Building A Pentest Lab With Vagrant
Intro Sec Con 2020

Hammerton Mwawuda

April 2020

hammy25 h_mwawuda Hammerton Mwawuda

Contents

1 Introduction

1.1 Definitions

1.1.1 What are virtual machines?

A virtual machine is a computer file, an image, that through the use of a virtualization software can be used to emulate an operating system within another operating system.

A virtual machine allows users to run an operating system in an app window and behaves like a full, separate computer.

1.1.2 What is Vagrant?

Vagrant is an open-source software that enables users to create and configure virtual environments.

For more information visit: https://www.vagrantup.com

1.2 Why Vagrant?

With Vagrant you can easily configure and run virtual machines. Your only requirements are Vagrant and a virtualization software such as VMware, VirtualBox, Parallels or Hyper-V.

Compared to the tedious process of actively setting up different virtual machines, Vagrant allows you to set up machines by specifying configurations in a file called a Vagrantfile and starting the machine with a single command.

2 Installing Vagrant and Oracle VirtualBox

2.1 Installing VirtualBox

Install the latest VirtualBox (We are using VirtualBox because it is readily available). Installing VirtualBox is beyond the scope of this tutorial. Download virtualbox here.

Also download Oracle VM VirtualBox Extension Pack from the same page you downloaded VirtualBox. This is to support USB 2.0 and USB 3.0. Lack of this can cause your Kali Linux not to start.

2.2 Installing Vagrant

Windows: Download installer package from here and install it.

Linux: Download the correct package file from here depending on your Linux distribution and install with a package manager or the following commands.

```
$ sudo dpkg -i vagrant_2.2.X.deb #Debian based using dpkg
$ sudo apt install vagrant_2.2.X.deb #Debian based using apt

$ sudo rpm -iv vagrant_2.2.X.rpm #Redhat based using rpm
$ sudo yum install vagrant_2.2.X.rpm #Redhat based using yum
```

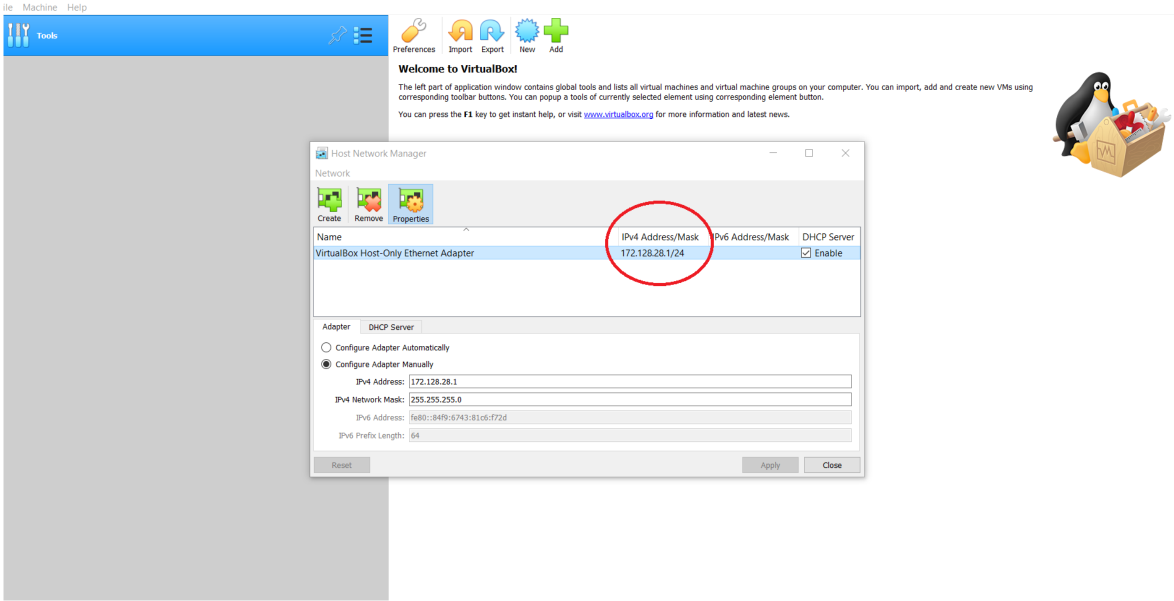
3 Setting Up the Virtual Machines

After installing Vagrant and VirtualBox we have achieved all our requirements and we are ready to start setting up our machines.

