# Operating Systems (OS) Security Requirements

Version 2.00

Version Date: March 28, 2022

**UNCONTROLLED COPY IF PRINTED**

**FORD CONFIDENTIAL**

# Revision History

| Date | Version | Created/Modified By | Notes |
|---|---|---|---|
| 12/06/2018 | 1.0 | Justin Mendenhall/jmenden2 | Initial Release Version |
| 03/28/2022 | 2.0 | Kelly Stephenson/steph46 | • Reformatted document to new template |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# Table of Contents

# Introduction

## 1.1 Purpose

This requirement document applies to all ECUs implementing a traditional embedded high level OS such as QNX, YOCTO, Linux derivatives, Android, and POSIX compliant OSs.

## 1.2 Terminology and Abbreviations

| | |
|---|---|
| Administrative Attack | An attack against the user or documented processes of how things should operate. Classic examples are phishing attacks or social engineering. |
| AES | Advanced Encryption Standard |
| CA | Certificate Authority |
| DID | UDS Data by Identifier |
| DoS | Denial of Service Attack |
| ECG | Enhanced Central Gateway |
| ECU | Electronic Control Unit |
| EOL | Manufacturing End of Line |
| Logical Attack | An attack involving an unintended hole in the functioning of the system such as a security hole that allows an attacker to circumvent intended security controls. |
| Physical Attack | An attack requiring physical access to the device and manipulating internal operations to cause it to perform unintended operations. |
| RNG | Random Number Generator |
| SOC | System On a Chip |
| TLS | Transport Layer Security (RFC 5246, RFC 6176) |
| UDS | Unified Diagnostic Service |

# Feature Design Requirements

## 3.1 Assumptions

## 3.2 Goals

Secure Operating System – OS security mechanisms must ensure that sensitive data is not disclosed, modified, or disrupted. All applications must be verified prior to executing to provide assurance that modifications have not been done.

## 3.3 Non-Goals

## 3.4 Architecture

### 3.4.1 Code Signing

#### 3.4.1.1 SW Signing

**###OS_SEC_00001### SW Signing**

Production ECUs shall authenticate all applications (including scripts) before proceeding with installation. Only code signed by

Ford shall be accepted for production. Only approved cryptographic libraries shall be used and each instance shall be

documented. Each processing domain within an ECU shall have a unique code-signing leaf certificate or Public/Private key pair.

AutoSAR or Canbedded Micro's shall follow the process defined on RQT-001403-020672

| **Requirement ID: ###OS_SEC_00001###** | |
|---|---|
| **Rationale** | |
| Code Signing verifies that the software or application that is being installed originated from a known and trusted source. Absent a vulnerability that bypasses code signing or permits injection, this feature prevents unauthorized software or applications from being installed on the system. | |
| **Acceptance Criteria** | **V&V Method** |
| • Verify that only authorized software, applications, and code are installed or executed on the system or module | DV Testing |
| **Notes** | |
| Verification of code, data, scripts and the originator through the use of a chain of trust approved by Ford. Chain of trust mechanism is a Ford approved public/private key pair and signature such as: X.509 Certificate PKCS #7, RSA-2048. Approved cryptographic libraries include but are not limited to: OpenSSL, boringSSL, TomCrypt. | |

| AutoSAR and Canbedded Micros reference: App_Signing _Requirements document<br><br>Other Modules reference: Embedded_Systems_Application _Signing | | | |
|---|---|---|---|
| **Version** | **Date** | **Author** | **Change** |
| 1.0 | 10/19/2021 | Jmenden2 | Initial version |

### 3.4.2 Secure Boot

#### 3.4.2.1 Secure Boot

## ###OS_SEC_00002### Secure Boot

The system shall have a method to verify that the system software (inlcuding bootloaders) has not been modified at system start-up. When modification is detected, the system shall ensure that unverified code is not executed. When multiple microcontrollers are used, it is expected that each microcontroller shall verify its SW stack independently.

