Payment Card Industry (PCI)

Secure Software Standard

Report on Validation Template for P2PE Applications for use with PCI Secure Software Standard v2.0

July 2026

Document Changes

Date	Version	Description
July 2026	1.0	Initial release to support the assessment of P2PE Applications using the *PCI Software Security Framework – Secure Software Standard*, v2.0.

Contents

ROV Template Overview

The use of this ROV Template is mandatory for submissions of P2PE Applications, assessed using the *PCI Software Security Framework – Secure Software Standard v2.0*, to PCI SSC for consideration of Acceptance and Listing. **Be advised that certain requirements are not applicable, and their related functionality, are not permissible, for P2PE Applications. These unique amendments for P2PE Applications are accounted for within this template.** **Refer to the PCI P2PE Program Guide, including its Appendix J, for additional information.**
Only qualified P2PE Application Assessors are permitted to assess Software Products as a P2PE Application intended for use in a P2PE Solution. Refer to the PCI P2PE Qualification Requirements and the latest PCI P2PE Program Guide for detailed information.
A P2PE Application is any software or other files with access to cleartext account data that is intended to be loaded onto a PCI-approved PTS POI device and used as part of a P2PE Solution. P2PE Applications are required to use the underlying validated SRED functions of the PTS POI device (its hardware and firmware) to accept and encrypt cleartext account data. P2PE Applications are not permitted to perform encryption of cleartext account data.

This ROV Template IS ONLY permissible for use by P2PE Application Assessors to assess P2PE Applications using the *PCI Software Security Framework – Secure Software Standard v2.0.*

Terminology

Terminology used in this ROV Template can be found in the *PCI Secure Software Standard v2.0* and the respective *PCI Secure Software Program Guide,* in addition to the *PCI P2PE Program Guide* . General terminology can be found at: https://www.pcisecuritystandards.org/glossary.

Purpose

This document, the *PCI Secure Software Standard – Report on Validation Template for P2PE Applications* is provided by PCI SSC to support assessments of P2PE Applications in accordance with the *PCI Software Security Framework – Secure Software Standard v2.0* (*PCI Secure Software Standard, Secure Software Standard*) and Program. P2PE Applications will be referred to generically as a Software Product within this document where appropriate.

A Software Product assessment involves thorough testing and assessment activities from which the assessor generates detailed workpapers for each security requirement and its associated test requirements. These workpapers contain comprehensive records of the assessment activities, including observations, configurations, process information, interview notes, documentation excerpts, references, and other evidence collected during the assessment.

A completed Secure Software Report on Validation (ROV) acts as a comprehensive summary of the testing activities performed, the information that is collected during the assessment, and the findings and observations. The information contained in a completed ROV must provide sufficient

detail and coverage to support the assessor's findings that the Software Product has met all requirements of the *PCI Secure Software Standard* and Program, including additional requirements of the PCI P2PE Program as a P2PE Application intended for use in P2PE Solutions.

Using this Document

This *ROV Template* contains the required reporting template for P2PE Application Assessors to document a Software Product assessment.

Tables have been included in this *ROV Template* to assist in capturing required information. The tables may be modified to add rows as needed. However, the assessor must not remove or otherwise modify the text in this *ROV Template* in any way.

The ROV Template must be completed thoroughly and accurately. Failure to provide a complete and accurate ROV in accordance with the instructions herein, and in accordance with the *PCI Secure Software Standard* and Program will result in the rejection of an associated Software Product submission. If there are questions about this ROV Template, contact software@pcisecuritystandards.org.

Reporting Instructions

The Keywords below are documented as part of the Test Requirement Methods, which are defined in the *PCI Secure Software Standard.* The Test Requirement Methods describe the activity the Secure Software Assessor must apply for each Test Requirement. The Keywords and the Test Requirement Methods are copied from the Standard herein for ease of reference.

These Keywords are also used to document and define the Reporting Instructions that are required to be followed by the assessor for each Test Requirement. The "<*Assessor Response*>" in the Findings and Observations are required to be populated based on these Reporting Instructions.

The software assessor can choose to include *additional* test activity, as denoted below, *in addition to* the test activity specified for the security requirement, as is reasonable and needed to assist in verifying if a security requirement is satisfied.

Note: *There are no "in-line' reporting instructions in this ROV Template. The Reporting Instructions below must be used and followed for each Test Requirement in the Findings and Observations, using the Keywords as applicable. Failure to adhere to these instructions and to properly document the assessor responses will result in a rejection of the submission.*

Keyword	Test Requirement Methods	Reporting Instructions
Examine	The assessor critically evaluates evidence. Common examples include, but are not limited to, software design and architecture documents (electronic or physical), source code, configuration and metadata files, bug tracking data, log files, and security-testing results. The choice of evidence that may be used to meet an "examination" requirement is deliberately left open for the tester to determine.	Detail the documentation, evidence, or equivalent **examined** by the assessor that was used to **verify** the Security Requirement is satisfied as instructed and stated in the Test Requirement. If the evidence examined is documented in a table in the ROV Template, provide the unique reference(s) from the table that corresponds to the Test Requirement.

Keyword	Test Requirement Methods	Reporting Instructions
Interview	The assessor converses with individual personnel. The purpose of interviews includes determining how an activity is performed, whether an activity is performed as defined, and whether personnel have particular knowledge or understanding of applicable policies, processes, responsibilities, or concepts.	Detail the entities and/or personnel **interviewed** by the assessor that were used to **verify** the Security Requirement is satisfied as instructed and stated in the Test Requirement. If the interview is documented in a table in the ROV Template, provide the reference(s) from the tables that corresponds to the Test Requirement.
Observe	The assessor watches an action or views something. Examples of observation subjects include personnel performing tasks or processes, software or system components performing a function or responding to input, system configurations/settings, environmental conditions, and physical controls. Observation may include the performance of tests, so that the output of those tests may be observed, potentially under changing conditions as the input is manipulated by the tester or other systems.	Detail the activity, processes, or evidence **observed** by the assessor that were used to **verify** the Security Requirement is satisfied as instructed and stated in the Test Requirement. If the observations are detailed in a table in the ROV Template, provide the reference(s) from the table that corresponds to the Test Requirement.
Perform	Carry out the specified activity in the test requirement. This may implicitly include other test activities, such as examination and testing (e.g., including static and/or dynamic analysis of the software). **Note**: static and dynamic analyses implicitly include "test" activity.	Detail the activity **performed** that was used to **verify** the Security Requirement is satisfied as instructed and stated in the Test Requirement. If the activity performed is detailed in a table in the ROV Template, provide the reference(s) from the table that corresponds to the Test Requirement. ***Note**: static and dynamic analyses implicitly include the "Test" keyword, and therefore also requires satisfying the "Test" Reporting Instruction.*

Keyword	Test Requirement Methods	Reporting Instructions
Test	The assessor evaluates the software to analyze its characteristics and behavior in various scenarios to assist in determining whether the associated security requirement is satisfied. Testing is generally carried out using either static and/or dynamic analysis as specified in the test requirements. Testing generally includes *both* positive and negative test activities. Both static and dynamic analyses are considered types of test activities. – **Positive testing**: Generally used to confirm information, attributes, and expected behavior based on vendor documentation. – **Negative testing**: Generally used to identify undocumented information, undocumented attributes, and unexpected behavior. This is significant when it is imperative to attempt to bypass or circumvent certain software behavior, such as security controls or sensitive asset protection mechanisms.	Detail the **testing** that was used to **verify** the Security Requirement is satisfied as instructed and stated in the Test Requirement. If the activity performed is detailed in a table in the ROV Template, provide the reference(s) from the table that corresponds to the Test Requirement. ***Note**: static and dynamic analyses implicitly include the "Test" keyword, and therefore also requires satisfying the "Test" Reporting Instruction.*
Verify	Assert something is reasonably true based on the evidence available. Generally, the test requirements and the associated testing activities are performed to "verify" something in order to validate the security requirement(s) is satisfied.	The purpose of every Test Requirement is to facilitate the assessment activity that contributes to verifying the associated Security Requirement is satisfied. The Assessor Response must include a description of how the Test Requirement Methods conducted led to the affirmation that the Security Requirement is satisfied. I.e., the testing performed is used to validate the finding for the Security Requirement. If such an affirmation cannot be made, absent any claims of a Technical Constraint or eligible Not Applicable Finding then the Assessor must determine a Not In Place Finding for the Security Requirement.

Reporting Expectations

DO	DO NOT
- Follow all instructions herein. - Read and understand the intent of each security requirement and test requirement.	- Do not report items as "In Place" unless they have been verified as being "In Place." - Do not include forward-looking statements or project plans in the "In Place" column.

- Provide a response for every reporting instruction, unless explicitly instructed otherwise. - Provide sufficient detail, information, and rationale to demonstrate a finding of “In Place” or “N/A.” - Describe how a security requirement was verified as the reporting instruction directs, not just that it was verified. - Ensure that all parts of the test requirements and reporting instructions are addressed. - Ensure the response covers all applicable systems, processes, components, APIs, and functions including those provided by third parties. - Perform an internal quality assurance review of the ROV for clarity, accuracy, and quality. - Provide useful, meaningful diagrams, as directed. - Provide full dates where dates are required.	- Do not repeat or echo the test requirements in the response. - Do not copy responses from one test requirement to another. - Do not copy responses from previous assessments. - Do not include information irrelevant to the assessment.

Findings

There are at most three mutually exclusive Findings for Security Objectives and Security Requirements, detailed below. A Security Objective Finding is a result of the cumulative Findings of all its underlying Security Requirements.

Finding	Description
In Place	The expected testing has been performed and the test requirements satisfied, which contribute to the verification that the associated Security Requirement is satisfied. Detailed testing must be performed and reporting provided that demonstrates how the assessor verified the In Place Finding. ***Note:*** *A Security Objective Finding is a result of the cumulative Findings of all its underlying Security Requirements. To achieve an In Place Finding for a Security Objective, either:* *(a) All underlying Security Requirements for that Security Objective are verified to be "In Place", or* *(b) There is a combination of "In Place" and "N/A" Findings for the underlying Security Requirements.* *However, one or more "Not In Place" Findings must result in a Not In Place Finding for the overarching Security Objective.*
Not Applicable (N/A)	"Not Applicable", or "N/A", is only acceptable as a Finding where the Security Requirement, through testing and review, is determined to not apply to the Software Product. All "N/A" responses require reporting on testing performed (including interviews conducted and documentation reviewed, as applicable) and must explain how it was determined that the Security Requirement does not apply within the scope of the assessment. ***Note: "****Not Applicable" Findings cannot be used to intentionally reduce the scope of the Software Product assessment.*
Not In Place	Some or all elements have **not** been satisfied, are in the process of being implemented, or require further testing before it will be known whether they are "In Place". ***Note****: One or more Not In Place Findings must result in a Not In Place Finding for the overarching Security Objective.*

ROV Template for use by P2PE Application Assessors

This ROV Template must be filled out completely and accurately, as instructed herein. Failure to adhere to the template instructions will result in rejection of an associated submission.

1 Software Vendor & Product Information

1.1 Software Vendor Information					
Vendor Company Name:			Vendor Company Website URL:		
Vendor Contact Name:			Vendor Contact Phone:		
Vendor Contact Email:			Vendor Company Mailing Address:		
Does the Vendor have a 'Qualified' (Non-Expired) PCI Secure SLC Listing?	☐ **Yes**	If Yes, provide the Vendor's SSLC Qualified Listing Reference #: (*Not applicable if the SSLC Qualified Listing is Expired*)			
Indicate if the Vendor affirms their SSLC-qualified processes were used to develop this Software Product: *Select 'N/A' if the Vendor either (a) does not have an SSLC-Qualified Listing, or (b) the SSLC Listing is Expired.*	☐ **Yes**	☐ **No**	☐ **N/A**		

1.2 Software Product Identification			
Software Product (P2PE Application) Name:		**Software Product (P2PE Application) Version(s):**	
Is the Software Product currently (or was it previously) Listed on the PCI SSC List of Validated Secure Software?	☐ **No** (*The Software Product has never been Listed.*)		
	☐ **Yes** *(Provide the Reference #)* ➔		
If 'Yes', is this a Reassessment as per the *PCI Secure Software Program Guide*? ***Note**: The Listed status must be Validated (Not Expired) to qualify as a Reassessment. Otherwise, it will be regarded as a New Assessment. Refer to the PCI Secure Software Program Guide for details.* *P2PE Applications assessed and Listed using the PCI P2PE Standard and Program on the List of Validated P2PE Applications are not eligible for Reassessment as part of the PCI Secure Software Standard and Program.*	☐ **Yes**, this is a Reassessment, and the Listed Secure Software Product is Not Expired.		

1.3 Type of Submission

Type of Assessment

Check the type of submission this P2PE Application assessment is associated with:

☐ **Separately-listed P2PE Application** (***On the List of Validated Secure Software***)	Complete this ROV Template **for each** unique P2PE Application (*i.e., only one P2PE Application per ROV*) intended to be Accepted and Listed on the PCI SSC Validated List of Secure Software. If the P2PE Application is to be associated with a P2PE Solution, the P2PE Application must be Accepted and Listed **prior to** the submission of the P2PE Solution such that the Validated P2PE Application listing reference number can be denoted as necessary in the respective submission of the P2PE Solution. Upon Acceptance and Listing of the P2PE Application of the List of Validated Secure Software, its listing will denote it is acceptable for use as a P2PE Application, and its Reference# can be used to include the P2PE Application into eligible P2PE Products. **Submit this P2PE Application ROV as part of the PCI Secure Software Program.** ***Note:*** *It is **NOT** permissible to use this ROV Template to list the P2PE Application on the List of Validated P2PE Applications as per the PCI P2PE Program.*
Component (*read the Note →*)	***Note:*** *It is not permissible to submit a P2PE Application assessment with a P2PE Component Assessment.*
☐ **Merchant-Managed Solution**	Complete this ROV Template for each P2PE Application used by the Merchant-Managed Solution. ***Note:*** *Assessments associated with a Merchant-managed Solution are NOT submitted to PCI SSC.*
☐ **P2PE Solution Only** (***not separately-listed***)	***Note:*** *If the application is intended to be separately listed as a Validated P2PE Application, then complete the P2PE Application assessment as a 'Separately-listed P2PE Application'.* If the application is only associated with the P2PE Solution denoted below, complete this ROV and submit it with all the required P2PE ROVs as dictated by the scope of the P2PE Solution assessment. The application will not be separately listed and is **only** validated for use in the P2PE Solution denoted below. This is regarded as a 'Solution-specific P2PE Application' in the PCI P2PE Program Guide. **Submit this P2PE Application ROV and the associated P2PE Solution ROV(s) as part of the PCI P2PE Program.**
	Enter the P2PE Solution Name (from the P2PE Solution ROV) that this P2PE Application is being submitted with:

2 P2PE Application Assessor Company Information

2.1 P2PE Application Assessor Company and P2PE Application Assessor			
P2PE Application Assessor Company Name:		Lead Assessor Name:	
P2PE Application Assessor Company Contact Name:		Lead Assessor Phone:	
P2PE Application Assessor Company Contact Email:		Lead Assessor Email:	
P2PE Application Assessor Contact Phone:		Lead Assessor Secure Software Certificate Number:	
		Lead Assessor P2PE Application Assessor Certificate Number:	
Internal SSF Assessor Company QA Review Affirm that internal QA was performed on the completed ROV.	☐ **Yes** (Internal QA has been performed in accordance with the ROV instructions herein and with the PCI Secure Software Program Requirements)		
Primary QA Reviewer Name:		Primary QA Reviewer Credentials:	
Primary QA Reviewer Email:		Primary QA Reviewer Phone:	
Provide details for additional Secure Software Assessors involved with this assessment.			
Secure Software Assessor Name:		Secure Software Assessor Email:	

2.2 Independence		
The current *PCI Software Security Framework – Qualification Requirements for Assessors* (SSF Qualification Requirements) section "Independence" specifies requirements for SSF Assessor Companies and their Assessor-Employees regarding disclosure of such services and/or offerings that could reasonably be viewed to affect independence of an assessment.		
Affirm the Independence requirements as stated in the *SSF Qualification Requirements* have been read and fully understood.	➔	☐ Affirm
Document any consultation services provided to the Software Vendor by the SSF Assessor Company as it relates to this Software Product and its assessment.		
Disclose all other products or services provided by the SSF Assessor Company to the Software Vendor that were reviewed during this assessment or that could reasonably be viewed to affect assessment independence.		
Describe the efforts made to ensure that no conflict of interest resulted from the above-mentioned products and services provided by the SSF Assessor Company.		

2.3 Subcontracting		
The current *PCI Software Security Framework – Qualification Requirements for Assessors* (SSF Qualification Requirements) section "Subcontracting" specifies requirements for SSF Assessor Companies regarding subcontracting.		
Affirm the Subcontracting requirements as stated in the *SSF Qualification Requirements* have been read and fully understood.	➔	☐ Affirm
Affirm if subcontracting was utilized as it relates to this Software Product assessment	➔	☐ **No**, subcontracting was **not** utilized. ☐ **Yes**, subcontracting **was** utilized. (*Complete the remainder of the table*)
If Yes, subcontracting was utilized for this Software Product assessment, affirm that per the SSF Qualification Requirements, prior written consent was obtained from PCI SSC.	➔	☐ **Yes**, prior consent **was** obtained.
Identify the companies and personnel that were subcontracted.		
Describe the assessment-related activity performed by the subcontractors.		

3 Assessment Overview

3.1 Summary of Findings

+ Mark the Finding for each Security Objective and Module from the Findings and Observations herein. + An overall Finding for a Security Objective of "In Place" can consist of "In Place" Findings, or from a mix of "In Place" and "Not Applicable" Findings, based on the underlying Security Requirement Findings, however they must not include any "Not in Place" Findings. + One or more "Not In Place" Findings for Security Requirements must result in the Finding for the relevant Security Objectives being marked as "Not In Place".	In Place	Not Applicable	Not In Place
Security Objective 1: Software Architecture, Composition, and Versioning	☐		☐
Security Objective 2: Sensitive Asset Identification	☐		☐
Security Objective 3: Sensitive Asset Storage and Retention	☐		☐
Security Objective 4: Sensitive Modes of Operation	☐	☐	☐
Security Objective 5: Sensitive Asset Protection Mechanisms	☐		☐
Security Objective 6: Sensitive Asset Output	☐		☐
Security Objective 7: Random Numbers	☐	☐	☐
Security Objective 8: Key Management	☐		☐
Security Objective 9: Cryptography	☐	☐	☐
Security Objective 10: Threats and Vulnerabilities	☐		☐
Security Objective 11: Secure Deployment and Management	☐		☐
Module A: Account-Data Protection	☐		☐
Module B: POI Device Software	☐		☐
Module C: Publicly-accessible Software	☐	☐	☐
Module D: Software Development Kits	☐	☒	☐

3.2 Date and Duration of Assessment	
Date this ROV was completed: (*Write out the date fully. E.g., January 01, 1900*)	
Timeframe of Assessment: (*Start to Finish*)	

3.3 Remote Assessment Activity	
Indicate if any of the assessment activity was performed remotely, that per the Remote Assessment section in the *PCI Secure Software Program Guide*, warrants the completion and provision of the "Addendum for ROC/ROV: Remote Assessments", as provided in Appendix A of the *PCI SSC Remote Assessment Guidelines and Procedures:*	☐ **No**, the assessment activity performed does not qualify as a Remote Assessment as described in the *PCI Secure Software Program Guide*.
	☐ **Yes** – the "Addendum for ROC/ROV: Remote Assessments", as provided in Appendix A of the *PCI SSC Remote Assessment Guidelines and Procedures,* has been completed and is being provided as part of the associated submission.

4 Required Dependencies

4.1 General Required Dependency Affirmation

The only applicable, and mandatory, Required Dependency for P2PE Applications are eligible PTS POI devices.

4.2 PCI-Listed PTS POI Device Required Dependencies

P2PE Applications by definition have "PTS POI Device Required Dependencies". This table is <u>required</u> to be filled out accordingly.

