C9AI Enterprise Security Assessment

Executive Summary

C9AI is designed with enterprise security and privacy as core principles. The application uses only well-established, audited dependencies and processes all AI operations locally, ensuring no sensitive data leaves your network.

External Dependencies Analysis

Direct Dependencies (8 total)

Package	Version	Purpose	Security Profile
chalk	^4.1.2	Terminal styling	50M+ weekly downloads, MIT license, no network access
chokidar	^3.5.3	File system monitoring	Core Node.js ecosystem, 40M+ downloads, widely audited
commander	^11.1.0	CLI framework	40M+ downloads, used by major projects (npm, git)
fs-extra	^11.2.2	Enhanced file operations	50M+ downloads, extends Node.js fs module
inquirer	^8.2.4	Interactive CLI prompts	20M+ downloads, terminal I/O only
node-fetch	^2.7.0	HTTP client	30M+ downloads, Node.js standard library
node-llama-cpp	^3.11.0	Local LLM runtime	Privacy-focused, **no external network calls**
yaml	^2.3.4	Configuration parsing	40M+ downloads, parsing only

Security Audit Results

- **0 vulnerabilities** found (npm audit)
- All dependencies are **mature, widely-used packages**
- No packages with known security issues
- All packages have **MIT or similar permissive licenses**

Privacy & Data Security

Local-First Architecture

- **AI processing**: 100% local via node-llama-cpp
- **No telemetry**: Zero analytics or usage tracking

- **No external API calls**: Except for optional cloud AI providers (user-controlled)
- **File operations**: Limited to user-specified directories only

Network Activity

- **Local LLM mode**: Zero network activity
- **Cloud AI mode**: Only when explicitly configured by user
- **No background connections**: No automatic updates or phone-home features

Data Handling

- **Configuration**: Stored locally in user directories
- **Models**: Downloaded once, cached locally
- **Processing**: All AI inference happens on-premises
- **Logs**: Local only, no external transmission

Compliance Considerations

Licenses

- **C9AI**: MIT License - allows enterprise use and modification
- **All dependencies**: MIT/ISC licenses - enterprise-friendly
- **No GPL**: No copyleft restrictions

Audit Trail

- Source code is fully open and auditable
- Dependency tree is transparent and minimal
- Build process is reproducible

Air-Gap Compatibility

- Can operate without internet after initial setup
- Local LLM models work offline
- No mandatory cloud dependencies

Enterprise Deployment Options

Installation Methods

1. **NPM Package**: Standard Node.js installation
2. **Standalone Executable**: Self-contained binary (coming soon)
3. **MSI Installer**: Windows enterprise deployment (in development)

Since we are creating an installer exe for enterprises. only the nodejs runtime is installed along with audited scripts and llama.cpp binary.

Configuration Management

- YAML-based configuration files
- Environment variable support
- No external configuration dependencies

Recommendations for Enterprise Adoption

Security Review Process

1. Review this document with your security team
2. Audit the open-source code repository
3. Run internal vulnerability scans on dependencies
4. Test in isolated environment first

Deployment Best Practices

1. Use local LLM mode for sensitive data
2. Configure appropriate file system permissions
3. Monitor resource usage during AI operations
4. Implement standard software deployment policies

Risk Mitigation

- **Low risk**: Minimal attack surface due to local-first design
- **Controlled dependencies**: Well-established packages only
- **No data exfiltration**: AI processing stays on-premises
- **Transparent operation**: Full source code availability

Support & Updates

- **Security updates**: Monitor npm audit results
- **Dependency updates**: Regular maintenance releases
- **Enterprise support**: Available through GitHub issues
- **Custom deployments**: Source code allows internal modifications

---

Document Version: 1.0
Last Updated: August 2025
Next Review: Quarterly dependency audit recommended