Study Plan & User Stories — GitHub Advanced Security (GHAS)

January 19, 2026

Contents

1 How to Use This Document

This document is a polished, standalone template for GHAS study planning using user stories. Each backlog item is rendered as a *story card* followed by a concrete *tasks* checklist. Duplicate a card for each item you want to track. Fields are intentionally concise and testable.

Writing Effective User Stories

Use this formula:

> *As a [**persona**], I want to [**do/achieve**], so that [**business outcome**].*

Good stories describe *one* valuable behavior, include acceptance criteria (BDD style), and tie to observable outcomes. Avoid implementation detail in the story—put it in tasks. Keep estimates small (1–5 SP).

Examples

- **Good:** *As an org admin, I want to enforce security checks via rulesets so that all PRs are gated on CodeQL and secret scanning.*
- **Good:** *As a security engineer, I want to author a custom CodeQL query pack so that we detect org-specific sinks.*
- **Anti-pattern:** *Set up all of GHAS this quarter.* (too broad, no persona, no outcome)

Non-Functional Tags

Use badges to call out cross-cutting concerns: Performance Security Reliability Accessibility Privacy i18n.

Prerequisites Checklist

- Admin access to a GitHub Enterprise/Team organization with GHAS licenses.
- Sample repositories (at least one compiled language project).
- Ability to create org & repo *rulesets*, enable security features, and view *Security overview*.

2 Study Roadmap (8 Weeks)

Each week is one primary story card (with BDD acceptance criteria) and a task checklist. Adjust estimates and personas to fit your context.

GHAS-1 — Foundations & Governance

Epic / Feature	Program Foundations / Org Governance
Business Value	Establish shared understanding of GHAS, fast feedback, and "keep main green" to reduce risk.
Priority / Estimate	Priority: Must SP: 3
Persona	Org admin / platform engineer
Dependencies	Test organization and 3 seed repositories
Assumptions / Risks	Time to enable features varies by repo; risk of noisy alerts initially

Story *As an org admin, I want to enable GHAS foundations and configure repository* ***rulesets*** *so that PRs are gated on security checks and the org baseline is measurable.*

Non-Functional Security Reliability Privacy

Acceptance Criteria (BDD)

Scenario	Happy path
Given	a sandbox org with 3 repos and permissions to manage security settings
When	rulesets and GHAS features are enabled per policy
Then	PRs require CodeQL and secret scanning checks; Security overview shows baseline metrics

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; tests green; security/a11y checks; docs updated; deployed/flagged.

- ☐ Enable on 3 repos: Dependency graph, Dependabot alerts/updates, secret scanning, code scanning (default setup).
- ☐ Create org rulesets enforcing: required checks (CodeQL, secret scanning), linear history, signed commits.
- ☐ Configure branch protections on `main` & `release/*`; block force-push and direct commits.
- ☐ Capture baseline in Security overview: open alerts by type, age > 30 days.
- ☐ Document governance in the platform handbook.

GHAS-2 — Code Scanning with CodeQL (Essentials)

Epic / Feature	Code Scanning
Business Value	Detect high-impact vulnerabilities early; create PR-gated signal.
Priority / Estimate	Priority: Must SP: 5
Persona	Security engineer / repo maintainer
Dependencies	GHAS-1 completed; languages identified
Assumptions / Risks	False positives must be triaged; build steps for compiled languages may require caching

Story *As a security engineer, I want to configure CodeQL default setup and PR checks so that critical issues are caught before merge.*

Non-Functional Security Reliability

Acceptance Criteria (BDD)

Scenario	Happy path
Given	repositories with CodeQL enabled
When	a PR introduces a vulnerable pattern
Then	the PR check fails, an alert is created, and triage notes are recorded

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; tests green; security/a11y checks; docs updated; deployed/flagged.

- ☐ Turn on *Default setup* for 3 repos; verify first analysis completes.
- ☐ Add schedules (nightly) and enable PR-only analysis for long builds.
- ☐ Define triage workflow: labels, assignees, SLAs; close or suppress top 10 alerts with justifications.
- ☐ Export SARIF from one run and archive in the security evidence folder.

GHAS-3 — CodeQL Deep Dive: CLI, Databases, Custom Queries

Epic / Feature	CodeQL Query Authoring
Business Value	Detect org-specific anti-patterns and reduce MTTR with precise alerts.
Priority / Estimate	Priority: Should SP: 8
Persona	Security engineer
Dependencies	GHAS-2; local dev environment for CodeQL CLI
Assumptions / Risks	Large projects may require extraction tuning; query quality must be validated

Story *As a security engineer, I want to author and ship a custom CodeQL query pack so that our repos detect org-specific vulnerabilities.*

Non-Functional Security Reliability Performance

Acceptance Criteria (BDD)

Scenario	Query pack in CI
Given	a CodeQL database for a compiled-language repo
When	a custom query identifies a tainted flow to a dangerous sink
Then	CI fails with a clear alert and remediation guidance

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; tests green; security/a11y checks; docs updated; deployed/flagged.

- ☐ Install CodeQL CLI; generate a local database for one compiled repo.
- ☐ Write one custom QL query; validate with unit tests and `codeql test`.
- ☐ Package queries into a query pack; reference it from the CodeQL workflow.
- ☐ Demonstrate SARIF upload from CLI; document process in handbook.

