Trivy - scan k8s image

Create trivy-k8s-scan.sh

```
#!/bin/bash
echo $imageName #getting Image name from env variable
docker run --rm -v $WORKSPACE:/root/.cache/ aquasec/trivy:0.17.2 -q image --exit-code 0 --severity LOW,MEDIUM,HIGH --light
$imageName
docker run --rm -v $WORKSPACE:/root/.cache/ aquasec/trivy:0.17.2 -q image --exit-code 1 --severity CRITICAL --light
$imageName
  # Trivy scan result processing
  exit_code=$?
  echo "Exit Code : $exit_code"
  # Check scan results
  if [[ ${exit_code} == 1 ]]; then
    echo "Image scanning failed. Vulnerabilities found"
    exit 1;
  else
    echo "Image scanning passed. No vulnerabilities found"
  fi;
```

Explain code:

docker run --rm -v $WORKSPACE:/root/.cache/ aquasec/trivy:0.17.2 -q image --exit-code 0 --severity LOW,MEDIUM,HIGH --light $imageName

This line runs the Trivy command in a Docker container. It uses `-v` option to mount the `$WORKSPACE` directory to the `/root/.cache/` directory inside the container for caching Trivy's database. The container is specified using the `aquasec/trivy:0.17.2` image with the `0.17.2` version. The Trivy scan is performed using the `image` option, and the `--exit-code` option is set to `0` to ensure that script does not exit if only low-severity vulnerabilities are found. The `--severity` option is set to `LOW,MEDIUM,HIGH` to include those severities in the report. The `--light` option indicates that the scan should use the lightweight database for faster scanning. `$imageName` variable is passed as a parameter to the Trivy command to scan the Docker image.

docker run --rm -v $WORKSPACE:/root/.cache/ aquasec/trivy:0.17.2 -q image --exit-code 1 --severity CRITICAL --light $imageName

Similar to the previous line, it runs the Trivy command in a Docker container to scan the Docker image, but it has a different severity criteria. This time, it scans for `CRITICAL` vulnerabilities, and sets the `--exit-code` option to `1`.

exit_code=$?
This line captures the exit code of the last executed command, which is the Trivy command. It stores it in the `$exit_code` variable.

echo "Exit Code : $exit_code"
This line prints the exit code of the Trivy command to the console, for debugging purposes.

if [[${exit_code} == 1]]; then
This line starts an if-else statement, where the exit code of the Trivy command is checked. If it is equal to `1`, it will execute the following lines.

```
  echo "Image scanning failed. Vulnerabilities found"
  exit 1;
```

These lines print a message that vulnerabilities have been found in the Docker image, and then exits the script with an exit code of `1`.

else

```
    echo "Image scanning passed. No vulnerabilities found"
  fi;
```

If the exit code of the Trivy command is not `1`, this block of code executes. It prints a success message indicating that there were no vulnerabilities found in the Docker image.

Add trivy scan into pipeline

```
stage('Vulnerability Scan - Kubernetes') {
    steps {
      parallel(
        "OPA Scan": {
          sh 'docker run --rm -v $(pwd):/project openpolicyagent/conftest test --policy opa-k8s-security.rego
k8s_deployment_service.yaml'
        },
        "Kubesec Scan": {
          sh "bash kubesec-scan.sh"
        },
        "Trivy Scan": {
          sh "bash trivy-k8s-scan.sh"
        }
      )
    }
  }
```

