

Forum: [Hack Forums](#)

Thread: [GIT SEC](#)

Oxycodone - August 12, 2026, 1:48 am

I will be giving away a vouch copy for someone wanting to do a review of this.

Current pricing is 420\$ a month.

You will get AI Credits to use to scan all your repos.

You will have 60 repo's limit per month. I can increase incase of dead / failed repos.

You will be given extra limits during your first few days just to ensure you understand how the platform works.

Only accepting 3 customers.

GIT-SEC

AUTOMATED GIT EXPOSURE DISCOVERY
& SECURITY INTELLIGENCE

Discover. Recover. Analyze. Understand.

TURN EXPOSED SOURCE CODE INTO ACTIONABLE INTELLIGENCE.

Git-Sec is a security operations platform designed to help organizations discover exposed Git repositories, securely recover their contents, identify sensitive data, and turn large volumes of source code into actionable security intelligence.

It brings repository discovery, credential analysis, static vulnerability scanning, AI-assisted review, file investigation, reporting, and operational workflows into one centralized platform.

01 // THE PROBLEM

Source-code exposure is rarely limited to a single misplaced .git directory.

GitHub and GitHub Enterprise

GitLab and self-managed GitLab

Bitbucket Cloud and Bitbucket Server

Gitea and Forgejo

Development, staging, legacy, and forgotten infrastructure

Public web servers containing recoverable Git metadata

Acquired companies and unmanaged business units

Traditional scanners may identify that .git/HEAD exists, but often stop there.

Security teams must then manually validate the finding, reconstruct repository objects, inspect thousands of files, search commit history, verify credentials, and create a usable report.

[center]Git-Sec automates that workflow.[/center]

02 // CORE PLATFORM CAPABILITIES

REPOSITORY DISCOVERY

Git-Sec discovers potential repository exposures through:

- Direct target submission
- Bulk URL imports
- Automated queue ingestion
- LeakIX-assisted discovery
- Git host and repository enumeration
- Exposed .git metadata detection
- Live endpoint verification
- Organization and account repository discovery

Targets are normalized, deduplicated, prioritized, and assigned to an organized processing queue.

SECURE REPOSITORY RECOVERY

The platform can recover repositories through:

- Standard Git clone
- Authenticated HTTPS clone
- Exposed .git reconstruction
- Git object and reference recovery
- Safe working-tree materialization
- Shared repository caching
- Automatic retry and repair of incomplete downloads

Git-Sec validates HEAD, Git configuration, references, object connectivity, tree structure, checkout paths, and resulting files before accepting a repository as successfully downloaded.

Incomplete repositories are not silently treated as healthy. Missing objects trigger a controlled repair workflow.

MULTI-PROVIDER TOKEN MANAGEMENT

Git-Sec supports managed credentials for:

- GitHub
- GitHub Enterprise
- GitLab
- Self-managed GitLab
- Bitbucket Cloud
- Bitbucket Server
- Gitea

Forgejo

The platform supports personal access tokens, OAuth credentials, bearer tokens, and supported username/app-password combinations.

It can:

- Detect providers automatically
- Recognize public and self-hosted installations
- Validate credentials using provider identity APIs
- Enumerate accessible repositories
- Securely pull private repositories
- Track token validity and usage
- Associate credentials with discovered repositories
- Encrypt tokens at rest
- Prevent credentials from being written into clone URLs or Git configuration

REPOSITORY FILE EXPLORER

Downloaded repositories become searchable security workspaces.

- Navigate complete directory trees
- Search and filter files
- Open source files directly
- Review files identified as security-sensitive
- Inspect configuration and infrastructure files
- Sort findings by severity, category, and relevance
- Review binary-file metadata safely
- Access cached content from verified live-file probes
- Automatically repair incomplete repository objects

The file viewer is designed around strict path containment and does not execute repository content.

CREDENTIAL & SECRET INTELLIGENCE

Git-Sec searches repositories and available history for:

- API tokens
- Cloud credentials
- Database connection strings
- Private keys
- OAuth secrets
- Service credentials
- Webhook URLs
- Deployment tokens
- Embedded Git credentials
- Environment files
- Internal service endpoints

Findings are normalized into structured records containing source file, type, severity, confidence, and supporting context.

