Secure Development Policy

Policy Owner: Director of Risk Management
Effective Date: March 21, 2024

Purpose

To ensure that information security is designed and implemented within the development lifecycle for applications and information systems.

Scope

All OnCue Technology, LLC applications and information systems that are business critical and/or process, store, or transmit Confidential data. This policy applies to all internal and external engineers and developers of OnCue Technology, LLC software and infrastructure.

Policy

This policy describes the rules for the acquisition and development of software and systems that shall be applied to developments within the OnCue Technology, LLC organization.

System Change Control Procedures

Changes to systems within the development lifecycle shall be controlled by the use of formal change control procedures. Change control procedures and requirements are described in the OnCue Technology, LLC Operations Security Policy.

All changes to a production or release branch shall be made through a pull request that is reviewed and approved by a developer other than the author before it is merged, as described in the Source Control Protections section of this policy.

Change control procedures shall ensure that development, testing and deployment of changes shall not be performed by a single individual without approval and oversight.

Software Version Control

All OnCue Technology, LLC software is version controlled in GitHub and synced between contributors (developers). Access to the central repository is restricted based on an employee's role. All code is written, tested, and saved in a local repository before being synced to the origin repository.

Source Control Protections

Production and release branches shall be protected. Changes shall reach a protected branch only through a pull request that has at least one approving review from a developer other than the author and passing automated checks. Direct pushes and force pushes to protected branches shall be disabled.

All developer accounts for GitHub, Google Workspace and Slack shall use multi-factor authentication. Repository access shall be removed within one business day of a developer's departure.

Technical Review of Applications after Operating Platform Changes

When operating platforms are changed, business critical applications shall be reviewed and tested to ensure that there is no adverse impact on organizational operations or security.

Restrictions on Changes to Software Packages

Modifications to third-party business application packages shall be discouraged, limited to necessary changes and all changes shall be strictly controlled.

Secure System Engineering Principles

Principles for engineering secure systems shall be established, documented, maintained and applied to any information system implementation efforts.

At a minimum, the following secure-by-design and privacy-by-design principles shall be applied:

Secure-by-design principles:

1. Minimize attack surface area
2. Establish secure defaults
3. The principle of Least privilege
4. The principle of defense in depth
5. Fail securely
6. Treat external services as untrusted
7. Separation of duties
8. Avoid security by obscurity
9. Keep security simple
10. Fix security issues correctly

Privacy-by-design principles:

1. Proactive not Reactive; Preventative not Remedial
2. Privacy as the Default Setting
3. Privacy Embedded into Design
4. Full Functionality – Positive-Sum, not Zero-Sum
5. End-to-End Security – Full Lifecycle Protection
6. Visibility and Transparency – Keep it Open
7. Respect for User Privacy – Keep it User-Centric

Software developers are expected to adhere to OnCue Technology, LLC's coding standards throughout the development cycle, including standards for quality, commenting, and security.

Secure Development Environment

OnCue Technology, LLC shall establish and appropriately protect environments for system development and integration efforts that cover the entire system development life cycle. The following environments shall be logically or physically segregated:

- Production
- Test / Staging
- Development

Third-Party Components and Dependencies

All third-party libraries and components used in OnCue Technology, LLC software shall be declared in the project's package manifests under version control. Automated dependency vulnerability alerts (GitHub Dependabot or equivalent) shall be enabled on all production repositories. Vulnerabilities in third-party components shall be remediated under the timelines in the Application Vulnerability Management section. A software bill of materials shall be produced from the manifests on request.

Secrets Management

Credentials, API keys, certificates, signing keys and other secrets shall not be stored in source code or committed to any repository. Secrets shall be stored in GitHub Actions encrypted secrets or in the company password manager, with access restricted by role. GitHub secret scanning and push protection shall be enabled on all repositories. Any secret found to be exposed shall be rotated immediately and the exposure treated as a security incident.

Release Integrity

All released installers, executables and update packages shall be code-signed with OnCue Technology, LLC's code signing certificate before distribution. Signing keys shall be stored in hardware or in the company password manager with access limited to release managers. Published builds shall be reproducible from a tagged commit in version control.

Outsourced Development

OnCue Technology, LLC shall supervise and monitor the activity of outsourced system development. Outsourced development shall adhere to all OnCue Technology, LLC standards and policies.

