Maine State Government
Dept. of Administrative & Financial Services
Office of Information Technology (OIT)

Mobile Application Policy

1.0 Purpose

1.1. This policy establishes governance, security requirements, and oversight for mobile applications that access, store, transmit, or process State Information Assets when used to conduct official business for the State of Maine. Security requirements are applied in a risk-based manner and aligned with the following, while accounting for platform-specific implementation differences between iOS and Android:

NIST SP 800-163 Rev. 1[1] – Vetting the Security of Mobile Applications
NIST SP 800-53 Rev. 5[2] – Security and Privacy Controls
OWASP Mobile Application Security Verification Standard (MASVS)[3]

2.0. Scope and Applicability

2.1. This policy applies to:

2.1.1. State-owned Devices: All mobile applications installed on State-owned or State-managed mobile devices, regardless of application purpose.

2.1.2. BYOD Devices: Mobile applications installed on personal BYOD (Bring Your Own Device) mobile devices only when such applications are used to conduct State business or directly access, store, transmit, or process State Information Assets. (see Mobile Device Policy[4])
Clarification: Enforcement actions on BYOD devices apply only to application access controls, such as revisions of entitlements or tokens. OIT does not modify personal device configuration.

2.1.3. Personnel: All personnel, including employees, contractors, and vendors, within The State of Maine Executive Branch.

2.2. This policy focuses strictly on the application layer. For policies governing mobile device configuration, mobile device management (MDM), or other device-level security controls, refer to the Mobile Device Policy[5].

[1] https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-163r1.pdf
[2] https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf
[3] https://mas.owasp.org/MASVS/
[4] https://www.maine.gov/oit/sites/maine.gov.oit/files/inline-files/MobileDevicePolicy.pdf
[5] https://www.maine.gov/oit/sites/maine.gov.oit/files/inline-files/MobileDevicePolicy.pdf

2.3 Applications on BYOD devices that are not used for State business and do not access State Information Assets are not subject to this policy. This policy does not apply to general-purpose mobile web browsers used to access State websites or web applications, which are governed by separate web and identity access policies.

2.4. Browser based applications such as Progressive Web Applications (PWAs) are governed by the Web Standards Policy[6], while installable apps are governed by this document. PWAs must comply with all secure coding, data handling, and privacy requirements defined in the Web Standards Policy. If a PWA becomes installable, it is governed by this Mobile Application Policy.

3.0 Roles and Responsibilities

3.1. Agency Management:

3.1.1 Shall be accountable for ensuring that all mobile applications used by agency personnel to conduct official business adhere to this policy.

3.1.2 Shall authorize requests for mobile applications based on valid business requirements and ensure funding is available for paid applications through approved channels. Once authorized internally, the Agency must submit a Request for Change (RFC) in the Enterprise Ticketing System to initiate the OIT technical review and implementation process.

3.2 Office of Information Technology (OIT):

3.2.1 Owns, executes, and enforces this Policy.

3.2.2 Shall maintain a vetting process for mobile applications in alignment with NIST SP 800-163 Rev. 1[7] to ensure they meet the security and privacy standards defined in Section 5.0.

3.2.3 Shall have the authority to restrict, revoke, or disable an application's or User's access to State Information Assets, require remediation, and mandate removal from State-managed devices when an application is found to violate the directives of this policy or pose a security risk.

3.2.4 On personal BYOD devices, enforcement actions are limited to application access controls, and revocation of User access privileges and do not include direct modification of personal devices.

3.3 OIT Security Architecture (SA)

3.3.1 Reviews application architecture, backend/API integrations, entitlements, data flows, and dependency risks.

[6] https://www.maine.gov/oit/sites/maine.gov.oit/files/inline-files/WebStandards.pdf

[7] https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-163r1.pdf

3.3.2 Ensures alignment with State architecture standards, NIST SP 800-53 Rev.5[8], and NIST SP 800-163 Rev.1[9]

3.3.3 Identifies required compensating controls or exceptions.

3.4 OIT Security Operations Center (SOC)

3.4.1 Performs and/or oversees mobile application security testing (SAST, DAST, API testing, TLS/pinning validation, crypto checks, logging/audit behavior)

3.4.2 Produces the Security Test Report (SRT) and validates remediation.

3.4.3 Supports re-vetting when application updates introduce material changes

3.5 OIT Risk Management (RM)

3.5.1 Defines mobile application risk assessment methodology.

3.5.2 Consolidates SOC and SA outputs and produces final risk rating and disposition (accept/mitigate/conditions).

3.5.3 Ensures consistency with enterprise risk tolerances.

3.6 State of Maine Personnel:

3.6.1. Shall only install and use mobile applications for State business that have been obtained through authorized Public App Stores.