If you open VirtualBox, you'll notice that there are no machines set up yet. As it should be if it is a fresh install. However, you should check if there are any virtual networks set. Some VirtualBox installations come with pre-configured network adapters.

Check that by going to File > Host Network Manager

If you see a VirtualBox Host-Only Adapter with the IPv4 AddressMask value of 172.128.28.124 delete it. This may interfere with Vagrant as it tries to set up a virtual network.



Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your computer. You can import, add and create new VMs using corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding element button.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.

Name	IPv4 Address/Mask	IPv6 Address/Mask	DHCP Server
VirtualBox Host-Only Ethernet Adapter	172.128.28.1/24		☑ Enable

○ Configure Adapter Automatically

● Configure Adapter Manually

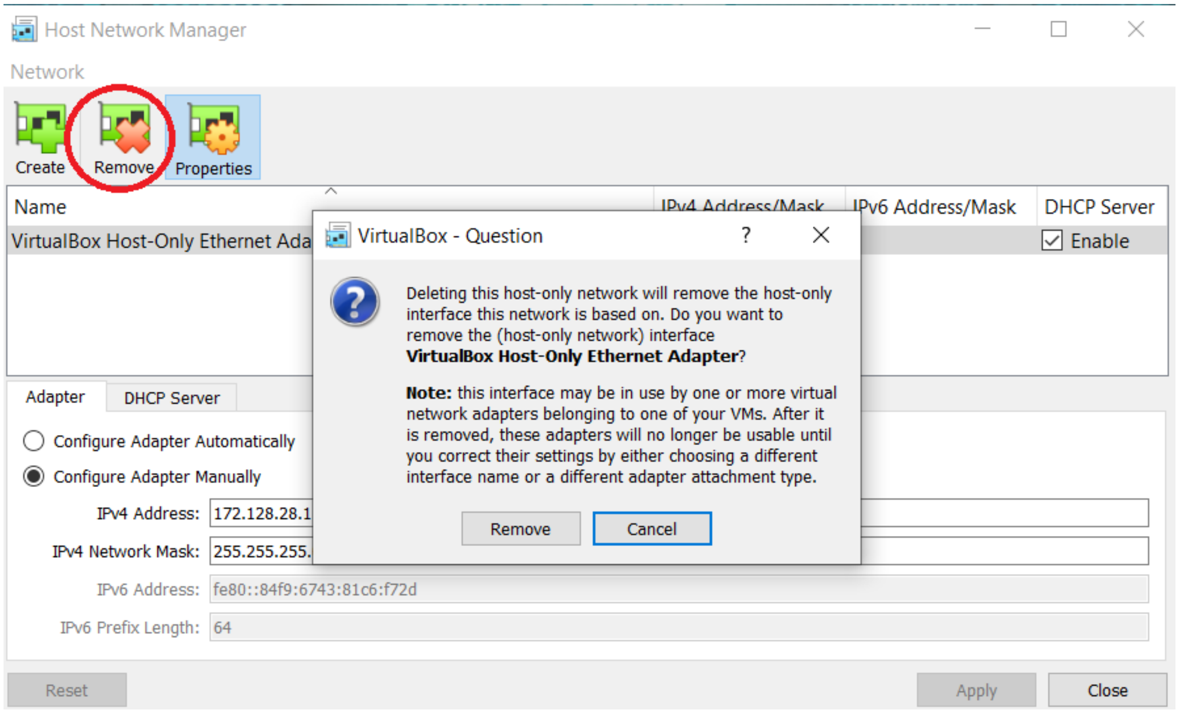
IPv4 Address: 172.128.28.1

IPv4 Network Mask: 255.255.255.0

IPv6 Address: fe80::84f9:6743:81c6:f72d

IPv6 Prefix Length: 64

Delete it with the *Remove* button.



