TOSCA Security Threat Model

Architecture Diagrams

Document Version: 1.0 **Date:** 2025-11-04 **Status:** Research Mode (v0.9.12-alpha) - Security Hardening Required for Production **Methodology:** STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege)

Executive Summary

Current Security Posture (v0.9.12-alpha): - **WARNING:** System NOT secure for clinical use or PHI storage - Database: UNENCRYPTED (plaintext SQLite) - Authentication: NOT IMPLEMENTED (no user login) - Authorization: NOT IMPLEMENTED (no role-based access control) - Communication: Local only (no network exposure) - Audit Trail: IMPLEMENTED (immutable event logging)

Risk Level: HIGH for production deployment, **ACCEPTABLE** for research-only use in controlled laboratory environment.

Required for Clinical Use (Phase 6+): 1. Database encryption (SQLCipher with AES-256-GCM) 2. User authentication (password hashing with bcrypt/argon2) 3. Role-based access control (technician, researcher, admin roles) 4. Session management (secure token-based sessions) 5. Audit log encryption (encrypted JSONL files)

System Overview

TOSCA Attack Surface:

```
User Interface (PyQt6 Desktop Application)
Attack Surface: Local physical access

                ▼
Application Core (Python 3.12)
Attack Surface: Code injection, logic flaws

                ▼
```

```
Hardware Abstraction Layer
Attack Surface: Serial port injection


              ▼
Physical Hardware (USB/Serial devices)
Attack Surface: Hardware tampering



Data Storage
- tosca.db (UNENCRYPTED SQLite)
- JSONL logs (PLAINTEXT)
- Session videos/images (UNENCRYPTED)
Attack Surface: File system access
```

Trust Boundaries: 1. **Operating System ↔ TOSCA Application:** Trust OS file permissions, user authentication 2. **TOSCA Application ↔ Hardware Devices:** Trust USB/serial device identity (no device authentication) 3. **TOSCA Application ↔ File System:** Trust file system permissions (no file encryption) 4. **User ↔ TOSCA Application:** NO TRUST (no user authentication implemented)

STRIDE Threat Analysis

1. Spoofing Identity

Threat 1.1: No User Authentication

Severity: CRITICAL **Status:** OPEN (Phase 6 mitigation required)

Description: - Anyone with physical access can operate the system - No password protection, no user login screen - No distinction between technician/researcher/unauthorized user

Attack Scenario: 1. Unauthorized person sits at TOSCA workstation 2. Application launches without authentication 3. Attacker can create subjects, run treatments, modify data

Impact: - Unauthorized laser treatments - Data integrity compromise - Regulatory non-compliance (21 CFR Part 11) - HIPAA violation if PHI stored

Likelihood: HIGH (physical access to lab required, but no technical barrier)

Mitigation (Phase 6): - Implement user authentication with password hashing (bcrypt/argon2) - Require login on application startup - Implement inactivity timeout (auto-logout after 15 minutes) - Store hashed passwords in encrypted database - Enforce password complexity requirements

Current Mitigation: - Physical access control to laboratory - System labeled “RESEARCH MODE - AUTHORIZED PERSONNEL ONLY”

Threat 1.2: No Hardware Device Authentication

Severity: MEDIUM **Status:** ACCEPTED (hardware supply chain trusted)

Description: - No verification of hardware device identity - Malicious device could be connected via USB/serial - System trusts any device on correct COM port

Attack Scenario: 1. Attacker replaces legitimate Arduino Uno with malicious clone 2. Malicious firmware sends false interlock signals 3. System believes safety interlocks are satisfied

Impact: - Safety interlock bypass - False photodiode readings - Laser operation in unsafe conditions

Likelihood: LOW (requires physical access, hardware knowledge, supply chain compromise)

Mitigation (Future): - Implement hardware device signature verification (challenge-response protocol) - Store known device serial numbers in encrypted database - Verify firmware version/checksum on connection

Current Mitigation: - Hardware supply chain controlled (direct purchase from manufacturers) - Laboratory access restricted - Visual inspection of hardware during setup

---

2. Tampering

Threat 2.1: Database Tampering (File System Access)

Severity: CRITICAL **Status:** OPEN (Phase 6 mitigation required)

Description: - `tosca.db` is plaintext SQLite file - Anyone with file system access can modify database directly - No database encryption, no integrity verification

