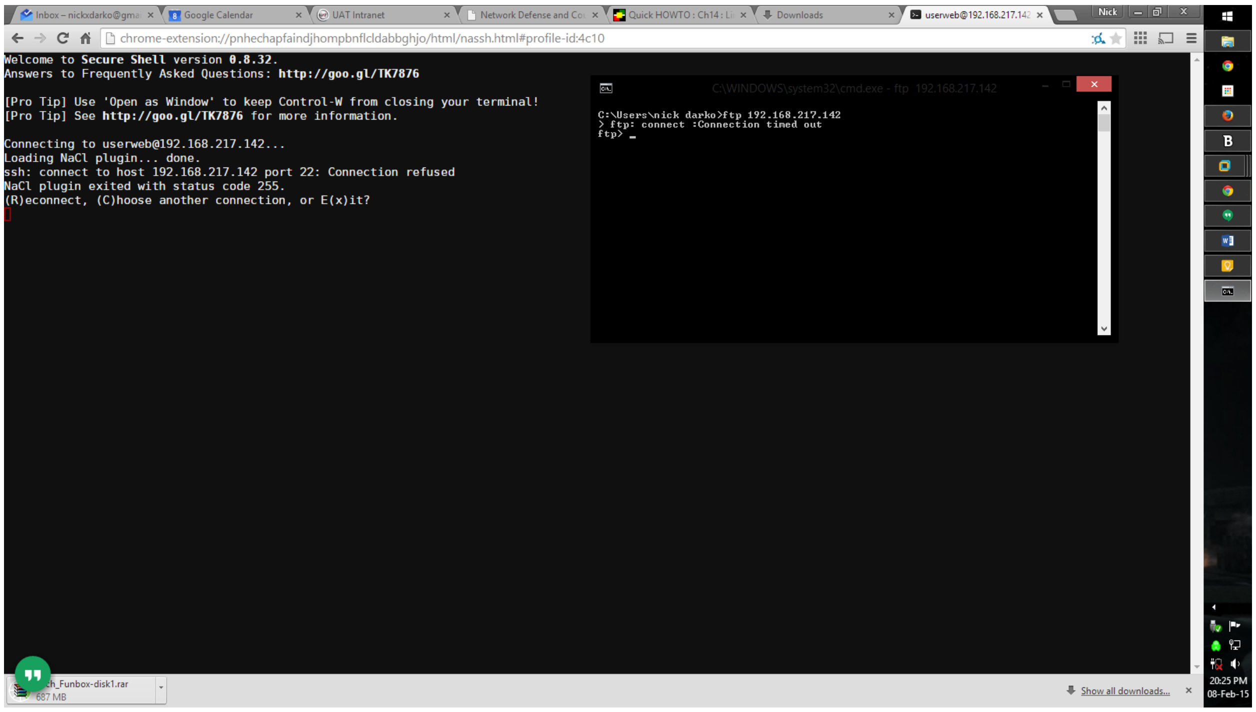
```
root@xubweb:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  192.168.217.141      anywhere             tcp dpt:ssh
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:ftp-data
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:ftp

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
REJECT     tcp  --  anywhere             anywhere             tcp spt:ssh reject-with icmp-port-unreachable
REJECT     tcp  --  anywhere             anywhere             tcp spt:ftp-data reject-with icmp-port-unreachable
REJECT     tcp  --  anywhere             anywhere             tcp spt:ftp reject-with icmp-port-unreachable
root@xubweb:~#
```

6) Outgoing SSH / FTP connections from Xubuntu web server VM successfully blocked.



```
root@xubweb:~# ssh userdev@192.168.217.143
ssh: connect to host 192.168.217.143 port 22: Connection refused
root@xubweb:~# ftp 192.168.217.143
ftp: connect: Connection refused
ftp> exit
root@xubweb:~#
```

7) Incoming SSH / FTP connections from host to Xubuntu web server VM successfully blocked.



```
Welcome to Secure Shell version 0.8.32.
Answers to Frequently Asked Questions: http://goo.gl/TK7876

[Pro Tip] Use 'Open as Window' to keep Control-W from closing your terminal!
[Pro Tip] See http://goo.gl/TK7876 for more information.

Connecting to userweb@192.168.217.142...
Loading NaCl plugin... done.
ssh: connect to host 192.168.217.142 port 22: Connection refused
NaCl plugin exited with status code 255.
(R)econnect, (C)hoose another connection, or E(x)it?
```

```
C:\Users\nick darko>ftp 192.168.217.142
> ftp: connect :Connection timed out
ftp> _
```

