**Interview Question**

**Domain**: Cloud Security

**Question**: How would you control access to a cloud network?

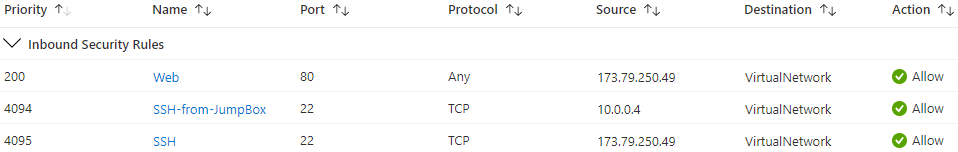
So, the goal is to control the access to our cloud network. In a previous project I deployed a cloud network through Microsoft Azure. I had to configure access controls because anyone with a username and the public IP would be able to access the network if I hadn’t. I created network security groups (NSGs) that initially denied all inbound traffic so we could configure our virtual machines (VMs) in a secure environment. Then key pairs were also generated and configured to the VMs inside the networks. The NSGs were configured with a whitelist of source IPs, ports and protocols allowed to pass through those firewalls. So, by implementing these controls I was able to specify the machines allowed to connect to the cloud network.

The kinds of access controls I implemented were the NSGs configured to the virtual networks (VNet & ELK\_Vnet) I had created. However, there were no NSGs or firewalls specific to each VM due to the strict whitelist rules in place, which dictated access to them through Jump Box. Both NSGs had customized inbound security rules which were needed to allow the machines to communicate.

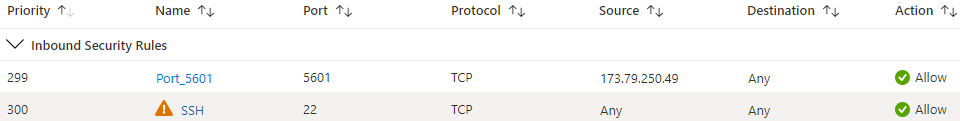
Authentication was achieved by generating a public key on my local machine to connect to Jump Box, then another public key to access the VMs from the Ansible Container, attached to Jump Box. Authorization was achieved by configuring inbound security rules on the NSGs to allow one remote user with a specific public key to access the VMs. Additionally, [webservers] and [elk] groups were created to dictate the host IPs where the playbook was going to run. And access to the VMs were achieved after the external machine was authenticated and authorized. These restrictions were necessary for the project to create a safe and secure networking environment. Furthermore, I audited the NSGs by changing the IPs and attempting to login to the VMs again.

Below are the custom inbound rules set for each NSG in the network:

* + VNet:



* + ELK\_Vnet:



Access to Jump Box worked by first creating it in Azure with the applicable resources and NSG inbound rules. Key pairs were generated on my local machine, and then I configured the SSH public key in Azure to allow my specific IP to connect to it. Access from Jump Box to the web servers worked by creating, attaching, and starting an ansible container. Once inside the container I generated another key pair, and through Azure configured the VMs to only allow a connection from that ansible container.

The advantage of deploying this network in a cloud environment is that it scales very easily. Admittedly, Azure Bastion would have been better than using Jump Box, because it’s an agent-less solution and is supposed to be the ‘replacement’ for Jump Box as a PaaS solution. A VPN wasn’t implemented here because cloud networks can scale easily, and VPNs create more complexity that impede network growth. Some other disadvantages of them are unpredictable performance and unreliable availability. Although the advantages of VPNs are things like added security, their low cost, and low maintenance, it wasn’t necessary to use one in this network because the configured security was enough to do the job, and the network was not connected to any sort of vital database. If the networks were breached, virtually no sensitive data would have been compromised.