**Connecting VMs to Mininet as Hosts**

Stephen Blystone ([smb032100@utdallas.edu](mailto:smb032100@utdallas.edu))

Document Version 1.0

Last Updated: Feb 24, 2018

Table of Contents

[**Revision History** 3](#_Toc507363878)

[**Planning Your Network** 3](#_Toc507363879)

[**Our Network Plan** 3](#_Toc507363880)

[**Our Network Diagram** 4](#_Toc507363881)

[**Setting Up VirtualBox Networking Adapters** 4](#_Toc507363882)

[**VM0 (Ubuntu with Mininet) Configuration** 6](#_Toc507363883)

[**VM1 (pfSense) Configuration** 6](#_Toc507363884)

[**VM2 (Linux) Configuration** 7](#_Toc507363885)

[**Python Code for Connecting VMs to Mininet on VM0** 9](#_Toc507363886)

[**Testing Connectivity** 9](#_Toc507363887)

[**APPENDIX** 10](#_Toc507363888)

[Instructions for Installing Snort (I have not gone all the way through them yet) 10](#_Toc507363889)

[References 10](#_Toc507363890)

# **Revision History**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Changes** |
| 1.0 | 2/24/2018 | Initial Version |
|  |  |  |

# **Planning Your Network**

* It is important to plan your network before you begin these instructions.

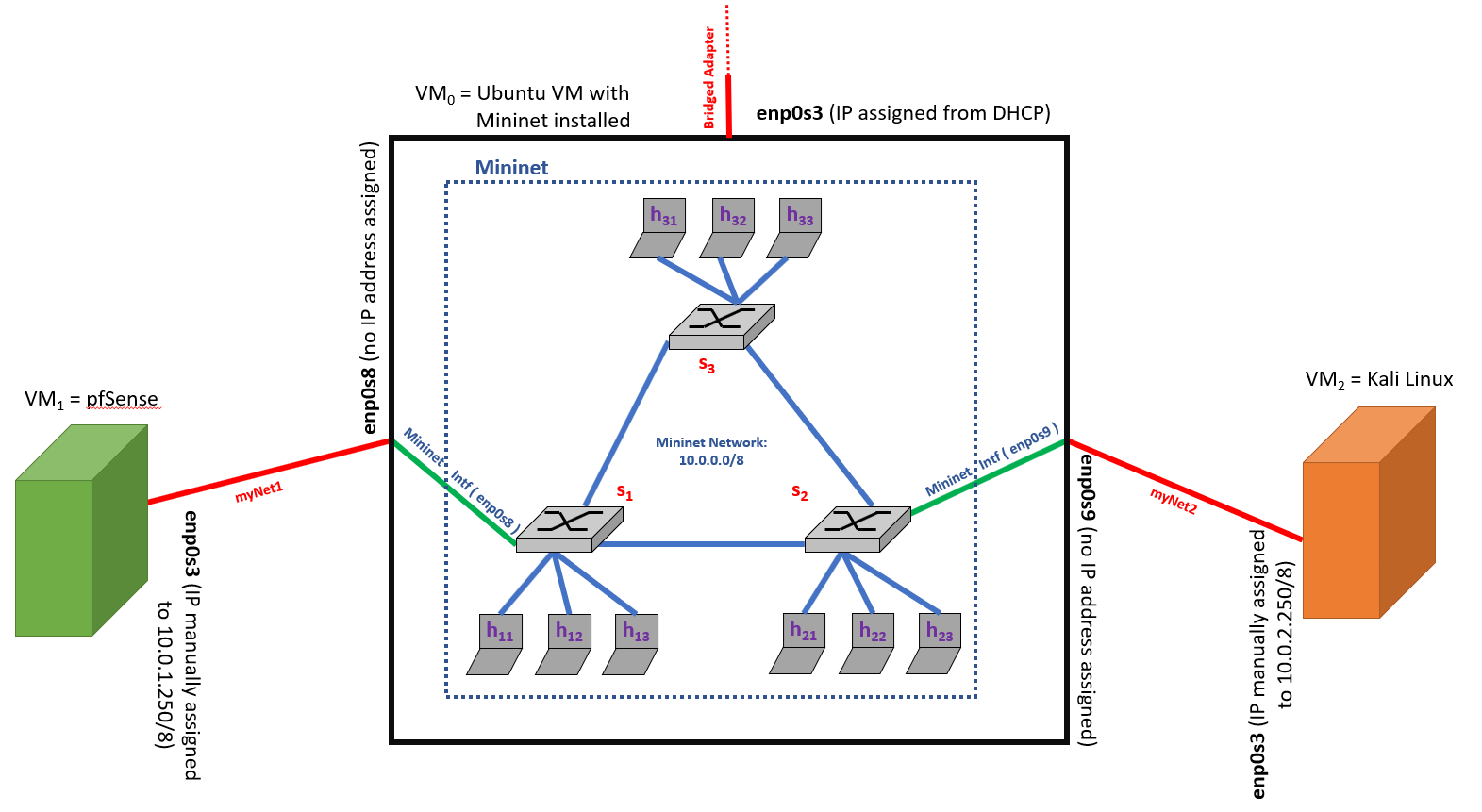
1. You must decide how many VMs you will connect to your main VM
   1. In our case we will connect 2 VMs to our main VM.
2. You must decide on names for the internal networks between the VMs. There must be a separate internal network between each VM and our main VM.
   1. In this case we need 2 separate internal networks (myNet1 and myNet2).
3. You must assign IP addresses to each VM that is not our main VM.
   1. IP assignments must be in the same range as our Mininet network.
   2. The default Mininet network is 10.0.0.0/8
      1. NOTE: /8 is the same as a subnet mask of 255.0.0.0
   3. I will assign an address of 10.0.1.250/8 to VM1
   4. I will assign an address of 10.0.2.250/8 to VM2
4. You must also decide what topology your Mininet network will use.
   1. For our purposes we will use a triangle topology of switches with three hosts on each switch.

