GH-500 // FINAL REVIEW GUIDE

CODE / SECRETS / SUPPLY CHAIN / OPERATIONS

Suite boundaries, prevention controls, alert workflows, CodeQL/SARIF, governance, abbreviations, and current limits.

01 // SUITES + ADMIN // D1 15-20% / D6 10-15%

PRODUCT BOUNDARIES // PICK THE CONTROL

GHAS	Umbrella for paid Secret Protection and Code Security products; many dependency features remain broadly available.
SECRET	Detect and prevent leaked credentials: secret scanning, push protection, validity checks, custom patterns, delegated bypass/dismissal, campaigns.
CODE	Find and fix code risk: CodeQL/third-party scanning, CLI, SARIF, autofix, dependency review, premium Dependabot controls, campaigns, overview.
SUPPLY	Dependency graph, Advisory Database, Dependabot alerts and updates, dependency review, SBOMs, attestations, release integrity.

AVAILABILITY // PUBLIC IS NOT PRIVATE

- Public repositories receive several security features without a paid product; enablement still varies by feature.
- Organization-owned private/internal repositories need the relevant product on GitHub Team or Enterprise Cloud. GHES support is version-specific.
- Do not infer that buying one suite enables the other. Confirm repository type, plan, product, platform, and feature state.

CONTROL PLANE // SCOPE + INHERITANCE

CONFIG	Repository-level feature settings bundled and applied by an organization or enterprise.
GLOBAL	Organization-wide analysis behavior inherited by repositories, such as private registries or CodeQL expansion.
STATE	Attached inherits the configuration. Unenforced settings may be overridden or detached; enforced settings cannot be changed below their owner.

- Enterprise policy sets the outer boundary; organization owners configure coverage; repository settings apply within that boundary.
- Use configurations and APIs for repeatable rollout. Check failures, removed/detached repositories, coverage gaps, and newly created repositories.

ROLES // LEAST PRIVILEGE

- Owners and security managers can view organization-wide security data. Repository admins configure local features; alert access can be delegated with custom roles.
- Delegated bypass and delegated alert dismissal separate the developer request from the reviewer decision. Preserve the reason and audit trail.

02 // SECRETS + SUPPLY CHAIN // D2 15-20% / D3 15-20%

SECRET RESPONSE // CREDENTIAL FIRST

- PREVENT -> DETECT -> VALIDATE -> REVOKE/ROTATE -> REMOVE -> CLOSE
- Treat a committed secret as exposed. Revoke or rotate it before cleanup; deleting a line or rewriting history does not invalidate the credential.
- Confirm provider, secret type, validity, locations, repository visibility, access scope, exposure window, and use. Then contain and document.

USER	Repository alert for a supported or custom pattern.
PUSH	Repository alert when push protection is bypassed.
PARTNER	Sent directly to a participating provider for a public leak; not shown as a repository user alert.

PUSH PROTECTION + CUSTOM DETECTION

- Push protection blocks a supported secret before exposure. A bypass is not remediation; choose a documented reason or delegated approval only when justified.
- Validity and extended metadata checks improve prioritization. Active does not prove use; inactive does not erase exposure history.
- Custom patterns use Hyperscan-compatible regular expressions. Test before publishing; add boundaries and positive/negative requirements to control noise.
- Exclusions automatically close matching-path alerts; use them narrowly and review their security effect.

DEPENDENCY FLOW // GRAPH BEFORE ALERT

GRAPH	Manifest/lock parsing plus dependency submission; tracks direct and transitive dependencies. Lock files give more exact versions.
ALERT	Graph data matched against GitHub Advisory Database vulnerabilities or malware.
SEC UPD	Alert-triggered pull request to the minimum secure version.
VER UPD	Scheduled pull request toward newer versions; requires dependabot.yml and is not vulnerability-triggered.
REVIEW	Pull-request diff for added/removed/updated dependencies, vulnerabilities, licenses, age, and policy gates before merge.
SBOM	Export the graph as SPDX-compatible inventory; a list of components is evidence, not a vulnerability verdict.

ADVISORY DATA // DO NOT MIX THE SIGNALS

- A repository alert is an affected-project finding. A security advisory describes a vulnerability or malware record and affected/fixed ranges.
- CVE identifies a vulnerability; CWE classifies a weakness; CVSS estimates severity; EPSS estimates exploitation likelihood; GHSA identifies a GitHub advisory.

03 // CODE SECURITY + CODEQL // D4 10-15%

SETUP CHOICE // LOW MAINTENANCE OR CONTROL

DEFAULT	GitHub-managed configuration. Starts from detected supported languages; customize languages, suite, runner, models, and config-file property without a workflow file.
ADVANCED	GitHub Actions workflow for explicit triggers, language matrix, build commands, query packs/suites, categories, runners, and schedule.
EXTERNAL	Run CodeQL CLI or a third-party SAST tool in external CI, then upload supported SARIF to GitHub code scanning.