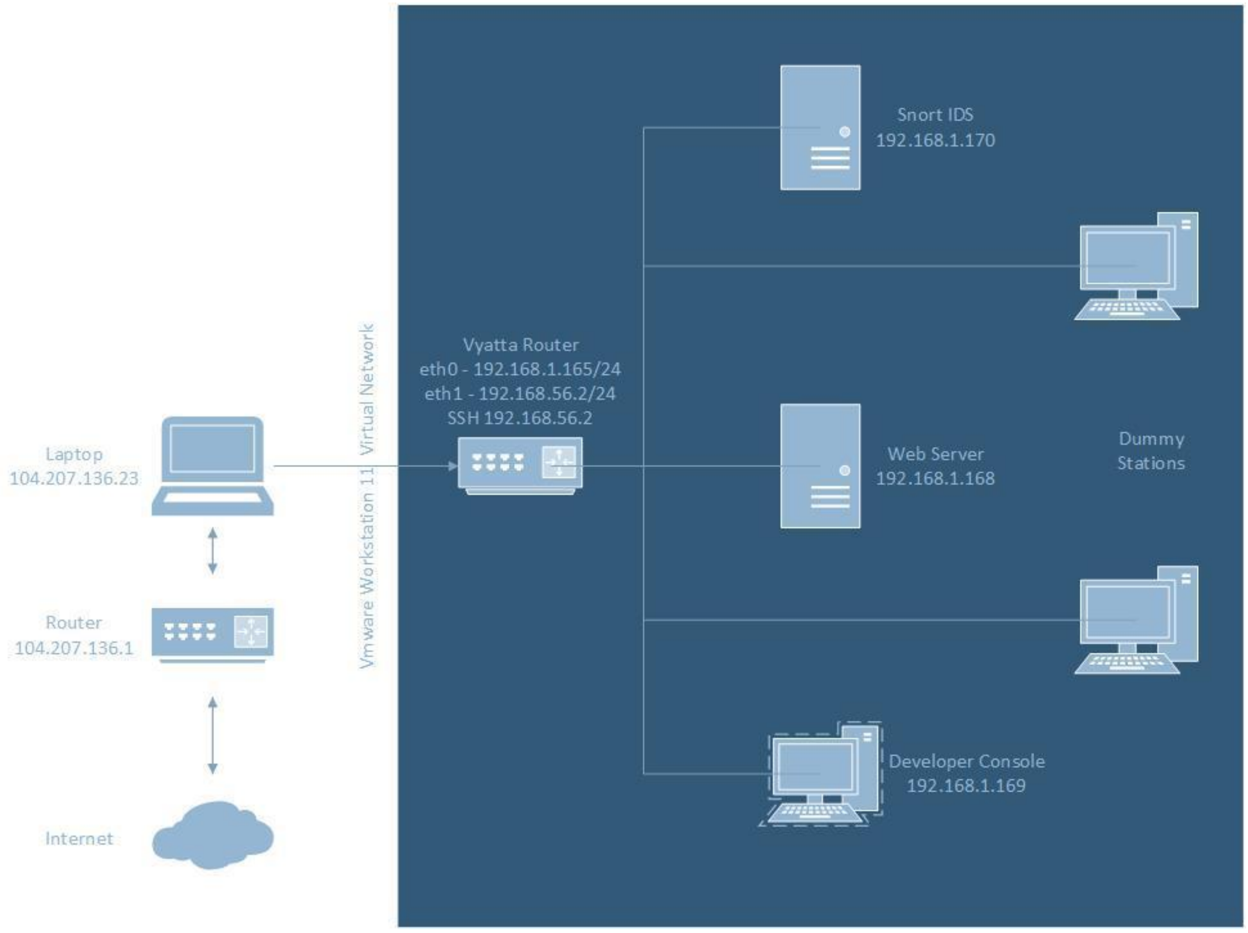
Lab/Activity Summary:

Minor issues came up a lot, but I couldn't find a way to delete specific rules and since the rules were so few, iptables –F came in handy a bit. Trying to get some of the connections to work was tricky which is what made me have to flush and readd rules.

