Payment Card Industry (PCI)

Marco de Seguridad del Software

Requisitos del Software Seguro y Procedimientos de Evaluación

Versión 1.2.1

Mayo de 2023

Cambios en los Documentos

Fecha	Versión	Descripción
Enero de 2019	1.0	Versión inicial
Abril de 2021	1.1	Actualización de v1.0. Consulte *el Marco de Seguridad del Software de PCI: Resumen de Cambios de los Requisitos de Software Seguro y los Procedimientos de Evaluación de la Versión 1.0 a 1.1* para obtener detalles sobre los cambios.
Septiembre de 2022	1.2	Actualización de v1.1. Consulte *el Marco de Seguridad del Software de PCI: Resumen de Cambios de los Requisitos de Software Seguro y los Procedimientos de Evaluación de la Versión 1.1 a 1.2* para obtener detalles sobre los cambios.
Mayo de 2023	1.2.1	Actualización de la v1.2 para corregir erratas.

DECLARACIONES: La versión en inglés del texto en este documento tal y como se encuentra en el sitio web de PCI SSC deberá considerar se, para todos los efectos, como la versión oficial de estos documentos y, si existe cualquier ambigüedad o inconsistencia entre este texto y el texto en inglés, el texto en inglés en dicha ubicación es el que prevalecerá.

Tabla de Contenido

Introducción

Para facilitar las transacciones de pago fiables y precisas, los sistemas y programas informáticos utilizados como parte del flujo de transacciones de pago deben diseñarse, desarrollarse y mantenerse de manera que protejan la integridad de las operaciones de pago y la confidencialidad de todos los datos confidenciales almacenados y procesados o transmitidos en asociación con transacciones de pago. Este documento, los *Requisitos de Software Seguro y los Procedimientos de Evaluación de la Industria de Tarjetas de Pago (PCI)* (en lo sucesivo, el "Estándares de Software Seguro PCI" o "este estándar") proporciona una base de requisitos de seguridad con los correspondientes procedimientos de evaluación correspondientes y una guía para la creación de software de pago seguro.

El *Estándar de Software Seguro de PCI* está diseñado para usarse como parte del Marco de Seguridad del Software (SSF) de PCI. Las entidades que deseen validar su software de pago bajo PCI SSF lo harían con este estándar.

Terminología

Se proporciona una lista de términos y definiciones aplicables en el *Glosario de Términos, Abreviaturas y Acrónimos del Marco de Seguridad de Software de PCI*, disponible en la Biblioteca de Documentos de PCI SSC: https://www.pcisecuritystandards.org/document_library.

Además, las definiciones de la terminología general de PCI se proporcionan en el Glosario PCI en el sitio web del PCI SSC en: https://www.pcisecuritystandards.org/pci_security/glossary.

Publicaciones Relacionadas

Además de los requisitos de seguridad y los procedimientos de evaluación para el software de pago definidos en este estándar, hay documentos adicionales disponibles para apoyar el uso de este estándar. Consulte las últimas versiones de (o los documentos sucesores de) las siguientes publicaciones de PCI SSC en la Biblioteca de Documentos del PCI SSC, para obtener más información:

Nombre del Documento	Descripción
Marco de Seguridad del Software de PCI: Preguntas Técnicas Frecuentes sobre el Estándar de Software Seguro ("Preguntas Técnicas Frecuentes sobre el Software Seguro, FAQS")	Las Preguntas Técnicas más Frecuentes (FAQ) proporcionan un mecanismo para abordar las cuestiones relacionadas con la interpretación y la aplicación del Estándar y el Programa PCI asociados. Las Preguntas Técnicas mas Frecuentes (FQA) se consideran "normativas" y deben considerarse completamente dentro del alcance de la actividad de evaluación del Estándar asociado.
Marco de Seguridad del Software de PCI: Estándar de Ciclo de Vida del Software Seguro de PCI ("El Estándar SLC Seguro")	Requisitos de seguridad adicionales para que las organizaciones de desarrollo de software garanticen que desarrollan y mantienen el software de forma segura durante todo el ciclo vital del software.
Marco de Seguridad del Software de PCI: Glosario de Términos, Abreviaturas y Acrónimos ("Glosario del SSF")	Describe términos importantes, abreviaturas y acrónimos utilizados a lo largo del Estándar del Software Seguro y la documentación de apoyo.
Marco de Seguridad del Software de PCI: Guía del Programa de Software Seguro ("Guía del Programa de Software Seguro")	Describe los requisitos del programa para que las entidades validen su software de pago para que cumpla con el Estándar de Software Seguro y para que su software aparezca y se mantenga en la Lista de Software de Pago Validado del PCI SSC.
Marco de Seguridad del Software de PCI: Plantilla de Software Seguro para el Informe de Validación ("Plantilla de Informes ROV de Software Seguro")	La plantilla obligatoria que los Evaluadores de SSF calificados deben utilizar para documentar los resultados de una Evaluación de Software Seguro e informar de dichos resultados al PCI SSC.
Marco de Seguridad del Software de PCI: Certificado de Validación de Software Seguro ("AOV de Software Seguro")	Un documento de plantilla proporcionado por PCI SSC que las Compañías y Proveedores de Evaluadores se Software Seguro deben usar certificar los resultados de una Evaluación de Software Seguro.
Marco de Seguridad del Software de PCI: Requisitos de Calificación para Evaluadores ("Requisitos de Calificación del SSF")	Describe la capacidad mínima y los requisitos de documentación relacionados que las Compañías Evaluadoras de SSF y sus Empleados Evaluadores deben cumplir para estar calificados para realizar Evaluaciones de Software Seguro.

Nombre del Documento	Descripción
Requisitos de Seguridad Modular del Punto de Interacción (POI) de la Seguridad de las Transacciones (PTS) con PIN PCI ("Estándar POI PTS de PCI")	Requisitos de seguridad que deben cumplir los dispositivos de aceptación de pagos para obtener la aprobación del dispositivo de Punto de Interacción (POI) de Seguridad de Transacciones con PIN (PTS) de la Industria de Tarjetas de Pago (PCI).
Acuerdo de Autorización del Vendedor ("VRA")	Establece los términos y condiciones que deben cumplir los Proveedores de Software de Pago validado para participar en los programas de PCI.

Roles y Responsabilidades de las Partes Interesadas

Existen numerosas partes interesadas involucradas en el mantenimiento y la gestión de los estándares PCI. A continuación, se describen los roles y responsabilidades de alto nivel de estas partes interesadas en relación con el Marco de Seguridad del Software PCI:

PCI SSC – Responsable de mantener los estándares, los programas de apoyo y la documentación relacionada asociada con el Marco de Seguridad del Software de PCI, incluidos, entre otros:

- Mantenimiento del Estándar de Software Seguro de PCI (este documento).
- Mantener toda la documentación de apoyo, incluidas las plantillas de reportes, los formularios de certificación, las preguntas frecuentes (FAQ) y la guía para ayudar a las entidades a implementar y evaluar este estándar.
- Proporcionar instrucciones y guía para los Evaluadores del SSF de acuerdo con los requisitos y procedimientos de evaluación de este estándar.
- Mantener una lista de todos los Evaluadores del SSF calificados para realizar evaluaciones según este estándar (en el Sitio Web de PCI SSC).
- Mantener un programa de aseguramiento de la calidad para los Evaluadores del SSF.

Marcas de Pago Participantes – Responsables de desarrollar y hacer cumplir sus respectivos programas de cumplimiento relacionados con los estándares PCI, incluidos, entre otros**:**

- Definir y hacer cumplir los requisitos, mandatos y plazos para el cumplimiento del Estándar de Software Seguro de PCI (este documento).
- Determinar las entidades que están obligadas a cumplir con esta estándar.
- Especificar los métodos de validación y la frecuencia.
- Identificar y hacer cumplir las multas o sanciones por incumplimiento.

Compañías Evaluadoras de SSF – Son responsables de mantener los obligatorios conocimientos, la experiencia y el equipo necesarios para ejecutar todas las actividades de evaluación, de cumplir todos los Requisitos de Calificación de los Evaluadores de SSF, de realizar las evaluaciones según este estándar y de generar el informe de evaluación que documente los resultados. Tenga en cuenta que no todas las Compañías Evaluadoras SSF están calificadas para realizar evaluaciones según este estándar. Para más información sobre las actividades de evaluación y los requisitos de cualificación de los evaluadores, consulte la *Guía del Programa de Software Seguro* de PCI y los *Requisitos de Calificación para Evaluadores de SSF,* respectivamente.

Vendedores/Proveedores/Desarrolladores de Software de Pago – Responsable de desarrollar, distribuir, mantener y operar (cuando corresponda) el software de pago y garantizar que su software de pago cumpla con todos los requisitos de seguridad aplicables definidos en este estándar.

Descripción General del Estándar de Software Seguro de PCI

Los requisitos de seguridad definidos en el *Estándar de Software Seguro de PCI* garantizan que el software de pago se diseñe, desarrolle y mantenga de manera que proteja las transacciones y los datos de pago, minimice las vulnerabilidades y se defienda contra ataques.

Alcance y Requisitos de Seguridad

Los requisitos de seguridad definidos en este estándar describen las características, controles, características y capacidades de seguridad que debe poseer el software de pago para proteger la integridad de las funciones de pago y la confidencialidad de los datos de pago confidenciales. Las características del software de pago que están dentro del alcance de estos requisitos incluyen, entre otras:

- Toda la funcionalidad del software de pago de extremo a extremo, que incluye:
  - Todas las funciones de pago.
  - Entradas y salidas.
  - Manejo de condiciones de error.
  - Interfaces y conexiones a otros archivos, sistemas y/o software.
  - Flujos de datos.
  - Mecanismos de seguridad, controles y contramedidas, como autenticación, autorización, validación, parametrizar, segmentación, registro, etc.
- Procesos utilizados por el vendedor, proveedor o desarrollador de software para identificar y apoyar los controles de seguridad del software.
- Guía que el vendedor, proveedor o desarrollador de software debe proporcionar a las partes interesadas y que describe:
  - Cómo implantar y utilizar el software de pago de forma segura.
  - Todas las opciones de configuración disponibles que pueden afectar a la seguridad del software de pago, incluidas las del entorno de ejecución y los componentes del sistema relacionados.
  - Cómo aplicar las actualizaciones de seguridad.
  - Cómo y dónde informar de los problemas de seguridad al vendedor, proveedor y/o desarrollador del software.

  Tenga en cuenta que se puede esperar que el vendedor, el proveedor o el desarrollador del software brinden dicha orientación incluso cuando la configuración específica:
  - No puede ser controlada por el vendedor, proveedor o desarrollador del software de pago después de que el software se instala en un entorno de producción; o

  - Son responsabilidad de la entidad que los implementa y no del vendedor, proveedor o desarrollador del software.
- Cualquier otro software, funcionalidad de software o servicio necesario para una implementación completa del software de pago, entre otros:
  - Funciones de software de código abierto y de terceros, bibliotecas, paquetes, componentes, servicios, y dependencias incrustadas o en las que se basa el software de pago para proporcionar la función prevista.
  - Características y funciones de una plataforma de apoyo o el entorno de ejecución en el que se basa el software de pago por motivos de seguridad.
  - Herramientas y funciones de terceros o personalizadas en las que se basa el software de pago para satisfacer los requisitos de seguridad de este estándar.

Módulos de Requisitos

El Estándar de Software Seguro de PCI incluye el concepto de "módulos" de requisitos, que son grupos distintos de requisitos relacionados con un tema o un tipo de software específico. Los módulos están destinados a aclarar cómo y cuándo se aplican los requisitos específicos a una determinada aplicación o función de pago.

