Building a Cyber Range for Penetration Testing

In this paper, I will discuss the need to a penetration range, and we will present a diagram for a range that will allow you to prepare for most of the different types of penetration testing engagements you may encounter.

Introduction

So, why do we want a penetration testing range? The number one reason is to provide us a "playground" where we can practice and perfect our skill without requiring authorization; moreover, without breaking the law in the process of learning how to hack.

This paper will take the approach of creating your own cyber range, so you can do your own in-house penetration testing. With the advent of the cloud based services, you could make the case that you can build a penetration testing environment in the cloud as well.

Additionally, the approach within in the paper will be that of the range being contained within a virtual environment that can be created on any machine, even a laptop if there are enough resources 😊. In my 2nd Edition of Building Virtual Pentesting Labs for Advanced Penetration Testing (https://www.packtpub.com/networking-and-servers/building-virtual-pentesting-labs-advanced-penetration-testing-second-edition). I dedicate an entire chapter to selecting the virtual software platform. In the book, also at the time of this paper, my preference is to use VmWare, but since many want to keep their costs low, in this paper I will use Virtual Box since it is free.

Design

To start with, we want to design our network architecture for our range. This is imperative and the design we choose and implement makes the penetration process a simple step by step methodology that does not change, all we do is change the targets, and apply the latest tools when there is a change in industry; therefore, we want a network design that provides the capability to represent our engagement, this is as follows:

1. External or blackbox testing
2. Gray box testing
3. Internal or white box testing

To accomplish our goal we want to create a number of different network segments that we can use to emulate these different approaches. Part of this will be establishing the Demilitarized Zone as well as public and private segments of the network architecture. The internal network will represent the internal machines that like in the majority of the existing networks will not be reachable from the outside external point due to RFC 1918 private IP addresses that are not routable from the Internet, so to test these machines you would emulate the client-side attacks, or as we like to refer to them "click here."

To meet the requirement of this goal, we need to create a total of 5 switches. Even though we are going to use Virtual Box for the exercise, being a VmWare veteran, my preference is to create the switch with the naming nomenclature of the VmWare environment. The five switches that we are going to create for the range are as follows:

1. VMnet08 – 192.168.177.0/24
2. VMnet02 – 10.2.0.0/24
3. VMnet03 – 10.3.0.0/24
4. VMnet04 – 10.4.0.0/24
5. VMnet05 – 10.5.0.0/24

A diagram of our five switches and their layout with our initial machines connected is shown in Figure 1

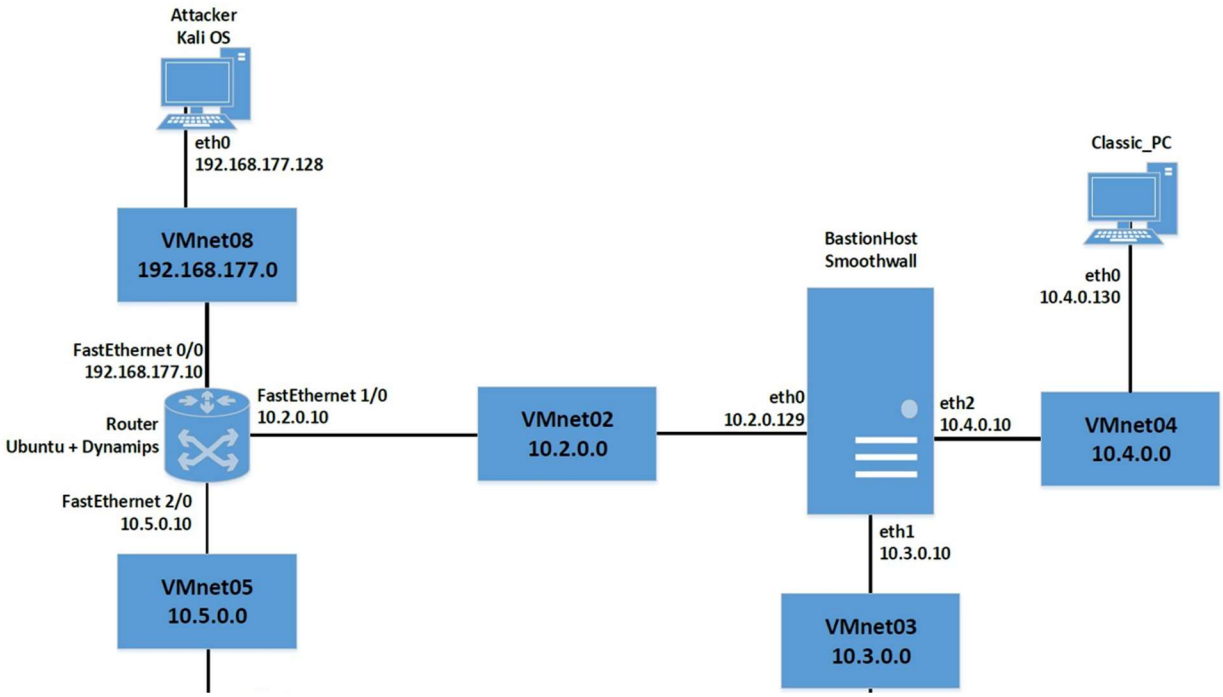

Figure 1:Initial network diagram

As the diagram shows, we have the Kali machine as our external attacker, and we have the Vmnet08 as our external zone switch, the VMnet05 is our off the data path DMZ, the VMnet03 is the screened subnet DMZ and the VMnet04 represents the inside machine. As discussed earlier in the paper, the network design allows for us to connect any machine that we want and we can place it into any of the zones which would represent the type of network we can encounter in our engagement. The Smoothwall firewall is used here, because it is easy to setup, and works well for us with a small RAM requirement as well. It is important to understand that the devices and the machines are interchangeable, and that is why the design is robust enough to allow you to practice and test against any vulnerability that you discover or is posted to the public. The router is the Dynamips machine, and does require a Cisco IOS, but it is not hard to setup an IPtables machine in the role of the router. You are in control of the design, and can add or delete machines and segments as required. In some cases you may encounter a 2nd Bastion Host firewall, for the most part it is the same process as we are covering in the paper, and in reality you really do not need to add it to be able to test the different environments you typically encounter in your testing.

