Fox ML Infrastructure — Security Controls Matrix

This document provides a concise summary of security controls implemented across Fox ML Infrastructure.
This matrix is designed for enterprise security reviews and SOC2-adjacent compliance assessments.

---

1. Executive Summary

Fox ML Infrastructure implements security controls appropriate for a client-hosted software platform with zero data processing.

Key characteristics: - **Client-hosted architecture** — Software runs on client infrastructure (no vendor infrastructure) - **Zero data processing** — No vendor data collection, storage, or processing - **Minimal attack surface** — Limited vendor infrastructure reduces attack surface - **Defense in depth** — Multiple layers of security controls

---

2. Access Control

2.1 Authentication

Authentication mechanisms:

- [OK] **GitHub authentication** — Two-factor authentication (2FA) required for repository access
- [OK] **SSH key authentication** — SSH keys for secure repository access
- [OK] **Email authentication** — Standard email authentication for support communications
- [OK] **Credential management** — Strong password policies and credential rotation

Access control principles: - **Principle of least privilege** — Access granted only to minimum required - **Multi-factor authentication** — 2FA required for critical systems - **Regular credential rotation** — Credentials rotated regularly

2.2 Authorization

Authorization mechanisms:

- [OK] **Repository-level permissions** — GitHub repository-level access controls
- [OK] **Role-based access** — Access based on commercial license tier
- [OK] **Client-specific repositories** — Isolated repositories for client-specific code
- [OK] **Read-only access** — Read-only access for certain documentation repositories

Authorization principles: - **Separation of duties** — Clear separation between development and client access - **Access reviews** — Regular review of access permissions - **Immediate revocation** — Immediate revocation upon termination or breach

---

3. Encryption

3.1 Encryption at Rest

Data encryption at rest:

- [OK] **Client data** — Client data encrypted by client (vendor does not store client data)
- [OK] **Repository encryption** — GitHub provides encryption at rest for repositories
- [OK] **Local backups** — Local backups encrypted (if applicable)
- [OK] **Email encryption** — Email stored in encrypted email systems

Note: Since vendor does not store client data, encryption at rest requirements are minimal.

3.2 Encryption in Transit

Data encryption in transit:

- [OK] **HTTPS/TLS** — All web traffic encrypted via HTTPS/TLS
- [OK] **SSH encryption** — Repository access encrypted via SSH
- [OK] **Email encryption** — Email communications encrypted (TLS)
- [OK] **API encryption** — All API communications encrypted (if applicable)

Encryption standards: - **TLS 1.2+** — Minimum TLS 1.2 for all encrypted communications - **Strong ciphers** — Strong cipher suites enabled - **Certificate validation** — Proper certificate validation

---

4. Logging and Monitoring

4.1 Logging

Logging capabilities:

- [OK] **Repository access logs** — GitHub provides access logs for repository access
- [OK] **Code change logs** — Git provides complete change history
- [OK] **Email logs** — Email systems provide communication logs
- [OK] **Application logs** — Software includes structured logging (client-managed)

Logging principles: - **Comprehensive logging** — Log all significant events - **Structured logging** — Structured log format for parsing - **Log retention** — Retain logs per retention policy - **Log integrity** — Protect logs from tampering

4.2 Monitoring

Monitoring capabilities:

- [OK] **Repository monitoring** — Monitor for unauthorized access or changes
- [OK] **Security alerts** — GitHub security alerts for vulnerabilities
- [OK] **Dependency monitoring** — Monitor dependencies for security vulnerabilities
- [OK] **Client reporting** — Clients can report security concerns

Monitoring principles: - **Continuous monitoring** — Monitor systems continuously - **Alert mechanisms** — Alert on suspicious activity - **Incident response** — Integrate with incident response procedures

---

5. Secrets Management

5.1 Secret Storage

Secret storage practices:

- [OK] **No hardcoded secrets** — No secrets hardcoded in source code
- [OK] **Environment variables** — Secrets stored in environment variables
- [OK] **Client-controlled** — Clients manage their own secrets
- [OK] **GitHub secrets** — GitHub secrets for CI/CD (if applicable)