Lab Memorandum

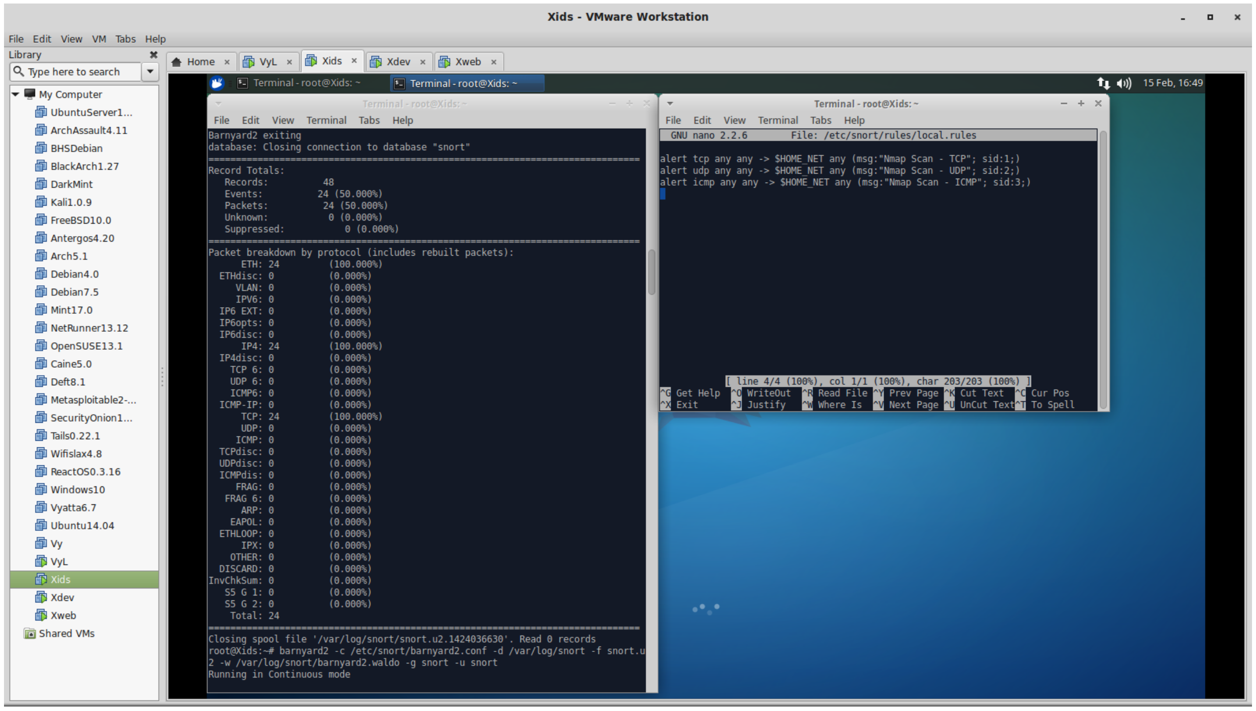
Professor:	**Barrett**
Course:	**NTW415, Network Defense & Countermeasures**
Student:	**Nick Somerville**
Date:	**15-Feb-15**
Lab / Activity:	**Final Project**

1) Brief description with network diagram.



This is my virtual network running off my laptop with Linux Mint complete with configured AT&T Motorola router on class A network. The Vyatta firewall / router VM is run ahead of the other VMs to ensure that they connect to Vyatta. Vyatta has inbound Internet connection on eth1 host-only adapter while outbound for the other VMs with NAT on eth0 NAT adapter. It’s also configured to use DHCP on outbound, listen for SSH connections on 192.168.56.2, and allow said SSH and HTTPS for other VMs connected (thwarting HTTP?). The Snort IDS server is configured with Snort to log, in my virtual network, ALL traffic in order to learn general from unusual. It's set to recognize any and all TCP, ICMP, and UDP connections on the Vyatta network with a constant connection since Snort and Barnyard2 start up on boot. The web server, which doesn't actually act as a web server since we never installed apache2 for the virtual network, is supposed to act as the web server for the virtual network. Any guest, dubbed dummy stations on the diagram below, are able to make an FTP connection into the web server, but users outside the network cannot. The only station permitted to create a SSH connection is the developer console and all outbound FTP and SSH is blocked. The developer console can accept SSH connections from any computer on any network inbound and outbound. It can accept FTP connections from any computer inbound or outbound as long as it is on the Vyatta network.

