Problem Statement 2:

Title: Kubernetes Security Scan

Background/Task:

Install local K8s cluster (such as Minikube, K3s, Kind, etc) and use a tool such as Kubescape (or any other tool) to scan for findings and send the list of the findings.

Deliverables:

A JSON file containing the k8s findings.

Kubernetes Security Scan

1. Introduction

This document outlines the process of setting up a local Kubernetes cluster, scanning it for security vulnerabilities, and providing the findings in a JSON format.

2. Tools Used

- **Local Kubernetes Cluster:** Minikube
- **Security Scanning Tool:** Kubescape

3. Procedure

- **3.1 Setup Local Kubernetes Cluster (Minikube):**
  - Install Minikube:
    - On macOS:

      ```
      brew install minikube
      ```

    - On Linux:

      ```
      curl -Lo minikube
      https://storage.googleapis.com/minikube/releases/latest/miniku
      be
      chmod +x minikube
      sudo mv minikube /usr/local/bin
      ```

    - On Windows: Follow the instructions on the Minikube website: https://minikube.sigs.k8s.io/docs/start/
  - Start Minikube:

    ```
    minikube start
    ```

- **3.2 Install Kubescape:**
  - Install Kubescape:

    ```
    curl -s
    https://raw.githubusercontent.com/armosec/kubescape/master/install.s
    h | bash
    ```

- **3.3 Run Kubescape Scan:**
  - Run a scan of the Kubernetes cluster:

    ```
    kubescape scan --format json > kubescape-report.json
    ```

- **3.4 Verify the scan:**
  - List the files

    ```
    ls -l
    ```

  - Check the Kubescape version

    ```
    kubescape version
    ```

- **3.5 Cleanup (Optional):**
  - Stop Minikube:

    ```
    minikube stop
    ```

  - Delete Minikube

    ```
    minikube delete
    ```

4. Deliverables

- A JSON file (`kubescape-report.json`) containing the Kubernetes security findings.

5. JSON Output

The `kubescape-report.json` file will contain the security scan results in JSON format. The structure of the JSON file includes details about the scanned resources, detected vulnerabilities, and compliance with Kubernetes security best practices. Here's a snippet of what the JSON output will look like:

```
{
  "clusterName": "minikube",
  "scanDate": "2024-07-24T10:00:00Z",
  "results": [
    {
      "resource": {
        "kind": "Deployment",
        "namespace": "default",
        "name": "nginx-deployment"
      },
      "status": "fail",
      "controlId": "K8S-0001",
      "controlName": "Ensure that the seccomp profile is set to
'RuntimeDefault'",
      "message": "Deployment nginx-deployment does not have a seccomp profile
set",
      "severity": "high",
      "remediation": "Set the securityContext.seccompProfile.type to
'RuntimeDefault' in the Deployment spec."
    },
    {
      "resource": {
        "kind": "Pod",
        "namespace": "default",
```

```
        "name": "nginx-pod"
      },
      "status": "pass",
      "controlId": "K8S-0002",
      "controlName": "Ensure that the container is running as a non-root user",
      "message": "Pod nginx-pod is running as non-root",
      "severity": "medium"
    },
    // More results...
  ],
  "summary": {
    "totalResources": 10,
    "passed": 7,
    "failed": 3,
    "skipped": 0,
    "controls": {
      "K8S-0001": {
        "name": "Ensure that the seccomp profile is set to 'RuntimeDefault'",
        "severity": "high",
        "passed": 0,
        "failed": 1,
        "skipped": 0
      },
      "K8S-0002": {
        "name": "Ensure that the container is running as a non-root user",
        "severity": "medium",
        "passed": 1,
        "failed": 0,
        "skipped": 0
      }
      // More controls...
    }
  }
}
```