

Security Review Methodology | CommitScope 2.4.2

01 / Executive decision

The decision to make

The company needs repeatable review of an exact Git commit, initiated by an employee using Claude Code, with an authorized source transfer and a protected handoff to an assigned human reviewer. A useful result is not a green badge; it is a bounded evidence package on which that reviewer can make and record a decision.

Recommendation: use CommitScope 2.4.2 as the local collection and verification workflow. Semgrep, Gitleaks, Trivy, independent Claude Hunter and Verifier, and human triage are all required. `READY_FOR_HUMAN_REVIEW` means handoff, not approval.

This is a rational fit to the stated workflow, not a measured claim of best detection, lowest cost, formal compliance, or universal security. It does not replace threat modeling, authorization tests, dynamic testing, a penetration test, or accountable risk acceptance. The source survey is checked on 2026-09-23. [S1] [S3] [S4] [S9]

02 / International practice

Complementary layers, not competing badges

NIST SSDF 1.1 is a final high-level secure-development framework; its 1.2 revision is a draft as of the survey date. OWASP SAMM organizes maturity work across Governance, Design, Implementation, Verification, and Operations. OWASP ASVS 5.0.0 supplies testable application-control requirements. These sources guide the program and its review questions; none is a claim that one tool completes every practice. [S1] [S2] [S3] [S4]

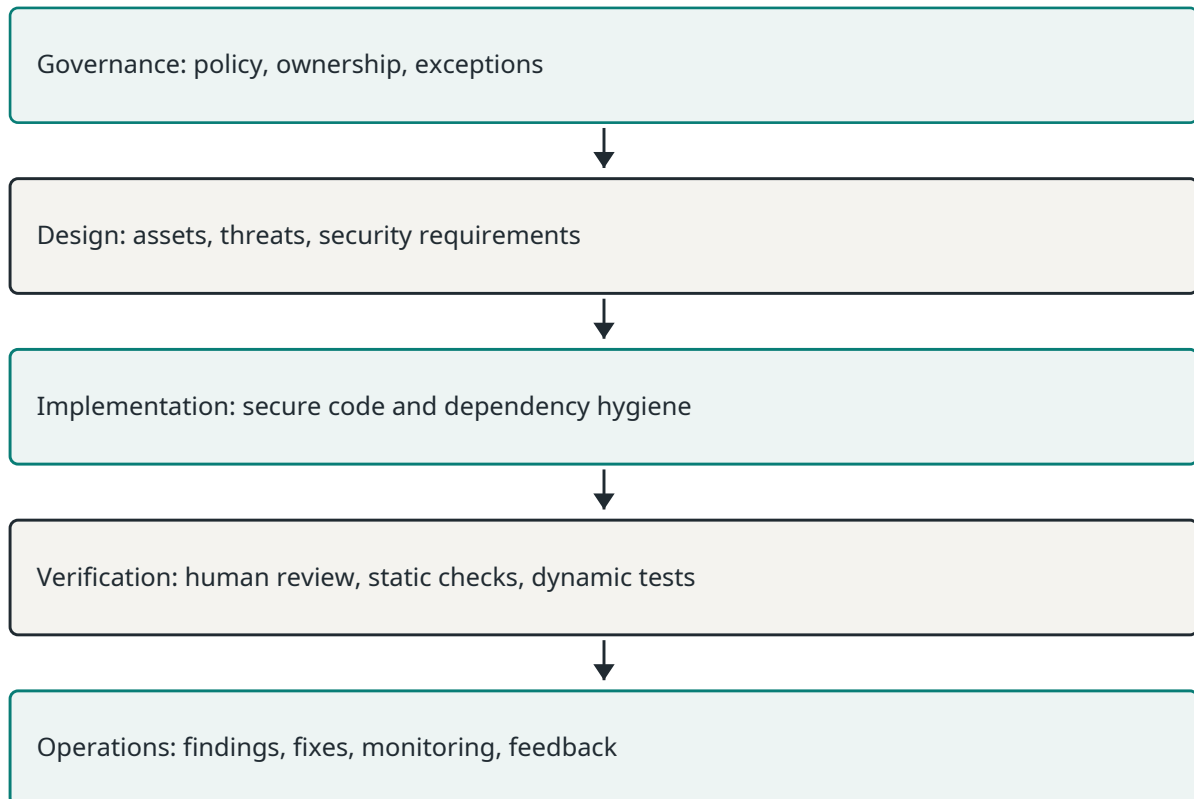


Figure 1. A layered security-review program; CommitScope addresses part of verification and evidence.