VirtualBox Host-Only Ethernet Ada ☑ Enable

Deleting this host-only network will remove the host-only interface this network is based on. Do you want to remove the (host-only network) interface **VirtualBox Host-Only Ethernet Adapter**?

Note: this interface may be in use by one or more virtual network adapters belonging to one of your VMs. After it is removed, these adapters will no longer be usable until you correct their settings by either choosing a different interface name or a different adapter attachment type.

○ Configure Adapter Automatically

● Configure Adapter Manually

IPv4 Address: 172.128.28.1

IPv4 Network Mask: 255.255.255.

IPv6 Address: fe80::84f9:6743:81c6:f72d

IPv6 Prefix Length: 64

Now we can proceed to creating our virtual pentesting lab.

4 Creating Virtual Pentesting Lab

4.1 Getting Vagrant file

I have provided a Vagrant file at *link*. It is a slight modification from the one provided by Rapid7 at *link*. The additions I've made involve the following changes:

- The Ubuntu 14.04 machine to use 512 MB of RAM
- The Windows server 2008 to use 1024 MB of RAM
- Include a Kali2020.1 machine using 2048 MB of RAM in the multi-machine environment

4.2 Understanding the Vagrantfile

You don't have to necessarily understand the Vagrantfile to complete the set up but understanding it might help you in future if you want to add machines or change the specifications of the machines.

The Vagrant file:

```ruby
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant.configure("2") do |config|
  config.vm.synced_folder '.', '/vagrant', disabled: true
  config.vm.define "ub1404" do |ub1404|
    ub1404.vm.box = "rapid7/metasploitable3-ub1404"
    ub1404.vm.hostname = "metasploitable3-ub1404"
    config.ssh.username = 'vagrant'
    config.ssh.password = 'vagrant'

    ub1404.vm.network "private_network", ip: "172.128.28.3"

    ub1404.vm.provider "virtualbox" do |v|
      v.name = "Metasploitable3-ub1404"
      v.memory = 512
    end
  end

  config.vm.define "win2k8" do |win2k8|
    # Base configuration for the VM and provisioner
    win2k8.vm.box = "rapid7/metasploitable3-win2k8"
    win2k8.vm.hostname = "metasploitable3-win2k8"
    win2k8.vm.communicator = "winrm"
    win2k8.winrm.retry_limit = 60
    win2k8.winrm.retry_delay = 10

    win2k8.vm.network "private_network", ip: "172.128.28.4"

    win2k8.vm.provider "virtualbox" do |w|
      w.name = "Metasploitable3-win2k8"
```

```
32      w.memory = 1024
33    end
34
35    # Configure Firewall to open up vulnerable services
36    case ENV['MS3_DIFFICULTY']
37      when 'easy'
38        win2k8.vm.provision :shell, inline: "C:\\startup\\disable_firewall.bat"
39      else
40        win2k8.vm.provision :shell, inline: "C:\\startup\\enable_firewall.bat"
41        win2k8.vm.provision :shell, inline: "C:\\startup\\configure_firewall.bat"
42    end
43
44    # Insecure share from the Linux machine
45    win2k8.vm.provision :shell, inline: "C:\\startup\\install_share_autorun.bat"
46    win2k8.vm.provision :shell, inline: "C:\\startup\\setup_linux_share.bat"
47    win2k8.vm.provision :shell, inline: "rm C:\\startup\\*" # Cleanup startup scripts
48  end
49
50  # Comment the kali Linux Setup if you are running this from a Kali Linux machine
51  config.vm.define "kali20201" do |kali|
52    kali.vm.box = "kalilinux/rolling"
53    kali.vm.hostname = "kali"
54
55    kali.vm.network "private_network", ip: "172.128.28.5"
56
57    kali.vm.provider "virtualbox" do |k|
58      k.name = "Kali20201"
59      k.memory = 2048
60    end
61    kali.vm.provision "shell", inline: <<-SHELL
62      apt-get update
63    SHELL
64  end
65 end
```