All keys and certificates required for this operation shall be stored and protected in a tamper resistant IC. The Operating System shall not execute unverified code. The Operating System may execute verified code prior to the system completing the secure boot process.

| **Requirement ID: ###OS_SEC_00002###** | |
|---|---|
| **Rationale** | |
| Secure Boot ensures that the software or data on the hosted the system is authentic at boot-up. If unauthorized software managed to be installed onto the system after a software update, Secure Boot will catch this and prevent the system from booting. Secure Boot requires hardware support to work effectively. | |
| **Acceptance Criteria** | **V&V Method** |
| • Verify that the software installed on the system has not been tampered or modified since its installation | DV Testing |
| **Notes** | |
| The system includes one or more microcontrollers (SOC, VMCU etc). Verified code is software or binaries whose signatures have been compared and have been found to be unmodified since its installation. A tamper resistant IC is an HSM or similar device (Reference HSM section for further details).<br><br>Reference NIST 800-147 Appendix A parts 4 and 5, and NIST 800-155. | |

| **Version** | **Date** | **Author** | **Change** |
|---|---|---|---|
| 1.0 | 10/19/2021 | Jmenden2 | Initial version |

### 3.4.3 Code Integrity Check

#### 3.4.3.1 Code Integrity Check

**###OS_SEC_00003### Code Integrity Check**

The OS shall verify that an application, persistent data, and shared libraries are authorized prior to each application execution by comparing the application and data against a known and trusted signature or hash. If the values do not match or the value cannot be trusted, the application shall not execute. This event shall be logged and reported.

| **Requirement ID: ###OS_SEC_00003###** | |
|---|---|
| **Rationale** | |
| Applications that have been tampered or modified after system startup will not be caught by secure boot during the current ignition cycle or current system runtime. Checking the application's signature prior to execution will identify the modification and prevent the application from executing.<br><br>This is viewed as the third prong in application integrity verification. The other two prongs are verification at application installation (code signing) and system start (secure boot). | |
| **Acceptance Criteria** | **V&V Method** |
| • Prevent tampered or unauthorized applications from executing. | DV Testing |
| **Notes** | |
| | |

| Version | Date | Author | Change |
|---|---|---|---|
| 1.0 | 10/19/2021 | Jmenden2 | Initial version |

### 3.4.4 Key Management and Hardware Security Module

#### 3.4.4.1 Private Key & Certificate Protection (ex. HSM requirement)

**###OS_SEC_00004### Private Key & Certificate Protection (ex. HSM requirement)**

Private keys/certificates and symmetric keys shall not be directly exposed to the core operating system, applications, or external hardware. A tamper resistant integrated circuit (IC) may be used to protect the keys as necessary if the ECU uses symmetric keys, private keys, or executes secure critical code. Public keys/certificates shall be protected against any unauthorized modification.

| Requirement ID: ###OS_SEC_00004### | | | |
|---|---|---|---|
| **Rationale** | | | |
| Hardware Security Modules and its various subtypes provide hardware based cryptographic acceleration. HSMs essentially provide a mechanism for protecting security keys or cryptographic keys by not directly exposing them. These devices can also be leveraged for secure boot. | | | |
| **Acceptance Criteria** | | | **V&V Method** |
| • Secure cryptographic keys and provide HW acceleration of cryptographic functions. | | | DV Testing |
| **Notes** | | | |
| The class of tamper resistant IC required is dependent on how many cryptographic keys are required for a given system. Types of tamper resistant IC are: Hardware Security Module (HSM), Secure Hardware Extension (SHE), Trusted Platform Module (TPM), ARM TrustZone.<br><br>Secure critical code is privileged code that needs to execute with full trust and can perform privileged operations | | | |
| **Version** | **Date** | **Author** | **Change** |
| 1.0 | 10/19/2021 | Jmenden2 | Initial version |

### 3.4.5 Memory Protections

#### 3.4.5.1 Memory Protection

## ###OS_SEC_00005### Memory Protection

High level Operating Systems shall implement and enable memory protection mechanisms for the stack and heap of applications. Enabled protections shall be configured as defined below:

- Stack Canaries shall select "strong flag" or greater
- ASLR protections shall be ON
- PIE shall be ON
- RELRO shall be ON
- Nx/DEP shall be ON

| Requirement ID: ###OS_SEC_00005### |
|---|
| **Rationale** |
| Memory protection covers a wide area and includes multiple items. Memory protections help detect modification to an application's stack and/or heap, randomize an application's address layout, and prevent memory locations from being executable. |

| Acceptance Criteria | V&V Method |
|---|---|
| • Detect memory modifications and prevent abnormal program execution. | DV Testing |

| Notes |
|---|
| Examples of memory protections include but are not limited to: Stack Canaries, Address Space Layout Randomization (ASLR), Position Independent (PIE), Read-Only Relocation (RELRO), No Execute (nX), and Data Execute Prevention (DEP). |

| Version | Date | Author | Change |
|---|---|---|---|
| 1.0 | 10/19/2021 | Jmenden2 | Initial version |

## 3.4.6 Rootless Operation

### 3.4.6.1 Rootless Operation

## ###OS_SEC_00006### Rootless Operation

Applications and processes shall not execute as root or with permissions equivalent to root. If an application or process requires root or root equivalent access, sign-off shall be required by the security team.

| Requirement ID: ###OS_SEC_00006### |
|---|
| **Rationale** |
| If a process or application has root, root-equivalent, or root-like permissions or capabilities, it can modify files on the system, add/remove accounts, and install applications. Running applications and processes with the minimum amount of permissions required reduces the potential harm it can create if can create, either by design or malicious intent. |

| Acceptance Criteria | V&V Method |
|---|---|
| • Restrict access to system interfaces, run applications under Principle of Least Privilege, and minimize data leakage. | DV Testing |

| Notes |
|---|
| Applications include but are not limited to: OS applications and services, applications, and scripts. |

| Version | Date | Author | Change |
|---|---|---|---|
| 1.0 | 10/19/2021 | Jmenden2 | Initial version |

## 3.4.7 Mandatory Access Controls (MAC)

### 3.4.7.1 Mandatory Access Controls (MAC)

### ###OS_SEC_00007### Mandatory Access Controls (MAC)

The OS shall support and enable a Mandatory Access Control scheme. This MAC scheme shall limit access to resources, including but not limited to system interfaces, data, and message paths. The MAC scheme shall be set to Enforcement mode or equivalent. Applications shall only be granted the minimum amount of privileges required. Access to resources, including but not limited to system interfaces, data, and message paths, shall be limited and audited

| **Requirement ID: ###OS_SEC_00007###** | |
|---|---|
| **Rationale** | |
| The system determines and enforces the permissions and access to a given object. The primary utility for enforcement is UID and GID. MAC provides a mechanism to have a rootless system, secure channel, and application sandboxing. Rootless system employs Principle of Least Privilege by where each process or application runs with as few privileges as possible. Secure channel secures the inter-process communication by restricting access to those paths. Additional communication protections can be provided by using cryptographic functions or libraries. Application sandboxing, which also requires appropriate use of Discretionary Access Controls, limits Application A from accessing resources controlled by Application B. | |
| **Acceptance Criteria** | **V&V Method** |
| • Restrict access to system interfaces, run applications under Principle of Least Privilege, and minimize data leakage. | DV Testing |
| **Notes** | |
| Example MAC schemes include: SELinux, AppArmor, SMACK, TOMOYO | |

| Version | Date | Author | Change |
|---|---|---|---|
| 1.0 | 10/19/2021 | Jmenden2 | Initial version |

### 3.4.7.2 Mandatory Access Controls (MAC) Manifest

### ###OS_SEC_00008### Mandatory Access Controls (MAC) Manifest

A manifest shall be generated enumerating all applications and their MAC permissions. The manifest shall be reviewed by the security team.

| **Requirement ID: ###OS_SEC_00007###** |
|---|
| **Rationale** |

The system determines and enforces the permissions and access to a given object. The primary utility for enforcement is UID and GID. MAC provides a mechanism to have a rootless system, secure channel, and application sandboxing. Rootless system employs Principle of Least Privilege by where each process or application runs with as few privileges as possible. Secure channel secures the inter-process communication by restricting access to those paths. Additional communication protections can be provided by using cryptographic functions or libraries. Application sandboxing, which also requires appropriate use of Discretionary Access Controls, limits Application A from accessing resources controlled by Application B.

| Acceptance Criteria | V&V Method |
|---|---|
| • Restrict access to system interfaces, run applications under Principle of Least Privilege, and minimize data leakage. | DV Testing |

**Notes**

The Manifest may also be referred to as the MAC Security Policy.

| Version | Date | Author | Change |
|---|---|---|---|
| 1.0 | 10/19/2021 | Jmenden2 | Initial version |

## 3.4.8 Discretionary Access Controls (DAC)

### 3.4.8.1 Discretionary Access Controls

### ###OS_SEC_00009### Discretionary Access Controls (DAC)

Access to files and executables shall be managed and audited. Permissions in POSIX based systems shall be 744, 755, or hosted in a read-only configuration (e.g. read-only partition). Non-POSIX based systems must use equivalent permissions.

