Using vagrant to create three machines and network between them.

Here is a quick explanation of what Vagrant is and how to use it. The gist is that I'm creating two virtual machines and giving them a private network for them to talk to each other.

https://scotch.io/courses/getting-started-with-vagrant-for-local-development/what-is-vagrant

```
chad@brakebills:~/code/Cybersecurity-work-class/classes/CIS311/week2$ cat Vagrantfile
# -*- mode: ruby -*-
# vi: set ft=ruby :

# Vagrant multi-machine sample setup

Vagrant.configure("2") do |config|
  config.vm.define :alpha do |alpha|
    alpha.vm.box = "hashicorp/precise64"
    alpha.vm.network :private_network, ip: "10.0.0.10"
    alpha.vm.hostname = "alpha"
  end

  config.vm.define :beta do |beta|
    beta.vm.box = "hashicorp/precise64"
    beta.vm.network :private_network, ip: "10.0.0.11"
    beta.vm.hostname = "beta"
  end

  config.vm.define :gamma do |gamma|
    gamma.vm.box = "hashicorp/precise64"
    gamma.vm.network :private_network, ip: "10.0.0.12"
    gamma.vm.hostname = "gamma"
  end
end
```

Start them up and then confirm that they are running. At this point I have three virtual machines running on my laptop. They are called **ALPHA, BETA,** and **GAMMA**.

```
chad@brakebills:~/code/Cybersecurity-work-class/classes/CIS311/week2$ ~/Downloads/vagrant status
Current machine states:

alpha                     running (virtualbox)
beta                      running (virtualbox)
gamma                     running (virtualbox)

This environment represents multiple VMs. The VMs are all listed
above with their current state. For more information about a specific
```

Next I'll connect to them in three separate windows using vagrant's ssh tool.

ALPHA (IP address 10.0.0.10)

```
chad@brakebills:~/code/Cybersecurity-work-class/classes/CIS311/week2$ ~/Downloads/vagrant ssh alpha
Welcome to Ubuntu 12.04 LTS (GNU/Linux 3.2.0-23-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '14.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Welcome to your Vagrant-built virtual machine.
Last login: Wed Feb 19 04:19:05 2020 from 10.0.2.2
vagrant@alpha:~$ hostname -I
10.0.2.15 10.0.0.10
```

BETA (IP address 10.0.0.11)

```
chad@brakebills:~/code/Cybersecurity-work-class/classes/CIS311/week2$ ~/Downloads/vagrant ssh beta
Welcome to Ubuntu 12.04 LTS (GNU/Linux 3.2.0-23-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '14.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Welcome to your Vagrant-built virtual machine.
Last login: Wed Feb 19 04:22:15 2020 from 10.0.2.2
vagrant@beta:~$ hostname -I
10.0.2.15 10.0.0.11
```

GAMMA (IP address 10.0.0.12)

```
chad@brakebills:~/code/Cybersecurity-work-class/classes/CIS311/week2$ ~/Downloads/vagrant ssh gamma
Welcome to Ubuntu 12.04 LTS (GNU/Linux 3.2.0-23-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '14.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Welcome to your Vagrant-built virtual machine.
Last login: Wed Feb 19 06:12:26 2020 from 10.0.2.2
vagrant@gamma:~$ hostname -I
10.0.2.15 10.0.0.12
```

So the next step is to use the tool tcpdump. It allows you to capture all network traffic coming into or out of a machine. (https://www.tecmint.com/12-tcpdump-commands-a-network-sniffer-tool/) So in essence I'm going to listen in on the ping traffic coming to and from each of these VM's.

ALPHA

```
vagrant@alpha:~$ tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 65535 bytes

vagrant@alpha:~$
```

BETA

```
vagrant@beta:~$ sudo tcpdump -nl -i any icmp &
[2] 1805
vagrant@beta:~$ tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 65535 bytes
```

GAMMA

```
vagrant@gamma:~$ sudo tcpdump -nl -i any icmp &
[1] 1829
vagrant@gamma:~$ tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 65535 bytes
```

Now over on **ALPHA**, I'll send a single ping to **BETA.** You can see the request go out to 10.0.0.11 and come back to 10.0.0.10

```
vagrant@alpha:~$ ping -c 1 10.0.0.11
PING 10.0.0.11 (10.0.0.11) 56(84) bytes of data.
05:52:26.158463 IP 10.0.0.10 > 10.0.0.11: ICMP echo request, id 2006, seq 1, length 64
05:52:26.160577 IP 10.0.0.11 > 10.0.0.10: ICMP echo reply, id 2006, seq 1, length 64
```

Over on **BETA** we immediately see that the ping request came in and a reply was sent.

```
05:54:44.543967 IP 10.0.0.10 > 10.0.0.11: ICMP echo request, id 2009, seq 1, length 64
05:54:44.544028 IP 10.0.0.11 > 10.0.0.10: ICMP echo reply, id 2009, seq 1, length 64
```

Over on **GAMMA** we don't see anything.

