User Stories by Chapter:
Application Security Program Guide

Compiled for Jordan Suber

Contents

How to Use This Template

Each card maps one chapter's *Learning Goals* to a concise story, binds the chapter's *Hands-on Objectives* to concrete *Tasks*, and verifies *Outcomes* via BDD-style Acceptance Criteria. Import these cards into your backlog, tag by risk tier, and iterate.

Required Data on Every Story

- **ID** (e.g., APPSEC-1), **Title** (actionable verb), **Epic/Feature**, **Business Value** (outcome/why)
- **Priority** (Must/Should/Could), **Estimate** (SP), **Persona**, **Dependencies**, **Assumptions/Risks**
- **Acceptance Criteria** (Gherkin-ish BDD), **Tasks** (checklist), **NFR** (Security, Privacy, Reliability, etc.)

Writing Effective User Stories (Quick Guide)

Template: As a *[persona]*, I want to *[do X]* so that *[value/why]*.
INVEST: Independent, Negotiable, Valuable, Estimable, Small, Testable.
Good: "As an AppSec lead, I want a *tiered SSDLC policy* so that *teams ship securely with minimal friction.*"
Anti-patterns: Vague "Research X"; multi-team mega-stories; outputs without value ("create doc") unless tied to decision/change.

1 Stories by Chapter

APPSEC-1 — Publish an AppSec Program Charter

Epic / Feature	Program Foundations
Business Value	align engineering, product, and risk on scope, value, and success criteria
Priority / Estimate	Priority: Must SP: 3
Persona	AppSec lead
Dependencies	Org strategy, security policy, product roadmap
Assumptions / Risks	Scope creep risk; time-box charter v1 and plan iterative updates

Story *As a AppSec lead, I want to Publish an AppSec Program Charter so that align engineering, product, and risk on scope, value, and success criteria.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

□ Draft a one-page charter: mission, scope, definitions, interfaces, success metrics.

□ Create a stakeholder map and RACI for threat modeling, testing, vuln mgmt, IR.

□ Review with Eng/Product/Risk; capture decisions and open questions.

□ Publish in the handbook repo; version as living document.

APPSEC-2 — Create a Control Dictionary & Traceability Matrix

Epic / Feature	Security Foundations
Business Value	give engineers clear, shared definitions and connect policies to app controls
Priority / Estimate	Priority: Must SP: 5
Persona	Security architect
Dependencies	Enterprise policies/standards
Assumptions / Risks	Terminology mismatch; include concrete code/config examples

Story *As a Security architect, I want to Create a Control Dictionary & Traceability Matrix so that give engineers clear, shared definitions and connect policies to app controls.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repositories, environments, and program context are available

When the *Hands-on Objectives* for this chapter are executed

Then the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Compile key concepts (authn, authz, logging, crypto, secrets, input validation).
- ☐ Map each enterprise policy to concrete application controls and test evidence.
- ☐ Add links to code samples, lints, and CI checks for each control.
- ☐ Publish as `/docs/control-dictionary.md` and keep PR-able.

APPSEC-3 — Build an Application Inventory & Tiering

Epic / Feature	Program Scope
Business Value	focus effort on highest-risk apps; enable tiered controls and SLAs
Priority / Estimate	Priority: Must SP: 5
Persona	Product security engineer
Dependencies	CMDB/source of truth; service catalog
Assumptions / Risks	Owner gaps; require ownership to promote to higher envs

Story *As a Product security engineer, I want to Build an Application Inventory & Tiering so that focus effort on highest-risk apps; enable tiered controls and SLAs.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Inventory apps/services/APIs with owners, data classes, exposure, tech stack.
- ☐ Define tiering model (e.g., P0–P3) with criteria and examples.
- ☐ Record lifecycle (active/sunset), compliance drivers, and repo links.
- ☐ Export registry to CSV/JSON; integrate with CI labels per repo.

APPSEC-4 — Stand Up an App Risk Register

Epic / Feature	Risk Management
Business Value	turn threats into tracked items tied to owners, dates, and treatments
Priority / Estimate	Priority: Must SP: 3
Persona	Risk manager
Dependencies	Inventory completed, risk rubric
Assumptions / Risks	Over-long registers stall; keep to top risks per app

Story *As a Risk manager, I want to Stand Up an App Risk Register so that turn threats into tracked items tied to owners, dates, and treatments.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- □ Define likelihood/impact rubric and treatment options.
- □ Run a 60–90 min risk workshop for two critical apps.
- □ Create entries with owner, due date, and linkage to epics/stories.
- □ Establish intake workflow (new risk → triage → acceptance).

APPSEC-5 — Publish Secure Reference Architectures

Epic / Feature	Secure Design Patterns
Business Value	give teams golden paths that bake in zero-trust and least privilege
Priority / Estimate	Priority: Should SP: 5
Persona	Security architect
Dependencies	Architecture council, platform patterns
Assumptions / Risks	Architecture drift; add linters/policies to reinforce

Story *As a Security architect, I want to Publish Secure Reference Architectures so that give teams golden paths that bake in zero-trust and least privilege.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

☐ Diagram monolith, microservices, async/event-driven, and serverless patterns.

☐ Annotate controls per tier (authn, mTLS, input validation, logging, backups).

☐ Provide IaC/app templates implementing the patterns.

☐ Add “choose-by-facts” table and decision records (ADRs).

APPSEC-6 — Adopt a Tiered SSDLC Policy

Epic / Feature	SSDLC Alignment
Business Value	embed right-sized checks by risk tier to shift left without friction
Priority / Estimate	Priority: Must SP: 5
Persona	AppSec lead
Dependencies	Engineering buy-in, CI access
Assumptions / Risks	Over-gating; start minimal and ratchet

Story *As a AppSec lead, I want to Adopt a Tiered SSDLC Policy so that embed right-sized checks by risk tier to shift left without friction.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Define controls per SDLC phase and per tier (ASVS/SSDF-aligned).
- ☐ Wire required checks in CI (lint, SAST, SCA) with pass/fail thresholds.
- ☐ Add DoD/DoR updates to team templates referencing security checks.
- ☐ Document exceptions/waivers with expiry and approval path.

APPSEC-7 — Launch the AppSec Champions Program

Epic / Feature	Operating Model & Teams
Business Value	scale AppSec via embedded advocates and faster issue resolution
Priority / Estimate	Priority: Should SP: 3
Persona	AppSec lead
Dependencies	Managers' support, time allocation
Assumptions / Risks	Attrition/adoption risk; include incentives and community time

Story *As a AppSec lead, I want to Launch the AppSec Champions Program so that scale AppSec via embedded advocates and faster issue resolution.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Define selection rubric, responsibilities, and incentives.
- ☐ Create monthly office hours and a champions Slack channel.
- ☐ Provide starter kit (checklists, threat modeling kit, PR review guide).
- ☐ Track participation and outcomes (bugs prevented, PRs reviewed).

