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

**Question 1: Faulty Firewall**

Suppose I have a firewall that is supposed to block SSH connections but instead lets them through. This is how I would debug it.

* I would first block all inbound traffic by default.
* After that, I would enable specific traffic to known services.
* Then, I would specify which IP addresses to access the HTTP servers that would be needed to be enabled. Examples of enabled common Server Management Ports such as SSH: Port 22, RDP: Port 3389; and Data Ports such as SQL: Server Port 1433, Oracle: Port 1521, and MySQL: Port 2206 would be ok when enabled with the proper rules setup.
* Next, I would specify the destination IP addresses
* Lastly, I would specify the destination port that would need to be enabled.

1. A Concrete Example Scenario

* In Project 1, we did allow SSH traffic to all of the VMs on the network.
* VMs Jump-Box VM, Web-1, Web-2, and ELKVM did accept SSH connections.
* If I am not able to connect to a VM that does not accept SSH connections. I will get a “Connection refused” error. This could mean that either the SSH server package is not set up properly or that the service isn’t currently running.

1. Explain the Solution Requirements

* If one of the Project 1 VMs accepted SSH connections, I would try to ping it first to see if it is running and listening to Port 22.
* I would check the general default configuration to Port 22. Make sure that the firewall rules are set correctly, and that it is not listening for Port 22.
* The last action I would take is to test that the new configurations. I make sure they are effective by pinging it and then trying to SSH into it.

1. Explain the Solution Details

* In the Azure UI, I would specifically open the Network Security Group pane and open the Network that needs to be looked at to investigate the problem.
* The specific configurations and controls I would check would be in the Inbound Security Rules.
* I would look at all default rules, specifically for the rules that are set for SSH into Port 22.
* To test my fix I would attempt to connect to my VMs by first trying to ping them and then trying to SSH into them to see if my fix is effective.

1. Identify Advantages/Disadvantages of the Solution

* My solution would not guarantee that the Project 1 network is now "immune" to all unauthorized access because there is no such thing as a 100% guarantee for immunity? Hackers are always trying to break the system. Updates to software and patches could also mess with configurations and leave a port disabled or open. It’s just best practice to keep watching and logging any changes.
* Audit logs and monitoring controls that I can add to ensure that I can identify any suspicious authentication attempts would be applications like: “SolarWinds Network Performance Monitor, Datadog Networking Monitoring, Logz.io Pro, Splunk, Kibana, and PRTG Network Monitor, etc.” The advantage of using these types of monitoring systems is that most are well-known and implement most of the same type of monitoring controls and logs. The disadvantage is that how they monitor and log is not all the same. Patches and vulnerabilities have been known to happen and can cause major damage and data compromise to the infrastructure.

**Question 2: Unsecured Web Server**

Suppose I find a server running HTTP on port 80, despite compliance guidelines requiring encryption in motion. What would I do? ​​

* I would close HTTP on Port 80 from the webserver.
* Next, I would make sure that the web server’s HTTP on Port 80 is not set to default and change it to HTTPS on Port 443. Also, generate an SSL certification to make sure the website is valid.
* Then, I would set the rules to make sure that HTTP on Port 80 is open to the proper inbound web traffic addresses making sure that the redirected traffic is to the correct recent version of the site. The secured HTTPS on Port 443 would be for certain personal.
* Lastly, I would test HTTP on Port 80 by using an application such as Autodesk Vault Server.

1. Concrete Example Scenario

* In Project 1, I did have a webserver running HTTP on port 80? Under the Red-Team-Security-Group for the sole purpose to allow web testing by DVWA.
* In a real deployment, I would specifically configure the RedTeamSG webserver to be set as HTTPS on Port 443. I would then set the HTTP to allow inbound web traffic to only specific web addresses. If I were to completely close the port then I would not be able to pentest with DVWA and Kibana. A closed port would just give my client no access to the internet.

1. Solution Requirements

* Running HTTP on port 80 without HTTPS on Port 443 can be a potential problem if rules and redirection to the incoming web traffic are not configured correctly. You have to watch out for SQL injections, cross-site request forgeries, scripting, and buffer overruns.
* I would reconfigure a server to serve HTTP traffic safely by redirecting and managing the addresses to the traffic to the most recent and secure HTTPS.
* This solution will fix the problem by redirecting the web traffic addresses from the server’s HTTP on Port 80 to HTTPS on Port 443.

1. Solution Details

* The tools and technologies I would use to implement this solution in Project 1. would be: Ansible playbook, DVWA, and Kibana.
* I would specifically use these tools to harden my deployment because they all have great features and real-world uses. For instance, the Ansible playbook would help to set up, configure, patch, and deploy all changes. I would then use DVWA to pentest with. Lastly, Kibana’s use for analytics, application monitoring.

1. Advantages and Disadvantages of the Solution

* The advantage of this solution will not break clients that used to communicate with the server over HTTP on Port 80 only if configured, set up, and deployed properly. All incoming web traffic will be set to be redirected from HTTP on Port 80 to HTTPS on Port 443. The disadvantage would be that when a new update to the software or a patch can be reset to default if not deployed and configured properly. This can leave Port 80 open and Port 443 closed.
* I have to do my due diligence in maintaining the solution in the long term. There will always be new software upgrades and patches that may have to be configured properly before deployment. It is not as simple as a "set it and forget it!”