TruffleHog Secret scanner

CONCEPTS

Every finding is verified by calling the provider's API. A verified hit is a live credential: rotate it first, clean history second.

Source git, files, S3...
↓ chunks
Detectors regex + keywords
↓ candidates
Verify call provider API
↓ status
Result verified / unknown / unverified

Unknown means verification failed (network or API error), not that the secret is safe. Unverified means detected but not confirmed live.

Exit code is 0 even with findings unless you pass --fail (then 183). Exit 1 means the scan itself had errors.

SCAN SOURCES

`trufflehog git <uri>`
Scan a git repo by URL or file:// path

`trufflehog git file://.`
Scan the local repo's history

`trufflehog filesystem <path>...`
Scan files and directories

`trufflehog github --org=<org>`
Scan every repo of an org

`trufflehog github --repo=<url>`
Scan one GitHub repo

`trufflehog docker --image <image>`
Scan a container image

`trufflehog s3 --bucket=<name>`
Scan an S3 bucket

GIT AND GITHUB OPTIONS

`--since-commit <ref>`
Scan only commits after this ref

GIT AND GITHUB OPTIONS (cont.)

`--branch <name>`
Scan one branch

`--token <token>`
GitHub token to avoid rate limits

RESULTS AND OUTPUT

`--results=verified`
Show only confirmed live secrets

`--results=verified,unknown`
Also keep unknown (verification failed)

`--json / -j`
Output one JSON object per finding

`--no-verification`
Skip API checks; report all matches

`--fail`
Exit 183 when results are found

DETECTORS AND FILTERS

`--include-detectors <names>`
Run only these detectors (names or IDs)

`trufflehog:ignore`
Comment on a line to skip its finding

ANALYZE AND INSTALL

`trufflehog help <command>`
Flags for one subcommand

`trufflehog --version`
Print the installed version

`brew install trufflehog`
Install on macOS or Linuxbrew

CI

`uses: trufflesecurity/trufflehog@main`
GitHub Action step; checkout with fetch-depth: 0

`trufflehog git file://. --since-commit main --branch <b> --fail`
PR-style scan of new commits