APPSEC-34 — Define Security Definition of Ready (DoR)

Definition of Done (DoD)

Epic / Feature	Agile AppSec Foundations
Business Value	bake security into the team's workflow gates so features ship with baseline controls
Priority / Estimate	Priority: Must SP: 3
Persona	Scrum Master
Dependencies	Agreed SSDLC policy; team working agreement
Assumptions / Risks	Too heavy gates can slow delivery; right-size to risk tiers

Story *As a Scrum Master, I want to Define Security Definition of Ready (DoR)*

Definition of Done (DoD) so that bake security into the team's workflow gates so features ship with baseline controls.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

- ☐ Document security DoR (threat model link, acceptance criteria, risk score)
- ☐ Document security DoD (tests green, SBOM present, secrets scan clean)
- ☐ Publish team board checklists and automate reminders

Acceptance Criteria

- ☐ Security DoR/DoD approved and referenced in sprint templates
- ☐ PR template includes security checklist
- ☐ Pipeline enforces key DoD checks (fail on critical issues)

APPSEC-35 — Create a Security Acceptance Criteria Library

Epic / Feature	Agile AppSec Foundations
Business Value	accelerate secure delivery by reusing well-formed security ACs per story type
Priority / Estimate	Priority: Must SP: 5
Persona	Product Owner
Dependencies	Secure coding standards; ASVS mapping
Assumptions / Risks	Generic ACs may not fit; allow tailoring per risk tier

Story *As a Product Owner, I want to Create a Security Acceptance Criteria Library so that accelerate secure delivery by reusing well-formed security ACs per story type.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repositories, environments, and program context are available

When the *Hands-on Objectives* for this chapter are executed

Then the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

□ Curate AC snippets for auth, input validation, logging, PII handling

□ Map ACs to ASVS controls and risk tiers

□ Add AC snippets to backlog templates and story examples

Acceptance Criteria

□ AC library lives in repo/Wiki and is referenced by >80% of new stories

□ Each AC mapped to ASVS section and test evidence type

APPSEC-36 — Stand Up a Security Backlog

Risk Triage Kanban

Epic / Feature	Agile AppSec Operations
Business Value	ensure visibility and flow for security work alongside product features
Priority / Estimate	Priority: Must SP: 5
Persona	Product Security Engineer
Dependencies	Control dictionary; risk register
Assumptions / Risks	Security items may be starved; set WIP and capacity policies

Story *As a Product Security Engineer, I want to Stand Up a Security Backlog*

Risk Triage Kanban so that ensure visibility and flow for security work alongside product features.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repositories, environments, and program context are available

When the *Hands-on Objectives* for this chapter are executed

Then the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

☐ Create categories (hardening, testing, debt, education)

☐ Define SLA classes (expedite for criticals, standard, fixed-date)

☐ Integrate with bug tracker and CWE/CVSS tagging

Acceptance Criteria

☐ Security backlog exists with WIP limits and classes of service

☐ Critical items auto-page, create expedite swimlane

APPSEC-37 — Sprint 0 Security Enablement

Epic / Feature	Agile AppSec Delivery
Business Value	set teams up for success with secure defaults before feature work begins
Priority / Estimate	Priority: Should SP: 8
Persona	DevOps Engineer
Dependencies	Reference architectures; templates available
Assumptions / Risks	Rushing Sprint 0 leads to gaps; time-box essentials

Story *As a DevOps Engineer, I want to Sprint 0 Security Enablement so that set teams up for success with secure defaults before feature work begins.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

☐ Provision repo templates with CI security jobs (SAST/SCA/secret scan)

☐ Generate baseline threat model and architecture diagram

☐ Seed env var policy, secret manager paths, logging/trace defaults

Acceptance Criteria

☐ New repos inherit security CI and pass baseline checks

☐ Threat model ADR committed and linked in README

APPSEC-38 — Security Champions Cadence

Office Hours

Epic / Feature	Agile AppSec Operations
Business Value	scale expertise via lightweight coaching and shared practices
Priority / Estimate	Priority: Should SP: 3
Persona	AppSec Lead
Dependencies	Champions program charter
Assumptions / Risks	Low attendance risk; align with sprint rituals

Story *As a AppSec Lead, I want to Security Champions Cadence*

Office Hours so that scale expertise via lightweight coaching and shared practices.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

☐ Hold bi-weekly office hours and monthly guild sessions

☐ Publish short playbooks and code examples

☐ Track engagement and topics to refine backlog

Acceptance Criteria

☐ Attendance recorded; >70% teams represented

☐ Two new playbooks published per quarter

APPSEC-39 — Security Code Review Checklist

Pairing

Epic / Feature	Agile AppSec Delivery
Business Value	catch issues early by enriching PR reviews with targeted security checks
Priority / Estimate	Priority: Should SP: 5
Persona	Senior Developer
Dependencies	Secure coding standards; code owners defined
Assumptions / Risks	Checklist fatigue; keep concise and role-based

Story *As a Senior Developer, I want to Security Code Review Checklist*

Pairing so that catch issues early by enriching PR reviews with targeted security checks.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

☐ Create language/framework-specific checklists (input, authz, logging)

☐ Enable CODEOWNERS for sensitive paths (auth, crypto, infra)

☐ Pilot pairing/mobbing for risky changes

Acceptance Criteria

☐ Checklist adopted in PR template; CODEOWNERS in repo

☐ Sampling shows >80% PRs include security review notes

APPSEC-40 — Security Test Harness in CI (Unit, Integration, e2e)

Epic / Feature	Agile AppSec Automation
Business Value	turn security ACs into repeatable tests that gate releases
Priority / Estimate	Priority: Must SP: 8
Persona	DevOps Engineer
Dependencies	CI runners; test data strategy
Assumptions / Risks	Flaky tests disrupt delivery; quarantine policy required

Story *As a DevOps Engineer, I want to Security Test Harness in CI (Unit, Integration, e2e) so that turn security ACs into repeatable tests that gate releases.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repositories, environments, and program context are available

When the *Hands-on Objectives* for this chapter are executed

Then the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

□ Translate ACs to tests (unit assertions, e2e negative cases)

□ Add security smoke tests to PR/merge workflows

□ Collect JUnit artifacts and trend failures

Acceptance Criteria

□ Security tests run on each PR and block on critical failures

□ Dashboard shows pass rates per repo

APPSEC-41 — Security SLOs/SLIs Error Budgets

Epic / Feature	Agile AppSec Metrics
Business Value	align risk tolerance with delivery by defining measurable targets
Priority / Estimate	Priority: Should SP: 5
Persona	Product Owner
Dependencies	Metrics dashboard pipeline
Assumptions / Risks	Vanity metrics risk; tie SLIs to outcomes (vuln age, MTTR)

Story *As a Product Owner, I want to Security SLOs/SLIs Error Budgets so that align risk tolerance with delivery by defining measurable targets.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repositories, environments, and program context are available

When the *Hands-on Objectives* for this chapter are executed

Then the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

☐ Define SLIs (critical vuln age, secrets incidents, SBOM freshness)

☐ Set SLOs per tier; error budget burn alerts

☐ Review in sprint review/ops review

Acceptance Criteria

☐ SLIs visible; SLOs approved by stakeholders

☐ Error budget policy documented and in use

APPSEC-42 — Manage Security Debt WIP Limits

Epic / Feature	Agile AppSec Operations
Business Value	prevent accumulation of risk by reserving capacity for security work
Priority / Estimate	Priority: Should SP: 3
Persona	Scrum Master
Dependencies	Security backlog with classes of service
Assumptions / Risks	Feature pressure can erode capacity; enforce WIP

Story *As a Scrum Master, I want to Manage Security Debt WIP Limits so that prevent accumulation of risk by reserving capacity for security work.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

