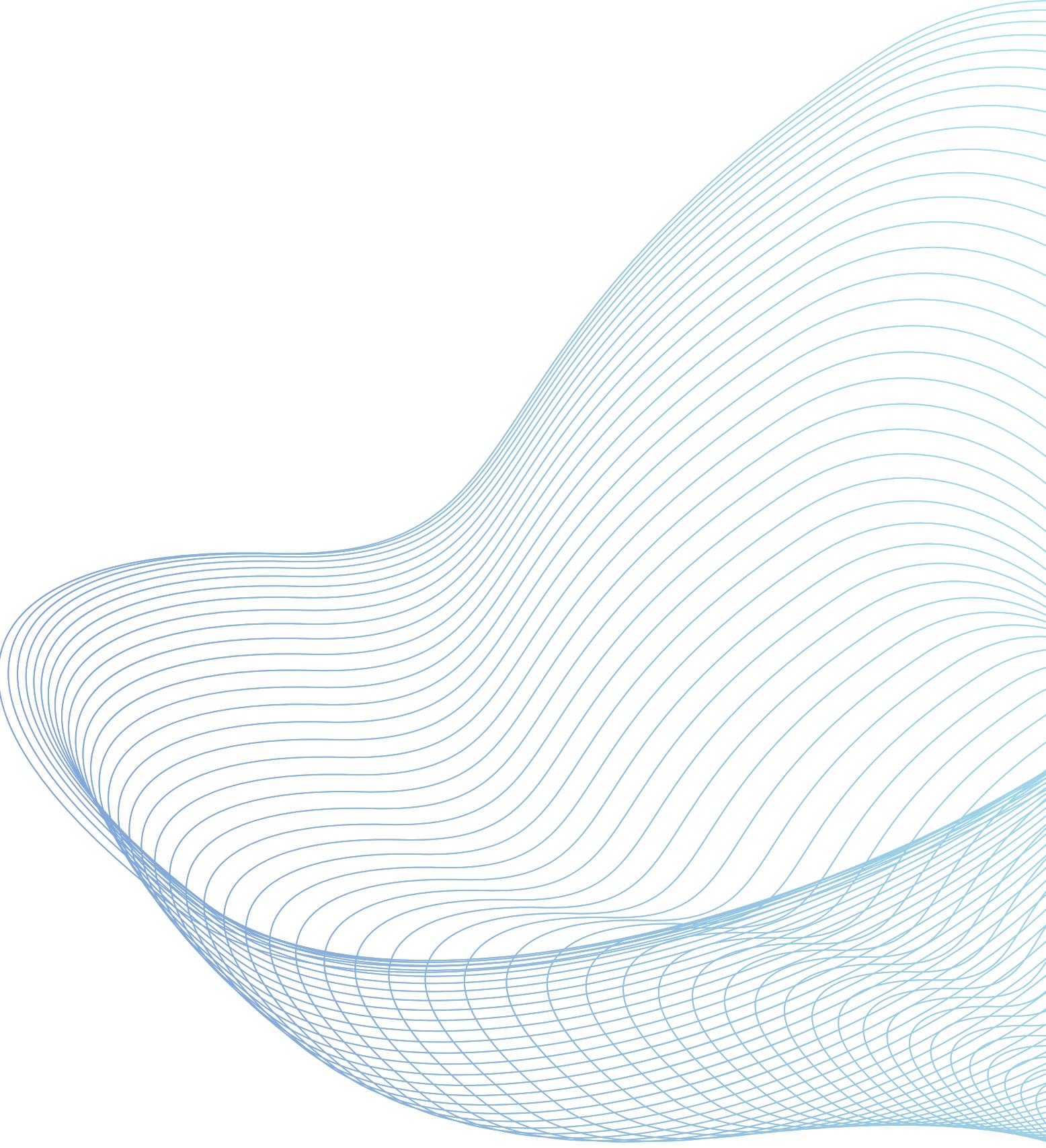




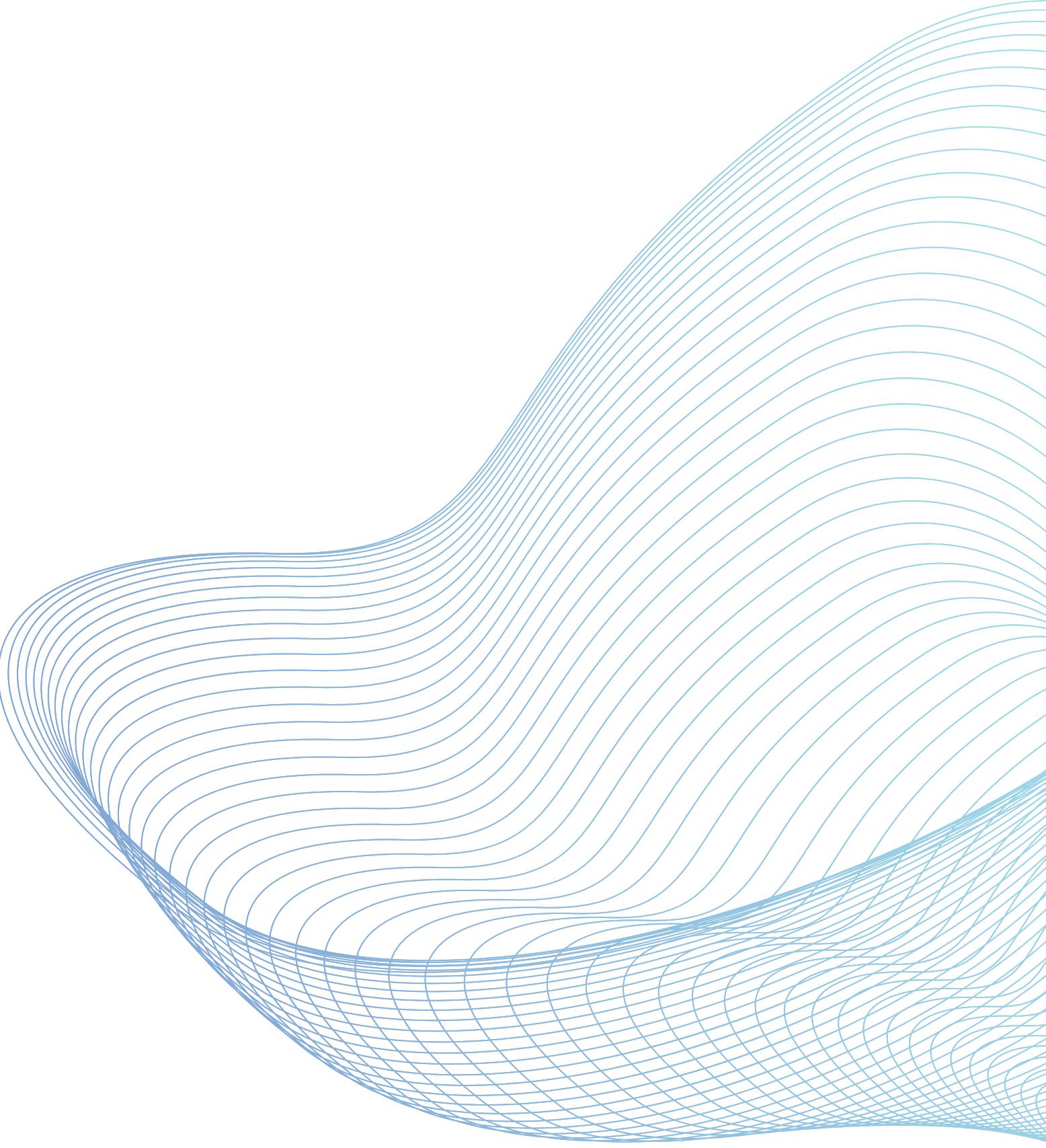
DEMO#2

Security : Mariam Osama



dst^{ny}
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.