Secret management principles: - **No secrets in code** — Never commit secrets to repositories - **Secret rotation** — Rotate secrets regularly - **Access control** — Limit access to secrets - **Audit trails** — Audit secret access

5.2 Secret Handling

Secret handling practices:

- [OK] **Secure transmission** — Secrets transmitted securely (encrypted channels)
- [OK] **No logging** — Secrets never logged
- [OK] **Client responsibility** — Clients responsible for their own secret management
- [OK] **Consulting secrets** — Consulting secrets handled per security policy

---

6. Network Security

6.1 Network Segmentation

Network segmentation:

- [OK] **Client-hosted** — Software runs on client networks (vendor has no network access)
- [OK] **No vendor network** — No vendor-managed network infrastructure
- [OK] **Isolated repositories** — Client repositories isolated from each other
- [OK] **No cross-client access** — No network access between client environments

Network security principles: - **Client-controlled** — All network security is client-controlled - **No vendor access** — Vendor has no network access to client systems - **Isolation** — Client environments isolated from each other

6.2 Network Monitoring

Network monitoring:

- [OK] **Client-managed** — Network monitoring is client-managed
- [OK] **No vendor monitoring** — Vendor does not monitor client networks
- [OK] **No data exfiltration** — No capability for data exfiltration

---

7. Package Integrity

7.1 Code Integrity

Code integrity controls:

- [OK] **Version control** — All code in version control (Git)
- [OK] **Signed commits** — Git commit signing (if applicable)
- [OK] **Tagged releases** — All releases tagged with semantic versioning
- [OK] **Audit trail** — Complete audit trail of all code changes

Code integrity principles: - **Immutable tags** — Release tags are immutable - **Change tracking** — All changes tracked in version control - **Code review** — Code reviewed before release - **Integrity verification** — Verify code integrity before deployment

7.2 Supply Chain Integrity

Supply chain integrity:

- [OK] **Explicit dependencies** — All dependencies explicitly declared
- [OK] **Dependency scanning** — Scan dependencies for vulnerabilities
- [OK] **No telemetry** — No outbound calls or telemetry
- [OK] **No embedded trackers** — No third-party tracking scripts
- [OK] **Open-source transparency** — Core platform open-source (AGPL-3.0)

Supply chain principles: - **Explicit dependencies** — No hidden dependencies - **Vulnerability scanning** — Regular vulnerability scanning - **No external calls** — No unauthorized external calls - **Transparency** — Transparent supply chain

---

8. Vulnerability Management

8.1 Vulnerability Detection

Vulnerability detection:

- [OK] **Dependency scanning** — Scan dependencies for known vulnerabilities
- [OK] **Code review** — Code review for security issues
- [OK] **Security alerts** — GitHub security alerts
- [OK] **Client reporting** — Clients can report vulnerabilities

8.2 Vulnerability Response

Vulnerability response:

- [OK] **Immediate assessment** — Assess vulnerabilities immediately
- [OK] **Patch releases** — Release security patches promptly
- [OK] **Client notification** — Notify clients of security issues
- [OK] **Incident response** — Follow incident response procedures

Vulnerability management principles: - **Rapid response** — Respond to vulnerabilities rapidly - **Patch management** — Release patches promptly - **Client communication** — Communicate

with clients transparently - **Continuous improvement** — Continuously improve vulnerability management

---

9. Incident Response

9.1 Incident Detection

Incident detection:

- [OK] **Monitoring** — Continuous monitoring for security incidents
- [OK] **Client reports** — Client reports of security concerns
- [OK] **Third-party notifications** — Notifications from service providers
- [OK] **Security audits** — Periodic security reviews

9.2 Incident Response

Incident response:

- [OK] **Incident response plan** — Documented incident response plan
- [OK] **Response procedures** — Clear response procedures
- [OK] **Client notification** — Timely client notification
- [OK] **Remediation** — Effective remediation procedures

See `legal/INCIDENT_RESPONSE_PLAN.md` for detailed incident response procedures.

