Information Security Standards

Connected Vehicle Mobile Applications Standard

Table of Contents

1.0 Summary

1.1 Introduction and Scope

This document defines cybersecurity standards for mobile applications in the Ford Connected Vehicle and Business-to-Consumer (B2C) environments.

This standard applies to all iOS and Android Mobile Applications that integrate with or interact with Ford vehicles and B2C mobility products and services.

1.1.1 Primary Audiences

Global Planning, Architecture, Engineering, Design and Development teams for Connected Vehicle, Mobility and Cyber Security.

1.2 Out of Scope

- Applications for mobile platforms other than iOS and Android, or for non-mobile clients.

  Note: *Contact the Connected Vehicle Cyber Security team for guidance on non-standard clients, Internet of Things (IoT) devices, or applications intended for other mobile platforms.*

- Endpoints or services that do not directly communicate with mobile applications, except where explicitly identified as an exception by the Business Owner.
- Anything in the vehicle other than direct interfaces with mobile applications.
- Anything relating to call centers, such as call center authentication.
- Requirements for security questions and answers used to recover passwords or related identity and access management issues.
- Issues related to mobile applications running on specific devices outside the normal installation process.

1.3 Assumptions

- Customer information is not retained on the device after specific logout operation by customer.
- Where appropriate, application controls will draw distinctions between Active (operations, functions) and Passive (screen with data elements, etc.) data usage.
- Procedural and security policies are prioritized and implemented.
  - Mobile Application policies are reviewed and updated where needed.
  - Contract Terms and Conditions provide guidelines and acceptable usage.
- Applications covered by this standard use IP-layer communications initiated over Wi-Fi or Cellular connections.
- For applications using Bluetooth Low Energy (BLE):
  - Specific features will only be available on mobile devices supporting BLE versions 4.2 or greater.
  - As of this revision, BLE payload is limited to 20 bytes.

1.4 Accountability and Data Ownership

1.4.1 Third Party Services

When Connected Vehicle or Mobility use cases in scope of this standard rely on third party development and hosting services, all third parties must act as responsible data stewards. Ford Business Owners must get documented permission from both Ford Office of General Counsel (OGC) and the third party to perform any security testing.

Ford Business Owners must understand:

- How third parties are securely transmitting, storing, and using data
- How third parties are controlling access
- How third parties ensure data integrity
- Where third parties are storing data and what the local regulations are
- Who third parties are sharing data with (internally and externally) and when they do
- How third parties classify data
- Data breach implications

Ford Business Owners must ensure that third parties:

- Securely isolate company data
- Apply the same protection standards to engaged subcontractors or other parties
- Prevent unauthorized access and modification of data
- Utilize approved encryption methods
- Implement secure deletion methods where required
- Understand Ford's data retention requirements
- Implement security monitoring
- Have incident response policies in place that are compliant with industry standards

1.4.2 Third Party Libraries

Approved third party libraries or plugins may be used to interface with mobile device OS services and application programming interfaces (APIs) where gaps exist.
Ongoing reviews and monitoring of third party library code and terms are required by the appropriate third party or Business Owner, including searching for vulnerabilities (see **Section 2.6** below) and code compliance.

- Ford must obtain permission to perform security and performance tests as needed. This includes offline libraries and production APIs.
- Ford OGC and Purchasing must review and approve all terms and contracts.
- Process must be defined for ethical disclosure of bugs and security vulnerabilities.

1.5 Progressive Risk Categories Summary

The development and implementation of appropriate controls depends on the relative level of data shared and used, and mapping certain controls to functions and interactions. The Progressive Risk Categories for the Connected Vehicle and B2C Mobile environments are listed below, with each of the four (4) categories building on lower-numbered levels.

Progressive Risk Categories align with the Confidentiality, Integrity, Availability (CIA) and Personally Identifiable Information (PII) ratings assigned to features and applications during the design phase. Assigning a Risk Category is a proactive step to help determine how to protect data and features in various contexts and situations.