TOOL

WALKTHROUGH

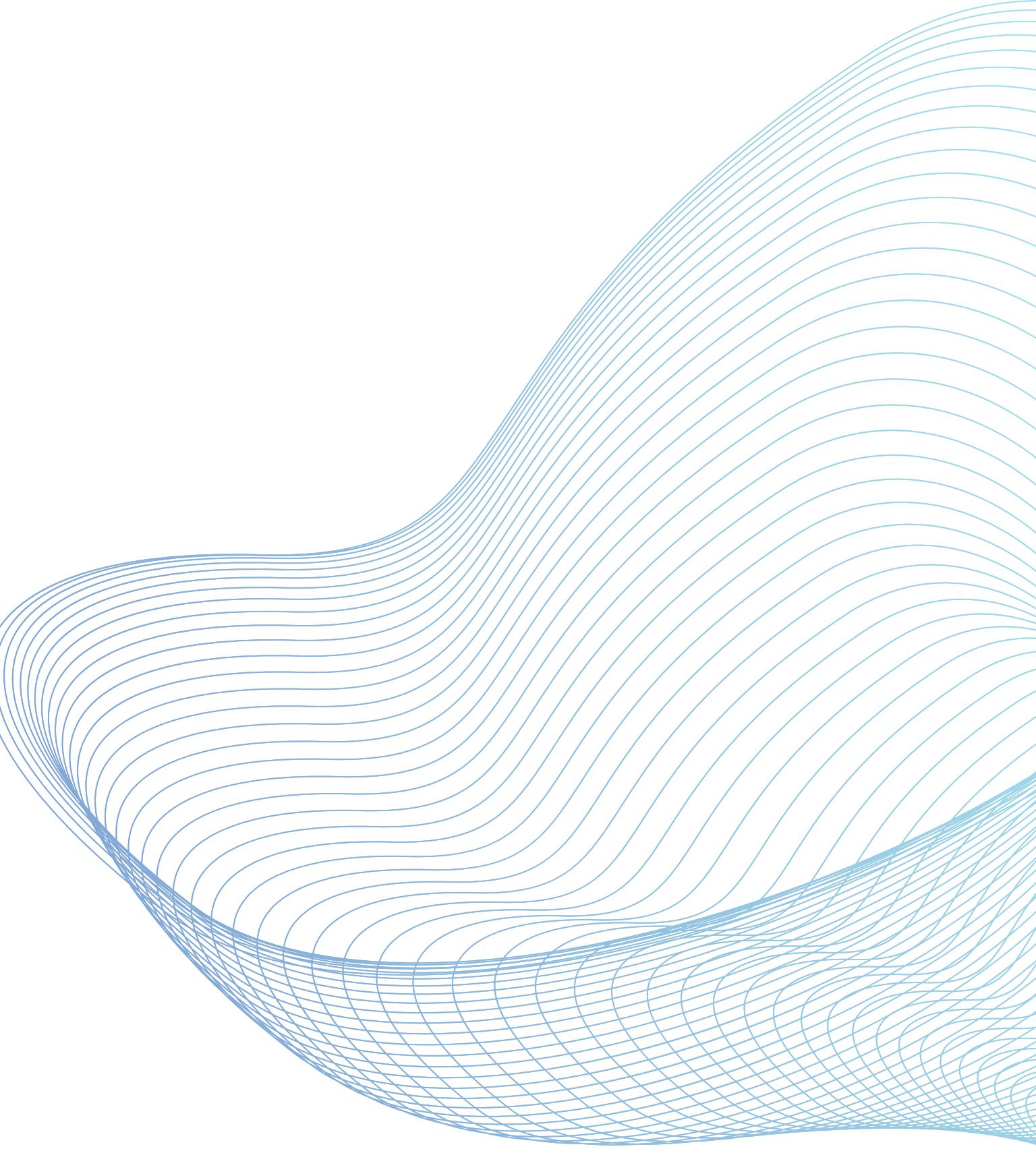
```
1 #!/bin/sh
2
3 # Function to print a message with timestamp
4 print_with_timestamp() {
5     echo "$(date -u +"%Y-%m-%dT%H:%M:%S") $1"
6 }
7
8 # Install kubectl if not present
9 if ! command -v kubectl &> /dev/null; then
10    print_with_timestamp "kubectl not found, installing..."
11    wget -O /usr/local/bin/kubectl https://storage.googleapis.com/kubernetes-release/release/v1.22.2/bin/linux/amd64/kubectl
12    chmod +x /usr/local/bin/kubectl
13 fi
14
15 print_with_timestamp "Cluster scan started."
16
17 # Iterate through all namespaces
18 for ns in $(kubectl get namespaces -o=jsonpath='{.items[*].metadata.name}'); do
19     print_with_timestamp "Scanning namespace: $ns"
20
21     # Iterate through deployments in the namespace
22     for deployment in $(kubectl get deployments -n $ns -o=jsonpath='{.items[*].metadata.name}'); do
23         print_with_timestamp "Scanning deployment: $deployment"
24
25         # Run vulnerability scan using Trivy
26         trivy_image=$(kubectl get deployment $deployment -n $ns -o=jsonpath='{.spec.template.spec.containers[0].image}')
27         trivy_results="/app/scan-results/${ns}_${deployment}_$(date -u +"%Y%m%d%H%M%S")_trivy-results.json"
28         trivy image --cache-dir /tmp/trivy --format json -o $trivy_results $trivy_image
29
30     # Run kube-bench
31     kube_bench_results="/app/scan-results/${ns}_${deployment}_$(date -u +"%Y%m%d%H%M%S")_kube-bench-results.json"
32     ./kube-bench > $kube_bench_results
33
34     # Run kube-hunter
35     kube_hunter_results="/app/scan-results/${ns}_${deployment}_$(date -u +"%Y%m%d%H%M%S")_kube-hunter-results.json"
36     ./kube-hunter --remote > $kube_hunter_results
37
38     # Run kubescape
39     kubescape_results="/app/scan-results/${ns}_${deployment}_$(date -u +"%Y%m%d%H%M%S")_kubescape-results.json"
40     ./kubescape scan framework nsa -o $kubescape_results
41
42     print_with_timestamp "Scanning completed for deployment: $deployment"
43 done
44 done
45
46 print_with_timestamp "Cluster scan completed."
47
```

```
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": {
```

dst^{ny}
Engage

SPRINT 3



Integrating the tool and Testing

The Tasks:

- Implementing the tool in live cluster
- Presisting the tool's results also notifying upon scanning
- Focusing on the cluster testing

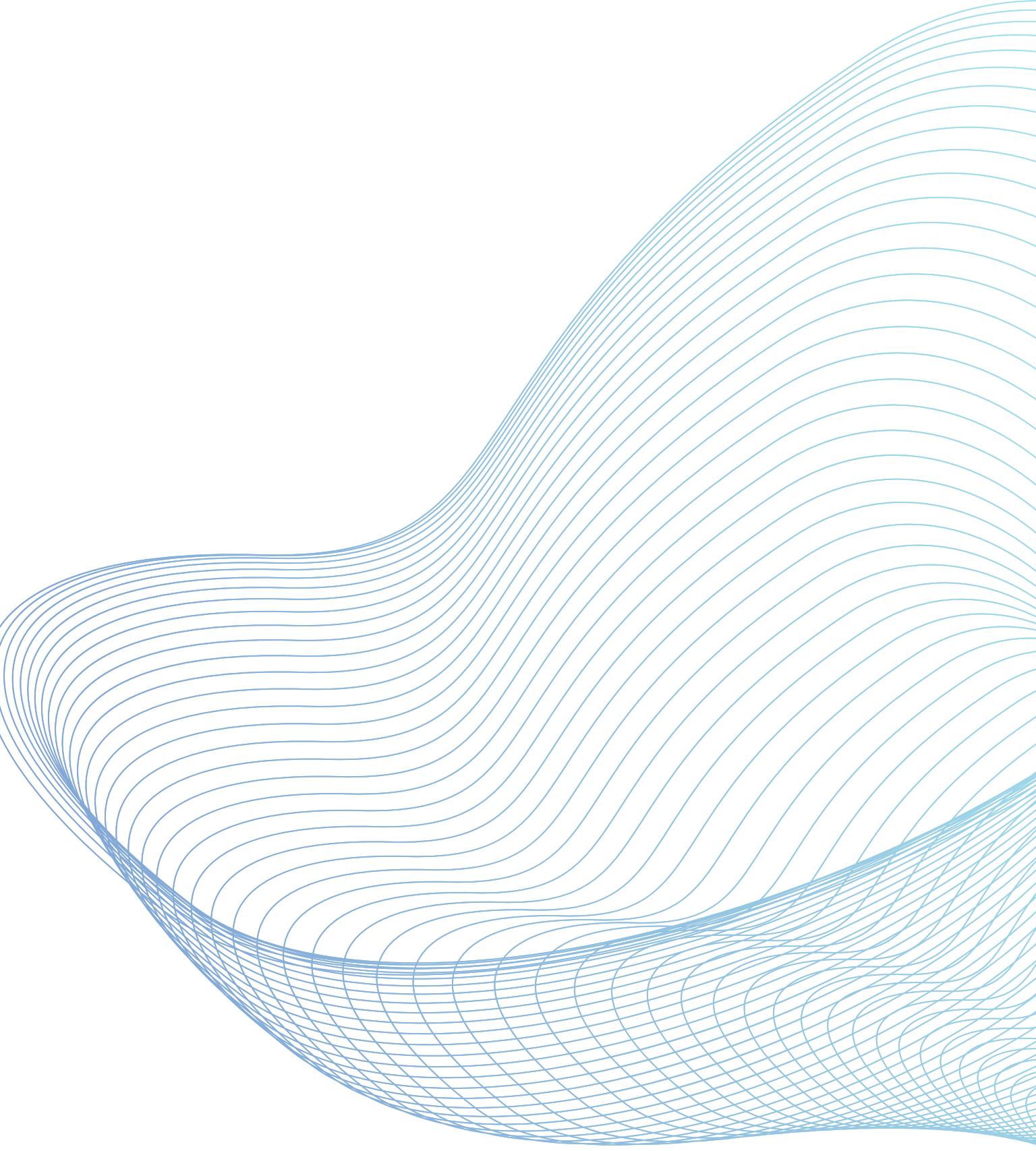


Regarding this sprint I will work on two things :

1. Using Kube-Goat playground to test the hardening best practices and the penetration scenarios
2. Will continue collecting test cases and understanding more about K8s cluster testing



SPRINT 4



Testing

The Tasks:

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



**THANK
YOU**

dstny
Engage

