Payment Card Industry (PCI)

Payment Application Data Security Standard

Requirements and Security Assessment Procedures

Version 3.1

May 2015

Document Changes

Date	Version	Description	Pages
October 1, 2008	1.2	To align content with new PCI DSS v1.2 and to implement minor changes noted since original v1.1.	
July 2009	1.2.1	Under "Scope of PA-DSS," align content with the PA-DSS Program Guide, v1.2.1, to clarify applications to which PA-DSS applies.	v, vi
		Under Laboratory Requirement 6, corrected spelling of "OWASP."	30
		In the Attestation of Validation, Part 2a, update "Payment Application Functionality" to be consistent with the application types listed in the PA-DSS Program Guide, and clarify annual re-validation procedures in Part 3b.	32, 33
October 2010	2.0	Update and implement minor changes from v1.2.1 and align with new PCI DSS v2.0. For details, please see *PA-DSS – Summary of Changes from PA-DSS Version 1.2.1 to 2.0.*	
November 2013	3.0	Update from PA-DSS v2. For details of changes, please see *PA-DSS – Summary of Changes from PA-DSS Version 2.0 to 3.0.*	
May 2015	3.1	Update from PA-DSS v3.0. See *PA-DSS – Summary of Changes from PA-DSS Version 3.0 to 3.1* for details of changes.	

Table of Contents

Introduction

Purpose of This Document

The PCI Payment Application Data Security Standard (PA-DSS) Requirements and Security Assessment Procedures define security requirements and assessment procedures for software vendors of payment applications. This document is to be used by Payment Application Qualified Security Assessors (PA-QSAs) conducting payment application assessments to validate that a payment application complies with the PA-DSS. For details on how to document a PA-DSS assessment and create the Report on Validation (ROV), the PA-QSA should refer to the *PA-DSS ROV Reporting Template,* available on the PCI Security Standards Council (PCI SSC) website—www.pcisecuritystandards.org.

Additional resources including Attestations of Validation, Frequently Asked Questions (FAQs) and the *PCI DSS and PA-DSS Glossary of Terms, Abbreviations, and Acronyms* are available on the PCI Security Standards Council (PCI SSC) website—www.pcisecuritystandards.org.

Relationship between PCI DSS and PA-DSS

Use of a PA-DSS compliant application by itself does not make an entity PCI DSS compliant, since that application must be implemented into a PCI DSS compliant environment and according to the *PA-DSS Implementation Guide* provided by the payment application vendor (per PA-DSS Requirement 13). The PA-DSS requirements are derived from the *Payment Card Industry Data Security Standard (PCI DSS) Requirements and Security Assessment Procedures*, which details what is required to be PCI DSS compliant (and therefore what a payment application must support to facilitate a customer's PCI DSS compliance). The PCI DSS can be found at www.pcisecuritystandards.org.

All applications that store, process, or transmit cardholder data are in scope for an entity's PCI DSS assessment, including applications that have been validated to PA-DSS. The PCI DSS assessment should verify the PA-DSS payment application is properly configured and securely implemented per PCI DSS requirements. If the payment application has undergone any customization, a more in-depth review will be required during the PCI DSS assessment, as the application may no longer be representative of the version that was validated to PA-DSS

PCI DSS may not apply directly to payment application vendors unless the vendor stores, processes, or transmits cardholder data, or has access to their customers' cardholder data. However, since these payment applications are used by the application vendor's customers to store, process, and transmit cardholder data, and their customers are required to be PCI DSS compliant, payment applications should facilitate, and not prevent, their customers' PCI DSS compliance. Just a few of the ways insecure payment applications can prevent compliance include:

1. Storage of magnetic-stripe data and/or equivalent data on the chip in the customer's network after authorization;
2. Applications that require customers to disable other features required by the PCI Data Security Standard, like anti-virus software or firewalls, in order to get the payment application to work properly; and
3. Vendors' use of unsecured methods to connect to the application to provide support to the customer.

Secure payment applications, when implemented in a PCI DSS-compliant environment, will minimize the potential for security breaches leading to compromises of primary account number (PAN), full track data, card verification codes and values (CAV2, CID, CVC2, CVV2), PINs and PIN blocks, and the damaging fraud resulting from these breaches.

Integrators and Resellers

Application vendors may engage integrators and resellers to sell, install, and/or maintain payment applications on their behalf. Integrator/resellers have a role to play in ensuring the secure installation and operation of payment applications, as they often provide onsite services to the vendor's customers and assist with the installation of validated PA-DSS payment applications. Incorrect configuration, maintenance or support of an application may lead to the introduction of security vulnerabilities into the customer's cardholder data environment, which could then be exploited by attackers. Application vendors should educate their customers, integrators, and resellers on how to install and configure the payment applications in a PCI DSS compliant manner.

PCI Qualified Integrators and Resellers (QIRs) are trained by the Council in PCI DSS and PA-DSS in order to securely implement payment applications. For more information on the PCI QIR program, please see www.pcisecuritystandards.org.

PCI DSS Applicability Information

PCI DSS applies to all entities involved in payment card processing—including merchants, processors, acquirers, issuers, and service providers. PCI DSS also applies to ***all*** other entities that store, process, or transmit cardholder data and/or sensitive authentication data.

Cardholder data and sensitive authentication data are defined as follows:

Account Data	
Cardholder Data includes:	**Sensitive Authentication Data includes:**
▪ Primary Account Number (PAN) ▪ Cardholder Name ▪ Expiration Date ▪ Service Code	▪ Full track data (magnetic-stripe data or equivalent on a chip) ▪ CAV2/CVC2/CVV2/CID ▪ PINs/PIN blocks

The primary account number (PAN) is the defining factor for cardholder data. If cardholder name, service code, and/or expiration date are stored, processed, or transmitted with the PAN, or are otherwise present in the cardholder data environment, they must be protected in accordance with all applicable PCI DSS requirements.

The table on the following page illustrates commonly used elements of cardholder data and sensitive authentication data, whether storage of that data is permitted or prohibited, and whether this data needs to be protected. This table is not meant to be exhaustive, but is presented to illustrate the different type of requirements that apply to each data element.

		Data Element	Storage Permitted	Render Stored Data Unreadable per PA-DSS Requirement 2.3
Account Data	**Cardholder Data**	Primary Account Number (PAN)	Yes	Yes
		Cardholder Name	Yes	No
		Service Code	Yes	No
		Expiration Date	Yes	No
	Sensitive Authentication Data[1]	Full Track Data [2]	No	Cannot store per PA-DSS Requirement 1.1
		CAV2/CVC2/CVV2/CID[3]	No	Cannot store per PA-DSS Requirement 1.1
		PIN/PIN Block[4]	No	Cannot store per PA-DSS Requirement 1.1

PA-DSS Requirements 2.2 and 2.3 apply only to PAN. If PAN is stored with other elements of cardholder data, only the PAN must be rendered unreadable according to PA-DSS Requirement 2.3.

Sensitive authentication data must not be stored after authorization, even if encrypted. This applies even where there is no PAN in the environment.

1 *Sensitive authentication data must not be stored after authorization (even if encrypted).*

2 *Full track data from the magnetic stripe, equivalent data on the chip, or elsewhere*

3 *The three- or four-digit value printed on the front or back of a payment card*

4 *Personal identification number entered by cardholder during a card-present transaction, and/or encrypted PIN block present within the transaction message*

Scope of PA-DSS

The PA-DSS applies to software vendors and others who develop payment applications that store, process, or transmit cardholder data and/or sensitive authentication data. For information related to eligibility of different types of applications, please see the *PA-DSS Program Guide*.

The scope of the PA-DSS assessment should include the following:

- Coverage of all payment application functionality, including but not limited to:
    1) End-to-end payment functions (authorization and settlement),
    2) Input and output,
    3) Error conditions,
    4) Interfaces and connections to other files, systems, and/or payment applications or application components,
    5) All cardholder data flows,
    6) Encryption mechanisms, and
    7) Authentication mechanisms.
- Coverage of guidance the payment application vendor is expected to provide to customers and integrators/resellers (see *PA-DSS Implementation Guide* later in this document) to ensure:
    1) Customer knows how to implement the payment application in a PCI DSS-compliant manner and
    2) Customer is clearly told that certain payment application and environment settings may prohibit their PCI DSS compliance.

  Note that the payment application vendor may be expected to provide such guidance even when the specific setting:
    1) Cannot be controlled by the payment application vendor once the application is installed by the customer, or
    2) Is the responsibility of the customer, not the payment application vendor.
- Coverage of all selected platforms for the reviewed payment application version (included platforms should be specified)
- Coverage of tools used by or within the payment application to access and/or view cardholder data (reporting tools, logging tools, etc.)
- Coverage of all payment application related software components, including third-party software requirements and dependencies
- Coverage of any other types of payment applications necessary for a full implementation
- Coverage of vendor's versioning methodology

PA-DSS Applicability to Payment Applications on Hardware Terminals

This section provides guidance for vendors who wish to gain PA-DSS validation for resident payment applications on hardware terminals (also known as standalone or dedicated payment terminals).

There are two ways for a resident payment application on a hardware terminal to achieve PA-DSS validation:

1. The resident payment application directly meets all PA-DSS requirements and is validated according to standard PA-DSS procedures.
2. The resident payment application does not meet all PA-DSS requirements, but the hardware that the application is resident on is listed on the PCI SSC's Approved PIN Transaction Security (PTS) Devices List as a current PCI PTS approved Point of Interaction (POI) device. In this scenario, it may be possible for the application to satisfy PA-DSS requirements through a combination of the PA-DSS and PTS validated controls.

The remainder of this section applies only to payment applications that are resident on a validated PCI PTS approved POI device.

If one or more PA-DSS requirements cannot be met by the payment application directly, they may be satisfied indirectly by controls tested as part of the PCI PTS validation. For a hardware device to be considered for inclusion in a PA-DSS review, the hardware device MUST be validated as a PCI PTS approved POI device and be listed on the PCI SSC's Approved PTS Devices List. The PTS validated POI device, which provides a trusted computing environment, will become a "***required dependency***" for the payment application, and the combination of application and hardware will be listed together on the PA-DSS List of Validation Payment Applications.

When conducting the PA-DSS assessment, the PA-QSA must fully test the payment application with its dependent hardware against all PA-DSS requirements. If the PA-QSA determines that one or more PA-DSS requirements cannot be met by the resident payment application, but they are met by controls validated under PCI PTS, the PA-QSA must:

1. Clearly document which requirements are met as stated per PA-DSS (as usual);
2. Clearly document which requirement was met via PCI PTS in the "In Place" box for that requirement;
3. Include a thorough explanation as to why the payment application could not meet the PA-DSS requirement;
4. Document the procedures that were conducted to determine how that requirement was fully met through a PCI PTS validated control;
5. List the PCI PTS validated hardware terminal as a required dependency in the Executive Summary of the Report on Validation.

Once the PA-QSA's validation of the payment application is complete and is subsequently accepted by the PCI SSC, the PTS validated hardware device will be listed as a dependency for the payment application on the PA-DSS List of Validated Applications.

Resident payment applications on hardware terminals that are validated through a combination of PA-DSS and PCI PTS controls must meet the following criteria:

1. Be provided together to the customer (both hardware terminal and application), OR, if provided separately, the application vendor and/or the integrator/reseller must package the application for distribution such that it will operate only on the hardware terminal it has been validated to run on.
2. Enabled by default to support a customer's PCI DSS compliance.
3. Include ongoing support and updates to maintain PCI DSS compliance.
4. If the application is separately sold, distributed, or licensed to customers, the vendor must provide details of the dependent hardware required for use with the application, in accordance with its PA-DSS validation listing.

PA-DSS Implementation Guide

Validated payment applications must be capable of being implemented in a PCI DSS-compliant manner. Software vendors are required to provide a *PA-DSS Implementation Guide* to instruct their customers and integrators/resellers on secure product implementation, to document the secure configuration specifics mentioned throughout this document, and to clearly delineate vendor, integrator/reseller, and customer responsibilities for meeting PCI DSS requirements. It should detail how the customer and/or integrator/reseller should enable security settings within the customer's network. For example, the *PA-DSS Implementation Guide* should cover responsibilities and basic features of PCI DSS password security even if this is not controlled by the payment application, so that the customer or integrator/reseller understands how to implement secure passwords for PCI DSS compliance.

The *PA-DSS Implementation Guide* must provide details on how to configure the payment application to meet the requirement(s) and not simply restate the requirements from the PCI DSS or PA-DSS. During an assessment, the PA-QSA must verify that that the instructions are accurate and effective. The PA-QSA must also verify that the *PA-DSS Implementation Guide* is distributed to customers and integrators/resellers.

Payment applications, when implemented according to the *PA-DSS Implementation Guide*, and when implemented into a PCI DSS-compliant environment, should facilitate and support customers' PCI DSS compliance.

Refer to *Appendix A: Summary of Contents for the PA-DSS Implementation Guide* for a comparison of responsibilities for implementing the controls specified in the *PA-DSS Implementation Guide*.

Payment Application Qualified Security Assessor (PA-QSA) Requirements

Only Payment Application Qualified Security Assessors (PA-QSAs) employed by Payment Application Qualified Security Assessor (PA-QSA) Companies are allowed to perform PA-DSS assessments. Please see the Payment Application QSA list at www.pcisecuritystandards.org for a list of companies qualified to perform PA-DSS assessments.

- The PA-QSA must utilize the testing procedures documented in this Payment Application Data Security Standard document.
- The PA-QSA must have access to a laboratory where the validation process is to occur.

Testing Laboratory

- Testing laboratories can exist in either of two locations: onsite at the PA-QSA location, or onsite at the software vendor's location.
- The testing laboratory should be able to simulate real-world use of the payment application.
- The PA-QSA must validate the clean installation of the lab environment to ensure the environment truly simulates a real world situation and that the vendor has not modified or tampered with the environment in any way.
- Please refer to *Appendix B: Confirmation of Testing Laboratory Configuration Specific to PA-DSS Assessment* in this document for detailed requirements for the laboratory and related laboratory processes.
- PA-QSA must complete and submit *Appendix B,* completed for the specific laboratory used for the payment application under review, as part of the completed PA-DSS Report on Validation (ROV).

Instructions and Content for Report on Validation

Instructions and content for the PA-DSS Report on Validation (ROV) are now provided in the *PA-DSS ROV Reporting Template.* The *PA-DSS ROV Reporting Template* must be used for creating the Report on Validation. Only compliant payment application ROVs should be submitted to PCI SSC. For details about the ROV submission process, refer to the *PA-DSS Program Guide*.

PA-DSS Completion Steps

This document contains the Requirements and Security Assessment Procedures table, as well as *Appendix B: Testing Laboratory Configuration for PA-DSS Assessments*. The Requirements and Security Assessment Procedures detail the procedures that must be performed by the PA-QSA.

The PA-QSA must perform the following steps:

1. Confirm the scope of the PA-DSS assessment.
2. Perform the PA-DSS assessment.
3. Complete the Report on Validation (ROV) using the *PA-DSS ROV Reporting Template*, including confirmation of the testing laboratory configuration used for the PA-DSS assessment.
4. Complete and sign an Attestation of Validation (both PA-QSA and software vendor). The Attestation of Validation is available on the PCI SSC website (www.pcisecuritystandards.org).
5. After completion, submit all of the above documents and the *PA-DSS Implementation Guide* to PCI SSC, according to the *PA-DSS Program Guide.*

Note:

PA-DSS submissions should not be made unless all PA-DSS requirements have been validated as being in place.

PA-DSS Program Guide

Please refer to the *PA-DSS Program Guide* for information about PA-DSS program management, including the following topics:

- Applicability of PA-DSS to different types of applications
- PA-DSS report submission and acceptance processes
- Annual renewal process for payment applications included on the List of Validated Payment Applications
- Notification responsibilities in the event a listed payment application is determined to be at fault in a compromise.

PCI SSC reserves the right to require revalidation due to significant changes to the Payment Application Data Security Standard and/or due to specifically identified vulnerabilities in a listed payment application.

PA-DSS Requirements and Security Assessment Procedures

The following defines the table column headings for PA-DSS Requirements and Security Assessment Procedures:

- **PA-DSS Requirements** – This column defines the security requirements for payment applications to be validated against
- **Testing Procedures** – This column defines the testing processes to be followed by the PA-QSA to validate that PA-DSS requirements have been met
- **Guidance** – This column describes the intent or security objective behind each PA-DSS requirement and is intended to assist understanding of the requirements. The guidance in this column does not replace or extend the PA-DSS requirements and testing procedures.

Note:

PA-DSS requirements must not be considered in place if any controls are not yet implemented or are scheduled to be completed at a future date.

Requirement 1: Do not retain full track data, card verification code or value (CAV2, CID, CVC2, CVV2), or PIN block data

PA-DSS Requirements	Testing Procedures	Guidance
1.1 Do not store sensitive authentication data after authorization (even if encrypted). If sensitive authentication data is received, render all data unrecoverable upon completion of the authorization process. Sensitive authentication data includes the data as cited in the following Requirements 1.1.1 through 1.1.3. ***Aligns with PCI DSS Requirement 3.2***	**1.1.a** If this payment application stores sensitive authentication data, verify that the application is intended only for issuers and/or companies that support issuing services.	Sensitive authentication data consists of full track data, card validation code or value, and PIN data. Storage of sensitive authentication data after authorization is prohibited. This data is very valuable to malicious individuals as it allows them to generate counterfeit payment cards and create fraudulent transactions. Entities that issue payment cards or that perform or support issuing services will often create and control sensitive authentication data as part of the issuing function. It is permissible for issuers and companies that support issuing services to store sensitive authentication data if there is a business justification and the data is stored securely. For non-issuing entities, retaining sensitive authentication data post-authorization is not permitted, and the application is required to have a mechanism for securely deleting the data so it is unrecoverable.
	1.1.b For all other payment applications, if sensitive authentication data (see 1.1.1 – 1.1.3 below) is stored prior to authorization, obtain and review methodology for securely deleting the data to verify that the data is unrecoverable.	

PA-DSS Requirements	Testing Procedures	Guidance
1.1.1 After authorization, do not store the full contents of any track from the magnetic stripe (located on the back of a card, equivalent data contained on a chip, or elsewhere). This data is alternatively called full track, track, track 1, track 2, and magnetic-stripe data. ***Note:*** *In the normal course of business, the following data elements from the magnetic stripe may need to be retained:* • *The accountholder's name,* • *Primary account number (PAN),* • *Expiration date, and* • *Service code* *To minimize risk, store only those data elements needed for business.* ***Aligns with PCI DSS Requirement 3.2.1***	**1.1.1** Install the payment application and perform numerous test transactions that simulate all functions of the payment application, including generation of error conditions and log entries. Use forensic tools and/or methods (commercial tools, scripts, etc.)[5] to examine all output created by the payment application and verify that the full contents of any track from the magnetic stripe on the back of the card or equivalent data on a chip are not stored after authorization. Include at least the following types of files (as well as any other output generated by the payment application): • Incoming transaction data • All logs (for example, transaction, history, debugging, error) • History files • Trace files • Non-volatile memory, including non-volatile cache • Database schemas • Database contents.	If full track data is stored, malicious individuals who obtain that data can use it to reproduce payment cards and complete fraudulent transactions.

