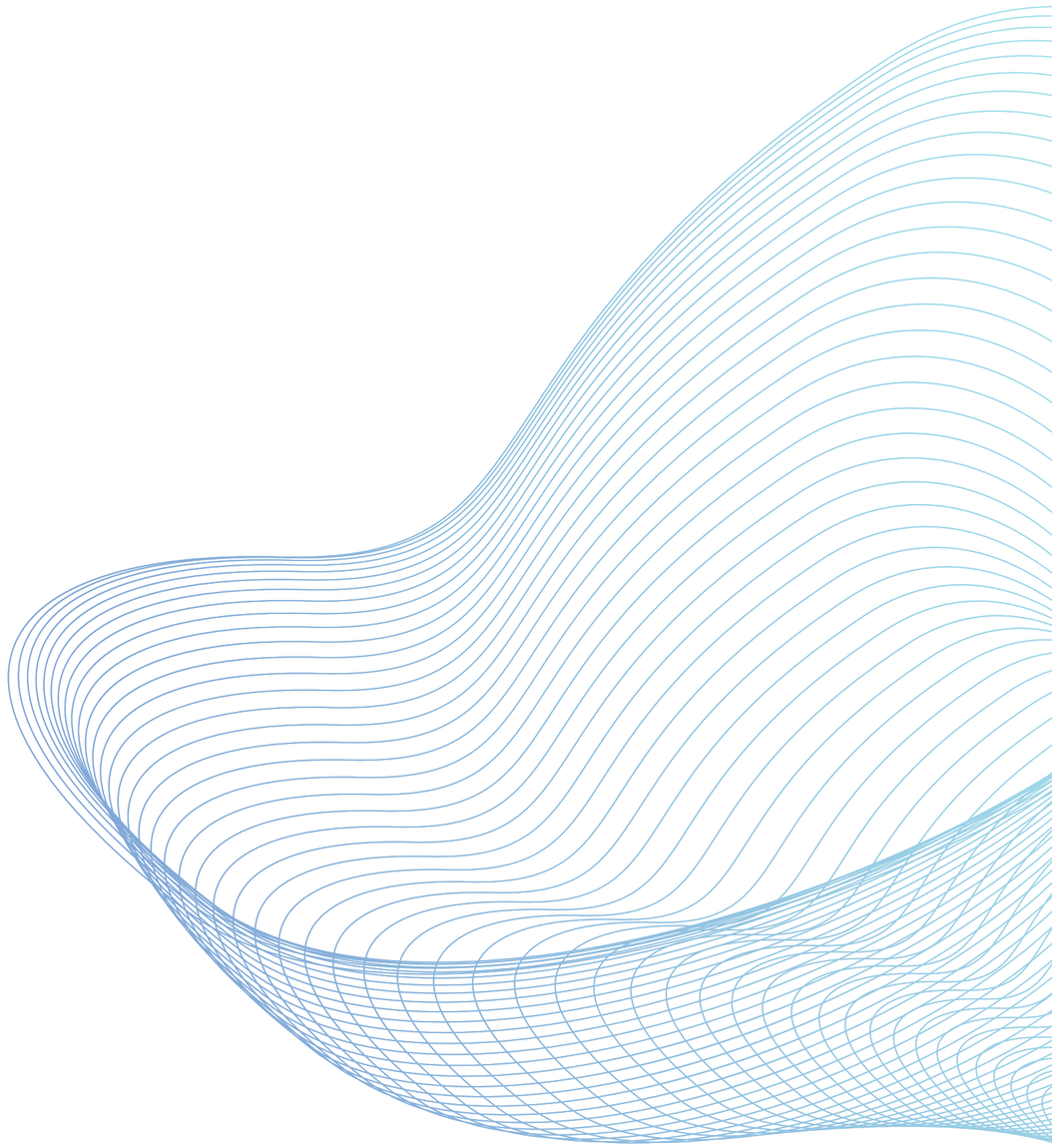
dstny
Engage

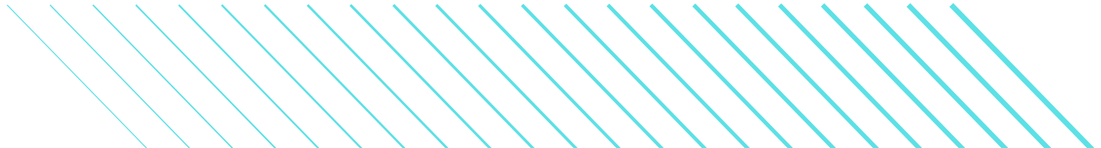
DEMO#2

Security : Mariam Osama

dstny
Engage

SPRINT 2





Scanning Tool

Where are we so far:

The Kubernetes Cluster Scanner Tool is a utility designed to perform security scans on Kubernetes deployments using various security tools. This tool automates the scanning process across different namespaces, identifying vulnerabilities and security risks in your Kubernetes environment.