- **Category 0 – Marketing & Public Data**
    - Applications intended to improve customer experience and acceptance; emphasis on ease of use with minimal intrusion.
    - Minimal credentials or PII needed; limited backend interaction with the application.
- **Category 1 – Basic Vehicle Function**
    - Applications that use remote vehicle access features to replicate functions (i.e., Remote Keyless Entry, Panic Alarm, etc.) along with certain vehicle personalization options.
    - Controls intended to protect the vehicle and vehicle components.
- **Category 2 – Digital Account Access**
    - Mobile applications that connect to personal account information, which may include third party financial services.
    - Requires access to PII and customer's financial information (including digital wallets and/or Ford Credit account information).
- **Category 3 – Command and Control**
    - Applications can initiate all vehicle access functions and enable movement (i.e., Drive Away, Passive Entry / Passive Start (PEPS)) along with limited automated functions.
    - Applies to highly and fully automated driving features managed remotely from an application, including scenarios where the driver is not actively controlling the vehicle.
    - Where Sensitive PII (SPII) is accessed or used, Category 3 must be assigned.

Note: *Category 3 will be the default for Vehicle-to-Vehicle (V2V) and Automated Vehicle (AV) functions and for features/applications not yet in the planning stage.*

2.0 Mobile Application Development Process controls

2.1 Process Controls Matrix

This table represents an at-a-glance overview of how the process controls (rows) in this chapter align with the Progressive Risk Categories (columns) summarized in **Section 1.5** above.

Control			0: Marketing & Public Data	1: Basic Vehicle Function	2: Digital Account Access	3: Command and Control
Secure Mobile Development	2.2.1	Application Signing	X	X	X	X
	2.2.2	Application Installation and Updates	X	X	X	X
	2.2.3	Error and Exception Messages	X	X	X	X
	2.2.4	Event Logging and Monitoring		X (partial)	X (partial)	X
	2.2.5	User Notification/Response Process		X	X	X
	2.2.6	Application Logout Procedure		X	X	X
	2.2.7	Permissions		X	X	X
	2.2.8	Input Validation		X	X	X
	2.2.9	Development and Debugging		X	X	X
	2.2.10	High Risk Device Warnings and Waivers			X	X
Tamper Protection	2.3.1	Application Obfuscation Technology		X (partial)	X (partial)	X
	2.3.2	Required Protections for Critical Functions		X (partial)	X (partial)	X
Authentication	2.4.1	Application Password	as appropriate	X	X	X
	2.4.2	Personal Identification Number (PIN)	as appropriate	X	X	X
	2.4.3	Session Management		X	X	X
	2.4.4	Device Authentication		X	X	X
	2.4.5	Multi-Factor Authentication			X	X
	2.4.6	Step-up Options			X	X
	2.4.7	Delegated Authorization / Authentication				X
Encryption	2.5.1	Transport Layer Protections (Secure Communications)		X	X	X
	2.5.2	Trusted Certificate Relationships		X	X	X
	2.5.3	Data Encryption		X	X	X
	2.5.4	Secure Data Storage (Data at Rest)			X	X
	2.5.5	Secure Data Usage (Data in Use)			X	X
Testing and Vulnerability	2.6.1	Verification and Testing	X (partial)	X	X	X
	2.6.2	Risk Assessments	X	X	X	X
	2.6.3	Vulnerability Identification		X	X	X
	2.6.4	Vulnerability Response Plan		X	X	X

2.2 Secure Mobile Development

Application Development and related teams must apply these controls as appropriate to the Risk Category. Where there are technical limitations, work with the Connected Vehicle Cyber Security team to identify and implement alternatives.

2.2.1 Application Signing

Each mobile application (iOS or Android) must use a unique digital signature in each development environment and must comply with NIST FIPS PUB 186-4. This includes non-PROD versions and build variants / flavors.

