Security Audit Report: Daniel_AI SafeCore

Executive Summary

The security audit for Daniel_AI SafeCore has been completed. The system's critical fail-safe protocols were verified through the "Patient Zero" simulation, and the API Gateway's security boundaries were validated. A critical routing configuration issue was identified and remediated during the audit.

Verification Scenarios

1. Patient Zero Simulation

Objective: Verify the "Fail-Safe Protocol" for data protection (Encryption, Tokenization, Access Control). **Result**: ✅ **PASSED**

- **Identity Tokenization**: Validated (UUIDs replaced PII).
- **Data Encryption**: Validated (AES-256-GCM encryption verified).
- **Access Control**: Validated (MFA-enforced access allowed; integrity checks passed).

2. API Gateway Security

Objective: Verify input sanitization and strict routing in the Edge Layer. **Result**: ✅ **PASSED** (After Remediation)

- **Unauthorized Access**: Correctly blocked (403).
- **Authorized Access**: Successfully ingested data (201).
- **Threat Detection**: Malicious payload (`<script>`) was intercepted and blocked by the AI Purifier.

3. Strict Compliance & Fail-Safe Integration

Objective: Verify "Strict Compliance" (Hash Chaining) and "Fail-Safe" (System Lockdown). **Result**: ✅ **PASSED**

- **Tamper Evidence**: Modification of an in-memory log entry was immediately detected by `verifyChainIntegrity()` .
- **System Lockdown**: Integrity failure triggered an immediate `SYSTEM_STATES.LOCKDOWN` .
- **Access Denial**: Subsequent operations were blocked with `System is in LOCKDOWN` .

4. Phase 2: Hardening & Infrastructure

Objective: Validate Advanced Defense (AI Purifier), Key Rotation, and Infrastructure. **Result**: ✅ **PASSED**

- **AI Purifier**: Scoring System verified. Blocked:
  - SQL Injection (Score: 0.8)
  - XSS (Score: 0.9)
  - Mass PHI Leakage (Score: 0.7 - Threshold Adjusted)
- **Key Rotation**: Automated rotation without downtime verified.
  - Data encrypted with V1 was readable after rotating to V2.
- **Infrastructure**: Terraform code audited manually.
  - **VPC**: Enforces private subnet isolation.
  - **Security Groups**: Ingress restricted to internal mTLS port (8443) only. No public ingress.
- **Design Compliance**: Corporate "Daniel-AI" Branding integrated into Gateway UI.

5. Phase 3: Advanced Security "10/10"

Objective: Prove "Holy Grail" capabilities: Context Awareness, Predictive Defense, Automated Scheduling, and Chaos Resilience. **Result**: ✅ **PASSED**

- **Context-Aware Purification**:
    - Blocked binary/base64 payload injected into a `name` field (Context Violation).
- **Predictive Observability**:
    - `SecurityAnalyst` detected Brute Force pattern (>5 failures/min) and **preemtively locked down** the system.
- **Chaos Engineering (Fail-Safe)**:
    - Simulated "Vault Service Death". System correctly **Failed Closed** (500 Error), writing 0 bytes to storage.
- **Automated Key Scheduler**:
    - Simulated passage of 35 days. Scheduler correctly triggered Master Key Rotation.

6. Phase 4: DevSecOps Code Scanner

Objective: Automate the analysis of new code to detect potentially dangerous lines. **Result**: ✅ **PASSED**

- **Static Analysis Engine**: Created `tools/secure_scanner.js` .
- **Coverage**: Scans for `eval()` , `exec()` , AWS Secrets (`AKIA...`), Private Keys, and Weak Crypto (`md5`).
- **Verification**: Ran scan on codebase. Result: **0 Critical/High Issues**.
- **Enforcement**: Ready for integration into pre-commit hooks or CI/CD pipelines.

Findings & Remediation

[CRITICAL] Improper Route Nesting in Gateway

> *[!WARNING] The clinical ingestion route was exposed at the root level (`/clinical/ingest`), bypassing the security middleware mounted at `/api` .*

- **Impact**: Potential for unauthenticated data ingestion if the attacker guessed the route.
- **Remediation**: Updated `src/gateway/server.js` to correctly prefix the route with `/api` , ensuring it sits behind the logic layer security boundary.
- **Status**: **FIXED**

Conclusion

The Daniel_AI SafeCore logic and data layers are functioning as designed. The Gateway is now correctly enforcing security policies.

Next Steps

- Implement key rotation drills.
- Expand AI Purifier ruleset.