Vulnerability Management Policy

How Splits identifies, classifies, and remediates vulnerabilities.

Effective	May 19, 2026	**Version**	1.0
Organization	Splits	**Owner**	Mustafa Alhelawe, Founder
Contact	security@splitshq.com	**Review cadence**	Annual

---

1. Purpose and Scope

This Vulnerability Management Policy describes how Splits identifies, classifies, and remediates security vulnerabilities in its application code, third-party dependencies, infrastructure, and developer endpoints. It applies to all assets used to develop or operate the Service.

2. Roles and Responsibilities

The Founder, Mustafa Alhelawe, is responsible for vulnerability identification, triage, remediation, and reporting. External researchers and users may report vulnerabilities to security@splitshq.com; reports are acknowledged within five business days.

3. Asset Inventory

Splits maintains an informal inventory of the assets within scope of this Policy:

- Application source code (Next.js web application, shared TypeScript packages) hosted on GitHub.
- Production environment hosted on Vercel.
- Database, authentication, realtime, and storage services hosted on Supabase.
- Third-party processors: Plaid (financial-account connectivity), Twilio (SMS), Google (optional OAuth sign-in).
- Developer endpoint: the Founder's macOS laptop used for development and deployment.

4. Vulnerability Identification

4.1 Dependency Scanning

GitHub Dependabot is enabled on the Splits repository with alerts, security updates, and version updates active. Dependabot continuously scans dependency manifests against the GitHub Advisory Database and opens pull requests to remediate vulnerable dependencies.

4.2 Build-Time Checks

Continuous integration may run `pnpm audit` against direct and transitive dependencies during builds. Findings rated High or Critical block release pipelines or are explicitly waived with justification.

4.3 Vendor-Managed Infrastructure

Vercel, Supabase, Plaid, and Twilio operate the underlying platform infrastructure and perform their own vulnerability scanning, patching, and security testing. Splits monitors vendor security advisories and applies any consumer-side mitigations promptly.

4.4 Endpoint Vulnerability Management

The Founder's development endpoint runs macOS with automatic security updates enabled. Built-in protections (XProtect, Gatekeeper, MRT) are active. As the team grows, formal endpoint detection and response (EDR) tooling may be adopted.

5. Classification and Risk Rating

Vulnerabilities are classified using a combination of CVSS v3.x base score (where available) and contextual factors specific to Splits' architecture and data flows. The classification scheme is:

Severity	CVSS base score	Typical example	Treatment
Critical	9.0 – 10.0	Unauthenticated remote code execution, exposure of plaintext credentials at scale.	Immediate triage; deploy within SLA below.
High	7.0 – 8.9	Authenticated privilege escalation, exposure of consumer financial data.	Triage within one business day.
Medium	4.0 – 6.9	Limited information disclosure, denial-of-service requiring specific conditions.	Triage during regular work week.
Low / Informational	< 4.0	Best-practice deviations, hardening opportunities with no direct exploit path.	Address in routine maintenance.

Contextual factors that may raise or lower a rating include the presence or absence of mitigating controls (such as RLS, encryption, or authentication gates), exploitability in Splits' specific configuration, and the sensitivity of any data exposed.

6. Remediation Service Level Agreements

Once a vulnerability has been confirmed and classified, it is remediated within the timeframes below, measured from the date of confirmation:

Severity	Remediation SLA	Notes
Critical	Within 7 calendar days	Out-of-band release if needed.
High	Within 30 calendar days	Bundled with next regular release.
Medium	Within 60 calendar days	Tracked in backlog with target date.
Low / Informational	Within 90 calendar days	Addressed during routine maintenance.

Where a remediation requires changes outside of Splits' control (for example, a vendor patch), Splits applies any available compensating controls within the same window and tracks the upstream fix to closure.

7. End-of-Life Software Monitoring

The Founder reviews end-of-life status of key software at least annually using public sources such as endoflife.date. The review covers:

- Node.js runtime version.
- Next.js and React major versions.
- Postgres version (as managed by Supabase).
- Direct dependencies that have entered or are nearing end-of-life.

Upgrades from end-of-life software are scheduled before the upstream end-of-life date wherever practical, and within 90 days afterward at the latest.

8. Endpoint Security

Developer endpoints are configured with full-disk encryption (FileVault on macOS), automatic OS security updates, a screen-lock timeout, and a strong device password or biometric unlock. Application installs are sourced from the App Store, the vendor's official distribution channel, or verified package managers.

9. Exception Process

If a vulnerability cannot be remediated within the applicable SLA, the Founder documents the exception, the compensating controls in place, and the planned remediation date. Exceptions are reviewed monthly until closed.

10. Reporting and Metrics

The Founder tracks at minimum: open vulnerabilities by severity, mean time to remediate by severity, and the count of active SLA exceptions. These metrics are reviewed during the annual policy review and during any post-incident retrospective.

11. Policy Review

This Policy is reviewed at least annually and updated as Splits' architecture, dependency footprint, or vendor mix changes materially. Current version v1.0, effective May 19, 2026.