Quiz 2

Student name: Kevin Martin

1(20): what is the firewall, what are the types of the firewall according to filtering directions. What type is the default ufw according to it's filtering direction(without setting up anything)? Can ufw be both direction? How? if you'd like to use ufw to block traffic, what is the first thing to do in VM(assume it is already enabled)? What is the reason?

A firewall is a program that observes the traffic passing through it and determines, based on the contents of the packet, whether or not it should be allowed through (or potentially “modified” where the packet is rerouted).

The types of firewalls according to their filtering directions are Ingress (which filters traffic coming into the network) and Egress (which filters traffic going out of the network.

Ufw is actually disabled by default, but once enabled, the default direction is ingress (denying incoming traffic).

Ufw can be set both directions. The syntax is very simple, just specify the ip address and then “to” or “from. For example, if the host machine has ip address 10.0.0.1 and you wanted to block outgoing web traffic, you would use the following:

*sudo ufw deny **from** 10.0.0.1 to any port 80*

Assuming it was already enabled in a VM, the first step would be to identify the ip address of the machine you want to block. Then, enter a command similar to the above.

2(20): How many types of firewall do we have according to it functions?

There are 3 types of firewalls: packet filter (stateless, single packet observation), stateful (multiple packet observation), and application (such as a web proxy, which looks at the layer above the UDP/TCP protocol).

3(20): Where does firewall run: Kernel or user space?

The firewall runs in the kernel space. There are two ways to implement: the first is to modify the kernel and recompile. The second is to use the Loadable Kernel Module, which allows us to modify the behavior of the kernel.

4(40): What is the module are you using to write firewall in this lab, is it in the user space or kernel space? how does it work? (not in code detail, in names detail, such as what do you call those connections function? where does this module run and how does it achieve it's purpose etc)

The module we are using to write the firewall is called Netfilter. It is written in the user space but runs in the kernel space. It works by creating hooks in the specified places in the kernel. If there is something connected to these hooks, such as our program, the code hooks the program into the kernel. When a packet crosses one of these hooks, if our code is runnning, it will examine the packet and either allow it to pass or drop, depending on the contents and the filtering rules of our program.

In order to write the fire wall we are using the Loadable Kernel Module in conjunction with netfilter, it is being written in the user space, however, it is then made into a kernel object which is then loaded into the kernel space via the terminal, in order to manipulate the packets they have to be "hooked" (netfilter implements these hooks which essentially boil down to