**Requirement ID: ###OS_SEC_00009###**

**Rationale**

Applications and users shall not have the ability to access, modify, or execute resources it does not own or is not a member of the owning group. Read access shall only be granted when needed. This prevents data leakage, prevents the system from acting in an unexpected way if configuration files, logs, or other data was modified, and prevents privilege escalation.

| Acceptance Criteria | V&V Method |
|---|---|
| • Access to system data, resources, and applications shall be limited, managed, and audited. | DV Testing |

**Notes**

| The owner of the file, directory, or process determines the access controls for that object. This protects files and process from unauthorized access and manipulation. | | | |
|---|---|---|---|
| **Version** | **Date** | **Author** | **Change** |
| 1.0 | 10/19/2021 | Jmenden2 | Initial version |

#### 3.4.8.2 Discretionary Access Controls: Default umask

### ###OS_SEC_00010### Discretionary Access Controls (DAC)

Default umask shall be 077.

| **Requirement ID: ###OS_SEC_00010###** | |
|---|---|
| **Rationale** | |
| Applications and users shall not have the ability to access, modify, or execute resources it does not own or is not a member of the owning group. Read access shall only be granted when needed. This prevents data leakage, prevents the system from acting in an unexpected way if configuration files, logs, or other data was modified, and prevents privilege escalation. | |
| **Acceptance Criteria** | **V&V Method** |
| • Default permissions are correctly set when a file is created. | DV Testing |
| **Notes** | |
| The owner of the file, directory, or process determines the access controls for that object. This protects files and process from unauthorized access and manipulation. | |

| **Version** | **Date** | **Author** | **Change** |
|---|---|---|---|
| 1.0 | 10/19/2021 | Jmenden2 | Initial version |

## 3.4.9 Secure Debug

#### 3.4.9.1 Secure Debug Port

### ###OS_SEC_00011### Secure Debug Port

All debug ports and services shall require a unique authentication credential per ECU for granting enablement or access or be disabled. When an ECU contains multiple micros, each micro shall have unique authentication credentials.

For Hardware Debug Ports, refer to ARL 020667

| Requirement ID: ###OS_SEC_00011### | |
|---|---|
| **Rationale** | |
| Debug ports should be disabled and removed as this reduces the attack surface. In cases where a debug port needs to be enabled or accessed, the utility, certificate, debug token, or password used to access or enable the debug port, the method shall be unique as this reduces the likelihood of other modules from being compromised through password reuse. | |
| **Acceptance Criteria** | **V&V Method** |
| • If a debug port or service needs to be accessed or enabled, a secure method is required. The credentials (e.g. password, certificate) shall be unique per each processor. | DV Testing |
| **Notes** | |
| Debug port access and enablement authentication credentials shall be unique per ECU. Debug port access and enablement events shall be logged and reported.<br><br>Unique passwords can be derived from approved random or psuedo-random functions. | |

| Version | Date | Author | Change |
|---|---|---|---|
| 1.0 | 10/19/2021 | Jmenden2 | Initial version |

## 3.4.10 Secure Diagnostics

### 3.4.10.1 Secure Diagnostics

## ###OS_SEC_00012### Secure Diagnostics

Diagnostic sessions shall authenticate the requestor and verify the requestor's permissions. Each requested secure diagnostic session event shall be logged in an auditable manner. Diagnostic session shall only be initiated by authenticated and authorized requestor for a given level of access.