3.6.2. Shall review application permission requests and comply with guidance provided by OIT regarding acceptable permission use for State business applications.

3.6.3. Shall not modify the mobile application or the device operating system (e.g., "jailbreaking" or "rooting") or otherwise attempt to bypass or defeat security controls.

4.0 Directives

4.1 Control Objectives: This policy establishes the following control objectives for mobile applications used by the State:

4.1.1. Applications must be acquired through OIT trusted distribution channels to reduce exposure to malicious or counterfeit software. When available, Microsoft Intune provides the most reliable and controlled method for managed application distribution.

4.1.2. Applications must maintain a consistent security control posture across supported mobile platforms, preventing security drift or patch gaps.

4.1.3. Applications must provide transparency regarding requested permissions and data access in support of informed governance decisions and user awareness.

4.2 Approved Acquisition Channels:

[8] https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf

[9] https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-163r1.pdf

4.2.1. Public App Store Requirement: To minimize the risk of malware and ensure baseline platform security vetting, all mobile applications used for State business must be obtained exclusively through official Public App Stores (Microsoft Intune, Google Play Store or Apple App Store), unless a waiver is granted by OIT.

4.2.2. Prohibition of Sideloading: The installation of applications from "unknown sources," third-party repositories, or direct "sideloading" (installing via APK or IPA files outside of the managed store) is strictly prohibited, unless a waiver is granted by OIT. Exception: Sideloaded builds are permitted only for internal QA/UAT testing, require an approved OIT waiver, and must be performed under SOC oversight, and cannot be used for production deployment.

4.2.3. CISA Guidance: This directive aligns with CISA Mobile App Adoption Best Practices[10], which identify official stores as the primary defense against counterfeit and malicious applications.

4.2.4. App Store Monitoring and Impersonation Defense: OIT shall periodically monitor public app stores for unauthorized applications that impersonate State of Maine agencies or services.

4.3 Change Management Requirement: The availability of an application on a Public App Store does not constitute authorization for deployment.

4.3.1. All new mobile application deployments[11] must follow the Normal Change process defined in the Change Management Policy and Procedures[12], requiring a formal Request for Change (RFC) and Security Impact Analysis (SIA) prior to installation.

4.3.2. Routine application updates may be managed as Standard Changes if they meet the criteria for low-risk, repeatable updates; however, they must still be logged in the Enterprise Ticketing System.

4.3.3 Detailed mobile application security testing procedures (Android and iOS) are maintained as appendices to the Application Deployment Certification (ADC) Policy [13] and are incorporated by reference into this Mobile Application Policy (see Appendices A & B). Digital accessibility requirements applicable to mobile applications are governed by the Digital Accessibility Policy[14]

[10] https://www.cisa.gov/sites/default/files/publications/Mobile%2520Application%2520Adoption%2520Best%2520Practices%2520Guide-508%2520compliant%2520FINAL%2520041316.pdf

[11] https://www.maine.gov/oit/sites/maine.gov.oit/files/inlinefiles/OIT_App_Deployment_Certification_Signoff_Template.pdf

[12] https://www.maine.gov/oit/sites/maine.gov.oit/files/inline-files/ChangeManagementPolicy.pdf

[13] https://www.maine.gov/oit/sites/maine.gov.oit/files/inline-files/ApplicationDeploymentCertification.pdf

[14] https://www.maine.gov/oit/sites/maine.gov.oit/files/inline-files/DigitalAccessibilityPolicy.pdf

4.4 Application Code and Security Control Requirements for Custom Development Mobile Applications:

4.4.1. Scope: The requirements in this section apply only to mobile applications developed specifically for the State of Maine under contract or other custom development arrangements. These requirements do not apply to commercially available, off-the-shelf, or third-party mobile applications obtained through public app stores.