Manual review and threat modeling ask whether a control is appropriate for the business flow. SAST, secret detection, SCA and IaC checks search repeatable classes of defects; dynamic tests exercise deployed behavior. AI-assisted analysis can form and challenge code-path hypotheses, but source comments and retrieved material are untrusted instructions. OWASP treats prompt injection as a distinct risk; Anthropic documents permissions and limits, not immunity. [S3] [S5] [S6] [S9]

03 / Options compared

Qualitative assessment under our constraints

The following judgments are the CommitScope team's inference for the agreed local, account-based process. They are not scores from NIST, OWASP, or a benchmark; coverage and cost depend on repository, policy, scanner rules, model, and reviewer skill.

Table 1. Review options: coverage, repeatability, and source transfer

Option	Coverage type	Repeatability	Confidentiality / transfer
Manual only	Deep context, bounded by reviewer time	Procedure-dependent	Can remain local
Scanners only	Known patterns and dependency/config data	High for pinned rules and DB	Usually local scan; DB updates need network
AI only	Contextual hypotheses, variable accuracy	Model and prompt dependent	Source request reaches provider
Central service	Depends on chosen stack and controls	Can centralize policy	Requires approved centralized custody
Local combined	Complementary machine checks and AI challenge	Exact commit and recorded versions	Bounded packet to provider by consent

Table 2. Review options: operating burden, cost drivers, and auditability

Option	Operating burden	Cost drivers	Evidence auditability
Manual only	Reviewer preparation and follow-up	Expert hours	Needs disciplined notes
Scanners only	Rules, DB and exception upkeep	Compute and triage	Raw/normalized output can be retained
AI only	Prompt, model and privacy governance	Quota and human checking	Weak without strict input/output record
Central service	Identity, hosting and access controls	Platform plus operations	Potentially strong if configured
Local combined	Employee setup and reviewer handoff	Scanner upkeep, model quota, review time	Manifest plus protected evidence; unsigned

The combined option is favored because it matches the approved employee-run and protected-handoff model while preserving complementary evidence. It is not sufficient on its own for runtime exposure, business-logic abuse, exploitability in production, or organization-wide policy assurance. [S1] [S4] [S9]

04 / Why this implementation

Decision criteria from the company workflow

Table 3. Mandatory constraints and the resulting design

Company constraint	CommitScope response	Residual control
Claude Code account required	Account-only Hunter and Verifier; no API fallback	Approve account and data terms
Employee starts locally	One commitscope review command	Train and authorize operator
Exact reviewed code	Clean full commit ID and immutable snapshot	Scope and change control
Approved source transfer	Explicit --allow-code-upload and screened bounded packet	Owner authorizes each transfer
Protected handoff	New restricted run directory outside target	Company storage and access policy
Assigned human reviewer	verify-review plus normalized and raw evidence	Identity, decision, exceptions
Supported employee hosts	macOS and glibc Linux x86_64/ARM64	Host enrollment and updates

Manual-only review cannot economically repeat machine-pattern checks for each commit. Scanner-only review cannot answer every reachability or business-invariant question. AI-only review lacks deterministic baseline checks and can be misled by untrusted repository text. A central service could be useful later, but changes custody, identity, and operating responsibilities. None of these is inherently inferior in every organization; the local combination fits these specific constraints.

Keep external controls: threat model and ASVS-derived requirements; dynamic tests and authorized reproduction; repository protections; reviewer identity and risk acceptance; data-transfer approval; retention and incident response. CommitScope collects evidence for those owners, not their authority.