| Requirement ID: ###OS_SEC_00012### | |
|---|---|
| **Rationale** | |
| Engineers or suppliers require different elevated access to troubleshoot the ECU or module. These credentials and elevated access could be used by malicious actors to modify intended functionality. | |
| **Acceptance Criteria** | **V&V Method** |

| | |
|---|---|
| • Only authenticated and authorized users shall be able to alter the system configuration. | DV Testing |

| Notes |
|---|
| Diagnostic session consists of elevated user privileges with the capability to access or modify the ECU or system. |

| Version | Date | Author | Change |
|---|---|---|---|
| 1.0 | 10/19/2021 | Jmenden2 | Initial version |

## 3.4.11 Calibration and Configuration

### 3.4.11.1 Calibration and Configuration Protection

### ###OS_SEC_00013### Calibration and Configuration Protection

Write access to calibration and configuration data in memory shall only be permitted to approved accounts, entities, or software update process. All modification events and all attempted modification attempts shall be logged and reported.

| Requirement ID: ###OS_SEC_00013### | |
|---|---|
| **Rationale** | |
| Using unauthorized calibration and configuration data may cause the system to operate in an unsafe manner, unsecure manner, or enabled unapproved features. | |
| **Acceptance Criteria** | **V&V Method** |
| • Ensure the use of authorized calibration and configuration data. | DV Testing |
| **Notes** | |
| Calibration and Configuration data are DIDs, configurable data (eg. config files). | |

| Version | Date | Author | Change |
|---|---|---|---|
| 1.0 | 10/19/2021 | Jmenden2 | Initial version |

## 3.4.12 Monitoring and Logging

### 3.4.12.1 Monitoring, Anomaly Detection, and Logging

### ###OS_SEC_00014### Monitoring, Anomaly Detection, and Logging

The system shall monitor and log security related events. Each event shall be written to a system log. The system shall implement an event specific counter, and each event shall increment the counter. The OS shall have a method to detect unauthorized modification of logs. Logs shall only be extracted by approved and authorized methods. A minimum 2MB rotating security log shall be retained.

| **Requirement ID: ###OS_SEC_00014###** | |
|---|---|
| **Rationale** | |
| Monitoring analyzes various aspects of the system, including but not limited to system performance, application execution, access events. Monitoring utilities exist to identify if an application or process exceeds its resource allocation or if it does not have access to its minimum allocation. Monitoring can also identify when an application becomes unresponsive or crashes.<br><br>Logging and reporting of this data is crucial to identifying problems or unexpected events. Logs should include a mechanism to identify if an entry is authentic or if it has been modified. This mechanism is generally referred to as Secure Logging. Options should exist to increase or decrease the verbosity of logs. Event counters are useful in the event that the logs roll over, you still know that an event occurred. | |
| **Acceptance Criteria** | **V&V Method** |
| • Provide an audit log of security related events | DV Testing |
| **Notes** | |
| Security related events include but are not limited to:<br><br>Software installation<br><br>Software installation attempts,<br><br>Attempts to initiate communication channels,<br><br>Attempts to initiate diagnostic sessions,<br><br>Attempts to access sensitive or critical data,<br><br>Privilege escalation attempts,<br><br>Application execution attempts,<br><br>System resets,<br><br>Direct memory access attempts,<br><br>Events occurring outside of defined behavior<br><br>Sensitive or critical data includes certificates, user credentials, passwords, keys, core configuration files.<br><br>Approved log extraction methods include: Signed utilities issued by a Ford approved entity (e.g. Hancock) | |

| Version | Date | Author | Change |
|---|---|---|---|
| 1.0 | 10/19/2021 | Jmenden2 | Initial version |

### 3.4.13 Kernel

#### 3.4.13.1 Kernel Version

## ###OS_SEC_00015### Kernel Version

The system shall use the latest long term support kernel. For Linux systems, this version shall not be less than version 4.0.

| Requirement ID: ###OS_SEC_00015### | |
|---|---|
| **Rationale** | |
| Using the latest Kernel reduces the potential number of vulnerabilities within the Kernel. | |
| **Acceptance Criteria** | **V&V Method** |
| • To reduce vulnerabilities and weaknesses within thie Kernel | DV Testing |
| **Notes** | |
| | |

| Version | Date | Author | Change |
|---|---|---|---|
| 1.0 | 10/19/2021 | Jmenden2 | Initial version |