All signing material must be removed from shared builds and securely stored.
Evidence that the application code has verified the signatures must be maintained to ensure the application has not been recompiled and signed with a different Key.

2.2.2 Application Installation and Updates

Mobile applications covered by this standard must be available for distribution or download from a Ford-approved source, such as Google Play, Apple App Store or the Ford Store.

Once installed, software updates must only be delivered via a Ford-approved source (see ISP 14.2.2/14.2.2.F System Change Control Procedures).

2.2.3 Error and Exception Messages

Displayed error or exception messages must not compromise user data nor application integrity, and must not contain:

- URIs, URLs, or paths to any resources on the mobile device or in the service layer.
- Unhandled exceptions. If an unexpected error occurs, the user should see a generic error response, with no application details returned.
- Default system messages. Modify default system errors to prevent leakage of architectural details.
- Passwords, PINs, tokens or other credentials in any form (plain text, encrypted, etc.).
- Variable names or any other functional details that would support reverse engineering.
- Valid usernames or that confirm invalid usernames. If username is invalid, the error response should be a generic authentication error that cannot be used to catalogue usernames.
- Exceptions that can identify cryptographic code:
    - Examples include *InvalidKeyException*, *GeneralSecurityException*, *NoSuchAlgorithmException*, *InvalidAlgorithmParameterException*, *UnsupportedEncodingException* and *BadPaddingException*.
    - Use generic exception messages instead, such as *GeneralException* or *ExceptionCode0x01*.

2.2.4 Event Logging and Monitoring

Log collection and monitoring enables visibility into application security, validation of controls effectiveness, forensic analysis, alert generation, and accountability for actions. Refer to ISP 12.4.1 Event Logging, ISP 12.4.2 Protection of Log Information, and ISP 12.4.3 Administrator and Operator Logs as appropriate.
Business Owners must identify any additional or feature-specific actions to be logged and monitored.
Centralized monitoring tools must be utilized to capture all transactional logs.
Logging events required for security or forensic analysis must be retained until delivery to an approved Ford Logging Service.
Functional logs visible outside application sandbox, or shared with third parties, must not contain:

- URIs, URLs, or paths to any resources on the mobile device or in the service layer.
- Passwords, PINs, tokens or other credentials in any form (plain text, encrypted, etc.).
- Variable names or any other functional details that would support reverse engineering.

- Sensitive or personally identifiable information (PII) as defined by Ford OGC.

2.2.5 User Notification/Response Process

Whenever an application performs a critical function (i.e. password change, password reset, OTA update, etc.), an out-of-band notification must be sent describing the operation and providing contact information that can be used to resolve any issues.
The content of the notification message must not contain sensitive information such as PII/SPII.

2.2.6 Application Logout Procedure

Whenever possible, log out procedures must include secure deletion or protection of all user data stored on the device – such as user and vehicle info, log data, or any user-associated keys.

2.2.7 Permissions

Following the principle of least privilege, application end users must explicitly grant permission allowing the application to access sensitive data.
Application extensions used to delegate access must undergo additional code review and testing determined by the Business Owner to prevent unintended privilege escalation.
All methods must be restricted to only the intended calling objects (i.e., application, user, etc.).

2.2.8 Input Validation

Input validation must be implemented to ensure unauthorized methods cannot be injected.

- Implement code integrity checks (mobile and cloud) to ensure accurate business logic.
- Validate all input data to ensure proper formatting and that complies with application logic.
    - Whitelists are preferred and must comply with NIST SP 800-53.
    - Client-side only validation is ***not*** acceptable for any data sent to the cloud.
- Test all inputs to ensure they can only be used as intended. This includes mobile application input fields, service layer APIs and forms, IPC method calls and exposed intents / activities.

Individual functional specifications for applications must identify specific tests for input limits and acceptable input constraints. Where these limits are documented or implied, a set of tests must validate that the component appropriately handles or ignores inputs with:

- Longer or shorter lengths than specified
- Invalid or non-printable characters
- Values outside of the documented range or list of categories
- SQL injection code (i.e., bind variables, input sanitation, etc.)