[5] *Forensic tool or method: A tool or method for uncovering, analyzing and presenting forensic data, which provides a robust way to authenticate, search, and recover computer evidence rapidly and thoroughly. In the case of forensic tools or methods used by PA-QSAs, these tools or methods should accurately locate any sensitive authentication data written by the payment application. These tools may be commercial, open-source, or developed in-house by the PA-QSA.*

PA-DSS Requirements	Testing Procedures	Guidance
1.1.2 After authorization, do not store the card verification value or code (three-digit or four-digit number printed on the front or back of a payment card used to verify card-not-present transactions). ***Aligns with PCI DSS Requirement 3.2.2***	**1.1.2** Install the payment application and perform numerous test transactions that simulate all functions of the payment application, including generation of error conditions and log entries. Use forensic tools and/or methods (commercial tools, scripts, etc.) to examine all output created by the payment application and verify that the three-digit or four-digit card verification code printed on the front of the card or the signature panel (CVV2, CVC2, CID, CAV2 data) is not stored after authorization. Include at least the following types of files (as well as any other output generated by the payment application): • Incoming transaction data • All logs (for example, transaction, history, debugging, error) • History files • Trace files • Non-volatile memory, including non-volatile cache • Database schemas • Database contents.	The purpose of the card validation code is to protect "card-not-present" transactions—Internet or mail order/telephone order (MO/TO) transactions—where the consumer and the card are not present. If this data is stolen, malicious individuals can execute fraudulent Internet and MO/TO transactions.
1.1.3 After authorization, do not store the personal identification number (PIN) or the encrypted PIN block. ***Aligns with PCI DSS Requirement 3.2.3***	**1.1.3** Install the payment application and perform numerous test transactions that simulate all functions of the payment application, including generation of error conditions and log entries. Use forensic tools and/or methods (commercial tools, scripts, etc.) to examine all output created by the payment application, and verify that PINs and encrypted PIN blocks are not stored after authorization. Include at least the following types of files (as well as any other output generated by the payment application). • Incoming transaction data • All logs (for example, transaction, history, debugging, error) • History files • Trace files • Non-volatile memory, including non-volatile cache • Database schemas • Database contents.	These values should be known only to the card owner or bank that issued the card. If this data is stolen, malicious individuals can execute fraudulent PIN-based debit transactions (for example, ATM withdrawals).

PA-DSS Requirements	Testing Procedures	Guidance
1.1.4 Securely delete any track data (from the magnetic stripe or equivalent data contained on a chip), card verification values or codes, and PINs or PIN block data stored by previous versions of the payment application, in accordance with industry-accepted standards for secure deletion, as defined, for example by the list of approved products maintained by the National Security Agency, or by other State or National standards or regulations. ***Note:*** *This requirement applies only if previous versions of the payment application stored sensitive authentication data.* ***Aligns with PCI DSS Requirement 3.2***	**1.1.4.a** Review the *PA-DSS Implementation Guide* prepared by the vendor and verify the documentation includes the following instructions for customers and integrators/resellers: • Historical data must be removed (track data, card verification codes, PINs, or PIN blocks stored by previous versions of the payment application). • How to remove historical data. • That such removal is absolutely necessary for PCI DSS compliance.	All of these elements of sensitive authentication data are not permitted to be stored post-authorization. If older versions of payment applications stored this information, the payment application vendor is required to provide instructions in the *PA-DSS Implementation Guide* as well as a secure wipe tool or procedure. If not securely deleted, this data could remain hidden on customer systems, and malicious individuals who obtain access to this information could use it to produce counterfeit payment cards, and/or to perform fraudulent transactions.
	1.1.4.b Examine payment application software files and configuration documentation to verify the vendor provides a secure wipe tool or procedure to remove the data.	
	1.1.4.c Verify, through the use of forensic tools and/or methods, that the secure wipe tool or procedure provided by vendor securely removes the data, in accordance with industry-accepted standards for secure deletion of data.	

PA-DSS Requirements	Testing Procedures	Guidance
1.1.5 Do not store sensitive authentication data on vendor systems. If any sensitive authentication data (pre-authorization data) must be used for debugging or troubleshooting purposes, ensure the following: • Sensitive authentication data is collected only when needed to solve a specific problem. • Such data is stored in a specific, known location with limited access. • The minimum amount of data is collected as needed to solve a specific problem. • Sensitive authentication data is encrypted with strong cryptography while stored. • Data is securely deleted immediately after use, including from: - Log files - Debugging files - Other data sources received from customers. ***Aligns with PCI DSS Requirement 3.2.***	**1.1.5.a** Examine the *software vendor's* procedures for troubleshooting customers' problems and verify the procedures include: • Collection of sensitive authentication data only when needed to solve a specific problem. • Storage of such data in a specific, known location with limited access. • Collection of only a limited amount of data needed to solve a specific problem. • Encryption of sensitive authentication data while stored. • Secure deletion of such data immediately after use. **1.1.5.b** Select a sample of recent troubleshooting requests from customers, and verify each event followed the procedure examined at 1.1.5.a. **1.1.5.c** Review the *PA-DSS Implementation Guide* prepared by the vendor and verify the documentation includes the following instructions for customers and integrators/resellers: • Collect sensitive authentication only when needed to solve a specific problem. • Store such data only in specific, known locations with limited access. • Collect only the limited amount of data needed to solve a specific problem. • Encrypt sensitive authentication data while stored. • Securely delete such data immediately after use.	If the vendor provides services to their customers that could result in the collection of sensitive authentication data (for example, for troubleshooting or debugging purposes), the vendor must minimize the collection of data, and ensure it is secured and securely deleted as soon as it is no longer needed. If troubleshooting a problem requires the application to be temporarily configured to capture sensitive authentication data (SAD), the application should be returned to its usual secure configuration (that is, to disable the collection of SAD) immediately upon completion of the necessary data capture. Once it is no longer needed, the SAD should be deleted in accordance with industry-accepted standards (for example, using a secure wipe program that ensures that the data is can never be retrieved).

Requirement 2: Protect stored cardholder data

PA-DSS Requirements	Testing Procedures	Guidance
2.1 Software vendor must provide guidance to customers regarding secure deletion of cardholder data after expiration of customer-defined retention period. ***Aligns with PCI DSS Requirement 3.1***	**2.1** Review the *PA-DSS Implementation Guide* prepared by the vendor and verify the documentation includes the following guidance for customers and integrators/resellers: • Cardholder data exceeding the customer-defined retention period must be securely deleted. • A list of all locations where the payment application stores cardholder data (so that customer knows the locations of data that needs to be deleted). • Instructions that customers need to securely delete cardholder data when no longer required for legal, regulatory, or business purposes. • Instructions on how to securely delete cardholder data stored by the payment application, including data stored on underlying software or systems (such as OS, databases, etc.) • Instructions for configuring the underlying software or systems (such as OS, databases, etc.) to prevent inadvertent capture or retention of cardholder data—for example, system backup or restore points.	To support PCI DSS Requirement 3.1, the vendor must provide details of all locations where the payment application may store cardholder data, including in any underlying software or systems (such as OS, databases, etc.), as well as instructions for securely deleting the data from these locations once the data has exceeded the customer's defined retention period. Customers and integrators/resellers must also be provided with configuration details for the underlying systems and software that the application runs on, to ensure these underlying systems are not capturing cardholder data without the customer's knowledge. The customer needs to know how the underlying systems could be capturing data from the application so they can either prevent it from being captured or ensure the data is properly protected.
2.2 Mask PAN when displayed (the first six and last four digits are the maximum number of digits to be displayed), such that only personnel with a legitimate business need can see the full PAN. ***Note:*** *This requirement does not supersede stricter requirements in place for displays of cardholder data—for example, legal or payment card brand requirements for point-of-sale (POS) receipts.* ***Aligns with PCI DSS Requirement 3.3***	**2.2.a** Review the *PA-DSS Implementation Guide* prepared by the vendor to verify the documentation includes the following guidance for customers and integrators/resellers: • Details of all instances where PAN is displayed, including but not limited to POS devices, screens, logs, and receipts. • Confirmation that the payment application masks PAN by default on all displays. • Instructions for how to configure the payment application such that only personnel with a legitimate business need can see the full PAN.	The display of full PAN on items such as computer screens, payment card receipts, faxes, or paper reports can result in this data being obtained by unauthorized individuals and used fraudulently. This requirement relates to protection of PAN displayed on screens, paper receipts, printouts, etc., and is not to be confused with PA-DSS Requirement 2.3 for protection of PAN when stored in files, databases, etc.

PA-DSS Requirements	Testing Procedures	Guidance
	2.2.b Install the payment application and examine all displays of PAN data, including but not limited to POS devices, screens, logs, and receipts. For each instance where PAN is displayed, verify that PAN is masked when displayed.	
	2.2.c Configure the payment application according to the *PA-DSS Implementation Guide* to allow only personnel with a legitimate business need to see the full PAN, For each instance where PAN is displayed, examine application configurations and displays of PAN to verify that instructions for masking PAN are accurate, and that only personnel with a legitimate business need can see the full PAN.	
2.3 Render PAN unreadable anywhere it is stored (including data on portable digital media, backup media, and in logs) by using any of the following approaches: • One-way hashes based on strong cryptography (hash must be of the entire PAN) • Truncation (hashing cannot be used to replace the truncated segment of PAN) • Index tokens and pads (pads must be securely stored) • Strong cryptography with associated key-management processes and procedures. *(Continued on next page)*	**2.3.a** Review the *PA-DSS Implementation Guide* prepared by the vendor to verify the documentation includes the following guidance for customers and integrators/resellers: • Details of any configurable options for each method used by the application to render cardholder data unreadable, and instructions on how to configure each method for all locations where cardholder data is stored by the payment application (per PA-DSS Requirement 2.1). • A list of all instances where cardholder data may be output for the customer to store outside of the payment application, and instructions that the customer is responsible for rendering PAN unreadable in all such instances.	Lack of protection of PANs can allow malicious individuals to view or download this data. One-way hash functions based on strong cryptography can be used to render cardholder data unreadable. Hash functions are appropriate when there is no need to retrieve the original number (one-way hashes are irreversible). The intent of truncation is that only a portion (not to exceed the first six and last four digits) of the PAN is stored. An index token is a cryptographic token that replaces the PAN based on a given index for an unpredictable value. A one-time pad is a system in which a randomly generated private key is used only once to encrypt a message that is then decrypted using a matching one-time pad and key. *(Continued on next page)*
	2.3.b Examine the method used to protect the PAN, including the encryption algorithms (if applicable). Verify that the PAN is rendered unreadable using any of the following methods: • One-way hashes based on strong cryptography • Truncation • Index tokens and pads, with the pads being securely stored • Strong cryptography, with associated key-management processes and procedures.	

PA-DSS Requirements	Testing Procedures	Guidance
Notes: • It is a relatively trivial effort for a malicious individual to reconstruct original PAN data if they have access to both the truncated and hashed version of a PAN. Where hashed and truncated versions of the same PAN are generated by a payment application, additional controls must be in place to ensure that hashed and truncated versions cannot be correlated to reconstruct the original PAN. • The PAN must be rendered unreadable anywhere it is stored, even outside the payment application (for example, log files output by the application for storage in the customer environment). ***Aligns with PCI DSS Requirement 3.4***	**2.3.c** If the application creates both hashed and truncated versions of the same PAN, examine methods for creating the hashed and truncated versions to verify that the hashed and truncated versions cannot be correlated to reconstruct the original PAN.	
	2.3.d Examine several tables or files from data repositories created or generated by the application to verify the PAN is rendered unreadable.	The intent of strong cryptography (as defined in the *PCI DSS and PA-DSS Glossary of Terms, Abbreviations, and Acronyms*) is that the encryption be based on an industry-tested and accepted algorithm (not a proprietary or "home-grown" algorithm), with strong cryptographic keys.
	2.3.e If the application creates or generates files for use outside the application (for example, files generated for export or backup), including for storage on removable media, examine a sample of generated files, including those generated on removable media (for example, back-up tapes), to confirm that the PAN is rendered unreadable.	
	2.3.f Examine a sample of audit logs created or generated by the application to confirm that the PAN is rendered unreadable or removed from the logs.	
	2.3.g If the software vendor stores the PAN for any reason (for example, because log files, debugging files, and other data sources are received from customers for debugging or troubleshooting purposes), verify that the PAN is rendered unreadable in accordance with Requirements 2.3.b through 2.3.f, above.	
2.4 Payment application must protect keys used to secure cardholder data against disclosure and misuse. ***Note:*** *This requirement applies to keys used to encrypt stored cardholder data, as well as to key-encrypting keys used to protect data-encrypting keys. Such key-encrypting keys must be at least as strong as the data-encrypting key.* ***Aligns with PCI DSS Requirement 3.5***	**2.4.a** Examine product documentation and interview responsible personnel to verify that controls are in place that restrict access to cryptographic keys used by the application.	Cryptographic keys must be strongly protected because those who obtain access will be able to decrypt data. The requirement for payment applications to protect keys from disclosure and misuse applies to both data-encrypting keys and key-encrypting keys. It is not intended that the key-encrypting keys be encrypted, however they are to be protected against disclosure and misuse as defined in Requirement 2.4
	2.4.b Examine system configuration files to verify that: • Keys are stored in encrypted format • Key-encrypting keys are stored separately from data-encrypting keys • Key-encrypting keys are at least as strong as the data encrypting keys they protect.	

PA-DSS Requirements	Testing Procedures	Guidance
	2.4.c Review the *PA-DSS Implementation Guide* prepared by the vendor and verify that customers and integrators/resellers are instructed to: • Restrict access to keys to the fewest number of custodians necessary. • Store keys securely in the fewest possible locations and forms.	There should be very few who have access to cryptographic keys, usually only those who have key custodian responsibilities.
2.5 Payment application must implement key-management processes and procedures for cryptographic keys used for encryption of cardholder data, including at least the following: ***Aligns with PCI DSS Requirement 3.6***	**2.5** Review the *PA-DSS Implementation Guide* prepared by the vendor and verify the documentation includes the following instructions for customers and integrators/resellers: • How to securely generate, distribute, protect, change, store, and retire/replace cryptographic keys, where customers or integrators/resellers are involved in these key-management activities. • A sample Key Custodian Form for key custodians to acknowledge that they understand and accept their key-custodian responsibilities.	The manner in which cryptographic keys are managed is a critical part of the continued security of the payment application. A good key-management process, whether it is manual or automated as part of the encryption product, is based on industry standards and addresses all key elements at 2.5.1 through 2.5.7. Providing guidance to customers on how to securely transmit, store and update cryptographic keys can help prevent keys from being mismanaged or disclosed to unauthorized entities. This requirement applies to keys used to encrypt stored cardholder data, and any respective key-encrypting keys.
2.5.1 Generation of strong cryptographic keys	**2.5.1.a** Review the *PA-DSS Implementation Guide* and verify it includes instructions for customers and integrators/resellers on how to securely generate cryptographic keys.	The payment application must generate strong keys, as defined in the *PCI DSS and PA-DSS Glossary of Terms, Abbreviations, and Acronyms* under "Strong Cryptography."
	2.5.1.b Test the application, including the methods used to generate cryptographic keys, to verify that the instructions in the *PA-DSS Implementation Guide* result in the generation of strong cryptographic keys.	
2.5.2 Secure cryptographic key distribution	**2.5.2.a** Review the *PA-DSS Implementation Guide* and verify it includes instructions for customers and integrators/resellers on how to securely distribute cryptographic keys.	The payment application must distribute keys securely, meaning the keys are not distributed in the clear, and only via authorized processes.

PA-DSS Requirements	Testing Procedures	Guidance
	2.5.2.b Test the application, including the methods used to distribute cryptographic keys, to verify that the instructions in the *PA-DSS Implementation Guide* result in the secure distribution of cryptographic keys.	
2.5.3 Secure cryptographic key storage	**2.5.3.a** Review the *PA-DSS Implementation Guide* and verify it includes instructions for customers and integrators/resellers on how to securely store cryptographic keys.	The payment application must store keys securely (for example, by encrypting them with a key-encrypting key).
	2.5.3.b Test the application, including the methods used to store cryptographic keys, to verify that the instructions in the *PA-DSS Implementation Guide* result in the secure storage of cryptographic keys.	
2.5.4 Cryptographic key changes for keys that have reached the end of their cryptoperiod (for example, after a defined period of time has passed and/or after a certain amount of cipher-text has been produced by a given key), as defined by the associated application vendor or key owner, and based on industry best practices and guidelines (for example, *NIST Special Publication 800-57*.	**2.5.4.a** Review the *PA-DSS Implementation Guide* and verify it includes the following instructions for customers and integrators/resellers: • Defined cryptoperiod for each key type used by the application. • Procedures for enforcing key changes at the end of the defined cryptoperiod.	A cryptoperiod is the time span during which a particular cryptographic key can be used for its defined purpose. Considerations for defining the cryptoperiod include, but are not limited to, the strength of the underlying algorithm, size or length of the key, risk of key compromise, and the sensitivity of the data being encrypted. Periodic changing of encryption keys when the keys have reached the end of their cryptoperiod is imperative to minimize the risk of someone's obtaining the encryption keys, and using them to decrypt data.
	2.5.4.b Test the application, including the methods for changing cryptographic keys, to verify the instructions in the *PA-DSS Implementation Guide* result in key changes at the end of the defined cryptoperiod.	
2.5.5 Retirement or replacement of keys (for example: by archiving, destruction, and/or revocation as applicable) as deemed necessary when the integrity of the key has been weakened (for example, departure of an employee with knowledge of a clear-text key component, etc.) or keys are suspected of being compromised. ***Note:*** *If retired or replaced cryptographic keys need to be retained, these keys must be securely archived (for example, by using a key-encrypting key). Archived cryptographic keys should be used only for decryption/verification purposes.*	**2.5.5.a** Review the *PA-DSS Implementation Guide* and verify it includes the following for customers and integrators/resellers: • Instructions that keys must be retired or replaced when the integrity of the key has been weakened, or there is a known or suspected compromise of a key. • Procedures for retiring or replacing keys (for example: by archiving, destruction, and/or revocation as applicable). • Procedures for ensuring that retired or replaced cryptographic keys are not used for encryption operations.	Keys that are no longer used or needed, or keys that are known or suspect compromised, should be revoked and/or destroyed to ensure that the keys can no longer be used. If such keys need to be kept (for example, to support archived, encrypted data) they should be strongly protected. The payment application should provide for and facilitate a process to replace keys that are due for replacement or that are known to be, or suspected of being, compromised.

PA-DSS Requirements	Testing Procedures	Guidance
	2.5.5.b Test the application, including the methods for retiring or replacing cryptographic keys, to verify that the instructions in the *PA-DSS Implementation Guide* result the retirement or replacement of keys (for example: by archiving, destruction, and/or revocation as applicable).	
	2.5.5.c Test the application with the retired/replaced keys to verify that the instructions in the *PA-DSS Implementation Guide* ensure the application does not use retired or replaced keys for encryption operations.	
2.5.6 If the payment application supports manual clear-text cryptographic key-management operations, these operations must enforce split knowledge and dual control. ***Note:*** *Examples of manual key-management operations include, but are not limited to: key generation, transmission, loading, storage and destruction.*	**2.5.6.a** Review the *PA-DSS Implementation Guide* and verify it includes the following for customers and integrators/resellers: • Details of any manual clear-text cryptographic key-management operations supported by the application. • Instructions for enforcing split knowledge and dual control for all such operations.	Split knowledge and dual control of keys are used to eliminate the possibility of one person having access to the whole key This control is applicable for manual key-management operations. Split knowledge is a method in which two or more people separately have key components that individually convey no knowledge of the original cryptographic key; each person knows only their own key component, and the individual key components convey no knowledge of the original cryptographic key). Dual control requires two or more people to perform a function, and no single person can access or use the authentication materials of another.
	2.5.6.b Test the application, including all manual clear-text cryptographic key-management operations, to verify that the instructions in the *PA-DSS Implementation Guide* result in split knowledge and dual control of keys being required for all manual clear-text key-management procedures.	
2.5.7 Prevention of unauthorized substitution of cryptographic keys	**2.5.7.a** Review the *PA-DSS Implementation Guide* and verify it includes instructions for customers and integrators/resellers on how to prevent unauthorized substitution of cryptographic keys.	The payment application should define methods for users of the application to ensure only authorized key substitutions can be made. The application configuration should include not allowing for or accepting substitution of keys coming from unauthorized sources or unexpected processes.
	2.5.7.b Test the application, including all methods for substituting keys, to verify that the instructions in the *PA-DSS Implementation Guide* prevent unauthorized substitution of cryptographic keys.	