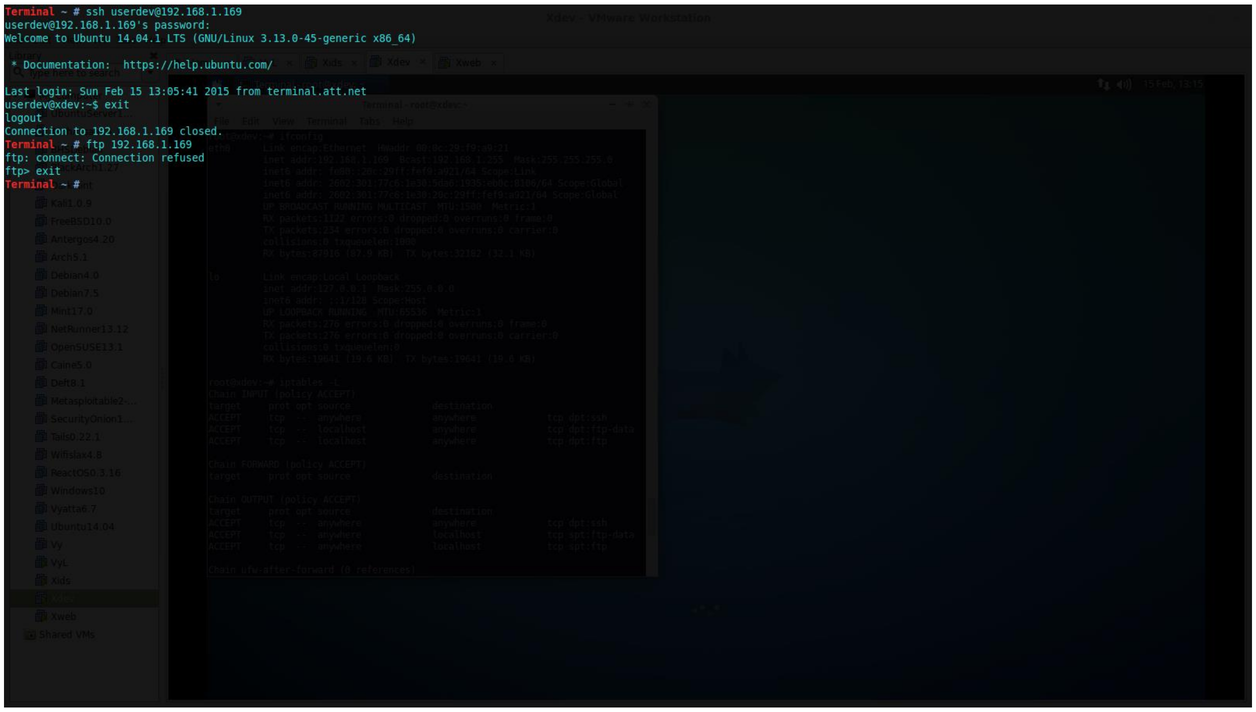
2) Perform Nmap scan with IDS detection.



```
Barnyard2 exiting
database: Closing connection to database "snort"
==============================================================================
Record Totals:
   Records:         48
   Events:         24 (50.000%)
   Packets:         24 (50.000%)
   Unknown:          0 (0.000%)
   Suppressed:          0 (0.000%)
==============================================================================
Packet breakdown by protocol (includes rebuilt packets):
      ETH: 24         (100.000%)
  ETHdisc: 0          (0.000%)
     VLAN: 0          (0.000%)
     IPV6: 0          (0.000%)
  IP6 EXT: 0          (0.000%)
  IP6opts: 0          (0.000%)
  IP6disc: 0          (0.000%)
      IP4: 24         (100.000%)
  IP4disc: 0          (0.000%)
    TCP 6: 0          (0.000%)
    UDP 6: 0          (0.000%)
    ICMP6: 0          (0.000%)
  ICMP-IP: 0          (0.000%)
      TCP: 24         (100.000%)
      UDP: 0          (0.000%)
     ICMP: 0          (0.000%)
  TCPdisc: 0          (0.000%)
  UDPdisc: 0          (0.000%)
  ICMPdis: 0          (0.000%)
     FRAG: 0          (0.000%)
   FRAG 6: 0          (0.000%)
      ARP: 0          (0.000%)
    EAPOL: 0          (0.000%)
  ETHLOOP: 0          (0.000%)
      IPX: 0          (0.000%)
    OTHER: 0          (0.000%)
  DISCARD: 0          (0.000%)
InvChkSum: 0          (0.000%)
   S5 G 1: 0          (0.000%)
   S5 G 2: 0          (0.000%)
    Total: 24
==============================================================================
Closing spool file '/var/log/snort/snort.u2.1424036630'. Read 0 records
root@Xids:~# barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.u
2 -w /var/log/snort/barnyard2.waldo -g snort -u snort
Running in Continuous mode
```