4.4.2. Unified Codebase for State-Commissioned Applications: To promote efficient security auditing, consistent security controls, and maintainability, all mobile applications developed specifically for the State of Maine must be implemented using a single shared codebase capable of producing both iOS and Android applications.

4.4.3. Approved Cross-Platform Frameworks: Unless an exception is approved, State-commissioned mobile applications shall be developed using an OIT-approved cross-platform framework (e.g., React Native[15] or Flutter[16]) that enables delivery of functionally equivalent iOS and Android applications from a single shared codebase.

4.4.4. Validation and Review: OIT may validate compliance with this section through documentation review, targeted testing, vendor attestations, or code review.

4.4.5. Routine application updates may be managed as Standard Changes if they meet the criteria for low-risk, repeatable updates; however, they must still be logged in the Enterprise Ticketing System.

4.4.6. Exceptions: Exceptions to this requirement may be granted by OIT where documented technical constraints necessitate platform-specific native development. Any approved exception must include justification and acknowledgment of the associated security and lifecycle maintenance implications.

4.5 Permission Transparency and Management:

4.5.1. Permission Transparency: Mobile applications must provide transparency regarding all permissions requested from the underlying mobile operating system, including install-time permissions, runtime permissions, and elevated or special permissions.

4.5.1.1 Pre-Installation Disclosure: Application store listings and supporting documentation must disclose categories of data collected, used, or shared to enable informed procurement and approval decisions.

4.5.1.2. Runtime Visibility: Applications must provide clear, user-facing explanations when requesting privacy-sensitive or elevated

[15] https://reactnative.dev/

[16] https://flutter.dev/

permissions at runtime, describing the functional purpose of the permission.

4.5.2. Least Privilege: Applications must adhere to the principle of least privilege and request only permissions necessary to support documented business functionality.

4.5.2.1. Data Minimization: Collection and retention of personally identifiable or sensitive data must be limited to what is required for approved business purposes and handled in accordance with applicable State data protection policies.

4.5.2.2. Consent and Control: Where applicable, applications must provide mechanisms for user consent and data control consistent with platform capabilities and State privacy requirements.

4.5.3. Permission Classification and Governance:

4.5.3.1. Install-Time Permissions: Install-time permissions must be documented and reviewed by OIT as part of the application vetting and approval process.

4.5.3.2. Runtime Permissions: Runtime permissions must be requested only when required for application functionality and must be reviewed by OIT to ensure alignment with approved business use.

4.5.3.3. Elevated or Special Permissions: Applications requesting elevated or privileged permissions designated by the mobile operating system as higher risk (e.g., Android Special App Access permissions or equivalent iOS entitlements) shall require a valid OIT waiver.

4.5.4. OWASP MASVS Alignment: The OWASP Mobile Application Security Verification Standard (MASVS)[17] shall be used by OIT as a reference framework to assess permission handling, platform interaction, and privacy controls, with depth of assessment commensurate with application risk and data sensitivity.

4.5.4.1. Verification Levels: Applications handling sensitive data may require deeper verification against applicable MASVS control groups, while lower-risk or commercially acquired applications may be assessed through documentation, attestation, or targeted testing.

4.5.5. Data Protection Requirements: Applications must protect “sensitive data” at rest and in transit using industry-standard cryptographic mechanisms and must prevent exposure of sensitive State data through logs, caches, notifications, or other non-secure channels.

4.5.5.1. Certificate Pinning for Data in Transit: To ensure the authenticity of the server and prevent man-in-the-middle (MITM) attacks, all network communications transmitting sensitive data must be protected by Transport Layer Security (TLS). The application must implement certificate pinning to validate and accept only the known,

[17] https://mas.owasp.org/MASVS/

trusted certificate(s) or public keys belonging to its authorized backend servers.

4.6 Backend Security

4.6.1. Backend Dependency Scope: Security vetting shall extend to backend services, APIs, and web services that mobile applications rely upon to access or process State Information Assets.