05 / Architecture and trust

What runs locally and what crosses the boundary

The employee chooses a clean exact commit and an approved policy outside the target. CommitScope exports a snapshot, runs pinned Semgrep, Gitleaks, and Trivy checks, and retains scanner output in the restricted run. It screens recognizable secrets and constructs a bounded source packet. Scanner/DB updates are separate network dependencies. A screening pass is not a guarantee that unknown secrets are absent. Version 2.4.2 adds five bounded Java Semgrep rules for direct JDBC SQL concatenation, process execution, deserialization, and disabled TLS checks; this is not exhaustive Java coverage.

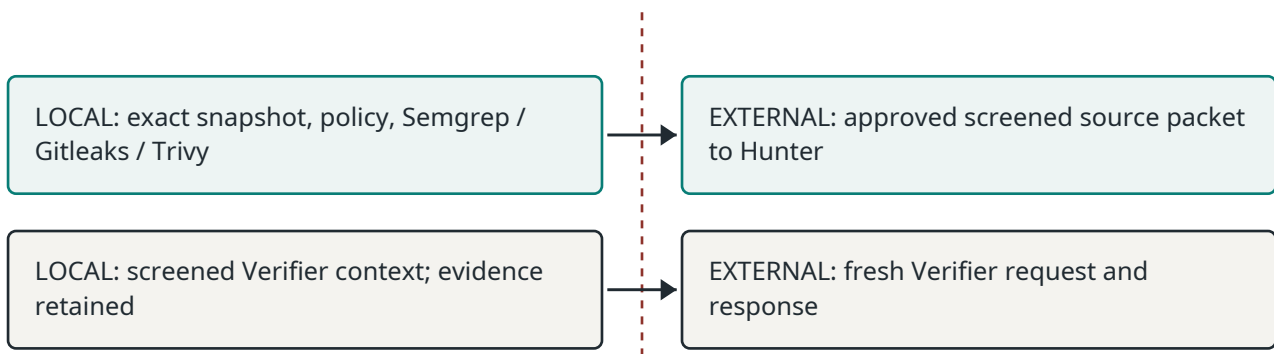


Figure 2. Dotted line: employee-controlled host versus Anthropic account service.

Hunter and Verifier are separate Claude Code account calls. The operator must approve sending the screened packet and check the intended account's terms, privacy settings, and retention: personal and commercial accounts differ. The repository and scanner/model output are untrusted input to the model. Anthropic describes security safeguards and data usage, but organizational permission to upload must be established locally. [S5] [S6] [S7] [S8]

Missing scanner coverage, authentication, model access, timeout, malformed AI output, or incomplete Hunter/Verifier coverage yields INCOMPLETE. The run preserves a protected evidence trail; its manifest is written last and is unsigned. verify-review checks internal consistency, not author identity or immutable custody. A human still decides.