Building the Network Switches

Start up the Virtual Box software, and once the program launches click on **File | Preferences | Network**. Once the window opens, click on the add icon, an example of this is shown in Figure 2

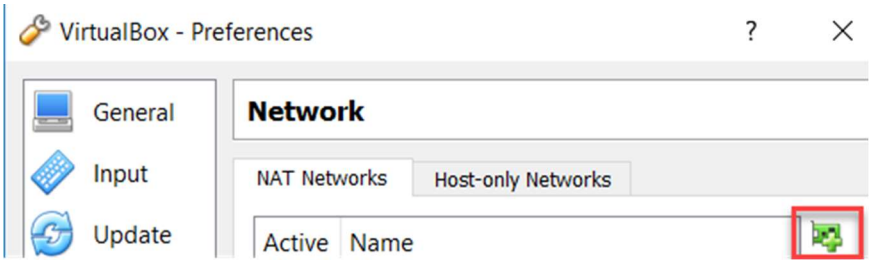

Figure 2: Network configuration in Virtual Box

When the add window opens double click the default name and in the name window enter the name for the switch, and then enter the IP address and select DHCP. You can elect to not set DHCP and perform the static addressing if desired, but the paper is recommending you set DHCP, so whenever you connect a machine an IP address is assigned, we can always enter a static IP if desired.

Once you have completed the configuration for your 5 switches you should see the configuration that is show in Figure 3

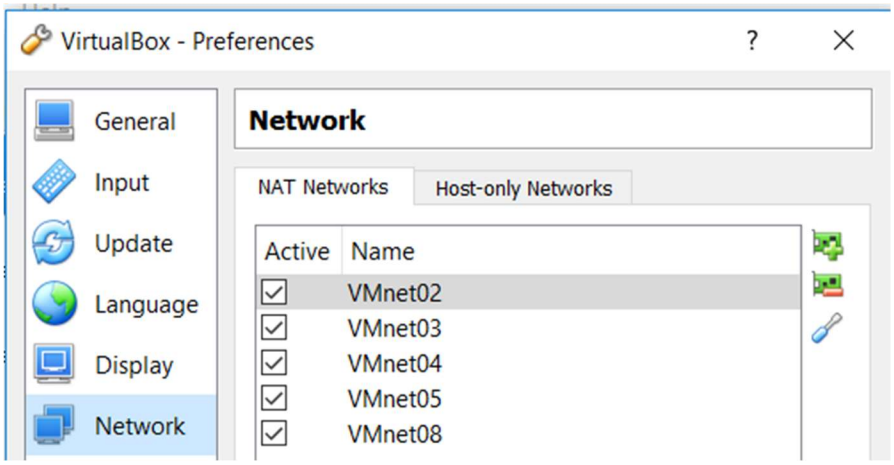

Figure 3: Configured NAT Networks

At this point the network is setup and established, so now all we have to do is add the components, these are the devices and the switches. The network diagram shows a router with 3 interfaces connected to the network and 3 of our segments; therefore, we will start there. This is the one area if you do not have or cannot get a Cisco IOS you will just configure a Linux machine with IPtables as your router device, there are plenty of references for this on the Internet, so we will not cover it here.

Adding to the Design

We want to establish the perimeter device as a Cisco since they continue to have the largest market share of the router market. Many will prefer to use the GNS3 graphical interface to the Dynamips software since it is maintained and updated more than the textual version of Dynagen, for our examples in this paper we will use the Dynagen software since, the GUI environment is for the most part point and click.

For this paper we are using the Ubuntu OS version 12.0.4 (this version works seamlessly with Dynagen) and we need to create a virtual machine for our router emulation software, we click on the Virtual Box **New** button and enter the details for the router as shown in Figure 4

