Security Scanner Usage Guide

This guide shows how to easily use the security scanner container to scan different types of targets.

Quick Start Examples

1. Scan Git Repository (Most Common)

```
# Mount your git repo and scan it
docker run --rm -v /path/to/your/repo:/scan security-scanner:latest \
  security-scanner --git-repo /scan

# Example: Scan current directory
docker run --rm -v $(pwd):/scan security-scanner:latest \
  security-scanner --git-repo /scan
```

2. Scan Kubernetes Manifests

```
# Scan K8s manifests in a directory
docker run --rm -v /path/to/k8s/manifests:/scan security-scanner:latest
\
  security-scanner --k8s-manifest /scan

# Scan specific manifest files
docker run --rm -v $(pwd):/scan security-scanner:latest \
  security-scanner --k8s-manifest /scan/deployment.yaml
/scan/service.yaml
```

PROF

3. Scan Terraform Code

```
# Scan Terraform directory
docker run --rm -v /path/to/terraform:/scan security-scanner:latest \
  security-scanner --terraform-code /scan
```

4. Scan Filesystem/Source Code

```
# General filesystem scan (for any source code)
docker run --rm -v /path/to/source:/scan security-scanner:latest \
  security-scanner --filesystem /scan
```

5. Docker Image Scanning (Using Remote Registry)

```
# Note: Since docker is not available inside container,
# this works by pulling images remotely via the scanning tools
docker run --rm security-scanner:latest \
  security-scanner --docker-image nginx:latest ubuntu:20.04
```

Advanced Usage

Using Configuration Files

```
# Mount config and target directories
docker run --rm \
  -v $(pwd):/scan \
  -v $(pwd)/my-config.yaml:/app/config/scan-config.yaml \
  security-scanner:latest \
  security-scanner --config /app/config/scan-config.yaml
```

Custom Output Directory

```
# Mount output directory to get reports on host
docker run --rm \
  -v $(pwd)/source:/scan \
  -v $(pwd)/reports:/app/reports \
  security-scanner:latest \
  security-scanner --git-repo /scan --output-dir /app/reports
```

Enable/Disable Specific Scanners

PROF

```
# Run only specific scanners
docker run --rm -v $(pwd):/scan security-scanner:latest \
  security-scanner --git-repo /scan \
  --enable-scanner trivy grype semgrep

# Disable specific scanners
docker run --rm -v $(pwd):/scan security-scanner:latest \
  security-scanner --git-repo /scan \
  --disable-scanner dockle hadolint
```

Set Severity Threshold

```
# Only show HIGH and CRITICAL findings
docker run --rm -v $(pwd):/scan security-scanner:latest \
  security-scanner --git-repo /scan --severity-threshold HIGH
```

Common Scan Patterns

Full Repository Security Audit

```
docker run --rm \
  -v $(pwd):/scan \
  -v $(pwd)/security-reports:/app/reports \
  security-scanner:latest \
  security-scanner \
    --git-repo /scan \
    --severity-threshold MEDIUM \
    --format json html sarif \
    --output-dir /app/reports
```

CI/CD Pipeline Integration

```
# Fail build on high/critical findings
docker run --rm -v $(pwd):/scan security-scanner:latest \
  security-scanner \
    --git-repo /scan \
    --severity-threshold HIGH \
    --fail-on-high \
    --format json
```

Multi-Target Scanning

```
# Scan multiple target types in one command
docker run --rm \
  -v $(pwd):/scan \
  -v $(pwd)/k8s:/k8s \
  -v $(pwd)/terraform:/tf \
  security-scanner:latest \
  security-scanner \
    --git-repo /scan \
    --k8s-manifest /k8s \
    --terraform-code /tf \
    --docker-image nginx:latest
```

PROF

Available Scanners

The container includes these security scanners:

- **trivy**: Vulnerability scanner for containers and filesystems

- **grype**: Vulnerability scanner by Anchore
- **syft**: Software bill of materials (SBOM) generator
- **dockle**: Container image linter for security
- **hadolint**: Dockerfile linter
- **checkov**: Static code analysis for infrastructure-as-code
- **conftest**: Policy testing for configurations
- **trufflehog**: Secrets scanner
- **gitleaks**: Git secrets scanner
- **semgrep**: Static analysis for multiple languages

Output Formats

Supported output formats:

- `json`: Machine-readable JSON reports
- `html`: Human-readable HTML reports
- `sarif`: SARIF format for CI/CD integration
- `xml`: XML format reports

Tips for Best Results

1. **Mount volumes properly**: Always mount your source code to `/scan` inside the container
2. **Use absolute paths**: Reference mounted paths with `/scan` prefix
3. **Output directory**: Mount an output directory to persist reports
4. **Large repositories**: Consider using `--timeout` for large codebases
5. **Parallel scanning**: Use `--no-parallel` if you encounter resource issues
6. **Configuration files**: Create reusable config files for consistent scanning

Troubleshooting

Common Issues

PROF

- **Permission errors**: Make sure mounted directories have proper read permissions
- **Out of memory**: Reduce `--max-workers` or use `--no-parallel`
- **Missing dependencies**: Run `--check-dependencies` to verify scanner availability
- **Large repositories**: Increase scanner timeouts with `--timeout`

Debug Commands

```
# List available scanners
docker run --rm security-scanner:latest security-scanner --list-scanners

# Check scanner dependencies
docker run --rm security-scanner:latest security-scanner --check-
dependencies

# Get help
docker run --rm security-scanner:latest security-scanner --help
```