Trivy Security scanner

CONCEPTS

Pick the target type (image, fs, repo, config, k8s, sbom), then choose what to look for with --scanners.

Target image, fs, repo, k8s
↓ analyze
Scanners vuln, secret, misconfig, license
↓ match
Databases cached vuln DB and checks
↓ filter
Report table, json, sarif, SBOM

Defaults: image, fs and repo run vuln + secret; config runs misconfig only. License scanning is off unless requested.

SCAN TARGETS

`trivy image <image>`
Scan an image (local daemon, then registry)

`trivy image --input <file.tar>`
Scan an image saved as tar

`trivy fs <dir>`
Scan a local project, incl. lock files

`trivy repo <path|url>`
Scan a local or remote git repo

`trivy rootfs <dir>`
Scan an extracted root filesystem

`trivy sbom <file>`
Scan an existing SBOM for vulns

SCANNERS

`--scanners vuln`
Only known vulnerabilities

`--scanners vuln,secret,misconfig`
Combine several scanners

`--scanners misconfig`
IaC and Dockerfile misconfigurations

`--scanners secret`
Hard-coded secrets only

SCANNERS (cont.)

`trivy config <dir>`
Misconfig scan of Terraform, K8s, Helm, Dockerfile

FILTERING

`--severity <list>`
Keep only these severity levels

`--ignore-unfixed`
Skip vulns that have no fix yet

`.trivyignore`
One finding ID per line; # comments

`--skip-dirs <glob>`
Do not scan matching directories

`--skip-files <glob>`
Do not scan matching files

OUTPUT AND CI

`-f table`
Human-readable table (default)

`-f json -o <file>`
Write JSON to a file

`-f sarif -o <file>`
SARIF for code-scanning tools

`--exit-code <n>`
Exit with n when issues are found

SBOM

`-f cyclonedx -o <file>`
Generate a CycloneDX SBOM

`-f spdx-json -o <file>`
Generate an SPDX JSON SBOM

KUBERNETES

`trivy k8s --report summary`
Cluster overview (default report)

DATABASE AND CACHE

`--skip-db-update`
Use cached vuln DB, no download