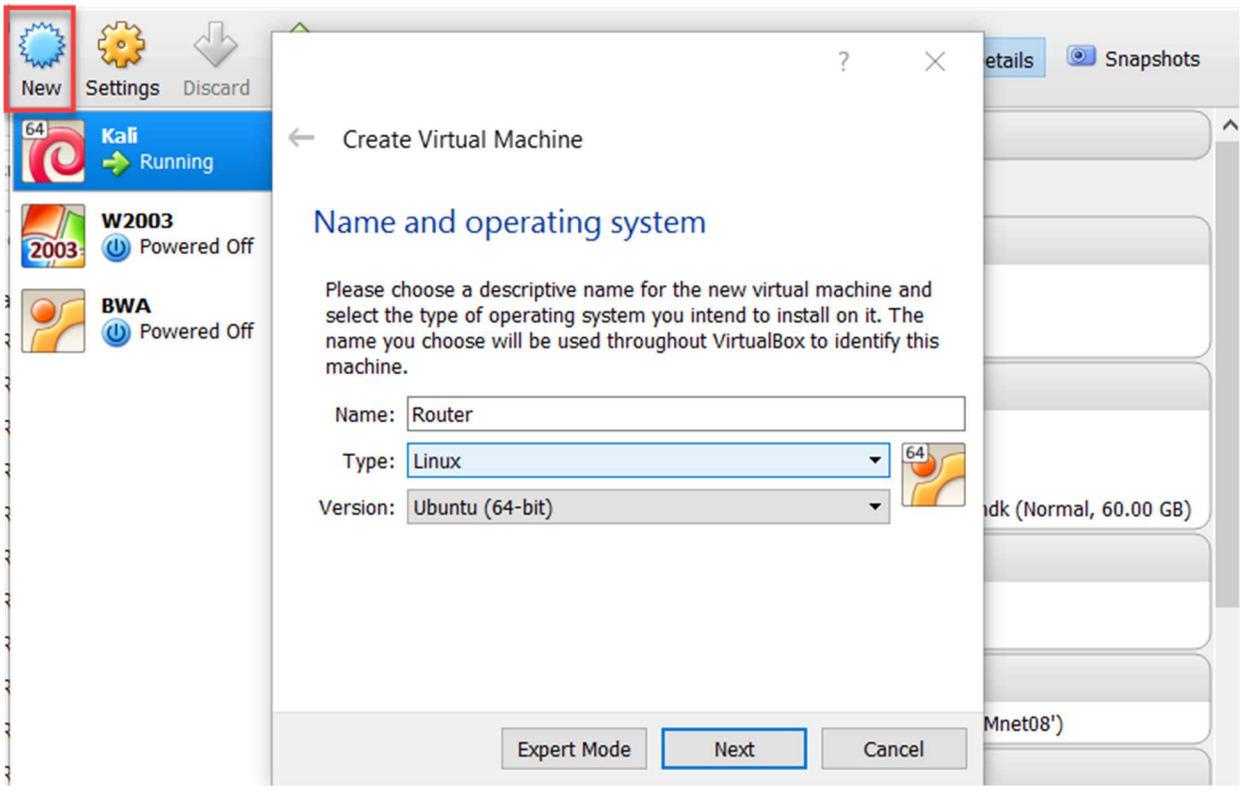

Figure 4: Creating the machine

Once we have setup the name of the system, we next want to configure the network cards that we are going to need for our router machine. In accordance with our diagram, we want to create the 3 network interfaces on the router machine. The interfaces will be connected to the following networks:

1. VMnet08 – 192.168.177.10 (eth0)
2. VMnet05 – 10.5.0.10 (eth1)
3. VMnet02 – 10.2.0.10 (eth2)

The 3 interfaces will represent a number of different network segments, it is important to note that the router emulation we are using for this paper is that of a Cisco 7200, and that provides us many interfaces to create zones and segments for our testing practice. An example of the completed router network configuration is shown in Figure 5

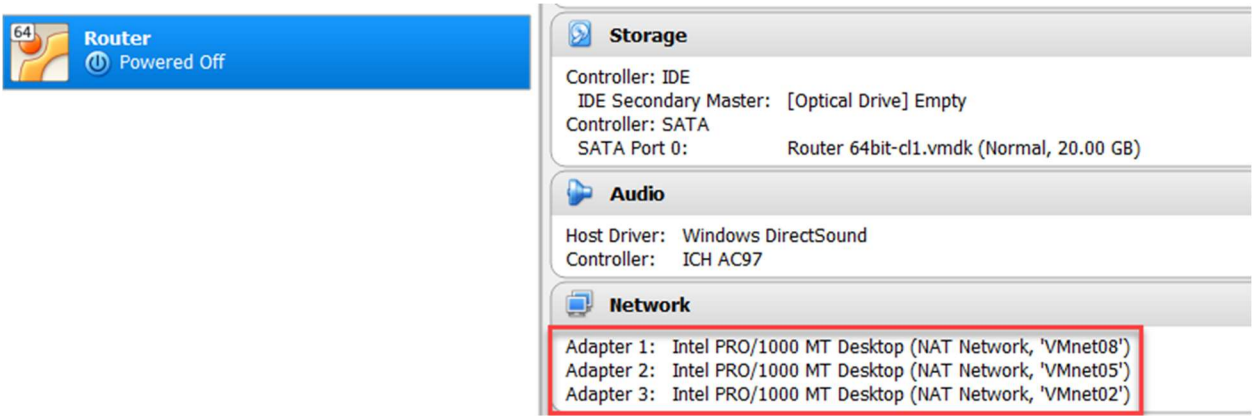

Figure 5 : Machine network adapters

For our purposes, we had a VMDK file from VMware, so we booted it there before creating the virtual box machine, and to get the software we just entered the following:

1. **apt-get install dynamips**
2. **apt-get install dynagen**

Once we finished this we copied the Cisco IOS image across to the machine. Once the network configuration is set, we are ready to power on the machine, and continue. After the machine boots up we need to configure the Cisco router, but before we do that we need to ensure our network cards are configured properly. There are a number of ways to do this, and in this case we use the more modern approach of using the tool within Ubuntu. My personal preference is to use the configuration file, but we will use the GUI here. Once you have your Ubuntu machine booted up, at the desktop right-click the network icon (the one with the arrows next to the time) and select the **Edit Connections.** Once the connections open, you will see the 3 network cards there, to configure the card, click on the **Edit** button, and this will open the interface configuration, we want to enter the configuration as follows for each card

1. eth0 IP Address – 192.168.177.10
    a. Netmask – 24
    b. Leave the gateway blank
2. eth1 IP Address – 10.5.0.10
    a. Netmask – 24
    b. Leave the gateway blank
3. eth2 IP Address – 10.2.0.10
    a. Netmask – 24
    b. Leave the gateway blank

