Technical Challenge Security Specialist

# Part 1: Cybersecurity Scenario

## 1. Threat Intelligence Report

Types of attack that exploit web application vulnerabilities could include and result in unauthorized access, data theft or compromise. Typical exploits are Brute Force Attack, cross-site scripting (XSS), cross-site request forgery (CSRF), XML External Entity (XXE), Path Traversal and SQL injection, etc.

Exploited vulnerability may provide access to the network and facilitate malicious code ability to compromise the network or system that often result in data britches or denial of service. For example, in the recent past unpatched and unsupported Windows Operating system computers have allowed malicious ransomware code to exploit Windows privilege escalation flaw and render company hostage to data access for ransom.

Preventive measures to avoid similar incidents in the future include regular patching and vulnerability assessment, the use of layers of difference, to include next generation firewall with support or in combination of Web Application Firewall (WAF)

Microsoft Azzure Services can be affected by vulnerability exploitation types of attacks include insecure Web App application on Azure App Services (Web App), Windows and Linux Azure Virtual Machines, Azure Container instances, Pods on Azure Kubernetes Services, Logic apps, Functional Apps, Azure API Gateway.

## 2. Incident Response Plan

Incident response plan to address the breach will include

Preparation, detection and Root Cause Analysis (RCA)/Identification and classification of incident, containment, recovery/ eradication/mitigation, and post-incident activity like lesson leaned and. documentation.

Steps for containment, eradication, and recovery will include patching of exploited and any vulnerabilities that has been identified and can be patched, ensure layered defense is set and configured properly, enable or ensure Web Application Firewall is set to prevention mode, restore system state from a known uncomposed state depending on the nature of the compromise.

Using Microsoft Azure cloud provider, we can utilize: -

* Microsoft Defender for Cloud, which is a cloud-native application protection platform (CNAPP) that is made up of security measures and practices that are designed to protect cloud-based applications and workloads from various cyber threats and vulnerabilities.
* Microsoft Sentinel is a cloud native security information and event management (SIEM) and security orchestration, automation, and response (SOAR) solution that runs in the Azure cloud.
* Azure Front Door provides a fast, reliable, and secure modern cloud content delivery network, integrated with intelligent threat protection, Azure Firewall Policies and Web Application Firewall.
* Azure Application Gateway is a web traffic (OSI layer 7) load balancer that enables you to manage traffic to your web applications.
* Azure DDoS Protection to prevent Distributed denial of service (DDoS) attacks.
* Azure Monitor and Alert
* Azure Update Manager is a service that helps manage updates for all your machines, including those running on Windows and Linux, across Azure, on premises, and on other cloud platforms.
* Azure Policy is a service in Azure, that you use to create, assign and manage policy definitions in your Azure environment.
* Azure Key Vault to encrypt keys and small secrets like passwords that use keys stored

Azure Backup provides independent and isolated backups to guard against accidental destruction of original data.

## 3. Network Security Measures

Using Microsoft Azure cloud provider, I would recommend network security measures enhancement that make effective use of the following:

* All Web applications should be configured privately and utilize private endpoint connection as backend to premium Azure Front Door Profiles thar are coupled with Azure Firewall Policy with WAP enable and set to prevention mode.
* Azure Firewall will be used to ensure all communication from the private Web Application are inspected and whitelisted to known callers, API and Web Services
* Deploy Azure DDoS Protection to prevent Distributed denial of service (DDoS) attacks on Virtual Networks
* Configure Microsoft Defender for Cloud for all resources and use its recommendation to improve security pasture.
* Deploy Hub and Spoke network with user defined routes to enable monitoring of traffic flow from the application virtual networks.

# Part 2: Container Security Implementation

Tasks:

## 1. Docker Security Best Practices

List and explain five Docker security best practices.

* Use Secure Container Registries from rusted registries like Docker Trusted Registry
* Avoid Running a Docker container with root permissions
* Scan images on private container registry for vulnerability and remediate where possible to improve overall security pasture.
* limiting its memory and CPU resource consumption to prevent single container draining system resources and result in poor performance of overall application.
* Use Docker Networks and APIs for Security for containers communication with each other for efficient communication.

Implement one of these practices in a [Dockerfile](https://github.com/almw/WebApplicationDemoSecurityScan/blob/main/Dockerfile) and provide the code.

## 2. Kubernetes Security Configuration

Describe three Kubernetes security features.

* Workload protection enforces Pod security standards to ensure that Pods and their containers are isolated appropriately.
* Kubernetes API Control plane access protection.
* Secret API provides basic protection for configuration values that require confidentiality.

Kubernetes YAML configuration that includes securityContext settings file “[demo-security-context.yaml](https://github.com/almw/WebApplicationDemoSecurityScan/blob/main/demo-security-context.yaml)”.

## 3. IaaS Security Measures

Infrastructure as a Service (IaaS) like any other cloud service has Shared Responsibility Model for Security. Cloud Service Provider is responsible for setup, secure and maintain the underlying infrastructure physical security, hardware, software like hypervisor, and consumer is responsible to secure part of system and system and application build on that owner/customer and the vendor share responsibilities. The customer is responsible for own data, and security of the deployed system.

# Part 3: CI/CD Pipeline Setup

Tasks:

## 1. Configuration Management with Terraform

Terraform deployment of a web server on a virtual machine.

## 2. CI/CD Pipeline Configuration

Jenkins pipeline configuration ([Jenkinsfile](https://github.com/almw/WebApplicationDemoSecurityScan/blob/main/Jenkinsfile)) that includes stages for building, testing, security scanning and deploying a sample application to Azure.

Appendix A.

Sample Project Code GitHub Repository <https://github.com/almw/WebApplicationDemoSecurityScan>