Git for Security Professionals

SEC204

Version control for the work security teams actually produce: detection rules, IR playbooks, scripts, policy-as-code, and compliance evidence. Put any artifact under Git with a full audit trail of who changed what and why, collaborate safely through branches and review, keep secrets out of the history and purge them when they get in, sign your work for provenance, and automate the checks that catch mistakes before they ship. You finish able to run version control as part of the craft, across detection, DFIR, architecture, and governance.

Modules	14
Lessons	63
Phases	5
Free preview	Course Orientation

The method

Security work produces artifacts that need history: detection rules, policies, playbooks, infrastructure definitions. This course teaches Git as the tool that gives them a provable audit trail, and runs the same loop for every workflow it covers.

1. Understand what Git is recording. Commits, trees and refs, before any command. Most Git confusion is a mental model problem rather than a syntax one.

2. Work in small, reviewable changes. A commit that does one thing can be reverted, reviewed and explained. A commit that does nine cannot.

3. Branch for anything that might not ship. Detection rules that need testing, policies awaiting sign-off. The branch is what makes the work safe to abandon.

4. Know how to recover. Reflog, reset, revert and the difference between them. The value of version control is entirely in the recovery, and the recovery is what nobody practices.

5. Sign what matters and automate the checks. Signed commits and CI validation turn a repository into evidence somebody else can rely on.

What this course assumes

No minimum experience and no prerequisite course. This course assumes you have never used Git, and it assumes you are not a software engineer.

What makes it go faster: comfort at a command line and something of your own worth versioning, a rule set or a policy pack. Neither is required.

What this course does not cover: software development, application security, and CI/CD as a build discipline. This is version control for security artifacts and the automation directly around them.

Full syllabus

Course Orientation

Course Orientation FREE PREVIEW

What Git for Security Professionals teaches: give your security work a memory.

Phase 1: Foundations

Git Foundations

The foundation the whole course stands on.

1. 1.1 Your First Repository
2. 1.2 The Four-Stage Model
3. 1.3 What a Commit Actually Is
4. 1.4 Reading History
5. 1.5 Working with a Remote
6. 1.6 A Repository for Security Work
7. Module Summary
8. Check My Knowledge

The Daily Workflow

Turn the foundations into fluency. Run the daily edit-stage-commit-sync loop as a discipline, write commit messages that hold up under review, stage selectively so every commit is one clean change, undo safely whichever way you got it wrong, stash and clean a working tree, and configure Git so...

1. 2.1 The Daily Loop
2. 2.2 Commit Messages That Hold Up
3. 2.3 Staging with Intent
4. 2.4 Undoing Safely
5. 2.5 Ignoring, Cleaning, and Stashing
6. 2.6 Making the Workflow Yours
7. Module Summary
8. Check My Knowledge

Phase 2: Collaboration

Branching and Collaboration

Work alongside a team without collisions.

1. 3.1 What a Branch Is
2. 3.2 Working on a Branch
3. 3.3 Merging
4. 3.4 Merge vs Rebase
5. 3.5 Collaborating Through a Remote
6. 3.6 The Pull Request Workflow
7. Module Summary
8. Check My Knowledge

Conflicts, History, and Recovery

Handle the situations collaboration produces.

1. 4.1 Understanding Merge Conflicts
2. 4.2 Conflicts in Practice
3. 4.3 Rewriting History with Interactive Rebase
4. 4.4 Rewriting History Safely
5. 4.5 Recovery with the Reflog
6. 4.6 Finding a Bad Commit with git bisect
7. Module Summary
8. Check My Knowledge

Phase 3: Securing Git

Securing Git

Secure the repository itself. Keep secrets out of history with ignore rules and the right habits, catch the ones that slip through with secret-scanning hooks and push protection, respond correctly when a secret leaks by rotating before you purge, prove who made a commit with signing, and enforce...

1. 5.1 Keeping Secrets Out of a Repository
2. 5.2 Secret Scanning and Pre-commit Hooks
3. 5.3 Responding to a Leaked Secret
4. 5.4 Verifying Commit Authorship with Signing
5. 5.5 Protecting Branches and the Repository
6. 5.6 Access, Credentials, and Repository Hygiene
7. Module Summary
8. Check My Knowledge

Phase 4: Automation and Capstone

CI/CD for Security Work

Turn a version-controlled security repository into a pipeline that ships.

1. 6.1 What CI/CD Is and Why It Matters for Security Work
2. 6.2 Your First Pipeline: Validating a Change
3. 6.3 Testing Security Content in the Pipeline
4. 6.4 Deploying Automatically
5. 6.5 Securing the Pipeline Itself
6. 6.6 Capstone: The Whole Workflow End to End
7. Module Summary
8. Check My Knowledge

Phase 0: Course Resources

Cheatsheets

The commands that matter, what each one actually moves, how to recover, and the security work Git makes possible.

1. The Model and the Daily Loop
2. Branching, Merging and Getting Back
3. Keeping Secrets Out, and What to Do When One Gets In
4. Pipelines for Security Content

Cookbooks
Ordered procedures for standing up a content repository, reviewing a change, responding to a leak, and recovering from a bad rewrite.

1. Standing Up a Security Content Repository
2. Responding to a Committed Credential
3. Reviewing a Detection Change
4. Recovering From a Bad Rewrite

Walkthroughs
Four cases reasoned end to end, including the green pipeline that deployed nothing and the commit signed by nobody.

1. The Pipeline That Deployed Nothing
2. The Commit Nobody Wrote
3. The Key That Had Been There Two Years
4. The Afternoon That Disappeared

Playground
A throwaway repository, a free host account, and the four experiments worth running deliberately.

Operational Reference
Every command, workflow, and pipeline snippet from the course in one searchable reference, organized by task, from daily workflow and history recovery to signing, branch protection, and CI/CD.

1. GT99.1 Operational Reference

References & Further Reading
Git documentation, commit-signing references, platform docs for branch protection and CI/CD, secret-scanning and history-rewriting tools, supply-chain security resources, and secure-development standards used throughout the Git for Security Professionals course.

1. GT100.1 References & Further Reading

Course Completion

Course Exam
Git for Security Professionals end-of-course exam: a committed credential reported as remediated because the file was deleted, testing whether you know the difference between a working tree and a history and what the only real remediation is.

1. Course Completion. Git for Security Professionals

Generated 2026-09-01 from the published course. The current syllabus is always at ridgelinecyber.com/training/courses/git-for-security-professionals/