***Note**: SRED is required for **<u>ALL</u>** PTS POI device hardware (HW) and firmware (FW) supported by the P2PE Application. It is NOT permissible for a P2PE Application to support non-SRED PTS POI devices (including any non-SRED HW/FW).*

- A New Assessment Software Product submission will not be considered for Acceptance if it claims a PTS POI device as a Required Dependency when the PTS Approval is Expired.

- If at any time prior to Acceptance of the Software Product submission, including during the PCI SSC AQM review process, the PTS POI device being claimed as a Required Dependency transitions to Expired, the Software Product submission will be rejected.

- There may be Secure Software Program allowances for the continued use of PTS POI devices for an eligible Software Product Reassessment. Refer to the PCI Secure Software Program Guide and the PCI Secure Software Technical FAQs to determine if there are any relevant allowances for Reassessments.

*- Entries such as "Any POI Device" are **<u>not</u>** permissible.*

PTS POI Device Testing

At least one unique combination of PTS POI device HW and FW (example #1) must be validated and functionally tested (as determined by the security requirements and associated test requirements) **<u>from each</u>** PTS approval that is being associated with the Software Product assessment.

Where the POI device FW is not monolithic (example #2), i.e., it is split into separate FW functionality (e.g., OS, SRED, OP), every FW required for the device to function as intended must be validated and functionally tested.

The Assessor must document, **<u>for each</u>** associated PTS approval, the supported POI device HW/FW(s) combinations that were validated and functionally tested, in addition to all eligible HW and software from the same PTS approval being supported by and intended to be listed for the Software Product. Note that all supported POI devices must be in accordance with the relevant security requirements. When populating the POI device version information, where the version tested is included in a wildcard version as shown in the PTS approval, document the wildcard version instead of the explicit version tested. E.g., if FW version 1.1 is tested, and the PTS approval denotes 1.x, then document 1.x.

Example:

PTS Approval #	Make / Mfr.	Model Name / Number	Hardware #(s)	Firmware #(s)	Applic #(s)	
9-12345	Anon	5000	1.x, 2.x, 3.x (Tested)	FW1.x FW2.x (Tested)	N/A	7-1.1, used for the POI device's RNG
0-54321	Ymous	100	HW1.x (Tested) HW2.x	OS: OS1.x OS2.x (Tested) SRED: S1.x (Tested) OP: OP1.x OP2.x(Tested) OP3.x	App1.1 (Tested)	Module B

Example 1

Example 2

This information must match the information as represented on the associated PTS approval exactly and be representative of the PTS POI devices included in the Software Product assessment.

Note: *'Applic' relates to a specific label on PTS approvals, which are also considered as firmware. Indicate N/A in the 'Applic' column as needed. However, if Applic software is used, it must be included.*

PTS Approval #	PTS Version #	Make / Mfr.	Model Name / Number	Hardware #(s)	Firmware #(s)	Applic #(s)

5 Detailed Software Product Information

5.1 Software Product Overview
Describe the overall software functionality and purpose, including a general overview of its sensitive assets:
Describe how the software is sold, distributed, or licensed to third parties (for example, licensed as software-as-a-service, stand-alone application, etc.). If the software is only used by the Software Vendor, indicate as such:
Describe how the software is designed (for example, as a standalone application, as a component or library, or as part of a suite of applications):
Describe a typical implementation of the software (for example, how it is configured in the execution environment or how it typically interacts with other systems or components).

5.2 Software Product Architectural Diagrams
The architectural diagram(s) should help facilitate the understanding of the Software Product and complement the information herein. The architectural diagrams should also complement the data/resource flow diagrams required herein. Ensure diagrams are clearly visible (not blurry) and comprehensible.

<Insert Architectural Diagrams Here>

5.3 Software Product Versioning				
- The associated *PCI Secure Software Program Guide* for v2.x contains details and criteria on versioning, which includes the use of wildcards and separators. It may be possible that related *Secure Software Technical FAQs* have also been published. - Affirm the published Secure Software Program information regarding allowable versioning is fully understood.			➔	☐ Affirm
Does the versioning include wildcards?	☐ **Yes**	If Yes, affirm the intended usage of wildcards is in accordance with the Program.	➔	☐ Affirm
Does the versioning include separators?	☐ **Yes**	If Yes, affirm the usage of separators is in accordance with the Program.	➔	☐ Affirm
Describe the format of the versioning scheme, such as the number of elements, number of digits used for each element, format of separators used between elements and character set used for element (consisting of alphabetic, numeric, and/or alphanumeric characters), use of wildcards, etc.				

6 Sensitive Asset Identification Information

This section aligns with the tables in the *PCI Secure Software Standard – Sensitive Asset Identification* document *(SAID)*, Appendix A.

While using the *SAID* Appendix A is optional, the information in this section **is required**, whether captured herein or provided within the SAID Appendix A.

If the *SAID* Appendix A is being used for this Software Product assessment, and is determined to be complete and accurate, the Secure Software Assessor does **not** need to duplicate the information in the respective tables herein, as applicable. Check the appropriate box for each table to indicate if the information will be provided in the *SAID* Appendix A as an additional document provided in the Portal as part of the relevant Software Product submission.

Note: *There may be slight differences in these tables herein from the tables in the SAID Appendix A respective to their presence within each unique document.*

6.1 Sensitive Asset Identification Affirmation

The *PCI Secure Software Standard – Sensitive Asset Identification* is essential to the assessment of a Software Product to the *PCI Secure Software Standard* and Program.		
Affirm the most recent version of *PCI Secure Software Standard – Sensitive Asset Identification* for v2.x has been read, fully understood, and used in the assessment activity as represented herein.	➔	☐ Affirm

6.2 Software Vendor Sensitive Asset Documentation

Check here if this information is being provided within the *SAID* Appendix A as part of the relevant Software Product submission.	☐ Yes

This table is provided to facilitate the required documentation for identifying and documenting sensitive assets, as well as make it easier to reference the documentation within other respective tables, as well as within the Findings and Observations.

Document ID: Self-assigned unique identifier used to allow for references to entries in this table.

Document Name: Name of the document.

Date / Version: The date and/or version number of the document.

Description: Describe the general content of the document and how it pertains to the Software Product assessment.

Document ID	Document Name	Date / Version	Description

6.3 Cryptographic Key Information

Check here if this information is being provided within the *SAID* Appendix A as part of the relevant Software Product submission.	☐ **Yes**

PCI Secure Software Requirements – This table is used to capture required information pertaining to requirements: 2-1.8[.x]
List all cryptographic key types used by the software that are associated with sensitive assets.

Key ID: Self-assigned unique identifier used for references to entries in this table. **Key Type:** E.g., DEK, KEK, PEK, MAC, Public, Private, etc. **Algorithm**: E.g., AES, RSA, DSA, SHA3, etc. **Key Mgmt**: E.g., DUKPT, MK/SK, Fixed, One-time use, etc.	**Key Length**: Full length (*include parity bits as applicable*) **Key Generation**: Generation method and origin **Key Destruction**: List destruction methods *for each* storage method ***Note:*** *As cryptographic keys are sensitive data, additional attributes for keys are accounted for in Table 6.4, which includes their storage location.*

Key ID	Key Type	Algorithm	Key Mgmt	Key Length (bits)	Fill out all the information below for each key type		
					Description & Purpose:		
					KEY	**Generation:**	
						Destruction:	
					Description & Purpose:		
					KEY	**Generation:**	
						Destruction:	
					Description & Purpose:		
					KEY	**Generation:**	
						Destruction:	
					Description & Purpose:		
					KEY	**Generation:**	
						Destruction:	

6.4 Sensitive Data Information

Check here if this information is being provided within the *SAID* Appendix A as part of the relevant Software Product submission. ☐ **Yes**

PCI Secure Software Requirements – This table is used to capture required information pertaining to requirements: 2-1, 2-1.1, 2-1.2

Sensitive Data ID: Self-assigned unique identifier used to allow for references to entries in this table.
Sensitive Data Type: The type of sensitive data. Refer to the "Sensitive Data Categories" in the *SAID* as a guide.
Sensitive Data Element: Specific sensitive data element in relation to the Sensitive Data Type.
Stored: Indicate with a 'Yes' or 'No' if the data element is stored.
Storage Locations: If stored, then document the location(s) where the data is stored persistently. Else if not stored, indicate 'N/A'.
Key ID: If the data element is a cryptographic key, populate the Key ID from Table 6.3. Else, indicate 'N/A'.
Doc ID: As applicable, references to entries in Table 6.2 herein.
Description / Use: Concisely describe the purpose/use of the sensitive data element within/by the software.

Sensitive Data ID	Sensitive Data Type	Sensitive Data Element	Stored	Storage Location(s)	Key ID	Doc ID	Description / Use

6.5 Sensitive Resource Information

Check here if this information is being provided within the *SAID* Appendix A as part of the relevant Software Product submission. ☐ **Yes**

PCI Secure Software Requirements – This table is used to capture required information pertaining to requirements: 2-2, 2-2.1, 2-2.2., 2-2.3

Sensitive Resource ID: Self-assigned identifier that can be used to allow for references to entries in this table.
Sensitive Resource Type: The type of resource. Refer to the "Sensitive Resource Categories" in the *SAID* as a guide.
Sensitive Resource Name: Unique identifier (names) of the individual resource in relation to the Sensitive Resource Type.
Stored: Indicate with a 'Yes' or 'No' if the sensitive resource is stored.
Storage Locations: If stored, then document the location(s) where the sensitive resource is stored persistently. Else if not stored, indicate 'N/A'.
Sensitive Data: If sensitive data elements associated with the sensitive resource, document the Sensitive Data IDs. Else indicate 'N/A'.
Doc ID: As applicable, references to entries in Table 6.2 herein.
Description / Use: Concisely describe the purpose/use of the sensitive resource within/by the software.

Sensitive Resource ID	Sensitive Resource Name	Sensitive Resource Type	Stored	Storage Location(s)	Sensitive Data IDs	Doc ID	Describe the purpose/use of the sensitive resource

6.6 Sensitive Functionality Information

Check here if this information is being provided within the *SAID* Appendix A as part of the relevant Software Product submission.	☐ Yes

PCI Secure Software Requirements – This table is used to capture required information pertaining to requirements: 2-3, 2-3.1, 2-3.2, 2-3.4, 2-3.5, 2-3.6

Sensitive Funct. ID: Self-assigned unique identifier used to allow for references to entries in this table.
Sensitive Funct. Name: Unique identifier (names) of the functionality in relation to the Sensitive Functionality Type, e.g., name of the function, process, etc.
Sensitive Funct. Cat.: The categorial type of functionality. Refer to the "Sensitive Functionality Categories" in the *SAID* as a guide.
Sensitive Data IDs: If sensitive data is associated with the functionality, enter the Sensitive Data IDs from Table 6.4. Else indicate 'N/A'.
Sensitive Resource IDs: If sensitive resources are associated with the functionality, enter the Sensitive Resource IDs from Table 6.5. Else indicate 'N/A'.
Externally Accessible: Indicate with a 'Yes' or 'No' if the functionality is externally accessible, e.g., as an API.
Sens. Mode of Op.: Indicate with a 'Yes' or 'No' if the functionality is a sensitive mode of operation.
Doc ID: As applicable, references to entries in Table 6.2.
Description / Use: Describe the purpose/use of the sensitive functionality within/by the software.

Sensitive Funct. ID	Sensitive Funct. Name	Sensitive Funct. Cat.	Sensitive Data IDs	Sensitive Resource IDs	Externally Accessible	Sens. Mode of Op.	Doc ID	Describe the purpose/use of the sensitive functionality

6.7 Sensitive Data and Resource Protection Information

Check here if this information is being provided within the *SAID* Appendix A as part of the relevant Software Product submission.	☐ Yes

PCI Secure Software Requirements – This table is used to capture required information pertaining to requirements: 2-1.5, 2-1.6, 2-2.6, 2-2.7

Sensitive Data ID: Reference to entries in the Sensitive Data Information table. Enter as many IDs as pertains to the same parameters for that row.
Sensitive Res. ID: Reference to entries in the Sensitive Resource Information table. Enter as many IDs as pertains to the same parameters for that row.
Protection Categories: Indicates the associated protection attributes, e.g., confidentiality and/or integrity protection. Refer to the *SAID* as a guide.
Protection Methods: Indicates the general method of protection, e.g., encrypted, truncated, hashed, etc.
Key ID: If the data is protected by a cryptographic key, populate the Key ID from Table 6.3.
Sen. Funct. ID: If the security mechanism pertains to Sensitive Functionality in Table 6.6, document the Sensitive Functionality ID.
Doc ID#: As applicable, references to entries in Table 6.2.
Description: Concisely describe the security mechanism used to protect the sensitive data/resource.

Sensitive Data ID	Sensitive Res. ID	Protection Categories	Protection Methods	Key ID(s)	Sens. Funct. ID	Doc ID	Description of security mechanism(s) implemented to protect the Sensitive Data / Resource

6.8 Sensitive Data / Resource Retention & Deletion Information

Check here if this information is being provided within the *SAID* Appendix A as part of the relevant Software Product submission.	☐ **Yes**

PCI Secure Software Requirements – This table is used to capture required information pertaining to requirements: 2-1.3[.x], 2-2.4[.x]

Sensitive Data ID: Reference to entries in the Sensitive Data Information table. Enter as many IDs as pertains to the same parameters for that row.

Sensitive Resource ID: Reference to entries in the Sensitive Resource Information table. Enter as many IDs as pertains to the same parameters for that row.

Configurable Retention: Indicate 'Yes' or 'No' if the retention period is configurable after the software is deployed (in use).

Retention Period: Document the retention period. If configurable, document if there is a defined range of allowable periods/settings.

Sen. Funct. ID: If the retention and/or deletion mechanism pertains to Sensitive Functionality in Table 6.6, document the Sensitive Functionality ID.

Doc ID: As applicable, references to entries in Table 6.2.

Deletion Method: Describe the method used to securely delete the sensitive asset or otherwise render it unrecoverable when it is no longer required.

Sensitive Data ID	Sensitive Resource ID	Configurable Retention	Retention Period(s)	Sens. Funct. ID	Doc ID	Describe the Deletion Methods

6.9 Sensitive Asset Flow Diagrams

Check here if this information is being provided within the *SAID* Appendix A as part of the relevant Software Product submission.	☐ **Yes**

PCI Secure Software Requirements – This is required information pertaining to requirements: 2-1.7, 2-2.8

Provide flow diagrams that show the details of all sensitive asset flows. The ID#s from the previous tables in this section should be used for easier references in the diagram(s). For each sensitive asset flow, identify the following:

- The sensitive data involved.
- The sensitive resources involved.
- The sensitive functionality involved, including all sensitive modes of operation.
- All components involved in the storage, processing, and/or transmission of the sensitive assets above.

Specify all types of sensitive asset flows, including any output to hardcopy, paper, or other external media. Sensitive asset flows must also indicate locations where sensitive assets cross trust boundaries and where it is passed to other applications or services that were not included in the assessment.

Ensure diagrams are clearly visible (not blurry) and comprehensible.

<Insert Flow Diagrams Here>

7 Findings and Observations - Overview

7.1 Reporting Instructions Affirmation

This ROV Template requires the use of defined Reporting Instructions based on Keywords used in the Test Requirements. Refer to the Reporting Instructions herein. It is imperative that the Reporting Instructions are followed and the expectations are satisfied in the Findings and Observations. Failure to do so will result in the rejection of the associated submission.

Affirm that the use and expectations of the defined Reporting Instructions herein have been read, fully understood, and satisfied in the Findings and Observations.	➔	☐ Affirm

7.2 Sampling

The *PCI Secure Software Standard* contains information on sampling ("Use of Sampling"), and sampling of the Software Product is not permitted.

Affirm that sampling of the Software Product cannot be used in the Software Product assessment	➔	☐ Affirm

7.3 Technical FAQs

The *PCI Secure Software Standard* and the *PCI Secure Software Program Guide* contain information regarding Technical FAQs, which relates to a separate document that is published and updated on an as needed basis. As Technical FAQs are mandatory for consideration as part of a Software Product assessment, it is imperative to review and understand them as part of the assessment activity.

Affirm that the most recent *PCI Secure Software Technical FAQs* for v2.x were reviewed thoroughly as part of this assessment.	➔	☐ Affirm

7.4 Documentation and Evidence

***Note**: Do **not** duplicate the documentation referenced in Table 6.2 specific to sensitive asset identification, unless it is also used for additional context as part of the assessment activity. However, avoid duplicating the self-assigned identifiers.*

Identify and list all the documents, materials, and other evidence obtained and reviewed during the assessment.

ID #: Self-assigned unique identifier used to allow for references to entries in this table.

Name: The title given to the documentation or evidence obtained.

Date: The date the document/evidence was created, or last updated, whichever is more recent.

Source: The entity who created and/or generated the documentation or evidence. Documentation and evidence are typically the responsibility of the Software Vendor or a third-party to create and maintain. However, it may be necessary for the Assessor to generate their own evidence as part of the assessment activity. Acceptable values for this field are "Vendor," "Assessor," or "Third-Party (*requires additionally identifying the third-party*)

Description/Purpose: A brief description of the contents and/or purpose of the documentation or evidence obtained.

ID #	Name	Date	Source	Description / Purpose

7.5 Interviews

Identify and list the individuals interviewed during the assessment.

If interviews are conducted, ensure the applicable Test Requirements include a reference to this table as applicable.

ID #: Self-assigned unique identifier used to allow for references to entries in this table.

Interviewee(s): The name of the individual(s) who participated in the interviews.

Job Title: The job title or job function of the interviewee(s).

Organization: The organization(s) represented by the interviewee(s).

Topics Covered: A high-level summary of the topics covered during each interview.

Interview Notes Ref#: The notes and/or audio files generated by the Assessor of the interview. Values in this column should include references to the appropriate documentation or evidence documented in Table 7.4.

ID #	Interviewee(s)	Job Title(s)	Organization	Topics Covered	Interview Notes Ref#

7.6 Testing

Identify and describe the testing performed during the assessment.

Tests may be grouped together if performed as part of a common test goal or objective. However, the details provided in each row should be sufficient to differentiate tests where variations are necessary to validate different Security Requirements.

ID #: Self-assigned unique identifier used to allow for references to entries in this table.

Description: A brief description of the types of testing performed (e.g., static analysis, dynamic analysis), the tools or methods used, etc.

Scope: The specific features, functions, or components assessed.

Objective / Purpose: The primary purpose of the test(s). May also include references to specific Security and/or Test Requirements.

ID #	Description	Scope	Objective / Purpose

7.7 Not Applicable Findings

Identify all Security Objectives, Security Requirements, and Modules marked as Not Applicable (N/A) in the Findings and Observations herein.

- Only Security Objectives and Security Requirements having the potential to be marked as N/A will have the N/A option available as a Finding.

- A "Not Applicable", or "N/A" finding is only acceptable where an appropriate degree of analysis and testing is used to determine the Finding.

- Mark entries below in the order they appear in the Findings and Observations herein.

- If an entire applicable Security Objective or Module is marked as N/A, do not list all underlying Security Requirements as N/A.

- Provide the analysis for the use of N/A for an entire module at the beginning of that Module's section where indicated herein.

Affirm that the criteria and use regarding Not Applicable (N/A) as a Finding is fully understood	➔	☐ Affirm

For each row entry below:

1. Document the Security Objectives, Modules, and/or Security Requirements in the Findings and Observations with a Not Applicable Finding.
2. Affirm that the Findings and Observations match this list for all Not Applicable Findings, and that each respective Security Objective, Module, and/or Security Requirement in the Findings and Observations contains a description of the analysis/testing performed that was used to determine the Finding.

***Note**: Discrepancies between this table and the Findings and Observations will result in the rejection of the associated submission.*

Security Objective / Security Requirement / Module	Affirmation (as per description above)
	☐ Affirm

7.8 Technical Constraints

The *PCI Secure Software Standard* contains information on Technical Constraints. It is imperative to read and understand the appropriate usage of Technical Constraints in a Software Product assessment.

- *A Technical Constraint is not a Compensating Control, and the use of Compensating Controls is not applicable to the PCI Secure Software Standard and Program.*
- *The use of a Technical Constraint must be documented herein as instructed.*
- *Mark entries below in the order they appear in the Findings and Observations herein.*

Affirm the criteria and use regarding Technical Constraints is fully understood.	➔	☐ Affirm
Affirm if Technical Constraints are being claimed for this assessment.	➔	☐ **No**, Technical Constraints are **not** being claimed. ☐ **Yes**, Technical Constraints **are** being claimed.