06 / From hypothesis to fix

Evidence must survive challenge and re-review

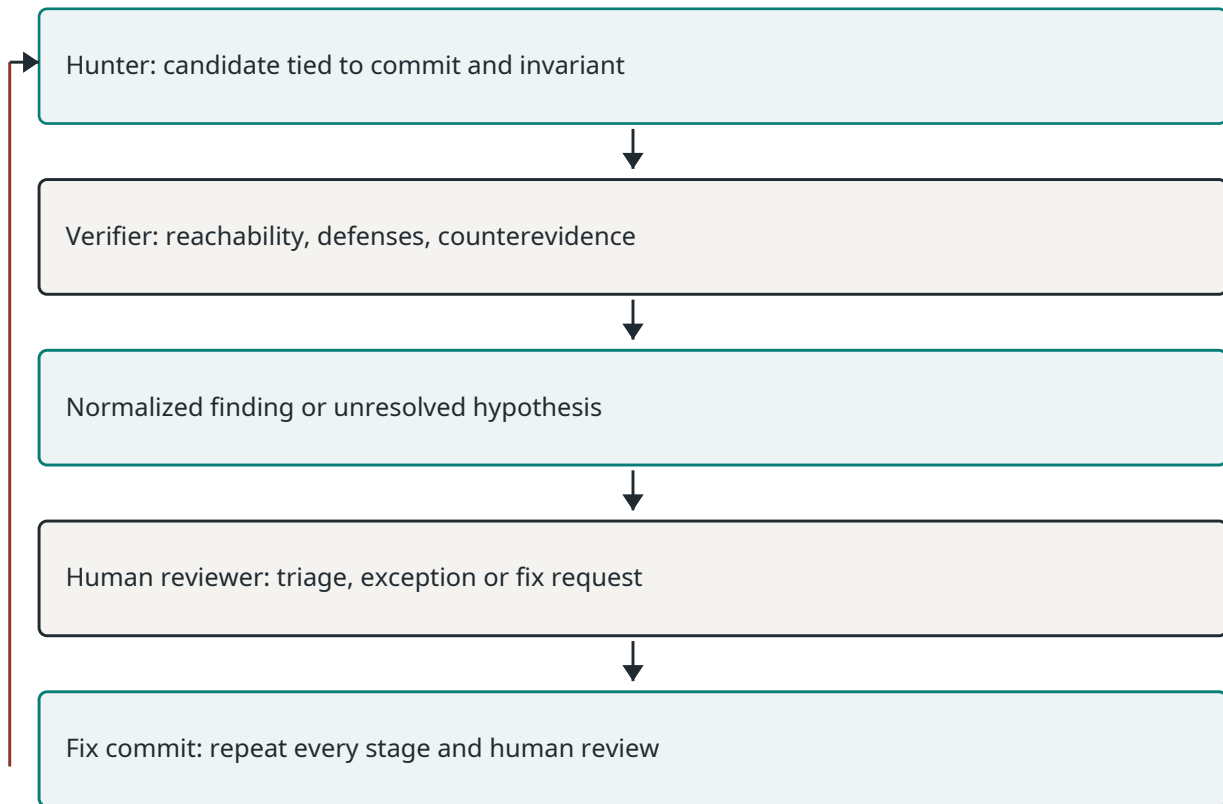


Figure 3. Finding lifecycle; a new commit always starts a full new run.

Synthetic example: a vulnerable order endpoint accepts an order ID and fetches that order without enforcing owner = current actor. Hunter proposes an IDOR path; Verifier checks authentication, the reachable handler, and any downstream ownership guard. The reviewer confirms or rejects the hypothesis using the code and authorized tests. A fix adds an ownership-enforcing lookup; the new commit receives a complete new run. This example illustrates method, not measured model recall.

A finding disappearing after a fix is not proof of remediation. Compare the two exact commits, test the security invariant in an authorized environment, inspect omissions and counterevidence, and record the reviewer's decision for each run.