Let's approach it line by line:

1. Line 1 - 2 : These are comments. Comments in the Vagrant file begin with the pound # sign. The comments here are just informing you that rest of the file is in Ruby syntax.

2. Line 4: This is where the configuration of our virtual machines begins. We are simply telling Vagrant to use version 2 of configuration and create an object called config to be used to configure the machines. At the time of writing, there are only two versions and two is the latest and most used.

3. Line 5 : Disables the creation of shared folders. Shared folders are meant to help you change some files inside the virtual machine from the host machine. We are

disabling it here because we don’t want to use that feature. Enabling it will also cause an error while running our *vagrantup* command since many machines are being configured. There will be a conflict.

4. Line 6 : Begins the configuration of our Ubuntu 14.04 machine.

5. Line 7 - 16 : We specify the settings of our Ubuntu 14.04 machine.

   - Line 7 : Defines the base box being used to set up the Ubuntu 14.04 box. This is defined in the Vagrant cloud.
   - Line 8 : Gives the machine a host name.
   - Lines 9 10 : Configure the ssh credentials of all the machines.
   - Line 12: Sets up the IP address of the Ubuntu 14.04 box. It creates a virtual network.
   - Line 14 - 16 : Instruct VirtualBox, which is our provider (Virtualization software) to create the box with the name ”MetaSploitable3-ub14.0.4” and to set up the box with RAM of 512MB.

6. Line 20 - 48: Configure our Windows 2008 Server. Similar to the setup of our Ubuntu 14.04 box above. The only differences are that it sets the Windows machine to communicate with the host machine through Windows Remote Management (winrm) at line 24 and winrm settings on line 25 & 26.

   Lines 35 through 48 are also new. These are commands that will be executed when the Windows Server 2008 machine is set up for the first time i.e. when the command *vagrantup* is ran for the first time.