Attack Scenario: 1. Attacker gains file system access (local admin, stolen backup drive) 2. Opens `tosca.db` with SQLite client 3. Modifies treatment events, deletes safety logs, changes subject data

Impact: - Audit trail compromise (regulatory non-compliance) - Data integrity loss (research validity questionable) - Evidence destruction (safety incidents concealed)

Likelihood: MEDIUM (requires file system access, but technically trivial)

Mitigation (Phase 6): - Migrate to SQLCipher (AES-256-GCM encryption) - Implement database integrity verification (HMAC signatures) - Store encryption key in hardware security module (HSM) or Windows DPAPI - Enable SQLite write-ahead logging (WAL) for corruption recovery - Implement backup integrity verification (checksums)

Current Mitigation: - Operating system file permissions (Windows NTFS) - Laboratory access control

---

Threat 2.2: JSONL Log File Tampering

Severity: HIGH **Status:** OPEN (Phase 6 mitigation required)

Description: - JSONL log files (photodiode, camera metadata) stored as plaintext - Append-only by software convention, not enforced - No file integrity verification

Attack Scenario: 1. Attacker gains file system access 2. Edits JSONL files to remove safety fault events 3. Modifies photodiode readings to conceal power excursions

Impact: - High-frequency safety data compromised - Audit trail incomplete - Safety incident evidence destroyed

Likelihood: MEDIUM (requires file system access)

Mitigation (Phase 6): - Encrypt JSONL files (AES-256-GCM, per-file keys) - Implement file integrity verification (HMAC per line) - Use OS-level file locking (prevent modification after session closure) - Consider WORM storage for audit logs (write-once, read-many)

Current Mitigation: - Operating system file permissions - Immutable append-only design (software enforced)

Threat 2.3: Arduino Firmware Tampering

Severity: HIGH **Status:** OPEN (future mitigation)

Description: - Arduino Uno firmware can be reflashed via USB - No firmware integrity verification on boot - Malicious firmware could bypass safety interlocks

Attack Scenario: 1. Attacker gains physical access to Arduino 2. Reflashes firmware with malicious code that reports “all interlocks OK” regardless of actual state 3. System believes safety interlocks satisfied when footpedal released

Impact: - Safety interlock bypass - Laser operation without deadman switch - Patient injury risk

Likelihood: LOW (requires physical access, hardware knowledge, Arduino ISP equipment)

Mitigation (Future): - Implement bootloader signature verification (signed firmware) - Use microcontroller with secure boot (e.g., ESP32-S3 with flash encryption) - Detect firmware version on connection, alert if unexpected version - Physical tamper-evident seals on Arduino enclosure

Current Mitigation: - Laboratory access control - Visual inspection during setup - Firmware version logged at system startup

---

3. Repudiation

Threat 3.1: No Non-Repudiation for User Actions

Severity: HIGH **Status:** OPEN (Phase 6 mitigation required)

Description: - No user authentication → no attribution of actions - Event logs record “tech_id” but no verification of tech identity - Operator can claim “someone else used my account”

Attack Scenario: 1. Technician performs unauthorized treatment 2. Claims “I didn’t do that, someone else was logged in” 3. No way to prove who actually performed action

Impact: - Regulatory non-compliance (21 CFR Part 11 requires electronic signatures) - Inability to attribute actions to individuals - Legal liability in adverse events

Likelihood: MEDIUM (requires malicious insider)

Mitigation (Phase 6): - Implement user authentication with unique usernames - Require password re-entry for critical actions (laser arming) - Implement electronic signatures for protocol approval - Log all user actions with authenticated user ID - Enforce session timeout to prevent unattended sessions

Current Mitigation: - Tech ID entry (honor system, no verification) - Event logging with tech_id (not cryptographically signed)

Threat 3.2: Event Log Modification (Covered in Tampering 2.1, 2.2)

Severity: CRITICAL **Status:** OPEN (Phase 6 mitigation required)

Description: - Event logs not cryptographically signed - Attacker with database access can modify/delete events - No proof logs haven’t been tampered with

Mitigation: See Threat 2.1 and 2.2

---

4. Information Disclosure

Threat 4.1: Database Encryption Missing

Severity: CRITICAL **Status:** OPEN (Phase 6 mitigation required)

Description: - `tosca.db` is plaintext SQLite - Contains subject demographics (if PHI entered) - Readable by anyone with file system access