2.2.9 Development and Debugging

All development, debugging and/or test code must be removed from the production environment before application publication. Refer to ISP 14.2.7.F Outsourced Development, ISP 14.2.8.F System Security Testing, ISP 14.2.9 System Acceptance Testing, and ISP 14.3.1 Protection of Test Data as appropriate.

- During application builds, disable developer debug access in production and test environments.
- Before publication, review stack trace reporting and system logging to ensure data leakage is not enabling other applications to collect sensitive information.

2.2.10 High Risk Device Warnings and Waivers

High risk devices ("rooted" for Android; "jailbroken" for iOS) undermine many security controls detailed in this standard. Application Terms and Conditions must include language to restrict, limit or prohibit the application's use on rooted or jailbroken devices.
The User interface must contain explicit warnings that the recommended security controls cannot be applied to rooted or jailbroken devices, and that Ford Motor Company does not recommend usage of its mobile applications on such unsecure devices. These warnings must use appropriate wording reviewed and approved by the Ford Office of General Counsel (OGC).

Note: *High-risk devices may be restricted from features that cannot be controlled, or required to utilize a more stringent "step up" or two-factor authentication model.*

Mobile applications must implement high risk device detection and logging. Once an application detects a rooted or jailbroken device, the following actions must be taken:

- **Category 3:** Detection must be logged to the Ford Security Incident and Event Management (SIEM) system.
- **Category 1 - 2:** Detection may be logged to the Ford SIEM system.
- Device may be flagged as "high risk" for purposes of forensics analysis.
- An in-application notification must notify the user that they are using an insecure device; acceptance of the notification must be logged.

2.3 Tamper Protection

2.3.1 Application Obfuscation and Tamper Detection Technology

Mobile applications must utilize application obfuscation technologies as an additional layer of protection, to prevent "white box" attacks and reverse engineering of the code base. This will include code obfuscation and cryptographic key protection for any sensitive information that could not be stored securely. Refer to ISP 14.1.3/14.1.3.F Protecting Application Services Transactions as appropriate.

2.3.2 Required Protections for Critical Functions

All sections of code that provide security-sensitive functionality, or have been identified as critical functionality during threat modeling, must be protected by additional obfuscation and tamper detection technologies. Please work with Connected Vehicle Cyber Security to determine appropriate protections for various categories of critical functions.

2.4 Authentication

This section applies to in scope interfaces where an entity must authenticate before continuing its operation, and applies equally to people, machines and any automated software system where the user or application is issued credentials to access in-scope infrastructure and services (internal or external to Ford).
Authentication provides a method of proving that the identification credentials used are valid, such as the combination of user ID and password.

2.4.1 Application Password

Mobile application passwords must comply with the ISP as a minimum, and must be actively maintained at regular intervals. Two-factor authentication must be used whenever privilege escalation is required.
Accounts must be disabled (locked out) after five (5) consecutive unsuccessful authentication attempts (lockout threshold) for a period of 15 minutes (lockout duration) with a notification to the User. If multiple lock outs occur without a successful password entry, the lock out interval should double in duration each time.

The maximum password length allowed is 128 characters.

There must be a secure process to allow User password resets.

2.4.2 Personal Identification Number (PIN)

Mobile applications may require an application PIN as an additional layer of authentication. PINs for all Connected Vehicle interfaces must comply with the Ford Information Security Policy (ISP) and must follow all guidelines listed in this document.

- PINs used by the application must be defined by the User.
- PINs must be masked, suppressed, or otherwise obscured whenever entered on screen.