☐ Reserve sprint capacity (e.g., 15–20%) for security items

☐ Set WIP limits and visual policies on the board

☐ Track debt burndown

Acceptance Criteria

☐ Capacity policy visible; burndown trends improving

☐ No sprint closes with critical debt untriaged

APPSEC-43 — Lightweight Risk Exception Time-Bound Waivers

Epic / Feature	Agile AppSec Governance
Business Value	enable pragmatic shipping while controlling residual risk
Priority / Estimate	Priority: Could SP: 3
Persona	Risk Manager
Dependencies	Risk register; waiver workflow
Assumptions / Risks	Waiver sprawl; enforce expirations and ownership

Story *As a Risk Manager, I want to Lightweight Risk Exception Time-Bound Waivers so that enable pragmatic shipping while controlling residual risk.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

☐ Define exception template (owner, risk, compensating controls, expiry)

☐ Automate reminders and revoke on expiry

☐ Report exceptions in QBRs

Acceptance Criteria

☐ All waivers have owners and expirations

☐ Expired waivers auto-alert and block releases if needed

APPSEC-44 — Security Chaos/Game Days

Epic / Feature	Agile AppSec Learning
Business Value	build muscle memory and validate controls under failure conditions
Priority / Estimate	Priority: Could SP: 5
Persona	SRE Lead
Dependencies	Staging environment; playbooks
Assumptions / Risks	Customer impact risk; run in staging with guardrails

Story *As a SRE Lead, I want to Security Chaos/Game Days so that build muscle memory and validate controls under failure conditions.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

- ☐ Design adversarial scenarios (secret leak, token theft, SSRF attempts)
- ☐ Run drills with cross-functional teams
- ☐ Capture learnings and convert to backlog items

Acceptance Criteria

- ☐ At least one drill per quarter with documented outcomes
- ☐ Follow-up stories created and prioritized

APPSEC-45 — Release Readiness Security Checklist

Epic / Feature	Agile AppSec Delivery
Business Value	ensure releases meet baseline security before go-live
Priority / Estimate	Priority: Must SP: 3
Persona	Release Manager
Dependencies	DoD gates; metrics dashboard
Assumptions / Risks	Last-minute crunch; automate checklist population

Story *As a Release Manager, I want to Release Readiness Security Checklist so that ensure releases meet baseline security before go-live.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

- □ Automate checklist (AC met, tests green, SBOM signed, secrets scan)
- □ Gate on unresolved criticals or expired waivers
- □ Publish release notes with security changes

Acceptance Criteria

- □ Checklist artifact attached to each release
- □ No release proceeds with critical blockers

APPSEC-46 — Continuous Education Micro-Learning

Epic / Feature	Agile AppSec Learning
Business Value	raise team capability with short, targeted security modules
Priority / Estimate	Priority: Could SP: 2
Persona	Learning Lead
Dependencies	Champions cadence; LMS
Assumptions / Risks	Low engagement; keep modules <10 min tied to current work

Story *As a Learning Lead, I want to Continuous Education Micro-Learning so that raise team capability with short, targeted security modules.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

□ Publish bite-size modules (e.g., XSS in React, JWT pitfalls)

□ Track completion and impact on defects

□ Reward champions/teams who complete modules

Acceptance Criteria

□ Module catalog live; >60% engineers complete at least one per quarter

□ Correlation shows reduced related defects over time

APPSEC-8 — Standardize Threat Modeling

Epic / Feature	Threat Modeling
Business Value	catch design flaws early and convert threats into actionable requirements
Priority / Estimate	Priority: Must SP: 5
Persona	Security champion
Dependencies	DFD notation, templates
Assumptions / Risks	Analysis paralysis; time-box sessions and prioritize

Story *As a Security champion, I want to Standardize Threat Modeling so that catch design flaws early and convert threats into actionable requirements.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

☐ Choose method (STRIDE/LINDDUN/misuse cases) and templates.

☐ Run two sessions on different architectures; capture DFDs and threats.

☐ Translate top threats into NFRs and tests.

☐ Add a reusable threats/mitigations catalogue to the wiki.

APPSEC-9 — Publish Secure Coding Standards

Epic / Feature	Secure Coding
Business Value	reduce recurring vulnerabilities and speed reviews with clear checklists
Priority / Estimate	Priority: Must SP: 3
Persona	Tech lead
Dependencies	Language stacks agreed
Assumptions / Risks	One-size-fits-none risk; tailor per language

Story *As a Tech lead, I want to Publish Secure Coding Standards so that reduce recurring vulnerabilities and speed reviews with clear checklists.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Write per-language standards (input validation, encoding, secrets, crypto).
- ☐ Add PR checklists and reviewer heuristics.
- ☐ Provide pre-commit hooks and code templates.
- ☐ Run a 45-min training; record and link in the repo.

APPSEC-12 — Enforce API Security Standards

Epic / Feature	API Security
Business Value	protect data and consumers via consistent auth, validation, and quotas
Priority / Estimate	Priority: Must SP: 5
Persona	API owner
Dependencies	OpenAPI/AsyncAPI specs
Assumptions / Risks	Shadow APIs; tie standard to inventory

Story *As a API owner, I want to Enforce API Security Standards so that protect data and consumers via consistent auth, validation, and quotas.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Write API security standard (authn/z, schema validation, rate limiting).
- ☐ Add contract tests and security tests to CI.
- ☐ Gate breaking changes and insecure defaults in PRs.
- ☐ Add discovery checks for undocumented endpoints.

APPSEC-10 — Operationalize SAST/SCA/DAST/IAST

Epic / Feature	Security Testing
Business Value	improve signal-to-noise and make security checks part of normal CI
Priority / Estimate	Priority: Must SP: 5
Persona	Automation engineer
Dependencies	Scanner licenses, CI capacity
Assumptions / Risks	Finding overload; enforce “new high/critical = fail”

Story *As a Automation engineer, I want to Operationalize SAST/SCA/DAST/IAST so that improve signal-to-noise and make security checks part of normal CI.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repositories, environments, and program context are available

When the *Hands-on Objectives* for this chapter are executed

Then the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Integrate SAST & SCA in CI; upload SARIF for code scanning.
- ☐ Stand up targeted DAST/IAST for a high-risk app.
- ☐ Establish severity thresholds, suppressions with expiry, and routing.
- ☐ Publish weekly trend reports and backlog hygiene metrics.

APPSEC-11 — Generate SBOMs & Sign Artifacts

Epic / Feature	Supply Chain Security
Business Value	improve provenance and compliance while enabling safe updates
Priority / Estimate	Priority: Must SP: 5
Persona	Release engineer
Dependencies	SBOM tool, signer
Assumptions / Risks	Tooling gaps; start with top languages/images

Story *As a Release engineer, I want to Generate SBOMs & Sign Artifacts so that improve provenance and compliance while enabling safe updates.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repositories, environments, and program context are available

When the *Hands-on Objectives* for this chapter are executed

Then the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

☐ Produce SBOM (CycloneDX/SPDX) during builds; attach to artifacts.

☐ Sign artifacts/images and verify in promotion gates.

☐ Document third-party source allowlist and review cadence.

☐ Add attestation checks to release workflow.