If Technical Constraints are being claimed, then for each row entry below:

1. Document the Security Requirements or Test Requirements in the Findings and Observations where a Technical Constraint is being claimed.
2. Affirm that the Findings and Observations match this list for all claims of a Technical Constraint, and that each respective Security Requirement and/or Test Requirement in the Findings and Observations contains a description, at a minimum, of the following.
   a. That a Technical Constraint is being claimed.
   b. The extent to which the security requirement or test requirement **can** be satisfied, if at all.
   c. The technical limitation of the implementation and why the limitation cannot be resolved or otherwise remediated.
   d. The residual risk that exists due to the technical limitation.
   e. If any alternative means have been leveraged or implemented to reduce the residual risk incurred by the technical limitation, whether within the software product itself and/or external to it.

Note: *Discrepancies between this table and the Findings and Observations will result in the rejection of the associated submission.*
IMPORTANT: *Technical Constraints cannot be claimed in order to bypass the use of the underlying SRED functions of the PTS POI device that are required to accept and encrypt cleartext account data.*

Security Requirement / Test Requirement	Affirmation (as per description above)
	☐ Affirm

8 Findings and Observations – Details

This section contains all Security Objectives, Security Requirements, and Test Requirements from the PCI Secure Software Standard, v2.0. In addition, it provides the requisite places for the Assessor to document their Findings and Observations of the assessment.

The use of this ROV Template is mandatory for submissions of Validated Secure Software Products to PCI SSC for consideration of Acceptance and Listing.

The ROV Template must be completed as instructed and accurately reflect the exact submission it is being used for.

Existing text must not be modified in any way. Only the determined Findings and documented Observations are to be populated by the Assessor.

Core – All Software

This section provides a minimum baseline set of security objectives and associated security requirements **required** for all software being assessed to this standard.

The applicability of the modules—which contain additional requirements—to the software assessment does not supersede any requirements in this section.

Security Objective 1: Software Architecture, Composition and Versioning

The architecture, composition, and versioning schema of the software are documented.

Notes: This section encapsulates the entirety of the software intended to be assessed to this standard and subsequently represented by its potential validation.

Select the overall Finding for this Security Objective ➔	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective 1: Software Architecture, Composition, and Versioning			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
1-1 The overall architecture of the software is documented	**In Place** ☐	[redacted]	**Not In Place** ☐
1-1.a Examine vendor documentation to **verify** the architecture of the software is documented. Leverage this information as relevant for the software assessment.	<Assessor Response>		
1-1.1 The security-relevant architectural aspects of the software design and implementation to protect sensitive assets are documented. **Implementation Notes** This requirement is in context of the overall security architecture, as the software itself is a sensitive asset. It should be demonstrable that the software is proactively and purposefully designed with security and the protection of sensitive assets as a significant objective. This requirement is not intended to duplicate the specific protection requirements for sensitive data and sensitive resources from Security Objectives 2 and 3.	**In Place** ☐	[redacted]	**Not In Place** ☐
1-1.1.a Examine vendor documentation to **verify** the security-relevant architectural aspects of the software design and implementation to protect sensitive assets are documented. Leverage this information as relevant for the software assessment, in particular for requirements in this Security Objective 5.	<Assessor Response>		

Security Objective 1: Software Architecture, Composition, and Versioning			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
1-2 The composition of the software, including its software and hardware dependencies, is documented in a bill of materials. **Implementation Notes** The software vendor can choose the format for the bill of materials.	**In Place** ☐		**Not In Place** ☐
1-2.a Examine evidence to **verify** the composition of the software, including its software and hardware dependencies, is accurately documented in a bill of materials.	<Assessor Response>		
1-3 The software is designed and built in a manner that restricts its overall composition to only what is required for its intended functionality.	**In Place** ☐		**Not In Place** ☐
1-3.a Examine vendor documentation to **verify** the software is designed and built in a manner that facilitates its overall composition being restricted to only what is required for its intended functionality. Leverage information from 1-1 and 1-2.	<Assessor Response>		
1-3.b Perform static analysis to **verify** the information from 1-3.a.	<Assessor Response>		
1-3.c Examine evidence of the software-build process to **verify** the information from 1-2.	<Assessor Response>		
1-3.1 This includes all *third-party elements*.	**In Place** ☐		**Not In Place** ☐
1-3.1.a Examine vendor documentation to **verify** the software is designed and built in a manner that only leverages third-party elements as needed for its intended functionality. Leverage information from 1-2 and 1-3.	<Assessor Response>		
1-4 Provenance information is documented to enable tracking across the software supply chain, including the following information, at a minimum: **Implementation Notes:** If information required in 1-4.1 through 1-4.3 cannot reasonably be obtained or otherwise is not available (e.g., an older library), the vendor needs to document or otherwise assert that the lack of information does not introduce any security-impacting risk to the software under assessment.	**In Place** ☐		**Not In Place** ☐
1-4.a Examine vendor documentation to **verify** the provenance information is documented and verify it includes the information required as denoted in 1-4.1 through 1-4.3.	<Assessor Response>		
1-4.1 The original source or supplier.	**In Place** ☐		**Not In Place** ☐
1-4.2 The name or reference as specified by the original source or supplier.	**In Place** ☐		**Not In Place** ☐

Security Objective 1: Software Architecture, Composition, and Versioning			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
1-4.3 The version or unique identifier.	**In Place** ☐	[redacted]	**Not In Place** ☐
1-5 The software versioning schema is documented and in accordance with the *Program*.	**In Place** ☐		**Not In Place** ☐
1-5.a Examine vendor documentation describing the versioning schema of the software and **verify** it is in accordance with the Program.	<Assessor Response>		
1-5.1 If wildcards are being used, the wildcarding schema is explicitly documented and will be implemented per the *Program* and used only for non-security-impacting changes to the software.	**In Place** ☐	**N/A** ☐	**Not In Place** ☐
1-5.1.a Examine vendor documentation describing the wildcarding schema of the software and **verify** it is in accordance with the Program and intended for only non-security impacting changes to the software.	<Assessor Response>		

Security Objective 2: Sensitive Asset Identification

The sensitive assets of the software are identified and documented

Notes:

1 - Refer to the *PCI Secure Software Standard – Sensitive Asset Identification* document for assistance in identifying and documenting sensitive assets.

2 - Accurate and complete identification and documentation for all sensitive assets is crucial—this information is relevant to, and required for, additional security objective sections and their associated security requirements in this standard.

3 - Software vendors are encouraged to identify the sensitive assets early and often in the design of their software to assist in the software being designed with intent to satisfy the security objectives and security requirements in this standard.

4 - If the software contains account data as defined by PCI DSS (which is considered sensitive data within this standard), refer to *Module A – Account Data Protection* in this standard for ***additional*** requirements and information. *Module A – Account Data Protection* does ***not*** circumvent any requirements in the "Core – All Software" section in this standard.

Select the overall Finding for this Security Objective ➔	**In Place** ☐		**Not In Place** ☐

Security Objective 2: Sensitive Asset Identification			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
2-1 The *sensitive data* associated with the software is identified and documented, including the following details, at a minimum:	**In Place** ☐		**Not In Place** ☐
This requirement is tested via 2.1.1 through 2-1.8.x.			
2-1.1 The description and use of all *sensitive data* are documented.	**In Place** ☐		**Not In Place** ☐
2-1.1.a Examine vendor documentation to **verify** it contains details that describe each sensitive data element and its use.	<Assessor Response>		
2-1.1.b Perform static analysis to **verify** the sensitive data identified in 2-1.1.a.	<Assessor Response>		
2-1.1.c Perform static analysis to identify any data elements that satisfy the definition of sensitive data and were not previously identified as sensitive data. The analysis is expected to check for qualifying sensitive data elements that have been unaccounted for. **Verify** that all sensitive data is identified.	<Assessor Response>		
2-1.2 The storage, including storage locations, of all *sensitive data* where storage is permissible, are documented.	**In Place** ☐		**Not In Place** ☐
2-1.2.a Examine vendor documentation to **verify** it contains details that describe whether or not each sensitive data element is, or can be, stored, in addition to the storage locations as applicable.	<Assessor Response>		

Security Objective 2: Sensitive Asset Identification			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
2-1.2.b Verify the sensitive data elements denoted as being stored, or capable of being stored, are permissible for storage. **Testing Notes** The software is analyzed and tested in 3-1.1.	<Assessor Response>		
2-1.3 The retention policies for all *sensitive data* are documented, which includes:	**In Place** ☐	[redacted]	**Not In Place** ☐
2-1.3.a Examine vendor documentation to **verify** it contains details that describe the retention policies attributed to, and implemented for, each sensitive data element that: **Testing Notes** The software is analyzed and tested in 3-1.2.	<Assessor Response>		
ROV Instruction: Use the instruction in 2-1.3.a to test the criteria in 2-1.3.1.a and 2-1.3.2.a.			
2-1.3.1 The retention policies for all *sensitive data* permissible to store in non-volatile memory.	**In Place** ☐	[redacted]	**Not In Place** ☐
2-1.3.1.a Is stored in non-volatile memory. Leverage information from 2-1.2.	<Assessor Response>		
2-1.3.2 The retention policies, for all *sensitive data* that only exists in volatile memory and are never stored, are documented.	**In Place** ☐	[redacted]	**Not In Place** ☐
2-1.3.2.a Only exists in volatile memory. Leverage information from 2-1.2.	<Assessor Response>		
2-1.4 The methods used to securely delete, or otherwise render the *sensitive data* unrecoverable once it is no longer required, are documented. **Implementation Notes:** This applies regardless of the form, e.g., cleartext, encrypted, etc.	**In Place** ☐	[redacted]	**Not In Place** ☐
2-1.4.a Examine vendor documentation to **verify** it contains details that describe the methods implemented for each sensitive data element to render it unrecoverable once it is no longer required. **Testing Notes** The software is analysed and tested in 3-1.3, 3-1.4, and 3-2.	<Assessor Response>		

Security Objective 2: Sensitive Asset Identification			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
2-1.5 The protection classifications for all *sensitive data*, in accordance with the appropriate protection needs for each type and its use in order to facilitate mitigating their unauthorized access, disclosure, modification, and/or misuse, are documented.	**In Place** ☐	[redacted]	**Not In Place** ☐
2-1.5.a Examine vendor documentation to **verify** it contains details that describe the protection classification attributed to each sensitive data element.	<Assessor Response>		
2-1.5.b Based on the evidence **examined** for 2-1.1 and in 2-1.5.a, **verify** the protection classification attribution for each sensitive data element facilitates mitigating its unauthorized access, disclosure, modification, and/or misuse.	<Assessor Response>		
2-1.6 The protection methods, based on the defined protection classifications, for all *sensitive data* to facilitate mitigating their unauthorized access, disclosure, modification, and/or misuse, are documented. **Implementation Notes:** All uses of cryptography to protect *sensitive data* must satisfy the definition of *strong cryptography*.	**In Place** ☐	[redacted]	**Not In Place** ☐
2-1.6.a Examine vendor documentation to **verify** it contains details that describe the protection methods attributed to, and implemented for, each sensitive data element.	<Assessor Response>		
2-1.6.b Examine vendor documentation to **verify** the protection methods employed for each sensitive data element are in accordance with and satisfy the protection classification attributions verified in 2-1.5.	<Assessor Response>		
2-1.7 The data flows for the *sensitive data* entering into, being processed within, and/or transmitted out of the software are documented	**In Place** ☐	[redacted]	**Not In Place** ☐
2-1.7.a Examine vendor documentation to **verify** the data flows for all sensitive data entering into, being processed within, and/or transmitted out of the software are documented.	<Assessor Response>		
2-1.8 For all cryptographic keys associated with *sensitive assets*, document the following additional information, at a minimum: **Implementation Notes:** All uses of cryptography to protect *sensitive assets* must satisfy the definition of *strong cryptography*.	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective 2: Sensitive Asset Identification			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
2-1.8.a Examine vendor documentation to **verify** it contains details that describe the protection methods attributed to, and implemented for, each sensitive data element.	<Assessor Response>		
2-1.8.1 Key type	**In Place** ☐	[redacted]	**Not In Place** ☐
2-1.8.1.a Examine vendor documentation to **verify** it denotes the key type for each cryptographic key.	<Assessor Response>		
2-1.8.2 Associated cryptographic algorithm	**In Place** ☐	[redacted]	**Not In Place** ☐
2-1.8.2.a Examine vendor documentation to **verify** it denotes the associated cryptographic algorithm for each cryptographic key.	<Assessor Response>		
2-1.8.3 Associated key management schema	**In Place** ☐	[redacted]	**Not In Place** ☐
2-1.8.3.a Examine vendor documentation to **verify** it denotes the associated key management schema for each cryptographic key.	<Assessor Response>		
2-1.8.4 Key length	**In Place** ☐	[redacted]	**Not In Place** ☐
2-1.8.4.a Examine vendor documentation to **verify** it denotes the associated key length for each cryptographic key.	<Assessor Response>		
2-1.8.5 Generation Method & Origin	**In Place** ☐	[redacted]	**Not In Place** ☐
2-1.8.5.a Examine vendor documentation to **verify** it describes the generation method for each cryptographic key, including the origin.	<Assessor Response>		
2-1.8.6 Destruction Method	**In Place** ☐	[redacted]	**Not In Place** ☐
2-1.8.6.a Examine vendor documentation to **verify** it describes the destruction method for each cryptographic key.	<Assessor Response>		
2-1.8.7 All associations with other *sensitive data*, as applicable.	**In Place** ☐	[redacted]	**Not In Place** ☐
2-1.8.7.a Examine vendor documentation to **verify** it denotes, as applicable, all associations between each cryptographic key to other sensitive data elements. Leverage the information examined as part of 2-1.1.	<Assessor Response>		
2-1.8.8 All associations with *sensitive resources*, as applicable.	**In Place** ☐	[redacted]	**Not In Place** ☐
2-1.8.8.a Examine vendor documentation to **verify** it denotes, as applicable, all associations between each cryptographic key to sensitive resources. Leverage the information examined as part of 2-2.1.	<Assessor Response>		
2-1.8.9 All associations with *sensitive functionality*, as applicable.	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective 2: Sensitive Asset Identification			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
2-1.8.9.a Examine vendor documentation to **verify** it denotes, as applicable, all associations between each cryptographic key with specific sensitive functionality. Leverage the information examined as part of 2-3.1.	<Assessor Response>		
2-2 The *sensitive resources* associated with the software are identified and documented, including the following details, at a minimum:	**In Place** ☐	[redacted]	**Not In Place** ☐
This requirement is tested via 2.2.1 through 2-2.8.	[redacted]		
2-2.1 The description and use of all *sensitive resources* are documented.	**In Place** ☐	[redacted]	**Not In Place** ☐
2-2.1.a Examine vendor documentation to **verify** it contains details that describe each sensitive resource and its use.	<Assessor Response>		
2-2.1.b Perform static analysis to **verify** the sensitive resources identified in 2-2.1.a.	<Assessor Response>		
2-2.1.c Perform static analysis to identify any resources that satisfy the definition of a sensitive resource and were not previously identified as a sensitive resource. The analysis is expected to check for qualifying sensitive resources that have been unaccounted for. **Verify** that all sensitive resources are identified. **Testing Notes:** The test requirement 2-2.1.c is intended as a means to corroborate the sensitive resources claimed by the vendor against the source code of the software product under assessment. A reasonable analysis should check for obvious qualifying sensitive resources that have been unaccounted for.	<Assessor Response>		
2-2.2 All *sensitive data* associated with each *sensitive resource* are documented.	**In Place** ☐	[redacted]	**Not In Place** ☐
2-2.2.a Examine vendor documentation to **verify** it denotes, as applicable, all sensitive data associated with each sensitive resource. Leverage the information examined as part of 2-1.1.	<Assessor Response>		
2-2.3 The storage, including storage locations, of all *sensitive resources* where storage is permissible are documented.	**In Place** ☐	[redacted]	**Not In Place** ☐
2-2.3.a Examine vendor documentation to **verify** it contains details that describe whether or not each sensitive resource is, or can be, stored, in addition to the storage locations as applicable.	<Assessor Response>		

Security Objective 2: Sensitive Asset Identification			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
2-2.3.b Verify the sensitive resources that can be stored are permissible for storage	<Assessor Response>		
2.2.3.c Examine vendor documentation to determine if there are any sensitive data elements contained within the sensitive resource and **verify** those sensitive data elements are permitted for storage.	<Assessor Response>		
2-2.4 The retention policies for all *sensitive resources* are documented, which includes:	**In Place** ☐	[redacted]	**Not In Place** ☐
2-2.4.a Examine vendor documentation to **verify** it contains details that describe the retention policies attributed to, and implemented for, each sensitive resource that is**:**	<Assessor Response>		
2-2.4.1 The retention policies for all *sensitive resources* permissible to store in non-volatile memory.	**In Place** ☐	[redacted]	**Not In Place** ☐
2-2.4.1.a Stored in non-volatile memory.	<Assessor Response>		
2-2.4.2 The retention policies for all *sensitive resources* that are not permissible to store that reside in volatile memory.	**In Place** ☐	[redacted]	**Not In Place** ☐
2-2.4.2.a Stored in volatile memory.	<Assessor Response>		
2-2.5 The methods used to securely delete, or otherwise render the *sensitive resources* unrecoverable once they are no longer required, are documented.	**In Place** ☐	[redacted]	**Not In Place** ☐
2-2.5.a Examine vendor documentation to **verify** it contains details that describe the methods implemented for each sensitive resource to securely delete it or otherwise render it unrecoverable once it is no longer required.	<Assessor Response>		
2-2.6 The protection classification for all *sensitive resources,* in accordance with the appropriate protection needs for each type and its use, are documented.	**In Place** ☐	[redacted]	**Not In Place** ☐
2-2.6.a Examine vendor documentation to **verify** it contains details that describe the protection classification attributed to each sensitive resource.	<Assessor Response>		
2-2.6.b Based on the evidence **examined** for 2-2.1 and in 2-2.6.a, **verify** the protection classification attribution for each sensitive resource facilitates mitigating its unauthorized access, disclosure, modification, and/or misuse.	<Assessor Response>		

Security Objective 2: Sensitive Asset Identification			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
2-2.7 The protection methods for all *sensitive resources,* to facilitate mitigating their unauthorized access, disclosure, modification, and/or misuse, are documented. **Implementation Notes:** All uses of cryptography to protect *sensitive resources* must satisfy the definition of *strong cryptography*.	**In Place** ☐	[redacted]	**Not In Place** ☐
2-2.7.a Examine vendor documentation to **verify** it contains details that describe the protection methods attributed to, and implemented for, each sensitive resource.	<Assessor Response>		
2-2.7.b Examine vendor documentation to **verify** the protection methods employed for each sensitive resource are in accordance with and satisfy the protection classification attributions verified in 2-2.6.	<Assessor Response>		
2-2.8 The resource flows for the *sensitive resources* entering into, being processed within, and/or transmitted out of the software are documented.	**In Place** ☐	[redacted]	**Not In Place** ☐
2-2.8.a Examine vendor documentation to **verify** the resource flows for all sensitive resources entering into, being processed within, and/or transmitted out of the software are documented.	<Assessor Response>		
2-3 The *sensitive functionality* of the software is identified and documented, including the following details, at a minimum:	**In Place** ☐	[redacted]	**Not In Place** ☐
This requirement is tested via 2.3.1 through 2-3.6. **Testing Notes** Identifying all sensitive functionality is required for the software assessment and is essential information used for numerous requirements throughout this standard.	[redacted]		
2-3.1 The description and use of all *sensitive functionality* are documented.	**In Place** ☐	[redacted]	**Not In Place** ☐
2-3.1.a Examine vendor documentation to **verify** it contains details that describe the sensitive functionality and its use.	<Assessor Response>		
2-3.1.b Perform static analysis to **verify** the sensitive functionality identified in 2-3.1.a.	<Assessor Response>		