# **Our Network Plan**

* We will create 3 VMs in VirtualBox.
* VM0 is our main VM that runs Ubuntu. Inside this we will install Mininet to run.
  + **NOTE**: The reason we need to have a VM running Ubuntu with Mininet installed instead of just the Mininet VM is because I was unable to get multiple interfaces working correctly on the Mininet Virtual Appliance. If you can get that working with multiple interfaces, then you can just assign the adapters as mentioned below, and that will work too.
  + VM0 has 3 Network Interfaces
    - Adapter 1 (enp0s3) is a Bridged Adapter that gets an IP address from your home router (by passing through your computer) using DHCP.
    - Adapter 2 (enp0s8) will connect to an internal network we are creating called myNet1. It will be used to connect VM0 with VM1.
      * **NOTE**: This interface will NOT be assigned an IP address.
    - Adapter 3 (enp0s9) will connect to an internal network we are creating called myNet2. It will be used to connect VM0 with VM2
      * **NOTE**: This interface will NOT be assigned an IP address.
* VM1 will be our VM running pfSense.
  + VM1 has 1 Network Interface
    - Adapter 1 (enp0s3) will connect to an internal network we are creating called myNet1. It will be used to connect VM0 with VM1.
      * We will manually assign this interface an IP address of 10.0.1.250 and a Subnet Mask of 255.0.0.0
* VM2 will be our VM running something else. For this document we will just use Kali Linux, but it could be used for Ubuntu, Snort, Parrot OS, or something else.
  + VM2 has 1 Network Interface
    - Adapter 1 (enp0s3) will connect to an internal network we are creating called myNet2. It will be used to connect VM0 with VM2.
      * We will manually assign this interface an IP address of 10.0.2.250 and a Subnet Mask of 255.0.0.0
* We will create a Python script that calls the Intf() method in the mininet.link file to connect our interfaces in VM0 to our external VMs.

# **Our Network Diagram**

* Below is the diagram showing the Network Plan and Mininet topology described in the previous sections:



# **Setting Up VirtualBox Networking Adapters**

* For VM0 we must configure 3 Network Adapters.