APPSEC-21 — Plan
Scope a Web App Penetration Test

Epic / Feature	Web App Penetration Testing (WAHH)
Business Value	gain explicit scope, rules of engagement, and safe test windows to prevent production impact
Priority / Estimate	Priority: Must SP: 3
Persona	Security tester
Dependencies	Signed RoE; test accounts; staging/prod window
Assumptions / Risks	Testing in prod may cause instability; throttle and monitor

Story *As a Security tester, I want to Plan Scope a Web App Penetration Test so that gain explicit scope, rules of engagement, and safe test windows to prevent production impact.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks
- □ Define scope (domains, apps, APIs), out-of-scope targets, and credentials
- □ Document test data handling and PII safeguards
- □ Align comms, SLAs for critical findings, and retest windows

cceptance Criteria
- □ RoE doc approved by stakeholders
- □ Test accounts provisioned with role variants (user, admin, support)
- □ Monitoring/alerting teams notified of test window

APPSEC-22 — Reconnaissance Application Mapping

Epic / Feature	Web App Penetration Testing (WAHH)
Business Value	discover hidden attack surface to prioritize testing and coverage
Priority / Estimate	Priority: Must SP: 5
Persona	Security tester
Dependencies	Scope confirmed; wordlists; proxy + crawler
Assumptions / Risks	Over-crawling may trigger rate limits; coordinate with SRE

Story *As a Security tester, I want to Reconnaissance Application Mapping so that discover hidden attack surface to prioritize testing and coverage.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repositories, environments, and program context are available

When the *Hands-on Objectives* for this chapter are executed

Then the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks
- ☐ Map URLs, parameters, methods with an intercepting proxy
- ☐ Enumerate endpoints, SPA routes, and undocumented APIs
- ☐ Fingerprint frameworks, versions, and third-party components

cceptance Criteria
- ☐ Site map exported with parameters and auth contexts
- ☐ List of potential high-risk surfaces identified (auth, upload, serialization)

APPSEC-23 — Test Authentication Session Management

Epic / Feature	Web App Penetration Testing (WAHH)
Business Value	prevent account takeover by finding flaws in login, MFA, and session controls
Priority / Estimate	Priority: Must SP: 8
Persona	Security tester
Dependencies	Accounts with/without MFA; password reset emails
Assumptions / Risks	Lockouts during testing; ensure customer impact safeguards

Story *As a Security tester, I want to Test Authentication Session Management so that prevent account takeover by finding flaws in login, MFA, and session controls.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks
- ☐ Probe MFA bypass, weak recovery flows, and magic-link abuse
- ☐ Assess session fixation/rotation, cookie flags, and idle timeouts
- ☐ Evaluate credential stuffing protections and lockout policies

cceptance Criteria
- ☐ Documented results for MFA, recovery, and session rotation
- ☐ Remediation guidance aligned to OWASP ASVS controls

APPSEC-24 — Test Authorization Access Control (IDOR/BOLA)

Epic / Feature	Web App Penetration Testing (WAHH)
Business Value	stop horizontal/vertical privilege escalation via broken object-level auth
Priority / Estimate	Priority: Must SP: 8
Persona	Security tester
Dependencies	Multiple role accounts; seeded cross-tenant data
Assumptions / Risks	Data exposure risk; use synthetic data

Story *As a Security tester, I want to Test Authorization Access Control (IDOR/BOLA) so that stop horizontal/vertical privilege escalation via broken object-level auth.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repositories, environments, and program context are available

When the *Hands-on Objectives* for this chapter are executed

Then the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks
- ☐ Fuzz identifiers (IDs, GUIDs) and object references for IDOR/BOLA
- ☐ Probe multi-tenant boundaries; confirm server-side checks
- ☐ Check mass assignment and insecure direct mapping in APIs

cceptance Criteria
- ☐ Evidence of any cross-tenant/object access or written ’no repro’ with proof
- ☐ Mitigations mapped to enforcement in controllers/middleware

APPSEC-25 — Injection Testing (SQL/NoSQL/Command/LDAP)

Epic / Feature	Web App Penetration Testing (WAHH)
Business Value	eliminate injection paths that lead to data breach or RCE
Priority / Estimate	Priority: Must SP: 13
Persona	Security tester
Dependencies	Safe test DB; command sandbox in staging
Assumptions / Risks	Potential data corruption; use read-only techniques where possible

Story *As a Security tester, I want to Injection Testing (SQL/NoSQL/Command/LDAP) so that eliminate injection paths that lead to data breach or RCE.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks
- □ Identify user-controlled inputs reaching interpreters
- □ Test with time-based, boolean, and error-based payloads
- □ Validate ORM parameterization and stored procedures

cceptance Criteria
- □ List of vulnerable sinks with PoC payloads, impact, and severity
- □ Verification that parameterization/escaping prevents injection

APPSEC-26 — Cross-Site Scripting (Reflected/Stored/DOM)

Epic / Feature	Web App Penetration Testing (WAHH)
Business Value	prevent account hijack and data theft via XSS in templates and SPA flows
Priority / Estimate	Priority: Must SP: 8
Persona	Security tester
Dependencies	CSP report URI; proxy instrumentation
Assumptions / Risks	False negatives in SPA due to client-side routing; exhaustive param coverage needed

Story *As a Security tester, I want to Cross-Site Scripting (Reflected/Stored/DOM) so that prevent account hijack and data theft via XSS in templates and SPA flows.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repositories, environments, and program context are available

When the *Hands-on Objectives* for this chapter are executed

Then the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks
- ☐ Probe contexts (HTML, attribute, JS, URL, style) for escaping failures
- ☐ Verify CSP, output encoding, and template auto-escape settings
- ☐ DOM XSS checks in dynamic frameworks

cceptance Criteria
- ☐ Any exploitable XSS documented with payload, context, and fix
- ☐ CSP evaluated; recommendations provided (nonce, strict-dynamic)

APPSEC-27 — CSRF SameSite Protections

Epic / Feature	Web App Penetration Testing (WAHH)
Business Value	block unauthorized state changes from cross-origin requests
Priority / Estimate	Priority: Should SP: 5
Persona	Security tester
Dependencies	Test harness for cross-origin forms/XHR/fetch
Assumptions / Risks	CSRF tests may trigger state changes; only use reversible actions

Story *As a Security tester, I want to CSRF SameSite Protections so that block unauthorized state changes from cross-origin requests.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

- ☐ Validate anti-CSRF tokens, double-submit, and origin checks
- ☐ Verify cookie SameSite, secure flags, and CORS policies
- ☐ Test JSON/GraphQL mutations for CSRF gaps

cceptance Criteria

- ☐ Critical state-changing routes confirmed protected or issues filed
- ☐ CORS and SameSite settings documented with recommendations

APPSEC-28 — File Upload Path Traversal RCE

Epic / Feature	Web App Penetration Testing (WAHH)
Business Value	prevent arbitrary code execution and data exposure via unsafe file handling
Priority / Estimate	Priority: Must SP: 8
Persona	Security tester
Dependencies	Isolated storage; antivirus/sandbox rules
Assumptions / Risks	Prod AV may quarantine test payloads; coordinate

Story *As a Security tester, I want to File Upload*
Path Traversal
RCE so that prevent arbitrary code execution and data exposure via unsafe file handling.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks
- ☐ Test MIME/type/extension checks and content-sniffing bypasses
- ☐ Probe image/polyglot payloads and storage path traversal
- ☐ Validate media processing libraries for RCE vectors

cceptance Criteria
- ☐ Uploads constrained by allowlist and verified server-side
- ☐ No traversal or remote execution demonstrated

APPSEC-29 — Deserialization Cryptographic Failures

Epic / Feature	Web App Penetration Testing (WAHH)
Business Value	mitigate code execution and privilege escalation through unsafe serialization and weak crypto
Priority / Estimate	Priority: Should SP: 8
Persona	Security tester
Dependencies	Known gadget chains in test env; key rotation docs
Assumptions / Risks	Key leakage risk; use dummy keys in tests