Attack Scenario: 1. Attacker steals laptop/hard drive 2. Mounts drive on external system 3. Opens `tosca.db` and reads all subject data

Impact: - HIPAA violation if PHI stored (fines up to $1.5M/year) - Subject privacy breach - Regulatory non-compliance

Likelihood: MEDIUM (requires theft/unauthorized access to physical media)

Mitigation (Phase 6): - Migrate to SQLCipher (AES-256-GCM) - Implement full disk encryption (Windows BitLocker) - Store encryption keys in hardware security module (HSM) - Implement key rotation policy (annually)

Current Mitigation: - System labeled “RESEARCH MODE - NO PHI” - Subject codes used (not patient names) - Operating system file permissions

---

Threat 4.2: Video/Image Files Unencrypted

Severity: HIGH **Status:** OPEN (Phase 6 mitigation required)

Description: - Session videos and images stored as unencrypted files - May contain identifiable anatomy or patient information

Attack Scenario: 1. Attacker gains file system access 2. Copies session videos/images 3. Identifies patients from anatomical features

Impact: - HIPAA violation - Subject privacy breach - Research ethics violation

Likelihood: MEDIUM

Mitigation (Phase 6): - Encrypt video files (AES-256-GCM) - Implement full disk encryption (BitLocker) - Consider file-level encryption for session folders - Implement access logging (who accessed which files)

Current Mitigation: - “RESEARCH MODE - NO PHI” labeling - Operating system file permissions - Laboratory access control

---

Threat 4.3: Application Memory Disclosure

Severity: LOW **Status:** ACCEPTED (low likelihood)

Description: - Application memory may contain sensitive data (subject IDs, session data) - Memory dumps could expose data if system crashes

Attack Scenario: 1. Application crashes during treatment 2. Crash dump written to disk 3. Attacker analyzes dump to extract subject data

Impact: - Limited information disclosure (single session data) - Unlikely to contain PHI in research mode

Likelihood: LOW

Mitigation (Future): - Disable automatic crash dump generation - Implement memory scrubbing for sensitive variables - Use Windows Data Execution Prevention (DEP) and Address Space Layout Randomization (ASLR)

Current Mitigation: - “RESEARCH MODE - NO PHI” - Limited sensitive data in memory

---

5. Denial of Service

Threat 5.1: USB Device Flooding

Severity: LOW **Status:** ACCEPTED (requires physical access)

Description: - Attacker connects malicious USB device - Floods USB bus with traffic - Causes camera/Arduino communication failure

Attack Scenario: 1. Attacker plugs in malicious USB device during treatment 2. USB bus saturated with traffic 3. Camera frame drops, Arduino watchdog timeout

Impact: - Treatment interruption (safety interlock failure) - Laser disabled (selective shutdown) - Minimal patient impact (safety preserved)

Likelihood: LOW (requires physical access, specialized hardware)

Mitigation: - Physical USB port access control (disable unused ports) - Laboratory access restriction

---

Threat 5.2: Disk Space Exhaustion

Severity: MEDIUM **Status:** PARTIALLY MITIGATED

Description: - Video recording consumes disk space rapidly (30 FPS, 1280x960) - No automatic cleanup of old session data - Disk full → application crash or inability to log

Attack Scenario: 1. System runs for extended period without disk cleanup 2. Hard drive fills with session videos 3. Application cannot write new events → crash or data loss

Impact: - Event logging failure - Application crash mid-treatment - Data loss (events not logged)

Likelihood: MEDIUM (will occur eventually without manual cleanup)

Mitigation (Current): - Disk space monitoring (manual check before sessions) - Operator training (monthly cleanup of old data)

Mitigation (Future): - Implement automatic disk space monitoring (alert at 80% full) - Implement automatic session folder archival (compress old sessions) - Implement session data retention policy (auto-delete after X years)

---

6. Elevation of Privilege

Threat 6.1: No Role-Based Access Control

Severity: HIGH **Status:** OPEN (Phase 6 mitigation required)

Description: - All users have full system access - No distinction between technician, researcher, administrator - Any user can modify protocols, delete data, change settings

Attack Scenario: 1. Research assistant (should have read-only access) launches TOSCA 2. Modifies treatment protocol 3. Technician unknowingly uses modified protocol on subject