07 / Controlled adoption

Proposed sequence and pilot evidence

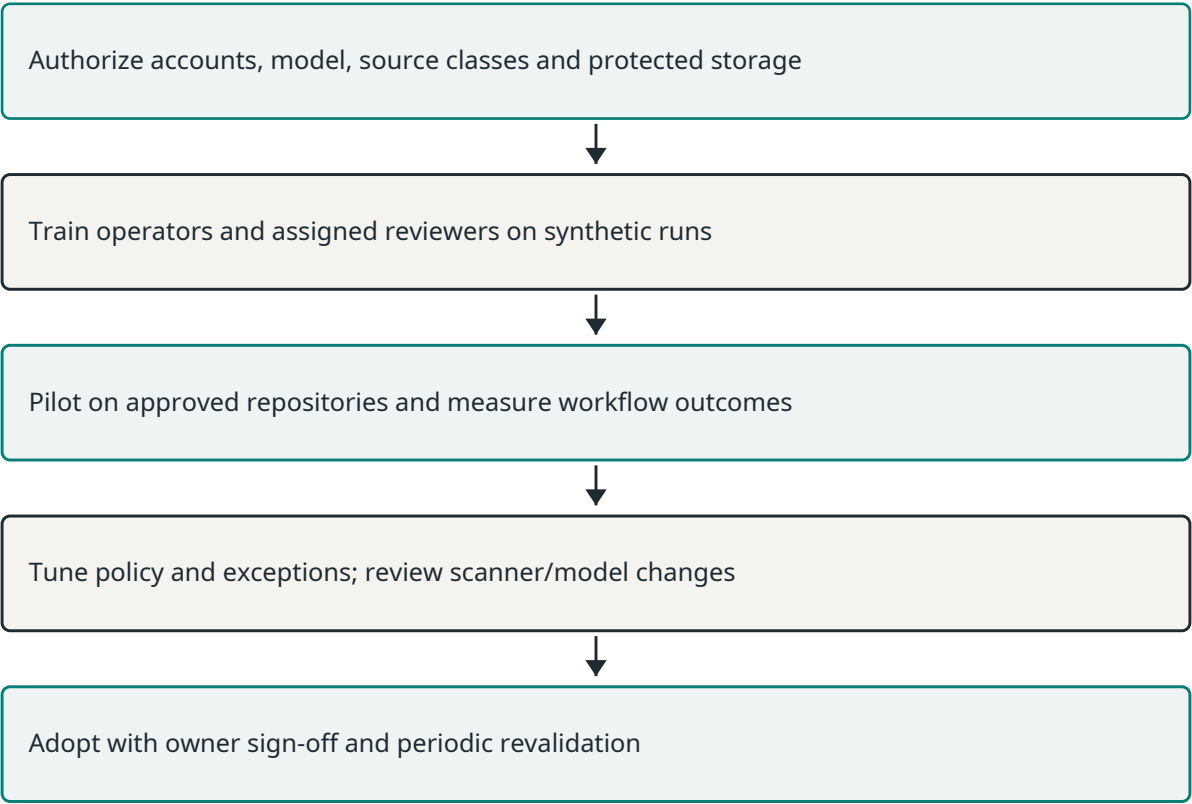


Figure 4. Proposed adoption stages with explicit human gates.

Table 4. Pilot measures to collect, not performance claims

Measure	Collection method	Decision owner
Completed versus INCOMPLETE	Count run states and causes	Tool owner
Reviewer effort	Time from handoff to recorded decision	Review lead
Confirmed versus rejected hypotheses	Human decision log with reasons	Security lead
Source-transfer exceptions	Approval and incident records	Data owner
Rule / model drift	Re-run controlled fixtures after updates	Tool owner

Document an exception owner, scope, expiry, and compensating check outside the tool. Revalidate after changes to scanner databases, pinned tools, model availability, policy, or source-transfer terms. Do not silently turn a failed required stage into a successful review.

08 / Limits and sources

What this conclusion does and does not establish

Practice survey checked on 2026-09-23. NIST SSDF 1.1 is final; the 1.2 revision is draft, so this methodology does not present it as a binding final standard. OWASP and Anthropic pages can change; re-check them when updating company policy. [S1] [S2]

Residual risks include unsupported languages or manifests, scanner false positives and false negatives, model error and prompt injection, unknown secrets in the packet, changes in vulnerability databases, unsigned evidence, storage tampering, and reviewer judgment. Record the exact scope and omissions. A clean run never certifies a real repository or approves a merge. [S5] [S6]

Use the paired CommitScope User Guide for the employee and reviewer walkthrough. The original protected run plus a separate human decision is the corporate evidence path; CI scanner reports alone are partial evidence.

Sources and verification date

[S1] NIST SP 800-218, SSDF 1.1 (final, 2022). <https://csrc.nist.gov/pubs/sp/800/218/final> (2026-09-23)

[S2] NIST SSDF publications: SP 800-218 Rev. 1 / SSDF 1.2 (draft).
<https://csrc.nist.gov/projects/ssdf/publications> (2026-09-23)

[S3] OWASP SAMM: model and business functions. <https://owasp samm.org/about/> (2026-09-23)

[S4] OWASP Application Security Verification Standard 5.0.0. <https://owasp.org/projects/asvs> (2026-09-23)

[S5] OWASP LLM01:2025 Prompt Injection. <https://genai.owasp.org/llmrisk/llm01-prompt-injection/> (2026-09-23)

[S6] Anthropic Claude Code: Security. <https://code.claude.com/docs/en/security> (2026-09-23)

[S7] Anthropic Claude Code: Data usage. <https://code.claude.com/docs/en/data-usage> (2026-09-23)

[S8] Anthropic Claude Code: Authentication. <https://code.claude.com/docs/en/iam> (2026-09-23)

[S9] OWASP Web Security Testing Guide 4.2. <https://wstg.owasp.org/v4.2/> (2026-09-23)