System Security Testing

Static analysis and dependency scanning shall run automatically on every pull request to a production branch, and a full security test pass shall be completed before each release. At least annually, OnCue Technology, LLC shall perform an independent security assessment of its products, performed by a qualified third party or by a qualified internal reviewer who did not

write the code under review. Findings shall be tracked to closure under the Application Vulnerability Management timelines.

No code shall be deployed to OnCue Technology, LLC production systems without documented, successful test results and evidence that identified security issues have been remediated or accepted under the Application Vulnerability Management timelines.

Application Vulnerability Management

Application code shall be scanned prior to deployment. Vulnerabilities shall be rated using the Common Vulnerability Scoring System (CVSS) and remediated within the following timelines from discovery:

Severity	CVSS score	Remediation timeline
Critical	9.0 – 10.0	Within 15 days of discovery
High	7.0 – 8.9	Within 30 days of discovery
Medium	4.0 – 6.9	Within 90 days of discovery
Low	0.1 – 3.9	Next scheduled release, or as resources permit

Where a fix is not available within the timeline, a compensating control or a documented risk acceptance approved by the Director of Risk Management shall be recorded before the deadline.

System Acceptance Testing

Acceptance testing programs and related criteria shall be established for new information systems, upgrades and new versions.

Prior to deploying code, a Release Checklist MUST be completed which includes a checklist of all Test Plans which show the completion of all associated tests and remediation of identified issues.

Protection of Test Data

Test data shall be selected carefully, protected and controlled. Confidential customer data shall be protected in accordance with all contracts and commitments. Customer data shall not be used for testing purposes without the explicit permission of the data owner and the Director of Risk Management.

Logging and Monitoring

Logging shall be enabled on the services OnCue Technology, LLC operates in production, including the GitHub organization audit log, Google Workspace and Slack administrative logs, and any licensing, update-distribution or web services. The company website is hosted on Wix, and its access and security logs are retained by the hosting provider. Logs under OnCue Technology, LLC control shall be retained for at least 90 days and reviewed when an incident or anomaly is suspected.

OnCue Technology, LLC does not operate a corporate network. Developer workstations shall run current operating system updates, full-disk encryption and endpoint protection.

Vulnerability Disclosure

OnCue Technology, LLC shall maintain a published security contact (security@oncuetech.com) for reporting vulnerabilities. Reports shall be acknowledged within five business days and handled under the Application Vulnerability Management section. Reporters acting in good faith shall not face legal action for responsible disclosure.

Use of AI Coding Assistants

AI-assisted development tools may be used. Every change produced with the help of such a tool shall be read and verified by the developer before it is committed; no AI-generated code shall be committed unreviewed. Code produced with such tools is subject to the same review, testing and security requirements as any other code and the author remains accountable for it. Confidential data, customer data and secrets shall not be entered into AI tools other than those approved by the Director of Risk Management, and approved tools shall be configured so that company inputs are not used for model training.

Acquisition of Third-Party Systems and Software

The acquisition of third-party systems and software shall be done in accordance with the requirements of the OnCue Technology, LLC Third-Party Management Policy.

Developer Training

Software developers shall be provided with secure development training appropriate to their role at least annually. Training content shall be determined by management but shall address the prevention of common vulnerabilities in Windows desktop applications. The following threats and vulnerabilities should be addressed as appropriate:

- safe handling of untrusted input, including parsing of documents, media, transcripts and case files received from outside the company
- path traversal and unsafe file handling when importing or extracting case data
- injection attacks against local databases and command execution
- unsafe deserialization of saved or imported data

- DLL search-order hijacking and unsafe loading of libraries
- avoidance of unnecessary local privilege and secure installer behavior
- integrity of software updates and installers
- secure storage of credentials and license data on the local machine
- correct use of cryptography
- prevention of the use of vulnerable third-party libraries
- secure use of any network-facing component, such as license activation or update services

Exceptions

Requests for an exception to this Policy must be submitted to the Director of Risk Management for approval.

Violations & Enforcement

Any known violations of this policy should be reported to the Director of Risk Management. Violations of this policy can result in immediate withdrawal or suspension of system and network privileges and/or disciplinary action in accordance with company procedures up to and including termination of employment.

Version	Date	Description	Author	Approved by
1.0	March 21, 2024	First Version	Harshit Kothari	Jim Groner
1.1	March 20, 2026	Annual refresh	Jim Groner	Jim Groner
1.2	November 23, 2026	Severity-based remediation timelines; new secure development controls; wording clarifications	Jim Groner, Director of Risk Management	David Cavin, Senior Programmer