PINs must be set immediately after the User authenticates for the first time or as soon as a new feature is added that requires an application PIN. If the option to remain logged into the application is set by default, the PIN must be set immediately upon successful authentication and before any additional functionality within the application can be accessed.
The user must re-enter the PIN after ten (10) minutes of inactivity. If the application is in the background or the screen is off, the 10 minute countdown should begin. Upon access within the countdown timeframe bringing the application to the foreground of the device, the countdown may be reset. After the countdown is expired, the PIN will be required.
After five (5) incorrect attempts to enter the PIN, the PIN must be locked for fifteen (15) minutes and the user must be sent an out-of-band notification. If multiple lock outs occur without a successful PIN entry, the lock out interval should double in duration each time.
PINs used in mobile applications must meet all of the following complexity requirements:

- The PIN must be at least four (4) digits. Six (6) or more digits are recommended; an eight-plus (8+) character alphanumeric passcode is preferred as it increases effort required for brute force guessing attempts.
- The PIN must be set immediately after the user chooses the option to remain logged into the application.
- The PIN must not have more than three similar numbers within a combination of four (4) digits. For example, combinations like 1111 or 5555, etc., are prohibited.

2.4.3 Session Management

Session IDs must be long enough to prevent brute force attacks such as enumerating a whole range of ID values to verify whether a session exists. Session IDs must be generated randomly to guard against attempts to guess the next session ID to be assigned. The minimum required length for a session ID is 16 bytes (128 bits).
All session IDs provided by standard request mechanisms must be validated using the appropriate session management mechanism.
Invalid session IDs must be ignored when detected.
By default, sessions must have a specific timeout value and must timeout automatically after a specified idle time. The specific length of time before a timeout occurs may change depending on use case; all use cases must be documented in the appropriate security specification (ACR, etc.).

2.4.4 Device Authentication

Some scenarios may require both the user and the device be authenticated. In these cases, a Universally Unique Identifier (UUID) may be used to identify a User / Mobile Device pair and must meet the following requirements:

- Uniquely represent a single user, a single device, and a single application instance.
- Never be duplicated.
- Not be shared nor be used for tracking purposes.

- Be securely generated and provisioned to the device during application installation or when opting into a new feature.

Device UUIDs must be invalidated if the application is uninstalled.

2.4.5 Multi-Factor Authentication

Any multi-factor authentication method used for additional authentication must differ from the initial authentication method.
Consult the Connected Vehicle Cyber Security team to identify an appropriate multi-factor authentication strategy.

2.4.6 Step-Up Options

Step-up authentication must be used for cases involving persistent login support for scenarios related to Vehicle tracking or location, Vehicle information, Customer profile data or any PII data.
Session tokens are often sufficient, but some use cases must also provide immediate step-up authentication for very sensitive transactions (this assumes initial login has already occurred).
Step up options may include PIN codes and/or biometrics such as fingerprint, voice, or facial recognition.
When step-up authentication is required, additional scrutiny must be applied to the input validation to ensure that transactions cannot be accessed without both initial authentication AND step-up authentication.
An approved step-up authentication method must be used whenever passwords are updated.

2.4.7 Delegated Authorization/Authentication

Passwords must not be used for API services. The best practice for authentication and authorization is to authenticate with the Identity Provider and immediately obtain an authorization token. This will prevent unnecessary transmission of usernames and passwords across untrusted networks. Industry standard methods are recommended for authorization tokens.
To prevent replay attacks, authorization tokens must have an expiration that correlates to the risk associated with the transaction.
To maintain integrity, tokens must contain a valid HMAC or digital signature.
For scenarios involving sensitive data or initiating any vehicle or financial transactions, authorization tokens must be digitally signed.
A separate token for the device may be required for certain tasks based on that token (which may be permanent or subject to expiration).

2.5 Encryption and Related Data Protections

Sensitive data must be protected by strong encryption whenever possible. Refer to ISP 10.1.1 Policy on the Use of Cryptographic Controls as appropriate.