PA-DSS Requirements	Testing Procedures	Guidance
2.6 Provide a mechanism to render irretrievable any cryptographic key material or cryptogram stored by the payment application, in accordance with industry-accepted standards. These are cryptographic keys used to encrypt or verify cardholder data. ***Note:*** *This requirement applies only if the payment application uses, or previous versions of the payment application used, cryptographic key materials or cryptograms to encrypt cardholder data.* ***Aligns with PCI DSS Requirement 3.6***	**2.6.a** Review the *PA-DSS Implementation Guide* prepared by the vendor and verify the documentation includes the following instructions for customers and integrators/resellers: • Procedures detailing how to use the tool or procedure provided with the application to render cryptographic material irretrievable. • That cryptographic key material should be rendered irretrievable whenever keys are no longer used and in accordance with key-management requirements in PCI DSS. • Procedures for re-encrypting historic data with new keys, including procedures for maintaining security of clear-text data during the decryption/re-encryption process.	Vendors should provide a mechanism so their customers can securely delete old cryptographic material when the customer no longer needs it. Note that the deletion of old cryptographic material is at the customers' discretion. Cryptographic key materials and/or cryptograms may be rendered irretrievable through the use of tools or processes including but not limited to: • Secure deletion, as defined, for example, in the list of approved products maintained by the National Security Agency, or by other State or National standards or regulations. • The deletion of the key-encrypting key (KEK) provided that residual data-encrypting keys only exist in encrypted form under the deleted KEK.
	2.6.b Examine final application product to verify the vendor provides a tool and/or procedure with the application to render cryptographic material irretrievable.	
	2.6.c Test the application, including the methods provided for rendering cryptographic key material irretrievable. Verify, through use of forensic tools and/or methods, that the secure wipe tool or procedure provided by the vendor renders the cryptographic material irretrievable, in accordance with industry-accepted standards.	
	2.6.d Test the methods for re-encrypting historic data with new keys, to verify the instructions in the *PA-DSS Implementation Guide* result in successful re-encryption of historic data with new keys.	

Requirement 3: Provide secure authentication features

PA-DSS Requirements	Testing Procedures	Guidance
3.1 The payment application must support and enforce the use of unique user IDs and secure authentication for all administrative access and for all access to cardholder data. Secure authentication must be enforced to all accounts generated or managed by the application by the completion of installation and for subsequent changes after installation. The application must enforce 3.1.1 through 3.1.11 below: ***Note:*** *The term "subsequent changes" as used throughout Requirement 3 refers to any application changes that result in user accounts reverting to default settings, changes to existing account configurations, and changes that generate new accounts or recreate existing accounts.* ***Note:*** *These password controls are not intended to apply to personnel who only have access to one card number at a time to facilitate a single transaction. These controls are applicable for access by personnel with administrative capabilities, for access to systems with cardholder data, and for access controlled by the payment application.* *This requirement applies to the payment application and all associated tools used to view or access cardholder data.* ***Aligns with PCI DSS Requirements 8.1 and 8.2***	**3.1.a** Examine the *PA-DSS Implementation* Guide created by the vendor to verify that customers and integrators/resellers are: • Provided clear and unambiguous directions on how the payment application enforces strong authentication for all authentication credentials that the application generates or manages, by: - Enforcing secure changes to authentication credentials by the completion of installation per Requirements 3.1.1 through 3.1.11). - Enforcing secure changes for any subsequent changes (after installation) to authentication credentials per Requirements 3.1.1 through 3.1.11). • Advised that, to maintain PCI DSS compliance, any changes made to authentication configurations would need to be verified as providing authentication methods that are at least as rigorous as PCI DSS requirements. • Advised to assign secure authentication to all default accounts in the environment. • For any default accounts that won't be used, assign secure authentication and then disable or do not use the accounts. • Provided clear and unambiguous directions for all authentication credentials used by the payment application (but which are not generated or managed by the application), on how, by the completion of installation and for any changes after installation, to change authentication credentials and create strong authentication per Requirements 3.1.1 through 3.1.11 below, for all application level and user accounts with administrative access and for all accounts with access to cardholder data.	By ensuring each user is uniquely identified—instead of using one ID for several employees—an application supports the PCI DSS requirements to maintain individual responsibility for actions and an effective audit trail per employee. This will help speed issue resolution and containment when misuse or malicious intent occurs. Secure authentication, when used in addition to unique IDs, helps protect users' IDs from being compromised, since anyone attempting to compromise an account would need to know both the unique ID and the password (or other authentication used).

PA-DSS Requirements	Testing Procedures	Guidance
3.1.1 The payment application does not use (or require the use of) default administrative accounts for other necessary software (for example, the payment application must not use the database default administrative account). ***Aligns with PCI DSS Requirement 2.1***	**3.1.1** Install and configure the payment application in accordance with the *PA-DSS Implementation Guide,* including configuring any administrative accounts for all necessary software. Test the payment application to verify the payment application does not use (or require the use of) default administrative accounts for necessary software.	Default administrative accounts (and passwords) are public knowledge, and known to anyone who is familiar with the payment application or underlying system components. If default administrative accounts and passwords are used, an unauthorized individual may be able to gain access to the application and data simply by logging in with publicly known credentials.
3.1.2 The application must enforce the changing of all default application passwords for all accounts that are generated or managed by the application, by the completion of installation and for subsequent changes after installation. This applies to all accounts, including user accounts, application and service accounts, and accounts used by the vendor for support purposes. ***Note:*** *This requirement cannot be met through specifying a user process or via instructions in the PA-DSS Implementation Guide. At the completion of installation, and upon subsequent changes, the application must technically prevent any default or built-in accounts from being used until the default password has been changed.* ***Aligns with PCI DSS Requirement 2.1***	**3.1.2** For all accounts generated or managed by the application, test the application as follows: **3.1.2.a** Install the application in accordance with the *PA-DSS Implementation Guide*, examine account and password settings and attempt to use all default passwords to verify that the application enforces changes to any default payment application passwords by completion of the installation process. **3.1.2.b** Test all application functionality that results in user accounts reverting to default settings, changes to existing account configurations, generation of new accounts, and recreation of existing accounts. For all types of changes performed, examine account and password settings and attempt to use all default passwords to verify that the application enforces changes to all default passwords upon completion of the change.	If the application doesn't enforce changing of default passwords, the application could be left exposed to unauthorized access by anyone knowledgeable of the default settings.
3.1.3 The payment application assigns unique IDs for user accounts. ***Aligns with PCI DSS Requirements 8.1.1***	**3.1.3** For all accounts that are generated or managed by the application, test the application as follows: **3.1.3.a** Install the payment application in accordance with the *PA-DSS Implementation Guide* and attempt to create different application accounts with the same user ID to verify that the payment application only assigns unique user IDs by completion of the installation process.	When each user is assigned a unique user ID, their access to and activities within the payment application can be traced back to the individual who performed them.

PA-DSS Requirements	Testing Procedures	Guidance
	3.1.3.b Test all application functionality that results in user accounts reverting to default settings, changes to existing account configurations, generation of new accounts, and recreation of existing accounts. For all types of changes performed, examine account settings and test application functionality to verify that unique user IDs are assigned for all accounts upon completion of the change.	
3.1.4 The payment application employs at least one of the following methods to authenticate all users: ▪ Something you know, such as a password or passphrase ▪ Something you have, such as a token device or smart card ▪ Something you are, such as a biometric. ***Aligns with PCI DSS Requirements 8.2***	**3.1.4** For all accounts generated or managed by the application, test the application as follows:	These authentication methods, when used in addition to unique IDs, help protect users' IDs from being compromised, since the one attempting the compromise needs to know both the unique ID and the password (or other authentication used).
	3.1.4.a Install the payment application in accordance with the *PA-DSS Implementation Guide* and test authentication methods to verify that the application requires at least one of the defined authentication methods for all accounts by completion of the installation process.	
	3.1.4.b Test all application functionality that results in user accounts reverting to default settings, changes to existing account configurations, generation of new accounts, and recreation of existing accounts. For all types of changes performed, test authentication methods to verify that the application requires at least one of the defined authentication methods for all accounts, upon completion of the change.	

PA-DSS Requirements	Testing Procedures	Guidance
3.1.5 The payment application does **not** require or use any group, shared, or generic accounts and passwords. ***Aligns with PCI DSS Requirement 8.5***	**3.1.5** For all accounts generated or managed by the application, test the application as follows:	If multiple users share the same authentication credentials (for example, user account and password), it becomes impossible to assign accountability for, or to have effective logging of, an individual's actions, since a given action could have been performed by anyone that has knowledge of the authentication credentials.
	3.1.5.a Install the payment application in accordance with the *PA-DSS Implementation Guide*, examine account settings and test application functionality to verify that, by completion of the installation process, the application does not require or use any group, shared, or generic accounts and passwords.	
	3.1.5.b Test all application functionality that results in user accounts reverting to default settings, changes to existing account configurations, generation of new accounts, and recreation of existing accounts. For all types of changes performed, examine account settings and test application functionality to verify that the application does not rely on or use any group, shared, or generic accounts and passwords upon completion of the change.	
3.1.6 The payment application requires that passwords meet the following: • Require a minimum length of at least seven characters. • Contain both numeric and alphabetic characters. Alternatively, the passwords/phrase must have complexity and strength at least equivalent to the parameters specified above.	**3.1.6** For all accounts generated or managed by the application, test the application as follows:	Malicious individuals will often try to find accounts with weak or non-existent passwords in order to gain access to an application or system. If passwords are short or simple to guess, it is relatively easy for a malicious individual to find these weak accounts, and compromise an application or system under the guise of a valid user ID. *(Continued on next page)*
	3.1.6.a Install the payment application in accordance with the *PA-DSS Implementation Guide* and examine account settings to verify that by completion of the installation process, the application requires passwords to require a minimum of the following complexity and strength: • Be at least seven characters in length. • Contain both numeric and alphabetic characters.	

PA-DSS Requirements	Testing Procedures	Guidance
	3.1.6.b Test all application functionality that results in user accounts reverting to default settings, changes to existing account configurations, generation of new accounts, and recreation of existing accounts. For all types of changes performed, examine account settings and test application functionality to verify that, upon completion of the change, the application requires passwords to require a minimum of the following complexity and strength: • Be at least seven characters in length • Contain both numeric and alphabetic characters.	This requirement specifies passwords be a minimum of seven characters in length and that both numeric and alphabetic characters should be used. For cases where this minimum cannot be met due to technical limitations, entities can use "equivalent strength" to evaluate their alternative. NIST SP 800-63-1 defines "entropy" as "a measure of the difficulty of guessing or determining a password or key." This document and others that discuss "password entropy" can be referenced for more information on entropy value and equivalent password strength for passwords of different minimum formats.
	3.1.6.c If the application uses a different minimum character set and length for passwords, calculate the entropy of the passwords required by the application, and verify that it is at least equivalent to the parameters specified above (that is, at least as strong as 7 characters in length with numeric and alphabetic characters),	
3.1.7 The payment application requires changes to user passwords at least once every 90 days. ***Aligns with PCI DSS Requirement 8.2.4***	**3.1.7** For all accounts generated or managed by the application, test the application as follows:	Passwords/phrases that are valid for a long time without being changed provide malicious individuals with more time to work on breaking the password/phrase.
	3.1.7.a Install the payment application in accordance with the *PA-DSS Implementation Guide* and examine account settings to verify that the application requires user passwords to be changed at least once every 90 days by completion of the installation process.	
	3.1.7.b Test all application functionality that results in user accounts reverting to default settings, changes to existing account configurations, generation of new accounts, and recreation of existing accounts. For all types of changes performed, examine account settings and test application functionality to verify that the application requires user passwords to be changed at least once every 90 days upon completion of the change.	

PA-DSS Requirements	Testing Procedures	Guidance
3.1.8 The payment application keeps password history and requires that a new password is different than any of the last four passwords used. ***Aligns with PCI DSS Requirement 8.2.5***	**3.1.8** For all accounts generated or managed by the application, test the application as follows:	If password history isn't maintained, the effectiveness of changing passwords is reduced, as previous passwords can be reused over and over. Requiring that passwords cannot be reused for a period of time reduces the likelihood that passwords that have been guessed or brute-forced will be used in the future.
	3.1.8.a Install the payment application in accordance with the *PA-DSS Implementation Guide* and examine account settings to verify that, by completion of the installation process, the application keeps password history and requires that a new password is different than any of the last four passwords used.	
	3.1.8.b Test all application functionality that results in user accounts reverting to default settings, changes to existing account configurations, generation of new accounts, and recreation of existing accounts. For all types of changes performed, examine account settings and test application functionality to verify that the application keeps password history and requires that a new password is different than any of the last four passwords used, upon completion of the change.	
3.1.9 The payment application limits repeated access attempts by locking out the user account after not more than six logon attempts. ***Aligns with PCI DSS Requirement 8.1.6***	**3.1.9** For all accounts generated or managed by the application, test the application as follows:	Without account-lockout mechanisms in place, an attacker can continually attempt to guess a password through manual or automated tools (for example, password cracking), until they achieve success and gain access to a user's account.
	3.1.9.a Install the payment application in accordance with the *PA-DSS Implementation Guide* and examine account settings to verify that, by completion of the installation process, the application locks out user accounts after not more than six invalid logon attempts.	
	3.1.9.b Test all application functionality that results in user accounts reverting to default settings, changes to existing account configurations, generation of new accounts, and recreation of existing accounts. For all types of changes performed, examine account settings and test application functionality to verify that the application locks out user accounts after not more than six invalid logon attempts, upon completion of the change.	

PA-DSS Requirements	Testing Procedures	Guidance
3.1.10 The payment application sets the lockout duration to a minimum of 30 minutes or until an administrator enables the user ID. ***Aligns with PCI DSS Requirement 8.1.7***	**3.1.10** For all accounts generated or managed by the application, test the application as follows:	If an account is locked out due to someone continually trying to guess a password, controls to delay reactivation of these locked accounts stops the malicious individual from continually guessing the password (they will have to stop for a minimum of 30 minutes until the account is reactivated). Additionally, if reactivation must be requested, the administrator can validate that it is the actual account owner requesting reactivation.
	3.1.10.a Install the payment application in accordance with the *PA-DSS Implementation Guide* and examine account settings to verify that, by completion of the installation process, the application sets the lockout duration to a minimum of 30 minutes or until administrator enables the user ID.	
	3.1.10.b Test all application functionality that results in user accounts reverting to default settings, changes to existing account configurations, generation of new accounts, and recreation of existing accounts. For all types of changes performed, examine account settings and test application functionality to verify that the application sets the lockout duration to a minimum of 30 minutes or until administrator enables the user ID, upon completion of the change.	
3.1.11 If a payment application session has been idle for more than 15 minutes, the application requires the user to re-authenticate to re-activate the session. ***Aligns with PCI DSS Requirement 8.1.8***	**3.1.11** For all accounts generated or managed by the application, test the application as follows:	When users walk away from an open session with access to the payment application, that connection may be used by others in the user's absence, resulting in unauthorized account access and/or account misuse.
	3.1.11.a Install the payment application in accordance with the *PA-DSS Implementation Guide* and examine account settings to verify that, by completion of the installation process, the application sets a session idle time out to 15 minutes or less	
	3.1.11.b Test all application functionality that results in user accounts reverting to default settings, changes to existing account configurations, generation of new accounts, and recreation of existing accounts. For all types of changes performed, examine account settings and test application functionality to verify that the application sets a session idle time out to 15 minutes or less, upon completion of the change.	

PA-DSS Requirements	Testing Procedures	Guidance
3.2 Software vendor must provide guidance to customers that all access to PCs, servers, and databases with payment applications must require a unique user ID and secure authentication. ***Aligns with PCI DSS Requirements 8.1 and 8.2***	**3.2** Examine the *PA-DSS Implementation Guide* created by vendor to verify customers and integrators/resellers are instructed to control access, via unique user ID and PCI DSS-compliant secure authentication, to any PCs, servers, and databases with payment applications and cardholder data.	If the application is installed on or accessed by systems that do not enforce strong identification and authentication controls, the strong authentication provided by the application could be bypassed, resulting in insecure access.
3.3 Secure all payment application passwords (including passwords for user and application accounts) during transmission and storage. ***Aligns with PCI DSS Requirement 8.2.1***	**3.**3 Perform the following:	If payment application passwords are stored or transmitted across the network without encryption, a malicious individual can easily intercept the password using a "sniffer," or directly access the passwords in files where they are stored, and use this stolen data to gain unauthorized access. Concatenating a unique input variable to each password before the hashing algorithm is applied reduces the effectiveness of brute force attacks. Examples of strong one-way cryptographic algorithms suitable for hashing passwords include PBKDF2 and Bcrypt.
3.3.1 Use strong cryptography to render all payment application passwords unreadable during transmission.	**3.3.1.a** Examine vendor documentation and application configurations to verify that strong cryptography is used to render all passwords unreadable at all times during transmission.	
	3.3.1.b For all types of application passwords, examine transmissions of passwords (for example, by logging into the application from another system, and authenticating the application to other systems) to verify strong cryptography is used to render all passwords unreadable at all times during transmission.	
3.3.2 Use a strong, one-way cryptographic algorithm, based on approved standards to render all payment application passwords unreadable during storage. Each password must have a unique input variable that is concatenated with the password before the cryptographic algorithm is applied. ***Note:*** *The input variable does not need to be unpredictable or secret*	**3.3.2.a** Examine vendor documentation and application configurations to verify that**:** • Stored passwords are rendered unreadable using a strong, one-way cryptographic algorithm, based on approved standards. • A unique input variable is concatenated with each password before the cryptographic algorithm is applied.	
	3.3.2.b For all types of application passwords, identify all locations where the application may store passwords, including within the application itself, on underlying systems, log files, registry settings, etc. For all locations and types of passwords, examine stored password files to verify that passwords are rendered unreadable using a strong, one-way cryptographic algorithm, with a unique input variable at all times when stored.	

PA-DSS Requirements	Testing Procedures	Guidance
3.4 Payment application must limit access to required functions/resources and enforce least privilege for built-in accounts: • By default, all application/service accounts have access to only those functions/resources specifically needed for purpose of the application/service account. • By default, all application/service accounts have minimum level of privilege assigned for each function/resource as needed for the application/service account. ***Aligns with PCI DSS Requirement 7***	**3.4.a** Install the payment application in accordance with the *PA-DSS Implementation Guide* and examine settings for built-in accounts to verify that, by completion of the installation process: • All application/service accounts have access to only those functions/resources specifically needed for purpose of the application/service account. • All application/service accounts have minimum level of privilege assigned for each function/resource as needed for the application/service account.	In order to limit access to cardholder data and sensitive functions to only those accounts that need such access, access needs and the level of privilege required must be defined for each built-in account, such that its assigned functions may be performed but that no additional, unnecessary access or privilege is granted. Assigning least privileges helps prevent users without sufficient knowledge about the application to incorrectly or accidentally change application configuration or alter its security settings. Enforcing least privilege also helps to minimize the scope of damage if an unauthorized person gains access to a user ID.
	3.4.b Test all application functionality that results in changes to built-in accounts, including those that result in user accounts reverting to default settings, changes to existing account settings, generation of new accounts, and recreation of existing accounts. For all types of changes performed, examine settings for built-in accounts and test application functionality to verify that upon completion of the change: • All application/service accounts have access to only those functions/resources specifically needed for purpose of the application/service account. • All application/service accounts have minimum level of privilege assigned for each function/resource as needed for the application/service account.	