Los requisitos de este estándar están organizados en los siguientes cuatro módulos de requisitos:

- **Requisitos Básicos ("Módulo Básico")**: Requisitos generales de seguridad que se aplican a todos los tipos de software de pago, independientemente de su función, diseño o tecnología subyacente.
- **Módulo A – Requisitos de Protección de los Datos de Tarjetahabientes ("Módulo de Protección de los Datos de Tarjetahabientes)**: Requisitos de seguridad adicionales para el software de pago que almacena, procesa o transmite los datos de tarjetahabientes.
- **Módulo B – Requisitos del Software del Terminal ("Módulo de Software del Terminal")**: Requisitos de seguridad adicionales para el software de pago diseñado específicamente para su implementación y operación en dispositivos POI aprobados por PCI.
- **Módulo C – Requisitos del Software Web ("Módulo de Software Web")**: Requisitos de seguridad adicionales para el software de pago que utiliza tecnologías, protocolos e idiomas de Internet para iniciar o apoyar transacciones de pago electrónico.

Aplicabilidad del Módulo de Requisitos

Cada módulo de requisitos incluye sus propios criterios de aplicabilidad. Se espera que el software evaluado según este estándar incluya la evaluación de todos los módulos aplicables. Como mínimo, el software de pago debe evaluarse para el Módulo Básico. Los módulos adicionales se incluyen en la evaluación cuando el software cumple con los criterios de aplicabilidad para esos módulos adicionales. Consulte la sección "Propósito y Alcance" dentro de cada módulo adicional para obtener más información sobre los criterios de aplicabilidad del módulo.

Tenga en cuenta que algunos requisitos definidos dentro de los módulos individuales son extensiones de los requisitos del Módulo Básico. Cuando se noten tales relaciones, los requisitos de los módulos deben evaluarse junto con sus requisitos "Básicos" asociados.

También tenga en cuenta que puede haber ciertos requisitos definidos dentro de un módulo que son similares a los requisitos de otros módulos o que pueden tener una aplicabilidad más amplia más allá de los módulos en los que están definidos. A menos que se indique lo contrario, tales requisitos están obligados a evaluarse solo en el contexto de ese módulo. Dicho esto, es probable que dichos requisitos se consoliden y/o se apliquen de manera más amplia en futuras actualizaciones de este estándar. Se anima a las entidades a identificar y aplicar los requisitos que puedan ser aplicables al software de pago de una entidad, independientemente de si la entidad está obligada a evaluar el módulo en el que se definen dichos requisitos.

Enfoque de los Requisitos Basado en Objetivos

El Marco de Seguridad del Software de PCI ha adoptado un enfoque "basado en objetivos" para definir los requisitos de seguridad en este estándar. El PCI SSC reconoce que no existe un enfoque de “talla única” para la seguridad del software y que los proveedores de software necesitan flexibilidad para determinar los controles del software de seguridad y las características más apropiados para abordar sus necesidades y riesgos empresariales específicos.

Un enfoque “basado en objetivos” es aquel que establece los requisitos de seguridad como un objetivo o resultado de seguridad deseado sin especificar necesariamente los métodos que se utilizarán para lograr el objetivo deseado. Este enfoque permite a las entidades implementar controles de seguridad del software basado en los riesgos identificados por el proveedor del software para una aplicación de software determinada. Para que este enfoque tenga éxito, los proveedores de software deben poseer una sólida práctica de gestión de riesgos como parte integral de su ciclo de vida de desarrollo del software (SDLC) y ser capaces de demostrar cómo los controles de seguridad implementados están apoyados por los resultados de sus prácticas de identificación y gestión de riesgos. Sin una práctica sólida de gestión de riesgos y evidencia disponible para respaldar la toma de decisiones basada en riesgos, el cumplimiento de los requisitos definidos en este estándar puede ser difícil de validar.

Requisito Frecuencia y Rigor

Dada la naturaleza del enfoque basado en objetivos del PCI SSC para los requisitos de seguridad, muchos requisitos de seguridad no especifican el nivel de rigor o la frecuencia de las actividades periódicas o recurrentes, como el período máximo en el que se debe proporcionar una actualización de seguridad para corregir las vulnerabilidades conocidas. En tales casos, el proveedor de software puede definir el nivel de rigor o frecuencia apropiado para sus necesidades comerciales. Sin embargo, el nivel de rigor o frecuencia elegido debe estar apoyado por evaluaciones de riesgos documentadas y las decisiones de gestión de riesgos resultantes. Además, el proveedor de software debe demostrar que su implementación proporciona una garantía continua de que los controles y actividades de seguridad del software son efectivos y satisfacen todos los objetivos de control relevantes.

Estructura de Requisito

Los requisitos de seguridad definidos en este estándar son los siguientes:

- **Objetivos de Control** – Los objetivos de seguridad de alto nivel que deben cumplirse. Los objetivos de control se establecen ampliamente para proporcionar a los proveedores de software la flexibilidad necesaria para determinar los mejores métodos para lograr el objetivo establecido. Independientemente de los métodos elegidos, se espera que el proveedor de software pueda producir evidencia clara e inequívoca para demostrar que los métodos elegidos son apropiados, suficientes y correctamente implementados para satisfacer el objetivo.
- **Requisitos de la Prueba** – Las actividades de evaluación que debe realizar un evaluador para determinar si se ha cumplido con un objetivo de control específico. Los requisitos de la prueba están destinados a proporcionar tanto al proveedor de software como al evaluador un entendimiento común de las tareas que se espera que lleve a cabo el evaluador durante la prueba. Los métodos específicos utilizados, los elementos evaluado y el personal entrevistado deben ser apropiados para el objetivo de control que se valida y para el software que se evalúa.
- **Orientación** – Información adicional para ayudar a los proveedores y evaluadores del software a comprender la intención de cada objetivo de control. La orientación también puede incluir las mejores prácticas que deben considerarse y los ejemplos de controles o métodos que pueden usarse para satisfacer el objetivo de control. La orientación no pretende excluir otros métodos que un proveedor del software pueda utilizar para cumplir con un objetivo de control, ni reemplaza ni modifica el objetivo de control al que se refiere.

Métodos de Prueba

Para apoyar la validación de su software según los requisitos de este estándar, se espera que los proveedores de software presenten evidencia de que han satisfecho los objetivos de control establecidos. Los requisitos de prueba identificados para cada objetivo de control describen las actividades que debe realizar el evaluador para confirmar que el software y/o el proveedor de software han cumplido los objetivos de control. Los requisitos de la prueba incluyen las siguientes actividades:

- **Evaluar**: El evaluador evalúa de forma crítica las pruebas de datos. Los ejemplos comunes incluyen documentos de diseño y arquitectura de software (electrónicos o físicos), código fuente, archivos de configuración y metadatos, datos de seguimiento de errores y otros resultados de los sistemas de desarrollo de software, y resultados de pruebas de seguridad. La elección de la evidencia que puede usarse para cumplir con un requisito de "examen" se deja deliberadamente abierta para que la determine el probador. Sin embargo, es un requisito de este estándar que el código fuente del software esté disponible para su revisión como parte de la evaluación. No es aceptable que se proporcione un informe de evaluación cuando no se haya evaluado o utilizado ningún código fuente en el proceso de realización de la prueba.
- **Entreviste**: El evaluador conversa con el personal. Los propósitos de tales entrevistas pueden incluir el determinar cómo se realiza una actividad, si una actividad se realiza tal como se define y si el personal tiene conocimientos o comprensión particulares de las políticas, procesos, responsabilidades o conceptos aplicables.

- **Prueba**: El evaluador evalúa el funcionamiento del software para analizar sus características y comportamiento en diversos escenarios. A menos que se indique lo contrario, las "pruebas" de software deben incluir pruebas funcionales utilizando herramientas y técnicas forenses. Ejemplos de estas herramientas y técnicas incluyen el uso de las pruebas de seguridad de análisis estático automatizado (SAST,), pruebas de seguridad de análisis dinámico (DAST,), pruebas de seguridad de aplicaciones interactivas (IAST,) y herramientas de análisis de composición de software (SCA,). Cuando se hace referencia explícita a las pruebas contradictorias, se deben usar herramientas y técnicas de prueba de penetración y otras herramientas y técnicas para intentar eludir los controles de seguridad del software o hacer que el software se comporte de manera no deseada.

Los elementos o procesos específicos a ser evaluados o probados, y el personal a ser entrevistado deben ser apropiados para el objetivo de control que se está validando y para la estructura organizacional, la cultura, las prácticas empresariales y los productos de software de cada entidad. Queda a discreción del evaluador determinar la idoneidad o adecuación de la evidencia proporcionada por la entidad para apoyar cada requisito de prueba. Cuando se especifican viñetas en un objetivo de control o requisito de prueba, se espera que cada viñeta sea validada como parte de la evaluación.

Al documentar los resultados de la evaluación, el evaluador identifica las actividades de comprobación realizadas y el resultado de cada actividad. Si bien se espera que el evaluador realice todos los requisitos de prueba definidos para cada objetivo de control, también es posible que un objetivo de control se valide utilizando métodos de prueba diferentes o adicionales. En tales casos, se espera que el evaluador documente por qué se utilizaron métodos de prueba alternativos y cómo esos métodos proporcionan al menos el mismo nivel de garantía que los requisitos de prueba establecidos. Además, cuando se utilizan términos como "periódico", "apropiado" y "razonable" en el requisito de la prueba, es responsabilidad del proveedor del software definir y defender sus decisiones sobre la frecuencia, solidez y madurez de los controles o procesos implementados.

Confianza en las Pruebas de Terceros

Se espera que el evaluador realice todos los requisitos de las pruebas. Sin embargo, un evaluador puede optar por confiar en las pruebas realizadas por un tercero, incluyendo el proveedor del software, para satisfacer un requisito de la prueba. El evaluador conserva la responsabilidad total de las actividades y los resultados de las pruebas, independientemente de si las pruebas las realiza el evaluador, el proveedor del software o de un tercero. Cuando el evaluador se basa en las pruebas de terceros, el evaluador deberá documentar y justificarlo siguiente:

- Cómo la evidencia proporcionada por el tercero apoya el mismo nivel de rigor que las pruebas realizadas por el evaluador, y
- Cómo el evaluador verificó que las pruebas de terceros en las que se basó el evaluador son apropiadas.

Cuando las pruebas de una entidad evaluada se utilicen con el fin de satisfacer los requisitos de las pruebas, el evaluador debe verificar primero que el proveedor de software esté calificado para el SLC seguro[1] antes de que se pueda confiar en las pruebas del proveedor de software.

[1]Consulte el Estándar del SLC Seguro de PCI y su Guía del Programa asociada para obtener mayor información sobre la calificación de Seguridad SLC.

Uso de Muestreo

Cuando proceda, el evaluador puede utilizar el muestreo como parte del proceso de prueba de acuerdo con una metodología de muestreo documentada. La metodología de muestreo del evaluador debe detallar cómo se eligen las muestras y debe proporcionarse al PCI SSC en el momento de la presentación del Informe de Validación (ROV).

La selección de muestras debe incluir una muestra representativa de todas las personas, procesos y tecnologías en el alcance de la evaluación del Software Seguro de PCI. Los tamaños de muestra deben ser lo suficientemente grandes para demostrar que la muestra refleja con precisión las características de la población más grande.

En todos los casos en que los hallazgos del evaluador se basan en una muestra representativa en lugar del conjunto completo de elementos aplicables, el evaluador debe anotar explícitamente este hecho en el ROV, detallar los elementos elegidos como muestras para la prueba y proporcionar referencias a las secciones aplicables de la metodología de muestreo del evaluador provista con el ROV. Cuando el evaluador selecciona muestras que no se alinean con la metodología de muestreo documentada del evaluador, el evaluador debe proporcionar una justificación en el ROV para cada instancia en la que se utilicen dichas muestras.

