**Project 1 Interview**

**Faulty Firewall**

The firewall that was implemented to block SSH connections to the server is faulty and needs to be revised and debugged. In the ELK stack project, I let only certain machines access others via SSH connection and others via connection through docker containers launched onto Web server machines. If the other machines could SSH into the machines that were not accepting SSH connections, this problem could come at a great cost to the server’s stability and availability. This all depends on the person who is accessing the server of course, but if a hacker were to gain access to the Web server virtual machines it could spell a lot of trouble for the organization they were built for. The main problem that comes to mind when thinking of this kind of vulnerability is a ransomware attack. The hacker could download and store all the files on the system on their own hard drives and require a monetary ransom, or other sort of payment, to get all the data back that they stole.

First off, if one of the machines I implemented on my virtual network was receiving SSH connection when it is not supposed to, I would turn to the NSG (Network security group) I created to see if all the security rules I set to protect the machines are still in place correctly. Making sure the network security rules are all set correctly is a crucial step to managing any network server or database. These rules protect what can go in and out of each machine or server whether it be HTTP, UDP, TCP or SSH connections. Certain rules that are set in place on my machines for the ELK project were allowing only my physical machine’s IP address to access the jumpbox and the specific IP addresses of other machines to access the Web machines or the ELK machine. The specific configuration I would check would be the NSG rules focused on the jumpbox, this machine has access to the web1 and web2 machines. Those VM’s control the DVWA servers and the docker containers which are connected to the ELK VM which has Ansible running on it as well. If the jumpbox is compromised, the whole network is compromised. The first thing I would check on this is the SSH rule I have to let only my IP address through. If that rule is still properly configured, I would then look at the SSH key, generate a new key, and change it just to be safe. After changing the SSH key I would then look at the other NSG rules just to make sure there are no flaws in the configurations and make sure the IP address of the machine is set to private. Since I have an Ubuntu virtual machine on my host machine, I would make sure that machine would not be able to SSH into the virtual network since it has a different IP address. If it is not able to SSH into the jumpbox, all is well. Though, I cannot stop there, I must make sure all the other machines are not accessible by my Ubuntu machine. I would go through the same steps to make sure all the NSG rules of the other machines are configured correctly, then test to see if my Ubuntu machine can SSH to the others.

After all the work done to the security rules and SSH keys, the network should be fortified against foreign SSH attempts. Though the SSH problem has been resolved, the application isn’t fully immune to unauthorized access, much more than a firewall must be put in place to assure it is 100% secure. One way we could monitor the activity on the servers could be Wireshark. This software gives us the ability to see the HTTP, TCP and UPD activity among other things to make sure there is no other suspicious activity is happening on the network. Another software that can be used is installed on the ELK stack itself, Metricbeat can monitor the metrics of the hardware such as CPU, incoming and outgoing data streams and disk usage. With these tools almost all angles can be covered to make sure the servers are safe and secure from attackers.