Security Objective 2: Sensitive Asset Identification			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
2-3.1.c Perform static analysis to identify any functionality that satisfies the definition of sensitive functionality and was not previously identified as sensitive functionality. The analysis is expected to check for qualifying sensitive functionality that has been unaccounted for. **Verify** that all sensitive functionality is identified. **Testing Notes:** The test requirement 2-3.1.c is intended as a means to corroborate the sensitive functionality claimed by the vendor against the source code of the software product under assessment. A reasonable analysis should check for obvious qualifying sensitive functionality that has been unaccounted for.	<Assessor Response>		
2-3.2 The external accessibility of the *sensitive functionality* is documented.	**In Place** ☐	**N/A** ☐	**Not In Place** ☐
2-3.2.a Examine vendor documentation to **verify** it contains details that describe the capability of external accessibility for all applicable sensitive functionality.	<Assessor Response>		
2-3.2.1 *Sensitive functionality* is only externally accessible where required, and the externally-accessible interface is as limited as possible.	**In Place** ☐	**N/A** ☐	**Not In Place** ☐
2-3.2.1.a Examine vendor documentation to **verify** the sensitive functionality that is externally accessible is required for the functionality of the software.	<Assessor Response>		
2-3.2.1.b Examine vendor documentation to **verify** the exposure of sensitive functionality that is externally accessible is as limited as possible.	<Assessor Response>		
2-3.3 The software design and implementation that facilitates mitigating the unauthorized access, disclosure, modification, and/or misuse of all *sensitive functionality*, as appropriate, are documented. **Implementation Notes:** This requirement is specific to sensitive functionality, whereas requirements in 5-1[.x] are specific to the overall security architecture of the software itself. There may be overlap that can be leveraged. All uses of cryptography to protect sensitive assets, which includes the software itself, must satisfy the definition of *strong cryptography*.	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective 2: Sensitive Asset Identification			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
2-3.3.a Examine vendor documentation to **verify** it contains details that describe how the software is designed and implemented to facilitate mitigating the unauthorized access, disclosure, modification, and/or misuse of all sensitive functionality, as appropriate. Leverage the information examined as part of 2-3.1 and 2-3.2.	<Assessor Response>		
2-3.4 All *sensitive data* associated with the *sensitive functionality* is documented.	**In Place** ☐	[redacted]	**Not In Place** ☐
2-3.4.a Examine vendor documentation to **verify** it denotes all sensitive data elements associated with the sensitive functionality. Leverage the information examined as part of 2-1.1 and 2-3.1.	<Assessor Response>		
2-3.5 All *sensitive resources* associated with the *sensitive functionality* are documented.	**In Place** ☐	[redacted]	**Not In Place** ☐
2-3.5.a Examine vendor documentation to **verify** it denotes all sensitive resources associated with the sensitive functionality. Leverage the information examined as part of 2-2.1 and 2-3.1.	<Assessor Response>		
2-3.6 All *sensitive functionality* that is a *sensitive mode of operation* are documented.	**In Place** ☐	**N/A** ☐	**Not In Place** ☐
ROV Instruction: <u>All 2-3.6.x test requirements are expected to be conducted</u>. If the assessment of security requirement 2-3.6 results in determining the finding is 'N/A', then security requirement 2-3.6 can be marked as 'N/A' (with the appropriate assessor response justifying the 'N'A finding documented in **<u>each</u>** 2-3.6.x test requirement as it relates to the prescribed test activity), provided the criteria for the use of 'N/A' as described herein is satisfied. I.e., even for a 'N/A' finding, each test requirement is expected to have been conducted and include an appropriate assessor response. This analysis and documentation are mandatory, as an 'N/A' finding here is referenced and relied upon as it relates to other Security Objectives and Security Requirements herein.			
2-3.6.a Examine vendor documentation to **verify** it denotes all sensitive functionality that is a sensitive mode of operation. Leverage the information examined as part of 2-3.1.	<Assessor Response>		
2-3.6.b Perform static analysis to **verify** the sensitive modes of operation identified in 2-3.6.a.	<Assessor Response>		
2-3.6.c Perform static analysis to identify any functionality that satisfies the definition of sensitive modes of operation and was not previously identified as a sensitive mode of operation. The analysis is expected to check for qualifying sensitive modes of operation that have been unaccounted for. **Verify** that all sensitive modes of operation are identified.	<Assessor Response>		

Security Objective 3: Sensitive Asset Storage and Retention

Sensitive assets are stored in a secure manner as appropriate to their data type/use and retained for only as long as necessary.

Notes: Leverage the relative documented information from Security Objective 2 once it is verified.

Select the overall Finding for this Security Objective ➔	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective 3: Sensitive Asset Storage and Retention			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
3-1 *Sensitive data* that is capable of being stored is:	**In Place** ☐	**N/A** ☐	**Not In Place** ☐
ROV Instruction: If the assessment of security requirement 2-1.1 results in determining the software product does not have any sensitive data and security requirement 2-1 is marked as 'N/A' (with the appropriate assessor response justifying the 'N'A' finding documented in **each** **2-1.1.a/b/c** test requirement as it relates to the prescribed test activity), then 3-1 can be marked as 'N/A'. For an 'N/A' finding for 3-1 via assessing 2-1.1, the remaining 3-1.x and 3-2 requirements can then be left blank. ***Note:*** *An 'N/A' finding due to zero sensitive data being identified is considered to be significant and of low probability.*			
3-1.a Examine vendor documentation to **verify** the requirements in 3-1.1 through 3-1.4. Leverage the information examined and verified in Security Objective 2, in particular all relevant requirements in 2-1.x, 2-2.x, and 2-3.x, to assist in the assessment.	<Assessor Response>		
3-1.1 Only stored if storage of the *sensitive data* type is permissible and only as necessary.	**In Place** ☐	[redacted]	**Not In Place** ☐
3-1.1.a Perform static analysis to **verify** that all sensitive data capable of being stored is only stored as necessary and is permissible to be stored.	<Assessor Response>		
3-1.1.b Perform static analysis to identify all sensitive data elements that are capable of being stored that were not previously identified as being stored. **Verify** at this point that all sensitive data capable of being stored is accounted for. The intent here is to uncover any sensitive data elements that have not been previously identified.	<Assessor Response>		
3-1.1.c Perform dynamic analysis to **verify** that all sensitive data capable of being stored is only stored as necessary and is permissible to be stored.	<Assessor Response>		
3-1.1.d Perform dynamic analysis to **verify** that all sensitive data that is not permissible to be stored is not capable of being stored.	<Assessor Response>		
3-1.2 Protected during storage in accordance with its type and use, which also takes into account that:	**In Place** ☐	[redacted]	**Not In Place** ☐
3-1.2.a Perform static analysis to **verify** that all sensitive data capable of being stored is protected in accordance with its type and use.	<Assessor Response>		

Security Objective 3: Sensitive Asset Storage and Retention			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
3-1.2.b Perform dynamic analysis to verify that all sensitive data capable of being stored is protected in accordance with its type and use. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the defined and implemented protection methods.	<Assessor Response>		
3-1.2.1 *Strong cryptography* is used where cryptography is implemented or required to protect *sensitive data* in storage.	**In Place** ☐	[redacted]	**Not In Place** ☐
3-1.2.1.a Perform static analysis to **verify** that cryptography leveraged to protect sensitive data in storage satisfies the definition of strong cryptography.	<Assessor Response>		
3-1.2.1.b Perform static and/or dynamic analysis as necessary to **verify** that the use of strong cryptography being leveraged to protect sensitive data in storage cannot be violated, bypassed, or otherwise circumvented.	<Assessor Response>		
3-1.3 Stored in accordance with the defined retention policies.	**In Place** ☐	[redacted]	**Not In Place** ☐
3-1.3.a Perform static analysis to **verify** the sensitive data is stored in accordance with the defined retention policies. Leverage the information from 2-1.3[.x].	<Assessor Response>		
3-1.3.b Perform dynamic analysis to **verify** that sensitive data is retained in accordance with the defined retention policies. Testing includes **but is not limited to**: – Exercising all configurable retention parameters. – Boundary testing all configurable retention periods. – Attempting to violate, bypass, or otherwise circumvent the defined retention policies. **Testing Notes** The testing and analysis performed for 3-1.3 will be related to, and leveraged for, the testing and analysis in 3-1.4.	<Assessor Response>		
3-1.4 Only stored until it is no longer necessary, at which time it is securely deleted, else it is rendered unrecoverable.	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective 3: Sensitive Asset Storage and Retention			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
3-1.4.a Perform static analysis to **verify** the sensitive data is securely deleted once it is no longer necessary to retain. If this is not possible due to a legitimate and verified technical constraint, then **verify** the sensitive data is rendered unrecoverable.	<Assessor Response>		
3-1.4.b Perform dynamic analysis to **verify** the analysis and findings in 3-1.4.a. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the methods employed to securely delete or render the sensitive data unrecoverable. – Where a technical constraint is stated, verify the technical constraint. – Attempting to recover sensitive data after being securely deleted or otherwise rendered unrecoverable.	<Assessor Response>		
3-2 *Sensitive data* is only retained in non-persistent memory for the duration necessary, after which time it is securely deleted, else it is rendered unrecoverable.	**In Place** ☐	[redacted]	**Not In Place** ☐
3-2.a Perform static analysis to **verify** the sensitive data is securely deleted once it is no longer necessary to retain. If this is not possible due to a legitimate and verified technical constraint, then **verify** the sensitive data is rendered unrecoverable.	<Assessor Response>		
3-2.b Perform dynamic analysis to **verify** the analysis and findings in 3-2.a. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the methods employed to securely delete or render the sensitive data unrecoverable. – Where a technical constraint is stated, verify the technical constraint. – Attempting to recover sensitive data after being securely deleted or otherwise rendered unrecoverable.	<Assessor Response>		
3-3 *Sensitive resources* that are capable of being stored are:	**In Place** ☐	[redacted]	**Not In Place** ☐
3-3.a Examine vendor documentation to **verify** the requirements in 3-3.1 through 3-3.4. Leverage the information examined and verified in 2-1.x, 2-2.x, and 2-3.x to assist in the assessment.	<Assessor Response>		

Security Objective 3: Sensitive Asset Storage and Retention			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
3-3.1 Only stored if storage of the *sensitive resource* is permissible, and only as necessary.	**In Place** ☐	[redacted]	**Not In Place** ☐
3-3.1.a Perform static analysis to **verify** that all sensitive resources capable of being stored are only stored as necessary and are permissible to be stored.	<Assessor Response>		
3-3.1.b Perform static analysis to identify all sensitive resources that are capable of being stored that were not previously identified as being stored. **Verify** at this point that all sensitive resources capable of being stored are accounted for. The intent here is to uncover any sensitive resources that have not been previously identified.	<Assessor Response>		
3-3.1.c Perform dynamic analysis to **verify** that all sensitive resources capable of being stored are only stored as necessary and are permissible to be stored.	<Assessor Response>		
3-3.1.d Perform dynamic analysis to **verify** that all sensitive resources that are not permissible to be stored are not capable of being stored.	<Assessor Response>		
3-3.2 Protected during storage in accordance with its type and use, which also takes into account that:	**In Place** ☐	[redacted]	**Not In Place** ☐
3-3.2.a Perform static analysis to **verify** that all sensitive resources capable of being stored are protected in accordance with their type and use.	<Assessor Response>		
3-3.2.b Perform dynamic analysis to **verify** that all sensitive resources capable of being stored are protected in accordance with their type and use. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the defined and implemented protection methods.	<Assessor Response>		
3-3.2.1 *Strong cryptography* is used where cryptography is implemented or required to protect the sensitive resource in storage.	**In Place** ☐	[redacted]	**Not In Place** ☐
3-3.2.1.a Perform static analysis to **verify** that cryptography leveraged to protect sensitive resources in storage satisfies the definition of strong cryptography.	<Assessor Response>		

Security Objective 3: Sensitive Asset Storage and Retention			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
3-3.2.1.b Perform static and/or dynamic analysis as necessary to **verify** that the use of strong cryptography being leveraged to protect sensitive resources in storage cannot be violated, bypassed, or otherwise circumvented.	<Assessor Response>		
3-3.3 Stored in accordance with the defined retention policies.	**In Place** ☐	[redacted]	**Not In Place** ☐
3-3.3.a Perform static analysis to **verify** the sensitive resources are stored in accordance with the defined retention policies that are reviewed and verified from all applicable requirements in Security Objective 2, in particular 2-2.4[.x].	<Assessor Response>		
3-3.3.b Perform dynamic analysis to **verify** that sensitive resources are retained in accordance with the defined retention policies. Testing includes, but is not limited to: – Exercising all configurable retention parameters. – Boundary testing all configurable retention periods. – Attempting to violate, bypass, or otherwise circumvent the defined retention policies. **Testing Notes** The testing and analysis performed for 3-3.3 will be related to, and leveraged for, the testing and analysis in 3-3.4.	<Assessor Response>		
3-3.4 Only stored until they are no longer necessary, at which time they are securely deleted, else they are rendered unrecoverable.	**In Place** ☐	[redacted]	**Not In Place** ☐
3-3.4.a Perform static analysis to **verify** the sensitive resources are securely deleted once they are no longer necessary to retain. If this is not possible due to a legitimate and verified technical constraint, then **verify** the sensitive resources are rendered unrecoverable.	<Assessor Response>		

Security Objective 3: Sensitive Asset Storage and Retention			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
3-3.4.b Perform dynamic analysis to **verify** the analysis and findings in 3-3.4.a. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the methods employed to securely delete or render the sensitive resources unrecoverable. – Where a technical constraint is stated, verify the technical constraint. – Attempting to recover sensitive resources after being securely deleted or otherwise rendered unrecoverable.	<Assessor Response>		
3-4 *Sensitive resources* are only retained in non-persistent memory for the duration necessary, at which time they are securely deleted, else they are rendered unrecoverable.	**In Place** ☐	[redacted]	**Not In Place** ☐
3-4.a Perform static analysis to **verify** the sensitive resources are securely deleted once they are no longer necessary to retain. If this is not possible due to a legitimate and verified technical constraint, then **verify** the sensitive resources are rendered unrecoverable.	<Assessor Response>		
3-4.b Perform dynamic analysis to **verify** the analysis and findings in 3-4.a. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the methods employed to securely delete or render the sensitive resources unrecoverable. – Where a technical constraint is stated, verify the technical constraint. – Attempting to recover sensitive resources after being securely deleted or otherwise rendered unrecoverable.	<Assessor Response>		

Security Objective 4: Sensitive Modes of Operation

The software securely implements sensitive modes of operation.

Notes: Refer to the *PCI Secure Software Standard – Sensitive Asset Identification* document for information regarding sensitive modes of operation.

The software is not required to implement a sensitive mode of operation; however, if it contains sensitive functionality that meets the definition of a sensitive mode of operation, then the requirements here in Security Objective 4 apply.

Select the overall Finding for this Security Objective ➔	**In Place** ☐	**N/A** ☐	**Not In Place** ☐

ROV Instruction: If the assessment of security requirement 2-3.6 results in determining the software does not implement a sensitive mode of operation and is therefore marked and documented appropriately and accurately as 'N/A', then Security Objective 4 can be marked as 'N/A', provided the criteria for the use of 'N/A' as described herein is satisfied. The remainder of this section and the 4-x requirements can then be left blank.

Security Objective 4: Sensitive Modes of Operation			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
4-1 *Sensitive modes of operation* are designed to facilitate mitigating their unauthorized access and minimizing their misuse during authorized access, which includes but is not limited to the following:	**In Place** ☐	[redacted]	**Not In Place** ☐
4-1.a Examine vendor documentation to **verify** the requirements in 4-1.1 through 4-1.9. Leverage the information examined and verified in Security Objective 2, in particular from 2-3.[x], to assist in the assessment.	<Assessor Response>		
4-1.1 *Strong authentication* is required for all access to a *sensitive mode of operation*.	**In Place** ☐	[redacted]	**Not In Place** ☐
4-1.1.a Examine vendor documentation to **verify** for each sensitive mode of operation that strong authentication is required before access is granted.	<Assessor Response>		
4-1.1.b Perform static analysis to **verify** the information from 4-1.1.a.	<Assessor Response>		
4-1.1.c Perform dynamic analysis to **verify** the analysis and findings from 4-1.1.a/b. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the implemented strong authentication mechanisms.	<Assessor Response>		
4-1.2 Implement a defined maximum number of failed access attempts in a defined period of time.	**In Place** ☐	[redacted]	**Not In Place** ☐
4-1.2.a Examine vendor documentation to **verify** a defined maximum number of failed access attempts within a defined period of time is implemented for each sensitive mode of operation.	<Assessor Response>		

Security Objective 4: Sensitive Modes of Operation			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
4-1.2.b Perform static analysis to **verify** the information from 4-1.2.a.	<Assessor Response>		
4-1.2.c Perform dynamic analysis to **verify** the analysis and findings from 4-1.2.a/b. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the threshold limits and/or duration of time it occurs in.	<Assessor Response>		
4-1.3 Implement a defined lockout period, initiated upon the allowable maximum number of failed access attempts in a defined period of time being reached.	**In Place** ☐	[redacted]	**Not In Place** ☐
4-1.3.a Examine vendor documentation to **verify** a defined lockout period is initiated upon the maximum number of failed access attempts within a defined period of time being reached for each sensitive mode of operation.	<Assessor Response>		
4-1.3.b Perform static analysis to **verify** the information from 4-1.3.a.	<Assessor Response>		
4-1.3.c Perform dynamic analysis to **verify** the analysis and findings from 4-1.3.a/b. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the defined lockout periods.	<Assessor Response>		
4-1.4 Are designed in a manner that failed access attempts do not disclose information that can assist in gaining unauthorized access.	**In Place** ☐	[redacted]	**Not In Place** ☐
4-1.4.a Examine vendor documentation to **verify** for each sensitive mode of operation that failed access attempts do not disclose information that can assist in gaining unauthorized access.	<Assessor Response>		
4-1.4.b Perform static analysis to **verify** the information from 4-1.4.a.	<Assessor Response>		
4-1.4.c Perform dynamic analysis to **verify** the analysis and findings from 4-1.4.a/b. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the methods employed in order to obtain useful information that can be used to gain unauthorized access.	<Assessor Response>		
4-1.5 Implement a timeout based on a defined period of inactivity, upon which the software exits the *sensitive mode of operation* and effectively returns to normal operation.	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective 4: Sensitive Modes of Operation			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
4-1.5.a Examine vendor documentation to **verify** for each sensitive mode of operation that an inactivity timeout has been implemented, upon which the software exits the sensitive mode of operation and returns to normal operation.	<Assessor Response>		
4-1.5.b Perform static analysis to **verify** the information from 4-1.5.a.	<Assessor Response>		
4-1.5.c Perform dynamic analysis to **verify** the analysis and findings from 4-1.5.a/b. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the implemented inactivity timeouts.	<Assessor Response>		
4-1.6 Implement a defined maximum duration of use timeout, upon which the software exits the *sensitive mode of operation* and effectively returns to normal operation.	**In Place** ☐	[redacted]	**Not In Place** ☐
4-1.6.a Examine vendor documentation to **verify** for each sensitive mode of operation that a maximum duration of use timeout has been implemented, upon which the software exits the sensitive mode of operation and returns to normal operation.	<Assessor Response>		
4-1.6.b Perform static analysis to **verify** the information from 4-1.6.a.	<Assessor Response>		
4-1.6.c Perform dynamic analysis to **verify** the analysis and findings from 4-1.6.a/b. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the implemented maximum duration of use timeouts.	<Assessor Response>		
4-1.7 The software is designed to retain, or facilitate the retention of, a record of all failed and successful access to *sensitive modes of operation*. **Implementation Notes** The software can create the record or otherwise provide the required and pertinent information such that a record can be created. The event information is the essential aspect that needs to be accounted for in 4-1.7[.x].	**In Place** ☐	[redacted]	**Not In Place** ☐
4-1.7.a Examine vendor documentation to **verify** the requirements in 4-1.7.1 through 4-1.7.7.	<Assessor Response>		
4-1.7.1 The records include information that can uniquely identify each failed access attempt.	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective 4: Sensitive Modes of Operation			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
4-1.7.1.a Examine vendor documentation to **verify** for each sensitive mode of operation that records include information that can uniquely identify each failed access attempt.	<Assessor Response>		
4-1.7.1.b Perform static analysis to **verify** the information from 4-1.7.1.a.	<Assessor Response>		
4-1.7.1.c Perform dynamic analysis to **verify** the analysis and findings from 4-1.7.1.a/b. Testing should include, but is not limited to: – Attempting to access each sensitive mode of operation using invalid information and verifying the subsequent record creation for the unique failed access attempt.	<Assessor Response>		
4-1.7.2 The records include information that can uniquely identify each successful access event, including traceability to the entity that access was granted to.	**In Place** ☐	[redacted]	**Not In Place** ☐
4-1.7.2.a Examine vendor documentation to **verify** for each sensitive mode of operation that records include information that can uniquely identify each successful access event, including the entity that access was granted to.	<Assessor Response>		
4-1.7.2.b Perform static analysis to **verify** the information from 4-1.7.2.a.	<Assessor Response>		
4-1.7.2.c Perform dynamic analysis to **verify** the analysis and findings from 4-1.7.2.a/b. Testing should include, but is not limited to: – Accessing each sensitive mode of operation using valid information and verifying the subsequent record creation for the unique successful access event.	<Assessor Response>		
4-1.7.3 The records include information that can uniquely identify the net effect, or change, resulting from access to the *sensitive mode of operation*.	**In Place** ☐	[redacted]	**Not In Place** ☐
4-1.7.3.a Examine vendor documentation to **verify** for each sensitive mode of operation that records include information that can uniquely identify the net effect, or change, resulting from access to the sensitive mode of operation.	<Assessor Response>		
4-1.7.3.b Perform static analysis to **verify** the information from 4-1.7.3.a.	<Assessor Response>		