Story *As a Security tester, I want to Deserialization Cryptographic Failures so that mitigate code execution and privilege escalation through unsafe serialization and weak crypto.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks
- ☐ Identify serialization formats (Java, PHP, JWT, protobuf) and trust boundaries
- ☐ Attempt known gadget chains; check object injection paths
- ☐ Assess JWT alg confusion, weak signing, and key exposure

cceptance Criteria
- ☐ Unsafe deserialization paths cataloged or remediated
- ☐ Crypto controls validated against ASVS (key mgmt, algs, rotation)

APPSEC-30 — SSRF/XXE
Server-Side Template Injection

Epic / Feature	Web App Penetration Testing (WAHH)
Business Value	stop lateral movement to internal services and metadata endpoints
Priority / Estimate	Priority: Must SP: 8
Persona	Security tester
Dependencies	Egress controls; canary endpoints
Assumptions / Risks	Risk of internal service impact; coordinate with platform team

Story *As a Security tester, I want to SSRF/XXE Server-Side Template Injection so that stop lateral movement to internal services and metadata endpoints.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks
- ☐ Probe URL fetchers and XML parsers for SSRF/XXE
- ☐ Validate denylists/allowlists, outbound proxy, and metadata protections
- ☐ Test template engines for SSTI to RCE chains

cceptance Criteria
- ☐ No internal egress or metadata access possible without policy
- ☐ Template engines hardened or issues raised with PoCs

APPSEC-31 — Business Logic Abuse Rate Limiting Automation

Epic / Feature	Web App Penetration Testing (WAHH)
Business Value	protect revenue and integrity by preventing workflow abuse and brute force
Priority / Estimate	Priority: Should SP: 5
Persona	Security tester
Dependencies	Analytics dashboards; throttling configs
Assumptions / Risks	Blocking legitimate users during tests; throttle carefully

Story *As a Security tester, I want to Business Logic Abuse Rate Limiting Automation so that protect revenue and integrity by preventing workflow abuse and brute force.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks
- ☐ Enumerate critical workflows (checkout, transfers, promotions)
- ☐ Test replay, race conditions, and coupon abuse
- ☐ Evaluate rate limiting, CAPTCHA, and bot defenses

cceptance Criteria
- ☐ Abuse scenarios documented with loss estimates and fixes
- ☐ Effective rate limits in place for sensitive endpoints

APPSEC-32 — Clickjacking
Caching
Sensitive Data Exposure

Epic / Feature	Web App Penetration Testing (WAHH)
Business Value	reduce data leakage and UI redress attacks
Priority / Estimate	Priority: Could SP: 3
Persona	Security tester
Dependencies	Response headers report; CDN config
Assumptions / Risks	Cache poisoning risk; test in staging when possible

Story *As a Security tester, I want to Clickjacking*
Caching
Sensitive Data Exposure so that reduce data leakage and UI redress attacks.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks
- ☐ Verify X-Frame-Options/Content-Security-Policy frame-ancestors
- ☐ Check cache-control on authenticated responses
- ☐ Scan for sensitive data in URLs, logs, and client storage

cceptance Criteria
- ☐ Headers configured defensively (no-store where needed)
- ☐ No sensitive data found in caches or client-side storage

APPSEC-33 — Report Triage Retest Findings

Epic / Feature	Web App Penetration Testing (WAHH)
Business Value	translate findings into engineering work, validate fixes, and build learning loops
Priority / Estimate	Priority: Must SP: 5
Persona	Security tester
Dependencies	Ticketing templates; CWE/CVRSS mapping
Assumptions / Risks	Fix regressions possible; ensure retest scripts are reusable

Story *As a Security tester, I want to Report Triage Retest Findings so that translate findings into engineering work, validate fixes, and build learning loops.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

- □ Create tickets with repro steps, impact, CWE, and severity
- □ Partner with owners on fixes and timelines
- □ Retest and close with evidence; update knowledge base

cceptance Criteria

- □ All critical/high issues triaged within SLA and retested
- □ KB updated with playbooks and examples

APPSEC-13 — Publish Cloud AppSec Baseline

Epic / Feature	Cloud-Native App Security
Business Value	set secure defaults for identity, secrets, network, and logging
Priority / Estimate	Priority: Should SP: 3
Persona	Cloud security engineer
Dependencies	Cloud org access
Assumptions / Risks	Drift risk; add config conformance packs

Story *As a Cloud security engineer, I want to Publish Cloud AppSec Baseline so that set secure defaults for identity, secrets, network, and logging.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Define shared-responsibility for app teams; list must-have controls.
- ☐ Provide bootstrap templates for logging/telemetry and secrets.
- ☐ Add guardrails and conformance checks.
- ☐ Document carve-outs and exception review.

APPSEC-14 — Harden Containers & Kubernetes

Epic / Feature	Container/K8s Security
Business Value	reduce runtime risk with minimal images and admission policies
Priority / Estimate	Priority: Must SP: 5
Persona	Platform engineer
Dependencies	Registry, admission controller
Assumptions / Risks	Breakages; start in warn mode, then enforce

Story *As a Platform engineer, I want to Harden Containers & Kubernetes so that reduce runtime risk with minimal images and admission policies.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- □ Create minimal, scanned base images; publish usage guidance.
- □ Enforce image provenance and vulnerability thresholds at admission.
- □ Apply Pod Security standards, RBAC, and NetworkPolicies.
- □ Add runtime policies for sensitive syscalls and egress.

APPSEC-15 — Centralize Secrets & Workload Identity

Epic / Feature	Secrets & IAM
Business Value	eliminate hardcoded secrets and reduce blast radius via least privilege
Priority / Estimate	Priority: Must SP: 3
Persona	Service owner
Dependencies	Secrets manager, IAM
Assumptions / Risks	Migration risk; migrate one app first

Story *As a Service owner, I want to Centralize Secrets & Workload Identity so that eliminate hardcoded secrets and reduce blast radius via least privilege.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Move secrets to a managed store with rotation.
- ☐ Adopt workload identity (mTLS/JWT/OIDC) for services.
- ☐ Review and minimize IAM policies per service.
- ☐ Add secrets scanning in CI and pre-commit.

APPSEC-47 — Define Policy-as-Code Strategy

Reference Architecture

Epic / Feature	Policy as Code
Business Value	create consistent, testable guardrails across repos, pipelines, cloud, and clusters
Priority / Estimate	Priority: Must SP: 5
Persona	Security Architect
Dependencies	SSDLC policy; cloud/K8s baselines; CI access
Assumptions / Risks	Too many frameworks increases toil; pick minimal viable set

Story *As a Security Architect, I want to Define Policy-as-Code Strategy*

Reference Architecture so that create consistent, testable guardrails across repos, pipelines, cloud, and clusters.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

- □ Select core frameworks and scopes: OPA/Rego (Conftest bundles), Gatekeeper/Kyverno (K8s), IaC checks (Terraform plans), pipeline policies
- □ Define target enforcement points: pre-commit, PR, CI, admission, deploy, runtime
- □ Write an ADR documenting choices, bundle layout, versioning, and promotion model (dev→stg→prod)

Acceptance Criteria

- □ Reference architecture approved by Platform, AppSec, and SRE
- □ Hello-world policy proven in one repo and one cluster in *audit* mode
- □ Docs published: “How policies run” + developer quickstart

