Security Policy

Thank you for helping keep **ALPHA** and its users safe.

Honest Security Posture

ALPHA is currently a **front-end reference implementation with simulated data**. Please read the following before
evaluating or reporting:

Claim shown in the UI	Actual status in this build
Post-quantum encryption (ML-KEM-1024, Kyber, Dilithium, SPHINCS+)	**Illustrative labels only.** No cryptographic operations are performed.
Zero-trust enforcement / trust scores	**Simulated values.** No policy engine is running.
Encrypted A2A channels	**Mock messages.** No network transport exists.
Self-healing actions	**Simulated events.** No remediation is executed.

Do **not** rely on this build to protect real data, credentials, or transactions. The target controls are described
in ARCHITECTURE.md and docs/THREAT_MODEL.md.

Supported Versions

Version	Supported
`main`	✅
Older commits, tags, forks	❌

Reporting a Vulnerability

Please do not open public issues or pull requests for suspected vulnerabilities.

1. Use GitHub's **Security → Report a vulnerability** (private vulnerability reporting), or
2. Email **adonis-jimenez@outlook.com** with subject `[ALPHA SECURITY]`.

Include: description, impact, reproduction steps, proof of concept (logs/screenshots), affected commit, and any
suggested mitigation.

Response targets

Stage	Target
Acknowledgement	3 business days
Initial triage & severity	7 business days
Fix or mitigation plan	30 days for High/Critical
Public disclosure	Coordinated, after a fix is available (default 90 days)

In Scope

- XSS, injection, or unsafe rendering in any module (e.g., chat, JSON payload viewer, memory browser)
- Secrets or credentials committed to the repository
- Dependency vulnerabilities with a reachable exploit path
- Misleading security claims that could cause unsafe adoption
- CI/CD workflow weaknesses (e.g., `pull_request_target` misuse, over-privileged tokens)
- Prompt-injection vectors once live LLM features ship

Out of Scope

- The absence of controls explicitly documented above as not yet implemented
- Denial of service via excessive traffic
- Social engineering of maintainers
- Findings requiring a compromised device or browser

Safe Harbor

Good-faith research that follows this policy, avoids privacy violations and data destruction, and gives us reasonable
time to respond will not be pursued legally.

Security Automation in this Repository

Workflow	Purpose
`ci.yml`	Lint, type-check, build
`codeql.yml`	Static analysis (JavaScript/TypeScript)
`security-scan.yml`	Gitleaks secret scan, Trivy filesystem scan, dependency review on PRs
`agentic-commerce-policy.yml`	Validates spend-mandate policy files against guardrail rules
`dependabot.yml`	Weekly dependency and Actions updates