Requirement 4: Log payment application activity

PA-DSS Requirements	Testing Procedures	Guidance
4.1 At the completion of the installation process, the "out of the box" default installation of the payment application must log all user access and be able to link all activities to individual users. ***Aligns with PCI DSS Requirement 10.1***	**4.1.a** Install the payment application. Test the application to verify that payment application audit trails are automatically enabled upon installation.	It is critical that the payment application has a process or mechanism that links users to the application resources accessed, generates audit logs, and provides the ability to trace back suspicious activity to a specific user. Post-incident forensic teams heavily depend on these logs to initiate the investigation.
	4.1.b Examine the *PA-DSS Implementation Guide* prepared by the vendor to verify the following instructions are included: • How to install the application so that logs are configured and enabled by default upon completion of the installation process. • How to set PCI DSS-compliant log settings, per PA-DSS Requirements 4.2, 4.3, and 4.4 below, for any logging options that are configurable by the customer after installation. • Logs should not be disabled and doing so will result in non-compliance with PCI DSS. • How to configure PCI DSS-compliant log settings for any third-party software components packaged with or required by the payment application, for any logging options that are configurable by the customer after installation.	
4.2 Payment application must provide automated audit trails to reconstruct the following events: ***Aligns with PCI DSS Requirement 10.2***	**4.2** Test the payment application by examining payment application audit log settings and audit log output, and perform the following:	Logging of the events in 4.2.1 – 4.2.7 enables an organization to identify and trace potentially malicious activities.
4.2.1 All individual user accesses to cardholder data from the application	**4.2.1** Verify all individual access to cardholder data through the payment application is logged.	Malicious individuals could obtain knowledge of a user account that has access to cardholder data through the application, or they could create a new, unauthorized account in order to access cardholder data. A record of all individual accesses to cardholder data can identify which accounts may have been compromised or misused.

PA-DSS Requirements	Testing Procedures	Guidance
4.2.2 All actions taken by any individual with administrative privileges as assigned in the application	**4.2.2** Verify actions taken by any individual with administrative privileges to the payment application are logged.	Accounts with increased privileges, such as an "administrator" account, have the potential to greatly impact the security or operational functionality of the application. Without a log of the activities performed, an organization is unable to trace any issues resulting from an administrative mistake or misuse of privilege back to the specific action and individual.
4.2.3 Access to application audit trails managed by or within the application	**4.2.3** Verify access to application audit trails managed by or within the application is logged.	Malicious users often attempt to alter audit logs to hide their actions, and a record of access allows an organization to trace any inconsistencies or potential tampering of the logs to an individual account.
4.2.4 Invalid logical access attempts	**4.2.4** Verify invalid logical access attempts are logged.	Malicious individuals will often perform multiple access attempts on targeted systems. Multiple invalid login attempts may be an indication of an unauthorized user's attempts to "brute force" or guess a password.
4.2.5 Use of, and changes to the application's identification and authentication mechanisms (including but not limited to creation of new accounts, elevation of privileges, etc.), and all changes, additions, deletions to application accounts with root or administrative privileges	**4.2.5** Verify use of and changes to the payment application's identification and authentication mechanisms (including but not limited to creation of new accounts, elevation of privileges, etc.), and all changes, additions, deletions to application accounts with root or administrative privileges are logged.	Without knowing who was logged on at the time of an incident, it is impossible to identify the accounts used. Additionally, malicious users may attempt to manipulate the authentication controls with the intent of bypassing them or impersonating a valid account. Activities including, but not limited to, creation of new accounts, escalation of privilege, or changes to access permissions may indicate unauthorized use of a system's authentication mechanisms.

PA-DSS Requirements	Testing Procedures	Guidance
4.2.6 Initialization, stopping, or pausing of the application audit logs	**4.2.6** Verify the following are logged: • Initialization of application audit logs • Stopping or pausing of application audit logs.	Turning the audit logs off (or pausing them) prior to performing illicit activities is a common practice for malicious users wishing to avoid detection. Initialization of audit logs could indicate that the log function was disabled by a user to hide their actions.
4.2.7 Creation and deletion of system-level objects within or by the application	**4.2.7** Verify the creation and deletion of system-level objects within or by the application is logged.	Malicious users often create or replace system-level objects on the target system in order to control a particular function or operation on that system. By logging when system-level objects, such as database tables or stored procedures, are created or deleted, it will be easier to determine whether such modifications were authorized.
4.3 Payment application must record at least the following audit trail entries for each event: ***Aligns with PCI DSS Requirement 10.3***	**4.3** Test the payment application by examining the payment application's audit log settings and audit log output, and, for each auditable event (from 4.2), perform the following:	By recording the details in 4.3.1 – 4.3.6 for the auditable events in 4.2, a potential compromise can be quickly identified, and with sufficient detail to know who, what, where, when, and how.
4.3.1 User identification	**4.3.1** Verify user identification is included in log entries.	
4.3.2 Type of event	**4.3.2** Verify type of event is included in log entries.	
4.3.3 Date and time	**4.3.3** Verify date and time stamp is included in log entries.	
4.3.4 Success or failure indication	**4.3.4** Verify success or failure indication is included in log entries.	
4.3.5 Origination of event	**4.3.5** Verify origination of event is included in log entries.	
4.3.6 Identity or name of affected data, system component, or resource	**4.3.6** Verify identity or name of affected data, system component, or resources is included in log entries.	

PA-DSS Requirements	Testing Procedures	Guidance
4.4. Payment application must facilitate centralized logging. ***Note:*** *Examples of this functionality may include, but are not limited to:* • Logging via industry standard log file mechanisms such as Common Log File System (CLFS), Syslog, delimited text, etc. • Providing functionality and documentation to convert the application's proprietary log format into industry standard log formats suitable for prompt, centralized logging. ***Aligns with PCI DSS Requirement 10.5.3***	**4.4.**a Examine the *PA-DSS Implementation Guide* prepared by the vendor to verify that customers and integrators/resellers are provided with: • A description of which centralized logging mechanisms are supported • Instructions and procedures for incorporating the payment application logs into a centralized logging environment. **4.4.b** Install and configure the payment application according to the *PA-DSS Implementation Guide* to verify that the instructions are accurate, and that functionality that facilitates a customer's ability to assimilate logs into their centralized log server is provided.	Without adequate protection of audit logs, their completeness, accuracy, and integrity cannot be guaranteed, and the audit logs can be rendered useless as an investigation tool after a compromise. Including payment application logs in a centralized logging system allows the customer to integrate and correlate their logs, and secure the logs consistently in their environment.

Requirement 5: Develop secure payment applications

PA-DSS Requirements	Testing Procedures	Guidance
5.1 The software vendor has defined and implemented a formal process for secure development of payment applications, which includes: • Payment applications are developed in accordance with PCI DSS and PA-DSS (for example, secure authentication and logging). • Development processes are based on industry standards and/or best practices. • Information security is incorporated throughout the software development life cycle. • Security reviews are performed prior to release of an application or application update. ***Aligns with PCI DSS Requirement 6.3***	**5.1.a** Examine documented software-development processes and verify that processes are based on industry standards and/or best practices.	Without the inclusion of security during the requirements definition, design, analysis, and testing phases of the software-development process, security vulnerabilities can be inadvertently or maliciously introduced into application code.
	5.1.b Verify documented software-development processes include procedures for the following: • Incorporating information security throughout the software development life cycle. • Developing payment applications in accordance with PCI DSS and PA-DSS Requirements.	
	5.1.c Verify documented software-development processes include: • Defined security reviews prior to release of an application or application update. • Procedures for security reviews to be performed to ensure the security objectives of PCI DSS and PA-DSS are being met.	
	5.1.d Interview software developers to confirm that documented processes are followed such that: • Information security is incorporated throughout the software development life cycle. • Payment applications are developed in accordance with PCI DSS and PA-DSS Requirements. • Security reviews are performed prior to release, to ensure that security objectives, including PCI DSS and PA-DSS requirements, are being met.	
5.1.1 Live PANs are not used for testing or development. ***Aligns with PCI DSS Requirement 6.4.3***	**5.1.1.a** Review software development processes to verify that they include procedures to ensure live PANs are not used for testing or development.	Payment card brands and many acquirers are able to provide account numbers suitable for testing in the event that realistic PANs are needed to test application functionality prior to release.
	5.1.1.b Observe testing processes and interview personnel to verify live PANs are not used for testing or development.	
	5.1.1.c Examine samples of test data to verify live PANs are not used for testing or development.	

PA-DSS Requirements	Testing Procedures	Guidance
5.1.2 Test data and accounts are removed before release to customer. ***Aligns with PCI DSS Requirement 6.4.4***	**5.1.2.a** Review software-development processes to verify they include procedures to ensure test data and accounts are removed before payment application is released to customers.	Test data and accounts should be removed from the application before it is released to customers, since inclusion of these items may give away information about key constructs within the application.
	5.1.2.b Observe testing processes and interview personnel to verify test data and accounts are removed before release to customer.	
	5.1.2.c Examine the final payment application product to verify test data and accounts are removed before release to customer.	
5.1.3 Custom payment application accounts, user IDs, and passwords are removed before payment applications are released to customers ***Aligns with PCI DSS Requirement 6.3.1***	**5.1.3.a** Review software-development processes to verify they include procedures to ensure custom payment application accounts, user IDs, and passwords are removed before payment application is released to customers.	Pre-release custom accounts, user IDs and passwords could be used as a back door for developers or other individuals with knowledge of those accounts to gain access to the application, and could facilitate compromise of the application and related cardholder data.
	5.1.3.b Observe testing processes and interview personnel to verify that custom payment application accounts, user IDs, and passwords are removed before payment application is released to customers.	
	5.1.3.c Examine the final payment application product to verify that custom payment application accounts, user IDs, and passwords are removed before payment application is released to customers.	

PA-DSS Requirements	Testing Procedures	Guidance
5.1.4 Payment application code is reviewed prior to release to customers after any significant change, to identify any potential coding vulnerability (using either manual or automated processes) to include at least the following: • Code changes are reviewed by individuals other than the originating code author, and by individuals who are knowledgeable in code-review techniques and secure coding practices. • Code reviews ensure code is developed according to secure coding guidelines. (See PA-DSS Requirement 5.2.) • Appropriate corrections are implemented prior to release. • Code-review results are reviewed and approved by management prior to release. • Documented code-review results include management approval, code author, and code reviewer, and what corrections were implemented prior to release. ***Note:*** *This requirement for code reviews applies to all payment application components (both internal and public-facing web applications), as part of the system development life cycle. Code reviews can be conducted by knowledgeable internal personnel or third parties.* ***Aligns with PCI DSS Requirement 6.3.2***	**5.1.4.a** Examine written software-development procedures and interview responsible personnel to verify the vendor performs code reviews for all significant application code changes (using either manual or automated processes) as follows: • Code changes are reviewed by individuals other than the originating code author, and by individuals who are knowledgeable in code-review techniques and secure coding practices. • Code reviews ensure code is developed according to secure coding guidelines. (See PA-DSS Requirement 5.2.) • Appropriate corrections are implemented prior to release. • Code-review results are reviewed and approved by management prior to release. • Code-review results are documented including management approval, code author, and code reviewer, and what corrections were implemented prior to release. **5.1.4.b** Examine code-review results for a sample of code changes to verify: • Code reviews were performed by a knowledgeable individual other than the code author. • Code reviews were developed according to secure coding guidelines. • Appropriate corrections were implemented prior to release. • Code-review results were reviewed and approved by management prior to release.	Security vulnerabilities in application code are commonly exploited by malicious individuals to gain access to a network and compromise cardholder data. In order to protect against these types of attacks, proper code-reviewing techniques should be used. Code-review techniques should verify that secure coding best practices were employed throughout the development process. The application vendor should incorporate relevant secure coding practices as applicable to the particular technologies used. Reviews should be performed by an individual knowledgeable in the technology and experienced in code-review techniques in order to identify potential coding issues. Assigning code reviews to someone other than the developer of the code allows an independent, objective review to be performed. Correcting coding errors before the code is released prevents faulty code from exposing customer environments to potential exploit. Faulty code is also far more difficult and expensive to address after it has been deployed. Including a formal review and signoff by management prior to release helps to ensure that code is approved and has been developed in accordance with policies and procedures.

PA-DSS Requirements	Testing Procedures	Guidance
5.1.5 Secure source-control practices are implemented to verify integrity of source code during the development process.	**5.1.5.a** Examine written software-development procedures and interview responsible personnel to verify the vendor maintains secure source control practices to verify integrity of source code during the development process.	Good source-code control practices help ensure that all changes to code are intended and authorized, and performed only by those with a legitimate reason to change the code. Examples of these practices include check-in and check-out procedures for code with strict access controls, and a comparison immediately before updating code to confirm that the last approved version hasn't been changed (for example, using a checksum).
	5.1.5.b Examine mechanisms and observe procedures for securing source code to verify integrity of source code is maintained during the development process.	
5.1.6 Payment applications are developed according to industry best practices for secure coding techniques, including: • Developing with least privilege for the application environment. • Developing with fail-safe defaults (all execution is by default denied unless specified within initial design). • Developing for all access point considerations, including input variances such as multi-channel input to the application.	**5.1.6.a** Examine software-development processes to verify that secure coding techniques are defined and include: • Developing with least privilege for the application environment. • Developing with fail-safe default (all execution is by default denied unless specified within initial design). • Developing for all access point considerations, including input variances such as multi-channel input to the application.	Developing applications with least privilege is the most effective way to ensure insecure assumptions aren't introduced into the application. Including fail-safe defaults could prevent an attacker from obtaining sensitive information about an application failure that could then be used to create subsequent attacks. Ensuring that security is applied to all accesses and inputs into the application avoids the likelihood that an input channel may be left open to compromise. Failure to consider these concepts while developing code could result in the release of an insecure application and potentially excessive remediation at a later time.
	5.1.6.b Interview developers to verify that applications are developed according to industry best practices for secure coding techniques, including: • Developing with least privilege for the application environment. • Developing with fail-safe defaults (all execution is by default denied unless specified within initial design). • Developing for all access point considerations, including input variances such as multi-channel input to the application.	

PA-DSS Requirements	Testing Procedures	Guidance
5.1.6.1 Coding techniques include documentation of how PAN and/or SAD are handled in memory.	**5.1.6.1.a** Examine coding techniques to verify they include documentation of how PAN and/or SAD are handled in memory.	Attackers use malware tools to capture sensitive data from memory. Minimizing the exposure of PAN/SAD while in memory will help reduce the likelihood that it can be captured by a malicious user or be unknowingly saved to disk in a memory file and left unprotected. This requirement is intended to ensure that consideration is given for how PAN and SAD are handled in memory. Understanding when and for how long sensitive data is present in memory, as well as in what format, will help application vendors to identify potential insecurities in their applications and determine whether additional protections are needed. Whether or not any coding techniques result from this activity will depend on the particular software being developed and the technologies in use.
	5.1.6.1.b Interview developers to verify that they consider how PAN/SAD are handled in memory during the application-development process.	

PA-DSS Requirements	Testing Procedures	Guidance
5.1.7 Provide training in secure development practices for application developers, as applicable for the developer's job function and technology used, for example: • Secure application design • Secure coding techniques to avoid common coding vulnerabilities (for example, vendor guidelines, OWASP Top 10, SANS CWE Top 25, CERT Secure Coding, etc.) • Managing sensitive data in memory • Code reviews • Security testing (for example, penetration-testing techniques) • Risk-assessment techniques. ***Note:*** *Training for application developers may be provided in-house or by third parties. Examples of how training may be delivered include on-the-job, instructor-led, and computer-based.*	**5.1.7a** Verify documented software-development processes require training in secure development practices for application developers as applicable for the developer's job function and technology used.	Ensuring developers are knowledgeable about secure development practices will help minimize the number of security vulnerabilities introduced through poor coding practices. Trained personnel are also more likely to identify potential security issues in the application design and code. Software-development platforms and methodologies change frequently, as do the threats and risks to software applications. Training in secure development practices should keep up to date with changing development practices.
	5.1.7.b Interview a sample of developers to verify that they are knowledgeable in secure development practices and coding techniques, as applicable to the technology used.	
	5.1.7.c Examine records of training to verify that all application developers receive training as applicable for their job function and technology used.	
5.1.7.1 Update training as needed to address new development technologies and methods used.	**5.1.7.1** Examine training materials and interview a sample of developers to verify that training is updated as needed to address new development technologies and methods used.	
5.2 Develop all payment applications to prevent common coding vulnerabilities in software-development processes. ***Note:*** *The vulnerabilities listed in PA-DSS Requirements 5.2.1 through 5.2.10 and in PCI DSS at 6.5.1 through 6.5.10 were current with industry best practices when this version of PA DSS was published. However, as industry best practices for vulnerability management are updated (for example, the OWASP Top 10, SANS CWE Top 25, CERT Secure Coding, etc.), the current best practices must be used for these requirements.* ***Aligns with PCI DSS Requirement 6.5***	**5.2** Verify that payment applications are not vulnerable to common coding vulnerabilities by performing manual or automated penetration testing that specifically attempts to exploit each of the following:	The application layer is high-risk and may be targeted by both internal and external threats. Without proper security, cardholder data and other confidential company information can be exposed. Requirements 5.2.1 through 5.2.10 are the minimum controls that should be in place. This list is composed of the most common coding vulnerabilities at the time that this version of the PA-DSS was published. As industry-recognized common coding vulnerabilities change, vendor coding practices should likewise be updated to match.

PA-DSS Requirements	Testing Procedures	Guidance
Note: *Requirements 5.2.1 through 5.2.6, below, apply to all payment applications (internal or external):*		
5.2.1 Injection flaws, particularly SQL injection. Also consider OS Command Injection, LDAP and XPath injection flaws as well as other injection flaws.	**5.2.1** Injection flaws, particularly SQL injection, are addressed by coding techniques that include: • Validating input to verify user data cannot modify meaning of commands and queries • Utilizing parameterized queries.	Injection flaws, particularly SQL injection, are a commonly used method for compromising applications. Injection occurs when user-supplied data is sent to an interpreter as part of a command or query. The attacker's hostile data tricks the interpreter into executing unintended commands or changing data, thereby exposing components inside the application to attacks such as buffer overflows. All input data should be validated by the application before being processed—for example, by checking for all alpha characters, mix of alpha and numeric characters, etc.
5.2.2 Buffer Overflow	**5.2.2** Buffer Overflows are addressed by coding techniques that include: • Validating buffer boundaries • Truncating input strings.	Buffer overflows occur when an application does not have appropriate bounds checking on its buffer space. This can cause the information in the buffer to be pushed out of the buffer's memory space and into executable memory space. When this occurs, the attacker has the ability to insert malicious code at the end of the buffer and then push that malicious code into executable memory space by overflowing the buffer. The malicious code is then executed and often enables the attacker remote access to the application and/or infected system.
5.2.3 Insecure cryptographic storage	**5.2.3** Insecure cryptographic storage is addressed by coding techniques that: • Prevent cryptographic flaws • Use strong cryptographic algorithms and keys.	Applications that do not utilize strong cryptographic functions properly to store data are at increased risk of being compromised and exposing authentication credentials and/or cardholder data.

PA-DSS Requirements	Testing Procedures	Guidance
5.2.4 Insecure communications	**5.2.4** Insecure communications are addressed by coding techniques that properly authenticate and encrypt all sensitive communications.	Applications that fail to adequately encrypt sensitive network traffic using strong cryptography are at increased risk of being compromised and exposing cardholder data.
5.2.5 Improper error handling	**5.2.5** Improper error handling is addressed by coding techniques that do not leak information via error messages (for example, by returning generic rather than specific error details).	Applications that leak information about their configuration, internal workings, or expose privileged information through improper error-handling methods are at risk of compromise. Attackers use these weaknesses to steal sensitive data or compromise the system altogether. If a malicious individual can create errors that the application does not handle properly, they can gain detailed system information, create denial-of-service interruptions, cause security to fail, or crash the application or system. For example, the message "incorrect password provided" tells an attacker that the user ID provided was accurate and that they should focus their efforts only on the password. Use more generic error messages, like "data could not be verified."
5.2.6 All “high risk” vulnerabilities as identified in the vulnerability identification process at PA-DSS Requirement 7.1	**5.2.6** Coding techniques address any “high risk” vulnerabilities that could affect the application, as identified in PA-DSS Requirement 7.1.	All vulnerabilities determined by the vendor’s vulnerability risk-ranking process (defined in PA-DSS Requirement 7.1) to be “high risk” and that could affect the application should be identified and addressed during application development.