2.5.1 Transport Layer Protection (Secure Communications)

To prevent eavesdropping and ensure confidentiality and integrity of data, Transport Layer Security (TLS) must be utilized for all **Category 1+** communications.
Non-standard communications devices or protocols not mentioned below must be approved by Connected Vehicle Cyber Security. Examples include Bluetooth, vehicle antennas and sensors, and vehicle modems.

- TLS version 1.2 or better must be used for all connections between Connected Vehicle components.
- Older versions of TLS and any version of Secure Sockets Layer (SSL) are not permitted.

- Service-to-service communications must use Mutual Transport Layer Security (mTLS).
- Sensitive data must never be sent over alternate channels (SMS, MMS, etc.).
- Mobile applications must use certificate pinning to prevent "man-in-the-middle" attacks.

2.5.2 Trusted Certificate Relationships

Mobile applications and cloud services may require additional identity verification beyond TLS. This ensures that the applications and services are communicating with a trusted endpoint, thereby avoiding connection spoofing. For **Category 1+** scenarios, Vehicle Cyber Security threat modeling will determine where additional verification will be required.

- Self-signed certificates must not be used.
- Data must only be delivered to the User if the Sender can provide a trusted signature.
- Mobile applications will only talk to trusted endpoints, as identified by an X.509 certificate issued by a Ford-owned or Ford-approved Certificate Authority.
- The Vehicle will not listen to any communications unless directed by an authenticated mobile application.
- It is the responsibility of the Ford Business Owner to ensure that all certificates are up to date and signed by an approved Certification Authority (CA) provider.
- An appropriate connection error message must display when an invalid or untrusted certificate is detected.
- Use of invalid or untrusted certificates must be logged, and the application must reject the connection without providing the user any option to proceed.

2.5.3 Data Encryption

To prevent data loss, sensitive data must be encrypted before transmission. Refer to ISP 9.2.4 Management of Secret Authentication Information of Users, ISP 9.4.3 Password Management System, and ISP 18.1.5 Regulation of Cryptographic Controls as appropriate.
Approved ciphers must be used for encryption. Encoding is not a valid substitute for encryption.
Cryptographic keys and other cryptographic material must be securely stored using native iOS KeyChain or Android KeyStore.
Access to cryptographic keys and encryption or decryption methods must be restricted so that other applications or users cannot view or execute.
Category 1+: Keys must not be accessed unless the User was previously authenticated.
Keys must be generated using an appropriate random number generator (RNG) standard, per NIST SP 800-90Ar1. RNGs must pass the randomness test as defined in NIST SP 800-22.
Category 1+: Keys must be generated using a hardware-backed secure element (Android TEE or iOS Secure Enclave).

2.5.4 Secure Data Storage (Data at Rest)

-

Applications must be secure by design with multiple layers of defense. When working with sensitive information, assume the application is running in an unsecured environment (such as a rooted / jailbroken device), so additional measures must be taken to ensure data is adequately secured.
Encrypt data at rest where technically feasible. Encrypt sensitive data and application data files with the appropriate techniques (see **Section 2.5.5**).

- If storage is required on the device, use approved encryption methods and key sizes.
- Mask how data is stored and accessed using substitution, shuffling, and other methods of obfuscating data.
- Never store data in plain text.

- Limit usage to RAM.

Sensitive data should only be stored on a mobile device if necessary to support functional requirements. If such storage is required, sensitive data must not be stored in a way that allows other applications or users to access the data.

- Sensitive data must **never** be stored in clear text – not even inside the keychain.
- Sensitive data must be encrypted whenever stored on disk.
- Sensitive data must never be stored on an SD card or any other external media, to prevent unauthorized access and data leakage. If an exception is granted allowing SD card use, data stored there must be securely encrypted.
- User credentials, tokens, keys, and all other confidential data must be stored inside the hardware-backed trust zone / secure element whenever possible.

Token lifetimes and other functions must be balanced against the proper Progressive Risk Category.

- **Category 1+:** Secure hardware backed storage must be used to store sensitive data and the cryptographic material used for encryption.
- **Category 2+:** Complete account numbers must not be stored and must never be displayed on-screen.