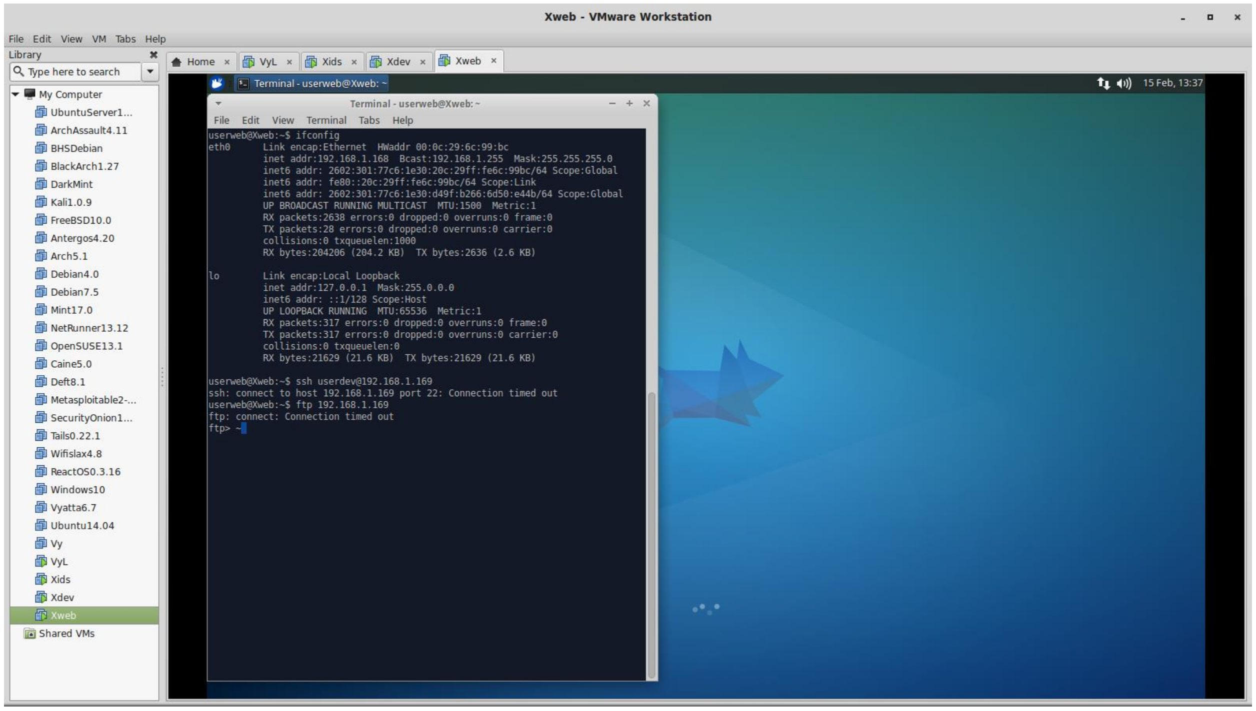
```
  GNU nano 2.2.6        File: /etc/snort/rules/local.rules

alert tcp any any -> $HOME_NET any (msg:"Nmap Scan - TCP"; sid:1;)
alert udp any any -> $HOME_NET any (msg:"Nmap Scan - UDP"; sid:2;)
alert icmp any any -> $HOME_NET any (msg:"Nmap Scan - ICMP"; sid:3;)

            [ line 4/4 (100%), col 1/1 (100%), char 203/203 (100%) ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

3) SSH into dev from host. Connected - Success. FTP into dev from host. Rejected – Failure/Success. As per week 4 assignment, the dev console blocks FTP connections that are not from the local network. Due to a time crunch, no guest was added, installed, and additionally configured to see FTP connection to act as "host," but FTP blocking was verified. Thus, blocking FTP from host was success.