Uso de una Plataforma de Prueba

Para garantizar que las pruebas de software cumplan con este estándar, puede ser necesario que el vendedor del software proporcione una plataforma de prueba. Se considera que una plataforma de pruebas es una funcionalidad de prueba especial que está separada o ausente del código a nivel de producción. La plataforma de la prueba debe depender de la mayor cantidad posible de funciones subyacentes del nivel de producción previstas. La plataforma de prueba solo sirve para proporcionar un marco de prueba que permite que la funcionalidad del software se ejerza fuera de un entorno de implementación a nivel de producción para verificar el cumplimiento del software con este estándar. Por ejemplo, puede ser necesario conceder privilegios elevados o capacidades de acceso con el fin de proporcionar visibilidad en el tiempo de ejecución de varias facetas de la operación del software. Otros ejemplos incluyen el proporcionar una función de prueba para iniciar una transacción de prueba o para realizar las funciones de autenticación. Queda a discreción del evaluador el solicitar cualquier funcionalidad de prueba que se considere necesaria para verificar el cumplimiento del software con los requisitos aplicables de este estándar.

Requisitos Básicos

Minimizar la Superficie de Ataque

La superficie de ataque del software se minimiza. La confidencialidad y la integridad de todos los activos críticos del software están protegidas y todas las características y funciones innecesarias se eliminan o deshabilitan.

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control 1: Identificación de los Activos Críticos Todos los activos críticos del software se identifican y clasifican.		
1.1 Se identifican todos los datos confidenciales almacenados, procesados o transmitidos por el software.	**1.1.a** El evaluador deberá evaluar la evidencia para confirmar que se mantiene la información que detalla todos los datos confidenciales que son almacenados, procesados y/o transmitidos por el software. Como mínimo, esto incluirá todos los datos de pago; las credenciales de autenticación; las claves criptográficas y los datos relacionados (como los IV y los datos tipo semilla para los generadores de números aleatorios); y los datos de configuración del sistema (como las entradas de registro, las variables del entorno de la plataforma, las solicitudes de datos en texto plano del software que permite la introducción de datos PIN, o los scripts de configuración).	Los controles de seguridad del software están diseñados y son implementados para proteger la confidencialidad o integridad de los activos críticos. Para asegurarse de que estos controles son los efectivos y adecuados, el proveedor del software debe identificar todos los datos confidenciales que el software recopila, almacena, procesa o transmite, así como todas las funciones y recursos confidenciales que proporciona o utiliza.
	1.1.b El evaluador deberá evaluar la evidencia para confirmar que se mantiene la información que describe dónde se almacenan los datos confidenciales. Esto incluye el almacenamiento de datos confidenciales en almacenamiento temporal (como memoria volátil), almacenamiento semipermanente (como discos RAM), almacenamiento no volátil (como medios de almacenamiento magnéticos y flash), o en ubicaciones o factores de forma específicos (como con un sistema integrado que solo es capaz de almacenamiento local).	
	1.1.c El evaluador deberá evaluar la evidencia para confirmar que se mantiene la información que describe los controles de seguridad que se implementan para proteger los datos confidenciales.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	1.1.d El evaluador deberá probar el software para validar la evidencia obtenida en los Requisitos de Prueba 1.1.c.	
	1.1.e El evaluador deberá evaluar la evidencia y probar el software para identificar los tipos de transacciones y/o los elementos de datos de la tarjeta que son apoyados con el software, y para confirmar que los datos de todos ellos están apoyados por la evidencia evaluada en los Requisitos de Prueba 1.1.a a 1.1.c.	
	1.1.f El evaluador deberá evaluar la evidencia y probar el software para identificar las implementaciones criptográficas que están apoyadas con el software (incluida la criptografía utilizada para el almacenamiento, el transporte y la autenticación), y para confirmar que los datos criptográficos para todas estas implementaciones son apoyados por la evidencia evaluada en los Requisitos de Prueba 1.1.a a 1.1.c, y que la evidencia describe si estos son implementados por el propio software, a través del software de terceros o como funciones del entorno de ejecución.	
	1.1.g El evaluador deberá evaluar la evidencia y probar el software para identificar las cuentas y las credenciales de autenticación apoyadas por el software (incluidas las cuentas predeterminadas y las creadas por el usuario) y para confirmar que estas cuentas y credenciales están apoyadas por la evidencia evaluada en los Requisitos de Prueba. 1.1.a a 1.1.c.	
	1.1.h El evaluador deberá evaluar las pruebas y probar el software para identificar las opciones de configuración proporcionadas por el software que pueden afectar los datos confidenciales (incluidos los proporcionados a través de archivos o scripts separados, funciones internas o menús y opciones), y para confirmar que estos están apoyados por la evidencia evaluada en los Requisitos de Prueba 1.1.a a 1.1.c.	

Objetivos de Control	Requisitos de las Pruebas	Guía
1.2 Se identifican todas las funciones confidenciales y los recursos sensibles proporcionados o utilizados por el software.	**1.2.a** El evaluador deberá evaluar la evidencia para confirmar que se mantiene la información que detalla todas las funciones confidenciales y los recursos confidenciales proporcionados o utilizados por el software. Como mínimo, esto incluirá todas las funciones diseñadas para almacenar, procesar o transmitir datos confidenciales, y aquellos servicios, archivos de configuración u otra información necesaria para la operación normal y segura de dichas funciones.	
	1.2.b El evaluador deberá evaluar la evidencia para confirmar que se mantiene información que describa claramente cómo y dónde se almacenan los datos confidenciales asociados con estas funciones y recursos. Esto incluye el almacenamiento de datos confidenciales en el almacenamiento temporal (como la memoria volátil), almacenamiento semipermanente (como discos RAM) y almacenamiento no volátil (como medios de almacenamiento magnéticos y flash). El evaluador deberá confirmar que esta información está apoyada por la evidencia evaluada en el Requisito de Prueba 1.1.a a 1.1.c.	
	1.2.c Cuando las funciones confidenciales o los recursos confidenciales sean proporcionados por software o sistemas de terceros, el evaluador deberá evaluar la evidencia y probar al software para confirmar que el software sigue correctamente la guía disponible para el software de terceros.. ***Nota:*** *Por ejemplo, revisando la política de seguridad de un sistema criptográfico aprobado de PTS o FIPS140-2 o 140-3.*	

Objetivos de Control	Requisitos de las Pruebas	Guía
1.3 Se clasifican los activos críticos.	**1.3** El evaluador evaluará la evidencia de para confirmar que: • El proveedor de software define los criterios para clasificar los activos críticos de acuerdo con los requisitos de confidencialidad, integridad y resistencia para cada activo crítico. • Se mantiene un inventario de todos los activos críticos con clasificaciones apropiadas.	Los activos críticos representan los datos confidenciales, las funciones y los recursos que tienen valor empresarial y que requieren protección de confidencialidad, integridad o resiliencia. Existen numerosas técnicas de análisis que pueden utilizarse para identificar los activos críticos, como el Análisis del Impacto de la Misión (MIA ,), el Análisis de Red de la Dependencia Funcional (FDNA,) y el Análisis de la Amenaza de la Misión. Se puede encontrar información y técnicas adicionales en publicaciones tales como los apéndices de la *Publicación Especial NIST 800-160* o en otras publicaciones de organismos de estándares de la industria como EMVCo, ISO o ANSI.

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivos de Control 2: Valores Predeterminados Seguros Los privilegios, las características y la funcionalidad predeterminados están restringidos solo a los necesarios para proporcionar una configuración predeterminada segura.		
2.1 Todas las funciones expuestas por el software están habilitadas por defecto sólo cuando y donde es una parte documentada y justificada de la arquitectura del software.	**2.1.a** El evaluador deberá evaluar la evidencia y probar el software para identificarcualquier API de software u otras interfaces que se proporcionen o expongan de manera predeterminada durante la instalación, la inicialización o el primer uso. Para cada una de estas interfaces, el evaluador deberá confirmar que el proveedor ha documentado y justificado su uso como parte de la arquitectura del software. Las pruebas incluirán los métodos para revelar cualquier funcionalidad expuesta del software (como el escaneo en busca de servicios de escucha cuando corresponda). ***Nota:*** *Esto incluye funciones que se habilitan automáticamente según sea obligatorio durante el funcionamiento del software.*	El software a menudo contiene funciones (por ejemplo, servicios web, interfaz administrativa, latido de la aplicación, etc.) que son opcionales y, por lo general, muchos usuarios no las utilizan. Por lo general, esta funcionalidad no recibe la misma atención que las funciones y los servicios de software estándar o esenciales, y a menudo contiene debilidades de seguridad que los usuarios malintencionados pueden aprovechar para eludir los controles de seguridad. Para facilitar la implementación segura, la configuración predeterminada del software solo debe exponer la funcionalidad segura que se haya revisado, justificado y aprobado. Esto debería incluir la configuración por defecto para todas las API del software, protocolos, demonios, oyentes, componentes, etc. Todos los servicios, protocolos o puertos innecesarios deben deshabilitarse o eliminarse. Para obtener orientación sobre los servicios, protocolos o puertos considerados inseguros, consulte los estándares y guías del sector (por ejemplo, del NIST, ENISA, etc.).
	2.1.b El evaluador probará el software para determinar si alguna de las interfaces identificadas en el Requisito de Prueba 2.1.a depende de recursos externos para la autenticación. Cuando se dependa de tales recursos, el evaluador deberá evaluar la evidencia para confirmar que se implementan métodos para garantizar que se mantenga la autenticación adecuada y que estos métodos se incluyan en la evaluación de otros requisitos aplicables en este estándar.	
	2.1.c El evaluador deberá proba el software para determinar si alguna de las interfaces identificadas en el Requisito de Prueba 2.1.a depende de recursos externos para la protección de datos confidenciales durante la transmisión. Cuando se dependa de tales recursos, el evaluador deberá evaluar la evidencia para confirmar que se implementan métodos para garantizar que se mantenga la autenticación adecuada y que estos métodos se incluyan en la evaluación de otros requisitos aplicables en este estándar.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	2.1.d El evaluador probará el software para determinar si alguna de las interfaces identificadas en el Requisito de Prueba 2.1.a expone funciones o servicios que tienen vulnerabilidades divulgadas públicamente mediante la realización de una búsqueda de los protocolos, métodos o servicios expuestos en repositorios públicos de vulnerabilidades como el que se mantiene dentro de la Base de Datos Nacional de Vulnerabilidad.	
	2.1.e Cuando existan vulnerabilidades conocidas en las interfaces expuestas, el evaluador deberá evaluar la evidencia y probar el software para confirmar lo siguiente: • Se implementan métodos para mitigar la explotación de estas debilidades. • Se documentan los riesgos que plantea el uso de protocolos, funciones o puertos vulnerables conocidos. • Se proporciona a las partes interesadas una guía clara y suficiente sobre cómo implementar correctamente la seguridad suficiente para cumplir con los objetivos de control aplicables en este estándar de acuerdo con el Objetivo de Control 12.1. ***Nota:*** *El evaluador debe hacer referencia a la información sobre amenazas del proveedor definida en el Objetivo de Control 4.1 para este elemento.*	