Security Objective 4: Sensitive Modes of Operation			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
4-1.7.3.c Perform dynamic analysis to **verify** the analysis and findings from 4-1.7.3.a/b. Testing should include, but is not limited to: – Accessing each sensitive mode of operation using valid information and verifying the subsequent record creation for the unique successful access event.	<Assessor Response>		
4-1.7.4 The software is designed to protect these records from compromise using *strong cryptography*.	**In Place** ☐	[redacted]	**Not In Place** ☐
4-1.7.4.a Examine vendor documentation to **verify** for each sensitive mode of operation that records are protected using strong cryptography.	<Assessor Response>		
4-1.7.4.b Perform static analysis to **verify** the information from 4-1.7.4.a.	<Assessor Response>		
4-1.7.4.c Perform dynamic analysis to **verify** the analysis and findings from 4-1.7.4.a/b. Testing should include, but is not limited to: – Attempting to bypass the protection mechanisms to gain access to cleartext records and/or the relative information.	<Assessor Response>		
4-1.7.5 The software is designed to require *strong authentication* to access these records.	**In Place** ☐	[redacted]	**Not In Place** ☐
4-1.7.5.a Examine vendor documentation to **verify** for each sensitive mode of operation that strong authentication is required to access associated records.	<Assessor Response>		
4-1.7.5.b Perform static analysis to **verify** the information from 4-1.7.5.a.	<Assessor Response>		
4-1.7.5.c Perform dynamic analysis to **verify** the analysis and findings from 4-1.7.5.a/b. Testing should include, but is not limited to: – Attempting to bypass the authentication mechanisms to gain access to records and/or the relevant information.	<Assessor Response>		
4-1.7.6 The records are retained for a defined retention period. **Implementation Notes** The software is not required to retain the records on the same system the software resides. The records can be offloaded elsewhere, in either physical and/or logical form. However, doing so still requires the records to be protected.	**In Place** ☐	[redacted]	**Not In Place** ☐
4-1.7.6.a Examine vendor documentation to **verify** for each sensitive mode of operation that records are retained for a defined retention period.	<Assessor Response>		

Security Objective 4: Sensitive Modes of Operation			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
4-1.7.6.b Perform static analysis to **verify** the information from 4-1.7.6.a.	<Assessor Response>		
4-1.7.6.c Perform dynamic analysis to **verify** the analysis and findings from 4-1.7.6.a/b. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the implemented record-retention parameters.	<Assessor Response>		
4-1.7.7 Records transmitted outside the software are protected in accordance with requirement 6-2.	**In Place** ☐	[redacted]	**Not In Place** ☐
4-1.7.7.a Verify that records associated with sensitive modes of operation (requirements 4-1.7[.x]) have been accounted for in the assessment of requirement 6-2[.x].	<Assessor Response>		
4-1.8 *Sensitive modes of operation* are designed in accordance with requirement 5-4.3.1.	**In Place** ☐	[redacted]	**Not In Place** ☐
4-1.8.a Verify that each sensitive mode of operation has been accounted for in the assessment of requirement 5-4.3.1 regarding secure authorization.	<Assessor Response>		
4-1.9 *Sensitive modes of operation* are designed in accordance with requirement 5-5.3.1.	**In Place** ☐	[redacted]	**Not In Place** ☐
4-1.9.a Verify that each sensitive mode of operation has been accounted for in the assessment of requirement 5-5.3.1 regarding mitigating inadvertently disclosing, exposing, or otherwise leaking sensitive assets.	<Assessor Response>		

Security Objective 5: Sensitive Asset Protection Mechanisms

The software is designed to protect itself and its underlying sensitive assets.

Notes: The mechanisms that can be employed to satisfy these requirements will be specific to each unique software implementation, including the particular programming languages used, third-party elements used, and underlying platform considerations. The security requirements are designed to provide the necessary flexibility for the software vendor in designing their software to satisfy this security objective.

Select the overall Finding for this Security Objective ➔	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective 5: Sensitive Asset Protection Mechanisms			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
5-1 Platform-based security mechanisms relied upon by the software to facilitate protecting *sensitive assets* have been evaluated. **Implementation Notes** Leveraging underlying platform-based security mechanisms is not required; however, their use does not supersede or otherwise replace any security requirements in this standard.	**In Place** ☐	**N/A** ☐	**Not In Place** ☐
ROV Instruction: If the assessment of security requirement 5-1 results in determining the finding is 'N/A', then security requirement 5-1 can be marked as 'N/A' (with the appropriate assessor response justifying the 'N'A finding documented in 5-1.a), provided the criteria for the use of 'N/A' as described herein is satisfied. The remaining 5-1 test requirements can then be left blank.			
5-1.a Examine vendor documentation to **verify** the platform mechanisms used, and to what extent they are relied on, to protect sensitive assets.	<Assessor Response>		
5-1.b Examine vendor documentation and all necessary additional evidence to **verify** that where security mechanisms leveraged by the software under assessment to this standard rely on previous approvals or certifications of the underlying platform, that the evidence and scope of the prior evaluation/certification are sufficient to ensure the required security to the sensitive assets they are being utilized to protect. **Testing Notes** Leverage this information as is relevant to the requirements in this section for Security Objective 5. This requirement does not prevent the use of underlying security mechanisms that do not have evidence of a prior evaluation; however, where any evidence of a prior evaluation does not exist, further assessment to confirm the security features being relied upon is expected.	<Assessor Response>		
5-2 The software is designed to facilitate pre-emptively mitigating *anomalous behavior* from occurring in order to protect sensitive assets, which also includes, but is not limited to:	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective 5: Sensitive Asset Protection Mechanisms			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
5-2.a Examine vendor documentation to **verify** how the software is designed to facilitate pre-emptively mitigating anomalous behavior from occurring to protect sensitive assets.	<Assessor Response>		
5-2.b Perform static analysis to **verify** the information from 5-2.a.	<Assessor Response>		
5-2.c Perform dynamic analysis to **verify** the analysis and findings from 5-2.a/b. Testing should include, but is not limited to: – Attempting to bypass or otherwise circumvent the implemented mechanisms.	<Assessor Response>		
5-2.1 Facilitating the mitigation of *anomalous behavior* as a result of input from external sources.	**In Place** ☐	[redacted]	**Not In Place** ☐
5-2.1.a Examine vendor documentation to **verify** how the software is designed to facilitate mitigating anomalous behavior as a result of input from external sources.	<Assessor Response>		
5-2.1.b Perform static analysis to **verify** the information from 5-2.1.a.	<Assessor Response>		
5-2.1.c Perform dynamic analysis to **verify** the analysis and findings from 5-2.1.a/b. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the implemented mechanisms. – Attempting to purposefully manipulate inputs with intent to cause risk to sensitive assets. **Testing Notes** The testing strategies will be highly contingent on, and should be catered to: the type of software, the programming language(s), the specific design and interfaces, etc. The goal is to verify there is demonstrable evidence that mechanisms are in place and seemingly effective at satisfying the requirement.	<Assessor Response>		
5-2.2 Facilitating the mitigation of *anomalous behavior* as a result of error conditions.	**In Place** ☐	[redacted]	**Not In Place** ☐
5-2.2.a Examine vendor documentation to **verify** how the software is designed to facilitate mitigating anomalous behavior as a result of expected error conditions.	<Assessor Response>		
5-2.2.b Perform static analysis to **verify** the information from 5-2.2.a.	<Assessor Response>		

Security Objective 5: Sensitive Asset Protection Mechanisms			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
5-2.2.c Perform dynamic analysis to **verify** the analysis and findings from 5-2.2.a/b. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the implemented mechanisms. – Attempting to purposefully manipulate the software with intent to cause risk to sensitive assets due to unexpected consequences of errors. **Testing Notes** The testing strategies will be highly contingent on, and should be catered to: the type of software, the programming language(s), the specific design and architecture, etc. The goal is to verify there is demonstrable evidence that mechanisms are in place and seemingly effective at satisfying the requirement.	<Assessor Response>		
5-2.3 Facilitating the mitigation of *anomalous behavior* as a result of retrieving or receiving externally-hosted *third-party elements* during runtime.	**In Place** ☐	[redacted]	**Not In Place** ☐
5-2.3.a Examine vendor documentation to **verify** if the software is capable of retrieving or receiving externally-hosted third-party elements during runtime.	<Assessor Response>		
5-2.3.b Perform static analysis to **verify** the information from 5-2.3.a. If it is determined that the software is capable of retrieving or receiving externally-hosted third-party elements during runtime, assess the remaining test requirements below.	<Assessor Response>		
5-2.3.c Leverage the information from 5-2.3.a/b and **perform** static analysis to **verify** the mechanisms implemented to facilitate the mitigation of anomalous behavior as a result of retrieving or receiving externally-hosted third-party elements during runtime.	<Assessor Response>		
5-2.3.d Perform dynamic analysis to **verify** the analysis and findings from 5-2.3.c. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the implemented mechanisms. For example, by forcing the software to retrieve third-party elements without any protective/defensive mechanisms being employed. This can include fetching an element with an invalid signature, no signature, malformed code, etc.	<Assessor Response>		

Security Objective 5: Sensitive Asset Protection Mechanisms			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
5-3 The software is designed to facilitate detecting suspected *anomalous behavior* in order to protect *sensitive assets*, which also includes, but is not limited to: **Implementation Notes** This is related to 5-2; however, it is not identical in intent. 5-2 facilitates designing the software to prevent anomalous behavior from ever occurring. However, as that is statistically impossible to achieve absolutely, this requirement 5-3 facilitates identifying and securely handling unexpected behavior should it occur.	**In Place** ☐	[redacted]	**Not In Place** ☐
5-3.a Examine vendor documentation to **verify** how the software is designed to facilitate detecting suspected anomalous behavior in order to protect sensitive assets.	<Assessor Response>		
5-3.b Perform static analysis to **verify** the information from 5-3.a.	<Assessor Response>		
5-3.c Perform dynamic analysis to **verify** the analysis and findings from 5-3.a/b. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the implemented mechanisms. **Testing Notes** The testing strategies will be highly contingent on, and should be catered to: the type of software, the programming language(s), the specific design and architecture, etc. The goal is to verify there is demonstrable evidence that mechanisms are in place and seemingly effective at satisfying the requirement.	<Assessor Response>		
5-3.1 The software is designed to facilitate mitigating, or at least minimizing, the impact of suspected *anomalous behavior,* or otherwise fails in a secure manner.	**In Place** ☐	[redacted]	**Not In Place** ☐
5-3.1.a Examine vendor documentation to **verify** how the software is designed to facilitate mitigating, or at least minimizing, the impact of suspected anomalous behavior, or otherwise fails in a secure manner.	<Assessor Response>		
5-3.1.b Perform static analysis to **verify** the information from 5-3.1.a.	<Assessor Response>		

Security Objective 5: Sensitive Asset Protection Mechanisms			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
5-3.1.c Perform dynamic analysis to **verify** the analysis and findings from 5-3.1.a/b. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the implemented mechanisms. **Testing Notes** The testing strategies will be highly contingent on, and should be catered to: the type of software, the programming language(s), the specific design and architecture, etc. The goal is to verify there is demonstrable evidence that mechanisms are in place and seemingly effective at satisfying the requirement.	<Assessor Response>		
5-3.2 The software is designed to provide an immediate indication of suspected *anomalous behavior*.	**In Place** ☐	[redacted]	**Not In Place** ☐
5-3.2.a Examine vendor documentation to **verify** how the software is designed to provide an immediate indication of suspected anomalous behavior.	<Assessor Response>		
5-3.2.b Perform static analysis to **verify** the information from 5-3.2.a.	<Assessor Response>		
5-3.2.c Perform dynamic analysis to **verify** the analysis and findings from 5-3.2.a/b. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the implemented mechanisms.	<Assessor Response>		
5-3.2.1 The mechanism used to provide an indication of suspected *anomalous behavior* is protected against compromise.	**In Place** ☐	[redacted]	**Not In Place** ☐
5-3.2.1.a Examine vendor documentation to **verify** how the software is designed to protect the indication of suspected anomalous behavior against compromise.	<Assessor Response>		
5-3.2.1.b Perform static analysis to **verify** the information from 5-3.2.1.a.	<Assessor Response>		
5-3.2.1.c Perform dynamic analysis to **verify** the analysis and findings from 5-3.2.1.a/b. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the implemented mechanisms.	<Assessor Response>		

Security Objective 5: Sensitive Asset Protection Mechanisms			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
5-3.3 The software is designed to retain, or facilitate the retention of, a record of suspected events of *anomalous behavior*. **Implementation Notes** The software can create the record or otherwise provide the required and pertinent information such that a record can be created. The event information is the essential aspect that needs to be accounted for in 5-3.3[.x].	**In Place** ☐	[redacted]	**Not In Place** ☐
5-3.3.a Examine vendor documentation to **verify** the requirements in 5-3.3.1 through 5-3.3.4.	<Assessor Response>		
5-3.3.1 The records include information that can uniquely identify the suspected *anomalous behavior* event.	**In Place** ☐	[redacted]	**Not In Place** ☐
5-3.3.1.a Examine vendor documentation to **verify** that records of suspected anomalous behavior include information that can uniquely identify each event.	<Assessor Response>		
5-3.3.1.b Perform static analysis to **verify** the information from 5-3.3.1.a.	<Assessor Response>		
5-3.3.1.c Perform dynamic analysis to **verify** the analysis and findings from 5-3.3.1.a/b. Testing should include, but is not limited to: – Triggering, or simulating, an anomalous behavior event and verifying the subsequent record creation for the unique event.	<Assessor Response>		
5-3.3.2 The software is designed to protect these records from compromise using *strong cryptography*.	**In Place** ☐	[redacted]	**Not In Place** ☐
5-3.3.2.a Examine vendor documentation to **verify** that records are protected using strong cryptography.	<Assessor Response>		
5-3.3.2.b Perform static analysis to **verify** the information from 5-3.3.2.a.	<Assessor Response>		
5-3.3.2.c Perform dynamic analysis to **verify** the analysis and findings from 5-3.3.2.a/b. Testing should include, but is not limited to: – Attempting to bypass the protection mechanisms to gain access to cleartext records and/or the relevant information.	<Assessor Response>		
5-3.3.3 The software is designed to require *strong authentication* to access these records.	**In Place** ☐	[redacted]	**Not In Place** ☐
5-3.3.3.a Examine vendor documentation to **verify** that strong authentication is required to access associated records.	<Assessor Response>		
5-3.3.3.b Perform static analysis to **verify** the information from 5-3.3.3.a.	<Assessor Response>		

Security Objective 5: Sensitive Asset Protection Mechanisms			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
5-3.3.3.c Perform dynamic analysis to **verify** the analysis and findings from 5-3.3.3.a/b. Testing should include, but is not limited to: – Attempting to bypass the authentication mechanisms to gain access to records and/or the relevant information.	<Assessor Response>		
5-3.3.4 The records are retained for a defined retention period. **Implementation Notes** The software is not required to retain the records on the same system the software resides. The records can be offloaded elsewhere, in either physical and/or logical form. However, doing so still requires the records to be protected.	**In Place** ☐	[redacted]	**Not In Place** ☐
5-3.3.4.a Examine vendor documentation to **verify** records of suspected anomalous behavior events are retained for a defined retention period.	<Assessor Response>		
5-3.3.4.b Perform static analysis to **verify** the information from 5-3.3.4.a.	<Assessor Response>		
5-3.3.4.c Perform dynamic analysis to **verify** the analysis and findings from 5-3.3.4.a/b. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the implemented record-retention parameters.	<Assessor Response>		
5-3.3.5 Records transmitted outside the software are protected in accordance with requirement 6-2.	**In Place** ☐	[redacted]	**Not In Place** ☐
5-3.3.5.a Verify that records associated with anomalous behavior events (requirements 5-3[.x]) have been accounted for in the assessment of requirement 6-2[.x].	<Assessor Response>		
5-4 The software is designed to facilitate securely implementing *authorized* access to sensitive assets, which includes access to:	**In Place** ☐	[redacted]	**Not In Place** ☐
5-4.a Examine vendor documentation to **verify** that the software is designed to facilitate securely implementing authorized access to sensitive assets, including the avoidance of known authorization-based flaws. Leverage this information for 5-4.[x].	<Assessor Response>		
5-4.1 *Sensitive data*	**In Place** ☐	[redacted]	**Not In Place** ☐
5-4.1.a Perform static analysis to **verify** the context of 5-4 in relation to sensitive data.	<Assessor Response>		

Security Objective 5: Sensitive Asset Protection Mechanisms			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
5-4.1.b Perform dynamic analysis to **verify** the analysis and findings from 5-4.1.a. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the authorized access mechanisms to sensitive data.	<Assessor Response>		
5-4.2 *Sensitive resources*	**In Place** ☐	[redacted]	**Not In Place** ☐
5-4.2.a Perform static analysis to **verify** the context of 5-4 in relation to sensitive resources.	<Assessor Response>		
5-4.2.b Perform dynamic analysis to **verify** the analysis and findings from 5-4.2.a. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the authorized access mechanisms to sensitive resources.	<Assessor Response>		
5-4.3 *Sensitive functionality*, which includes:	**In Place** ☐	[redacted]	**Not In Place** ☐
5-4.3.a Perform static analysis to **verify** the context of 5-4 in relation to sensitive functionality.	<Assessor Response>		
5-4.3.b Perform dynamic analysis to **verify** the analysis and findings from 5-4.3.a. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the authorized access mechanisms to sensitive functionality.	<Assessor Response>		
5-4.3.1 *Sensitive modes of operation*	**In Place** ☐	**N/A** ☐	**Not In Place** ☐
ROV Instruction: If the assessment of security requirement 2-3.6 results in determining the software does not implement a sensitive mode of operation and is therefore marked and documented appropriately and accurately as 'N/A', then security requirement 5-4.3.1 can be marked as 'N/A' above, provided the criteria for the use of 'N/A' as described herein is satisfied. The test requirements 5-4.3.1.x can then be left blank.			
5-4.3.1.a Perform static analysis to **verify** the context of 5-4 in relation to sensitive modes of operation.	<Assessor Response>		
5-4.3.1.b Perform dynamic analysis to **verify** the analysis and findings from 5-4.3.1.a. Testing should include, but is not limited to: – Attempting to violate, bypass, or otherwise circumvent the authorized access mechanisms to sensitive modes of operation.	<Assessor Response>		