An example for eth0 is shown in Figure 6

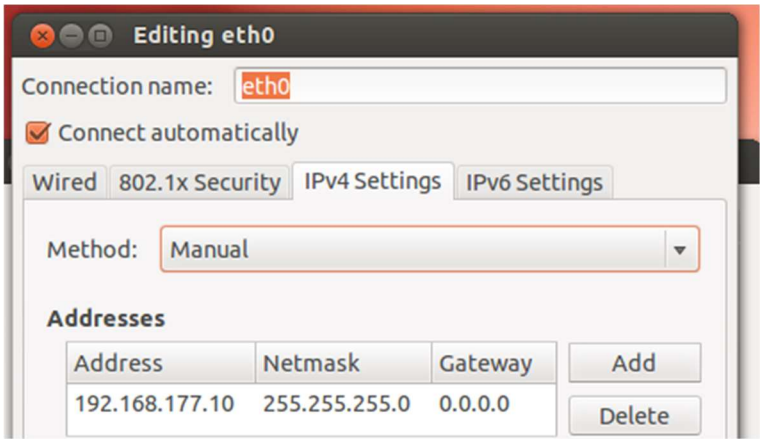

Figure 6: Editing network card in Ubuntu

Once the network cards are configured, we are ready to boot up the router and configure it, for those of you without a Cisco IOS image, just use IPtables. As mentioned it is available in most distributions, even Kali has it, the rules are just wide open. As we first begin our architecture, this is the way we will leave our devices and that is wide open, anything else can complicate our ability to reach the targets, so we always start with nothing filtering until we get all of our testing working then we add the protections. It is amazing, but the majority of the competitions that I am aware of have the network virtually flat or very close to it.

Before we boot machines and do the configuration we will create all of the network machines, and connect them to the architecture for the first layer. When we look at the network diagram, the second device is the Bastion Host which is representing our firewall, for the purposes of this paper we are going to use the Smoothwall firewall, this can be any firewall you want and could also be a machine with IPtables on it. It really is representing a device within the network, and it is not important which product or software you use here. Another favorite is the pfSense firewall. A note on the pfSense firewall selection, it will require that you enable the RFC 1918 addresses, because by default they are blocked. You will notice that the free and open source version of Smoothwall has not been update recently, but it does server our purposes of providing segments and an easy to use interface to configure our rules and filtering,

You can get the iso image for Smoothwall from www.smoothwall.org and once you have the image. Create the virtual machine by following the following steps

1. In the Virtual Box software, click on the **New.** In the window that opens, enter the name of the virtual machine as **BastionHost**
2. Select the platform as Linux with the version to match the version that you have downloaded
3. Accept the default setting of **512 MB** of RAM, this is more than enough for our Smoothwall machine
4. Click on **Next** and accept the default to create the virtual disk, and click on **Create**
5. Accept the default setting of creating a Virtual Disk Image and click **Next**
6. We want to keep the default setting of dynamically allocated for the hard disk and click **Next**
7. You can leave the default size set, and click on **Create**

8. This will result in the machine being created, the next step is to create the network interfaces, we need 3 of them as follows
    a. Adapter 1 – VMnet02
    b. Adapter 2 – VMnet03
    c. Adapter 3 – VMnet04
9. Once you have created the network interfaces, an example of the result of this is shown in the Figure 7

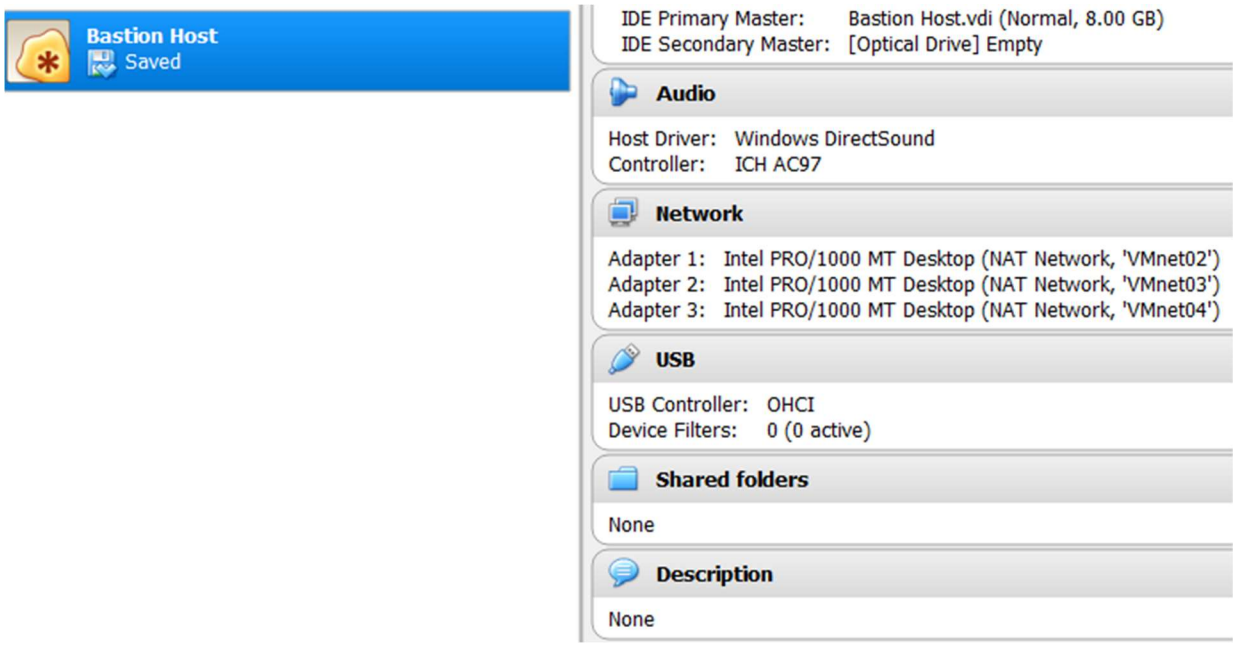

Figure 7: Bastion Host configuration

We next want to create our attacker machine, and as our diagram shows we are using the Kali machine for this, so we can download the image from www.kali.org, then we just need to create the machine using the same process that we have explored earlier within the paper, so we will not repeat the steps here, and the configuration of the completed machine is shown in Figure 8.