APPSEC-48 — Author Baseline Policy Library

Epic / Feature	Policy as Code
Business Value	codify critical controls (secrets, SBOM, least privilege, network) with reusable rules
Priority / Estimate	Priority: Must SP: 8
Persona	Policy Engineer
Dependencies	Reference architecture; control dictionary
Assumptions / Risks	Over-blocking risk; start with *audit* severity and tune

Story *As a Policy Engineer, I want to Author Baseline Policy Library so that codify critical controls (secrets, SBOM, least privilege, network) with reusable rules.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repositories, environments, and program context are available

When the *Hands-on Objectives* for this chapter are executed

Then the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

- □ Write baseline policies: repo (branch protection, required checks), CI (required SAST/SCA), IaC (public buckets, open SGs, unencrypted volumes), K8s (PSa, runAsNonRoot, image provenance), Cloud (IAM wildcard deny)
- □ Provide passing/failing examples and unit tests (e.g., `rego` tests) for each rule
- □ Tag rules by tier (P0–P3) and map to ASVS/SSDF controls

Acceptance Criteria

- □ Library stored as versioned bundles with tests passing in CI
- □ Each rule has rationale, remediation text, and references

APPSEC-49 — Build Local Dev Tooling

Pre-Commit Experience

Epic / Feature	Policy as Code
Business Value	shift-left feedback via IDE/CLI so engineers fix before PR
Priority / Estimate	Priority: Should SP: 5
Persona	Developer Experience Lead
Dependencies	Baseline policy library
Assumptions / Risks	Tool friction; ensure fast local runs

Story *As a Developer Experience Lead, I want to Build Local Dev Tooling*

Pre-Commit Experience so that shift-left feedback via IDE/CLI so engineers fix before PR.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repositories, environments, and program context are available

When the *Hands-on Objectives* for this chapter are executed

Then the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

□ Publish `make policy-test` + `pre-commit` hooks (conftest, yaml/json/plan inputs)

□ Ship IDE tasks/snippets and a sample app showing policy passes/fails

□ Document troubleshooting and rule suppression with expiry metadata

Acceptance Criteria

□ New repos enable pre-commit in <5 min and get local results <2s

□ Suppressions require owner, ticket, expiry; flagged in CI on expiry

APPSEC-50 — Integrate Policies into CI/CD

Admission & Deploy Gates

Epic / Feature	Policy as Code
Business Value	prevent risky changes by gating merges and deploys with policy checks
Priority / Estimate	Priority: Must SP: 8
Persona	Platform Engineer
Dependencies	CI runners; admission controller; registry access
Assumptions / Risks	Breaking builds en masse; roll out by cohort and audit-first

Story *As a Platform Engineer, I want to Integrate Policies into CI/CD*

Admission & Deploy Gates so that prevent risky changes by gating merges and deploys with policy checks.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

☐ Add conftest checks to PRs (IaC, manifests, pipeline config); publish SARIF annotations

☐ Install Gatekeeper/Kyverno; onboard namespaces in *audit* then *enforce*

☐ Enforce image provenance/SBOM signature at admission; block on criticals

Acceptance Criteria

☐ CI fails for new critical violations; admission denies non-compliant pods/images

☐ Rollout plan tracked; <2% false-positive rate post-tuning

APPSEC-51 — Exceptions/Waivers as Code

Epic / Feature	Policy as Code
Business Value	enable pragmatic delivery with time-bound, reviewable exceptions
Priority / Estimate	Priority: Must SP: 5
Persona	Risk Manager
Dependencies	Risk register; waiver workflow
Assumptions / Risks	Shadow waivers; require owners and expirations

Story *As a Risk Manager, I want to Exceptions/Waivers as Code so that enable pragmatic delivery with time-bound, reviewable exceptions.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

□ Define waiver schema (owner, risk, justification, compensating controls, expiry)

□ Store waivers near code (YAML/CRD); policies read waivers at evaluate-time

□ Auto-alert before expiry; block builds on expired waivers

Acceptance Criteria

□ All policy suppressions reference a waiver ID and ticket

□ Quarterly review report lists active/expired waivers by service

APPSEC-52 — Policy Telemetry

Dashboards & Coverage

Epic / Feature	Policy as Code
Business Value	observe adoption, denials, and drift to guide improvements
Priority / Estimate	Priority: Should SP: 5
Persona	Program Manager
Dependencies	Logging backend; metrics stack
Assumptions / Risks	Noisy logs; sample and aggregate wisely

Story *As a Program Manager, I want to Policy Telemetry*

Dashboards & Coverage so that observe adoption, denials, and drift to guide improvements.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

- □ Collect decision logs (OPA), admission denials, CI failures; tag by app/tier/team
- □ Build dashboard: pass/fail rates, top rules hit, time-to-fix, waiver counts
- □ Track coverage: % repos with CI checks; % namespaces enforcing; % images verified

Acceptance Criteria

- □ Monthly report shows improving coverage and reduced critical violations
- □ Error budget alerts for rising denial rates or stale waivers

APPSEC-53 — Policy Bundles Registry

Versioning & Promotion

Epic / Feature	Policy as Code
Business Value	safely evolve policies via semantic versions and environment promotion
Priority / Estimate	Priority: Should SP: 3
Persona	Release Engineer
Dependencies	OCI registry or artifact store
Assumptions / Risks	Drift across envs; automate promotions

Story *As a Release Engineer, I want to Policy Bundles Registry*

Versioning & Promotion so that safely evolve policies via semantic versions and environment promotion.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repositories, environments, and program context are available

When the *Hands-on Objectives* for this chapter are executed

Then the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

Tasks

- ☐ Package policy bundles; publish to OCI registry with semver and changelogs
- ☐ Automate promotion (dev→stg→prod) after smoke-tests
- ☐ Define deprecation policy and migration guides for breaking changes

Acceptance Criteria

- ☐ Envs reference immutable bundle digests
- ☐ Rollbacks possible by pinning previous versions

APPSEC-54 — Define Security Vision, Threats, and Controls

Epic / Feature	Security as Code Foundations
Business Value	align the team on risks and codify controls that will be enforced by pipelines
Priority / Estimate	Priority: Must SP: 3
Persona	Platform engineer
Dependencies	Sample app repo; sandbox account
Assumptions / Risks	Over-scoping threat model; keep to top 5 risks

Story *As a Platform engineer, I want to Define Security Vision, Threats, and Controls so that align the team on risks and codify controls that will be enforced by pipelines.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repositories, environments, and program context are available

When the *Hands-on Objectives* for this chapter are executed

Then the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

☐ Add `docs/security-vision.md`: goals, assumptions, non-goals.

☐ Create a 1-page STRIDE-lite model for the app & cloud footprint.

☐ Publish a control catalog CSV with owner, evidence, and CI gate mapping.

☐ Link all of the above from the README; set a quarterly review.

APPSEC-55 — Bootstrap IaC & CI Foundations

Epic / Feature	Security as Code Foundations
Business Value	create a reproducible base that enables automated security checks
Priority / Estimate	Priority: Must SP: 5
Persona	DevOps engineer
Dependencies	Artifact bucket/registry; CI runners
Assumptions / Risks	Leaked secrets risk; adopt OIDC and pre-commit scanners

Story *As a DevOps engineer, I want to Bootstrap IaC & CI Foundations so that create a reproducible base that enables automated security checks.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

☐ Provision minimal VPC, registry, and CI roles via IaC (encrypted by default).