Security Objective 5: Sensitive Asset Protection Mechanisms			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
5-5 The software is designed to facilitate mitigating inadvertently disclosing, exposing, or otherwise leaking *sensitive assets*, which includes: **Implementation Notes** There is going to inevitably be partial overlap here with requirements in other Security Objectives (e.g., encrypting sensitive data can help prevent disclosure). Requiring authentication to a sensitive mode of operation can help protect sensitive assets. Implementation according to least privilege can also help. This requirement, however, is more encompassing and overarching in consideration of the overall software architecture. Refer to the guidance for more information.	**In Place** ☐	[redacted]	**Not In Place** ☐
5-5.a Examine vendor documentation to **verify** the software is designed to facilitate mitigating inadvertently disclosing, exposing, or otherwise leaking sensitive assets. This information will be used to assist in the assessment for 5-5.1 through 5-5.3[.x]. **Testing Notes** The intent here is not to reconfirm testing/evidence from other requirements. This is not the same context as 5-4 regarding authorized access. This requirement 5-5 is regarding unexpected disclosure/exposure in unintended ways.	<Assessor Response>		
5-5.1 *Sensitive data*	**In Place** ☐	[redacted]	**Not In Place** ☐
5-5.1.a Perform static analysis to **verify** the software is designed to facilitate mitigating inadvertently disclosing, exposing, or otherwise leaking sensitive data. Leverage, in part, the information from Security Objective 2 regarding sensitive data, including information regarding sensitive functionality related to sensitive data.	<Assessor Response>		
5-5.1.b Perform dynamic analysis to **verify** the analysis and findings from 5-5.1.a. Testing should include, but is not limited to: – Attempting to find unexpected ways to gain access to sensitive data.	<Assessor Response>		
5-5.2 *Sensitive resources*	**In Place** ☐	[redacted]	**Not In Place** ☐
5-5.2.a Perform static analysis to **verify** the software is designed to facilitate mitigating inadvertently disclosing, exposing, or otherwise leaking sensitive resources. Leverage, in part, the information from Security Objective 2 regarding sensitive resources, including information regarding sensitive functionality related to sensitive resources.	<Assessor Response>		

Security Objective 5: Sensitive Asset Protection Mechanisms			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
5-5.2.b Perform dynamic analysis to **verify** the analysis and findings from 5-5.2.a. Testing should include, but is not limited to: – Attempting to find unexpected ways to gain access to sensitive resources.	<Assessor Response>		
5-5.3 *Sensitive functionality*, which includes:	**In Place** ☐	[redacted]	**Not In Place** ☐
5-5.3.a Perform static analysis to **verify** the software is designed to facilitate mitigating inadvertently disclosing, exposing, or otherwise leaking sensitive functionality. Leverage, in part, the information from Security Objective 2 regarding sensitive functionality.	<Assessor Response>		
5-5.3.b Perform dynamic analysis to **verify** the analysis and findings from 5-5.3.a. Testing should include, but is not limited to: – Attempting to find unexpected ways to gain access to sensitive functionality.	<Assessor Response>		
5-5.3.1 *Sensitive modes of operation*	**In Place** ☐	**N/A** ☐	**Not In Place** ☐
ROV Instruction: If the assessment of security requirement 2-3.6 results in determining the software does not implement a sensitive mode of operation and is therefore marked and documented appropriately and accurately as 'N/A', then security requirement 5-5.3.1 can be marked as 'N/A' above, provided the criteria for the use of 'N/A' as described herein is satisfied. The test requirements 5-5.3.1.x can then be left blank.			
5-5.3.1.a Perform static analysis to **verify** the software is designed to facilitate mitigating inadvertently disclosing, exposing, or otherwise leaking sensitive modes of operation. Leverage, in part, the information from Security Objective 2 regarding sensitive modes of operation.	<Assessor Response>		
5-5.3.1.b Perform dynamic analysis to **verify** the analysis and findings from 5-5.3.1.a. Testing should include, but is not limited to: – Attempting to find unexpected ways to gain access to sensitive modes of operation.	<Assessor Response>		

Security Objective 6: Sensitive Asset Output

The software is designed to facilitate the protection of the sensitive assets it outputs.

Notes: "Output" refers to any form of output of a sensitive asset from the software that results in the software essentially relinquishing direct control and protection of that asset. This can include, but is not limited to: transmission over a network, output to the operating system, output to a co-resident application on the same underlying hardware platform, output to shared memory, output through a hardware interface/port, etc.

IMPORTANT: P2PE Applications require the use of the underlying SRED functions of the PTS POI device to encrypt cleartext account data. Refer to Module B.

Select the overall Finding for this Security Objective ➔	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective 6: Sensitive Asset Output			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
6-1 All forms of *sensitive assets* that are capable of being output from the software are identified and documented. **Implementation Notes** While the flow diagrams in 2-1.7 and 2-2.8 are used to document all sensitive data and sensitive resources being output from the software, they are confirmed here in the context of Security Objective 6. This requirement is in regard to any output: cleartext, encrypted, truncated, hashed, and/or any other form.	**In Place** ☐	[redacted]	**Not In Place** ☐
6-1.a Leverage the information from 2-1[.x] and 2-2[.x], in addition to the flow information from 2-1.7 and 2-2.8 and **perform** static analysis to verify all sensitive assets that are capable of being output from the software are identified and documented, including all forms of potential output (encrypted, hashed, truncated, cleartext, etc.). **Testing Notes** This should correlate with information from requirements 2-1[.x] and 2-2[.x]. If it does not, the discrepancies must be reported to the software vendor and subsequently accounted for in the software, all affected documentation, and all affected requirements in this standard.	<Assessor Response>		
6-2 *Strong cryptography* is used to protect *sensitive assets* output from the software using individual data-level and/or session-level protection, which includes:	**In Place** ☐	[redacted]	**Not In Place** ☐
ROV Instruction: Either security requirement 6-2.1 and/or 6-2.2 need to be satisfied – i.e., both cannot be 'N/A'.			
6-2.a Examine vendor documentation to **verify** strong cryptography is used to protect sensitive assets being output from the software.	<Assessor Response>		

Security Objective 6: Sensitive Asset Output			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
6-2.b Based on 6-2.a, leverage the information from 6-1 and **verify** either 6-2.1 and/or 6-2.2 as applicable.	<Assessor Response>		
6-2.1 If data-level encryption is used to protect *sensitive assets* being output from the software, the associated cryptographic algorithms and cryptographic keys are documented and satisfy the use of *strong cryptography*.	**In Place** ☐	**N/A** ☐	**Not In Place** ☐
ROV Instruction: If the assessment of security requirement 6-2.1 results in determining the finding is 'N/A', then security requirement 6-2.1 can be marked as 'N/A' (with the appropriate assessor response justifying the 'N'A finding documented in 6-2.1.a), provided the criteria for the use of 'N/A' as described herein is satisfied. The remaining 6-2.1 test requirements can then be left blank.			
6-2.1.a Examine vendor documentation to **verify** the cryptographic algorithms and associated key lengths are documented and satisfy the use of strong cryptography.	<Assessor Response>		
6-2.1.b Perform static analysis to **verify** the software is designed to encrypt the applicable cleartext sensitive assets identified in 6-1 using strong cryptography.	<Assessor Response>		
6-2.1.c Perform dynamic analysis to **verify** the analysis and findings from 6-2.1.a/b. Testing should include, but is not limited to: – Attempting to obtain applicable sensitive assets being output in cleartext from the software.	<Assessor Response>		
6-2.2 If the software facilitates establishing a channel to transmit *sensitive assets*, a *secure channel* is used, which includes:	**In Place** ☐	**N/A** ☐	**Not In Place** ☐
ROV Instruction: If the assessment of security requirement 6-2.2 results in determining the finding is 'N/A', then security requirement 6-2.2 can be marked as 'N/A' (with the appropriate assessor response justifying the 'N'A finding documented in 6-2.2.a), provided the criteria for the use of 'N/A' as described herein is satisfied. The remaining 6-2.2 test requirements can then be left blank, as well as the remaining 6-2.2[.x] security requirements and their associated test requirements.			
6-2.2.a Examine vendor documentation to **verify** if the software facilitates the establishment of any channels to transmit sensitive assets.	<Assessor Response>		
6-2.2.b Based on 6-2.2.a, leverage the information from 6-1 and **verify** 6-2.2.1 through 6-2.2.7.	<Assessor Response>		
6-2.2.c Perform static analysis to **verify** the analysis and findings from 6-2.2.a/b.	<Assessor Response>		
6-2.2.d Perform dynamic analysis to **verify** the analysis and findings from 6-2.2.a/b/c.	<Assessor Response>		
6-2.2.1 The implementation of each *secure channel* is documented.	**In Place** ☐		**Not In Place** ☐

Security Objective 6: Sensitive Asset Output			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
6-2.2.1.a Examine vendor documentation to **verify** each secure channel is identified and documented.	<Assessor Response>		
6-2.2.2 The endpoints of each *secure channel* are documented.	**In Place** ☐	[redacted]	**Not In Place** ☐
6-2.2.2.a Examine vendor documentation to **verify** the endpoints of each secure channel are identified and documented.	<Assessor Response>		
6-2.2.3 The root of trust used for each *secure channel* is documented.	**In Place** ☐	[redacted]	**Not In Place** ☐
6-2.2.3.a Examine vendor documentation to **verify** the root of trust of each secure channel is identified and documented.	<Assessor Response>		
6-2.2.4 The cryptography supported for each *secure channel* is documented and satisfies the use of *strong cryptography*.	**In Place** ☐	[redacted]	**Not In Place** ☐
6-2.2.4.a Examine vendor documentation to **verify** the cryptography supported for each secure channel is identified, documented, and satisfies the use of strong cryptography.	<Assessor Response>		
6-2.2.5 The establishment of each *secure channel* and how mutual authentication is guaranteed is documented.	**In Place** ☐	[redacted]	**Not In Place** ☐
6-2.2.5.a Examine vendor documentation to **verify** the mutual authentication details of each secure channel are identified and documented.	<Assessor Response>		
6-2.2.6 Secret or private cryptographic keys used to establish and maintain *secure channels* are unique per session, except by chance.	**In Place** ☐	[redacted]	**Not In Place** ☐
6-2.2.6.a Examine vendor documentation to **verify** the secret or private cryptographic keys used to establish and maintain each secure channel are unique per session, except by chance.	<Assessor Response>		
6-2.2.7 The *secure channels* are implemented to mitigate downgrade attacks.	**In Place** ☐	[redacted]	**Not In Place** ☐
6-2.2.7.a Examine vendor documentation to **verify** each secure channel is implemented in a manner that mitigates downgrade attacks.	<Assessor Response>		
6-2.3 *Sensitive assets* are protected with *strong cryptography* if they are capable of being transmitted by the software via end-user messaging technologies.	**In Place** ☐	**N/A** ☐	**Not In Place** ☐
ROV Instruction: If the assessment of security requirement 6-2.3 results in determining the finding is 'N/A', then security requirement 6-2.3 can be marked as 'N/A' (with the appropriate assessor response justifying the 'N'A finding documented in 6-2.3.a), provided the criteria for the use of 'N/A' as described herein is satisfied. The remaining 6-2.3 test requirements can then be left blank.			

Security Objective 6: Sensitive Asset Output		
Security Requirements and Test Requirements		**Assessor's Findings and Observations**
	6-2.3.a Examine vendor documentation to **verify** that if sensitive assets are capable of being transmitted using end-user messaging technologies that they are protected using strong cryptography.	<Assessor Response>
	6-2.3.b Perform static analysis to **verify** the information from 6-2.3.a.	<Assessor Response>
	6-2.3.c Perform dynamic analysis to **verify** the information from 6-2.3.a/b.	<Assessor Response>

Security Objective 7: Random Numbers

Random numbers used in association with sensitive assets are generated using a process that ensures sufficient entropy and lack of statistical correlation.

Notes: The software is not required to use random values; however, if it does, in association with sensitive assets, then this Security Objective 7 applies.

For P2PE Applications, be advised of requirement B-2.7, which requires the use of the underlying PTS POI device's RNG.

Select the overall Finding for this Security Objective ➔	**In Place** ☐	**N/A** ☐	**Not In Place** ☐

ROV Instruction: If the assessment of security requirement 7-1 results in determining the software does not implement random values associated with sensitive assets, Security Objective 7 and security requirement 7-1 can be marked as 'N/A' (with the appropriate assessor response justifying the 'N'A finding documented in 7-1.a), provided the criteria for the use of 'N/A' as described herein is satisfied. The remaining 7-x requirements can then be left blank.

Security Objective 7: Random Numbers			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
7-1 The *sensitive assets* associated with random numbers are identified and documented.	**In Place** ☐	**N/A** ☐	**Not In Place** ☐
7.1.a Examine vendor documentation to **verify** if the software uses random values, and if so, all correlations with the sensitive assets identified and verified in Security Objective 2.	<Assessor Response>		
7.1.b Perform static analysis and **verify** the analysis and findings from 7-1.a. All use cases of random values being used in association with sensitive assets must be accounted for.	<Assessor Response>		
7-1.1 The software uses an RNG, including the relative entropy source, that supports and provides the security strength equal to or greater than the security strength of the strongest cryptographic key supported. **Implementation Notes** It is permitted for the software to leverage an RNG implementation provided by the underlying platform, especially if it is a sufficient hardware-based RNG. This can also be used to provide the entropy sources into a DRNG, provided requirement 7-1.1 is satisfied. "Security strength" can be used interchangeably with "effective key strength".	**In Place** ☐	[redacted]	**Not In Place** ☐
7-1.1.a Examine vendor documentation to **verify** the random-number generator implementation, which includes the entropy source, supports the security strength equal to or greater than the security strength of the strongest cryptographic key supported. Demonstrable test evidence must corroborate this finding.	<Assessor Response>		

Security Objective 7: Random Numbers			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
7-1.2 Documentation includes justification for the random-number generator implementation being leveraged and its appropriateness in relation to *sensitive assets*.	**In Place** ☐	[redacted]	**Not In Place** ☐
7-1.2.a Examine vendor documentation to **verify** the justification for the RNG implementation used and its appropriateness in relation to sensitive assets is documented and explained.	<Assessor Response>		
ROV Instruction: P2PE Applications require the use of the underlying PTS POI device's RNG for all account-data related data and functions. Refer to requirement B-2.7. It is permissible for a P2PE Application to implement its own RNG as per below for non-account-data related data and functions.			
7-1.3 If the software is designed with its own random-number generator implementation, then the following applies: **Implementation Notes** By definition, this is a deterministic random-number generator (DRNG) and therefore requires a sufficient entropy source. For software being assessed to Module B, refer to B2-7.	**In Place** ☐	**N/A** ☐	**Not In Place** ☐
ROV Instruction: If the assessment of security requirement 7-1.3 results in determining the finding is 'N/A', then security requirement 7-1.3 can be marked as 'N/A' (with the appropriate assessor response justifying the 'N'A finding documented in 7-1.3.a), provided the criteria for the use of 'N/A' as described herein is satisfied. The remaining 7-1.3[.x] security and test requirements can then be left blank.			
7-1.3.a Examine vendor documentation to **verify** if the software design includes its own random-number generator implementation. If it does, assess 7-1.3[.x].	<Assessor Response>		
7-1.3.1 The DRNG algorithm is designed and implemented based on industry-recognized standards.	**In Place** ☐	[redacted]	**Not In Place** ☐
7-1.3.1.a Examine vendor documentation to **verify** the random-number generator implementation is designed based on industry-recognized standards.	<Assessor Response>		
7-1.3.1.b Perform static analysis to **verify** the analysis and findings from 7-1.3.1.a.	<Assessor Response>		
7-1.3.1.c Perform dynamic analysis to **verify** the analysis and findings from 7-1.3.1.a/b.	<Assessor Response>		
7-1.3.2 A trusted source of entropy is used for seed values being input into the DRNG.	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective 7: Random Numbers			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
7-1.3.2.a Examine vendor documentation to **verify** the source of entropy used for seed values being input into the software RNG (DRNG) is documented.	<Assessor Response>		
7-1.3.3 The [re]seeding period is frequent enough to account for the deterministic nature of the DRNG.	**In Place** ☐	[redacted]	**Not In Place** ☐
7-1.3.3.a Examine vendor documentation to **verify** the seeding period.	<Assessor Response>		
7-1.3.4 The seed values are protected from disclosure and modification using *strong cryptography*.	**In Place** ☐	[redacted]	**Not In Place** ☐
7-1.3.4.a Examine vendor documentation to **verify** the protection mechanisms implemented to facilitate the mitigation of the seed values from disclosure/modification.	<Assessor Response>		
7-1.3.4.b Perform static analysis to **verify** the analysis and findings from 7-1.3.4.a, in the context of the software under assessment, regarding the protection of seed values from their initial existence within the software up to their use in the RNG and their subsequent secure deletion.	<Assessor Response>		

Security Objective 8: Key Management

Cryptographic keys associated with sensitive assets are managed securely.

Notes: These requirements relate to all cryptographic keys associated with sensitive assets, which includes sensitive data, sensitive resources, sensitive functionality, and sensitive modes of operation. Where there is an association between a cryptographic key and a sensitive asset, that inherently qualifies the cryptographic key, and any associated key material, as sensitive data and needs to be identified in Security Objective 2.

Leverage the information from Security Objective 2 regarding cryptographic keys, as well as for certificates that qualify as a sensitive resource.