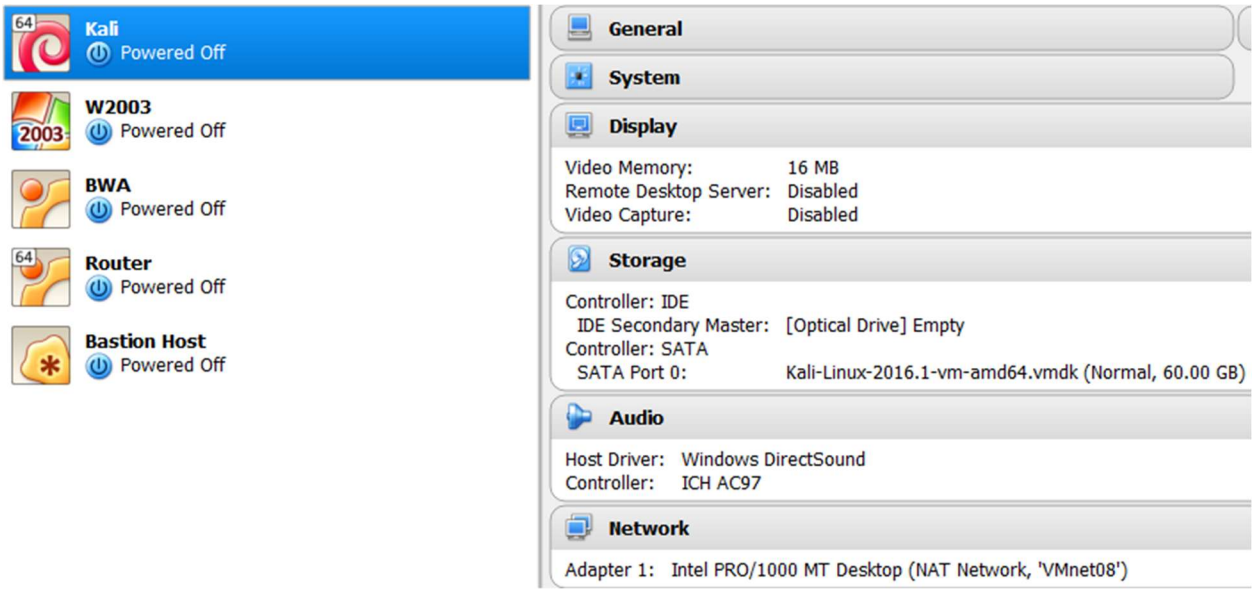

Figure 8: Kali configuration

We now have a network built with a number of machines and the switches to connect those machines together. The Smoothwall machine requires us to install it, and we will do that now. In the Virtual Box Manager, select the Bastion Host virtual machine, and click **Edit.** Once the editor opens, select the **Start** button, since there currently is no configuration in the machine, browse to the location of the ISO image you are going to use and select it and click on **Start.** This will start the installation process for the machine. Follow through the installation prompts until you get to the name entry, for the machine in this paper we will name it BastionHost. Read the explanation of the types of installations, and it is recommended that you leave it at the **Half-open** state, if you want an easier path to the other network segments then install it in an Open state. In the **Network Configuration Type** we want to press enter and select the option for the **Green + Orange and Red** as indicated in Figure 9

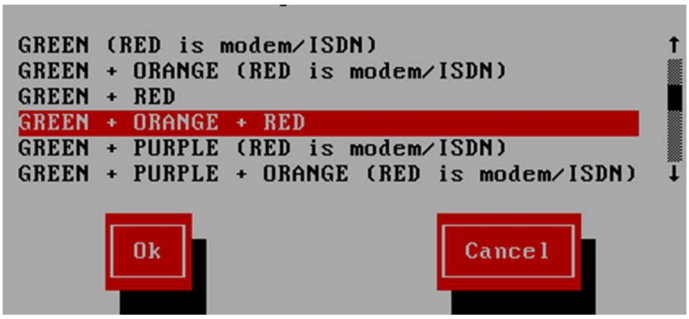

Figure 9: Network Configuration Type

The next thing we need to set is the card assignments, this can be tricky in Virtual Box, so I recommend you get the MAC address that is listed in the Adapter Advanced settings for each network card, so you assign the correct adapter to the right switch. This is a common problem and can make for frustration later, and another area where Vmware excels at over Virtual Box. We want to assign the cards as follows:

Red: – VMnet02

Green – VMnet03

Orange – VMnet04

Once the cards have been allocated the next step is to assign the IP addresses, for the example, and the diagram our addresses are as follows:

Red – IP address – 10.2.0.129

Green – IP Address – 10.3.0.10

Orange – IP Address – 10.4.0.10

The representation here, is the Red interface is the external threat, the Green is the insider, and the Orange defines our DMZ that we can place any number of services on. This provides the range with virtually all of the potential scenarios you could face in a professional security and penetration testing engagement; furthermore, the process is in place, and you can continue to build more and more segments and add machines, or just manipulate the interfaces if you are memory challenged. Once you have assigned the IP addresses to the interfaces, the remainder of the configuration is straightforward and dependent on what scenario you are selecting, so you can configure these as you choose. The DHCP server for the Green interface is good to Enable as it provides you the capability to place any machine on the VMnet04 switch and then have instant connectivity. Once you have configured the settings you should have three network interfaces defined as shown in Figure 10

```
       valid_lft forever preferred_lft forever
2: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc sfq state UP qlen 1000
    link/ether 08:00:27:ce:bf:2d brd ff:ff:ff:ff:ff:ff
    inet 10.2.0.129/24 scope global eth2
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fece:bf2d/64 scope link
       valid_lft forever preferred_lft forever
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc sfq state UP qlen 1000
    link/ether 08:00:27:0b:a2:9f brd ff:ff:ff:ff:ff:ff
    inet 10.3.0.10/24 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe0b:a29f/64 scope link
       valid_lft forever preferred_lft forever
4: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc sfq state UP qlen 1000
    link/ether 08:00:27:82:ed:5d brd ff:ff:ff:ff:ff:ff
    inet 10.4.0.10/24 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe82:ed5d/64 scope link
       valid_lft forever preferred_lft forever
```