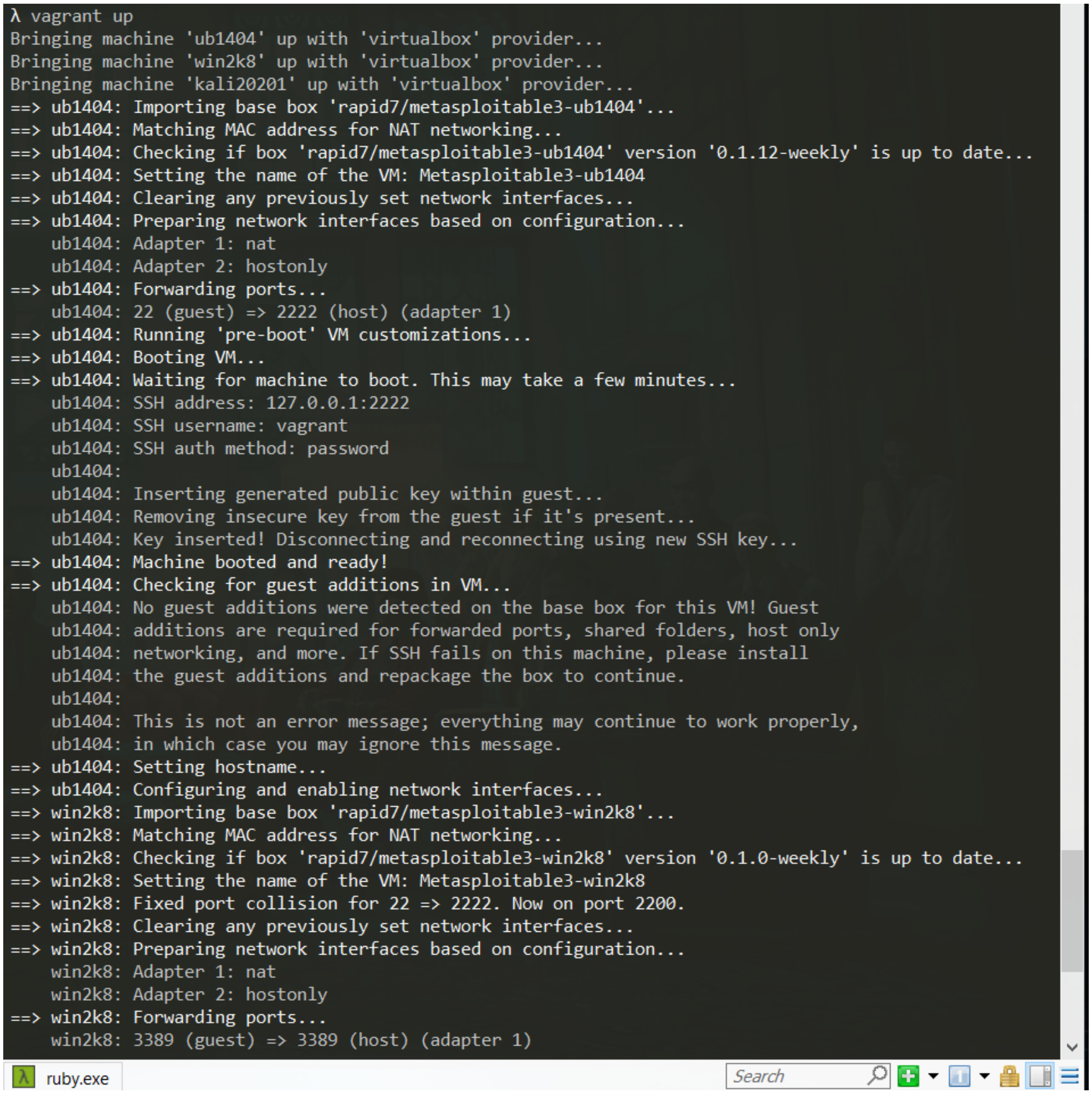
7. line 51 - 64 : Set-up the Kali Linux box. Line 61 through 64 tell the Kali Linux box to update the first time it loads.

8. Line 65 : Ends the configuration of the machines

4.3 Starting the Virtual Machines

To start the virtual machines (our lab) we run vagrant up!

```
$ vagrant up
```



```
λ vagrant up
Bringing machine 'ub1404' up with 'virtualbox' provider...
Bringing machine 'win2k8' up with 'virtualbox' provider...
Bringing machine 'kali20201' up with 'virtualbox' provider...
==> ub1404: Importing base box 'rapid7/metasploitable3-ub1404'...
==> ub1404: Matching MAC address for NAT networking...
==> ub1404: Checking if box 'rapid7/metasploitable3-ub1404' version '0.1.12-weekly' is up to date...
==> ub1404: Setting the name of the VM: Metasploitable3-ub1404
==> ub1404: Clearing any previously set network interfaces...
==> ub1404: Preparing network interfaces based on configuration...
    ub1404: Adapter 1: nat
    ub1404: Adapter 2: hostonly
==> ub1404: Forwarding ports...
    ub1404: 22 (guest) => 2222 (host) (adapter 1)
==> ub1404: Running 'pre-boot' VM customizations...
==> ub1404: Booting VM...
==> ub1404: Waiting for machine to boot. This may take a few minutes...
    ub1404: SSH address: 127.0.0.1:2222
    ub1404: SSH username: vagrant
    ub1404: SSH auth method: password
    ub1404:
    ub1404: Inserting generated public key within guest...
    ub1404: Removing insecure key from the guest if it's present...
    ub1404: Key inserted! Disconnecting and reconnecting using new SSH key...
==> ub1404: Machine booted and ready!
==> ub1404: Checking for guest additions in VM...
    ub1404: No guest additions were detected on the base box for this VM! Guest
    ub1404: additions are required for forwarded ports, shared folders, host only
    ub1404: networking, and more. If SSH fails on this machine, please install
    ub1404: the guest additions and repackage the box to continue.
    ub1404:
    ub1404: This is not an error message; everything may continue to work properly,
    ub1404: in which case you may ignore this message.
==> ub1404: Setting hostname...
==> ub1404: Configuring and enabling network interfaces...
==> win2k8: Importing base box 'rapid7/metasploitable3-win2k8'...
==> win2k8: Matching MAC address for NAT networking...
==> win2k8: Checking if box 'rapid7/metasploitable3-win2k8' version '0.1.0-weekly' is up to date...
==> win2k8: Setting the name of the VM: Metasploitable3-win2k8
==> win2k8: Fixed port collision for 22 => 2222. Now on port 2200.
==> win2k8: Clearing any previously set network interfaces...
==> win2k8: Preparing network interfaces based on configuration...
    win2k8: Adapter 1: nat
    win2k8: Adapter 2: hostonly
==> win2k8: Forwarding ports...
    win2k8: 3389 (guest) => 3389 (host) (adapter 1)
```