```
Terminal ~ # ssh userdev@192.168.1.169
userdev@192.168.1.169's password:
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Last login: Sun Feb 15 13:05:41 2015 from terminal.att.net
userdev@xdev:~$ exit
logout
Connection to 192.168.1.169 closed.
Terminal ~ # ftp 192.168.1.169
ftp: connect: Connection refused
ftp> exit
Terminal ~ #
```

4) SSH into dev from web server. Blocked - Success. FTP into dev from host. Blocked - Success.

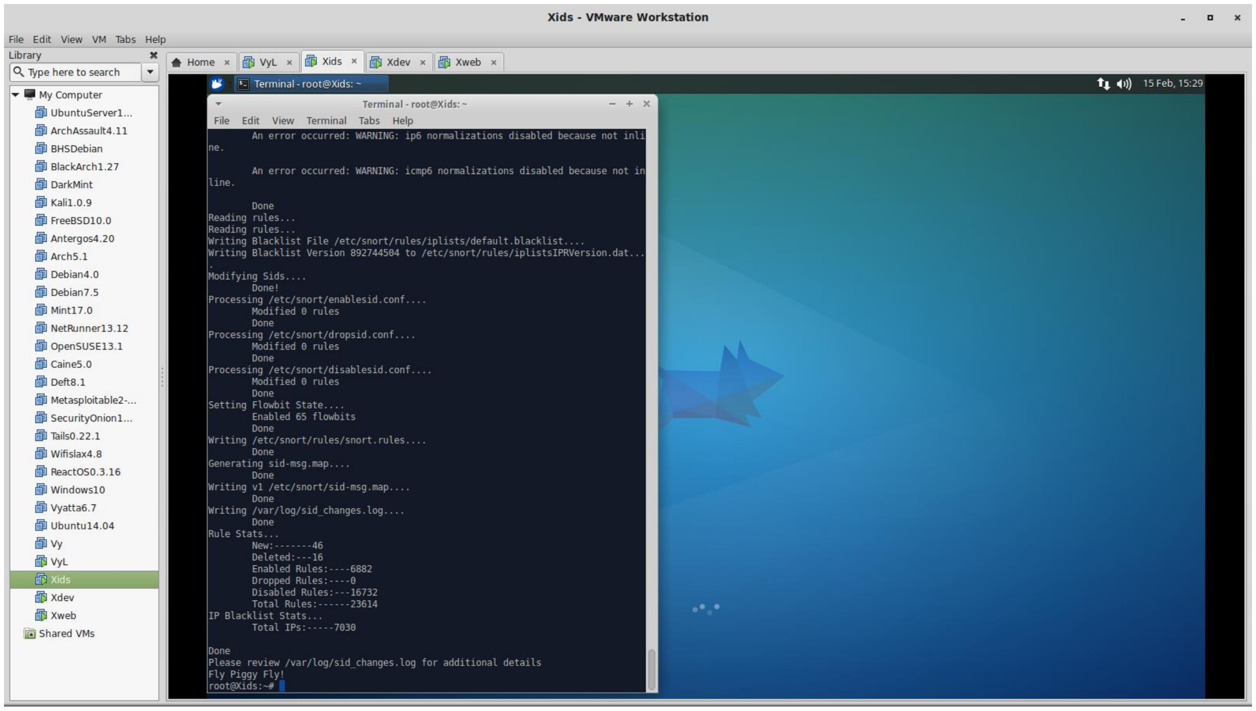


```
userweb@Xweb:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:6c:99:bc
          inet addr:192.168.1.168  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: 2602:301:77c6:1e30:20c:29ff:fe6c:99bc/64 Scope:Global
          inet6 addr: fe80::20c:29ff:fe6c:99bc/64 Scope:Link
          inet6 addr: 2602:301:77c6:1e30:d49f:b266:6d50:e44b/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2638 errors:0 dropped:0 overruns:0 frame:0
          TX packets:28 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:204206 (204.2 KB)  TX bytes:2636 (2.6 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:65536  Metric:1
          RX packets:317 errors:0 dropped:0 overruns:0 frame:0
          TX packets:317 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:21629 (21.6 KB)  TX bytes:21629 (21.6 KB)

