

User Manual: CMMC Compliance Companion

Client-Side Compliance Orchestration Utility

Document Version: 2.1

Date: November 19, 2025

Classification: Public

System Applicability: Development Workstations & Air-Gapped Production Systems

1. Introduction

1.1 Purpose

The **CMMC Compliance Companion** is a single-file, client-side web application designed to standardize the implementation of a CMMC-compliant secure development architecture. It acts as an operational bridge between high-level compliance policy documents and technical execution.

1.2 Scope

This utility is authorized for use in:

1. **Cloud Development Environments:** For generating supply chain security configurations.
2. **Air-Gapped FIPS Workstations:** For generating cryptographic keys and audit logs in an isolated environment.

1.3 Compliance Targets

This tool is hardcoded to enforce parameters compliant with:

- **NIST FIPS 186-5:** Enforces RSA-3072 / SHA-384 for all key generation commands.
- **CMMC SC.L2-3.13.8:** Standardizes cryptographic authentication for software components via Kyverno policies.
- **NIST SP 800-171:** Facilitates audit logging and configuration management.

2. Deployment & Security Architecture

2.1 Zero-Transmission Architecture

This application operates under a strict **Local-Execution Model**.

- **No Backend:** The application has no server-side component. All logic executes within the user's browser memory.
- **No Data Exfiltration:** No user input (Organization Name, Keys, Checklist Status) is transmitted over the network.
- **Persistence:** Data is stored exclusively in the browser's localStorage. Clearing browser cache wipes this data.

2.2 Installation Procedures

Scenario A: Internet-Connected (Cloud Dev)

1. Save the index.html file to a local directory.
2. Open the file using a modern web browser (Chrome, Firefox, Edge).
3. External libraries (React, Tailwind, Lucide, JSZip) are loaded via secure CDN (unpkg.com).

Scenario B: Air-Gapped (Secure Workstation)

To maintain the air-gap boundary, the application must be "localized" before transfer.

1. **Download Dependencies:** On an internet-connected machine, download the following JavaScript files:
 - react.development.js
 - react-dom.development.js
 - babel.min.js
 - lucide.js
 - tailwindcss.js
 - jszip.min.js (New in v2.1)
2. **Modify Source:** Edit index.html to reference these files locally (e.g., <script src="./react.development.js"></script>).
3. **Transfer:** Copy the folder containing the HTML and JS files to the secure workstation via the approved USB transfer procedure.
4. **Execution:** Open index.html in the offline browser.

3. Operational Modules

3.1 Module 1: MOK Tools (Machine Owner Key)

Purpose: Eliminates human error during the complex command-line generation of FIPS-compliant keys.

Inputs:

- Common Name (CN)
- Organization (O)
- Operator Name & Witness Name (for Dual Control)
- **Validity Period:** Configurable days (default: 3650) (New in v2.1)

Outputs:

1. **Key Generation Command:** A pre-formatted openssl command.
 - *Audit Control:* The algorithm is hardcoded to rsa:3072 and -sha384. The user cannot accidentally downgrade to weaker encryption.
2. **Audit Log Entry:** A formatted text block timestamped ISO-8601.
 - *Usage:* This output must be saved as MOK-AUDIT-ENTRY.txt or printed for the physical binder.

3.2 Module 2: Security Config

Purpose: Generates Infrastructure-as-Code (IaC) artifacts for supply chain security.

Features:

- **Kyverno Policy Generator:** Creates a Kubernetes ClusterPolicy to enforce image signing.
- **SLSA Workflow Generator:** Creates GitHub Actions YAML for Level 3 provenance.
- **Input Validation:** The Public Key field performs real-time validation. If the input does not contain standard PEM headers (----BEGIN PUBLIC KEY ----), the field borders turn red and an error is displayed.

3.3 Module 3: Workflow Roadmap

Purpose: A persistent interactive checklist to track compliance progress across critical phases.

Features:

- **Visual Progress Tracking:** Real-time progress bars for each phase (MOK Setup, Cloud Dev, Air-Gap Transfer). (New in v2.1)

- **Data Storage:** Progress is saved automatically to localStorage.

3.4 Global Features (New in v2.1)

- **Onboarding Wizard:** A guided tour for first-time users explaining the tool's purpose and air-gap capability.
- **Bulk Export:** A "Export All" button that packages all generated configurations, audit logs, and progress status into a single ZIP file (cmmc-export-[DATE].zip) for easy archiving.

4. Audit Procedures

4.1 Verification for Assessors

To verify the integrity of this tool during an assessment:

1. **Inspect Source:** View Page Source in the browser.
2. **Verify Hardcoding:** Search for rsa:3072. This proves the tool forces FIPS compliance.
3. **Verify Isolation:** Open the Network tab in Developer Tools. Input data into the fields. Observe that **zero** network requests are triggered by the input.

4.2 Troubleshooting

- **Issue:** Application is blank.
 - *Cause:* Script blocked by strict browser policy or missing internet connection (if using CDN version).
 - *Fix:* Use the Localized/Air-Gap version described in Section 2.2.
- **Issue:** "Invalid PEM Format" error in Security Config.
 - *Cause:* Copied key is missing headers or has extra whitespace.
 - *Fix:* Ensure the key includes the -----BEGIN... and -----END... lines exactly.
- **Issue:** Export button does not work.
 - *Cause:* jszip.min.js failed to load.
 - *Fix:* Ensure the library is present in the folder (air-gapped) or internet is accessible (cloud).

5. Version Control

Version	Date	Author	Changes
1.0	Nov 19, 2025	StackArc3	Initial Release. Includes MOK Manager.
1.1	Nov 19, 2025	StackArc3	Added Security Config.
1.2	Nov 19, 2025	StackArc3	Added PEM validation and Persistent Checklists.
2.1	Nov 19, 2025	StackArc3	Added Bulk Export (ZIP), Onboarding Wizard, Visual Progress Bars, and Customizable MOK Validity.

End of Document

nls