When the command completes execution. A Kali Linux box should pop up! **The credentials are username : vagrant and password : vagrant. Those are also the credentials of the other machines.**

You can access the other machines through the Kali Linux now and practise your pentesting!

5 What Next?

Now you have a whole lab to practise your skills on. Let the hacking commence!

I have included some commands to help you manage the Vagrant boxes:

5.1 Useful Vagrant commands to manage the machines.

If you want to ssh into the machines

(a)

```
$ vagrant ssh ub1404 #ssh into the Ubuntu 14.04 virtual machine
$ vagrant ssh win2K8 #ssh into the Windows 2008 Server virtual machine
$ vagrant ssh kali20201 #ssh into the Kali Linux virtual machine
```

If you want to shutdown the machines. This can be in situations where you are using too many resources and don't need all machines to be operating. You can switch off one and continue to hack with two.

(b)

```
$ vagrant halt ub1404 #shutdown the Ubuntu 14.04 virtual machine
$ vagrant halt win2K8 #shutdown the Windows 2008 Server virtual machine
$ vagrant halt kali20201 #shutdown the Kali Linux virtual machine
$ vagrant halt #shutdown all virtual machines
```

If you want to suspend the machines

(c)

```
$ vagrant suspend ub1404 #suspend the Ubuntu 14.04 virtual machine
$ vagrant suspend win2K8 #suspend the Windows 2008 Server virtual machine
$ vagrant suspend kali20201 #suspend the Kali Linux virtual machine
$ vagrant suspend #suspend all virtual machines
```

If you want to reload the machines

(d)

```
$ vagrant reload ub1404 #reload the Ubuntu 14.04 virtual machine
$ vagrant reload win2K8 #reload the Windows 2008 Server virtual machine
$ vagrant reload kali20201 #reload the Kali Linux virtual machine
$ vagrant reload #reload all virtual machines
```

If you want to delete the machines

(e)

```
$ vagrant destroy ub1404 #delete the Ubuntu 14.04 virtual machine
$ vagrant destroy win2K8 #delete the Windows 2008 Server virtual machine
$ vagrant destroy kali20201 #delete the Kali Linux virtual machine
$ vagrant destroy #delete all virtual machines
```

5.2 Useful Resources:

There are a lot of hacking resources online but here are a few useful ones.

- Cyber Security basics - https://cybering.cc/
- Metasploitable3 Vulnerabilities wiki - Wiki
- A YouTube playlist on Metasploitable3 hacking - YouTube Playlist
- Awesome hacking GitHub repositoty - Repo