There are significant security considerations when deploying virtual networks. The first step in mitigating these potential vulnerabilities is controlling the “who” and “how” of accessing the network.

In Project 1, a cloud network was deployed using Microsoft Azure in which access controls were configured and updated throughout the design process. These access controls included creating Network Security Groups and being strategic in developing the network architecture. Taking these steps ensured that only designated machines have access to specific resources within the virtual network, resulting in a more secure, hardened network.

As previously stated, Network Security Groups were created around the VNET and the ELK server subnet. These NSG’s contained rules to control the protocol of the network traffic as well as allowable ports to connect to each VM. An inbound rule was configured to block all traffic to the subnet upon the initial creation of the VNET before any VM’s were added to the system. This barrier to the internet was necessary in order to create a secure environment in which to begin designing the network architecture. Without this rule, the network would be vulnerable to outside actors being able to access the machines directly through the internet. During the initial phase of this project, a class member failed to implement this rule and her network was compromised by an outside threat actor, illustrating the importance of this step in securing your network.

After the Jump-Box and other 3 VMs were created, a rule was added to allow SSH from only my public IP address ( local machine ) to the Jump-Box, and from the Jump-Box to the other VMs as this would be the administrator for the other VMs. It is important to note that the method of authentication for SSH onto the network is using SSH keys as they are more secure than a simple password that could be brute forced. After installing an Ansible container on the Jump-Box, this rule was updated to only allow SSH from the Ansible container within the Jump-box to the 3 VMs. Once a load-balancer was established, the NSG was updated to allow my local machine to connect Port 80 ( HTTP ) on the VNET in order to access the DVWA server.

The network architecture was configured in such a way as to restrict the number of ways the VMs inside of the subnet could be accessed. Specifically, each machine could only be accessed via SSH directly from the Ansible container on the Jump-Box VM. This method of “fanning in” is effective because it reduces the attack surface of the subnet down to only one gateway router – the Jump-Box.

This solution easily scales thanks to a combination of one administrator ( Jump-Box ) and the use of containers. Using these containers give the ability to create Ansible playbooks that can automate the process of building out new containers on the new VMs and configuring them to arbitrary specifications.

For the purposes of this project, use of a Jump-Box was a sufficient security measure to manage access. If deemed necessary, another secure way of connecting would be via a VPN. This creates an encrypted “tunnel” or connection from a local network to a remote network, giving the user access as if he/she were locally connected. This is beneficial because it allows for a user to access resources from a remote private network through secure channels. Some disadvantages of using a VPN is that they are not designed for continuous use so excessive loads can negatively impact performance. It is also very costly to scale and maintain performance at the same time. VPN’s are most appropriate for enterprises requiring a group of employees to work remotely and have the resources to accommodate a robust performing but more costly VPN.