PA-DSS Requirements	Testing Procedures	Guidance
Note: *Requirements 5.2.7 through 5.2.10, below, apply to web-based applications and application interfaces (internal or external):*		Web applications have unique security risks based upon their architecture as well as the relative ease and occurrence of compromise.
5.2.7 Cross-site scripting (XSS)	**5.2.7** Cross-site scripting (XSS) is addressed by coding techniques that include: • Validating all parameters before inclusion • Utilizing context-sensitive escaping.	XSS flaws occur whenever an application takes user supplied data and sends it to a web browser without first validating or encoding that content. XSS allows attackers to execute script in the victim's browser that can hijack user sessions, deface web sites, introduce worms, etc.
5.2.8 Improper access control such as insecure direct object references, failure to restrict URL access, and directory traversal)	**5.2.8** Improper access control, such as insecure direct object references, failure to restrict URL access, and directory traversal is addressed by coding technique that include: • Proper authentication of users • Sanitizing input • Not exposing internal object references to users • User interfaces that do not permit access to unauthorized functions.	A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, database record, or key, as a URL or form parameter. Attackers can manipulate those references to access other objects without authorization. An attacker who can enumerate and navigate the directory structure of a website (directory traversal) could gain access to unauthorized information as well as further insight into the workings of the site for later exploitation. User interfaces that permit access to unauthorized functions could result in unauthorized individuals gaining access to privileged credentials or cardholder data. Limiting access to data resources will help prevent cardholder data from being presented to unauthorized resources.

PA-DSS Requirements	Testing Procedures	Guidance
5.2.9 Cross-site request forgery (CSRF)	**5.2.9** Cross-site request forgery (CSRF) is addressed by coding techniques that ensure applications do not rely on authorization credentials and tokens automatically submitted by browsers.	A CSRF attack forces a logged-on victim's browser to send a pre-authenticated request to a vulnerable web application, which then enables the attacker to perform any state-changing operations the victim is authorized to perform (such as updating account details, making purchases, or even authenticating to the application).
5.2.10 Broken authentication and session management	**5.2.10** Broken authentication and session management is addressed via coding techniques that commonly include: • Flagging session tokens (for example cookies) as "secure" • Not exposing session IDs in the URL • Incorporating appropriate time-outs and rotation of session IDs after a successful login.	Secure authentication and session management prevents unauthorized individuals from compromising legitimate account credentials, keys, or session tokens that would otherwise enable the intruder to assume the identity of an authorized user.
5.3 Software vendor must follow change-control procedures for all application changes. Change-control procedures must follow the same software development processes as new releases (as defined in PA-DSS Requirement 5.1), and include the following: ***Aligns with PCI DSS Requirement 6.4.5***	**5.3.a** Examine the vendor's change-control procedures for software modifications, and: • Verify the procedures follow documented software-development processes as defined in Requirement 5.1 • Verify that the procedures require items 5.3.1 – 5.3.4 below.	If not properly managed, the impact of software updates and security patches might not be fully realized and could have unintended consequences.
	5.3.b Interview developers to determine recent payment application changes. Examine recent payment application changes and trace them back to related change-control documentation. For each change examined, verify the following was documented according to the change-control procedures:	
5.3.1 Documentation of impact	**5.3.1** Verify that documentation of customer impact is included in the change-control documentation for each change.	The impact of the change should be documented so that all affected parties can plan appropriately for any processing changes.
5.3.2 Documented approval of change by appropriate authorized parties	**5.3.2** Verify that documented approval by appropriate authorized parties is present for each change.	Approval by authorized parties indicates that the change is a legitimate and approved change sanctioned by management.

PA-DSS Requirements	Testing Procedures	Guidance
5.3.3 Functionality testing to verify that the change does not adversely impact the security of the system	**5.3.3.a** Verify that functionality testing was performed to verify that the change does not adversely impact the security of the system. **5.3.3.b** Verify that all changes (including patches) are tested for compliance with 5.2 before being released.	Thorough testing should be performed to verify that the security of the payment application is not reduced by implementing a change. Testing should validate that all existing security controls remain in place, are replaced with equally strong controls, or are strengthened after any change to the application.
5.3.4 Back-out or product de-installation procedures	**5.3.4** Verify that back-out or product de-installation procedures are prepared for each change.	For each change, there should be back-out procedures in case the change fails or adversely affects the security of the application, to allow the application to be restored back to its previous state.
5.4 The payment application vendor must document and follow a software-versioning methodology as part of their system development lifecycle. The methodology must follow the procedures in the *PA-DSS Program Guide* for changes to payment applications and include at least the following:	**5.4** Examine documented software development processes to verify they include the software vendor's versioning methodology, and that the versioning methodology must be in accordance with the PA-DSS Program Guide. Verify that the documented versioning methodology is required to be followed for the payment application, including all changes to the payment application.	Without a thoroughly defined versioning methodology, changes to applications may not be properly identified, and customers and integrators/resellers may not understand the impact of a version change to the application.
5.4.1 The versioning methodology must define the specific version elements used, as follows: • Details of how the elements of the version scheme are in accordance with requirements specified in the *PA-DSS Program Guide*. • The format of the version scheme, including number of elements, separators, character set, etc. (consisting of alphabetic, numeric, and/or alphanumeric characters). *(Continued on next page)*	**5.4.1.a** Examine the documented versioning methodology to verify it includes the following: • Details of how the elements of the version numbering scheme are in accordance with requirements specified in the *PA-DSS Program Guide.* • The format of the version numbering scheme is specified and includes details of number of elements, separators, character set, etc. (e.g., 1.1.1.N, consisting of alphabetic, numeric, and/or alphanumeric characters). • A definition of what each element represents in the version-numbering scheme (e.g., type of change, major, minor, or maintenance release, wildcard, etc.) • Definition of elements that indicate use of wildcards.	Payment application vendor versioning methodology should include a defined version scheme that specifically identifies the elements being used, format of the version, hierarchy of different version elements, and so on, for the particular payment application. The version scheme should clearly specify how each of the various elements is used in the version number. *(Continued on next page)*

PA-DSS Requirements	Testing Procedures	Guidance
• Definition of what each element represents in the version scheme (for example, type of change, major, minor, or maintenance release, wildcard, etc.) • Definition of elements that indicate use of wildcards. ***Note:*** *Wildcards may only be substituted for elements of the version number that represent non-security impacting changes. Refer to Requirement 5.4.3 for additional requirements on the use of wildcards.*	**5.4.1.b** Verify the elements of the version scheme are in accordance with the types of changes specified in the PA-DSS Program Guide.	The version scheme can be indicated in a number of ways—for example, N.NN.NNA, where "N" indicates a numeric element and "A" indicates an alphabetic element. The versioning scheme should include identification of the character set (for example, 0-9, A-Z, etc.) that can be used for each element in the version. Without a properly defined version scheme, changes made to the application may not be accurately represented by the version number format.
	5.4.1.c Select a sample of recent payment application changes, the version numbers assigned, and the change-control documentation that specifies the type of application change, and verify that the elements in the version number match the applicable change and the parameters defined in the documented versioning methodology.	
	5.4.1.d Interview a sample of developers and verify that they are knowledgeable in the version scheme, including the acceptable use of wildcards in the version number.	
5.4.2 The versioning methodology must indicate the type and impact of all application changes in accordance with the *PA-DSS Program Guide,* including: • Descriptions of all types and impacts of application changes. • Specific identification and definition of changes that: - Have no impact on functionality of the application or its dependencies - Have impact on application functionality but no impact on security or PA-DSS requirements - Have impact to any security functionality or PA-DSS requirement. • How each type of change ties to a specific version number.	**5.4.2.a** Examine the software vendor's documented versioning methodology to verify the version methodology includes: • Description of all types and impacts of application changes (for example, changes that have no impact, low impact, or high impact to the application) • Specific identification and definition of changes that: - Have no impact on functionality of the application or its dependencies - Have impact on application functionality but no impact on security or PA-DSS requirements - Have impact to any security functionality or PA-DSS requirement. • How each type of change ties to a specific version number.	
	5.4.2.b Verify that the versioning methodology is in accordance with the *PA-DSS Program Guide* requirements.	
	5.4.2.c Interview personnel and observe processes for each type of change to verify that the documented methodology is followed for all types of changes.	

PA-DSS Requirements	Testing Procedures	Guidance
	5.4.2.d Select a sample of recent payment application changes and review the change control documentation that specifies the type of application change to verify that the version assigned to the change matches the type of change according to the documented methodology.	
5.4.3 The versioning methodology must specifically identify whether wildcards are used and, if so, how they are used. The following must be included: • Details of how wildcards are used in the versioning methodology. • Wildcards are never used for any change that has an impact on security or any PA-DSS requirements. • Any element of the version number used to represent a non-security-impacting change (including a wildcard element) must never be used to represent a security impacting change. • Wildcard elements must not precede version elements that could represent security-impacting changes. Any version elements that appear after a wildcard element must not be used to represent security-impacting changes. ***Note:*** *Wildcards may only be used in accordance with the PA-DSS Program Guide.*	**5.4.3.a** Examine the software vendor's documented versioning methodology to verify that it includes specific identification of how wildcards are used, including: • Details of how wildcards are used in the versioning methodology. • Wildcards are never used for any change that has an impact on security or any PA-DSS requirements. • Any element of the version number used to represent a non-security-impacting change (including a wildcard element) must never be used to represent a security impacting change. • Any elements to the right of a wildcard cannot be used for a security-impacting change. Version elements reflecting a security-impacting change must appear "to the left of" the first wildcard element. **5.4.3.b** Verify that any use of wildcards is in accordance with the *PA-DSS Program Guide* requirements. For example, elements that appear after a wildcard element cannot be used for a security impacting change. **5.4.3.c** Interview personnel and observe processes for each type of change to verify that: • Wildcards are never used for any change that has an impact on security or any PA-DSS requirements. • Elements of the version number used to represent non-security-impacting changes (including a wildcard element) are never used to represent a security impacting change.	A PA-DSS "wildcard" element can optionally be used in the version scheme to represent multiple non-security-impacting changes. A wildcard is the only variable element of the vendor's version scheme and is used to indicate there are only minor, non-security-impacting changes between each version represented by the wildcard element. For example, a version number of 1.1.x would cover specific versions 1.1.2 and 1.1.3, etc., letting a customer know that the code base between them is effectively unchanged except for cosmetic or other minor types of changes. Any use of wildcards must be predefined in the vendor's versioning methodology and be used only in accordance with the *PA-DSS Program Guide*. ***Note:*** *Use of a wildcard is optional and is not required.*

PA-DSS Requirements	Testing Procedures	Guidance
	5.4.3.d Select a sample of recent payment application changes and review the change control documentation that specifies the type of application change. Verify that: • Wildcards are not used for any change that has an impact on security or any PA-DSS requirements. • Elements of the version number used to represent non-security-impacting changes (including a wildcard element) are not used to represent a security impacting change.	
5.4.4 The vendor's published versioning methodology must be communicated to customers and integrators/resellers.	**5.4.4** Verify the *PA-DSS Implementation Guide* includes a description of the vendor's published versioning methodology for customers and integrators/resellers, and includes the following: • Details of versioning scheme, including the format of the version scheme (number of elements, separators, character set, etc.). • Details of how security-impacting changes will be indicated by the version scheme. • Details of how other types of changes will affect the version. • Details of any wildcard elements that are used, including confirmation that they will never be used to represent a security-impacting change.	Ensuring the vendors' versioning methodology is included in the *PA-DSS Implementation Guide* will provide customers and integrators/resellers the necessary information to understand which version of the payment application they are using as well as the types of changes that have been made to each version of the payment application.
5.4.5 If an internal version mapping to published versioning scheme is used, the versioning methodology must include mapping of internal versions to the external versions.	**5.4.5.a** Examine the documented version methodology to verify it includes a mapping of internal versions to published external versions.	Some payment application vendors have versioning methodologies for internal use or reference that differ from the versioning methodology used for external (or public) releases. In these situations it is important that both versioning methodologies are well defined and documented, and the relationship between them is thoroughly documented.
	5.4.5.b Examine recent changes to confirm that internal version mapping to published versioning scheme is updated in accordance with the type of change, as defined in the documented methodology.	

PA-DSS Requirements	Testing Procedures	Guidance
5.4.6 Software vendor must have a process in place to review application updates for conformity with the versioning methodology prior to release.	**5.4.6.a** Examine documented software-development processes and the versioning methodology to verify there is a process in place to review application updates for conformity with the versioning methodology prior to release. **5.4.6.b** Interview software developers and observe processes to verify that application updates are reviewed for conformity with the versioning methodology prior to release.	It is critical that payment application vendors have a process in place to ensure product updates match the intent and scope of the planned release, and that those changes are accurately communicated to customers. Otherwise, changes could be made to an application that may negatively impact a customer's application security without their knowledge.
5.5 Risk assessment techniques (for example, application threat-modeling) are used to identify potential application security design flaws and vulnerabilities during the software-development process. Risk assessment processes include the following: • Coverage of all functions of the payment application, including but not limited to, security-impacting features and features that cross trust-boundaries. • Assessment of application decision points, process flows, data flows, data storage, and trust boundaries. • Identification of all areas within the payment application that interact with PAN and/or SAD or the cardholder data environment (CDE), as well as any process-oriented outcomes that could lead to the exposure of cardholder data. • A list of potential threats and vulnerabilities resulting from cardholder data flow analyses and assign risk ratings (for example, high, medium, or low priority) to each. • Implementation of appropriate corrections and countermeasures during the development process. • Documentation of risk assessment results for management review and approval.	**5.5** Examine written software-development procedures and interview responsible personnel to verify the vendor uses risk assessment techniques as part of the software-development process, and that the processes include: • Coverage of all functions of the payment application, including but not limited to, security-impacting features and features that cross trust boundaries. • Assessment of application decision points, process flows, data flows, data storage, and trust boundaries. • Identification of all areas within payment applications that interact with PAN/SAD or the cardholder data environment (CDE), as well as any process-oriented outcomes that could lead to the exposure of cardholder data. • A list of potential threats and vulnerabilities resulting from cardholder data flow analyses, and assign risk ratings (e.g. high, medium, or low priority) to each. • Implementation of appropriate corrections and countermeasures during the development process. • Documentation of risk assessment results for management review and approval.	To maintain the quality and security of payment applications, risk assessment techniques should be employed by application vendors during the software-development process. Threat modeling is a form of risk assessment that can be used to analyze an application's constructs and data flows for opportunities where confidential information may be exposed to unauthorized application users. These processes allow software developers and architects to identify and resolve potential security issues early in the development process, improving application security and minimizing development costs.

PA-DSS Requirements	Testing Procedures	Guidance
5.6 Software vendor must implement a process to document and authorize the final release of the application and any application updates. Documentation includes: • Signature by an authorized party to formally approve release of the application or application update • Confirmation that secure development processes were followed by the vendor.	**5.6.a** Examine documented processes to verify that final release of the application and any application updates must be formally approved and documented, including a signature by an authorized party to formally approve the release and confirmation that all SDLC processes were followed.	Someone within the payment application vendor's organization should be responsible for reviewing and ensuring all aspects of the secure development processes (as defined in Requirements 5.1 through 5.5) were performed. Without a formal review and acknowledgment from a responsible party, critical security processes may be missed or excluded, resulting in a faulty or less secure application.
	5.6.b For a sample of recent releases of application and application updates, review approval documentation to verify it includes • Formal approval and signature by an authorized party • Confirmation that that all secure development processes were followed.	

Requirement 6: Protect wireless transmissions

PA-DSS Requirements	Testing Procedures	Guidance
6.1 For payment applications using wireless technology, change wireless vendor defaults, including but not limited to default wireless encryption keys, passwords, and SNMP community strings. The wireless technology must be implemented securely. ***Aligns with PCI DSS Requirements 1.2.3 & 2.1.1***	**6.1** For payment applications developed for use with wireless technology, and for all wireless applications bundled with the payment application, verify that the wireless applications do not use vendor default settings, as follows:	The exploitation of wireless technology is a common method for malicious individuals to gain access to the network and cardholder data. If wireless networks are not implemented with sufficient security configurations (including changing default settings), wireless sniffers can eavesdrop on the traffic, easily capture data and passwords, and easily enter and attack a network. For these reasons, payment applications must not require the use of default or insecure wireless settings. If firewalls do not restrict access from wireless networks into the CDE, malicious individuals that gain unauthorized access to the wireless network can easily connect to the CDE and compromise account information.
	6.1.a Examine the *PA-DSS Implementation Guide* prepared by the vendor to verify it includes the following for customers and integrators/resellers: • The payment application enforces changes of default encryption keys, passwords and SNMP community strings at installation for all wireless components controlled by the application. • Procedures for changing wireless encryption keys and passwords, including SNMP strings, anytime anyone with knowledge of the keys/passwords leaves the company or changes positions. • Instructions for changing default encryption keys, passwords, and SNMP community strings on any wireless components provided with, but not controlled by, the payment application. • Instructions to install a firewall between any wireless networks and systems that store cardholder data. • Details of any wireless traffic (including specific port information) that the wireless function of the payment application would use. • Instructions to configure firewalls to deny or—if such traffic is necessary for business purposes—permit only authorized traffic between the wireless environment and the cardholder data environment.	

PA-DSS Requirements	Testing Procedures	Guidance
	6.1.b Install the application according to the *PA-DSS Implementation Guide* and test application and wireless settings to verify the following, for all wireless functionality managed by the payment application: • Encryption keys were changed from default at installation. • Default SNMP community strings on wireless devices were changed at installation. • Default passwords/passphrases on access points were changed at installation. • Firmware on wireless devices is updated to support strong encryption for authentication and transmission over wireless networks. • Other security-related wireless vendor defaults were changed, if applicable.	
	6.1.c For all wireless functionality managed by the payment application, follow instructions in the *PA-DSS Implementation Guide* for changing wireless encryption keys, passwords/passphrases, and SNMP strings. Verify that the *PA-DSS Implementation Guide* instructions are accurate and result in changed wireless encryption keys, passwords and SNMP strings.	
	6.1.d For all wireless components provided with, but not controlled by, the payment application, follow instructions in the *PA-DSS Implementation Guide* for changing default encryption keys, passwords/passphrases and SNMP community strings. Verify the *PA-DSS Implementation Guide* instructions are accurate and result in changed wireless encryption keys, passwords, and SNMP strings.	
	6.1.e Install the application and test wireless functions to verify the wireless traffic and ports used by the application are in accordance with those documented in the *PA-DSS Implementation Guide*.	

PA-DSS Requirements	Testing Procedures	Guidance
6.2 For payment applications using wireless technology, payment application must facilitate use of industry best practices (for example, IEEE 802.11i) to implement strong encryption for authentication and transmission. ***Note:*** *The use of WEP as a security control is prohibited.* ***Aligns with PCI DSS Requirement 4.1.1***	**6.2.a** For payment applications developed for use with wireless technology, test all wireless functionality to verify the application uses industry best practices (for example, IEEE 802.11.i) to provide strong encryption for authentication and transmission.	Malicious users use free and widely available tools to eavesdrop on wireless communications. Use of strong cryptography can help limit disclosure of sensitive information across wireless networks. Strong cryptography for authentication and transmission of cardholder data is required to prevent malicious users from gaining access to data on a wireless network or utilizing wireless networks to access other systems or data.
	6.2.b For all wireless applications bundled with the payment application, test wireless functionality to verify that industry best practices (for example, IEEE 802.11.i) are used to provide strong encryption for authentication and transmission.	
	6.2.c Examine the *PA-DSS Implementation Guide* prepared by the vendor to verify it includes the following instructions for customers and integrators/resellers: • How to configure the application to use industry best practices (for example, IEEE 802.11.i) for strong encryption for authentication and transmission, and/or • How to configure all wireless applications bundled with the payment application to use industry best practices for strong encryption for authentication and transmission.	
6.3 Provide instructions for customers about secure use of wireless technology, ***Note:*** *This requirement applies to all payment applications, regardless of whether the application is developed for use with wireless technologies.* **Aligns with *PCI DSS Requirements 1.2.3, 2.1.1, & 4.1.1***	**6.3** Examine the *PA-DSS Implementation Guide* prepared by the vendor to verify customers and integrators/resellers are instructed on PCI DSS-compliant wireless settings as follows: • Instructions to change all wireless default encryption keys, passwords, and SNMP community strings upon installation. • Instructions to change wireless encryption keys, passwords, and SNMP strings anytime anyone with knowledge of the keys/passwords leaves the company or changes positions. • Instructions to install a firewall between any wireless networks and systems that store cardholder data and to configure firewalls to deny or, if traffic is necessary for business purposes, permit only authorized traffic between the wireless environment and the cardholder data environment. • Instructions to use industry best practices (for example, IEEE 802.11.i) to provide strong encryption for authentication and transmission.	Payment application vendors will need to provide instructions to customers for configuring the application to support the use of wireless technologies, even if the application is not designed explicitly for use in a wireless environment. Wireless networks are commonplace, and customers should be made aware of common wireless security settings that should be implemented to ensure the security of the payment application.