---

10. Business Continuity

10.1 Backup and Recovery

Backup and recovery:

- [OK] **Repository backups** — GitHub provides repository redundancy
- [OK] **Local backups** — Local backups of critical repositories
- [OK] **Email backups** — Email systems provide redundancy
- [OK] **Documentation backups** — Documentation in version control

10.2 Business Continuity

Business continuity:

- [OK] **Business continuity plan** — Documented business continuity plan
- [OK] **Recovery procedures** — Clear recovery procedures
- [OK] **RTO/RPO targets** — Defined recovery time and point objectives
- [OK] **Client communication** — Communication during disruptions

See `legal/BUSINESS_CONTINUITY_PLAN.md` for detailed business continuity procedures.

---

11. Compliance and Audit

11.1 Compliance

Compliance controls:

- [OK] **GDPR principles** — Adheres to GDPR principles
- [OK] **CCPA principles** — Adheres to CCPA principles
- [OK] **Export compliance** — Complies with export control regulations
- [OK] **Data protection** — Data protection and privacy controls

11.2 Audit

Audit controls:

- [OK] **Audit trails** — Complete audit trails of all activities
- [OK] **Documentation** — Comprehensive security documentation
- [OK] **Access logs** — Access logs for audit purposes
- [OK] **Change logs** — Change logs for code and configuration

---

12. Security Controls Summary

12.1 Control Categories

Security controls by category:

Category	Controls	Status
Access Control	Authentication, Authorization, Credential Management	[OK] Implemented
Encryption	Encryption at Rest, Encryption in Transit	[OK] Implemented
Logging & Monitoring	Logging, Monitoring, Alerting	[OK] Implemented
Secrets Management	Secret Storage, Secret Handling	[OK] Implemented
Network Security	Network Segmentation, Network Monitoring	[OK] Client-Controlled
Package Integrity	Code Integrity, Supply Chain Integrity	[OK] Implemented
Vulnerability Management	Vulnerability Detection, Vulnerability Response	[OK] Implemented
Incident Response	Incident Detection, Incident Response	[OK] Implemented
Business Continuity	Backup and Recovery, Business Continuity	[OK] Implemented
Compliance & Audit	Compliance, Audit	[OK] Implemented

12.2 Control Effectiveness

Control effectiveness:

- **High effectiveness** — Access control, encryption, logging, package integrity
- **Medium effectiveness** — Vulnerability management, incident response
- **Client-dependent** — Network security, secrets management (client-controlled)

---

13. Continuous Improvement

13.1 Security Enhancements

Security enhancements:

- **Regular reviews** — Regular security reviews and assessments
- **Process improvements** — Continuous improvement of security processes
- **Tooling enhancements** — Enhance security tooling and monitoring
- **Training** — Security training and awareness (if applicable)

13.2 Maturity Progression

Security maturity:

- **Current state** — Appropriate for client-hosted software platform
- **Future enhancements** — SOC2 certification (if applicable), enhanced monitoring
- **Scalability** — Controls designed to scale with business growth

---

14. Contact

For security controls questions:

Jennifer Lewis
Fox ML Infrastructure LLC
Email: **jenn.lewis5789@gmail.com**
Subject: *Security Controls Inquiry — Fox ML Infrastructure*

---

15. Related Documents

- `legal/SECURITY.md` — Security statement
- `legal/INFOSEC_SELF_ASSESSMENT.md` — Information security self-assessment
- `legal/INCIDENT_RESPONSE_PLAN.md` — Incident response plan
- `legal/BUSINESS_CONTINUITY_PLAN.md` — Business continuity plan

---

16. Summary

Key Security Controls:

1. [OK] **Access Control** — Strong authentication and authorization
2. [OK] **Encryption** — Encryption at rest and in transit
3. [OK] **Logging & Monitoring** — Comprehensive logging and monitoring

4. [OK] **Secrets Management** — Secure secret storage and handling
5. [OK] **Network Security** — Client-controlled network security
6. [OK] **Package Integrity** — Code and supply chain integrity
7. [OK] **Vulnerability Management** — Vulnerability detection and response
8. [OK] **Incident Response** — Documented incident response procedures
9. [OK] **Business Continuity** — Backup and recovery procedures
10. [OK] **Compliance & Audit** — Compliance and audit controls

This matrix provides a comprehensive summary of security controls for enterprise security reviews.