4.6.2. Backend Security Expectations: Backend services supporting mobile applications must implement security controls appropriate to the sensitivity of the data and functionality provided, including authentication, authorization, input validation, logging, and secure communication.

4.6.3. Risk-Based Assessment: Backend components shall be assessed using a risk-based approach to identify material security weaknesses, including misconfigurations or known vulnerabilities, that could expose State Information Assets.

4.6.3. Backend Assessment Modality:

4.6.3.1. SA leads backend/API architecture evaluation.

4.6.3.2. SOC executes backend/API security testing (authN/authZ, input validation, rate limiting, error handling).

4.6.3.3. RM issues the risk rating and disposition.

4.6.3.4. Backend assessment artifacts are included in Section 4.8 (Mobile Application Security Testing & Certification).

4.7. Application Updates and Re-Vetting:

4.7.1. Update Classification: Mobile application updates shall be classified based on the nature and risk of the change, including but not limited to changes in functionality, permissions, data handling, authentication mechanisms, or backend integrations.

4.7.2. Re-Vetting Triggers: Applications must undergo re-vetting when updates introduce material changes that affect security posture, including, but not limited to:

4.7.2.1. New or expanded permissions.

4.7.2.2. Changes to data collection, storage, or transmission.

4.7.2.3. Modifications to authentication or authorization mechanisms.

4.7.2.4. Introduction of new backend services or third-party dependencies.

4.7.3. The Application Owner must notify OIT when updates introduce material changes (permissions, data collection/storage, authentication, authorization, backend integrations, or other security-impacting changes).

4.7.4. Material changes require submission of an RFC with SIA. SOC retests modified components; SA validates architectural changes; RM re-evaluates risk disposition.

4.7.5. Routine and Security Updates: Routine updates, including vendor-provided maintenance releases and security patches that do not introduce material security changes, may be deployed without full re-vetting, provided they are logged in the Enterprise Ticketing System.

4.7.6. Exception Handling: OIT may require additional review of any update when warranted by emerging threat intelligence, vulnerability disclosures, or observed security concerns.

4.8. Mobile Application Security Testing & Certification

4.8.1. Scope: All mobile applications must undergo security testing appropriate to the data sensitivity and risk level, as determined by RM.

4.8.2. Testing Domains:

4.8.2.1. Static Analysis (SAST)

4.8.2.2. Dynamic Analysis (DAST)

4.8.2.3. API/backend security testing

4.8.2.4. TLS/pinning and session protections

4.8.2.5. Cryptography and secure storage validation

4.8.2.6. Permissions/entitlements review

4.8.2.7. Privacy/data minimization assessment

4.8.2.8. Logging/audit behavior testing

4.8.2.9. Secure build/deployment configuration review

4.8.3. Standards Alignment; Testing references:

4.8.3.1. NIST SP 800-163 Rev. 1[18]

4.8.3.2. NIST SP 800-53 Rev. 5 (SA-11, CA-8, RA-5, SC-8/12/13/23, SI-10/11, AU-2/12, PT-2/3/6)[19]

4.8.4. Required Certification Artifacts

4.8.4.1. SOC Security Test Report (STR)

4.8.4.2. SA Architecture & Integration Review Summary

4.8.4.3. RM Risk Rating & Disposition

4.8.4.4. Permission & entitlement justification

4.8.4.5. Privacy evaluation

4.8.4.6. Final certification-to-deploy memo or conditional approval

4.9. Compliance:

4.9.1. Non-Compliance: Failure to comply with this policy may result in restriction or revocation of access to State Information Assets, mandatory remediation actions, or removal of affected applications from State-managed devices.

4.9.2. Continued Use: Continued use of a non-compliant mobile application after notification may result in additional enforcement actions in accordance with applicable State policies.