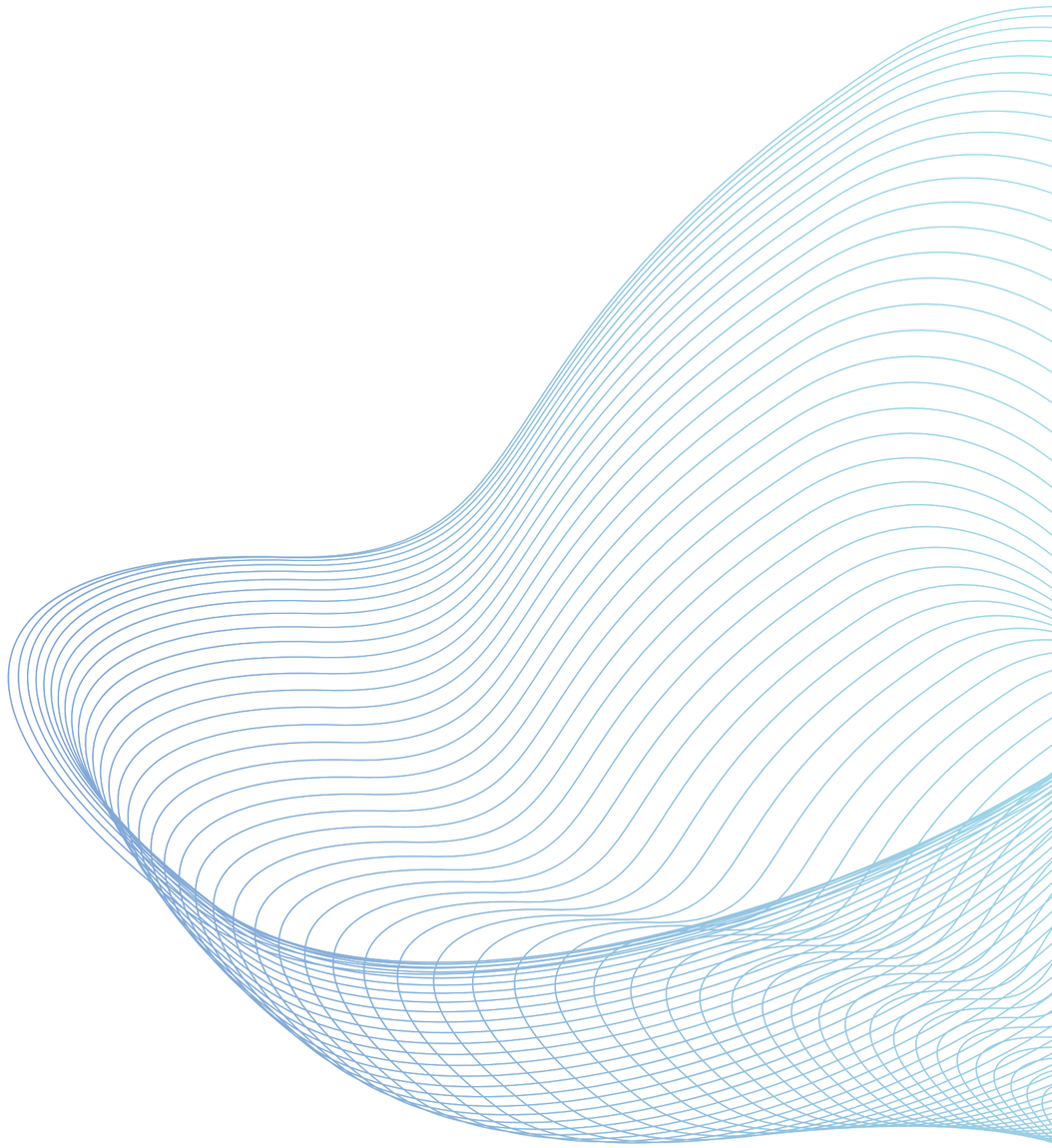
```bash
#!/bin/sh

# Function to print a message with timestamp
print_with_timestamp() {
    echo "$(date -u +"%Y-%m-%dT%H:%M:%SZ") $1"
}

# Install kubectl if not present
if ! command -v kubectl &> /dev/null; then
    print_with_timestamp "kubectl not found, installing..."
    wget -O /usr/local/bin/kubectl https://storage.googleapis.com/kubernetes-release/release/v1.22.2/bin/linux/amd64/kubectl
    chmod +x /usr/local/bin/kubectl
fi

print_with_timestamp "Cluster scan started."

# Iterate through all namespaces
for ns in $(kubectl get namespaces -o=jsonpath='{.items[*].metadata.name}'); do
    print_with_timestamp "Scanning namespace: $ns"

    # Iterate through deployments in the namespace
    for deployment in $(kubectl get deployments -n $ns -o=jsonpath='{.items[*].metadata.name}'); do
        print_with_timestamp "Scanning deployment: $deployment"

        # Run vulnerability scan using Trivy
        trivy_image=$(kubectl get deployment $deployment -n $ns -o=jsonpath='{.spec.template.spec.containers[0].image}')
        trivy_results="/app/scan-results/${ns}_${deployment}_$(date -u +"%Y%m%d%H%M%S")_trivy-results.json"
        trivy image --cache-dir /tmp/trivy --format json -o $trivy_results $trivy_image

        # Run kube-bench
        kube_bench_results="/app/scan-results/${ns}_${deployment}_$(date -u +"%Y%m%d%H%M%S")_kube-bench-results.json"
        ./kube-bench > $kube_bench_results

        # Run kube-hunter
        kube_hunter_results="/app/scan-results/${ns}_${deployment}_$(date -u +"%Y%m%d%H%M%S")_kube-hunter-results.json"
        ./kube-hunter --remote > $kube_hunter_results

        # Run kubescape
        kubescape_results="/app/scan-results/${ns}_${deployment}_$(date -u +"%Y%m%d%H%M%S")_kubescape-results.json"
        ./kubescape scan framework nsa -o $kubescape_results

        print_with_timestamp "Scanning completed for deployment: $deployment"
    done
done

print_with_timestamp "Cluster scan completed."