Select the overall Finding for this Security Objective ➔	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective 8: Key Management			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
8-1 Cryptographic keys that are generated by the software use an entropy source as input that is at least equal to the intended effective strength of the key being generated. **Implementation Notes** The software is not required to generate its own cryptographic keys; however, if it does, then the generation process must be sufficient, including the sourced entropy.	**In Place** ☐	**N/A** ☐	**Not In Place** ☐
ROV Instruction: If the assessment of security requirement 8-1 results in determining the finding is 'N/A', then security requirement 8-1 can be marked as 'N/A' (with the appropriate assessor response justifying the 'N'A finding documented in 8-1.a), provided the criteria for the use of 'N/A' as described herein is satisfied. The remaining 8-1 test requirements can then be left blank.			
8-1.a Examine vendor documentation to **verify** if the software implements cryptographic key generation.	<Assessor Response>		
8-1.b Examine vendor documentation to **verify** the source and expected entropy used in the key generation process.	<Assessor Response>		
8-1.c Perform static analysis to **verify** the analysis and findings from 8-1.a/b.	<Assessor Response>		
8-1.d Examine vendor documentation and necessary evidence/testing to **verify** the expected entropy used in the key generation process is at least equal to the intended effective strength of the key(s) being generated. If evidence is not provided, then **perform** all necessary testing to verify the requirement is satisfied. **Testing Notes** Leverage the information from Security Objective 7 as applicable.	<Assessor Response>		

Security Objective 8: Key Management			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
8-2 Secret and private cryptographic keys are established and maintained in the software in a manner that facilitates their confidentiality.	**In Place** ☐	**N/A** ☐	**Not In Place** ☐
ROV Instruction: If the assessment of security requirement 2-1.8[.x] results in determining the software does not utilize secret or private keys in association with sensitive assets, then security requirement 8-2 can be marked as 'N/A' (with the appropriate assessor response justifying the 'N'A finding documented in 8-2.a), provided the criteria for the use of 'N/A' as described herein is satisfied. The remaining 8-2.1 security and test requirements can then be left blank.			
8-2.a Perform static analysis to **verify** the confidentiality of cryptographic keys is protected.	<Assessor Response>		
8-2.1 Cleartext cryptographic keys are not stored in non-volatile memory.	**In Place** ☐		**Not In Place** ☐
8-2.1.a Perform static analysis to **verify** cleartext cryptographic keys are not stored in non-volatile memory.	<Assessor Response>		
8-3 Cryptographic keys are only used for a single, predetermined purpose.	**In Place** ☐		**Not In Place** ☐
8-3.a Perform static analysis to **verify** cryptographic keys are only used for a single, predetermined purpose.	<Assessor Response>		
8-3.b Perform dynamic analysis to **verify** the analysis and findings from 8-3.a. Attempt to use keys for more than one purpose, or otherwise for a purpose they are not defined to be used for.	<Assessor Response>		
8-4 Cryptographic keys are only protected with a key of equal or greater strength. **Implementation Notes** There are two considerations: 1. If a lesser bit-strength key (Key_1) is needed to encrypt a greater bit-strength key (Key_2), the effective key strength of Key_2 is reduced to that of Key_1. E.g., if a 2048-bit RSA is used to encrypt a 128 or 256-bit AES key, the resulting effective key strength will be that of the 2048-bit RSA key (i.e., 112 bits). 2. Strong cryptography is still required.	**In Place** ☐		**Not In Place** ☐
8-4.a Perform static analysis to **verify** cryptographic keys are only protected with a key of equal or greater strength. Factor in the allowable considerations in the *Implementation Notes*.	<Assessor Response>		

Security Objective 8: Key Management			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
8-4.b Perform dynamic analysis to **verify** the analysis and findings from 8-4.a. Attempt to use keys (or key strengths) that are inadequate to protect other keys and are not in accordance with requirement 8-4.	<Assessor Response>		
8-5 Public keys are protected for integrity and authenticity and are authenticated before they are used.	**In Place** ☐	**N/A** ☐	**Not In Place** ☐
ROV Instruction: If the assessment of security requirement 2-1.8[.x] results in determining the software does not utilize public keys in association with sensitive assets, then security requirement 8-5 can be marked as 'N/A' (with the appropriate assessor response justifying the 'N'A finding documented in 8-5.a), provided the criteria for the use of 'N/A' as described herein is satisfied. The remaining 8-5 test requirements can then be left blank.			
8-5.a Leverage the information from Security Objective 2 regarding cryptographic keys. **Perform** static analysis to **verify** public keys are protected for integrity and authenticity and are authenticated before they are used.	<Assessor Response>		
8-5.b Perform dynamic analysis to **verify** the analysis and findings from 8-5.a. Attempt to use public keys in a manner that is not in accordance with requirement 8-5.	<Assessor Response>		
8-6 Cryptographic-key derivation and key-check functions are required to be implemented:	**In Place** ☐	**N/A** ☐	**Not In Place** ☐
8-6.a Leverage information from Security Objective 2, in particular from 2-3[.x], for 8-6.1 and 8-6.2.	<Assessor Response>		
8-6.1 As one-way functions.	**In Place** ☐		**Not In Place** ☐
8-6.1.a Perform static analysis to **verify** cryptographic key derivation and key check functions are implemented as one-way functions.	<Assessor Response>		
8-6.2 In a manner that does not expose information about the cryptographic keys used in the derivation or check process.	**In Place** ☐		**Not In Place** ☐
8-6.2.a Perform static analysis to **verify** cryptographic key derivation and key check functions are implemented in a manner that does not expose information about the cryptographic keys used in the derivation or check process.	<Assessor Response>		
8-7 The software facilitates the capability to revoke or otherwise cease all use of suspected compromised cryptographic keys/certificates.	**In Place** ☐		**Not In Place** ☐

Security Objective 8: Key Management		
	Security Requirements and Test Requirements	**Assessor's Findings and Observations**
	8-7.a Examine vendor documentation to **verify** software facilitates the capability to revoke or otherwise cease all use of suspected compromised cryptographic keys/certificates.	<Assessor Response>
	8-7.b Perform static analysis to **verify** the analysis and findings from 8-7.a.	<Assessor Response>

Security Objective 9: Cryptography

The use of cryptography in association with sensitive assets satisfies the definition of strong cryptography.

Notes: Strong cryptography is the minimum baseline allowed for the use of cryptography regarding sensitive assets. Refer to the definition of strong cryptography in the 'terminology' section in this standard.

Select the overall Finding for this Security Objective ➔	**In Place** ☐	**N/A** ☐	**Not In Place** ☐

ROV Instruction: **All 9-1.x test requirements are expected to be conducted.** If the assessment of security requirement 9-1 results in determining the finding is 'N/A', then security requirement 9-1 can be marked as 'N/A' (with the appropriate assessor response justifying the 'N'A finding documented in **each** 9-1.x test requirement as it relates to the prescribed test activity), provided the criteria for the use of 'N/A' as described herein is satisfied. I.e., even for a 'N/A' finding, each test requirement is expected to have been conducted and include an appropriate assessor response. If 9-1 is determined to be 'N/A', then Security Objective 9 can be marked as 'N/A' herein.

Security Objective 9: Cryptography			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
9-1 The use of cryptography related to *sensitive assets* that is not already accounted for elsewhere in this standard satisfies the definition of *strong cryptography*.	**In Place** ☐	**N/A** ☐	**Not In Place** ☐
9-1.a Examine vendor documentation to **verify** if the software is leveraging cryptography, in relation to sensitive assets, that is not otherwise already accounted for by the other explicit requirements in this standard.	<Assessor Response>		
9-1.b Perform static analysis to **verify** the analysis and findings from 9-1.a.	<Assessor Response>		
9-1.c Verify, based on the analysis and findings from 9-1.a/b, that either: 1. There is no additional use of cryptography, in relation to sensitive assets, that is not otherwise accounted for elsewhere in this standard. Or, 2. There is additional use of cryptography, in relation to sensitive assets, that is not otherwise accounted for elsewhere in this standard, and it satisfies the definition of strong cryptography. This should relate to information captured in Security Objective 2.	<Assessor Response>		

Security Objective 10: Threats and Vulnerabilities

The software is designed and updated to account for known threats and vulnerabilities in order to protect sensitive assets.

Notes: [No Notes]

Select the overall Finding for this Security Objective ➔	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective 10: Threats and Vulnerabilities			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
10-1 The software is designed and updated as needed to mitigate known threats and vulnerabilities that could pose risks to sensitive assets.	**In Place** ☐	[redacted]	**Not In Place** ☐
10-1.a Examine vendor documentation to **verify** the software is designed to mitigate known threats and vulnerabilities that could pose risks to sensitive assets.	<Assessor Response>		
10-1.b Examine vendor documentation to **verify** that processes exist to update the software as needed to account for newly-discovered relevant threats and vulnerabilities.	<Assessor Response>		
10-1.1 Known security issues and vulnerabilities are accounted for in the following, at a minimum:	**In Place** ☐	[redacted]	**Not In Place** ☐
10-1.1.a Leverage the information from 10-1 to assess 10-1.1.[1-3].	<Assessor Response>		
10-1.1.1 The programming languages used to develop the software. **Implementation Notes** The term "programming language" is being used generically. This includes all constructs that generally consist of defined syntax and associated semantics being used to "create/implement" the software.	**In Place** ☐	[redacted]	**Not In Place** ☐
10-1.1.1.a Examine vendor documentation to **verify** the software is designed with an awareness of avoiding known security issues based on the underlying programming languages used.	<Assessor Response>		
10-1.1.1.b Perform static analysis in order to **verify** the software is designed in accordance with 10-1.1.	<Assessor Response>		

Security Objective 10: Threats and Vulnerabilities			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
10-1.1.2 *Third-party elements* **Implementation Notes** There is an obvious limit of knowledge and degree of association of third-party elements used by the software. The scope here is all third-party elements where the software vendor has a choice regarding their use by, or within, the software.	**In Place** ☐	[redacted]	**Not In Place** ☐
10-1.1.2.a Examine vendor documentation to **verify** the software is designed with an awareness of avoiding known security issues in all third-party elements being used by, or within, the software. Leverage information from Security Objective 1 regarding the composition of the software, as well as noted dependencies, including the documented provenance information.	<Assessor Response>		
10-1.1.2.b Perform static analysis to **verify** the software is designed in accordance with requirement 10-1.1.2 and the information from 10-1.1.2.a.	<Assessor Response>		
10-1.1.3 Protocols **Implementation Notes** It is possible this might overlap with 10-1.1.1 and/or 10-1.1.2. However, the context of protocols and their potential risk warrants being accounted for explicitly.	**In Place** ☐	[redacted]	**Not In Place** ☐
10-1.1.3.a Examine vendor documentation to **verify** the software is designed with an awareness of avoiding known security issues in all utilized protocols.	<Assessor Response>		
10-1.1.3.b Perform static analysis, and research as needed, to determine if there are known security-relevant issues with any of the protocols being used by the software. **Verify** the software is designed in accordance with requirement 10-1.1.3 and the information from 10-1.1.3.a.	<Assessor Response>		
10-2 Unsupported dependencies are either avoided, or the potential risk of their use to the software and *sensitive assets* is demonstrably mitigated.	**In Place** ☐	[redacted]	**Not In Place** ☐
ROV Instruction: If the assessment of security requirement 10-2 results in determining unsupported dependencies are in fact avoided, then security requirement 10-2 can be marked as 'In Place' (with the appropriate assessor response justifying the 'In Place' finding documented in 10-2.a), provided the criteria for the use of 'In Place' as described herein is satisfied. The remaining 10-2 test requirements can then be left blank.			

Security Objective 10: Threats and Vulnerabilities		
	Security Requirements and Test Requirements	**Assessor's Findings and Observations**
	10-2.a Examine vendor documentation to **verify** if the software utilizes unsupported dependencies. Leverage the relevant information from Security Objective 1.	<Assessor Response>
	10-2.b Examine vendor documentation to **verify** the potential risk relative to the use of the unsupported dependencies is documented and accounted for.	<Assessor Response>
	10-2.c Perform static analysis, to verify the findings and analysis from 10-2.b is appropriate and seemingly effective in accounting for the potential risk of the use of the documented unsupported dependencies.	<Assessor Response>

Security Objective 11: Secure Deployment and Management

Software is deployed and managed in a secure manner.

Notes: [No Notes]

Select the overall Finding for this Security Objective ➔	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective 11: Secure Deployment and Management			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
11-1 The processes involving the release and delivery of the software facilitate maintaining its security and integrity.	**In Place** ☐	[redacted]	**Not In Place** ☐
11-1.a Examine vendor documentation to **verify** the processes involving the release and delivery of the software are intended to facilitate maintaining the security and integrity of the software.	<Assessor Response>		
11-2 The processes to provide security-relevant software updates facilitate prompt deployment to all affected customers.	**In Place** ☐	[redacted]	**Not In Place** ☐
11-2.a Examine vendor documentation to **verify** the processes involving the release and delivery of security-relevant software updates facilitate prompt deployment to affected customers.	<Assessor Response>		
11-3 The software vendor maintains and provides current secure implementation guidance to customers, which includes, at a minimum, instructions regarding performing the following procedures in a secure manner:	**In Place** ☐	[redacted]	**Not In Place** ☐
11-3.a Examine vendor documentation, including the implementation guidance, to **verify** the requirements in 11-3.1 through 11-3.7.	<Assessor Response>		
11-3.b Leveraging the information from 11-3.a, **perform** the functions in 11-3.1 through 11-3.7 by following the implementation guidance. **Verify** the accuracy of the implementation guidance in completing each activity in a secure manner.	<Assessor Response>		
11-3.1 Installation	**In Place** ☐	[redacted]	**Not In Place** ☐
11-3.2 Integration	**In Place** ☐		**Not In Place** ☐
11-3.3 Configuration	**In Place** ☐		**Not In Place** ☐
11-3.4 Operation	**In Place** ☐		**Not In Place** ☐
11-3.5 Updates	**In Place** ☐		**Not In Place** ☐
11-3.6 Removal	**In Place** ☐		**Not In Place** ☐

Security Objective 11: Secure Deployment and Management			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
11-3.7 Version information retrieval	**In Place** ☐		**Not In Place** ☐
11-4 Mechanisms are implemented to verify the integrity of the software as part of the initial installation and for all updates.	**In Place** ☐		**Not In Place** ☐
11-4.a Examine vendor documentation to **verify** the mechanisms that are implemented to verify the integrity of the software as part of the initial installation and for updates.	<Assessor Response>		
11-4.b Perform dynamic analysis to **verify** the information from 11-4.a. Testing should include attempts to bypass or circumvent the integrity verification mechanisms.	<Assessor Response>		
11-5 The software provides a mechanism to provide its software version information.	**In Place** ☐		**Not In Place** ☐
11-5.a Leverage the information from 11-3.7 and **verify** that the software provides a mechanism to provide its software version information.	<Assessor Response>		
11-6 The software is designed to force changing all default authentication values/credentials/accounts that facilitate access to sensitive assets upon software installation, initialization, or otherwise before business use.	**In Place** ☐		**Not In Place** ☐
11-6.a Leverage the information from 11-3.[x] and **perform** static analysis to **verify** the software is designed to force changing all default authentication values/credentials/accounts that facilitate access to sensitive assets upon software installation, initialization, or otherwise before business use.	<Assessor Response>		
11-6.b Perform dynamic analysis to **verify** the information from 11-6.a. Testing should include attempts to leverage default values/credentials/accounts to gain access to sensitive assets.	<Assessor Response>		

Module A – Account-Data Protection

Additional security requirements for software that stores, processes, and/or transmits account data (as defined in PCI DSS).

ROV Instruction: A P2PE Application by definition has the potential to access cleartext account data. Therefore, Module A is mandatory for P2PE Application assessments. In addition, a P2PE Application is required to use the underlying PTS POI device's SRED functionality to securely encrypt and transmit account data. Note that PCI DSS requirements do not supersede the explicit requirements in Module B for the encryption and transmission of account data.

Affirm that it is understood that Module A is mandatory for P2PE Applications and must be accounted for in the assessment.	➔	☐ Affirm

Security Objective A1: Securing Account Data

Sensitive Authentication Data (SAD) and Primary Account Numbers (PANs) are handled in accordance with PCI DSS requirements.

Notes: The handling of PAN and SAD by the software is accounted for generically in relative requirements regarding sensitive data in the "Core – All Software" section in this standard. Refer to the *PCI Secure Software Standard – Sensitive Asset Identification* document for additional assistance regarding PAN and SAD.

These requirements in "Module A – Account-Data Protection" are intended to facilitate explicitly satisfying respective requirements regarding PAN and SAD in PCI DSS. Refer to the latest version of PCI DSS for complete information and expectations regarding the handling of PAN and SAD as it pertains to the software under assessment for this PCI Secure Software Standard.

Select the overall Finding for this Security Objective ➔	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective A1: Securing Account Data			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
A1-1 *SAD* is handled in accordance with PCI DSS requirements.	**In Place** ☐	[redacted]	**Not In Place** ☐
A1-1.a Examine vendor documentation to **verify** how SAD is managed within the software. Leverage all requisite information from the requirements in Security Objective 2 as is relevant.	<Assessor Response>		
A1-1.b Based on the use of SAD by the software, **examine** the latest version of PCI DSS and determine the requirements that apply to the software based on its use of SAD.	<Assessor Response>		
A1-1.c Based on evidence from A1-1.a/b, **perform** static and/or dynamic analysis on the software as necessary to **verify** the software manages SAD in accordance with the latest version of PCI DSS.	<Assessor Response>		
A1-2 *PAN* is handled in accordance with PCI DSS requirements.	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective A1: Securing Account Data		
	Security Requirements and Test Requirements	**Assessor's Findings and Observations**
	A1-2.a Examine vendor documentation to **verify** how PAN is managed within the software. Leverage all requisite information from the requirements in Security Objective 2 as is relevant.	<Assessor Response>
	A1-2.b Based on the use of PANs by the software, **examine** the latest version of PCI DSS and determine the requirements that apply to the software based on its use of PANs.	<Assessor Response>
	A1-2.c Based on evidence from A1-2.a/b, **perform** static and/or dynamic analysis as necessary to **verify** the software manages PANs in accordance with the latest version of PCI DSS.	<Assessor Response>

Module B – POI Device Software

Additional security requirements for software intended for deployment and execution on POI devices that have been evaluated and approved in accordance with the PCI PTS POI Standard and Program.

ROV Instruction: A P2PE Application by definition is required to be deployed on eligible PTS POI devices with SRED. Therefore, Module B is mandatory for P2PE Application assessments.

Note: *Support for non-SRED PTS POI devices, including individual hardware and/or firmware, is not permissible for P2PE Applications.*

Affirm that it is understood that Module B is mandatory for P2PE Applications and must be accounted for in the assessment.	➔	☐ Affirm

Security Objective B1: PTS Approval

The software is designed for the secure integration and operation on an approved PTS POI device.

Notes: Refer to the *PCI SSC's List of Approved PTS Devices at:* https://listings.pcisecuritystandards.org/assessors_and_solutions/pin_transaction_devices

Select the overall Finding for this Security Objective ➔	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective B1: PTS Approval			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
B1-1 The software is intended for deployment and operation on devices that have been evaluated and approved per the PCI PTS POI Standard and its Program.	**In Place** ☐	[redacted]	**Not In Place** ☐
B1-1.a Examine the software vendor's documentation to **verify** the PTS POI devices the software is intended to operate on.	<Assessor Response>		
B1-1.b Examine the *PCI SSC List of Approved PTS Devices* to **verify** the following information for each PTS device model: – Model Name/Number – Hardware Version Number(s) – Firmware Version Number (s) – Application ("Applic") Number(s) – SRED (mandatory, including all HW and FW) – Open Protocols (OP) (as applicable) – PTS approval number(s)	<Assessor Response>		

Security Objective B2: Approved POI Device Functionality

The software is designed to use approved functionality of the POI device's hardware and firmware.

Notes: It is not the intent to "reassess" the PTS POI firmware. The intent is to verify the functionality of the software being assessed to this standard as it relates to the PTS POI device, which includes its already-approved hardware and firmware.

Depending on the POI devices being supported, which include the individual HW/FW, it may be possible that both B2-1 and/or B2-2 apply to the software. For example, there could be a build-time option that includes different functionality (depending on the target platform), which might result in different functionality, depending on the underlying HW/FW POI platform.

Select the overall Finding for this Security Objective ➔	**In Place** ☐		**Not In Place** ☐

Security Objective B2: Approved POI Device Functionality			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
ROV Instruction: Support for non-SRED PTS POI devices, including individual hardware and/or firmware, is not permissible for P2PE Applications. Therefore, requirement B2-1 does not pertain to, and is not an option for, P2PE Application assessments.			
If the POI device's HW/FW is not approved to SRED: **B2-1** The software uses *strong cryptography* to protect *account data*.			
ROV Instruction: If the assessment of security requirement B2-1 results in determining that **all** supported POI device HW and FW **is** approved to SRED, then security requirement B2-1 can be marked as 'N/A' (with the appropriate assessor response justifying the 'N'A finding documented in B2-1.a), provided the criteria for the use of 'N/A' as described herein is satisfied. The remaining B2-1 test requirements can then be left blank.			
B2-1.a Examine the software vendor's documentation to **verify** all mechanisms used to protect account data, both from the underlying POI device platform and within the software itself. Leverage all information/testing from relevant *Core – All Software* requirements.			
B2-1.b Perform static analysis [of the software being assessed] to **verify** that all use of cryptography leveraged to protect account data satisfies the definition of strong cryptography.			