Figure 10: Configured IP addresses

As a reminder, match the MAC address to the interface and do not worry about what eth0 is assigned. As long as the MAC address matches the machines should talk. We now have the firewall setup, and we next want to configure the first layer of our architecture, and we do that in the Dynamips router. Once you are in the machine, you start the Dynamips tool with the command **dynamips -H 7200** this starts the router emulator on the port 7200, the next step is to load the configuration file. The example in the paper here is the default config file, and it is shown in Figure 11

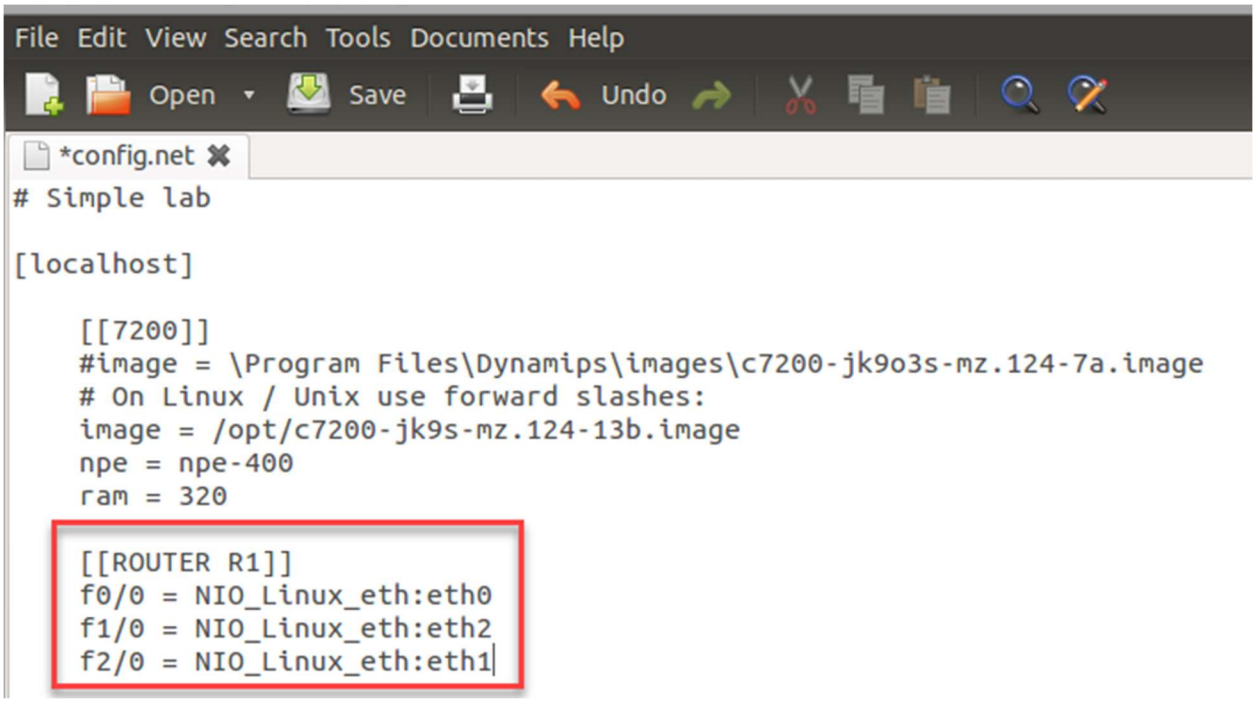

Figure 11: Dynagen configuration file

The key of the configuration is the router interfaces, and as is shown we are using 3 of them to match our network diagram earlier in the paper. Once we are done reviewing our configuration all that we need to do is start the router, and we enter **dynagen config.net**

Once the router has started we enter **console R1** and this provides us access into the router the same as if we connected to the console cable, once in we enter the enable mode with **en** and then configuration mode by entering **conf t** and then we have the router. And from here it is a matter of configuration. We have the device setup in our architecture, and we can move on to the next setup since we cannot within this paper go through all of the configuration and possibilities you now have with the router, you have the device and can do any router configuration with a 7200 that you choose. An example of the 3 interfaces when configured is shown in Figure 12

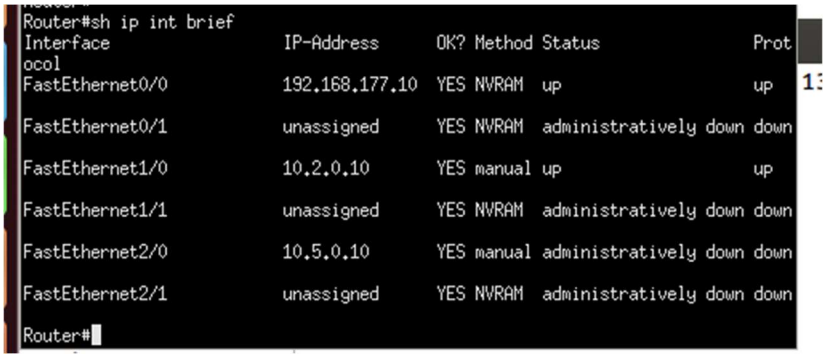

Figure 12: Cisco 7200 interfaces

At this point we just need to ping the Smoothwall interface that is connected to the Red side, and this is shown in Figure 13

