Design for Security Project Overview

1. Based on the architecture diagram, and the steps you have taken so far to upload data and access the application web service, identify at least 2 obvious poor practices as it relates to security. Include justification.

Poor practice 1

The security group for the trusted public subnet has no ingress rules, which allows unrestricted access to the web service instance.

Poor practice 2

The data is not secured. Files that get placed into the secret recipe vault are not encrypted with a key when uploaded to the bucket, which if downloaded by anyone can also be read.

2. Research and analyze which of the vulnerabilities appear to be related to the code that was deployed for the environment in this project.

Subnets are assigned a public IP address - Reassign to a private IP address

EC2 EBS volumes are not encrypted - Encrypt EBS volumes with a KMS key

Load balancer does not have HTTP to HTTPS redirection configured - Reconfigure to redirect from HTTP to HTTPS

Load balancer does not have deletion protection enabled - Enable load balancer with deletion protection

S3 buckets do not have lifecycle policy configured - Enable a lifecycle policy

S3 buckets do not have polices that require requests to use SSL - Enable a policy that requests to use SSL

EC2 instances are associated with a public IP - Disassociate EC2 instances from the public IP

3. GuardDuty findings

Describe GuardDuty findings that were detected related to the brute force attack

The findings that were detected were 2 instances of unauthorized access to the EC2 instance by SSH bruteforce.

Research the AWS Guard Duty documentation page and explain how GuardDuty may have detected this attack - i.e. what was its source of information.

GuardDuty was able to detect this attack through monitoring of the vpc flow logs on port 22.

4. Simulated Attack

Identify 2-3 changes that can be made to our environment to prevent an ssh brute force attack from the internet.

Update the security group of the EC2 instance to exclude access by SSH to the web application.

Implement AWS WAF to protect the web application.

Neither instance should have had access to the secret recipes bucket, in the event that instance API credentials were compromised how could we have prevented access to sensitive data.

Update bucket policy to restrict access to specific users.

5. Identify tools that will allow you to do the following:

Scan infrastructure as code templates

Regula

Example vulnerability

An IAM policy that is non-restrictive or allows access to all resources and actions.

#Scan AMI's or containers for OS vulnerabilities

AWS Inspector

Example vulnerability

Allowing SSH password login

#Scan an AWS environment for cloud configuration vulnerabilities

AWS GuardDuty

Example vulnerability

API calls from IP addresses that are on threat lists

DevOpsPipeline