1. Create your VM
2. Go to Settings -> Network
3. On Adapter 1:
   1. Enable Network Adapter
   2. Attached to: Bridged Adapter
4. On Adapter 2:
   1. Enable Network Adapter
   2. Attached to: Internal Network
   3. Name: myNet1
      1. You can name this whatever you want, but it must match the name you give the Internal Network on VM1
   4. Click Advanced
      1. Change Promiscuous Mode to “Allow All”
         1. **NOTE**: This is required because the traffic that is coming in from myNet1 is destined for the Mininet network running inside your VM, and not for the VM’s interface connecting to myNet1.
         2. Promiscuous Mode tells your VM’s network interface to accept all packets arriving on myNet1 even if they are not for the address assigned on that interface.
5. On Adapter 3:
   1. Enable Network Adapter
   2. Attached to: Internal Network
   3. Name: myNet2
      1. You can name this whatever you want, but it must match the name you give the Internal Network on VM2
   4. Click Advanced
      1. Change Promiscuous Mode to “Allow All”
         1. **NOTE**: This is required because the traffic that is coming in from myNet2 is destined for the Mininet network running inside your VM, and not for the VM’s interface connecting to myNet2.
         2. Promiscuous Mode tells your VM’s network interface to accept all packets arriving on myNet2 even if they are not for the address assigned on that interface.

* For VM1 we must configure 1 Network Adapter

1. Create your VM
2. Go to Settings -> Network
3. On Adapter 1:
   1. Enable Network Adapter
   2. Attached to: Internal Network
   3. Name: myNet1
      1. You can name this whatever you want, but it must match the name you give the Internal Network for Network Adapter 2 on VM0
   4. Click Advanced
      1. Change Promiscuous Mode to “Allow All”
         1. **NOTE**: This is required because the traffic that is coming in from myNet1 is destined for the Mininet network running inside your VM, and not for the VM’s interface connecting to myNet1.
         2. Promiscuous Mode tells your VM’s network interface to accept all packets arriving on myNet1 even if they are not for the address assigned on that interface.

* For VM2 we must configure 1 Network Adapter

1. Create your VM
2. Go to Settings -> Network
3. On Adapter 1:
   1. Enable Network Adapter
   2. Attached to: Internal Network
   3. Name: myNet2
      1. You can name this whatever you want, but it must match the name you give the Internal Network for Network Adapter 3 on VM0
   4. Click Advanced
      1. Change Promiscuous Mode to “Allow All”
         1. **NOTE**: This is required because the traffic that is coming in from myNet2 is destined for the Mininet network running inside your VM, and not for the VM’s interface connecting to myNet2.
         2. Promiscuous Mode tells your VM’s network interface to accept all packets arriving on myNet2 even if they are not for the address assigned on that interface.

# **VM0 (Ubuntu with Mininet) Configuration**

* This section assumes that you have already installed the Ubuntu OS and have the VM running with the VirtualBox Network Adapters configured as stated in “Setting Up VirtualBox Networking Adapters” above.

1. Install Mininet

sudo apt-get install mininet

1. This is all that needs done on VM0 until after VM1 and VM2 are both ready.

# **VM1 (pfSense) Configuration**

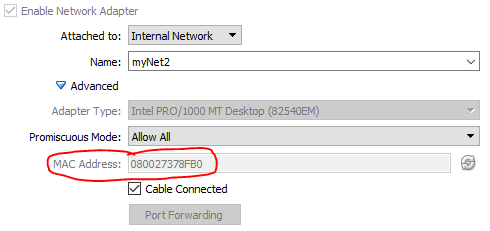
* This section assumes that you have already installed the pfSense OS and have the VM running with the VirtualBox Network Adapters configured as stated in “Setting Up VirtualBox Networking Adapters” above.

1. Enter option 2 “Set interface(s) IP address”
2. Configure IPv4 address WAN interface via DHCP? (y/n) n
3. Enter the new WAN IPv4 address: 10.0.1.250
4. Enter the new WAN IPv4 subnet bit count (1 to 31): 8
5. For a WAN, enter the new WAN IPv4 upstream gateway address.  
   For a LAN, press <ENTER> for none: <ENTER>
6. Configure IPv6 address WAN interface via DHCP6? (y/n) n
7. Enter the new WAN IPv6 address. Press <ENTER> for none: <ENTER>
8. Do you want to enable the DHCP server on WAN? (y/n) n
9. Do you want to revert to HTTP as the webConfigurator protocol? (y/n) n
10. VM1 is now set up to talk to VM0

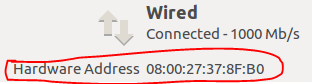
# **VM2 (Linux) Configuration**

* This section assumes that you have already installed the Ubuntu OS and have the VM running with the VirtualBox Network Adapters configured as stated in “Setting Up VirtualBox Networking Adapters” above.

