

Assignment One - Setting up the Lab Environment

Introduction

In this assignment, I set up a firewall using PFSense and VMWare Workstation Pro 16 running on a Windows 10 host machine. The process was straight forward, but that doesn't mean I didn't run into issues. Hopefully by the end of reading this document, anyone with similar issues or questions about setting up a firewall properly will be able to use my mistakes as a lesson and will have an easier time accomplishing this task.

Network Setup

The first thing I did to create the firewall lab environment was to set up the virtual network in VMWare Workstation. This can be found by navigating to *Edit > Virtual Network Editor* in the menu bar at the top of the screen. Once the editor is opened, it's good practice to reset the virtual network settings to default using the *restore defaults* button at the bottom of the dialogue box. This is helpful because some previously configured settings may not play well with the required network settings of this PFSense installation. Once the defaults have been restored, the new virtual networks can be added. To do this, I selected the *Add Network* button under the list of configured networks, then chose VMnet2 as the network to add. To make the networks more organized, it would be smart to rename the newly added network to Windows. Ensure the network is set to *Host-Only* and the *Use local DHCP service to distribute IP addresses to VMs* option is unchecked. Once DHCP is disabled, static IP addresses can now be configured. Refer to the Figure 1 below for the IP settings I used. Now that the windows network is set-up repeat these steps for the Linux network and rename accordingly. Choose a different network (I chose VMnet3) for the Linux network to sit on. Now, the windows interfaces should be changed to reflect the names configured in VMware. Right-click the network icon in the lower right corner on the taskbar, then select *Open Network and Sharing Settings*. Click on *Change Adapter Settings*, then right click VMnet2 and rename it to *Windows – VNET*. Do the same with VMnet3 and the Linux network. Now, the virtual network should be completely set up, and the next phase of the installation can be commenced.

Figure 1-1:

New Network Name	Original Network Name	Subnet IP	Subnet Mask
Windows	VMnet2	192.168.156.0	255.255.255.0
Linux	VMnet3	192.168.156.128	255.255.255.128

Firewall Installation

To install the firewall application, I first went to www.pfsense.org to download the latest ISO of the PFSense image. It is a large and compressed file, so once its downloaded, I extracted the contents to my computer using WinRAR. Then, I navigated to the VMware home page and clicked *Create a New Virtual Machine*, selected *Custom* and clicked next. On the next screen, the *Choose the Virtual Machine Hardware Compatibility* dialogue window should be present. Select *Workstation 12.x* and click next. Now the iso that was downloaded before must be selected. I clicked browse, then navigated to the drive where I stored the ISO file. Next, the machine was named PFSense. I also wanted to install the machine onto my portable SSD so that I can work on it from another machine, so I had to change the default path to one on that drive. Once that's complete, I kept everything default and just hit next until the *Specify Disk Capacity* Window appeared. I chose 20GB, as the PFSense machine doesn't need a large amount of storage. I hit next until the *Ready to Create Virtual Machine* dialogue box appeared. The PFSense box requires two interfaces, a WAN and a LAN, so I clicked on *Customize Hardware* and added a second network adapter, which was set to the Windows virtual network I had set up before. Now, the machine can be booted and the PFSense installer can begin its process. I went through the default menus until I had the option to select *Auto-UFS (UEFI)*. This will begin the installation. After it installed, I went through all the default menus and said no to anything being asked until it was ready to reboot. Upon reboot, the LAN adapter must be configured to match the local windows network. To do this, select option 2 (configure IP addresses), then select the LAN adapter (this should also be option 2). Set this IP Address as 192.168.156.1. Once the LAN adapter is configured, the Windows 10 host should be able to access that IP from a browser and be greeted with the PFSense landing page, where the default PFSense login (see below) will be put in. PFSense will then direct to the guided set up wizard, which can be configured to whatever specifications are needed.

Default PFSense Login

Username: admin

Password: pfsense

Troubleshooting

As any IT professional will tell you, no installation is complete without at least a few issues. The first one I ran into was when I tried to install the ISO, and upon restart, the VM would boot back into the install prompt instead of the PFSense interface. After trying for a long while to figure out why this happened, I found that the boot path of the VM was set to BIOS, and NOT UEFI. To fix this, I went into the settings of the VM in VMware, clicked advanced, then changed the booth path to UEFI. PFSense then booted right into the intended interface. Another issue I ran into was that once I had PFSense up and running, I just could not get the Windows Host to communicate with the PFSense VM. The solution to this problem was to reset the virtual network editor to default, then rebuild it from scratch. This cleared any network settings I created in the past that might have interfered with the current configuration. I also had to make sure that both the network cards attached to the VM were set to the right network. At first, the second NIC was attached to another VMNet, instead of the Windows one. Once that was switched, I was able to ping the VM. An issue that others may run into is forgetting to set the IP address of the LAN. Without doing this, there is no way for the host to communicate with the PFSense VM. You'll also want to make sure that there is a rule allowing for packets to be sent from the LAN to the interface. This should be enabled by default, but it can't hurt to check it if this is causing any issues.