#### **Domain: Network Security**

**Problem 1 Statement**

Problem detected on port 22, reporting all SSH connections coming through on Firewall with security rules to block SSH connections.

We created a security rule to allow SSH connections only from the host IP address, and to connect with the Jump-Box-Provisioner virtual machine for management. Only Jump-Box-Provisioner VM has the capability to enable SSH connections. In the case that a user tries to connect to Web-1 or Web-2 VM using port 22, connections will be rejected because we configured the access with the SSH key from the ansible container created inside the Jump-Box-Provisioner with the purpose of susceptibility to brute force attacks. Additionally, we can add that in the case in which an attacker steals the key remotely, they will be able to connect to the VM if they also successfully compromise the development machine. Explaining this with more detail we will connect the workstation machine SSH/HTTP to the internet. It will be filtered by the Red Team Network Security Group firewall or ELK Security Group firewall and each of those firewalls is going to allow the SSH traffic to the Jump Box VM or ELK VM. It is important to note that DVWA VM and containers have the capability to allow connections with ELK.

In the case that Web-1 or Web-2 allow SSH connections, we would take the assumption that one inbound security rule on the network security group was changed or we did a misconfiguration. To be able to detect any failures, we recommend double-checking the Network Security Groups all the SSH inbound security rules and evaluate if there are any changes on the source IP addresses. To test the effectiveness of the configuration we recommend the SSH connection from an external IP address and evaluate if the VM’s are allowing SSH connections.

The previous scenario and solutions provided does not guarantee that the network is comprehensively immune due to the fact that our development machine can be always compromised, and we can evaluate any suspicious authentication of the /var/auth.log file.

**Problem 2: Statement Unsecured Web Server**

Server running on port 80 detected, compliance guidelines require encryption in motion.

To be able to install our DVWA container, first we created an installation YAML file specifying the launching and downloading of the ansible docker container using port 80 as a published port as well as host port 80 using the hypertext transfer protocol in a non encrypted way. We did the installation of an azure load balancer that distributes incoming traffic among healthy virtual machines and we added a health probe using TCP and port 80 to make sure all the Virtual machines were able to receive traffic, after that we exposed the Red Team Network Security Group to port 80. To do this we created a load balancing rule from the Load Balancer to the Red Team Virtual Network but we specified the internal IP addresses of Web-1 and Web-2 VMs. Doing these changes, we were able to reach the DVWA app over the internet using the following template [http://[load\_balancer\_IPadress]/setup.ph](about:blank)p. In this case we wanted to do this configuration to be able to set up an attacking training machines environment and avoid any connection interruptions. In the real world the best option to do any configurations should be using port 443 HTTPS because we would like to add an encrypted filter to the data transference.

Running HTTP on port 80 can represent a potential problem due the exposure to a non safe network interaction, and also because our network can be susceptible to attacks like SQL injection to be able to have a safe HTTP traffic we recommend to change the ports on the load balancing rule to 443, and this could help because we are using the default safest way of data encapsulation.

#### **Domain: Cloud Security**

**Problem 3 Statement Cloud Access Control to a Cloud Network**

We deployed a cloud network that we configured access to a handful of IP addresses adding rules to the NSG, VPN and Load balancer rules, this controls are necessary because we would like to restrict the access specially we are talking about sensitive data of any organization, so as a part of this project deployment we created an Azure Red Team Resource Group containing the Azure Red Team Virtual Network and Subnetwork that also groups two environments: 1 Red Team (RT firewall, RT load balancer, Jump-Box-Provisioner, and Web servers VM’s), 2 ELK (ELK firewall and ELK VM).

Aboarding the Red team security group we restricted the access through the Jum-Box-Provisioner VM, we decide to add the load balancer in order to efficiently allocate resources to the Virtual Environment created, we did this configurations because we wanted to ensure the installation and deployment of the Ansible docker container, also we decide to configure all the deployment under the Red Team Security Group to be able to ensure network connections and be able to complete cloud security training without any connection interruptions and evaluate data repositories using the ELK VM and containers.

As a result we created two Network Security Groups with a different set of security rules, in both of them we restricted external SSH connections to any of the VM’s created:

1. Red Team Security Group with the following inbound security rules: Port\_8080 that allows port 80 traffic from the internet to the VM from my public IP and has the VN as a destination, Jump-Box-Access that allows SSH access from the Jumpbox provisioner VM and has the VN as a destination too and finally SSH inbound rule that allows SSH connections from my host IP address to the Jump-Box provisioner VM, all the above rules use TCP as a Protocol.
2. ELK VM Security Group with the default-allow-ssh that as the name describes allows SSH connection with the ELK VM and VN using TCP protocol.

We created Red Team and ELK Virtual networks with the purpose to communicate Virtual Machines with each other, to be able to access to the Jump-Box since the host machine terminal we ran the command ssh with the Jumpboxuser followed by the public IP address and once the Jump-Box user prompted on the terminal we accessed to each if the Web VMs using the ssh command, user and private IP we could confirm that we were having the proper connection because users for the Web VMs were prompted to corroborate that we have access to internet we run ping command to a random website and the output revealed the packets transferred.

Some of the advantages of a VPN are that VPNs provide secure internet connections via encryption, also all the data traveling through the VPN is encrypted ensuring relatively secure remote connections, one of the main disadvantages can be that in case of receive an attack if the configuration of the network environment is not redundant and the attacker obtain privileges as a root he can make a considerable damage on the information and affect the organization, also corporate VPN can be susceptible to phishing and social engineering attacks. Corporate VPNs with the right policies can be beneficial in large organizations with a controlled network access.

**Containers**

Containers are lightweight software components that bundle the application, its dependencies, and its configuration in a single image, running in isolated user environments on a traditional operating system on a traditional server or in a virtualized environment.

We deployed Ansible docker containers to create a virtual environment and attacker machines without spend a lot of resources and also creating this cloud computing environment it is not necessary to aboard the costs of a physical environment, other advantage is that in terms of security we are isolating each environment and having all the installation YAML files and playbooks we can deploy these environments easily and without complications, also we can have a faster resource provisioning because containers don’t use hypervisor and have to share resources like virtual machines.

We decided to use ansible to deploy thirsty\_rubin docker container and install the DVWA container, with the ELK VM we deployed ELK server and it was running properly and we installed beats data collection tools, specifically filebeat and metricbeat, filebeat was installed in all the Web servers machines to monitor and collect the log files from Apache, Microsoft Azure tools, Nginx web server and MySQL databases.

Accessing to Kibana we were able to evaluate the data graphs on the data panes for Unique Visitors, Source country, Visitors by OS, Responses codes Over time (200, 404, and 503), File type scatter Plot, Heatmap filtering data of the most active hours by country and the data transferred between machines.