userweb@Xweb:~$ ssh userdev@192.168.1.169
ssh: connect to host 192.168.1.169 port 22: Connection timed out
userweb@Xweb:~$ ftp 192.168.1.169
ftp: connect: Connection timed out
ftp> ~
```

5) See #3

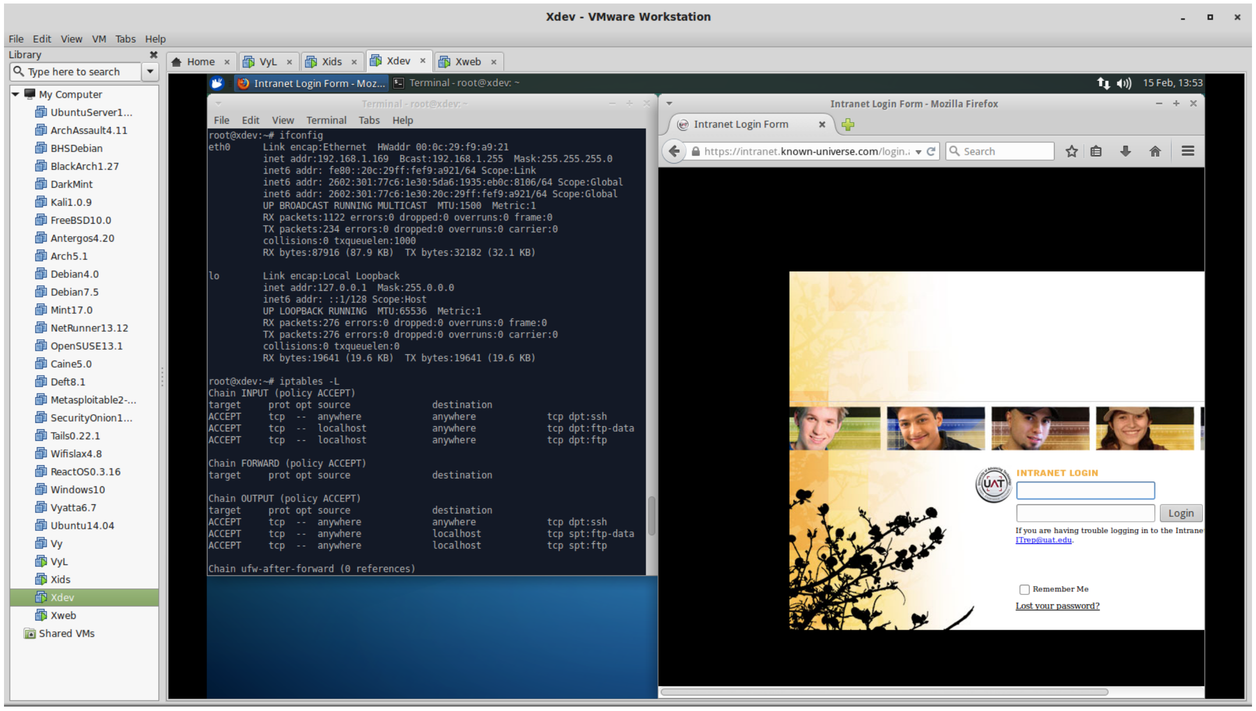
6) See #4

7) Snort rules updated via Pulled Pork. See screenshot after #8 for Vyatta connection.



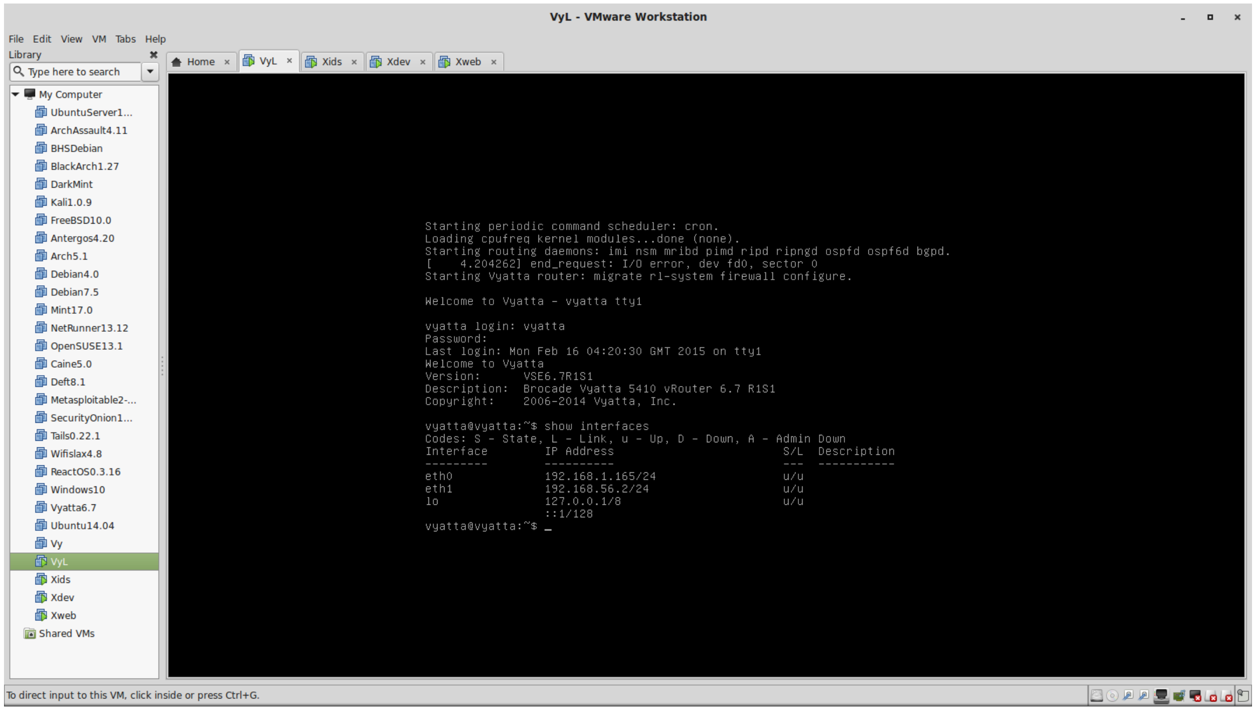
```
        An error occurred: WARNING: ip6 normalizations disabled because not inli
ne.