GHAS-4 — Secret Scanning & Push Protection

Epic / Feature	Secret Scanning
Business Value	Prevent leaked credentials from entering history; speed incident response.
Priority / Estimate	Priority: Must SP: 5
Persona	Platform engineer / repo maintainer
Dependencies	GHAS-1
Assumptions / Risks	Exclusions required for test data; bypass governance must be defined

Story *As a platform engineer, I want to enable secret scanning with push protection so that high-confidence secrets are blocked before commit.*

Non-Functional Security Reliability Privacy

Acceptance Criteria (BDD)

Scenario	Blocked push
Given	push protection enabled on 3 repos
When	a developer attempts to push a simulated token
Then	the push is blocked; bypass requires justification and is auditable

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; tests green; security/a11y checks; docs updated; deployed/flagged.

☐ Enable secret scanning and push protection on 3 repos.

☐ Add `secret_scanning.yml` to exclude noisy paths (e.g., test fixtures).

☐ Simulate a blocked push with a dummy token; capture the developer UX and audit event.

☐ Define delegated bypass roles and documentation.

GHAS-5 — Supply Chain: Dependabot, Advisories, PVR

Epic / Feature	Supply Chain Security
Business Value	Reduce exposure from vulnerable dependencies; handle inbound reports securely.
Priority / Estimate	Priority: Should SP: 5
Persona	Security engineer / maintainer
Dependencies	GHAS-1
Assumptions / Risks	Update noise; coordination required for coordinated disclosure

Story *As a maintainer, I want Dependabot updates and Private Vulnerability Reporting so that we remediate CVEs quickly and accept reports responsibly.*

Non-Functional Security Reliability

Acceptance Criteria (BDD)

Scenario	Weekly updates
Given	Dependabot alerts & updates enabled on study repos
When	critical advisories exist
Then	grouped PRs are raised and merged within SLA; PVR workflow is validated end-to-end

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; tests green; security/a11y checks; docs updated; deployed/flagged.

☐ Configure `dependabot.yml`: weekly schedule, grouped minor bumps, auto-merge for safe updates.

☐ Enable Private Vulnerability Reporting; publish one test advisory and triage to closure.

☐ Build a remediation dashboard: open alerts, aging, MTTR.

GHAS-6 — Org Reporting & Workflow Hardening

Epic / Feature	Security Overview & Actions Hardening
Business Value	Drive remediation through metrics; protect CI from supply-chain risks.
Priority / Estimate	Priority: Should SP: 5
Persona	Security program owner / platform engineer
Dependencies	GHAS-1..5
Assumptions / Risks	Fork PRs need safe permissions; action pinning reduces risk but needs maintenance

Story *As a program owner, I want org-level dashboards and hardened workflows so that leaders see progress and CI remains trustworthy.*

Non-Functional Security Reliability Performance

Acceptance Criteria (BDD)

Scenario	Dashboard-driven remediation
Given	Security overview with feature adoption metrics
When	teams review weekly
Then	MTTR for High/Critical < 7 days; adoption > 90% on target repos

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; tests green; security/a11y checks; docs updated; deployed/flagged.

☐ Build an adoption scorecard: feature enablement %, alert MTTR, backlog trend.

☐ Harden Actions: least-privilege tokens, OIDC to cloud, pin actions by SHA, required checks on protected branches.

☐ Create an incident runbook: secret exfiltration, vulnerability disclosure, CodeQL regression.

GHAS-7 — Capstone: End-to-End Implementation

Epic / Feature	Capstone
Business Value	Prove value on a production-like repo; socialize rollout approach.
Priority / Estimate	Priority: Must SP: 8
Persona	Security engineer / repo owner
Dependencies	GHAS-1..6
Assumptions / Risks	Coordination with repo owners; change management for required checks

Story *As a repo owner, I want an end-to-end GHAS setup so that our main branch stays clean and secure.*

Non-Functional Security Reliability Privacy

Acceptance Criteria (BDD)

Scenario	E2E success
Given	a target repo
When	rulesets, CodeQL (with custom pack), secret scanning w/ push protection, Dependabot, and PVR are configured
Then	PRs are gated; main has zero critical alerts; dashboard reflects improvements

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; tests green; security/a11y checks; docs updated; deployed/flagged.

- □ Apply all security features and rulesets to the capstone repo.
- □ Integrate the custom CodeQL query pack; verify failing PR then fix and re-run.
- □ Demo results and metrics to stakeholders; capture lessons learned.

GHAS-8 — Rollout Plan & (Optional) Certification

Epic / Feature	Program Rollout
Business Value	Scale GHAS across the org; validate skills via certification.
Priority / Estimate	Priority: Should SP: 3
Persona	Program owner
Dependencies	GHAS-7
Assumptions / Risks	Team readiness varies; certification optional

Story *As a program owner, I want a 90-day rollout and training plan so that GHAS adoption is consistent and measurable.*

Non-Functional Security Reliability

Acceptance Criteria (BDD)

Scenario	Rollout approved
Given	a pilot completed and metrics available
When	the 90-day rollout plan is reviewed
Then	leadership signs off; training & enablement assets are published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; tests green; security/a11y checks; docs updated; deployed/flagged.

- ☐ Create a 90-day rollout plan: scope, milestones, enablement sessions, metrics.
- ☐ Prepare a GHAS playbook: setup steps, ruleset recipes, CodeQL pack usage, secret scanning patterns, PVR guide.
- ☐ (Optional) Schedule the GitHub Advanced Security certification after a passing practice exam.

3 Appendix: Quick Reference

Story Template *As a [persona], I want to [goal], so that [business outcome].*

Acceptance Criteria Use Given/When/Then with observable outcomes. Cover happy and negative paths. Include data boundaries and permissions.

Definitions **Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. **Definition of Done:** All ACs pass; tests green; security/a11y checks; docs updated; deployed/flagged.