- Choose default setup unless the required build, runner, query, trigger, or external-CI control exceeds its customization surface.
- Default setup requires Actions, no conflicting advanced setup, and public visibility or Code Security enablement.

CODEQL PIPELINE // DATA FLOW

- SOURCE -> EXTRACT/BUILD -> DATABASE -> QUERIES -> SARIF -> ALERT
- A source introduces untrusted data; propagation carries taint/data; a sanitizer neutralizes it; a sink performs a security-sensitive operation.
- Path queries explain flow from source to sink. Review steps, assumptions, language/framework models, and whether the path is reachable.

NONE	Create the database without a build where supported; also used for interpreted languages.
AUTOBUILD	CodeQL chooses a supported build path; self-hosted runners need the build tooling.
MANUAL	Provide exact build commands when generated code, variants, or custom tooling determine coverage.

QUERIES + RESULTS // PRECISION TRADEOFF

DEFAULT	High-precision built-in suite with fewer false positives.
EXTENDED	Default plus lower-precision/severity queries; broader coverage and more review work.
CUSTOM	QL packs and .qls suites; custom suites require advanced setup. Pin pack versions for reproducibility.

- Use categories/analysis origins to keep results from different languages, builds, tools, or configurations distinct.
- Alert details include rule, location, severity/security severity, path, branches, and first introduction. Default-branch state is canonical.
- Autofix is a proposed code change: review the explanation, tests, side effects, and residual paths before merge.

TROUBLESHOOT // COVERAGE BEFORE GREEN

- Check language detection, source roots, generated/vendor exclusions, build success, permissions, Actions policy, runner capacity, private registries, upload category, and SARIF size/noise.
- A successful workflow can still analyze too little code. Verify extracted source and expected alerts, not only a green job.

04 // SECURITY OPERATIONS // D5 15-20%

PRIORITIZE // CONTEXT BEATS ONE NUMBER

- Combine exploitability, reachability, production/deployment context, business criticality, exposure, severity, asset ownership, and fix availability.
- CVSS answers impact/severity. EPSS answers estimated exploitation likelihood. Neither proves reachability, compromise, or business impact.
- Prefer prevention and pre-merge controls, then use overview, filters, ownership, campaigns, autofix, and APIs to burn down existing risk.

ALERT LIFECYCLE // EVIDENCE + OWNERSHIP

- TRIAGE -> ASSIGN -> VALIDATE -> FIX/ROTATE -> VERIFY -> CLOSE
- Record why an alert is fixed, dismissed, revoked, used in tests, false positive, or accepted. Delegated dismissal keeps review independent.
- Security campaigns group focused code or secret alerts across repositories. Give managers, developers, deadline, guidance, and a measurable exit condition.
- Audit configuration changes, bypasses, dismissals, notifications, API automation, campaign progress, and repositories outside coverage.

NUMBERS // CURRENT OFFICIAL LIMITS

PATTERN	500 custom patterns per organization/enterprise; 100 per repository.
EPSS	0-1 probability (0-100%) for exploitation in the next 30 days; percentile is relative rank.
DEFAULT	Scan pushes to default/protected branches, pull requests targeting them except forks, and weekly.
SARIF	Version 2.1.0 subset; gzip file <=10 MB; <=25,000 results/run; top 5,000 displayed by severity.
CAMPAIGN	Up to 1,000 alerts per campaign; up to 10 active campaigns.

DECISION TRAPS // NAME THE CONTROL POINT

- Leaked credential -> Secret Protection. Vulnerable dependency -> graph/Dependabot. Vulnerable code path -> code scanning/CodeQL.
- Bypass permits a blocked push; dismissal changes alert disposition. Neither revokes a credential or fixes code.
- Security configuration controls repository feature settings; global settings control organization-wide analysis behavior.
- Default setup optimizes maintenance; advanced setup optimizes control. External SARIF imports findings but does not make the tool CodeQL.
- Security update targets the minimum patched version; version update follows the configured update strategy.
- Closing an alert is workflow state. Verify the underlying secret, dependency, or code risk is actually removed.

ABBREVIATIONS

GHAS GitHub Advanced Security
GHEC GitHub Enterprise Cloud
GHES GitHub Enterprise Server
SDLC software development life cycle
CI/CD continuous integration/delivery
RBAC role-based access control
SAST static application security testing
SCA software composition analysis
SARIF Static Analysis Results Interchange Format
SBOM software bill of materials
SPDX Software Package Data Exchange
PAT personal access token

CVE vulnerability ID | CWE weakness class | CVSS severity | EPSS exploitation probability | GHSA GitHub Security Advisory ID

SOURCES: [1] study guide [2] products [3] secrets [4] supply chain [5] CodeQL [6] SARIF [7] advisories [8] campaigns