So at this point I have two machines, I have a way to trigger traffic from one machine to the other, and on the receiving machine I can see that incoming traffic. Now onto sending a ping with a forged return address. For this example, I'm going to use hping3 but there are a lot of tools that do the same thing. (https://tools.kali.org/information-gathering/hping3) So let's set the spoofed return address to **GAMMA**(10.0.0.12) So you see the ping leave but nothing comes back.

On **ALPHA**, we see the ping leave but no response comes back.

```
vagrant@alpha:~$ sudo hping3 -1 -S 10.0.0.11 -a 10.0.0.12 -c 1
HPING 10.0.0.11 (eth1 10.0.0.11): icmp mode set, 28 headers + 0 data bytes
05:56:26.865697 IP 10.0.0.12 > 10.0.0.11: ICMP echo request, id 57863, seq 0, length 8
```

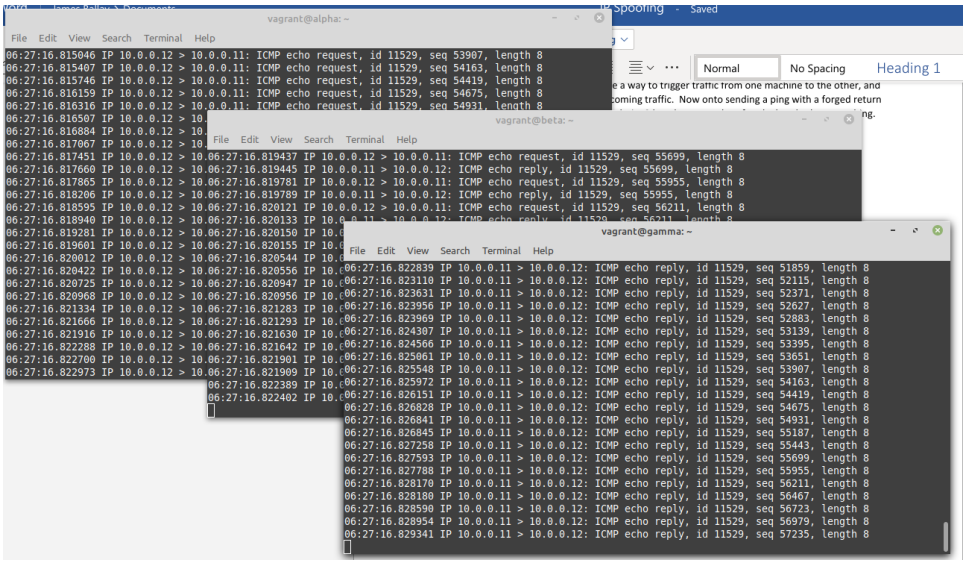
Over on **BETA** we see an incoming ping packet but this time it looks like it is coming from **GAMMA** instead of **ALPHA**. **BETA** takes this packet at face value and sends a response.

```
06:21:31.609352 IP 10.0.0.12 > 10.0.0.11: ICMP echo request, id 64007, seq 0, length 8
06:21:31.609408 IP 10.0.0.11 > 10.0.0.12: ICMP echo reply, id 64007, seq 0, length 8
```

Over on **GAMMA** it sees a ping reply even though it never sent a ping out. So without doing anything it just had to waste resources accepting that packet and processing it.

```
06:21:31.621021 IP 10.0.0.11 > 10.0.0.12: ICMP echo reply, id 64007, seq 0, length 8
```

Now in a simple situation like this, the impact is trivial to **GAMMA**. But let's say instead of 1 attacking machine there were hundreds or thousands.



From the perspective of **GAMMA** it's being flooded with traffic from **BETA** but in reality it's being triggerd on **ALPHA**. And this was a simple example just using ping.