Impact: - Unauthorized protocol modifications - Data deletion by non-administrators - Regulatory non-compliance

Likelihood: MEDIUM (requires malicious or careless insider)

Mitigation (Phase 6): - Implement role-based access control (RBAC): - **Technician:** Run treatments, create

sessions, view data - **Researcher:** Read-only access to data, export reports - **Administrator:** Full access, user management, protocol approval - Require elevated privileges for sensitive actions (protocol modification, data deletion) - Implement audit logging of privilege escalation

Current Mitigation: - Laboratory access control (only trained technicians) - "RESEARCH MODE" labeling

---

Threat 6.2: Python Code Injection (SQL Injection, Command Injection)

Severity: HIGH **Status:** MITIGATED (parameterized queries, input validation)

Description: - User input fields (subject code, session notes) could contain malicious code - SQL injection possible if queries not parameterized - Command injection possible if user input passed to shell commands

Attack Scenario: 1. Attacker enters `'; DROP TABLE subjects; --` as subject code 2. If SQL query not parameterized, database table deleted

Impact: - Database corruption - Data loss - Application crash

Likelihood: LOW (requires malicious input, mitigated by SQLAlchemy ORM)

Mitigation (Current): - SQLAlchemy ORM used (automatic parameterized queries) - Input validation for all user fields (regex whitelisting) - No direct SQL query construction from user input - No shell command execution with user-controlled parameters

Best Practice Enforcement:

```
# SECURE (SQLAlchemy ORM - parameterized)
subject = session.query(Subject).filter_by(subject_code=user_input).first()

# INSECURE (direct SQL - NEVER USE)
# query = f"SELECT * FROM subjects WHERE subject_code = '{user_input}'"
```

Security Controls Summary

Threat Category	Current Controls	Phase 6 Required Controls
Spoofing	Physical access control	User authentication (bcrypt/argon2), session management
Tampering	OS file permissions	Database encryption (SQLCipher), integrity verification (HMAC)
Repudiation	Event logging (unsigned)	Electronic signatures, authenticated user IDs
Information Disclosure	"NO PHI" policy	Database encryption, full disk encryption (BitLocker)
Denial of Service	Manual disk monitoring	Automatic disk monitoring, archival, retention policy
Elevation of Privilege	Laboratory access control	Role-based access control (RBAC), privilege auditing

---

Phase 6 Security Roadmap

Priority 1 (CRITICAL - Required for Clinical Use): 1. **Database Encryption:** Migrate to SQLCipher with AES-256-GCM 2. **User Authentication:** Implement login with password hashing (bcrypt/argon2) 3. **Full Disk Encryption:** Enable Windows BitLocker on all TOSCA workstations

Priority 2 (HIGH - Recommended for Clinical Use): 4. **Role-Based Access Control:** Implement technician/researcher/admin roles 5. **Audit Log Encryption:** Encrypt JSONL files with AES-256-GCM 6.

Electronic Signatures: Implement 21 CFR Part 11 compliant e-signatures

Priority 3 (MEDIUM - Enhanced Security): 7. **Hardware Device Authentication:** Implement Arduino firmware signature verification 8. **Session Management:** Auto-logout after inactivity, session token expiration 9. **Disk Space Monitoring:** Automatic alerts and archival

Priority 4 (LOW - Future Enhancements): 10. **Hardware Security Module (HSM):** Store encryption keys in hardware token 11. **Tamper-Evident Seals:** Physical seals on Arduino and PC enclosures 12. **Network Isolation:** Air-gapped network for clinical deployment

---

Compliance Requirements

FDA 21 CFR Part 11 (Electronic Records; Electronic Signatures)

- **MISSING:** User authentication, electronic signatures
- **IMPLEMENTED:** Audit trail (needs encryption + integrity verification)

HIPAA (Health Insurance Portability and Accountability Act)

- **MISSING:** Database encryption, access control, audit log encryption
- **CURRENT STATUS:** "NO PHI" policy prevents HIPAA violations in research mode

IEC 62304 (Medical Device Software Lifecycle)

- **IMPLEMENTED:** Software architecture, risk management, verification/validation
- **MISSING:** Security risk analysis (THIS DOCUMENT addresses gap)

Document Owner: Security Engineer **Last Updated:** 2025-11-04 **Next Review:** Upon Phase 6 (Pre-Clinical Validation) initiation