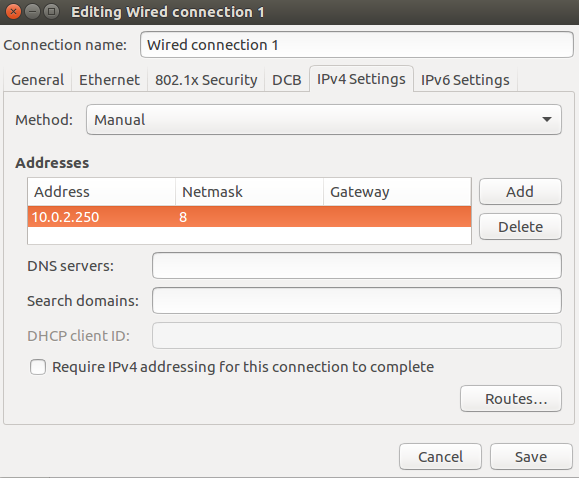
1. Go to System Settings
2. Go to Network Settings
3. Select the interface that connects to myNet2
   1. If you need help determining your interface, compare the MAC Address (Hardware Address) on the interface in Linux with the VirtualBox Network Adapter MAC Address for the Adapter in the Advanced menu.
      1. For example: MAC Address in the VirtualBox Network Adapter



* + 1. MAC Address (Hardware Address) in Network Settings in Linux



1. Click Options
2. Click IPv4 Settings
3. Select Method: Manual
4. Click “Add”
5. Enter the IP Address: 10.0.2.250
6. Enter the Netmask: 8
7. Leave Gateway blank.
8. Example:



1. Click Save
2. Toggle your network connection
   1. Click On to turn it Off
   2. Click Off to turn it back On
3. VM2 is now set up to talk to VM0

# **Python Code for Connecting VMs to Mininet on VM0**

* This code is based off the hwintf.py example on the Mininet GitHub page:  
  <https://github.com/mininet/mininet/blob/master/examples/hwintf.py>
* networkTopology.py creates the interfaces from the Open vSwitches in the Mininet network to the network interfaces on VM0
  + The networkTopology.py file is on our GitHub site:

<https://github.com/blynotes/CS6301_SDN/blob/master/Code/src/networkTopology.py>

* Since we have Promiscuous mode enabled, traffic from VM1 and VM2 can flow to and from their respective Open vSwitches in our Mininet network.

1. Run networkTopology.py to create our Mininet network and create the Intf’s from Mininet to the network interfaces on VM0

sudo python networkTopology.py

* 1. If you get an error about the mininet module, make sure you ran “sudo apt-get install mininet” in the VM0 section above.
  2. **NOTE**: The script mentions “you may need to reconfigure the interfaces for the Mininet hosts…”. You can ignore this if you followed the Network Diagram shown previously. This message is only if some of your host IPs are the same as your VM1 or VM2 IPs.

1. Make sure you connected to the controller and there were no errors.
   1. If you could not connect to the controller, check the IP address and port number and update accordingly in the file.

# **Testing Connectivity**

1. First confirm you can ping from h11 to h22
   1. If you cannot, make sure your controller is working correctly and allowing forwarding packets.
2. …TO BE CONTINUED…

# **APPENDIX**

## Instructions for Installing Snort (I have not gone all the way through them yet)

* Really detailed walkthrough:

<http://www.ubuntu-howtodoit.com/?p=138>

* Shorter walkthrough, but I ran into issues with this tutorial:

<https://www.upcloud.com/support/installing-snort-on-ubuntu/>

* Snort homepage

<https://snort.org/#get-started>

## References

* Good intro for creating custom Mininet topologies, integrating Mininet into Atom Editor, and creating Datacenter topologies with Mininet

<https://inside-openflow.com/2016/06/29/custom-mininet-topologies-and-introducing-atom/>