Objetivos de Control	Requisitos de las Pruebas	Guía
	2.1.f El evaluador evaluará la evidencia para identificar cualquier módulo de terceros utilizado por el software y para confirmar que tales funciones expuestas por cada módulo están incapacitadas, no se puede acceder a ellas a través de métodos de mitigación implementados por el software, o están formalmente documentadas y justificadas por el vendedor del software. Cuando se impida el acceso a funciones de terceros a través de métodos de protección implementados, el evaluador deberá probar el software para confirmar que no se basa en la falta de conocimiento de dichas funciones como método de mitigación de la seguridad al simplemente no documentar una interfaz API accesible de otro modo, y para confirmar que los métodos de protección son efectivos para prevenir el uso inseguro de tales funciones de terceros.	
2.2 Todos los controles, características y funciones de seguridad del software están habilitados en la instalación, inicio o primer uso del software. ***Nota:*** *Los controles específicos de seguridad del software obligatorios para proteger la integridad y la confidencialidad de los datos confidenciales, las funciones confidenciales y los recursos confidenciales se capturan en la sección de Mecanismos de Protección de Software.*	**2.2.a** El evaluador deberá evaluar la evidencia y probará el software para identificar todos los controles de seguridad del software, las características y funciones en las que se basa el software para la protección de los activos críticos y para confirmar que todos están habilitados en el momento de la instalación, inicio o primer uso del software. **2.2.b** Cuando los controles, las características y las funciones de seguridad del software se habilitan solo después del inicio o el primer uso, el evaluador deberá probar el software para confirmar que los datos confidenciales se procesan solo después de que se complete este proceso de inicio.	Como se ha señalado anteriormente en la guía, los controles de seguridad del software se diseñan e implementan para proteger la confidencialidad e integridad de los activos críticos. Algunos ejemplos de dichos controles de seguridad del software incluyen los mecanismos de autenticación y autorización, controles criptográficos y controles para evitar la filtración de datos confidenciales. La configuración predeterminada del software debe dar lugar a una configuración de software segura y no debe depender de que el usuario final sea un experto en la materia para garantizar una configuración segura. A tal efecto, todos los controles de seguridad del software disponibles deben estar activos en la instalación, inicio o primer uso del software, dependiendo de cómo se implemente el software.

Objetivos de Control	Requisitos de las Pruebas	Guía
	2.2.c Cuando sea obligatoria la entrada o interacción del usuario para habilitar controles, características o funciones de seguridad del software (como la instalación de certificados), el evaluador deberá evaluar la evidencia para confirmar que se proporciona a las partes interesadas una guía clara y suficiente sobre la configuración de estas opciones, de acuerdo con el Objetivo de Control 12.1.	
2.3 Las credenciales o claves de autenticación predeterminadas para las cuentas integradas no se utilizan después de la instalación, inicio o primer uso.	**2.3.a** El evaluador deberá evaluar la evidencia para identificar las credenciales, claves, certificados y otros activos críticos predeterminados utilizados para la autenticación del software. ***Nota:*** *El evaluador deberá remitirse a la evidencia obtenida en las pruebas de los Objetivos de Control 1, 5 y 7 para determinar los mecanismos de autenticación y control de acceso, las claves y otros activos críticos utilizados para la autenticación.*	Para protegerse contra el acceso no autorizado, el software de pago debe evitar el uso de cuentas integradas hasta que se puedan cambiar las credenciales de autenticación predeterminadas. Las cuentas integradas con credenciales conocidas, como contraseñas predeterminadas o vacías o claves predeterminadas, a menudo se pasan por alto durante la instalación, la configuración inicial o el uso, y un usuario malintencionado puede usarlas para omitir los controles de acceso. Por lo tanto, el software no debe utilizar ni depender de las credenciales predeterminadas para su funcionamiento en el momento de la instalación, inicio o primer uso.
	2.3.b El evaluador deberá probar el software para confirmar que todas las credenciales, claves, certificados y otros activos críticos predeterminados utilizados para la autenticación del software están apoyados por la evidencia evaluada. ***Nota:*** *Se espera que este análisis incluya, pero no necesariamente se limite a, el uso de las herramientas de análisis de entropía para buscar las claves criptográficas codificadas, búsquedas de llamadas a funciones criptográficas comunes y estructuras como S-Boxes y funciones de biblioteca de números grandes (y rastrear estas funciones hacia atrás para buscar las claves codificadas), así como verificar las cadenas que contengan nombres de cuentas de usuario comunes o valores de contraseña.*	

Objetivos de Control	Requisitos de las Pruebas	Guía
	2.3.c Cuando sea obligatorio la participación o interacción del usuario para deshabilitar o cambiar cualquier credencial de autenticación o claves para las cuentas integradas, el evaluador deberá evaluar la evidencia para confirmar que se proporciona una guía sobre la configuración de estas opciones a las partes interesadas de acuerdo con el Objetivo de Control 12.1.	
	2.3.d El evaluador probará el software para confirmar que las credenciales de autenticación predeterminadas o las claves para las cuentas integradas no se utilicen por los mecanismos de autenticación y control de acceso implementados por el software después de la instalación, inicio o primer uso del software. ***Nota:*** *El evaluador debe referirse a la evidencia obtenida en la prueba del Objetivo de Control 5 para determinar los mecanismos de autenticación y control de acceso implementados por el software.*	
	2.3.e El evaluador deberá probar el software para confirmar que las claves criptográficas utilizadas para la autenticación no se utilizan para otros fines, como la protección de datos confidenciales durante el almacenamiento y la transmisión. ***Nota:*** *El evaluador debe consultar la evidencia obtenida en la prueba del Objetivo de Control 6 para determinar los controles de seguridad del software implementados para proteger los datos confidenciales.*	

Objetivos de Control	Requisitos de las Pruebas	Guía
2.4 Los privilegios y recursos solicitados por el software de su entorno de ejecución se limitan a los necesarios para el funcionamiento del software.	**2.4.a** El evaluador deberá evaluar las pruebas del proveedor para identificar los privilegios y recursos obligatorios del software y para confirmar que se mantiene la información que describe y justifica razonablemente todos los privilegios y recursos obligatorios, incluidos los permisos explícitos para acceder a los recursos, como cámaras, contactos, etc.	En muchos ataques al software o sistemas subyacentes, el software se utiliza a menudo para ejecutar las funciones en los sistemas operativos subyacentes o para abusar de los recursos externos accesibles. Cuando el software requiere permisos excesivos, dichos permisos pueden ser aprovechados por un usuario malintencionado. Para minimizar la superficie de ataque del software, este solo debe solicitar y recibir los privilegios mínimos obligatorios para su funcionamiento. Por ejemplo, las cuentas del servicio del sistema que usa el software para operar, o las cuentas que usa el software para acceder a los componentes subyacentes, como una base de datos o invocar las llamadas a los servicios web, no deben requerir permisos que excedan el mínimo necesario para que el software realice sus operaciones. El mismo concepto se aplica a los recursos utilizados por el software. El software debe tener acceso solo a los recursos mínimos obligatorios para que funcione como se espera. Por ejemplo, las aplicaciones móviles que no requieren acceso a la cámara o a las fotografías no deberían solicitar dicho acceso a menos que sean una parte necesaria de la arquitectura del software. Del mismo modo, el software no debe tener acceso a archivos confidenciales (por ejemplo, /etc/passwd) a menos que exista una necesidad legítima de que el software acceda a esos archivos.
	2.4.b Cuando no sea posible limitar el acceso debido a la arquitectura de la solución o el entorno de ejecución en el que se ejecuta el software, el evaluador deberá evaluar la evidencia para identificar todos los mecanismos implementados por el software para evitar el acceso no autorizado, la exposición o la modificación de los activos críticos y confirmar que se proporciona una guía sobre la implementación y configuración adecuada de estos mecanismos a las partes interesadas de acuerdo con el Objetivo de Control 12.1.	
	2.4.c El evaluador deberá probar el software para confirmar que los permisos y privilegios de acceso se asignan de acuerdo con la evidencia evaluada el Requisito de Prueba 2.4.a. Cuando sea posible, el evaluador utilizará las herramientas adecuadas para la plataforma en la que esté instalado el software para revisar los permisos y privilegios del propio software, así como los permisos y privilegios de cualesquiera recursos, archivos o elementos adicionales generados o cargados por el software durante su uso. ***Nota:*** *Cuando la prueba anterior no sea posible, el evaluador deberá justificar por qué es así y que la prueba que se ha realizado es suficiente.*	
	2.4.d Cuando el entorno de ejecución del software proporcione funciones heredadas para que las utilicen las versiones anteriores del software, el evaluador deberá evaluar las pruebas del proveedor y probar el software para confirmar que no se utilizan y que solo se implementan las funciones recientes y seguras. Por ejemplo, el software debe "apuntar" a las versiones más recientes de las API proporcionadas por el entorno en el que se ejecutan, cuando estén disponibles.	

Objetivos de Control	Requisitos de las Pruebas	Guía
2.5 Los privilegios predeterminados para las cuentas integradas se limitan a los necesarios para el propósito y la función previstos.	**2.5.a** El evaluador deberá evaluar la evidencia para identificar todas las cuentas predeterminadas proporcionadas por el software y confirmar que los privilegios asignados a estas cuentas están justificados y son razonables.	En apoyo al principio del "privilegio mínimo", las cuentas integradas solo deben tener los privilegios obligatorios para la función prevista de la cuenta, incluyendo el acceso a datos y recursos confidenciales, así como la capacidad de ejecutar funciones confidenciales. Por ejemplo, una cuenta de administrador integrada puede requerir la capacidad de configurar el software y las cuentas de usuario asociadas, pero no la capacidad de acceder a áreas que contienen datos confidenciales. La aplicación del principio de privilegio mínimo a las cuentas de usuario ayuda a evitar que los usuarios que no tengan conocimiento suficiente sobre el software cambien de forma incorrecta o accidental la configuración del software o su configuración de seguridad. La aplicación de privilegios mínimos también ayuda a minimizar los efectos del acceso no autorizado a las cuentas de usuario del software. Para limitar el acceso a datos, funciones y recursos confidenciales solo a aquellas cuentas que requieren dicho acceso, el nivel de privilegio y acceso obligatorio debe definirse y documentarse para cada cuenta integrada en una matriz de acceso, de modo que se puedan realizar sus funciones asignadas, pero no se conceden accesos o privilegios adicionales o innecesarios.
	2.5.b El evaluador deberá probar el software para confirmar que todas las cuentas predeterminadas proporcionadas o utilizadas por el software están apoyadas por la evidencia evaluada en el Requisito de Prueba 2.5.a.	
	2.5.c El evaluador deberá evaluar la evidencia y probar el software para confirmar que las interfaces expuestas, como las API, están protegidas frente a los intentos de usuarios no autorizados de modificar los privilegios de las cuentas y elevar los derechos de acceso de los usuarios.	