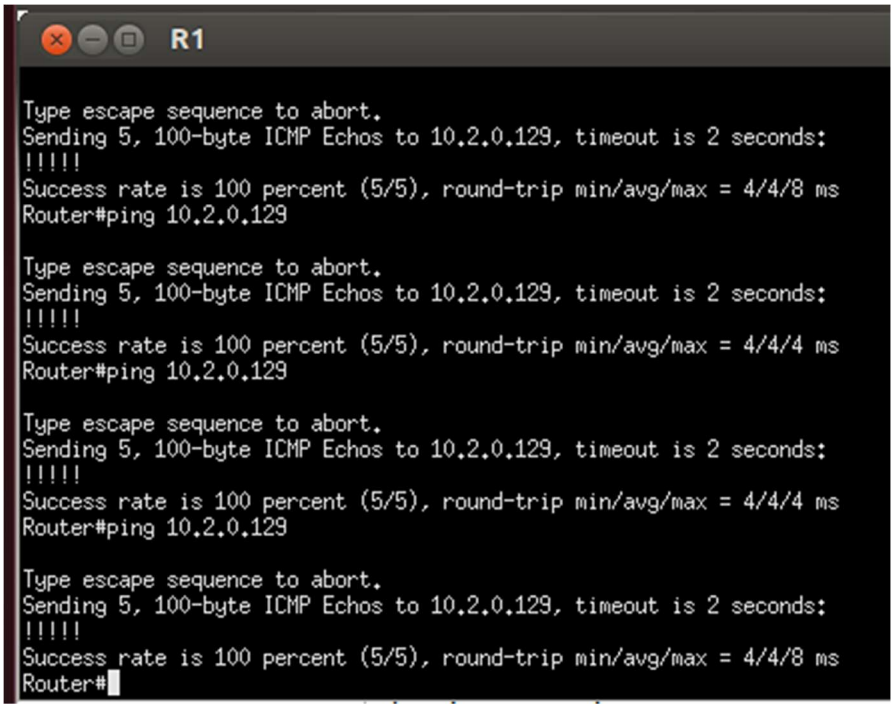

Figure 13: Successful ping across the architecture

We now have a 3 interface router that we can use to filter and control traffic just like our potential client would and look for where we can establish a bypass or at least access into the other networks.

We are now ready to configure the Smoothwall machine, and add access to the DMZ so that we can have a services subnet to test from the external position. We are using the OWASP Broken Wep Application virtual machine . We like to use this since it has many things we can practice with. We will connect the machine to our VMnet03 switch and then all we have to do is create the rule in Smoothwall, bring up our external machine and access it. The configuration for Smoothwall is done via the Green interface, so you need to connect a machine to the Green interface then access the Smoothwall machine using port 441. Once you have accessed the machine and logged in you will be in the main configuration area for Smoothwall, click on **Networking** to access the rules area. This is shown in Figure 14

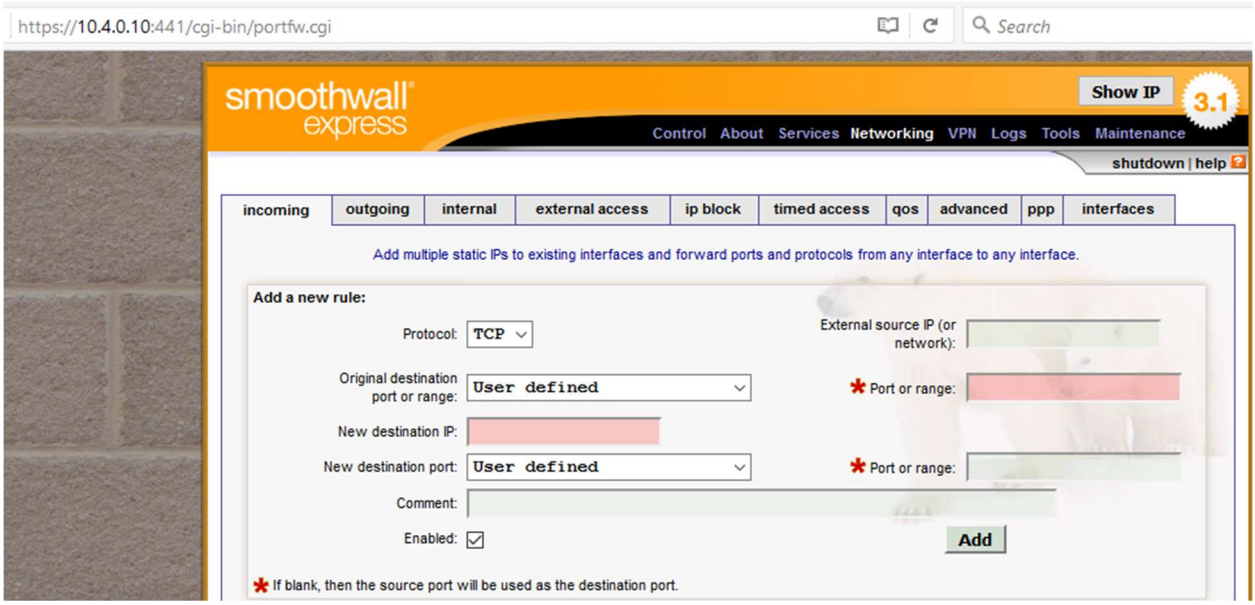

Figure 14: Incoming network configuration page

This is where we need to add the services that we want to reach for our testing purposes, since we have added the OWASP BWA virtual machine, we want to add port 80, enter the information as shown in Figure 15

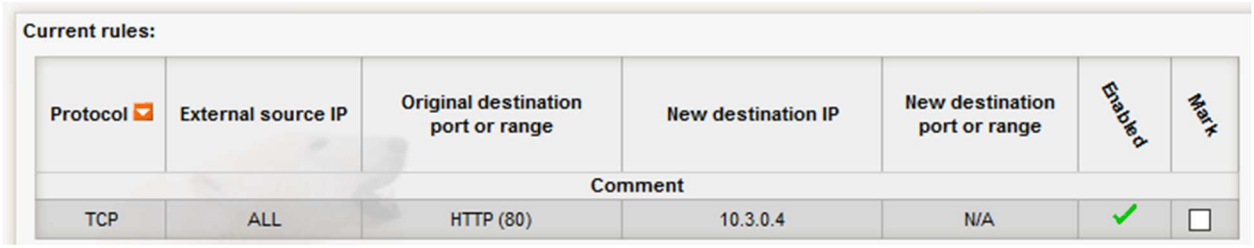

Figure 15: Completed DMZ rule

Then all we have to do now is connect the virtual machine to the switch and then we have our target to practice with; moreover, we have an entire enterprise architecture for a range. The one challenge that is left to you as the reader is, you have to configure the routing. The way to think of this is, at each device and/or machine you have to pass through to get to your target, there has to be a route, so if you look at the routing table and do not see the network of your destination then you have to add a route, and this takes place to and from the target, so whenever you have problems always check the routes. An example of the virtual machine being accessed via our enterprise architecture is shown in Figure 16