Connected Vehicle Cyber Security recommends the following additional protections:

- Use Checksums to validate the integrity of hashed data.
- Use Key derivation functions to keep multiple secret values secure.

2.5.5 Secure Data Usage (Data in Use)

To ensure trusted communications, utilize appropriate TLS protection and certificate pinning or stapling where possible (See **Section 2.5.1**).
Do not mix TLS and non-TLS content when displaying sensitive information. Vulnerabilities in non-TLS content could expose TLS secured content.
Exercise due care to ensure that sensitive data is properly cleared.

- Access must be restricted to minimize risk.
- Do not allow access to sensitive data while the application is not running.
- User must enter device authorization PIN in order to access sensitive data.
- Limit visibility into data caching, keyboards, copy/paste buffers, cookies and session tokens.
- Do not keep sensitive data in RAM longer than necessary.
- Set key variables to null after usage.
- Sensitive data must not be stored within Application snapshots/screenshots (i.e. clipboard).

Do not use "immutable objects" to hold credentials. Any object used to store a credential should be ephemeral and must be securely set to null immediately after use. Credentials should not be available in memory longer than required.
For additional scenario controls, see the *Mobile Device Credential Storage Specification*.

2.6 Testing and Vulnerability Management

2.6.1 Verification and Testing

Upgrades and changes to systems not controlled by Ford may affect how an implemented mobile solution functions. Systems develop at different rates, and periodic reviews of Ford mobile applications and software will determine if updates are necessary.

- **Threat Models:**
  - All mobile device applications must undergo a threat model exercise, and identified risks mitigated or accepted before deployment to Production.
  - Post launch threat modeling is required as appropriate.
- **Ethical Hacks / Penetration Testing:**
  - All mobile application services must undergo an ethical hack to identify and further understand any security risks or vulnerabilities associated with the service.
  - Any issues found must be documented and resolved before Production launch, and the results of these tests may lead to further controls requirements.
- **Vulnerability Scanning:**
  - All codebases affecting a mobile application must be regularly scanned for vulnerabilities during the development process and after release.
- **Fuzz Testing:**
  - All mobile device applications must be fuzz tested to ensure they properly handle invalid and unexpected data.

2.6.2 Risk Assessments

Use Risk Assessments to understand the value of information and assets that need protection and help prioritize remediation efforts. To ensure timely and effective remediation of known vulnerabilities, the following actions must be taken by the Business Owner as soon as vulnerabilities are identified in the hardware, software, organization, network, or components directly affecting the mobile application:

- All components must be identified and categorized, with accurate versions and complete disclosure of all dependencies. This includes mapping to the Common Vulnerability Scoring System (CVSS), Common Vulnerabilities and Exposures (CVE), and other internal metadata (vector, vehicle type, etc.).
- Impact and likelihood of a compromise must be quantified and evaluated. Factors to consider could include financial loss, data theft, physical theft, physical safety, etc.
- Remediation priority must be established, documented, and communicated to management.

2.6.3 Vulnerability Identification

Vulnerabilities may be identified through the testing methods described in **Section 2.6.1** or may be uncovered and disclosed publicly by independent research. In order to limit public disclosure of vulnerabilities, some controls have been defined below that must be applied to all mobile applications:

- New versions of software and code libraries (developed by Ford or other third parties) must be vetted for security posture and supportability.
- A formal reporting mechanism must be in place that enables ethical disclosure and prompt remediation.

2.6.4 Vulnerability Response Plan

Keeping applications and service level code patched ensures the latest security measures are installed. It is critical to identify an escalation procedure and incident response plan in advance.

All applications must have a vulnerability response plan documented and signed off as part of their control review process.