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivos de Control 3: Retención de Datos Confidenciales Se minimiza la retención de los datos confidenciales.		
3.1 El software sólo conserva los datos confidenciales absolutamente necesarios para que el software proporcione la funcionalidad prevista.	**3.1.a** El evaluador evaluará las evidencias para identificar los datos confidenciales recopilados por el software para su uso más allá de cualquier transacción, el período de tiempo predeterminado durante el cual se retienen y si el período de retención es configurable por el usuario, y para confirmar que el propósito para retener los datos confidenciales de esta manera está justificado y es razonable. ***Nota:*** *El evaluador debe referirse a la evidencia obtenida en la prueba del Objetivo de Control 1.1 para determinar los datos confidenciales retenidos por el software.*	Para evitar la divulgación no autorizada de datos confidenciales a las partes no autorizadas, el software debe conservar los datos confidenciales solo durante el tiempo necesario para realizar la operación específica para la cual se recopilan los datos confidenciales. La retención de datos confidenciales por más tiempo del obligatorio presenta la oportunidad de que los datos se manejen o usen incorrectamente o se divulguen accidentalmente. Este objetivo de control diferencia entre datos confidenciales transitorios retenidos temporalmente y datos confidenciales persistentes que se retienen de forma más permanente. Los ejemplos de datos confidenciales transitorios incluyen los datos de tarjetahabientes de retención en la memoria hasta que se recibe la autorización de pago. Los ejemplos de datos confidenciales persistentes incluyen el almacenamiento de datos de tarjetahabientes en disco para apoyar transacciones de pago recurrentes.
	3.1.b El evaluador deberá probar el software para confirmar que todas las funciones o servicios disponibles diseñados para la conservación de los datos confidenciales estén respaldados por la evidencia evaluada en el Requisito de Prueba 3.1.a. ***Nota:*** *El evaluador debe referirse la evidencia obtenida en la prueba del Objetivo de Control 1.2 para determinar las funciones y servicios confidenciales proporcionados o utilizados por el software.*	

Objetivos de Control	Requisitos de las Pruebas	Guía
	3.1.c El evaluador deberá evaluar la evidencia y probar el software para determinar si el software facilita el almacenamiento de datos confidenciales persistentes con el fin de depurar, encontrar errores o probar los sistemas, y para confirmar que dichos datos están protegidos durante el almacenamiento de acuerdo con el Objetivo de Control 6.1. Cualquier función que permita el almacenamiento de datos confidenciales para estos fines debe habilitarse explícitamente a través de una interfaz que requiere interacción y autorización por parte del usuario y retiene los datos solo durante el tiempo necesario de acuerdo con los criterios razonables del proveedor. El cierre del software debe dar lugar a la terminación de este estado de depuración, de modo que requiera una rehabilitación explícita la próxima vez que se ejecute el software, y cualquier dato confidencial se elimine de forma segura según el Objetivo de Control 3.4.	
	3.1.d Cuando sea obligatoria la participación o interacción del usuario para configurar el período de retención de datos confidenciales, el evaluador deberá evaluar la evidencia para confirmar que se proporciona a los interesados una guía sobre la configuración de estas opciones, de conformidad con el Objetivo de Control 12.1.	
3.2 Los datos confidenciales transitorios solo se conservan durante el tiempo necesario para cumplir con un propósito comercial legítimo.	**3.2.a** El evaluador evaluará las evidencias para identificar todos los datos confidenciales que retiene el software para uso transitorio, lo que desencadena la eliminación segura de estos datos, y para confirmar que los fines para retener los datos están justificados y son razonables. Esto incluye los datos que se almacenan solo en la memoria durante el funcionamiento del software. ***Nota:*** *El evaluador debe remitirse a las evidencias obtenidas en las pruebas del Objetivo de Control 1.1 para determinar los datos confidenciales transitorios retenidos temporalmente por el software.*	Los elementos de los datos confidenciales recopilados junto con operaciones de software sólo deben conservarse durante el tiempo obligatorio para completar dicha operación o transacción relacionada. Una vez completado el procesamiento del pago, todos los datos confidenciales transitorios deben eliminarse de forma segura de todas las ubicaciones en las que se hayan conservado, de manera que ningún proceso, componente, función, aplicación o usuario posterior dentro del entorno pueda acceder a los datos confidenciales o capturarlos. *(continúa en la página siguiente)*

Objetivos de Control	Requisitos de las Pruebas	Guía
	3.2.b El evaluador deberá probar el software para confirmar que todas las funciones o servicios disponibles que conservan los datos confidenciales transitorios están apoyadas por la evidencia evaluada en el Requisito de Prueba 3.2.a y no utilizan objetos inmutables. ***Nota:*** *El evaluador debe consultar la evidencia obtenida en la prueba del Objetivo de Control 1.2 para determinar las funciones y servicios confidenciales que retienen datos confidenciales transitorios.*	Los vendedores de software también deben ser conscientes y tener en cuenta cómo otros aspectos de la arquitectura del software (como el lenguaje de desarrollo del software y el entorno operativo) pueden afectar cómo y dónde se conservan los datos confidenciales transitorios. Por ejemplo, el uso del sistema operativo de particiones de intercambio o archivos de memoria virtual puede hacer que la información que debería haber sido transitoria persista más de lo previsto. Si se deben usar datos confidenciales para depurar o solucionar problemas, el software solo debe capturar la cantidad mínima de datos necesarios y almacenarlos de forma segura en una ubicación conocida.
	3.2.c El evaluador deberá evaluar la evidencia y probar el software para determinar si el software facilita el almacenamiento de datos confidenciales transitorios con el fin de depurar, encontrar errores o probar los sistemas, y para confirmar que dichos datos están protegidos de acuerdo con el Objetivo de Control. 6.1. Cualquier función que permita el almacenamiento de datos confidenciales transitorios para estos fines debe habilitarse explícitamente a través de una interfaz que requiere la interacción y autorización del usuario. El cierre del software debe dar como resultado la terminación de este estado de depuración, de modo que requiera una rehabilitación explícita la próxima vez que se ejecute el software, y cualquier dato confidencial transitorio se elimine de forma segura de acuerdo con el Objetivo de Control 3.5.	
	3.2.d Cuando la retención de datos confidenciales transitorios requiera la participación o interacción del usuario, el evaluador deberá evaluar la evidencia para confirmar que se proporciona la guía sobre la configuración de estas opciones a las partes interesadas de acuerdo con el Objetivo de Control 12.1.	

Objetivos de Control	Requisitos de las Pruebas	Guía
3.3 El software protege la confidencialidad y la integridad de los datos confidenciales (tanto transitorios como persistentes) durante la retención. ***Nota:*** *La sección de los Mecanismos de Protección del Software incluye varios controles específicos de seguridad del software que es obligatorio implementar para proteger los datos confidenciales durante su almacenamiento, procesamiento o transmisión. Estos controles de seguridad del software deben analizarse para determinar su aplicabilidad a los tipos de datos confidenciales retenidos por el software.*	**3.3.a** El evaluador deberá examinar la evidencia para identificar los métodos implementados para proteger los datos confidenciales durante el almacenamiento.	El software debe mantener controles y mecanismos de seguridad para proteger todos los datos confidenciales mientras el software los retiene. Algunos ejemplos de controles de seguridad del software incluyen escribir en una ubicación de memoria segura o usar la criptografía para hacer que los datos sean ilegibles.
	3.3.b Cuando los datos confidenciales se almacenan fuera de las variables temporales dentro del propio código, el evaluador deberá probar el software para confirmar que los datos confidenciales están protegidos mediante criptografía robusta u otros métodos que brinden un nivel de seguridad equivalente.	
	3.3.c Cuando los métodos de protección utilicen criptografía, el evaluador examinará las evidencias y probará el software para confirmar que la implementación criptográfica cumple con el Objetivo de Control 7 de este estándar.	
	3.3.d Cuando los datos confidenciales estén protegidos mediante métodos distintos a la criptografía robusta, el evaluador deberá evaluar la evidencia y probar el software para confirmar que las protecciones están presentes en todos los entornos en los que el software está diseñado para ejecutarse y se implementan correctamente.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	3.3.e Cuando sea obligatoria la participación o interacción del usuario para configurar el período de retención de datos confidenciales, el evaluador deberá evaluar la evidencia para confirmar que se proporciona a los interesados una guía sobre la configuración de estas opciones, de conformidad con el Objetivo de Control 12.1.	
3.4 El software elimina de forma segura los datos confidenciales cuando ya no sea obligatorio.	**3.4.a** El evaluador deberá evaluar la evidencia para identificar los métodos implementados para hacer que los datos confidenciales persistentes sean irrecuperables y para confirmar que los datos confidenciales se vuelven irrecuperables una vez que se completa el proceso.	La eliminación segura es el proceso de hacer que los datos sean irrecuperables para otras personas, procesos o sistemas. La eliminación segura puede ser obligatoria al final de una operación específica del software o al finalizar los requisitos de retención especificados por el usuario. En el último caso, el software debería poder eliminar de forma segura los datos confidenciales después de que expire el período de retención especificado por el usuario. Sólo en circunstancias en las que la conservación de datos confidenciales esté explícitamente permitida, los datos deberán conservarse una vez completado el procesamiento de la transacción.
	3.4.b El evaluador examinará las evidencias y probará el software para identificar cualquier problema a nivel de plataforma o de implementación que complique la eliminación segura de datos confidenciales no transitorios y para confirmar que cualquier dato confidencial no transitorio se elimina de forma segura utilizando un método que garantice que los datos sean irrecuperables. Los métodos pueden incluir (pero no se limitan necesariamente a) la sobre-escritura de los datos, la eliminación de claves criptográficas (de suficiente fuerza) que se hayan utilizado para cifrar los datos o funciones específicas de la plataforma que permiten una eliminación segura. Los métodos deben adaptarse a los problemas específicos de la plataforma, como los algoritmos de nivelación de desgaste flash o sobre-aprovisionamiento de SSD, lo cual puede complicar los métodos sencillos de sobre-escritura.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	3.4.c El evaluador probará el software utilizando herramientas forenses para identificar cualquier residuo de datos confidenciales no transitorios en el entorno de la ejecución y para confirmar que los métodos atestiguados por el proveedor del software se hayan implementado y aplicado correctamente a todos los datos confidenciales. Este análisis debe adaptarse a las estructuras de datos y los métodos utilizados para almacenar los datos confidenciales (por ejemplo, examinando los sistemas de archivos en el nivel de asignación y traduciendo los formatos de datos para identificar elementos de datos confidenciales) y cubrir todos los tipos de datos confidenciales no transitorios. ***Nota:*** *Cuando no sea posible realizar las pruebas forenses de algunos o de todos los aspectos de la plataforma, el evaluador deberá evaluar las evidencias adicionales para confirmar la eliminación segura de los datos confidenciales. Dicha evidencia puede incluir (pero no necesariamente se limita a) volcados de memoria y almacenamiento de sistemas de desarrollo, evidencia de seguimientos de memoria de sistemas emulados o evidencia de extracción física de datos realizada en el sitio por el proveedor de software.*	

Objetivos de Control	Requisitos de las Pruebas	Guía
3.5 Los datos confidenciales transitorios se eliminan de forma segura de las instalaciones de almacenamiento temporal automáticamente por el software una vez que se cumple con el propósito para el cual se conservan.	**3.5.a** El evaluador deberá examinar la evidencia para identificar los métodos implementados para hacer que los datos confidenciales transitorios sean irrecuperables y para confirmar que los datos confidenciales son irrecuperables una vez que se completa el proceso. ***Nota:*** *Esto incluye los datos que solo pueden almacenarse temporalmente en la memoria del programa o variables durante el funcionamiento del software.*	Cuando los datos confidenciales sólo se conservan temporalmente para realizar una función específica (como una operación de pago), es obligatorio disponer de mecanismos para eliminar de forma segura los datos confidenciales una vez completada la función específica. Los datos confidenciales transitorios a menudo se borran de las ubicaciones de almacenamiento temporal después de que se completa el procesamiento. Sin embargo, esos datos pueden permanecer residentes en la memoria volátil (RAM) o en otras ubicaciones de almacenamiento durante períodos más largos de lo previsto (por ejemplo, en archivos y particiones de intercambio o archivos de registro). Los proveedores del software deben tener en cuenta todas las ubicaciones donde se almacenan los datos confidenciales, independientemente de la duración prevista del almacenamiento, y asegurarse de que dichos datos se eliminen de forma segura una vez que se cumpla el propósito para el cual el software recopiló los datos.
	3.5.b El evaluador deberá evaluar la evidencia y probar el software para identificar cualquier problema de plataforma o nivel de implementación que complique el borrado de dichos datos confidenciales transitorios, como capas de abstracción entre el código y el entorno de ejecución del hardware, y para confirmar que los métodos han sido implementados para minimizar el riesgo planteado por estas complicaciones.	
	3.5.c El evaluador deberá probar el software para identificar cualquier residuo de datos confidenciales en el entorno de ejecución y para confirmar que los métodos implementados se implementen correctamente y se cumplan para todos los datos confidenciales transitorios. Este análisis debe adaptarse a las estructuras de datos y los métodos utilizados para almacenar los datos confidenciales (por ejemplo, examinando los sistemas de archivos en el nivel de asignación y traduciendo los formatos de datos para identificar elementos de datos confidenciales) y cubrir todos los tipos de datos confidenciales no transitorios. ***Nota:*** *Cuando no sea posible realizar las pruebas forenses de algunos o de todos los aspectos de la plataforma, el evaluador deberá evaluar las evidencias adicionales para confirmar la eliminación segura de los datos confidenciales. Dicha evidencia puede incluir (pero no necesariamente se limita a) volcados de memoria y almacenamiento de sistemas de desarrollo, evidencia de seguimientos de memoria de sistemas emulados o evidencia de extracción física de datos realizada en el sitio por el proveedor de software.*	