AUTOMATED VULNERABILITY DETECTION

Every downloaded repository can undergo automatic defensive analysis.

The platform combines:

- Local deterministic security rules
- Secret detection
- Source-code pattern analysis
- Configuration review
- Dependency and CVE detection
- Infrastructure-file analysis
- Data-flow-oriented AI review
- Exploitability and confidence classification

Potential findings include:

- SQL injection
- Command injection
- Server-side request forgery
- Cross-site scripting
- Authentication weaknesses
- Authorization failures
- Unsafe deserialization
- Path traversal
- Open redirects
- Embedded credentials
- Insecure deployment configuration
- Supply-chain and dependency risks

AI-ASSISTED DIRECTORY SCANNING

Git-Sec can automatically select security-relevant files and analyze them with configured AI providers.

The AI pipeline:

- Treats repository content as untrusted data
- Ignores instructions embedded in source files
- Redacts credential-like values before external analysis
- Excludes sensitive files from external transmission
- Validates AI-returned file paths
- Produces structured, reviewable findings
- Supports free-model routing when configured
- Falls back to deterministic local scanning

AI findings include severity, confidence, affected file, line location, evidence, potential impact, and remediation guidance.

GIT CHECKOUT PROTECTION

Repositories can contain hostile metadata and development-environment traps.

Git-Sec's checkout pipeline therefore disables:

- Git hooks

- SSH execution
- External Git protocols
- Local file protocols
- Credential helpers
- Smudge and process filters
- Automatic submodule recursion
- Executable filesystem monitors
- Attacker-controlled askpass programs
- Repository templates
- Symlink materialization

Before checkout, the platform checks for:

- Path traversal
- .git path aliases
- Control characters
- Unicode and case collisions
- Excessively large blobs
- Oversized expanded trees
- Invalid object connectivity
- Malformed references
- Unsafe Git configuration

IDE & AI-AGENT TRAP DETECTION

Git-Sec detects repository content capable of influencing development tools, including:

- .vscode/tasks.json
- Automatic folder-open tasks
- .kiro tool hooks
- .kiro steering instructions
- Cursor workspace rules
- Repository-local AI instructions
- Git-hook payloads
- Reverse-shell patterns
- Download-and-execute commands
- Miner indicators

Harmless security-canary tests prove that the scanner detects these paths while the hardened Git runner prevents checkout hooks from executing.

SCALABLE QUEUE ARCHITECTURE

Git-Sec separates interactive API traffic from expensive background work.

Production uses:

- PostgreSQL-backed durable queues
- Atomic job claiming
- Dedicated background workers
- Priority scheduling
- Per-user concurrency controls
- Round-robin tenant fairness
- Duplicate-target suppression

Stale-job recovery
Automatic retry and repair
Queue depth and worker metrics

One large customer cannot monopolize the candidate window.

Jobs are interleaved across users while retaining priority and age order within each user's queue.

[center]
GIT-SEC

EXPOSURE DISCOVERY.
REPOSITORY RECOVERY.
SECURITY INTELLIGENCE.

One platform. One workflow. From exposed Git metadata to actionable findings.
[/center]
[contract=tid]Start Contract[/contract]

DeathPenalty666 - August 12, 2026, 9:08 am
Awsome, Good luck with the Sales!!! I have some free time in the next days if you want a review)

Oxycodone - August 12, 2026, 11:30 am
(Aug 12, 2026 09:08 AM)DeathPenalty666 Wrote: [\[CITE\]](#) Awsome, Good luck with the Sales!!! I have some free time in the next days if you want a review)
Message me @IvanBreachy on telegram will set you up asap.

DeathPenalty666 - August 12, 2026, 12:15 pm
(Aug 12, 2026 11:30 AM)Oxycodone Wrote: [\[CITE\]](#) Message me @IvanBreachy on telegram will set you up asap.
?