Security Objective B2: Approved POI Device Functionality			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
B2-1.c Perform dynamic analysis as necessary to **verify** that the use of strong cryptography being leveraged to protect account data cannot be violated, bypassed, or otherwise circumvented. **Testing Notes** Testing needs to determine the functionality of the underlying POI HW/FW that is being leveraged, in addition to all relevant functionality implemented in the software being assessed. The analysis and testing required will depend on the unique instance of the implementation. The intent is not to reassess the POI device's firmware. All firmware functionality being called on or otherwise leveraged by the software can be noted and confirmed. However, all relevant functionality being implemented in the software directly needs to be properly assessed.			
ROV Instruction: P2PE Applications must rely on the SRED functions of the underlying PTS POI device to accept and encrypt cleartext account data. A P2PE Application is not permitted to encrypt cleartext account data. The analysis of B2-2 must include verifying at a minimum that (a) the P2PE Application leverages the underlying PTS POI device's SRED functionality to accept and encrypt cleartext account data, and that (b) the P2PE Application does not contain any functionality to directly accept and/or encrypt cleartext account data.			
B2-2 The software does not bypass or circumvent the PTS-approved SRED-related functions of the POI device.	**In Place** ☐	[redacted]	**Not In Place** ☐
B2-2.a Examine the software vendor's documentation to **verify** how the software is designed to leverage the PTS POI device's SRED-approved functionality for each relevant PTS POI device, including the specific HW/FW approved to SRED.	<Assessor Response>		
B2-2.b Perform static analysis to **verify** B2-2.a.	<Assessor Response>		
B2-2.c Perform dynamic analysis as deemed necessary to **verify** B2-2.a/b.	<Assessor Response>		
ROV Instruction: P2PE Applications must rely on the SRED functions of the underlying PTS POI device to accept and encrypt cleartext account data. Therefore, requirement B2-2[.x] does not pertain, and is not an option, for P2PE Application assessments.			
B2-2.1 If the software requires *account data* related encryption functionality that the POI device's firmware and SRED-related functionality do not provide, then:			

Security Objective B2: Approved POI Device Functionality			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
ROV Instruction: If the assessment of security requirement B2-2.1 results in a finding of 'N'A, then security requirement B2-2.1 can be marked as 'N/A' (with the appropriate assessor response justifying the 'N'A finding documented in B2-2.1.a), provided the criteria for the use of 'N/A' as described herein is satisfied. The remaining B2-2.1[.x] security requirements can then be left blank.			
B2-2.1.a Examine the software vendor's documentation to **verify** if the software implements additional account data-related encryption functionality. If it does, assess B2-2.1.[.x].			
B2-2.1.1 The software vendor documents the limitations of the POI device's firmware and SRED functionality, including the additional functionality necessary to be implemented in the software directly.			
B2-2.1.1.a Examine the software vendor's documentation to **verify** the noted limitations of the POI device's firmware and SRED functionality, including the additional functionality deemed necessary to be implemented in the software directly.			
B2-1.1.1.b Perform static analysis [of the software being assessed] to **verify** the additional functionality deemed necessary to be implemented in the software directly.			
B2-2.1.2 *Account data* is protected in accordance with B2-1.			
B2-2.1.2.a Verify account data is protected in accordance with B2-1.			
B2-3 The software does not share *cleartext account data* directly with other non-firmware software on the POI device.	**In Place** ☐	[redacted]	**Not In Place** ☐
B2-3.a Examine the software vendor's documentation to **verify** the software does not contain any functionality to share cleartext account data with any other software other than the PTS POI device's firmware.	<Assessor Response>		
B2-3.b Perform static analysis to **verify** B2-3.a.	<Assessor Response>		
B2-3.c Perform dynamic analysis as deemed necessary to **verify** B2-3.a/b.	<Assessor Response>		
B2-4 The software does not output *cleartext account data* outside of the POI device, which also includes:	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective B2: Approved POI Device Functionality			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
B2-4.a Examine the software vendor's documentation to **verify** the software does not contain any functionality that is capable of outputting cleartext account data from the PTS POI device.	<Assessor Response>		
B2-4.b Perform static analysis to **verify** B2-4.a.	<Assessor Response>		
B2-4.c Perform dynamic analysis to **verify** B2-4.a/b.	<Assessor Response>		
B2-4.1 The software does not facilitate the visual presentation of *cleartext account data.*	**In Place** ☐	[redacted]	**Not In Place** ☐
B2-4.1.a Examine the software vendor's documentation to **verify** the software does not contain any functionality that is capable of facilitating the visual presentation of cleartext account data.	<Assessor Response>		
B2-4.1.b Perform static analysis to **verify** B2-4.1.a.	<Assessor Response>		
B2-4.1.c Perform dynamic analysis to **verify** B2-4.1.a/b.	<Assessor Response>		
B2-4.2 The software does not facilitate the audible presentation of *cleartext account data*.	**In Place** ☐	[redacted]	**Not In Place** ☐
B2-4.2.a Examine the software vendor's documentation to **verify** the software does not contain any functionality that is capable of facilitating the audible presentation of cleartext account data.	<Assessor Response>		
B2-4.2.b Perform static analysis to **verify** B2-4.2.a.	<Assessor Response>		
B2-4.2.c Perform dynamic analysis to **verify** B2-4.2.a/b.	<Assessor Response>		
B2-5 The software does not implement its own Open Protocols, and if needed, only uses the available Open Protocol functions of the POI device's firmware.	**In Place** ☐	[redacted]	**Not In Place** ☐
B2-5.a Examine the software vendor's documentation to **verify** the software does not implement its own functionality that is considered an "Open Protocol" as defined in the PCI PTS POI Standard. It is acceptable for the software to use the approved Open Protocols of the PTS POI device's firmware.	<Assessor Response>		
B2-5.b Perform static analysis to **verify** B2-5.a.	<Assessor Response>		
B2-5.c Perform dynamic analysis as deemed necessary to **verify** B2-5.a/b.	<Assessor Response>		

Security Objective B2: Approved POI Device Functionality			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
B2-6 The software facilitates the management or use of shared platform resources in a secure manner and in accordance with applicable POI device guidance.	**In Place** ☐	[redacted]	**Not In Place** ☐
B2-6.a Examine the software vendor's documentation to **verify** the software facilitates the management or use of shared platform resources in a secure manner and in accordance with applicable POI-device guidance.	<Assessor Response>		
B2-6.b Perform static analysis to **verify** B2-6.a.	<Assessor Response>		
B2-6.c Perform dynamic analysis as deemed necessary to **verify** B2-6.a/b.	<Assessor Response>		
B2-7 The software does not implement its own random-number generator (RNG) and, if needed, only uses the available RNG functions of the POI device's firmware.	**In Place** ☐	[redacted]	**Not In Place** ☐
B2-7.a Examine the software vendor's documentation to **verify** the software does not implement its own random-number generator (RNG) and, if needed, only uses the available RNG functions of the POI device's firmware.	<Assessor Response>		
B2-7.b Perform static analysis to **verify** B2-7.a.	<Assessor Response>		
B2-7.c Perform dynamic analysis as deemed necessary to **verify** B2-7.a/b.	<Assessor Response>		

Security Objective B3: Authentication

The software is designed to facilitate the required authentication by the POI device, per the PCI PTS POI Standard.

Notes: [No Notes]

Select the overall Finding for this Security Objective ➔	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective B3: Authentication			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
If the software is capable of facilitating loading additional files/content into the POI device: **B3-1** The software vendor includes additional details in the secure implementation guidance for the process to facilitate the authentication of the files/content by the POI device's firmware.	**In Place** ☐	**N/A** ☐	**Not In Place** ☐
ROV Instruction: If the assessment of security requirement B3-1 results in a finding of 'N'A, then security requirement B3-1 can be marked as 'N/A' (with the appropriate assessor response justifying the 'N'A finding documented in B3-1.a), provided the criteria for the use of 'N/A' as described herein is satisfied. The remaining B3-1 test requirements can then be left blank.			
B3-1.a Examine the software vendor's documentation to **verify** if the software is capable of facilitating loading additional files/content into the POI device.	<Assessor Response>		
B3-1.b Perform static analysis to **verify** B3-1.a.	<Assessor Response>		
B3-1.c Perform dynamic analysis to **verify** B3-1.a/b.	<Assessor Response>		
B3-1.d Examine the implementation documentation from Security Objective 11 and **verify** it includes secure implementation guidance for the processes to facilitate the authentication of the files/content by the POI device's firmware.	<Assessor Response>		
B3-2 The software vendor includes additional details in the secure implementation guidance for the process to facilitate the authentication of the software by the POI device's firmware.	**In Place** ☐	[redacted]	**Not In Place** ☐
B3-2.a Examine the implementation documentation from Security Objective 11 and **verify** it includes secure implementation guidance for the process to facilitate the authentication of the software itself by the POI device's firmware.	<Assessor Response>		

Module C – Publicly-accessible Software

Additional security requirements for software that contains, even if only in part, an interface that is accessible via public networks.

ROV Instruction: If Module C is being indicated as 'N/A' in tables 3.1 and 7.7, mark 'N/A' here with the appropriate assessor response justifying the 'N/A' finding, provided the criteria for the use of 'N/A' as described herein is satisfied. The remainder of this Module C section can then be left blank.

Module C is N/A ☐	<Assessor Response>

Security Objective C1: HTTP Headers

The software securely implements HTTP Headers.

Notes: There is significant potential risk related to HTTP headers. However, there are also dedicated informative resources available online to assist in utilizing and implementing HTTP headers in a [more] secure manner. As this information is subject to change based on known issues/threats, proactive research is prudent.

Select the overall Finding for this Security Objective ➔	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective C1: HTTP Headers			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
C1-1 HTTP security-related headers are used and configured in a manner to increase the security of the software.	**In Place** ☐	[redacted]	**Not In Place** ☐
C1-1.a Perform research as needed to determine the current recommendations/options of HTTP security-related headers and their particular configuration options. Leverage this information in the subsequent test requirements.	<Assessor Response>		
C1-1.b Examine vendor documentation to **verify** that security-related headers are used and configured in a manner to increase the security of the software. Where a particular header, or its more secure configuration, isn't being leveraged, assess C1-1.1.	<Assessor Response>		
C1-1.c Perform static analysis and/or dynamic analysis to **verify** the information and analysis from C1-1.b.	<Assessor Response>		
C1-1.1 Where a security-related header or its more secure configuration cannot be leveraged, vendor documentation explains the constraint.	**In Place** ☐	[redacted]	**Not In Place** ☐
C1-1.1.a Examine vendor documentation to **verify** that where a security-related header, or its more secure configuration, cannot be leveraged, vendor documentation explains the constraint.	<Assessor Response>		

Security Objective C1: HTTP Headers			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
C1-2 HTTP headers or their configuration options that are known to be vulnerable, or otherwise could negatively impact the security of the software, are avoided.	**In Place** ☐	[redacted]	**Not In Place** ☐
C1-2.a Perform research as needed to determine the current HTTP headers that have known security-impacting concerns. Leverage this information in the subsequent test requirements.	<Assessor Response>		
C1-2.b Examine vendor documentation to **verify** that HTTP headers that are known to be vulnerable or otherwise could negatively impact the security of the software are avoided. Where that isn't being accommodated, assess C1-2.1.	<Assessor Response>		
C1-2.c Perform static analysis and/or dynamic analysis to **verify** the information and analysis from C1-2.b.	<Assessor Response>		
C1-2.1 Where the use of a header or its configuration satisfying this condition cannot be avoided, vendor documentation explains the constraint.	**In Place** ☐	[redacted]	**Not In Place** ☐
C1-2.1.a Leverage information from C1-2 to **verify** that the use of HTTP headers that are known to be vulnerable or otherwise could negatively impact the security of the software cannot be avoided, vendor documentation explains the constraint.	<Assessor Response>		

Security Objective C2: Input Protection Mechanisms

The software securely implements and configures mechanisms to protect against input-related exploitation.

Notes: These requirements are intended as an extension to requirement 5-2.1 for input considerations that pose additional and unique risk for publicly-exposed web-based software implementations.

Select the overall Finding for this Security Objective ➔	In Place ☐	[redacted]	Not In Place ☐

Security Objective C2: Input Protection Mechanisms			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
C2-1 The software is designed to facilitate mitigating injection attacks.	**In Place** ☐	[redacted]	**Not In Place** ☐
C2-1.a Perform research as needed to determine the current recommendations/options regarding mitigating injection attacks as it pertains to the unique software implementation being assessed (e.g., the programming languages used, the API interface for inputs, etc.). Leverage this information in the subsequent test requirements.	<Assessor Response>		
C2-1.b Examine vendor documentation to **verify** that the software is designed to facilitate mitigating anomalous behavior from content being uploaded.	<Assessor Response>		
C2-1.c Perform static analysis to **verify** the information and analysis from C2-1.b.	<Assessor Response>		
C2-1.d Perform dynamic analysis to **verify** the information and analysis from C2-1.b/c. Testing should include attempts to exploit content uploads in a manner intended to compromise the software and/or underlying sensitive assets.	<Assessor Response>		
C2-2 The software is designed to implement secure deserialization.	**In Place** ☐	[redacted]	**Not In Place** ☐
C2-2.a Perform research as needed to determine the current recommendations/options regarding secure deserialization techniques that apply to the unique implementation of the software being assessed. Leverage this information in the subsequent test requirements.	<Assessor Response>		
C2-2.b Examine vendor documentation to **verify** that the software is designed to implement secure deserialization.	<Assessor Response>		
C2-2.c Perform static analysis to **verify** the information and analysis from C2-2.b.	<Assessor Response>		

Security Objective C2: Input Protection Mechanisms			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
C2-2.d Perform dynamic analysis to **verify** the information and analysis from C2-2.b/c. Testing should include attempts to exploit the deserialization functionality.	<Assessor Response>		
C2-3 The software is designed to securely implement and configure the use of parser and interpreter functionality.	**In Place** ☐	[redacted]	**Not In Place** ☐
C2-3.a Perform research as needed to determine the current recommendations/options regarding secure parser/interpreter implementations and configurations that apply to the unique implementation of the software being assessed. Leverage this information in the subsequent test requirements.	<Assessor Response>		
C2-3.b Examine vendor documentation to **verify** that the software is designed to securely implement and configure the use of parser and interpreter functionality.	<Assessor Response>		
C2-3.c Perform static analysis to **verify** the information and analysis from C2-3.c.	<Assessor Response>		
C2-3.d Perform dynamic analysis to **verify** the information and analysis from C2-3.b/c. Testing should include attempts to exploit the parser/interpreter functionality.	<Assessor Response>		
C2-4 The software is designed to facilitate mitigating *anomalous behavior* from content being uploaded.	**In Place** ☐	[redacted]	**Not In Place** ☐
C2-4.a Examine vendor documentation to **verify** that the software is designed to facilitate mitigating anomalous behavior from content being uploaded.	<Assessor Response>		
C2-4.b Perform static analysis to **verify** the information and analysis from C2-4.a.	<Assessor Response>		
C2-4.c Perform dynamic analysis to **verify** the information and analysis from C2-4.a/b. Testing should include attempts to exploit content uploads in a manner intended to compromise the software and/or underlying sensitive assets.	<Assessor Response>		
C2-5 The software is designed to facilitate mitigating resource starvation.	**In Place** ☐	[redacted]	**Not In Place** ☐
C2-5.a Examine vendor documentation to **verify** that the software is designed to facilitate mitigating resource starvation.	<Assessor Response>		

Security Objective C2: Input Protection Mechanisms		
Security Requirements and Test Requirements		**Assessor's Findings and Observations**
	C2-5.b Perform static analysis to **verify** the information and analysis from C2-5.a.	<Assessor Response>
	C2-5.c Perform dynamic analysis to **verify** the information and analysis from C2-5.a/b. Testing should include attempts to exploit the implemented resource starvation mitigation mechanisms.	<Assessor Response>

Security Objective C3: Session Management

The software securely implements and manages sessions.

Notes: [No Notes]

Select the overall Finding for this Security Objective ➔	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective C3: Session Management			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
C3-1 The software securely implements and manages sessions, which includes but is not limited to accounting for the following:	**In Place** ☐	[redacted]	**Not In Place** ☐
C3-1.a Perform research as needed to determine mechanisms and strategies regarding session management (including all contexts from C3-1.[x]) that apply to the unique implementation of the software being assessed. Leverage this information in the subsequent test requirements.	<Assessor Response>		
C3-1.b Examine vendor documentation to **verify** that the software is designed to securely implement and manage sessions.	<Assessor Response>		
C3-1.1 Session identifier token exchange mechanisms	**In Place** ☐	[redacted]	**Not In Place** ☐
C3-1.1.a Examine vendor documentation to **verify** that the software is designed to securely manage and implement session identifier token exchange mechanisms.	<Assessor Response>		
C3-1.1.b Perform static analysis to **verify** the information and analysis from C3-1.1.a.	<Assessor Response>		
C3-1.1.c Perform dynamic analysis to **verify** the information and analysis from C3-1.1.a/b.	<Assessor Response>		
C3-1.2 Session-identifier attributes	**In Place** ☐	[redacted]	**Not In Place** ☐
C3-1.2.a Examine vendor documentation to **verify** that the software is designed to securely manage and implement session-identifier attributes.	<Assessor Response>		
C3-1.2.b Perform static and/or dynamic analysis to **verify** the information and analysis from C3-1.2.a.	<Assessor Response>		
C3-1.3 Session timeouts	**In Place** ☐	[redacted]	**Not In Place** ☐
C3-1.3.a Examine vendor documentation to **verify** that the software is designed to securely manage and implement session timeouts.	<Assessor Response>		

Security Objective C3: Session Management	
Security Requirements and Test Requirements	**Assessor's Findings and Observations**
C3-1.3.b Perform static and/or dynamic analysis to **verify** the information and analysis from C3-1.3.a.	<Assessor Response>
C3-1.4 Session termination and re-instantiation	**In Place** ☐ **Not In Place** ☐
C3-1.4.a Examine vendor documentation to **verify** that the software is designed to securely manage and implement session termination and re-instantiation.	<Assessor Response>
C3-1.4.b Perform static and/or dynamic analysis to **verify** the information and analysis from C3-1.4.a.	<Assessor Response>
C3-1.5 Concurrent sessions	**In Place** ☐ **Not In Place** ☐
C3-1.5.a Examine vendor documentation to **verify** that the software is designed to securely manage and implement concurrent sessions.	<Assessor Response>
C3-1.5.b Perform static and/or dynamic analysis to **verify** the information and analysis from C3-1.5.a.	<Assessor Response>
C3-1.6 Use of secure session-management implementations	**In Place** ☐ **Not In Place** ☐
C3-1.6.a Examine vendor documentation to **verify** that the software is designed to utilize known-good session-management implementations.	<Assessor Response>
C3-1.6.b Perform static and/or dynamic analysis to **verify** the information and analysis from C3-1.6.a.	<Assessor Response>

Security Objective C4: User Authentication

The software securely implements authorized user authentication for access to sensitive assets.

Notes: [No Notes]

Select the overall Finding for this Security Objective ➔	**In Place** ☐	[redacted]	**Not In Place** ☐

Security Objective C4: User Authentication			
Security Requirements and Test Requirements	**Assessor's Findings and Observations**		
C4-1 The software authenticates authorized user access to *sensitive assets* via publicly-accessible interfaces.	**In Place** ☐	[redacted]	**Not In Place** ☐
C4-1.a Examine vendor documentation to **verify** that the software authenticates authorized user access to sensitive assets via publicly-accessible interfaces.	<Assessor Response>		
C4-1.b Perform static analysis to **verify** the information and analysis from C4-1.a.	<Assessor Response>		
C4-1.c Perform dynamic analysis to **verify** the information and analysis from C4-1.a/b. Testing should include ensuring non-authorized users cannot be authenticated or otherwise provided access. Leverage information and testing from 5-4.	<Assessor Response>		

Module D – Software Development Kits

Additional security requirements for software that is a software development kit (SDK).

ROV Instruction: A P2PE Application is not permitted to be provided as an SDK. Therefore, Module D is not an option for P2PE Application assessments. Leave this Module blank.

Module D is N/A ☒

Security Objective D1: SDK Integrity

The SDK is designed and delivered in a manner that facilitates its integrity.

Notes: [No Notes]

Select the overall Finding for this Security Objective ➔	**In Place** ☐	**Not In Place** ☐

Security Objective D1: SDK Integrity		
Security Requirements and Test Requirements	**Assessor's Findings and Observations**	
D1-1 The SDK is designed to mitigate the tampering of its execution and the compromise of its sensitive assets.	**In Place** ☐	**Not In Place** ☐
D1-1.a Examine the software vendor's documentation to **verify** the SDK is designed to mitigate the tampering of its execution and the compromise of its sensitive assets.	<Assessor Response>	
D1-1.b Perform static analysis to **verify** D1-1.a	<Assessor Response>	
D1-1.c Perform dynamic analysis as deemed necessary to **verify** D1-1.a/b. Sample tests might be to reconstruct the code or functionality in an effort to alter the SDK code, attempt to hook the SDK, or otherwise reverse engineer the SDK with the intent of attempting to compromise its execution and/or sensitive assets.	<Assessor Response>	
D1-2 The SDK is designed to facilitate the integrating application being able to validate the SDK's integrity and authenticity to the greatest extent possible.	**In Place** ☐	**Not In Place** ☐
D1-2.a Examine the software vendor's documentation to **verify** the SDK is designed to facilitate the integrating application being able to validate the SDK's integrity and authenticity to the greatest extent possible.	<Assessor Response>	
D1-2.b Perform static analysis to **verify** D1-2.a.	<Assessor Response>	
D1-2.c Perform dynamic analysis as deemed necessary to **verify** D1-2.a/b.	<Assessor Response>	