Objetivos de Control	Requisitos de las Pruebas	Guía
3.6 El software no divulga los datos confidenciales a través de canales no deseados.	**3.6.a** El evaluador deberá evaluar la evidencia para confirmar que el proveedor de software ha realizado un análisis exhaustivo para tener en cuenta todos los vectores de ataque de divulgación de datos confidenciales, incluidos, entre otros: • Mensajes de error, registros de errores o volcados de memoria. • Entornos de ejecución que pueden ser vulnerables a ataques remotos de canal lateral para exponer datos confidenciales, como ataques que explotan la temporización de la caché o la predicción de bifurcaciones dentro del procesador de la plataforma. • Almacenamiento automático o exposición de datos confidenciales por parte del entorno de ejecución subyacente, como a través de archivos de intercambio, registro de errores del sistema, ortografía del teclado y funciones de autocorrección. • Sensores o servicios proporcionados por el entorno de ejecución que pueden usarse para extraer o filtrar datos confidenciales, como mediante el uso de un acelerómetro para capturar la entrada de una frase de contraseña que se utilizará como semilla para generar una clave criptográfica, o mediante la captura de los datos confidenciales mediante el uso de cámaras o interfaces de Comunicación de Campo Cercano (NFC,).	El proveedor del software o el propio software deben aplicar las medidas proactivas para garantizar que los datos confidenciales no se "filtren" inadvertidamente. La divulgación de datos confidenciales a las partes no autorizadas a menudo se produce a través de las salidas o los canales desconocidos o no deseados. Por ejemplo: los datos confidenciales podrían divulgarse involuntariamente a través de rutinas del manejo de los errores o excepciones, los canales de registro o depuración, los servicios y / o componentes de terceros, o mediante el uso de los recursos compartidos como memoria, disco, archivos, teclados, pantallas y funciones. Deberían aplicarse mecanismos de protección, ya sean de carácter procesal o programático, para garantizar que los datos confidenciales no se divulguen accidentalmente por esos medios.
	3.6.b El evaluador deberá evaluar la evidencia, incluidos los resultados del análisis descrito en el Requisito de Prueba 3.6.a, y probar el software para confirmar que se implementan métodos para proteger contra la divulgación no intencional de datos confidenciales. Dichos métodos pueden incluir el uso de criptografía para proteger los datos, o el uso de cegamiento o enmascaramiento de operaciones criptográficas (cuando lo apoye el entorno de ejecución).	

Objetivos de Control	Requisitos de las Pruebas	Guía
	3.6.c Cuando los métodos de protección requieran la intervención o interacción del usuario, el evaluador deberá evaluar la evidencia para confirmar que se proporciona a las partes interesadas una guía sobre la configuración y el uso adecuado de dichos métodos, de acuerdo con el Objetivo de Control 12.1.	
	3.6.d El evaluador deberá probar el software para identificar cualquier residuo de datos confidenciales en el entorno de ejecución y para confirmar que los métodos de protección se implementen correctamente y que el software no exponga ni revele datos confidenciales a usuarios no autorizados.	

Mecanismos de Protección de Software

Los controles de seguridad del software se implementan para proteger la integridad y la confidencialidad de los activos críticos.

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivos de Control 4: Protección de los Activos Críticos Los activos críticos están protegidos contra los escenarios de ataque.		
4.1 Se identifican los escenarios de ataque aplicables al software. ***Nota:*** *Este objetivo de control es una extensión del Objetivo de Control 10.1. La validación de ambos objetivos de control debe realizarse al mismo tiempo.*	**4.1.a** El evaluador deberá evaluar la evidencia para confirmar que el proveedor de software ha identificado y documentado escenarios de ataque relevantes para el software.	Los proveedores del software deben evaluar el diseño de su software de pago para identificar los escenarios de ataque aplicables al software, y deben documentar los resultados de ese análisis. La documentación debe describir los diversos aspectos del código que podrían ser atacados (incluidas las tareas o acciones que los marcos de trabajo y las bibliotecas realizan en nombre del software), la dificultad de montar un ataque con éxito, las técnicas de mitigación utilizadas para protegerse contra dichos ataques y la metodología empleada para medir la probabilidad y el impacto de cada método de ataque potencial. Cuando el software se basa en controles de seguridad del entorno de ejecución, el vendedor del software debe revisar y hacer referencia a la documentación de implementación de la plataforma (como las Políticas de Seguridad para dispositivos POI aprobados por la PCI o los módulos criptográficos aprobados por FIPS140-2 o 140-3) y debe confirmar que el software y su documentación asociada se adaptan correcta y completamente a la guía de estos documentos.
	4.1.b El evaluador deberá evaluar las evidencias para determinar si se utilizó algún método o directriz específico estándar de la industria para identificar los escenarios de ataque relevantes. Cuando no se utilicen dichos estándares industriales, el evaluador confirmará que la metodología empleada proporciona una cobertura equivalente para los escenarios de ataque aplicables al software que se está evaluando.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	4.1c El evaluador deberá evaluar la evidencia para confirmar lo siguiente: • Se asigna un propietario formal del software. Puede tratarse de la función de una persona específica o un nombre específico, pero la evidencia debe mostrar claramente a una persona responsable de la seguridad del software. • Se define una metodología para medir la probabilidad y el impacto de cualquier explotación del sistema. • Se documentan los métodos y tipos de amenazas genéricos que pueden aplicarse al software. • Todos los activos críticos administrados por el sistema y todos los recursos confidenciales utilizados por el sistema están documentados. • Se documentan todos los puntos de entrada y salida de datos confidenciales, así como el modelo de autenticación y confianza aplicado a cada uno de estos puntos de entrada/salida. • Todos los flujos de datos, segmentos de red y límites de autenticación/privilegios están documentados. • Se documentan todas las IPs estáticas, dominios, URL o puertos obligatorios por el software para su funcionamiento. • Se documentan las consideraciones para los elementos de la criptografía como modos de cifrado, protección contra los ataques de tiempo, oráculos acolchados, fuerza bruta, ataques de la "tabla arco iris" y ataques de diccionario contra el dominio de entrada. • Se documentan los detalles o supuestos de implementación del entorno de ejecución, como las configuraciones de la red y las configuraciones de seguridad del sistema operativo. *(continúa en la página siguiente)*	

Objetivos de Control	Requisitos de las Pruebas	Guía
	• Se documentan las consideraciones para el entorno de ejecución del software, el tamaño de la base de instalación y las superficies de ataque que deben mitigarse. Los ejemplos de dichas superficies de ataque pueden incluir solicitudes de usuario inseguras o pilas de protocolos, o el almacenamiento de datos confidenciales después de la autorización o el uso de métodos inseguros.	
4.2 Se implementan los controles de seguridad del software para mitigar los ataques del software.	**4.2.a** El evaluador deberá evaluar la evidencia para confirmar que uno o más métodos de mitigación están definidos para cada una de las amenazas identificadas en el Requisito de Prueba 4.1.a o que se proporcione una justificación para la ausencia de mitigaciones.	Una vez que se identifican los escenarios de ataque, se debe mitigar el riesgo de que ocurran. Los proveedores del software deben definir e implementar los mecanismos para proteger el software de los ataques y reducir la probabilidad y el impacto de una ejecución exitosa. Cualquier escenario de ataque que no se mitigue lo suficiente debe estar razonablemente justificado.
	4.2.b Cuando las mitigaciones dependan de la configuración dentro del software, el evaluador deberá probar el software para confirmar que dicha configuración se aplica de forma predeterminada en la instalación, inicialización o primer uso del software.	La naturaleza exacta de los mecanismos de protección dependerá de los escenarios de ataque, la plataforma de desarrollo y los lenguajes, marcos, bibliotecas y API de desarrollo de software utilizados por el software, así como el entorno de ejecución al que está destinado el software para ser desplegado. Para minimizar la superficie de ataque del software, el software se puede desarrollar utilizando los principios del diseño seguro como defensa por capas, segmentación y aislamiento de las aplicaciones (lógicas) y respuesta adaptativa. Algunos ejemplos de los controles de seguridad del software son la validación de entradas y salidas, la autenticación, la parametrización, el escape, la segmentación, el registro, etc. Para obtener una guía sobre la implementación de técnicas y enfoques de resiliencia cibernética, consulte los estándares y guías de la industria, como la *Publicación Especial NIST 800-160 .*
	4.2.c Cuando la entrada o interacción del usuario pueda deshabilitar, eliminar o evitar tales mitigaciones, el evaluador deberá evaluar la evidencia y probar el software para confirmar que dicha acción requiere autenticación y autorización y que se proporciona una guía sobre el riesgo de tales acciones a las partes interesadas. de acuerdo con el Objetivo de Control 12.1.	
	4.2.d Cuando las mitigaciones dependan de las características del entorno de ejecución, el evaluador deberá evaluar la evidencia para confirmar que se proporcione una guía a las partes interesadas sobre cómo habilitar dichos entornos de acuerdo con el Objetivo de Control 12.1.	
	4.2.e Cuando el entorno de ejecución proporcione API para consultar el estado de los controles de mitigación, el evaluador deberá probar el software para confirmar que las verificaciones del software para estas mitigaciones estén implementadas y activas antes del lanzamiento y periódicamente durante la ejecución.	