☐ Pipeline builds container, runs linters and SCA, pushes image to registry.

☐ Enable pre-commit hooks (`tfsec`/`cfn-lint`, `hadolint`, secrets scan).

☐ Protect `main`: require passing checks; show badge in README.

APPSEC-56 — Preventive & Detective Controls as Code

Epic / Feature	Security as Code Controls
Business Value	block misconfigs before deploy and surface evidence automatically
Priority / Estimate	Priority: Must SP: 8
Persona	Security champion
Dependencies	Working CI; IaC modules
Assumptions / Risks	False positives; add waivers with time-boxed expiry

Story *As a Security champion, I want to Preventive & Detective Controls as Code so that block misconfigs before deploy and surface evidence automatically.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Write guard/OPA policies: no public buckets, encryption-at-rest, deny wildcard IAM.
- ☐ Enable Security Hub/GuardDuty/Config rules; encrypt logs with KMS.
- ☐ Add a `policy-check` job that fails on violations and posts rule summaries.
- ☐ Emit a control-coverage matrix artifact and link in job summary.

APPSEC-57 — Centralize Telemetry & Alerts

Epic / Feature	Security as Code Observability
Business Value	improve detection/triage via standard logs, metrics, and alarms as code
Priority / Estimate	Priority: Must SP: 5
Persona	SRE / observability engineer
Dependencies	KMS keys; log shipping
Assumptions / Risks	Alert fatigue; tune severities and routes

Story *As a SRE / observability engineer, I want to Centralize Telemetry & Alerts so that improve detection/triage via standard logs, metrics, and alarms as code.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Enable org CloudTrail; VPC Flow Logs; cluster audit logs with retention.
- ☐ Emit app logs as structured JSON with correlation IDs.
- ☐ Create alarms for auth failures, 5xx spikes, throttling, and unusual egress.
- ☐ Build a dashboard JSON and link it from the README.

APPSEC-58 — Automate Access (IAM, RBAC, IRSA)

Epic / Feature	Security as Code Access Control
Business Value	reduce standing privileges and make access auditable end-to-end
Priority / Estimate	Priority: Must SP: 5
Persona	Cloud security engineer
Dependencies	EKS/ECS/OIDC configured
Assumptions / Risks	Privilege creep; schedule periodic reviews

Story *As a Cloud security engineer, I want to Automate Access (IAM, RBAC, IRSA) so that reduce standing privileges and make access auditable end-to-end.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Adopt IRSA/OIDC for workloads; remove node-wide credentials.
- ☐ Generate least-priv IAM with Access Analyzer and validate in CI.
- ☐ Define Kubernetes RBAC via GitOps; separate dev/ops permissions.
- ☐ Add break-glass role with MFA and session recording.

APPSEC-59 — Secrets Hygiene as Code

Epic / Feature	Security as Code Secrets
Business Value	prevent credential leaks and shrink blast radius
Priority / Estimate	Priority: Must SP: 5
Persona	Dev lead
Dependencies	Pre-commit configured
Assumptions / Risks	Developer friction; provide quick-fix guidance

Story *As a Dev lead, I want to Secrets Hygiene as Code so that prevent credential leaks and shrink blast radius.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

☐ Add secrets scanning in pre-commit and CI with org allowlist.

☐ Block merges on new high-sev matches; allow time-bound waivers.

☐ Publish rotation runbook; integrate auto-revocation for leaked keys.

APPSEC-60 — Vault Integration & Rotation

Epic / Feature	Security as Code Secrets
Business Value	eliminate static credentials and automate rotation evidence
Priority / Estimate	Priority: Must SP: 5
Persona	Platform engineer
Dependencies	Secrets manager/Vault; CSI driver
Assumptions / Risks	Migration risk; start with one service

Story *As a Platform engineer, I want to Vault Integration & Rotation so that eliminate static credentials and automate rotation evidence.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Inject app config via CSI/env-from; remove plaintext secrets from repo.
- ☐ Configure rotation for DB/API keys; surface status in CI.
- ☐ Add policy test that fails if opaque K8s Secrets hold known sensitive patterns.

APPSEC-61 — Container Hardening as Code

Epic / Feature	Security as Code Supply Chain
Business Value	standardize minimal, non-root images and enforce at deploy
Priority / Estimate	Priority: Should SP: 5
Persona	Senior developer
Dependencies	Registry; base images
Assumptions / Risks	Breakages from base changes; canary rollout

Story *As a Senior developer, I want to Container Hardening as Code so that standardize minimal, non-root images and enforce at deploy.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Provide hardened base images (non-root, pinned digests) and usage guide.
- ☐ Add `hadolint` & `trivy image` with thresholds to CI.
- ☐ Enforce rootless, read-only FS via Helm/K8s manifests.

APPSEC-62 — SBOM, Provenance & Signing

Epic / Feature	Security as Code Supply Chain
Business Value	improve provenance and verify artifacts automatically
Priority / Estimate	Priority: Must SP: 5
Persona	Release engineer
Dependencies	Cosign/Sigstore; CycloneDX/SPDX
Assumptions / Risks	Tooling variance; start with top services

Story *As a Release engineer, I want to SBOM, Provenance & Signing so that improve provenance and verify artifacts automatically.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Generate SBOMs during build and publish as CI artifacts.
- ☐ Sign images and attest build provenance; verify at admission.
- ☐ Document KMS key rotation for signing; add failure runbook.

APPSEC-63 — Security Unit & Contract Tests

Epic / Feature	Security as Code Testing
Business Value	convert requirements to executable checks that block risky changes
Priority / Estimate	Priority: Must SP: 5
Persona	QA engineer
Dependencies	AC library; test data
Assumptions / Risks	Flaky tests; add quarantine/nightly runs

Story *As a QA engineer, I want to Security Unit & Contract Tests so that convert requirements to executable checks that block risky changes.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Add negative unit tests (authz, validation, encoding boundaries).
- ☐ Generate contract tests from OpenAPI (auth scopes, rate limits, schema).
- ☐ Publish JUnit; gate merges on critical failures.

APPSEC-64 — API Security Tests in CI

Epic / Feature	Security as Code Testing
Business Value	prevent BOLA/IDOR and unsafe defaults with repeatable checks
Priority / Estimate	Priority: Should SP: 5
Persona	Security tester
Dependencies	OpenAPI/GraphQL schema
Assumptions / Risks	Synthetic data required; avoid real PII

Story *As a Security tester, I want to API Security Tests in CI so that prevent BOLA/IDOR and unsafe defaults with repeatable checks.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Fuzz IDs with multi-identity accounts to detect IDOR/BOLA.
- ☐ Validate scopes/claims on sensitive endpoints; test CSRF/CORS.
- ☐ Fail pipeline on exploitable findings; auto-file tickets with repro.

APPSEC-65 — Continuous Fuzzing as Code

Epic / Feature	Security as Code Testing
Business Value	discover edge-case bugs via coverage-guided fuzzing
Priority / Estimate	Priority: Could SP: 5
Persona	DevOps engineer
Dependencies	Fuzz harnesses
Assumptions / Risks	Compute cost; run nightly for depth

Story *As a DevOps engineer, I want to Continuous Fuzzing as Code so that discover edge-case bugs via coverage-guided fuzzing.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Add fuzzers for parsers/critical libs; short run on PRs.
- ☐ Extended fuzz nightly; publish minimized crashes as artifacts.