Figure 16: Connection to the OWASP machine in the DMZ

We now have our target, so now since this is a web application machine we can start to deploy our web application testing tool arsenal across the enterprise. The best thing is, we now are doing it through two layers of protection, so once we have this we then can add filtering and develop obfuscation and other techniques to defeat the filters when we test them. Enjoy experimenting with the enterprise you have created. When you have created this, then try your luck at creating the network architecture that I have on my laptop that is shown in Figure 17

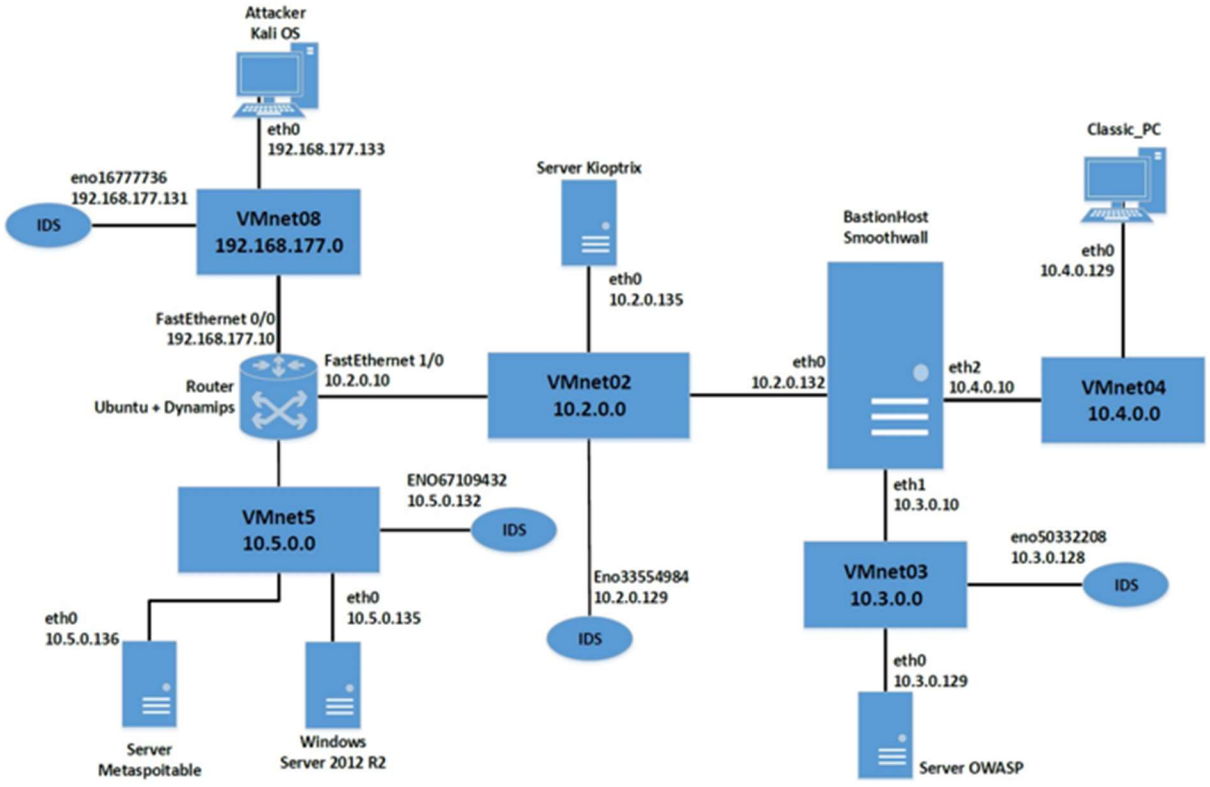

Figure 17: Complete penetration testing range

Summary

In this paper we have created the switches and several machines for our penetration testing range. The goal was to create the network segments then test them. Have fun with it, this is what penetration testing is all about, creating your own range so that you can enhance, hone and perfect your skills, feel free to connect with me on LinkedIN. Enjoy building your range!

Bio

Kevin Cardwell served as the leader of a 5 person DoD Red Team that achieved a 100% success rate at compromising systems and networks for six straight years. He has conducted over 500 security assessments across the globe. His expertise is in finding weaknesses and determining ways clients can mitigate or limit the impact of these weaknesses.

He currently works as a free-lance consultant and provides consulting services for companies throughout the world, and as an advisor to numerous government entities within the US, Middle East, Africa, Asia and the UK . He is an Instructor, Technical Editor and Author for Computer Forensics, and Hacking courses. He is the author of the Center for Advanced Security and Training (CAST) Advanced Network Defense and Advanced Penetration Testing courses. He is technical editor of the Learning Tree Course Penetration Testing Techniques and Computer Forensics. He has presented at the Blackhat USA, Hacker Halted, ISSA and TakeDownCon conferences as well as many others. He has chaired the Cybercrime and Cyberdefense Summit in Oman and was Executive Chairman of the Oil and Gas Cyberdefense Summit. He is author of Building Virtual Pentesting Labs for Advanced Penetration Testing, Advanced Penetration Testing for Highly Secured Environments 2nd Edition and Backtrack: Testing Wireless Network Security. He holds a BS in Computer Science from National University in California and a MS in Software Engineering from the Southern Methodist University (SMU) in Texas.