**Gitleaks - Secrets Detection Tool**

**Gitleaks** is a fast, open-source **SAST (Static Application Security Testing)** tool used to detect **hardcoded secrets like API keys, passwords, tokens, and credentials** in your Git repositories or any source code folder.

**✅ 1. INSTALLATION ON UBUNTU**

**🔹 Option 1: Install via Linux Internal Package**

sudo apt install gitleaks

**🔹 Option 2: Download Precompiled Binary**

curl -sSL https://github.com/gitleaks/gitleaks/releases/latest/download/gitleaks-linux-amd64 -o gitleaks

chmod +x gitleaks

sudo mv gitleaks /usr/local/bin/

**🔹 Verify Installation**

gitleaks version

**✅ 2. BASIC USAGE COMMANDS**

**🔸 Scan Current Directory (non-git)**

gitleaks detect --source .

This scans the current working directory **recursively** for secrets in plain files, not just Git history.

**🔸 Scan with Redacted Secrets (hide actual values)**

gitleaks detect --source . --redact

The --redact flag replaces detected secrets with \*\*\* to avoid accidental exposure in logs/reports.

**🔸 Save Scan Results to File**

gitleaks detect --source . -r b.txt

This saves results to b.txt in plain format.

**🔸 Save Redacted Output to File**

gitleaks detect --source . -r b.txt --redact

Redacted + Saved Output.

**✅ 3. ADVANCED USAGE**

**🔸 Scan a Specific File Only**

gitleaks detect --source . --path config.yaml

**🔸 Set Custom Config File**

gitleaks detect --source . --config-path path/to/gitleaks.toml

Allows use of custom regex patterns, allowlists, etc.

**🔸 Exit with Zero Even if Leaks Found (for CI testing)**

gitleaks detect --source . --exit-code 0

**🔸 Scan with Verbose Output**

gitleaks detect --source . --verbose

**🔸 Scan with JSON Output**

gitleaks detect --source . -f json -r report.json

Formats:

* json
* csv
* sarif

**✅ 4. CONFIGURATION WITH GITLEAKS.TOML**

You can customize your detection rules using a **.gitleaks.toml** configuration file.

[[rules]]

description = "AWS Secret Access Key"

regex = '''AKIA[0-9A-Z]{16}'''

tags = ["key", "AWS"]

[allowlist]

description = "Ignore specific files or paths"

files = ["test\_config.yaml", "README.md"]

Use your config:

gitleaks detect --source . --config-path .gitleaks.toml

**✅ 5. SAMPLE OUTPUT (REDACTED)**

{

"description": "AWS Secret Access Key",

"file": "terraform.tfvars",

"line": 14,

"secret": "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*",

"ruleID": "aws-access-key",

"commit": "",

"repo": "",

"author": "",

"email": "",

"date": ""

}

**✅ 6. CLEAN UNINSTALL (Binary)**

sudo rm /usr/local/bin/gitleaks

**✅ 7. BEST PRACTICES**

* ✅ Always use --redact when scanning in pipelines.
* ✅ Run on pull requests to catch exposed secrets early.
* ✅ Add .gitleaks.toml to your repo to customize scans.
* ❌ Never store secrets in version control, even temporarily.
* ✅ Rotate any secret found in historical commits.

**✅ 8. FULL COMBO EXAMPLE**

gitleaks detect \

--source . \

--redact \

--config-path .gitleaks.toml \

-f json \

-r gitleaks-report.json