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control 5: Autenticación y Control de Acceso El software implementa métodos robustos de autenticación y control de acceso para proteger la confidencialidad, la integridad y la resiliencia de los activos críticos..		
5.1 El acceso a los activos críticos está autenticado.	**5.1.a** El evaluador deberá evaluar la evidencia para confirmar que los requisitos de autenticación están definidos (es decir, el tipo y número de factores) para todos los roles basados en la clasificación de activos críticos, el tipo de acceso (por ejemplo, local, sin consola, remoto) y el nivel de privilegio. ***Nota:*** *El evaluador debe remitirse a la evidencia obtenida en la prueba del Objetivo de Control 1.3 para determinar las clasificaciones de todos los activos críticos.* **5.1.b** El evaluador deberá evaluar la evidencia y probar el software para confirmar que el acceso a los activos críticos está autentificado y que los mecanismos de autenticación están implementados correctamente. **5.1.c** Cuando el software recomiende, sugiera, dependa o apoye de otro modo el uso de mecanismos externos (como VPN de terceros, funciones de escritorio remoto, etc.) para proporcionar un acceso seguro no desde la consola al sistema en el que se ejecuta el software o directamente al software en sí, el evaluador deberá evaluar la evidencia para confirmar que se proporciona orientación sobre cómo configurar correctamente los mecanismos de autenticación a las partes interesadas de acuerdo con el Objetivo de Control 12.1.	La autenticación segura garantiza la responsabilidad individual de las acciones y permite que el software mantenga una pista de la auditoría efectiva de la actividad del usuario. Esto agiliza la resolución de problemas y la contención cuando se produce un uso indebido o malintencionado. Los mecanismos de autenticación deben cubrir todos los recursos no públicos administrados o accesibles a través del software, así como las funciones confidenciales que pueden alterar el funcionamiento del software o afectar la seguridad de los datos confidenciales y los recursos confidenciales. Los ejemplos de los métodos de autenticación incluyen: • Algo que usted conoce, como una contraseña o frase de tipo contraseña. • Algo que usted tiene, como un dispositivo *token* o una tarjeta inteligente. • Algo que es suyo, como su biometría Para garantizar que los mecanismos de autenticación implementados sean adecuados para abordar el riesgo de acceso no autorizado a datos confidenciales o recursos confidenciales o al uso indebido de una función confidencial, el proveedor deberá analizar las amenazas e identificar el nivel de autenticación obligatorio para todos los tipos de usuarios y funciones. Por ejemplo, a un usuario con acceso limitado a datos confidenciales y recursos confidenciales podría obligársele a realizar la autenticación utilizando un único factor de autenticación (por ejemplo, una contraseña o una frase de paso), mientras que a un usuario capaz de exportar toda la base de datos podría obligársele una autenticación multifactor.

Objetivos de Control	Requisitos de las Pruebas	Guía
	5.1.d El evaluador deberá evaluar las evidencias para confirmar que los datos confidenciales asociados a las credenciales de autenticación, incluidas las claves públicas, se identifican como un activo crítico.	Otros factores como el tipo de acceso (por ejemplo, local, fuera de la consola o acceso remoto) y el nivel de privilegio (por ejemplo, la capacidad de invocar funciones confidenciales como pausar el registro o cambiar los privilegios de acceso) pueden influir en el nivel de autenticación que debe ser obligatorio.
5.2 El acceso a activos críticos requiere una identificación única.	**5.2.a** El evaluador deberá evaluar la evidencia y probar el software para confirmar que todos los métodos de autenticación implementados requieren una identificación única.	El software no debe requerir el uso de cuentas grupales, compartidas o genéricas. El uso de cuentas grupales o compartidas hace que sea más difícil determinar qué personas ejecutan acciones específicas, ya que una acción determinada podría haberla realizado cualquier persona que tenga conocimiento de las credenciales de autenticación de las cuentas compartidas o grupales.
	5.2.b Cuando las interfaces, como las API, permitan el acceso automatizado a los activos críticos, el evaluador deberá evaluar la evidencia y probar el software para confirmar que es obligatoria la identificación única de los diferentes programas o sistemas que acceden a los activos críticos (por ejemplo, mediante el uso de múltiples claves públicas) y que se proporcione una guía a las partes interesadas sobre la configuración de una credencial única para cada programa o sistema de acuerdo con el Objetivo de Control 12.1.	
	5.2.c Cuando la identificación se suministre a través de una interfaz que no sea de consola, el evaluador deberá probar el software para confirmar que las credenciales de autenticación están protegidas contra ataques que intentan interceptarlas en tránsito.	
	5.2.d El evaluador deberá evaluar la evidencia para confirmar que la orientación brindada a las partes interesadas según el Objetivo de Control 12.1 señala específicamente que los parámetros de identificación y autenticación no deben compartirse entre individuos, programas o de ninguna manera que impida la identificación única de cada acceso a un activo crítico.	

Objetivos de Control	Requisitos de las Pruebas	Guía
5.3 Los métodos de autenticación (incluidas las credenciales de sesión) son lo suficientemente fuertes y robustos como para proteger las credenciales de autenticación de ser falsificadas, suplantadas, filtradas, adivinadas o eludidas.	**5.3.a** El evaluador deberá evaluar la evidencia para confirmar que los métodos de autenticación implementados por el software se evalúan para identificar vulnerabilidades conocidas o métodos de ataque que involucren el método de autenticación y cómo la implementación de estos métodos mitiga contra tales ataques. El evaluador también deberá confirmar que la evidencia evaluada demuestra que se consideró la implementación utilizada en el software. Por ejemplo, una huella digital puede ser identificable de forma única para un individuo, pero la capacidad de falsificar o de otro modo eludir dicha tecnología puede depender en gran medida de la forma en que se implemente la solución. ***Nota:*** *El evaluador debe remitirse a la evidencia obtenida en la prueba del Objetivo de Control 4.1 para determinar los escenarios de ataque aplicables al software.*	El proveedor del software debe evaluar, documentar y justificar el uso de los métodos de autenticación implementados para demostrar que son lo suficientemente fuertes como para proteger las credenciales de autenticación en el caso de un uso específico previsto del software o en el escenario de implementación. Por ejemplo, si el software utiliza autenticación biométrica, es posible que el proveedor pueda identificar todos los puntos en los que un usuario malintencionado puede atacar el autenticador e implementar mitigaciones para abordar esos riesgos. El mecanismo de autenticación implementado en el software podría depender de sensores adicionales para garantizar que la muestra biométrica proporcionada sea la de un ser humano vivo y no una muestra falsificada. En algunos casos de uso o escenarios de implementación, un mecanismo de autenticación que se basa en un único método de autenticación puede que no sea suficiente. En tales circunstancias, el proveedor de software puede querer implementar estrategias de mitigación adicionales (por ejemplo, un mecanismo de autenticación de múltiples factores). Para apoyar la afirmación de que el mecanismo de autenticación implementado es lo suficientemente fuerte y robusto, el proveedor debe adoptar una metodología aceptada por la industria para asignar niveles de garantía (por ejemplo, *NIST SP800-63-3 y NIST SP800-63B.*
	5.3.b El evaluador deberá evaluar la evidencia para confirmar que los métodos de autenticación implementados son robustos y que la robustez de los métodos de autenticación se evaluó utilizando métodos aceptados por la industria. ***Nota:*** *La evaluación del proveedor y la justificación de la robustez incluyen la consideración de la ruta completa de las credenciales del usuario, desde cualquier fuente de entrada (como una interfaz hombre máquina u otro programa), a través de la transición al entorno de ejecución del software (incluyendo cualquier transmisión conmutada o de red y el cruce a través de la pila de software del entorno de ejecución antes de que el propio software lo procese).*	
	5.3.c El evaluador deberá probar el software para confirmar que los métodos de autenticación se implementen correctamente y no expongan vulnerabilidades.	

Objetivos de Control	Requisitos de las Pruebas	Guía
5.4 De forma predeterminada, todo acceso a los activos críticos está restringido solo a aquellas cuentas y servicios que requieren dicho acceso.	**5.4.a** El evaluador deberá evaluar las evidencias para confirmar que se mantiene la información que identifica y justifica el acceso obligatorio para todos los activos críticos.	Para garantizar que el software protege la confidencialidad e integridad de los activos críticos, los privilegios de acceso a dichos activos críticos deben restringirse en función de los requisitos de acceso definidos por el proveedor. Existen varios enfoques para implementar la restricción de los privilegios, tales como la administración de privilegios basada en la confianza, la restricción de uso basada en atributos y los privilegios dinámicos. Para reducir la superficie de ataque del software, los mecanismos de autorización del software podrían limitar el acceso a los activos críticos sólo a aquellas cuentas que necesiten dicho acceso (el principio del "privilegio mínimo"). Otras técnicas incluyen la implementación del Control de Acceso Basado en Roles (RBAC,), el control de acceso basado en atributos (ABAC,), el ajuste de privilegios basado en el tiempo y la revocación dinámica de la autorización de acceso.
	5.4.b El evaluador deberá evaluar la evidencia y probar el software para identificar el nivel de acceso que se brinda a los activos críticos y para confirmar que dicho acceso se correlaciona con la evidencia evaluada en el Requisito de Prueba 5.4.a. Las pruebas para confirmar que el acceso a los activos críticos está correctamente restringido deben incluir los intentos de acceso a los activos críticos a través de la cuentas de usuario, roles o servicios que no deben tener los privilegios obligatorios.	

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control 6: Protección de Datos Confidenciales Los datos confidenciales están protegidos en reposo y en tránsito.		
6.1 Los datos confidenciales están protegidos en cualquier lugar donde se almacenen.	**6.1.a** El evaluador deberá evaluar las evidencias para confirmar que los requisitos de protección de todos los datos confidenciales están definidos, incluidos los requisitos para hacer ilegibles los datos confidenciales con consideraciones de confidencialidad en cualquier lugar en el que se almacenen de forma persistente.	Los datos confidenciales deben protegerse dondequiera que se almacenen. En algunos casos, la integridad puede ser la principal preocupación. En otros casos, puede ser la confidencialidad de los datos confidenciales lo que debe protegerse. A veces, tanto la integridad como la confidencialidad deben garantizarse. El tipo de datos y el propósito para el que se generan a menudo determinarán la necesidad de protección de la integridad o confidencialidad. En todos los casos, estos requisitos de protección deben estar claramente definidos. En los casos en que la confidencialidad de los datos confidenciales sea una preocupación, es imperativo saber dónde y durante cuánto tiempo se conservan los datos. El proveedor debe tener detalles de todas las ubicaciones donde el software puede almacenar datos confidenciales, incluso en cualquier software o sistema subyacente, y documentación que detalle los controles de seguridad utilizados para proteger los datos. Los datos confidenciales que requieren protección de confidencialidad, cuando se almacenan de forma persistente, deben protegerse para evitar el acceso malicioso o accidental. Los ejemplos de métodos para hacer que los datos confidenciales sean ilegibles incluyen el uso de un hash unidireccional o el uso de criptografía robusta con procesos de administración de claves asociados. *(continúa en la página siguiente)*
	6.1.b El evaluador deberá evaluar la evidencia y probar el software para confirmar que se implementan los controles de seguridad para proteger los datos confidenciales durante el almacenamiento y que se abordan todos los requisitos de protección definidos y los escenarios de ataque identificados. ***Nota:*** *El evaluador debe remitirse a la evidencia obtenida en la prueba del Objetivo de Control 1.1 para determinar todos los datos confidenciales retenidos por el software, y del Objetivo de Control 4.1 para identificar todos los escenarios de ataque aplicables al software.*	
	6.1.c Cuando se utilice la criptografía para asegurar datos confidenciales, el evaluador deberá evaluar la evidencia y probar el software para confirmar que los métodos que implementan la criptografía para asegurar los datos confidenciales cumplen con el Objetivo de Control 7.	
	6.1.d Cuando se utilicen tokens de índice para proteger datos confidenciales, el evaluador deberá evaluar la evidencia y probar el software para confirmar que se generan de una manera que asegure que no haya correlación entre el valor y los datos confidenciales a los que se hace referencia (sin acceso al software para realizar la correlación como parte de una característica formalmente definida y evaluada de ese software, como la "detokenización").	