        An error occurred: WARNING: icmp6 normalizations disabled because not in
line.

        Done
Reading rules...
Reading rules...
Writing Blacklist File /etc/snort/rules/iplists/default.blacklist....
Writing Blacklist Version 892744504 to /etc/snort/rules/iplistsIPRVersion.dat...
.
Modifying Sids....
        Done!
Processing /etc/snort/enablesid.conf....
        Modified 0 rules
        Done
Processing /etc/snort/dropsid.conf....
        Modified 0 rules
        Done
Processing /etc/snort/disablesid.conf....
        Modified 0 rules
        Done
Setting Flowbit State....
        Enabled 65 flowbits
        Done
Writing /etc/snort/rules/snort.rules....
        Done
Generating sid-msg.map....
        Done
Writing v1 /etc/snort/sid-msg.map....
        Done
Writing /var/log/sid_changes.log....
        Done
Rule Stats...
        New:-------46
        Deleted:---16
        Enabled Rules:----6882
        Dropped Rules:----0
        Disabled Rules:---16732
        Total Rules:------23614
IP Blacklist Stats...
        Total IPs:-----7030

Done
Please review /var/log/sid_changes.log for additional details
Fly Piggy Fly!
root@Xids:~#
```

8) Browsed to UAT Intranet login page from dev console. See screenshot after #8 for Vyatta connection.



```
root@xdev:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:f9:a9:21
          inet addr:192.168.1.169  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fef9:a921/64 Scope:Link
          inet6 addr: 2602:301:77c6:1e30:5da6:1935:eb0c:8106/64 Scope:Global
          inet6 addr: 2602:301:77c6:1e30:20c:29ff:fef9:a921/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1122 errors:0 dropped:0 overruns:0 frame:0
          TX packets:234 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:87916 (87.9 KB)  TX bytes:32182 (32.1 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:65536  Metric:1
          RX packets:276 errors:0 dropped:0 overruns:0 frame:0
          TX packets:276 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19641 (19.6 KB)  TX bytes:19641 (19.6 KB)

root@xdev:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:ftp-data
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:ftp

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             localhost            tcp spt:ftp-data
ACCEPT     tcp  --  anywhere             localhost            tcp spt:ftp

Chain ufw-after-forward (0 references)
```

Vyatta connection screenshot



```
Starting periodic command scheduler: cron.
Loading cpufreq kernel modules...done (none).
Starting routing daemons: imi nsm mribd pimd ripd ripngd ospfd ospf6d bgpd.
[    4.204262] end_request: I/O error, dev fd0, sector 0
Starting Vyatta router: migrate rl-system firewall configure.

Welcome to Vyatta - vyatta tty1

vyatta login: vyatta
Password:
Last login: Mon Feb 16 04:20:30 GMT 2015 on tty1
Welcome to Vyatta
Version:      VSE6.7R1S1
Description:  Brocade Vyatta 5410 vRouter 6.7 R1S1
Copyright:    2006-2014 Vyatta, Inc.

vyatta@vyatta:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth0             192.168.1.165/24                  u/u
eth1             192.168.56.2/24                   u/u
lo               127.0.0.1/8                       u/u
                 ::1/128
vyatta@vyatta:~$ _
```

Lab/Activity Summary:

This was a fairly challenging and hard assignment that needed the assignments from week 1, 3, and 4 done correctly in order to succeed. Good thing week 1 and 4 were done correctly, but that means 3 days was spent on week 3's Snort IDS. I finally managed to get it to work with a few jury rigged commands here and there. Thus, we have a live, fully functional virtual network! I really hope to have more assignments this challenging in the future!