APPSEC-66 — Release Readiness as Code

Epic / Feature	Security as Code Release
Business Value	ensure releases meet baseline security and ship evidence
Priority / Estimate	Priority: Must SP: 3
Persona	Release manager
Dependencies	Previous SAC stories complete
Assumptions / Risks	Last-minute surprises; precompute checklist

Story *As a Release manager, I want to Release Readiness as Code so that ensure releases meet baseline security and ship evidence.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Generate release checklist (AC met, tests green, SBOM present, signatures valid, secrets scan clean).
- ☐ Block release on criticals or expired waivers; publish security notes.

APPSEC-67 — Runtime Detection Rules as Code

Epic / Feature	Security as Code Runtime
Business Value	detect abuse/misuse with declarative runtime policies
Priority / Estimate	Priority: Should SP: 5
Persona	SRE lead
Dependencies	Centralized logs/metrics
Assumptions / Risks	Noise risk; tune with incident feedback

Story *As a SRE lead, I want to Runtime Detection Rules as Code so that detect abuse/misuse with declarative runtime policies.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Deploy eBPF/Falco rules for exec in containers, sensitive file access, outbound spikes.
- ☐ Route alerts with enriched context (pod, image digest, commit SHA).

APPSEC-68 — Compliance Mapping & Validations

Epic / Feature	Security as Code Compliance
Business Value	prove control effectiveness continuously
Priority / Estimate	Priority: Should SP: 5
Persona	Program manager
Dependencies	Control catalog
Assumptions / Risks	Stale mappings; auto-generate from source

Story *As a Program manager, I want to Compliance Mapping & Validations so that prove control effectiveness continuously.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

□ Map controls to CIS/SSDF in machine-readable form (CSV/OSCAL).

□ Schedule validations (InSpec/Conftest) and export pass/fail to a lake.

□ Generate monthly effectiveness report with trends.

APPSEC-69 — Drift Detection & Auto-Remediation

Epic / Feature	Security as Code Operations
Business Value	reduce exposure by catching and fixing drift quickly
Priority / Estimate	Priority: Should SP: 5
Persona	Platform engineer
Dependencies	GitOps desired state
Assumptions / Risks	False remediation risk; start with suggest/fix PRs

Story *As a Platform engineer, I want to Drift Detection & Auto-Remediation so that reduce exposure by catching and fixing drift quickly.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Enable drift detectors; post annotated diffs to PRs.
- ☐ Auto-open remediation PRs for low-risk drifts; page on critical drift.

APPSEC-70 — Evidence Pipeline & Dashboards

Epic / Feature	Security as Code Metrics
Business Value	make posture visible and self-serve to product teams
Priority / Estimate	Priority: Must SP: 3
Persona	Data engineer
Dependencies	CI artifacts; logs; SBOMs
Assumptions / Risks	Data sprawl; define a minimal schema

Story *As a Data engineer, I want to Evidence Pipeline & Dashboards so that make posture visible and self-serve to product teams.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

□ Ingest JUnit, SARIF, SBOMs, attestations into a lake with app/tier labels.

□ Build dashboards: pass/fail rates, vuln age, waiver counts, coverage %.

□ Publish team scorecards and quarterly trend reports.

APPSEC-16 — Unify Vulnerability Intake & SLAs

Epic / Feature	Vulnerability Management
Business Value	prioritize by exploitability and asset criticality to reduce MTTR
Priority / Estimate	Priority: Must SP: 5
Persona	Vuln management owner
Dependencies	Scanner feeds, ticketing
Assumptions / Risks	Duplicate noise; dedupe by CWE/package/asset

Story *As a Vuln management owner, I want to Unify Vulnerability Intake & SLAs so that prioritize by exploitability and asset criticality to reduce MTTR.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- □ Define prioritization (CVSS/EPSS + criticality + exposure).
- □ Create unified intake and dedup logic across code/deps/containers/infra.
- □ Set SLAs per tier and auto-create tickets with owners and due dates.
- □ Build dashboard (age buckets, MTTR, reopen rate).

APPSEC-17 — Integrate AppSec into Incident Response

Epic / Feature	App IR
Business Value	speed containment and comms for app-specific incidents
Priority / Estimate	Priority: Should SP: 3
Persona	IR lead
Dependencies	On-call schedule, playbooks
Assumptions / Risks	Confusion in roles; publish contact matrix

Story *As a IR lead, I want to Integrate AppSec into Incident Response so that speed containment and comms for app-specific incidents.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Write app-centric playbooks (auth bypass, data exfil, supply-chain).
- ☐ Define evidence capture and comms templates (legal/regulatory triggers).
- ☐ Run a tabletop; record actions and owners.
- ☐ Add lessons learned template and review cadence.

APPSEC-18 — Set AI/ML Security Guardrails

Epic / Feature	AI/ML Security
Business Value	prevent model abuse and data leakage with standards and tests
Priority / Estimate	Priority: Could SP: 5
Persona	ML product owner
Dependencies	Model inventory, logs
Assumptions / Risks	Novel threats; start with one model/feature

Story *As a ML product owner, I want to Set AI/ML Security Guardrails so that prevent model abuse and data leakage with standards and tests.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Threat-model one ML feature (prompt injection, data poisoning, model theft).
- ☐ Add adversarial test cases and output filters.
- ☐ Log model interactions for abuse patterns.
- ☐ Document red-team scenarios and escalation paths.

APPSEC-19 — Automate Evidence & ChatOps

Epic / Feature	Automation & Orchestration
Business Value	reduce toil and raise adoption with bots, policies-as-code, and summaries
Priority / Estimate	Priority: Should SP: 3
Persona	Automation engineer
Dependencies	Bot account, APIs
Assumptions / Risks	Alert fatigue; keep messages concise with links

Story *As a Automation engineer, I want to Automate Evidence & ChatOps so that reduce toil and raise adoption with bots, policies-as-code, and summaries.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- ☐ Auto-comment PRs with scanner summaries and fix hints.
- ☐ Scaffold “new service” with secure defaults via a bot command.
- ☐ Export evidence (SBOM, test reports, approvals) automatically.
- ☐ Maintain an automation backlog with value stream mapping.

APPSEC-20 — Ship Metrics Dashboard & Maturity Plan

Epic / Feature	Metrics & Maturity
Business Value	prove risk reduction and align roadmap with measurable outcomes
Priority / Estimate	Priority: Must SP: 3
Persona	Program manager
Dependencies	Data sources, dashboard tool
Assumptions / Risks	Metric cargo-cult; define glossary and collection method

Story *As a Program manager, I want to Ship Metrics Dashboard & Maturity Plan so that prove risk reduction and align roadmap with measurable outcomes.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repositories, environments, and program context are available

When the *Hands-on Objectives* for this chapter are executed

Then the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and 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.

Tasks

- □ Choose north-star KPIs (risk reduced, MTTR, escape rate) and definitions.
- □ Build a dashboard with trends and targets; segment by tier/team.
- □ Run baseline maturity assessment (e.g., SAMM) and publish a 12-month plan.
- □ Review quarterly and adjust priorities based on results.

Capstone & Milestones (Reference)

Foundation: Charter, control dictionary, inventory/tiering, risk register.
Build-in Security: Reference architectures, SSDLC, champions, secure coding, testing.
Platform Guardrails: SBOM/signing, API/cloud/K8s baselines, secrets/IAM, **policy as code**, **security as code**.
Operate & Improve: Vuln SLAs, App IR, AI/ML guardrails, automation, metrics+maturity.