3.0 Appendices

3.1 References

- Ford Information Security Policy (ISP)
- Content from the 2016 "Symantec Connected Vehicle Cyber Security Capability Study"
- Federal Information Processing Standards (FIPS) Publication 186-4 "Digital Signature Standard (DSS)"
- National Institute of Standards and Technology (NIST) Special Publication 800-53 "Security and Privacy Controls for Federal Information Systems and Organizations"
- National Institute of Standards and Technology (NIST) Special Publication 800-90A Revision 1 "Recommendation for Random Number Generation Using Deterministic Random Bit Generators"
- National Institute of Standards and Technology (NIST) Special Publication 800-22 "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications"

3.2 Glossary

Terms and Acronyms used in this document.

- API – Application Programming Interface
- AV – Autonomous Vehicle
- B2C – Business to Consumer
- BLE – Bluetooth Low Energy
    - BLEM – Bluetooth Low Energy Module
    - BLEAM – Bluetooth Low Energy Antenna Module
    - BLE CM – Bluetooth Low Energy Control Module
- CA – Certificate Authority
- Certificate Pinning – Method to detect and block many kinds of man-in-the-middle attacks, by checking the sender's certificate against one stored on the device. Also called "public key pinning."
- CIA – Confidentiality, Integrity, Availability
- CVE – Common Vulnerabilities and Exposures
- CVSS – Common Vulnerability Scoring System
- DoS or DDoS – Denial of Service or Distributed Denial of Service. An attack type.
- HMAC – Hash Message Authentication Code
- HSM – Hardware Security Module
- IoT – Internet of Things
- IPC – Inter-process Communication
- KeyChain/KeyStore – Cryptographic key storage feature in iOS and Android respectively
- MMS – Multimedia Messaging Service
- mTLS – Mutual Transport Layer Security
- OGC – Office of General Counsel (Ford legal team)
- PEPS – Passive Entry / Passive Start
- PII – Personally Identifiable Information
- RNG – Random Number Generator
- Root/Rooting – Process enabling "super user" access on Android devices (related terms for iOS are "Jailbreak/Jailbroken")
- Sensitive – Data that is Confidentiality rating **1 or higher** and/or PII rating **2 or higher.** Sensitive data can change per application and may be re-evaluated on a case-by-case basis.
- SDN – Service Delivery Network (Ford-specific terminology)

    - NG-SDN – Next Generation SDN
- Secure Element – Secure hardware-backed storage on a mobile device; generally part of a SoC (see below); often compared to HSM. (Secure Enclave for iOS; TEE for Android)
- SIEM – Security Incident and Event Management
- SMS – Short Message Service
- SoC – System on Chip
- SSL – Secure Sockets Layer
- TEE – Trusted Execution Environment
- TLS – Transport Layer Security
- Threat – Any activity that that could negatively impact an application or its associated assets:
    - Threat Actor – individual capable of exploiting a vulnerability
    - Threat Vector – the method used by an actor to attack
    - Threat Target – anything of value to the actor
- UUID – Universally Unique Identifier
- V2V – Vehicle to Vehicle
- Vulnerability – Any weakness that can be exploited to produce unintended consequences (maliciously or by accident).

3.3 Mobile Application States

Term	iOS Equivalent States	Android Equivalent States
Foreground/Active	• Active • Inactive	• Resumed • Paused • Started
Background/Service	• Background	• Services • Created
Not Running	• Suspended • Not Running	• Stopped • Destroyed

3.4 Enhanced Authentication Methods

Method	Description
Step-up Authentication	Authorization procedure that requires an authorized User to complete an additional authentication challenge.
Two-Factor Authentication (2FA)	2FA is an authorization procedure requiring a User successfully complete two different authentication challenges.
Multi-Factor Authentication (MFA)	MFA generally includes 2FA, but may also refer to additional authentication challenges or methods.

Revision History

Change Date	Change Description
5-Dec-2017	Version 1.0: Initial publication
23-Feb-2018	Version 2.0: No content changes. Hyperlinks to ISP sections added, in support of Policy Portal release.
15-July-2018	Version 3.0: URLs updated; no content changes made.