Requirement 7: Test payment applications to address vulnerabilities and maintain payment application updates

PA-DSS Requirements	Testing Procedures	Guidance
7.1 Software vendors must establish a process to identify and manage vulnerabilities, as follows: ***Note:*** *Any underlying software or systems that are provided with or required by the payment application (for example, web servers, third-party libraries and programs) must be included in this process.* ***Aligns with PCI DSS Requirement 6.1***	**7.1.a** Examine vulnerability management process documentation to verify procedures are defined to: • Identify new security vulnerabilities using reputable sources for obtaining security vulnerability information • Assign a risk ranking to all identified vulnerabilities • Test payment applications and updates for the presence of vulnerabilities prior to release. **7.1.b** Verify that processes to identify new vulnerabilities and implement corrections into payment application apply to all software provided with or required by the payment application (for example, web servers, third-party libraries and programs).	Vendors need to keep up-to-date with new vulnerabilities that may impact their applications, including vulnerabilities in underlying components or software packaged with or required by the application. Payment application vendors knowledgeable of vulnerabilities within their own applications or in underlying components should then be able to resolve those vulnerabilities prior to release, or implement other mechanisms to reduce the likelihood that the vulnerability may be exploited in the event a third-party security patch is not immediately available.
7.1.1 Identify new security vulnerabilities using reputable sources for obtaining security vulnerability information.	**7.1.1** Interview responsible personnel and observe processes to verify new security vulnerabilities are identified: • In both the payment application and any underlying software or systems provided with or required by the payment application • Using reputable sources (such as software/systems vendor websites, NIST's NVD, MITRE's CVE, and the DHS's US-CERT websites).	Reputable sources should be used for vulnerability information and/or patches in third-party software components. Sources for vulnerability information should be trustworthy and often include vendor websites, industry news groups, mailing lists, or RSS feeds. Examples of industry sources include NIST's National Vulnerability Database, MITRE's Common Vulnerabilities and Exposures list, and the US Department of Homeland Security's US-CERT websites.

PA-DSS Requirements	Testing Procedures	Guidance
7.1.2 Assign a risk ranking to all identified vulnerabilities, including vulnerabilities involving any underlying software or systems provided with or required by the payment application. ***Note:*** *Risk rankings should be based on industry best practices as well as consideration of potential impact. For example, criteria for ranking vulnerabilities may include consideration of the CVSS base score, and/or the classification by the vendor, and/or impact to application functionality.* *Risk rankings should, at a minimum, identify all vulnerabilities considered to be a "high risk" to the application. In addition to the risk ranking, vulnerabilities may be considered "critical" if they pose an imminent threat, impact critical application components, or would result in a potential compromise if not addressed.*	**7.1.2** Interview responsible personnel and observe processes to verify new security vulnerabilities are assigned a risk ranking, including vulnerabilities involving any underlying software or systems provided with or required by the payment application.	Once the vendor identifies a vulnerability that could affect their application, the risk that the vulnerability poses must be evaluated and ranked. This requires a process to actively monitor industry sources for vulnerability information. Classifying the risks (for example, as "high," "medium," or "low") allows vendors to identify, prioritize and address the highest risk items (for example, by releasing high-priority patches more quickly), and reduce the likelihood that vulnerabilities posing the greatest risk to customer environments will be exploited.
7.1.3 Test payment applications and updates for the presence of vulnerabilities prior to release	**7.1.3** Interview responsible personnel and observe processes to verify that payment applications are tested for the presence of vulnerabilities prior to release.	Adequate testing should be included in any payment application vendor's vulnerability management process to ensure that any identified vulnerabilities have been properly addressed prior to release. *Examples of testing methods may include penetration testing and/or fuzz-testing techniques to identify potential vulnerabilities—for example, by injecting malformed or unexpected data, or modifying the bit size of the data.*
7.2 Software vendors must establish a process for timely development and deployment of security patches and upgrades.	**7.2** Examine process documentation for the development and distribution of security patches and upgrades to verify the process include procedures for 7.2.1 through 7.2.2:	Software updates to address security vulnerabilities should be developed and released to customers as quickly as possible once a critical vulnerability has been identified, to minimize the timeframe and likelihood that the vulnerability could be exploited.

PA-DSS Requirements	Testing Procedures	Guidance
7.2.1 Patches and updates are delivered to customers in a secure manner with a known chain of trust.	**7.2.1** Interview responsible personnel and observe processes to verify patches and updates are delivered to customers in a secure manner with a known chain of trust.	Security patches must be distributed in a manner that prevents malicious individuals from intercepting the updates in transit, modifying them, and then redistributing them to unsuspecting customers.
7.2.2 Patches and updates are delivered to customers in a manner that maintains the integrity of the patch and update code.	**7.2.2.a** Interview responsible personnel and observe processes to verify patches and updates are delivered to customers in a manner that maintains the integrity of the patch and update code.	Security updates should include a mechanism within the update process to verify the update code has not been replaced or tampered with. Examples of integrity checks include, but are not limited to, checksums, digitally-signed certificates, etc.
	7.2.2.b Interview responsible personnel and observe application update processes to verify patches and updates are integrity-tested on the target system prior to installation.	
	7.2.2.c Verify that the integrity of patch and update code is maintained by running the update process with arbitrary code and determining that the system will not allow the update to occur.	
7.3 Include release notes for all application updates, including details and impact of the update, and how the version number was changed to reflect the application update.	**7.3.a** Examine processes for releasing updates and interview personnel to verify release notes are prepared for all updates, including details and impact of the update, and how the version number was changed to reflect the application update.	Release notes provide customers with details about software updates, including which files may have changed, which application functionality was modified, as well as any security-related features that may be affected. Release notes should also indicate how a particular patch or update affects the overall version number associated with the patch release.
	7.3.b Examine release notes for a sample of application updates and verify they were provided with the update.	

Requirement 8: Facilitate secure network implementation

PA-DSS Requirements	Testing Procedures	Guidance
8.1 The payment application must be able to be implemented into a secure network environment. Application must not interfere with use of devices, applications, or configurations required for PCI DSS compliance. *For example, payment application cannot interfere with installation of patches, anti-malware protection, firewall configurations, or any other device, application, or configuration required for PCI DSS compliance.* ***Aligns with PCI DSS Requirements 1, 3, 4, 5, and 6***	**8.1.a** Install the application in a PCI DSS compliant laboratory environment according to the *PA-DSS Implementation Guide.* Test the payment application to obtain evidence that it can run in a network that is fully compliant with PCI DSS. **8.1.b** Test the application and underlying systems to verify that the payment application does not preclude the use of or interfere with PCI DSS functions on underlying systems—for example, the application does not inhibit installation of patches or anti-malware updates, or interfere with the operation of other PCI DSS functions.	Payment applications should be designed and developed in such a way that the installation and operation of the application must not prevent an organization in implementing other controls required for PCI DSS compliance. For example, the payment application must be able to operate in an environment that runs anti-virus solutions (for example, doesn't require these solutions to be turned off or uninstalled).
8.2 The payment application must only use or require use of necessary and secure services, protocols, daemons, components, and dependent software and hardware, including those provided by third parties, for any functionality of the payment application. *For example, if NetBIOS, file-sharing, Telnet, FTP, etc., are required by the application, they are secured via SSH, S-FTP, TLS, IPSec, or other technology.* *Note: SSL and early TLS are not considered strong cryptography. Payment applications must not use, or support the use of, SSL or early TLS. Applications that use or support TLS must not allow fallback to SSL.* ***Aligns with PCI DSS Requirement 2.2.3***	**8.2.a** Examine system services, protocols, daemons, components, and dependent software and hardware enabled or required by the payment application. Verify that only necessary and secure services, protocols, daemons, components, dependent software and hardware are enabled by default "out of the box". **8.2.b** Install the application and test application functions to verify that if the application supports any insecure services, daemons, protocols or components, they are securely configured by default "out of the box". **8.2.c** Verify that the *PA-DSS Implementation Guide* documents all required protocols, services, components, and dependent software and hardware that are necessary for any functionality of the payment application, including those provided by third parties.	There are many protocols that a business may need (or have enabled by default) that are commonly used by malicious individuals to compromise a system or network. The payment application should not require the use of an insecure protocol, service, daemon, etc. If the application does support the use of insecure services, daemons, protocols or components, they must be secured by default.

PA-DSS Requirements	Testing Procedures	Guidance
8.3 The payment application must not require use of services or protocols that preclude the use of or interfere with normal operation of two-factor authentication technologies for securing remote access to the payment application that originates from outside the customer environment. ***Note:*** *Two-factor authentication requires that two of the three authentication methods (see below) be used for authentication. Using one factor twice (for example, using two separate passwords) is not considered two-factor authentication. The authentication methods, also known as a factors, are:* • Something you know, such as a password or passphrase • Something you have, such as a token device or smart card • Something you are, such as a biometric *Examples of two-factor technologies include RADIUS with tokens, TACACS with tokens, or other technologies that facilitate two-factor authentication.* ***Aligns with PCI DSS Requirement 8.3***	**8.3.a** Examine payment application functionality to verify it does not require use of any services or protocols that preclude the use of or interfere with the normal operation of two-factor authentication technologies for remote access. **8.3.b** Identify remote-access mechanisms supported by the application and verify that the mechanisms do not prevent two-factor authentication.	Payment applications should be designed and developed in such a way that the installation and operation of the application must not require an organization to use services or protocols that would inhibit that organization from implementing and operating two-factor authentication solutions for secure remote access. For example, the application should not, by default, use port 1812 (which is universally known to be assigned to RADIUS by RFC 2865) if RADIUS is intended to be a supported authentication and authorization technology.

Requirement 9: Cardholder data must never be stored on a server connected to the Internet

PA-DSS Requirements	Testing Procedures	Guidance
9.1 The payment application must be developed such that any web server and any cardholder data storage component (for example, a database server) are not required to be on the same server, nor is the data storage component required to be on the same network zone (such as a DMZ) with the web server. ***Aligns with PCI DSS Requirement 1.3.7***	**9.1.a** Identify all payment application data storage components (for example, databases) and all web servers. Install data storage components and web servers on different servers and test application functionality across the different servers, Verify the payment application does not require any data storage component (such as a database) to be installed on the same server as a web server in order to function.	Any web-server component of a payment application is at substantially higher risk of compromise given the open nature of public networks (Internet, public wireless, etc.) and the nature and volume of attacks that can originate from those networks. Cardholder data storage components require a higher level of protection than public-facing application components. If cardholder data is located within the DMZ, it is easier for an external attacker to access this information, since there are fewer layers to penetrate. For the same reason, web servers should never be stored on the same server as the data storage component. If a malicious individual were able to compromise an account on the web server, they could also have compromised the cardholder database with no additional effort required.
	9.1.b Install data storage components and web servers on different network zones. Test all application functions across the network zones to verify that the payment application does not require any data storage component (such as a database) to be installed on the same network zone as a web server in order to function.	
	9.1.c Examine the *PA-DSS Implementation Guide* prepared by the vendor to verify it includes the following for customers and integrators/resellers: • Instructions not to store cardholder data on public-facing systems (for example, web server and database server must not be on same server). • Instructions on how to configure the payment application to use a DMZ to separate the Internet from systems storing cardholder data (for example, installing a web server component in a DMZ and installing a data storage component on an internal different network zone). • A list of services/ports that the application needs to use in order to communicate across two network zones (so the customer can configure their firewall to open only required ports).	

Requirement 10: Facilitate secure remote access to payment application

PA-DSS Requirements	Testing Procedures	Guidance
10.1 Two-factor authentication must be used for all remote access to the payment application that originates from outside the customer environment. ***Note:*** *Two-factor authentication requires that two of the three authentication methods be used for authentication (see PA-DSS Requirement 3.1.4 for descriptions of authentication methods).* **Aligns with *PCI DSS Requirement 8.3***	**10.1.a** Examine the *PA-DSS Implementation Guide* prepared by the vendor to verify it contains the following for customers and integrators/resellers: • Instructions that all remote access originating from outside the customer's network to the payment application must use two-factor authentication in order to meet PCI DSS requirements. • A description of two-factor authentication mechanisms supported by the application. • Instructions for configuring the application to support two-factor authentication (two of the three authentication methods described in PA DSS Requirement 3.1.4).	Two-factor authentication requires two methods of authentication for access originating from outside the network. Payment application vendors will need to provide instructions to customers for configuring the application to support the specified two-factor authentication mechanisms in order to ensure those mechanisms can be implemented properly and meet applicable PCI DSS requirements. The requirement for two-factor authentication applies only where the remote access originates from outside the customer environment
	10.1.b If the application vendor has remote access to a customer's payment application that originates from outside the customer environment, examine vendor policies to verify that the vendor supports customer requirements for two-factor authentication for all such access.	
10.2 Any remote access into the payment application must be performed securely, as follows:	**10.2** Verify that any remote access is performed as follows:	Any remote-access mechanism employed by the payment application vendor and/or integrators/resellers—for example, to support services being delivered by those providers—should support all applicable PCI DSS requirements.
10.2.1 If payment application updates are delivered via remote access into customers' systems, software vendors must tell customers to turn on remote-access technologies only when needed for downloads from vendor, and to turn off immediately after download completes. Alternatively, if delivered via virtual private network (VPN) or other high-speed connection, software vendors must advise customers to properly configure a firewall or a personal firewall product to secure "always-on" connections. ***Aligns with PCI DSS Requirements 1 and 12.3.9***	**10.2.1.a** If payment application updates are delivered via remote access into customers' systems, examine the *PA-DSS Implementation Guide* prepared by the vendor and verify it contains: • Instructions for customers and integrators/resellers regarding secure use of remote-access technologies, specifying that remote-access technologies used by vendors and business partners should be activated only when needed and immediately deactivated after use. • Recommendation for customers and integrators/resellers to use a securely configured firewall or a personal firewall product if computer is connected via VPN or other high-speed connection, to secure these "always-on" connections, per PCI DSS Requirement 1.	

PA-DSS Requirements	Testing Procedures	Guidance
	10.2.1.b If the vendor delivers payment application and/or updates via remote access to customer networks, observe the vendor's methods for delivering payment application and/or updates via remote access to customer networks, and verify the vendor method includes: • Activation of remote-access technologies to customer networks only when needed and immediate deactivation after use. • If remote access is via VPN or other high-speed connection, the connection is secured according to PCI DSS Requirement 1.	
10.2.2 If vendors or integrators/resellers can access customers' payment applications remotely, a unique authentication credential (such as a password/phrase) must be used for each customer. ***Aligns with PCI DSS Requirements 8.5.1***	**10.2.2** If vendors or integrators/resellers can access customers' payment applications remotely, examine vendor processes and interview personnel to verify that a unique authentication credential (such as a password/phrase) is used for each customer they have access to.	To prevent the compromise of multiple customers' environments through the use of a single set of credentials, vendors with remote-access accounts to customer environments should use a different authentication credential for each customer. Avoid the use of repeatable formulas to generate password that are easily guessed. These credentials become known over time and can be used by unauthorized individuals to compromise the vendor's customers.

PA-DSS Requirements	Testing Procedures	Guidance
10.2.3 Remote access to customers' payment applications by vendors, integrators/resellers, or customers must be implemented securely, for example: • Change default settings in the remote-access software (for example, change default passwords and use unique passwords for each customer). • Allow connections only from specific (known) IP/MAC addresses. • Use strong authentication and complex passwords for logins (See PA-DSS Requirements 3.1.1 through 3.1.11). • Enable encrypted data transmission according to PA-DSS Requirement 12.1. • Enable account lockout after a certain number of failed login attempts. (See PA-DSS Requirements 3.1.9 through 3.1.10.) • Establish a VPN connection via a firewall before access is allowed. • Enable the logging function. • Restrict access to customer environments to authorized integrators/resellers personnel. ***Aligns with PCI DSS Requirements 2, 8 and 10***	**10.2.3.a** Examine the *PA-DSS Implementation Guide* prepared by the vendor, and verify that customers and integrators/resellers are instructed that all remote access to the payment application must be implemented securely for example: • Change default settings in the remote-access software (for example, change default passwords and use unique passwords for each customer). • Allow connections only from specific (known) IP/MAC addresses. • Use strong authentication and complex passwords for logins (See PA-DSS Requirements 3.1.1 through 3.1.11). • Enable encrypted data transmission according to PA-DSS Requirement 12.1. • Enable account lockout after a certain number of failed login attempts. (See PA-DSS Requirement 3.1.9 through 3.1.10.) • Establish a VPN connection via a firewall before access is allowed. • Enable the logging function. • Restrict access to customer environments to authorized personnel. **10.2.3.b** If the software vendor can access customers' payment applications remotely, observe the vendor's remote-access methods and interview personnel to verify the remote access is implemented securely	Payment application vendors will need to provide instructions to customers and integrators/resellers for configuring the application to support secure remote access in order to ensure those mechanisms can be implemented properly and meet PCI DSS requirements. This requirement applies to all types of remote access used to access the customer environment.

Requirement 11: Encrypt sensitive traffic over public networks

PA-DSS Requirements	Testing Procedures	Guidance
11.1 If the payment application sends, or facilitates sending, cardholder data over public networks, the payment application must support use of strong cryptography and security protocols (for example, TLS, IPSEC, SSH, etc.) to safeguard sensitive cardholder data during transmission over open, public networks, including at least the following: • Only trusted keys and certificates are accepted. • The protocol in use only supports secure versions or configurations • The encryption strength is appropriate for the encryption methodology in use ***Note:*** *SSL and early TLS are not considered strong cryptography. Payment applications must not use, or support the use of, SSL or early TLS. Applications that use or support TLS must not allow fallback to SSL* *Examples of open, public networks include but are not limited to:* • The Internet • Wireless technologies, including but not limited to 802.11 and Bluetooth • Cellular technologies, for example, Global System for Mobile Communications (GSM), Code division multiple access (CDMA) • General Packet Radio Service (GPRS) • Satellite communications ***Aligns with PCI DSS Requirement 4.1***	**11.1.a** If the payment application sends, or facilitates sending, cardholder data over public networks, verify that strong cryptography and security protocols are provided with the application, or that use thereof is specified. **11.1.b** Examine the *PA-DSS Implementation Guide* prepared by the vendor, and verify the vendor includes directions for customers and integrators/resellers to use the strong cryptography and security protocols provided by or specified for use with the application, including: • Instructions that strong cryptography and security protocols must be used if cardholder data is ever transmitted over public networks. • Instructions for verifying that only trusted keys and/or certificates are accepted. • How to configure the payment application to use only secure versions and secure implementations of security protocols. • How to configure the payment application to prevent fallback to an insecure version or configuration (e.g. if TLS is used, the application must not allow fallback to SSL). • How to configure the payment application to use the proper encryption strength for the encryption methodology in use. **11.1.c** If strong cryptography and security protocols are provided with the payment application, install and test the application according to instructions in the *PA-DSS Implementation Guide*, and verify: • The protocol is implemented by default to use only trusted keys and/or certificates. • The protocol is implemented by default to use only secure configurations and does not support insecure versions or configurations. • The protocol is implemented by default to not allow fallback to an insecure version or configuration (e.g. if TLS is used, the application must not allow fallback to SSL). • Proper encryption strength is implemented for the encryption methodology in use.	Because it is easy and common for a malicious individual to intercept and/or divert data while in transit, sensitive information must be encrypted during transmission over public networks. Secure transmission of cardholder data requires using trusted keys/certificates, a secure protocol for transport, and proper encryption strength to encrypt cardholder data. Note that some protocol implementations (such as SSL, SSH version 1.0, and early TLS) have documented vulnerabilities, such as buffer overflows, that an attacker can use to gain control of the affected system. Regardless of which security protocols are used by the payment application, ensure they are configured by default to use only secure configurations and versions to prevent an insecure connection being used.