```

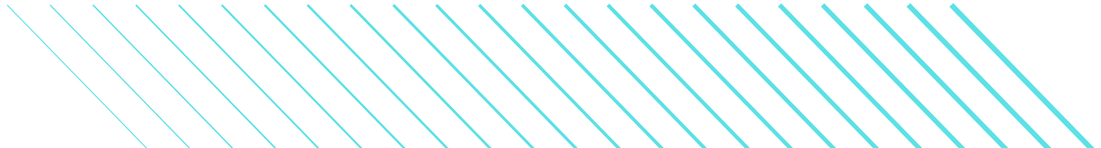
```
/app/scan-results # ls
default_cluster-scanner_20230824160124_trivy-results.json
default_cluster-scanner_20230824160418_kube-bench-results.json
default_cluster-scanner_20230824160418_kube-hunter-results.json
default_python-deployment_20230824160418_trivy-results.json
default_python-deployment_20230824160749_kube-bench-results.json
default_python-deployment_20230824160749_kube-hunter-results.json
hello_hello-world-deployment_20230824160750_trivy-results.json
hello_hello-world-deployment_20230824160846_kube-bench-results.json
hello_hello-world-deployment_20230824160846_kube-hunter-results.json
kube-system_coredns_20230824160846_trivy-results.json
kube-system_coredns_20230824160916_kube-bench-results.json
kube-system_coredns_20230824160916_kube-hunter-results.json
kubernetes-dashboard_dashboard-metrics-scraper_20230824160916_trivy-results.json
kubernetes-dashboard_dashboard-metrics-scraper_20230824161020_kube-bench-results.json
kubernetes-dashboard_dashboard-metrics-scraper_20230824161020_kube-hunter-results.json
```


```
      {
        "VulnerabilityID": "CVE-2022-40897",
        "PkgName": "setuptools",
        "PkgPath": "usr/local/lib/python3.9/site-packages/setuptools-58.1.0.dist-info/METADATA",
        "InstalledVersion": "58.1.0",
        "FixedVersion": "65.5.1",
        "Status": "fixed",
        "Layer": {
          "Digest": "sha256:d271c014c3a0adbd37dae2ce6245711e2876ace65aa5c9b7859a6c515063c45f",
          "DiffID": "sha256:ddab61a14989358634f27d99fed2910a51c05aba9f83ec630386edcf80c8bb53"
        },
        "SeveritySource": "nvd",
        "PrimaryURL": "https://avd.aquasec.com/nvd/cve-2022-40897",
        "DataSource": {
          "ID": "osv",
          "Name": "Python Packaging Advisory Database",
          "URL": "https://github.com/pypa/advisory-db"
        },
        "Title": "Regular Expression Denial of Service (ReDoS) in package_index.py",
        "Description": "Python Packaging Authority (PyPA) setuptools before 65.5.1 allows remote attackers to cause a denial of service via HTML in a crafted package
or custom PackageIndex page. There is a Regular Expression Denial of Service (ReDoS) in package_index.py.",
        "Severity": "MEDIUM",
        "CweIDs": [
          "CWE-1333"
        ],
        "CVSS": {
          "ghsa": {
            "V3Vector": "CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H",
            "V3Score": 7.5
          },
          "nvd": {
```

dstny
Engage

SPRINT 3





Integrating the tool and Testing

The Tasks:

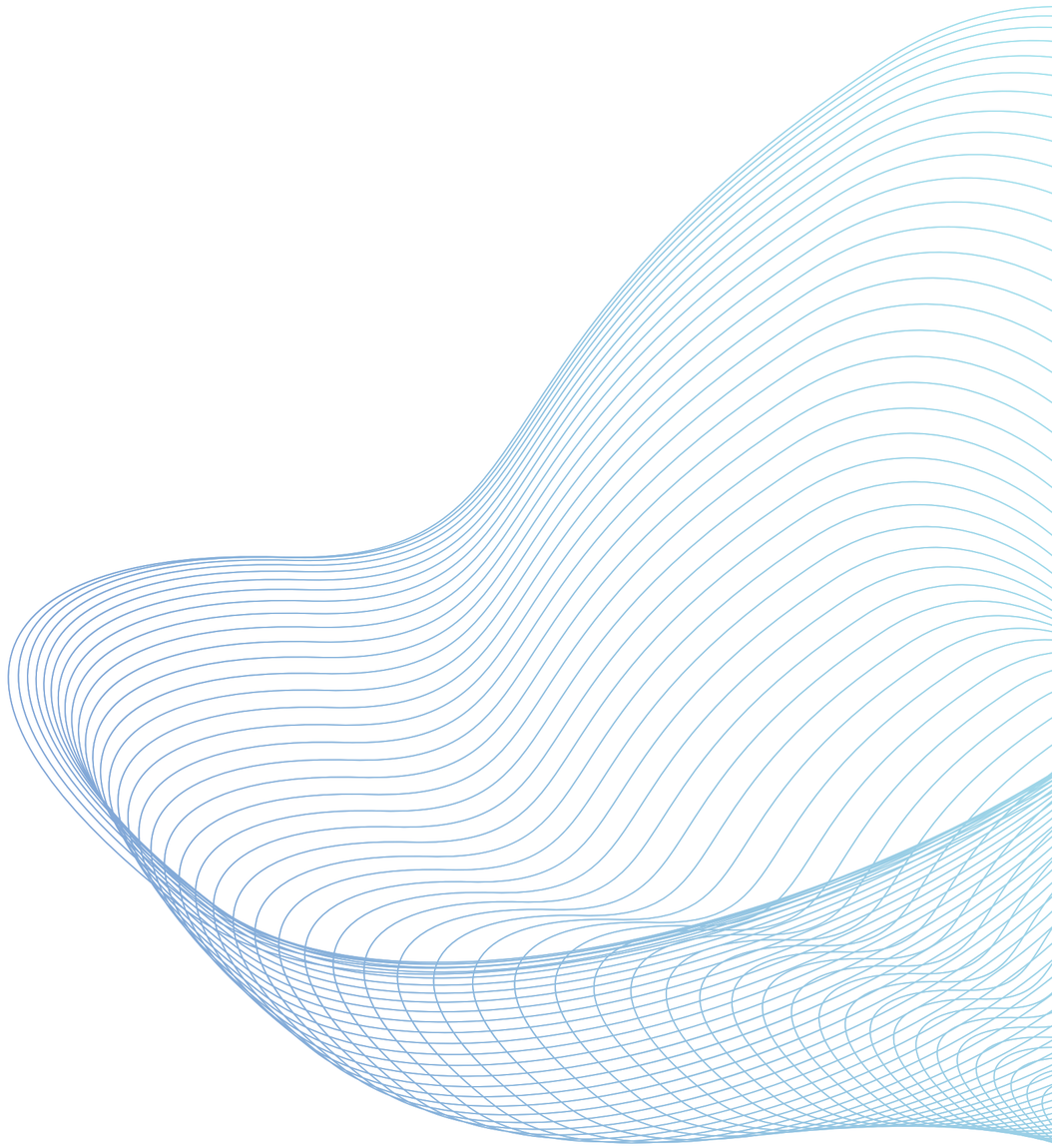
- Implementing the tool in live cluster
- Focusing on the cluster testing

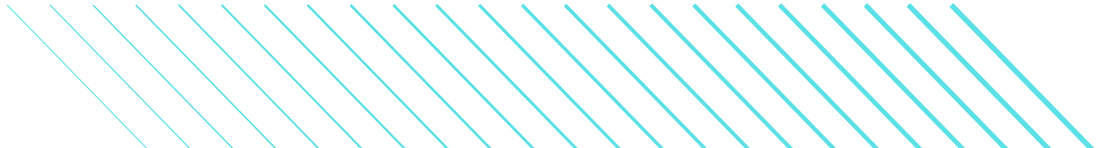
Regarding this sprint I will work on two things :

1. Create a K8s playground to test the hardening best practices and the penetration scenarios
2. Will continue collecting test cases and understanding more about K8s cluster testing

dstny
Engage

SPRINT 4





Testing

The Tasks:

- Testing the cluster
- Analysis of Test Results
- Reporting and Recommendations

THANK YOU

dstiny

Engage