After build, we will fail in trivy scan with result like below.

```
Java (jar)
==========
Total: 5 (CRITICAL: 5)

┌──────────────────────────────────────────────┬────────────────┬──────────┬───────────────────┬──────────────────────┬──────────────────────────────────────────────────────────┐
│                   Library                    │ Vulnerability  │ Severity │ Installed Version │    Fixed Version     │                          Title                           │
├──────────────────────────────────────────────┼────────────────┼──────────┼───────────────────┼──────────────────────┼──────────────────────────────────────────────────────────┤
│ org.springframework.boot:spring-boot-starter-web (app.jar) │ CVE-2022-22965 │ CRITICAL │ 2.3.12.RELEASE    │ 2.6.6, 2.5.12        │ spring-framework: RCE via Data Binding on JDK 9+         │
│                                              │                │          │                   │                      │ https://avd.aquasec.com/nvd/cve-2022-22965               │
├──────────────────────────────────────────────┤                │          │                   │                      │                                                          │
│ org.springframework:spring-beans (app.jar)   │                │          │ 5.2.15.RELEASE    │ 5.2.20.RELEASE, 5.3.18 │                                                          │
├──────────────────────────────────────────────┼────────────────┤          │                   │                      │                                                          │
│ org.springframework:spring-web (app.jar)     │ CVE-2016-1000027 │        │                   │ 6.0.0                │ spring: HttpInvokerServiceExporter readRemoteInvocation  │
│                                              │                │          │                   │                      │ method untrusted java deserialization                    │
│                                              │                │          │                   │                      │ https://avd.aquasec.com/nvd/cve-2016-1000027             │
├──────────────────────────────────────────────┼────────────────┤          │                   │                      │                                                          │
│ org.springframework:spring-webmvc (app.jar)  │ CVE-2022-22965 │          │                   │ 5.2.20, 5.3.18       │ spring-framework: RCE via Data Binding on JDK 9+         │
│                                              │                │          │                   │                      │ https://avd.aquasec.com/nvd/cve-2022-22965               │
├──────────────────────────────────────────────┼────────────────┤          ├───────────────────┼──────────────────────┼──────────────────────────────────────────────────────────┤
│ org.yaml:snakeyaml (app.jar)                 │ CVE-2022-1471  │          │ 1.26              │ 2.0                  │ SnakeYaml: Constructor Deserialization Remote Code Execution │
│                                              │                │          │                   │                      │ https://avd.aquasec.com/nvd/cve-2022-1471                │
└──────────────────────────────────────────────┴────────────────┴──────────┴───────────────────┴──────────────────────┴──────────────────────────────────────────────────────────┘

Exit Code : 1
Image scanning failed. Vulnerabilities found
script returned exit code 1
```

```
┌──────────────────────────────────────────────┬────────────────┬──────────┬───────────────────┬──────────────────────┬──────────────────────────────────────────────────────────┐
│                   Library                    │ Vulnerability  │ Severity │ Installed Version │    Fixed Version     │                          Title                           │
├──────────────────────────────────────────────┼────────────────┼──────────┼───────────────────┼──────────────────────┼──────────────────────────────────────────────────────────┤
│ ch.qos.logback:logback-core (app.jar)        │ CVE-2021-42550 │ MEDIUM   │ 1.2.3             │ 1.2.9                │ logback: remote code execution through JNDI call from within │
│                                              │                │          │                   │                      │ its configuration file...                                │
│                                              │                │          │                   │                      │ https://avd.aquasec.com/nvd/cve-2021-42550               │
├──────────────────────────────────────────────┼────────────────┼──────────┼───────────────────┼──────────────────────┼──────────────────────────────────────────────────────────┤
│ com.fasterxml.jackson.core:jackson-databind (app.jar) │ CVE-2020-36518 │ HIGH │ 2.11.4            │ 2.12.6.1, 2.13.2.1   │ jackson-databind: denial of service via a large depth of │
│                                              │                │          │                   │                      │ nested objects                                           │
│                                              │                │          │                   │                      │ https://avd.aquasec.com/nvd/cve-2020-36518               │
│                                              ├────────────────┤          │                   ├──────────────────────┼──────────────────────────────────────────────────────────┤
│                                              │ CVE-2021-46877 │          │                   │ 2.13.1, 2.12.6       │ jackson-databind 2.10.x through 2.12.x before 2.12.6 and │
│                                              │                │          │                   │                      │ 2.13.x before ...                                        │
│                                              │                │          │                   │                      │ https://avd.aquasec.com/nvd/cve-2021-46877               │
│                                              ├────────────────┤          │                   ├──────────────────────┼──────────────────────────────────────────────────────────┤
│                                              │ CVE-2022-42003 │          │                   │ 2.12.7.1, 2.13.4.1   │ jackson-databind: deep wrapper array nesting wrt         │
│                                              │                │          │                   │                      │ UNWRAP_SINGLE_VALUE_ARRAYS                               │
│                                              │                │          │                   │                      │ https://avd.aquasec.com/nvd/cve-2022-42003               │
│                                              ├────────────────┤          │                   ├──────────────────────┼──────────────────────────────────────────────────────────┤
│                                              │ CVE-2022-42004 │          │                   │ 2.12.7.1, 2.13.4     │ jackson-databind: use of deeply nested arrays            │
│                                              │                │          │                   │                      │ https://avd.aquasec.com/nvd/cve-2022-42004               │
├──────────────────────────────────────────────┼────────────────┤          ├───────────────────┼──────────────────────┼──────────────────────────────────────────────────────────┤
│ org.apache.tomcat.embed:tomcat-embed-core (app.jar) │ CVE-2022-45143 │          │ 9.0.46            │ 9.0.69               │ tomcat: JsonErrorReportValve injection                   │
│                                              │                │          │                   │                      │ https://avd.aquasec.com/nvd/cve-2022-45143               │
│                                              ├────────────────┼──────────┤                   ├──────────────────────┼──────────────────────────────────────────────────────────┤
│                                              │ CVE-2021-43980 │ LOW      │                   │                      │ Apache Tomcat: Information disclosure                    │
│                                              │                │          │                   │                      │ https://avd.aquasec.com/nvd/cve-2021-43980               │
├──────────────────────────────────────────────┤                │          │                   │                      │                                                          │
│ org.apache.tomcat.embed:tomcat-embed-websocket (app.jar) │                │          │                   │                      │                                                          │
```

We can fix CVE-2022-45143 by updating tomcat to version 9.0.69 and CVE-2022-22965 by updating spring-boot to version 2.6.6 or 2.5.12. And build again