PA-DSS Requirements	Testing Procedures	Guidance
11.2 If the payment application facilitates sending of PANs by end-user messaging technologies (for example, e-mail, instant messaging, chat), the payment application must provide a solution that renders the PAN unreadable or implements strong cryptography, or specify use of strong cryptography to encrypt the PANs. ***Aligns with PCI DSS Requirement 4.2***	**11.2.a** If the payment application allows and/or facilitates sending of PANs by end-user messaging technologies, verify that a solution that renders the PAN unreadable or implements strong cryptography is provided, or that use thereof is specified.	E-mail, instant messaging, and chat can be easily intercepted by packet-sniffing during delivery traversal across internal and public networks. Do not utilize these messaging tools to send PAN unless the payment application provides for the use of strong cryptography with these technologies or renders the PAN unreadable.
	11.2.b Examine *PA-DSS Implementation Guide* prepared by the vendor, and verify the vendor includes directions for customers and integrators/resellers to use a solution provided with or specified for use with the application, including: • Procedures for using the defined solution to render the PAN unreadable or secure the PAN with strong cryptography. • Instruction that PAN must always be rendered unreadable or secured with strong cryptography whenever it is sent via end-user messaging technologies.	
	11.2.c If a solution is provided with the payment application, install and test the application to verify that the solution renders the PAN unreadable or implements strong cryptography.	

Requirement 12: Encrypt all non-console administrative access

PA-DSS Requirements	Testing Procedures	Guidance
12.1 If the payment application facilitates non-console administrative access, encrypt all such access with strong cryptography using technologies such as SSH, VPN, or TLS, for web-based management and other non-console administrative access. ***Notes:*** • *Clear-text protocols such as Telnet or rlogin must never be used for administrative access.* • *SSL and early TLS are not considered strong cryptography. Payment applications must not use, or support the use of, SSL or early TLS. Applications that use or support TLS must not allow fallback to SSL.* ***Aligns with PCI DSS Requirement 2.3***	**12.1.a** Install the payment application in a lab and test non-console administrative connections to verify that a strong encryption method is invoked before the administrator's password is requested. **12.1.b** Examine payment application configuration settings to verify that clear-text protocols, such as Telnet and rlogin, are not used by the payment application for non-console administrative access. **12.1.c** Examine the *PA-DSS Implementation Guide* prepared by the vendor, and verify it includes instructions for customers and integrators/resellers how to configure the application to use strong cryptography, using technologies such as SSH, VPN, or TLS, for encryption of non-console administrative access.	If remote administration is not done with secure authentication and encrypted communications, sensitive administrative or operational level information (like administrator's passwords) can be revealed to an eavesdropper. A malicious individual could use this information to access the application and/or network, modify permissions, and steal data.
12.2 Instruct customers to encrypt all non-console administrative access with strong cryptography, using technologies such as SSH, VPN, or TLS for web-based management and other non-console administrative access. ***Note:*** *Clear-text protocols such as Telnet or rlogin must never be used for administrative access.* ***Aligns with PCI DSS Requirement 2.3***	**12.2** Examine the *PA-DSS Implementation Guide* prepared by the vendor and verify it includes instructions for customers and integrators/resellers to implement strong cryptography, using technologies such as SSH, VPN, or TLS, for encryption of all non-console administrative access.	Payment application vendors will need to provide instructions to customers and integrators/resellers for configuring the application to use strong cryptography for encryption of all non-console administrative access. Doing so helps to ensure the security controls are implemented properly and meet PCI DSS and PA-DSS guidelines.

Requirement 13: Maintain a PA-DSS Implementation Guide for customers, resellers, and integrators

PA-DSS Requirements	Testing Procedures	Guidance
13.1 Develop, maintain, and disseminate a *PA-DSS Implementation Guide(s)* for customers, resellers, and integrators that accomplishes the following:	**13.1** Examine the *PA-DSS Implementation Guide* and related vendor processes, and interview personnel to verify: • The *PA-DSS Implementation Guide* is disseminated to all customers, resellers, and integrators with the application. • The vendor has a mechanism in place to provide the *PA-DSS Implementation Guide* to customers, resellers, and integrators upon request.	A well-designed and detailed *PA-DSS Implementation Guide* helps guide customers and integrators/resellers in the implementation of appropriate security measures and configurations within the payment application and its underlying components in order to meet the relevant PCI DSS and PA-DSS guidelines for protecting cardholder data.
13.1.1 Provides relevant information specific to the application for customers, resellers, and integrators to use.	**13.1.1** Examine the *PA-DSS Implementation Guide* and verify it: • Clearly identifies the payment application name and version to which it applies. • Provides details of all application dependencies that are required in order for the application to be configured in a PCI DSS compliant manner.	
13.1.2 Addresses all requirements in this document wherever the *PA-DSS Implementation Guide* is referenced.	**13.1.2** Examine the *PA-DSS Implementation Guide* and, using Appendix A as a reference, verify the *PA-DSS Implementation Guide* covers all related requirements in this document.	
13.1.3 Includes a review at least annually and upon changes to the application or to the PA-DSS requirements, and is updated as needed to keep the documentation current with all changes affecting the application, as well as to the requirements in this document.	**13.1.3.a** Examine the *PA-DSS Implementation Guide* and interview personnel to verify the *PA-DSS Implementation Guide* is reviewed: • At least annually • Upon changes to the application • Upon changes to these PA-DSS requirements.	With each application update, system functionality and, in some cases, critical application security mechanisms are modified or introduced. If the *PA-DSS Implementation Guide* is not kept current with the latest versions of the payment application, customers and integrators/resellers could overlook or misconfigure critical application security controls that could ultimately enable an attacker to bypass such security mechanisms and compromise sensitive data.
	13.1.3.b Verify the *PA-DSS Implementation Guide* is updated as needed to keep current with: • Changes to the PA-DSS requirements • Changes to the application or its dependencies.	

PA-DSS Requirements	Testing Procedures	Guidance
	13.1.3.c Examine the *PA-DSS Implementation Guide* and related vendor processes, and interview personnel to verify the vendor has a mechanism in place to communicate updates to customers, resellers, and integrators, and provide updated versions as needed.	

Requirement 14: Assign PA-DSS responsibilities for personnel, and maintain training programs for personnel, customers, resellers, and integrators

PA-DSS Requirements	Testing Procedures	Guidance
14.1 Provide training in information security and PA-DSS for vendor personnel with PA-DSS responsibility at least annually.	**14.1** Examine training materials and interview responsible personnel to verify that all vendor personnel with PA-DSS responsibility receive training in PA-DSS and information security at least annually.	In order for a payment application to be designed effectively to meet PA-DSS guidelines, payment application vendor personnel should be knowledgeable in PA-DSS and their responsibilities with regards to ongoing PA-DSS assessments. It is the responsibility of the payment application vendor to ensure their personnel are properly educated in these areas.
14.2 Assign roles and responsibilities to vendor personnel including the following: • Overall accountability for meeting all the requirements in PA-DSS • Keeping up-to-date within any changes in the PCI SSC PA-DSS Program Guide • Ensuring secure coding practices are followed • Ensuring integrators/resellers receive training and supporting materials • Ensuring all vendor personnel with PA-DSS responsibilities, including developers, receive training	**14.2.a** Examine documented responsibilities to verify that responsibility for the following roles is formally assigned: • Overall accountability for meeting all the requirements in PA-DSS • Keeping up-to-date within any changes in the PCI SSC PA-DSS Program Guide • Ensuring secure coding practices are followed • Ensuring integrators/resellers receive training and supporting materials • Ensuring all vendor personnel with PA-DSS responsibilities, including developers, receive training.	Within each payment application vendor organization, a responsible party (either an individual or a team) should be assigned formal responsibility for PA-DSS to ensure all PA-DSS requirements are met accordingly.
	14.2.b Interview personnel assigned responsibility for the following roles to confirm that roles and responsibilities are defined and understood: • Overall accountability for meeting all the requirements in PA-DSS • Keeping up-to-date within any changes in the PCI SSC PA-DSS Program Guide • Ensuring secure coding practices are followed • Ensuring integrators/resellers receive training and supporting materials • Ensuring all vendor personnel with PA-DSS responsibilities, including developers, receive training.	

PA-DSS Requirements	Testing Procedures	Guidance
14.3 Develop and implement training and communication programs for payment application integrators and resellers. Training should include at least the following: • How to implement the payment application and related systems and networks in a PCI DSS-compliant manner • Coverage of all items noted for the *PA-DSS Implementation Guide* throughout this document (and in Appendix A).	**14.3.a** Examine the training materials for integrators and resellers, and confirm the materials include the following: • Training on how to implement the payment application and related systems and networks in a PCI DSS compliant manner • Coverage of all items noted for the *PA-DSS Implementation Guide* throughout this document (and in Appendix A).	Incorrect configuration, maintenance or support of an application may lead to security vulnerabilities being introduced into the customer's cardholder data environment, which could then be exploited by attackers. Application vendors should provide training for integrator/resellers in the secure installation and configuration of the application to ensure that, when installed in the customer's environment, the application facilitates PCI DSS compliance It is the responsibility of the payment application vendor to provide integrators and resellers with training in these areas.
	14.3.b Examine the vendor's communication programs and related vendor processes, and interview vendor personnel to verify: • Training materials are provided to integrators and resellers • The vendor has a mechanism in place to provide updated materials to integrators and resellers upon request.	
	14.3.c Interview a sample of integrators and resellers to verify that they received the training and training materials from the application vendor.	
	14.3.d Examine evidence of integrators and resellers receipt of the training materials from the software vendor.	
14.3.1 Review training materials at least annually and upon changes to the application or to PA-DSS requirements. Update the training materials as needed to keep the documentation current with new payment application versions and changes to PA-DSS requirements.	**14.3.1.a** Examine the training materials for integrators and resellers and verify the materials are: • Reviewed at least annually and upon changes to the application or to PA-DSS requirements • Updated as needed to keep the documentation current with new payment application versions and changes to PA-DSS requirements.	Training materials for payment application vendor personnel, integrators and resellers should be updated at least annually to ensure the materials remain current with the latest versions of the applications and PA-DSS requirements. Use of outdated training materials could render training programs ineffective, leading to poorly designed security functions within the application or improper application configurations by integrators and resellers.
	14.3.1.b Examine the distribution process for new payment application versions and verify that updated documentation is distributed to integrators and resellers with the updated payment application.	
	14.3.1.c Interview a sample of integrators and resellers to verify they received updated training materials from the application vendor.	

Appendix A: Summary of Contents for the *PA-DSS Implementation Guide*

The intent of this Appendix is to summarize those PA-DSS requirements that have related *PA-DSS Implementation Guide* topics, to explain the content for the *PA-DSS Implementation Guide* provided to customers and integrators/resellers (see "PA-DSS Implementation Guide" on page 11), and to spell out responsibilities for implementing the related controls.

PA-DSS Requirement	PA-DSS Topic	Required Implementation Guide Content	Control Implementation Responsibility
1.1.4	Delete sensitive authentication data stored by previous payment application versions.	The following instructions must be provided for customers and integrators/resellers: ▪ Historical data must be removed (track data, card verification codes, PINs, or PIN blocks stored by previous versions of the payment application), ▪ How to remove historical data. ▪ Such removal is absolutely necessary for PCI DSS compliance.	**Software Vendor**: Provide tool or procedure for customers to securely remove sensitive authentication data stored by previous versions, per PA-DSS Requirement 1.1.4. **Customers & Integrators/Resellers**: Delete any historical data per the *PA-DSS Implementation Guide* and PA-DSS Requirement 1.1.4.
1.1.5	Delete any sensitive authentication data (pre-authorization) gathered as a result of troubleshooting the payment application.	The following instructions must be provided for customers and integrators/resellers: ▪ Sensitive authentication data (pre-authorization) must only be collected when needed to solve a specific problem. ▪ Such data must be stored only in specific, known locations with limited access. ▪ Only collect a limited amount of such data as needed to solve a specific problem. ▪ Sensitive authentication data must be encrypted while stored. ▪ Such data must be securely deleted immediately after use.	**Software Vendor**: Do not store sensitive authentication data; and perform any troubleshooting of customer's problems according to PA-DSS Requirement 1.1.5.a. **Customers & Integrators/Resellers:** Do not store sensitive authentication data; and troubleshoot any problems per the *PA-DSS Implementation Guide* and PA-DSS Requirement 1.1.5.a.

PA-DSS Requirement	PA-DSS Topic	Required Implementation Guide Content	Control Implementation Responsibility
2.1	Securely delete cardholder data after customer-defined retention period.	The following must be provided for customers and integrators/resellers: ▪ Instruction that cardholder data exceeding the customer-defined retention period must be securely deleted. ▪ A list of all locations where payment application stores cardholder data, so that customer knows the locations of data that needs to be deleted. ▪ Instruction that customers need to securely delete cardholder data when no longer required for legal, regulatory, or business purposes. ▪ How to securely delete cardholder data stored by the payment application, including data stored on underlying software or systems (such as OS, databases, etc.). ▪ How to configure the underlying software or systems (such as OS, databases, etc.) to prevent inadvertent capture or retention of cardholder data.	**Software Vendor**: Provide guidance to customers that cardholder data exceeding customer-defined retention periods must be securely deleted where such data is stored by the payment application and underlying software or systems, and how to securely delete cardholder data stored by the payment application. **Customers & Integrators/Resellers**: Securely delete cardholder data exceeding customer-defined retention period, per the *PA-DSS Implementation Guide* and PA-DSS Requirement 2.1.
2.2	Mask PAN when displayed so only personnel with a business need can see the full PAN.	The following must be provided for customers and integrators/resellers: ▪ Details of all instances where PAN is displayed, including but not limited to POS devices, screens, logs, and receipts. ▪ Confirmation that the payment application masks PAN by default on all displays. ▪ Instructions on how to configure the payment application such that only personnel with a legitimate business need can see the full PAN.	**Software Vendor**: Provide instructions to customers for masking PAN so only personnel with a business need can see the full PAN. **Customers & Integrators/Resellers**: Mask displays of PAN so only personnel with a business need can see the full PAN, per the *PA-DSS Implementation Guide* and PA-DSS Requirement 2.2.

PA-DSS Requirement	PA-DSS Topic	Required Implementation Guide Content	Control Implementation Responsibility
2.3	Render PAN unreadable anywhere it is stored (including data on portable digital media, backup media, and in logs).	The following must be provided for customers and integrators/resellers: ▪ Details of any configurable options for each method used by the application to render cardholder data unreadable, and instructions on how to configure each method for all locations where cardholder data is stored by the payment application (per PA-DSS Requirement 2.1). ▪ A list of all instances where cardholder data may be output for the customer to store outside of the payment application, and instructions that the customer is responsible for rendering PAN unreadable in all such instances.	**Software Vendor**: Provide instructions to customers for rendering PAN unreadable anywhere it is stored or output by the application. **Customers & Integrators/Resellers**: Render PAN unreadable anywhere it is stored per the *PA-DSS Implementation Guide* and PA-DSS Requirement 2.3.
2.4	Protect keys used to secure cardholder data against disclosure and misuse.	The following instructions must be provided for customers and integrators/resellers: ▪ Restrict access to keys to the fewest number of custodians necessary. ▪ Store keys securely in the fewest possible locations and forms.	**Software Vendor**: Provide guidance to customers that keys used to secure cardholder data should be stored securely in the fewest possible locations, and access to keys must be restricted to the fewest possible custodians. **Customers & Integrators/Resellers**: Store keys securely in the fewest possible locations, and restrict access to keys to the fewest possible custodians, per the *PA-DSS Implementation Guide* and PA-DSS Requirement 2.4.
2.5	Implement key-management processes and procedures for cryptographic keys used for encryption of cardholder data.	The following must be provided for customers and integrators/resellers: ▪ Instructions on how to securely generate, distribute, protect, change, store, and retire/replace cryptographic keys, where customers or integrators/resellers are involved in these key-management activities. ▪ A sample Key Custodian Form for key custodians to acknowledge that they understand and accept their key-custodian responsibilities.	**Software Vendor**: Provide instructions to customers that access cryptographic keys used for encryption of cardholder data to implement key-management processes and procedures. **Customers & Integrators/Resellers**: Implement key-management processes and procedures for cryptographic keys used for encryption of cardholder data per *PA-DSS Implementation Guide* and PA-DSS Requirement 2.5.

PA-DSS Requirement	PA-DSS Topic	Required Implementation Guide Content	Control Implementation Responsibility
2.5.1 – 2.5.7	Implement secure key-management functions.	Provide instructions for customers and integrators/resellers on how to perform key-management functions including: ▪ Generation of strong cryptographic keys. ▪ Secure cryptographic key distribution. ▪ Secure cryptographic key storage. ▪ Cryptographic key changes for keys that have reached the end of their cryptoperiod. ▪ Retirement or replacement of keys as deemed necessary when the integrity of the key has been weakened or keys are suspected of being compromised. ▪ Split knowledge and dual control for any manual clear-text cryptographic key-management operations supported by the payment application. ▪ Prevention of unauthorized substitution of cryptographic keys.	**Software Vendor**: Provide instructions to customers to implement secure key-management functions. **Customers & Integrators/Resellers**: Implement secure key-management functions for cryptographic keys per *PA-DSS Implementation Guide* and PA-DSS Requirements 2.5.1 – 2.5.7.
2.6	Provide a mechanism to render irretrievable cryptographic key material or cryptograms stored by the payment application.	The following must be provided for customers and integrators/resellers: ▪ Procedures detailing how to use the tool or procedure provided with the application to render cryptographic material irretrievable. ▪ Instruction that cryptographic key material be rendered irretrievable whenever keys are no longer used and in accordance with key-management requirements in PCI DSS. ▪ Instructions on how to re-encrypt historic data with new keys, including procedures for maintaining security of clear-text data during the decryption/re-encryption process.	**Software Vendor**: Provide tool or procedure to securely remove cryptographic key material or cryptograms stored by the application, and provide tool or procedure to re-encrypt historic data with new keys. **Customers & Integrators/Resellers**: Delete any historical cryptographic material in accordance with key-management requirements per *PA-DSS Implementation Guide* and PA-DSS Requirement 2.6.

PA-DSS Requirement	PA-DSS Topic	Required Implementation Guide Content	Control Implementation Responsibility
3.1	Use unique user IDs and secure authentication for administrative access and access to cardholder data.	The following must be provided for customers and integrators/resellers: ▪ Directions on how the payment application enforces strong authentication for any authentication credentials (for example, users, passwords) that the application generates or manages, by: - Enforcing secure changes to authentication credentials by the completion of installation per PA-DSS requirements 3.1.1 through 3.1.11. - Enforcing secure changes to authentication credentials for any subsequent changes (after installation) per PA-DSS requirements 3.1.1 through 3.1.11. ▪ That, to maintain PCI DSS compliance, any changes made to authentication configurations would need to be verified as providing authentication methods that are at least as rigorous as PCI DSS requirements. ▪ Assign secure authentication to all default accounts in the environment ▪ For any default accounts that won't be used, assign secure authentication and then disable or do not use the accounts. ▪ How to change and create authentication credentials when such credentials are not generated or managed by the payment application, per PA-DSS Requirements 3.1.1 through 3.1.11, by the completion of installation and for subsequent changes after installation, for all application level accounts with administrative access or access to cardholder data.	**Software Vendor**: For all authentication credentials generated or managed by the application, ensure payment application enforces customer's use of unique user IDs and secure authentication for accounts/passwords, per PA-DSS Requirements 3.1.1 through 3.1.11. For authentication credentials not generated or managed by the payment application, ensure the *PA-DSS Implementation Guide* provides clear and unambiguous guidance for customers and integrators/resellers on how to change and create secure authentication credentials per PA-DSS Requirements 3.1.1 through 3.1.11. **Customers & Integrators/Resellers**: Establish and maintain unique user IDs and secure authentication per the *PA-DSS Implementation Guide* and PA-DSS Requirements 3.1.1 through 3.1.11.