[18] https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-163r1.pdf

[19] https://csrc.nist.gov/projects/cprt/catalog#/cprt/framework/version/SP_800_53_511/home?element=AC

4.9.3. Enforcement Examples:

4.9.3.1. Suspension of backend/API access

4.9.3.2. Mandatory update/patch prior to re-enablement

4.9.3.3. Removal from Apple Business Manager or Managed Google Play

4.9.3.4. Revocation of API tokens or service keys

5.0 Definitions

5.1. Application Code Base: The source code, libraries, configurations, and build artifacts used to develop and deploy a mobile application. Mobile applications may employ platform-specific, cross-platform, or hybrid development approaches based on technical and security requirements.

5.2. Information Assets: The full spectrum of information technology resources owned, managed, or used by the State, including data, applications, systems, services, and supporting infrastructure.

5.3. Mobile Application (Mobile App): Software designed to run on smartphones, tablets, or other mobile devices and intended to access, store, transmit, or process State Information Assets, distinct from desktop or purely web-based applications.

5.4. Mobile Device: A computing or communication device running a mobile operating system, such as Android or iOS, as opposed to a workstation or server operating system.

5.5. Permissions: Privileges requested by a mobile application to access specific device capabilities or data, including but not limited to camera, microphone, contacts, location, storage, or system services.

5.6. Progressive Web Application (PWA): a type of web application that can be installed on a device as a standalone application. PWAs are installed using capabilities provided by the device's web browser, including offline caching.

5.7. Public App Store: *A*n official, platform-specific digital distribution platform for mobile applications operated by the platform vendor, including the Apple App Store for iOS and the Google Play Store for Android.

5.8. Request for Change (RFC): A formal request for a change to an information system or service, recorded and tracked through the Office of Information Technology's Enterprise Ticketing System.

5.9. Security Impact Analysis: A structured assessment to determine how a change to an information system affects its security control posture. For the purposes of this policy, mobile application vetting activities fulfill the Security Impact Analysis function.

5.10. Security Control Posture: The aggregate set of security controls, behaviors, and protections implemented by a mobile application, including authentication, authorization, encryption, data handling, session management, and update practices, as assessed across supported platforms.

6.0. Abbreviations

6.1. ADC: Application Deployment Certification
6.2. API: Application Programming Interface
6.3. APK: Android Package Kit
6.4. ATS: App Transport Security
6.5. BYOD: Bring Your Own Device
6.6. CA: Certificate Authority
6.7. CIO: Chief Information Officer
6.8. CISA: Cybersecurity and Infrastructure Security Agency
6.9. DAST: Dynamic Application Security Testing
6.10. HTTPS: Hypertext Transfer Protocol Secure
6.11. IPA: iOS App Store Package
6.12. iOS: iPhone Operating System
6.13. MAP: Mobile Application Policy
6.14. MASVS: Mobile Application Security Verification Standard
6.15. MDM: Mobile Device Management
6.16. MITM Man-in-the-Middle
6.17. MobSF: Mobile Security Framework
6.18. NIST: National Institute of Standards and Technology
6.19. OIT: Office of Information Technology
6.20. OWASP: Open Worldwide Application Security Project
6.21. PWA: Progressive Web Application
6.22. QA: Quality Assurance
6.23. RFC: Request for Change
6.24. SA: Security Architecture
6.25. SAST: Static Application Security Testing
6.26. SIA: Security Impact Analysis
6.27. SOC: Security Operations Center
6.28. SP: Special Publication
6.29. SRT / STR: Security Test Report
6.30. TLS: Transport Layer Security
6.31. UAT: User Acceptance Testing
6.32. XML: Extensible Markup Language

7.0. Document Information

7.1. Initial Issue Date: March 23, 2026

7.2. Latest Revision Date: March 23, 2026

7.3. Point of Contact: PolicyTeam.OIT@maine.gov.

7.4. Approved by: Chief Information Officer

7.5. Legal Citation: Title 5, Chapter 163: Office of Information Technology[20]

7.6. Distribution: Internet[21]

[20] http://legislature.maine.gov/statutes/5/title5ch163sec0.html

[21] https://www.maine.gov/oit/policies-standards

Appendix A – Android Mobile App Security Testing Procedures

Application Deployment Certification (ADC)

1. Purpose

Defines Android testing procedures referenced by MAP (§4.3.3, §4.7A). Testing depth is determined by RM and executed by SOC with SA review.

