DarkMatter

INFORMATION SECURITY OVERVIEW

Security Architecture and Controls for AI Execution Lineage

darkmatterhub.ai | hello@darkmatterhub.ai | Version 1.0

Architecture Overview

DarkMatter is a hosted SaaS platform (Node.js on Railway, PostgreSQL on Supabase) with optional self-hosted deployment for customers with data residency requirements. All context payloads are transmitted over TLS 1.3 and stored encrypted at rest. Enterprise customers may supply their own AES-256-GCM encryption key (BYOK) so that DarkMatter stores only ciphertext.

Security Controls

Control	Implementation	Status
Encryption in transit	TLS 1.3 enforced on all endpoints and dashboard traffic. No unencrypted connections accepted.	**Active**
Encryption at rest	AES-256 via Supabase managed encryption on all stored data.	**Active**
BYOK encryption	Client-provided AES-256-GCM key. DarkMatter stores only ciphertext. Key is never stored on DarkMatter systems.	**Enterprise**
Authentication	Supabase JWT for dashboard access. Bearer token API keys for agent API. Keys are hashed at rest and never stored in plaintext.	**Active**
Rate limiting	Per-endpoint limits enforced: auth 20 req/15min, API 120 req/min, feedback 5 req/hr.	**Active**
SSRF protection	Webhook URLs validated against blocklist of private IP ranges, loopback addresses, and cloud metadata endpoints.	**Active**
Security headers	helmet.js applied: HSTS, X-Frame-Options, X-Content-Type-Options, referrer policy.	**Active**
Row-level security	Supabase RLS policies enforce tenant isolation at the database layer. Users can only access their own agents and commits.	**Active**
Tamper-evident chain	SHA-256 parent hash chaining. Each commit hashes its payload plus the parent integrity hash. Modification of any node breaks all downstream hashes.	**Active**
Input sanitization	All user input sanitized and length-limited before processing or storage. HTML injection prevented.	**Active**
Secret management	All secrets stored in Railway environment variables. No secrets in source code or version control.	**Active**

Shared Responsibility Model

Security responsibilities are divided between DarkMatter and the customer.

Responsibility	DarkMatter	Customer
Platform infrastructure and hosting security	**Yes**	

Encryption in transit and at rest	**Yes**	
Database security and backups	**Yes**	
Rate limiting and abuse prevention	**Yes**	
API key hashing and secure storage	**Yes**	
BYOK key custody, storage, and rotation		**Customer**
Agent API key rotation and revocation		**Customer**
Secure storage of API keys in customer systems		**Customer**
Content and classification of committed payloads		**Customer**
Agent authorization and access control logic		**Customer**
Network security for self-hosted deployments		**Customer**

Incident Response and Vulnerability Disclosure

Security issues should be reported to hello@darkmatterhub.ai. DarkMatter is working toward SOC 2 Type 2 certification.