PA-DSS Requirement	PA-DSS Topic	Required Implementation Guide Content	Control Implementation Responsibility
3.2	Use unique user IDs and secure authentication for access to PCs, servers, and databases with payment applications.	Instruct customers and integrators/resellers to use unique user names and secure authentication to access any PCs, servers, and databases with payment applications and/or cardholder data, per PA-DSS requirements 3.1.1 through 3.1.11.	**Software Vendor**: Ensure payment application supports customer's use of unique user IDs and secure authentication for accounts/passwords if set by vendor to access PCs, servers, and databases, per PA-DSS requirements 3.1.2 through 3.1.9. **Customers & Integrators/Resellers**: Establish and maintain unique user IDs and secure authentication per the *PA-DSS Implementation Guide* and PA-DSS requirements 3.1.1 through 3.1.11.
4.1	Implement automated audit trails.	Provide instructions for implementing automated audit trails to include: ▪ How to install the application so that logs are configured and enabled by default upon completion of the installation process. ▪ How to set PCI DSS-compliant log settings, per PA-DSS Requirements 4.2, 4.3 and 4.4, for any logging options that are configurable by the customer after installation. ▪ Logs must be enabled, and disabling the logs will result in non-compliance with PCI DSS. ▪ How to configure PCI-compliant log settings for any third-party software components packaged with or required by the payment application, for any logging options that are configurable by the customer after installation.	**Software Vendor**: Ensure payment application supports customer's use of compliant logs per PA-DSS Requirements 4.2, 4.3 and 4.4. **Customers & Integrators/Resellers:** Establish and maintain PCI DSS-compliant logs per the *PA-DSS Implementation Guide* and PA-DSS Requirements 4.2, 4.3 and 4.4.
4.4	Facilitate centralized logging.	Provide a description of which centralized logging mechanisms are supported, as well as instructions and procedures for incorporating the payment application logs into a centralized logging server.	**Software Vendor**: Ensure payment application supports centralized logging in customer environments per PA-DSS Requirement 4.4. **Customers & Integrators/Resellers:** Establish and maintain centralized logging per the *PA-DSS Implementation Guide* and PA-DSS Requirement 4.4.

PA-DSS Requirement	PA-DSS Topic	Required Implementation Guide Content	Control Implementation Responsibility
5.4.4	Implement and communicate application versioning methodology.	Provide a description of the vendor's published versioning methodology and include guidance for the following: ▪ Details of versioning scheme, including the format of the version scheme (number of elements, separators, character set, etc.). ▪ Details of how security-impacting changes will be indicated by the versioning scheme. ▪ Details of how other types of changes will affect the version. ▪ Details of any wildcard elements that are used, including that they will never be used to represent a security-impacting change.	**Software Vendor**: Document and implement a software-versioning methodology as part of the system development lifecycle. The methodology must follow the procedures in the *PA-DSS Program Guide* for changes to payment applications, per PA-DSS Requirement 5.5. **Customers & Integrators/Resellers:** Understand which version of the payment application they are using, and ensure validated versions are in use.

PA-DSS Requirement	PA-DSS Topic	Required Implementation Guide Content	Control Implementation Responsibility
6.1	Securely implement wireless technology.	For payment applications developed for use with wireless technology, the following must be provided for customers and integrators/resellers: ▪ Instruction that the payment application enforces changes of default encryption keys, passwords, and SNMP community strings at installation for all wireless components controlled by the application. ▪ Procedures for changing wireless encryption keys and passwords, including SNMP strings, anytime anyone with knowledge of the keys/passwords leaves the company or changes positions. ▪ Instructions for changing default encryption keys, passwords, and SNMP community strings on any wireless components provided with, but not controlled by, the payment application. ▪ Instructions to install a firewall between any wireless networks and systems that store cardholder data. ▪ Details of any wireless traffic (including specific port information) that the wireless function of the payment application would use. ▪ Instructions to configure firewalls to deny or (if such traffic is necessary for business purposes) permit only authorized traffic between the wireless environment and the cardholder data environment.	**Software Vendor**: Instruct customers and integrators/resellers, that if wireless technology is used with the payment application, the wireless vendor default settings must be changed per PA-DSS Requirement 6.1. **Customers & Integrators/Resellers**: For wireless implemented into the payment environment by customers or integrators/resellers, change vendor defaults per PA-DSS Requirement 6.1 and install a firewall per the *PA-DSS Implementation Guide* and PCI DSS Requirement 2.1.1.

PA-DSS Requirement	PA-DSS Topic	Required Implementation Guide Content	Control Implementation Responsibility
6.2	Secure transmissions of cardholder data over wireless networks.	For payment applications developed for use with wireless technology, include instructions for using industry best practices (for example, IEEE 802.11i) to implement strong encryption for authentication and transmission of cardholder data. This includes: ▪ How to configure the application to use industry best practices (for example, IEEE 802.11.i) for strong encryption for authentication and transmission, and/or ▪ How to configure all wireless applications bundled with the payment application to use industry best practices for strong encryption for authentication and transmission.	**Software Vendor**: Instruct customers and integrators/resellers, that if wireless technology is used with the payment application, secure encrypted transmissions must be implemented per PA-DSS Requirement 6.2. **Customers & Integrators/Resellers**: For wireless implemented into the payment environment by customers or integrators/resellers, use secure encrypted transmissions per the *PA-DSS Implementation Guide* and PA-DSS Requirement 6.2.
6.3	Provide instructions for secure use of wireless technology.	Provide instructions for PCI DSS-compliant wireless settings, including: ▪ Instructions to change all wireless default encryption keys, passwords, and SNMP community strings upon installation. ▪ Instructions to change wireless encryption keys, passwords, and SNMP strings anytime anyone with knowledge of the keys/passwords leaves the company or changes positions. ▪ Instructions to install a firewall between any wireless networks and systems that store cardholder data, and to configure firewalls to deny or control (if such traffic is necessary for business purposes) any traffic from the wireless environment into the cardholder data environment. ▪ Instructions to use industry best practices (for example, IEEE 802.11.i) to provide strong encryption for authentication and transmission.	**Software Vendor**: Instruct customers and integrators/resellers, to secure wireless technologies per PA-DSS Requirement 6.3. **Customers & Integrators/Resellers**: Secure wireless technologies per the *PA-DSS Implementation Guide* and PA-DSS Requirement 6.2.

PA-DSS Requirement	PA-DSS Topic	Required Implementation Guide Content	Control Implementation Responsibility
8.2	Use only necessary and secure services, protocols, components, and dependent software and hardware, including those provided by third parties.	Document all required protocols, services, components, and dependent software and hardware that are necessary for any functionality of the payment application.	**Software Vendor:** Ensure payment application supports customer's use of only necessary and secure protocols, services, etc., by 1) having only necessary protocols, services, etc., established "out of the box" by default, 2) having those necessary protocols, services, etc., securely configured by default, and 3) by documenting necessary protocols, services, etc., as a reference for customers and integrators/resellers. **Customers and Integrators/Resellers:** Use the documented list from the *PA-DSS Implementation Guide* to ensure only necessary and secure protocols, services, etc., are used on the system, in accordance with PA-DSS Requirement 5.4.
9.1	Store cardholder data only on servers not connected to the Internet.	The following must be provided for customers and integrators/resellers: ▪ Instructions not to store cardholder data on public-facing systems (for example, web server and database server must not be on same server). ▪ Instructions on how to configure the payment application to use a DMZ to separate the Internet from systems storing cardholder data. ▪ A list of services/ports that the application needs to use in order to communicate across two network zones (so the customer can configure their firewall to open only required ports).	**Software Vendor**: Ensure payment application does not require cardholder data storage in the DMZ or on Internet-accessible systems, and will allow use of a DMZ per PA-DSS Requirement 9. **Customers & Integrators/Resellers**: Establish and maintain payment applications so that cardholder data is not stored on Internet-accessible systems, per the *PA-DSS Implementation Guide* and PA-DSS Requirement 9

PA-DSS Requirement	PA-DSS Topic	Required Implementation Guide Content	Control Implementation Responsibility
10.1	Implement two-factor authentication for all remote access to payment application that originates from outside the customer environment.	Provide the following for customers and integrators/resellers: ▪ Instruction that all remote access originating from outside the customer's network to the payment application must use two-factor authentication in order to meet PCI DSS requirements. ▪ Description of the two-factor authentication mechanisms supported by the application. ▪ Instructions on how to configure the application to support two-factor authentication (two of the three authentication methods described in PA DSS Req. 3.1.4).	**Software Vendor**: Ensure payment application supports customers' use of two-factor authentication for all remote access to the payment application that originates from outside the customer environment, per PA-DSS Requirement 10.2. **Customers & Integrators/Resellers**: Establish and maintain two-factor authentication for all remote access to payment application that originates from outside the customer environment, per the *PA-DSS Implementation Guide* and PA-DSS Requirement 10.2.
10.2.1	Securely deliver remote payment application updates.	If payment application updates are delivered via remote access into customers' systems, provide the following: ▪ Instructions for activation of remote-access technologies for payment application updates only when needed for downloads, and turning access off immediately after download completes, per PCI DSS Requirement 12.3.9. ▪ Instructions that, if computer is connected via VPN or other high-speed connection, receive remote payment application updates via a securely configured firewall or personal firewall per PCI DSS Requirement 1.	**Software Vendor**: Deliver remote payment application updates securely per PA-DSS 10.3 **Customers & Integrators/Resellers**: Receive remote payment application updates from vendor securely, per the *PA-DSS Implementation Guide,* PA-DSS Requirement 10.3 and PCI DSS Requirement 1.

PA-DSS Requirement	PA-DSS Topic	Required Implementation Guide Content	Control Implementation Responsibility
10.2.3	Securely implement remote-access software.	Include instructions that all remote access to the payment application must be implemented securely, for example: ▪ Change default settings in the remote-access software (for example, change default passwords and use unique passwords for each customer). ▪ Allow connections only from specific (known) IP/MAC addresses. ▪ Use strong authentication and complex passwords for logins (See PA-DSS Requirements 3.1.1 through 3.1.11). ▪ Enable encrypted data transmission according to PA-DSS Requirement 12.1. ▪ Enable account lockout after a certain number of failed login attempts (See PA-DSS Requirement 3.1.9 through 3.1.10). ▪ Establish a Virtual Private Network ("VPN") connection via a firewall before access is allowed. ▪ Enable the logging function. ▪ Restrict access to customer environments to authorized integrator/reseller personnel.	**Software Vendor**: (1) If vendor can access customers' payment applications remotely, implemented secure remote access such as those specified in PA-DSS Requirement 10.3.2. (2) Ensure payment application supports customers' use of remote-access security features. **Customers & Integrators/Resellers**: Use remote-access security features for all remote access to payment applications, per the *PA-DSS Implementation Guide* and PA-DSS Requirement 10.3.2.

PA-DSS Requirement	PA-DSS Topic	Required Implementation Guide Content	Control Implementation Responsibility
11.1	Secure transmissions of cardholder data over public networks.	If the payment application sends, or facilitates sending, cardholder data over public networks, include instructions for implementing and using strong cryptography and security protocols for secure cardholder data transmission over public networks, including: ▪ Required use of strong cryptography and security protocols if cardholder data is ever transmitted over public networks. ▪ Instructions for verifying that only trusted keys and/or certificates are accepted. ▪ How to configure the payment application to use only secure versions and secure implementations of security protocols. ▪ How to configure the payment application to use the proper encryption strength for the encryption methodology in use.	**Software Vendor**: Ensure payment application supports customer's use of strong cryptography and security protocols for transmissions of cardholder data over public networks, per PA-DSS Requirement 11.1. **Customers & Integrators/Resellers**: Establish and maintain strong cryptography and security protocols for transmissions of cardholder data, per the *PA-DSS Implementation Guide* and PA-DSS Requirement 11.1.
11.2	Encrypt cardholder data sent over end-user messaging technologies.	If the payment application facilitates sending of PANs by end-user messaging technologies, include instructions for implementing and using a solution that renders the PAN unreadable or implements strong cryptography, including: ▪ Procedures for using the defined solution to render the PAN unreadable or secure the PAN with strong cryptography. ▪ Instruction that PAN must always be rendered unreadable or secured with strong cryptography whenever it is sent via end-user messaging technologies.	**Software Vendor**: Provide or specify use of a solution that renders the PAN unreadable or implements strong cryptography, and ensure payment application supports the encryption or rendering unreadable of PANs if sent with end-user messaging technologies, per PA-DSS Requirement 11.2. **Customers & Integrators/Resellers**: Render unreadable or encrypt with strong cryptography all PANs sent with end-user messaging technologies, per the *PA-DSS Implementation Guide* and PA-DSS Requirement 11.2.

PA-DSS Requirement	PA-DSS Topic	Required Implementation Guide Content	Control Implementation Responsibility
12.1	Encrypt non-console administrative access.	If the payment application facilitates non-console administrative access, include instructions on how to configure the application to use strong cryptography (such as SSH, VPN, or TLS) for encryption of all non-console administrative access to payment application or servers in cardholder data environment.	**Software Vendor**: If the payment application facilitates non-console administrative access, ensure payment application implements strong encryption for non-console administrative access, per PA-DSS Requirement 12.1. **Customers & Integrators/Resellers**: Encrypt all non-console administrative access, per the *PA-DSS Implementation Guide* and PA-DSS Requirement 12.1
12.2	Encrypt non-console administrative access.	Include instructions for customers and integrators/resellers to implement strong cryptography, using technologies such as SSH, VPN, or TLS, for encryption of all non-console administrative access.	**Software Vendor**: Ensure payment application supports customer's encryption of non-console administrative access, per PA-DSS Requirement 12.2. **Customers & Integrators/Resellers**: Encrypt all non-console administrative access, per the *PA-DSS Implementation Guide* and PA-DSS Requirement 12.2

Appendix B: Testing Laboratory Configuration for PA-DSS Assessments

For each PA-DSS assessment conducted, the PA-QSA must confirm the status and capabilities of the laboratory used to conduct the testing for the PA-DSS assessment. This confirmation must be submitted along with the completed *Report of Validation (ROV)*.

For each Laboratory Validation Procedure, the PA-QSA must indicate whether the laboratory used for the assessment and the laboratory undergoing these validation procedures was the PA-QSA's laboratory or software vendor's laboratory. PA-QSAs are required to maintain a testing laboratory that meets all of the requirements set out below and use their own laboratory to conduct assessments whenever possible. The software vendor's laboratory may only be used when necessary (for example, the PA-QSA does not have the mainframe, AS400, or Tandem the payment application runs on) and after verifying that all laboratory requirements are met.

The PA-QSA must confirm all items in the table below, as well as:

- **Identification of location and owner of lab(s) used for the PA-DSS review**
- **Description of laboratory testing architecture and environment in place for the PA-DSS review**
- **Description of how the real-world use of the payment application was simulated in the laboratory for the PA-DSS review**

The *PA-DSS ROV Reporting Template* provides details of the laboratory validation, which must be provided for each assessment.

Laboratory Requirement	Laboratory Validation Procedure
1. Install payment application per vendor's installation instructions or training provided to customer.	**1.** Verify that the vendor's installation manual or training provided to customers was used to perform the default installation for the payment application product on all platforms listed in the PA-DSS report to simulate real-world customer experience.
2. Install and test all payment application versions listed in PA-DSS report.	**2.a** Verify that all common implementations (including region/country specific versions) of the payment application to be tested were installed.
	2.b Verify that all payment application versions and platforms were tested, including all necessary system components and dependencies
	2.c Verify that all critical payment application functionalities were tested for each version.
3. Install and implement all PCI DSS required security devices.	**3.** Verify that all security devices required by PCI DSS (for example, firewalls and anti-virus software) were implemented on test systems.
4. Install and/or configure all PCI DSS required security settings.	**4.** Verify all PCI DSS-compliant system settings, patches, etc., were implemented on test systems for operating systems, system software, and applications used by the payment application.

Laboratory Requirement	Laboratory Validation Procedure
5. Simulate real-world use of the payment application.	**5.a** The laboratory simulates the 'real world' use of the payment application, including all systems and applications where the payment application is implemented. For example, a standard implementation of a payment application might include a client/server environment within a retail storefront with a POS machine, and back office or corporate network. The laboratory simulates the total implementation.
	5.b The laboratory uses only test card numbers for the simulation/testing – live PANs are not used for testing. ***Note:*** *Test cards can usually be obtained from the vendor or a processor or acquirer.*
	5.c The laboratory runs the payment application's authorization and/or settlement functions and all output is examined per item 6 below.
	5.d The laboratory and/or processes map all output produced by the payment application for every possible scenario, whether temporary, permanent, error processing, debugging mode, log files, etc.
	5.e The laboratory and/or processes simulate and validate all functions of the payment application, to include generation of all error conditions and log entries using both simulated 'live' data and invalid data.
6. Provide capabilities for, and test using, the following penetration testing methodologies:	**6.a** Use of forensic tools/methods: Forensic tools/methods were used to search all identified output for evidence of sensitive authentication data (commercial tools, scripts, etc.), per PA-DSS Requirement 1.1.1–1.1.3.[6]
	6.b Attempt to exploit application vulnerabilities: Current vulnerabilities (for example, the OWASP Top 10, SANS CWE Top 25, CERT Secure Coding, etc.), were used to attempt to exploit the payment application(s), per PA-DSS Requirement 5.2.
	6.c Laboratory and/or processes attempted to execute arbitrary code during the payment application update process: Run the update process with arbitrary code per PA-DSS Requirement 7.2.2.

[6] Forensic tool or method: A tool or method for uncovering, analyzing and presenting forensic data, which provides a robust way to authenticate, search, and recover computer evidence rapidly and thoroughly. In the case of forensic tools or methods used by PA-QSAs, these tools or methods should accurately locate any sensitive authentication data written by the payment application. These tools may be commercial, open-source, or developed in-house by the PA-QSA.

Laboratory Requirement	Laboratory Validation Procedure
7. Use vendor's lab ONLY after verifying all requirements are met.	If use of the software vendor's lab is necessary (for example, the PA-QSA does not have the mainframe, AS400, or Tandem the payment application runs on), the PA-QSA can either (1) use equipment on loan from the vendor or (2) use the vendor's lab facilities, provided that this is detailed in the report together with the location of the tests. For either option, the PA-QSA verifies that the vendor's equipment and lab meet the following requirements:
	7.a The PA-QSA verifies that the vendor's lab meets all above requirements specified in this document and documents the details in the report.
	7.b The PA-QSA must validate the clean installation of the remote lab environment to ensure the environment truly simulates a real world situation and that the vendor has not modified or tampered with the environment in any way.
	7.c All testing is executed by the PA-QSA (the vendor cannot run tests against their own application).
	7.d All testing is either (1) performed while onsite at the vendor's premises, or (2) performed remotely via a network connection using a secure link (for example, VPN).
	7.e Use only test card numbers for the simulation/testing—do not use live PANs for testing. These test cards can usually be obtained from the vendor or a processor or acquirer.
8. Maintain an effective quality assurance (QA) process.	**8.a** PA-QSA QA personnel verify that all versions and platforms identified in the PA-DSS report were included in testing.
	8.b PA-QSA QA personnel verify that all PA-DSS requirements were tested against.
	8.c The PA-QSA QA personnel verify that PA-QSA laboratory configurations and processes meet requirements and were accurately documented in the report.
	8.d PA-QSA QA personnel verify that the report accurately presents the results of testing.