2. Pre-Validation

- Obtain QA/UAT APK pointing to non-production API endpoints.
- Set up test accounts and device/emulator.
- Configure Burp Suite proxy; install Burp CA certificate.
- Validate HTTPS interception.
- Confirm test builds are segregated from production.

3. Static Analysis (SAST)

- Scan APK using MobSF.
- Review AndroidManifest.xml for:
    - Exported components
    - Cleartext traffic
    - Debuggable flags
    - Dangerous permissions
- Identify hardcoded secrets, misuse of cryptography, insecure storage.
- Map to NIST: SA-11, RA-5, SI-02, SC-28, SC-12/SC-13.

4. Dynamic Analysis (DAST) / API Testing

- Validate TLS 1.2+ and certificate validation behavior.
- Document pinning behavior and bypass techniques (test only).
- Test authentication/authorization flows.
- Test input validation, error handling, and session management.
- Manipulate API requests for access control verification and rate limiting.
- NIST mapping: CA-8, SC-8, SC-23, SI-10/11.

5. Permissions & Privacy Review

- Inventory install-time/runtime permissions and “Special App Access.”
- Validate least privilege, data minimization, runtime transparency.
- Ensure no sensitive data in logs, caches, or notifications.
- NIST mapping: AC-3/AC-6, PT-2/3/6, AU-2/12, SC-28.

6. Backend/API Security

- SA reviews architecture, integrations, and service design.
- SOC tests API authN/authZ, throttling, error handling, and input validation.
- NIST mapping: AC-3, SC-7, SI-10/11.

7. Required Artifacts

(Referenced in MAP §4.7A.4)

- SOC: Security Test Report (STR).
- SA: Architecture & Integration Review Summary.
- RM: Risk Rating & Disposition.
- Permission/entitlement justification.
- Privacy evaluation.

8. Retesting & Change Management

- Material changes require RFC + SIA.
- SOC retests impacted components.
- SA validates updated architecture.
- RM re-evaluates risk/disposition.

Appendix B – iOS Mobile App Security Testing Procedures

Application Deployment Certification (ADC)

1. Purpose

Defines iOS testing procedures referenced by MAP (§4.3.3, §4.7A). Testing depth determined by RM and executed by SOC with SA review.

2. Pre-Validation

- Obtain QA/UAT IPA via TestFlight, MDM, or sideloading with correct provisioning profile.
- Confirm non-production endpoints.
- Configure Burp Suite; install/trust Burp CA profile and enable full trust.
- Validate HTTPS interception from Safari/WebView.

3. Static Analysis (SAST)

- Scan IPA using MobSF.
- Review Info.plist and entitlements for:
  - ATS (App Transport Security) settings
  - Sensitive APIs
  - Background modes
  - URL schemes
- Identify hardcoded secrets, insecure crypto, insecure storage.
- NIST mapping: SA-11, SI-02, SC-28, SC-12/SC-13.

4. Dynamic Analysis (DAST) / API Testing

- Validate TLS/pinning posture.
- Test authentication, authorization, input validation, error handling.
- Test session lifecycle: creation, rotation, revocation.
- NIST mapping: CA-8, SC-8, SC-23, SI-10/11.

5. Permissions & Privacy Review

- Inventory entitlements (camera, mic, location, contacts).
- Confirm least privilege and user transparency.
- Ensure logs/caches/notifications do not leak sensitive data.
- NIST mapping: AC-3/AC-6, PT-2/3/6, AU-2/12, SC-28.

6. Backend/API Security

- SA reviews backend design and integrations.
- SOC tests API authN/authZ, throttling, error responses, and input validation.
- Mapping: AC-3, SC-7, SI-10/11.

7. Required Artifacts

(Referenced in MAP §4.7A.4)

- SOC: Security Test Report (STR).
- SA: Architecture & Integration Review Summary.
- RM: Risk Rating & Disposition.
- Permission/entitlement justification.
- Privacy evaluation.

8. Retesting & Change Management

- Material changes require RFC + SIA.
- SOC retests affected areas.
- SA validates design.
- RM re-evaluates risk/disposition.