**Question 1: Faulty Firewall**

*Suppose you have a firewall that's supposed to block SSH connections, but instead lets them through. How would you debug it?*

The issue: A firewall is letting SSH connections through instead of blocking them.

Example Scenario: In Project 1, no external SSH traffic was allowed to the backend DVWA web VMs or the ELK server VM. The Jumpbox accepted SSH traffic from my IP only and the others accepted SSH connections from an ansible container within the jumpbox. If an SSH connection is attempted to the machines it fails.

Solution requirements: If one of the VMs was accepting connections, I would assume the error was in my Network Security group configurations. I would check to make sure only the IP address of my jumpbox was allowed to connect to the machine's TCP port 22. I would also check to make sure the default block all incoming traffic is set to a lower priority number than any rule that may allow all SSH traffic. In order to test new configurations I would attempt SSH connections from IP addresses other than the one I specifically want to allow access with.

Solution details:

The specific Azure panes I opened were

- Network Security Groups (Selected security group to investigate rules for)
- Inbound Security Rules (Investigated various rules)

Configurations and controls checked (And what was looked for):

- Source (Set to "IP Addresses", any if denying all incoming)
- Source IP Addresses (Set to Jumpbox internal IP, all if denying all)
- Destination (Set to "Virtual Network", set to any if denying all)
- Destination Port Ranges (Set to 22)
- Action (Set to allow for access, deny if configuring to deny all incoming)
- Priority (Set with a number low enough to take effect over any rule that may allow for more access)

How to attempt to connect to VMs for testing:

1. Turn VMs on
2. Do NOT ssh into jumpbox but do open powershell
3. Attempt to ssh into web VM from command line (ssh azadmin@<web server external IP>)

4. Wait for error message “ssh: connect to host 137.135.103.144 port 22: Connection timed out”

Solution Advantages/Disadvantages:

- Advantage: The network is now set to only allow SSH access from a specific static IP, making it more hardened against attacks because it’s not accessible via public IP address.
- Disadvantage: This solution assumes the host computer connecting to the Jumpbox and ansible containers has not been breached, a breach on this level would compromise the security of this solution as it only accounts for outside threats.
- I would supplement this solution with an IDS set to alert to failed ssh authentication attempts in order to collect information on suspicious auth attempts

**Question 1: Cloud Access Control**

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

The Problem: How do you make sure the only people in your cloud network are people you want there?

Example Scenario: In Project 1 we deployed a cloud network instead of an on site physical network. Access controls configured include the Jumpbox and ansible machines with SSH Keys instead of passwords and Network Security Group settings (As partially outlined in response 1). Limiting network access to the jumpbox only and limiting jumpbox access to my IP keeps the number of individual machines able to access the network low. This is a security benefit because every additional point of connection is a potential point of access for an attacker. The Network Security Group was configured to restrict SSH access from the internet to the machines on the internal network. This was necessary to keep unauthorized people from connecting to the backend VMs via the internet while still keeping Port 22 exposed to my specific IP address.

Solution Details:

NSG rules

- I set a rule to allow SSH connections from the ansible container in my jumpbox to the ELK server and deny all other connection attempts
- I set a rule to allow HTTP connections via port 5601 from my external IP address
- I make sure the default deny all rule is in place but has a higher number than the rules allowing specific sorts of connections

Jump Box Access + Jump Box to web servers:

- SSH from command line to jump box's external IP (ssh azadmin@<jump box ip>)
- Start ansible container (sudo docker start <container name>)
- Attach to container (sodo docker attach <container name>)
- SSH from container to web servers (ssh azadmin@<server IP>)

Advantages/Disadvantages:

- Advantages: Easy to add machines to NSG and have all access controls apply to any deployment under that group, jump point as single point of access provides security benefits
- Disadvantages: Complicated