Objetivos de Control	Requisitos de las Pruebas	Guía
	6.1.e Cuando los métodos de protección se basen en las propiedades de seguridad del entorno de ejecución, el evaluador deberá evaluar la evidencia y probar el software para confirmar que estas propiedades de seguridad son válidas para todas las plataformas en las que se pretende implantar el software.	Cuando la integridad de los datos confidenciales se convierte en una preocupación, la criptografía robusta con prácticas apropiadas de gestión de claves es un método que podría utilizarse para satisfacer los requisitos de protección de la integridad durante el almacenamiento.
	6.1.f Cuando los métodos de protección se basen en las propiedades de seguridad del software de terceros, el evaluador deberá evaluar la evidencia y probar el software para confirmar que no existen vulnerabilidades o problemas no mitigados con el software que proporciona las propiedades de seguridad.	
6.2 Los datos confidenciales están protegidos durante la transmisión.	**6.2.a** El evaluador deberá evaluar la evidencia para identificar las ubicaciones dentro del software donde los datos confidenciales se transmiten fuera del entorno de ejecución física y para confirmar que se definen los requisitos de protección para la transmisión de todos los datos confidenciales.	Para evitar que personas malintencionadas intercepten o desvíen los datos confidenciales durante el tránsito, debe estar protegido durante la transmisión. Un método para proteger los datos confidenciales en tránsito es cifrarlos utilizando una criptografía robusta antes de transmitirlos. Como alternativa, el software podría establecer un canal autenticado y cifrado utilizando únicamente claves y certificados de confianza (para la autenticación) y una fuerza de cifrado adecuada para los protocolos seleccionados.
	6.2.b El evaluador deberá evaluar la evidencia y probar el software para confirmar que para cada uno de los métodos de ingreso y egreso que permiten la transmisión de datos confidenciales fuera del entorno de ejecución física, los datos se cifran con criptografía reforzada antes de la transmisión o se transmiten a través de un canal cifrado usando criptografía robusta. ***Nota:*** *El evaluador debe consultar la evidencia obtenida en la prueba del Objetivo de Control 1.1 para determinar los datos confidenciales almacenados, procesados o transmitidos por el software.*	
	6.2.c Cuando se dependa de características de terceros o del entorno de ejecución para la seguridad de los datos transmitidos, el evaluador evaluará la evidencia para confirmar que se proporciona a los interesados una guía sobre cómo configurar dichas características de acuerdo con el Objetivo de Control 12.1.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	6.2.d Cuando se utilice el cifrado de la capa de transporte para asegurar la transmisión de datos confidenciales, el evaluador deberá evaluar el software para confirmar que todos los métodos de entrada y salida imponen una versión segura del protocolo con autenticación del punto final antes de la transmisión de esos datos confidenciales.	
	6.2.e Cuando los métodos implementados para encriptar datos confidenciales permitan el uso de diferentes tipos de criptografía o diferentes niveles de seguridad, el evaluador deberá probar el software, incluyendo la captura de transmisiones del software, para confirmar que el software hace cumplir el uso de la criptografía robusta en todos los casos de transmisión.	
6.3 El uso de la criptografía cumple con todos los requisitos de la criptografía aplicables dentro de este estándar.	**6.3.a** Cuando se confíe en la criptografía (total o parcialmente) para la seguridad de los activos críticos, el evaluador deberá evaluar la evidencia y probar el software para confirmar que el uso de la criptografía cumple con el Objetivo de Control 7. ***Nota:*** *El evaluador debe remitirse al Objetivo de Control 7 para identificar todos los requisitos para la implementación adecuada y correcta de la criptografía.*	Siempre que se utilice la criptografía para cumplir los requisitos de seguridad del software de este estándar, debe realizarse de acuerdo con los requisitos de seguridad específicos relacionados con el uso de la criptografía (incluidos los del Objetivo de Control 7). Por ejemplo, almacenar una clave criptográfica (utilizada para proteger datos confidenciales) en un archivo de texto sin formato no se consideraría como una seguridad suficiente a menos que haya controles adicionales para evitar que el archivo que contiene la clave criptográfica sea accedido o modificado por, o expuesto a, partes no autorizadas. En las versiones actuales de la norma *NIST SP 800-175* o en otras guías industriales relacionadas con la ISO o el ANSI se pueden encontrar más guías sobre los usos adecuados de los algoritmos criptográficos.
	6.3.b Cuando se dependa de métodos criptográficos proporcionados por software de terceros o de aspectos del entorno de ejecución o de la plataforma en la que se ejecuta la aplicación para la protección de datos confidenciales, el evaluador deberá evaluar la evidencia y probar el software para confirmar que se proporciona a las partes interesadas una guía sobre la configuración de estos métodos durante la instalación, inicialización o primer uso del software, de acuerdo con el Objetivo de Control 12.1.	
	6.3.c Cuando se utilice criptografía asimétrica como RSA o ECC para proteger la confidencialidad de los datos confidenciales, el evaluador deberá evaluar la evidencia y probará el software para confirmar que las claves privadas no se utilizan para brindarle protección de confidencialidad para los datos.	

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control 7: Uso de la Criptografía La criptografía se usa de forma adecuada y correcta.		
7.1 Se utilizan algoritmos y métodos criptográficos estándar de la industria para proteger los activos críticos. Los algoritmos y métodos criptográficos estándar de la industria son los reconocidos por los organismos de normalización aceptados por la industria, como el NIST, ANSI, ISO y EMVCo. No se utilizan algoritmos y parámetros criptográficos que se sabe que son vulnerables.	**7.1.a** El evaluador deberá evaluar la evidencia para determinar cómo se utiliza la criptografía para la protección de activos críticos y para confirmar que: • Se utilizan los algoritmos criptográficos y los modos de funcionamiento estándar de la industria . • El uso de cualquier otro algoritmo se realiza en conjunción con los algoritmos estándar de la industria. • La implementación de algoritmos no estándar no reduce la fuerza de la clave criptográfica equivalente proporcionada por los algoritmos estándar de la industria.	No todos los algoritmos criptográficos son suficientes para proteger los datos confidenciales. Es un principio bien establecido en la seguridad del software el utilizar solo las implementaciones criptográficas reconocidas basadas en los estándares actuales aceptados por la industria, como los de los organismos de la industria como NIST, ANSI, ISO y EMVCo. El uso de implementaciones criptográficas propietarias puede aumentar el riesgo de que los datos se vean comprometidos, ya que las implementaciones propietarias no suelen someterse al mismo nivel de pruebas al que se han sometido las implementaciones aceptadas por la industria. Solo deben utilizarse aquellas implementaciones que hayan sido sometidas a suficientes pruebas (por ejemplo, por el NIST, el ANSI u otros organismos industriales reconocidos).
	7.1.b El evaluador examinará la evidencia, incluida la información sobre la amenaza del vendedor obtenida en el Requisito de Prueba 4.1.a, y probará el software para confirmar que: • En el software sólo se utilizan algoritmos criptográficos y modos de funcionamiento documentados, y • Se implementan métodos de protección para mitigar ataques comunes a las implementaciones criptográficas (por ejemplo, el uso del software como oráculo de descifrado, ataques de fuerza bruta o de diccionario contra el dominio de entrada de los datos confidenciales, la reutilización de parámetros de seguridad como los IV, o el recifrado de múltiples conjuntos de datos utilizando valores clave aplicados linealmente, como valores clave XOR en cifradores de flujo o libretas de un solo uso).	
	7.1.c Cuando las implementaciones criptográficas requieran un valor único por operación o sesión de cifrado, el evaluador deberá evaluar la evidencia y probar el software para confirmar que las implementaciones criptográficas no exponen vulnerabilidades. Por ejemplo, esto puede incluir el uso de un IV único para un modo de operación de cifrado de flujo, un valor "k" único y aleatorio para una firma digital.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	7.1.d Cuando se utilice relleno antes o durante el cifrado, el evaluador deberá evaluar la evidencia y probar el software para confirmar que la operación de cifrado incorpora siempre un método de relleno estándar aceptado por la industria.	
	7.1.e Cuando se utilicen funciones hash para proteger datos confidenciales, el evaluador deberá evaluar la evidencia y probar el software para confirmar que: • Sólo se utilizan para este fin algoritmos y métodos hash aprobados y resistentes a las colisiones, y • Se utiliza un valor de asalto de fuerza apropiada que se genera utilizando un generador de números aleatorios seguro para garantizar que el hash resultante tiene suficiente entropía. ***Nota:*** *El evaluador deberá consultar el Objetivo de Control 7.3 para obtener más información sobre los generadores de números aleatorios seguros.*	
7.2 El software apoya los procesos y procedimientos de gerencia de claves estándar de la industria. Los procesos y procedimientos estándares de gestión claves de la industria son los reconocidos por los organismos de estándares de la industria, como NIST, ANSI e ISO.	**7.2.a** El evaluador deberá la evaluar la evidencia para confirmar que se mantiene la información que describe lo siguiente para cada clave especificada en el inventario: • Etiqueta o nombre de la clave • Ubicación de la clave • Fecha de entrada en vigor • Fecha de expiración • Propósito/tipo de clave Método/algoritmo de generación de clave utilizado • Longitud de la clave	Ya sea que se implemente dentro o fuera del software, la manera en que se administran las claves criptográficas es una parte fundamental de la seguridad continua del software de pago y los datos confidenciales que maneja. Aunque los procesos de gestión de claves criptográficas se implementan a menudo como procedimientos operativos, el software debe apoyar prácticas seguras de gestión de claves basadas en los estándares de la industria o en las mejores prácticas. *(continúa en la página siguiente)*

Objetivos de Control	Requisitos de las Pruebas	Guía
	7.2.b El evaluador deberá evaluar la evidencia y probar el software para validar la evidencia evaluada en el Requisito de Prueba 7.2.a y para confirmar que: • Todas las claves criptográficas utilizadas para proporcionar seguridad a activos críticos (confidencialidad, integridad y autenticidad) y otros servicios de seguridad del software tienen un propósito único, y ninguna clave se utiliza para las operaciones de cifrado y autenticación. • Todas las claves tienen métodos de generación definidos, y no se comparten claves criptográficas secretas o privadas en las que se basa la seguridad de los activos críticos entre las instancias del software, excepto cuando se utiliza una clave secreta o privada común para asegurar el almacenamiento de otras claves criptográficas que se generan durante el proceso de instalación, inicio o primer uso del software (por ejemplo, criptografía de la caja blanca). • Todas las claves criptográficas tienen una fuerza de bits equivalente de al menos 128 bits de acuerdo con los estándares de la industria. • Todas las claves tienen un período de cifrado definido alineado con los estándares de la industria, y se implementan métodos para retirar y/o actualizar cada clave al final del periodo de encriptación definido.	Las prácticas de gestión de claves estándar de la industria deben aplicarse a lo siguiente: • Generación de claves criptográficas fuertes • Distribución segura de claves criptográficas. • Almacenamiento seguro de claves criptográficas. • Cambios de clave criptográfica para las claves que han llegado al final de su criptoperiodo. • El retiro o la sustitución de claves. • Aplicación del conocimiento dividido y del doble control (cuando el software admite operaciones manuales de gestión de claves criptográficas en texto claro). • Prevención de sustitución no autorizada de claves criptográficas. • La implementación de un mecanismo para hacer irrecuperable cualquier material de clave criptográfica o criptograma almacenado por el software de pago. Este requisito se aplica a las claves utilizadas para cifrar datos confidenciales y a cualquier clave de cifrado correspondiente.

Objetivos de Control	Requisitos de las Pruebas	Guía
	• La integridad y confidencialidad de todas las claves criptográficas secretas y privadas administradas por el software están protegidas cuando se almacenan (por ejemplo, cifradas con una clave de cifrado de claves que es al menos tan robusta como la clave de cifrado de datos y se almacena por separado de la clave de cifrado de datos, o como al menos dos componentes de clave de longitud completa o claves compartidas, de acuerdo con un método aceptado por la industria). • Todas las claves tienen un proceso de generación o inyección definido, y este proceso asegura una entropía suficiente para la clave. • Todas las funciones de generación de claves deben implementar funciones unidireccionales u otros procesos de generación de claves irreversibles, y no se utilizan modos de cálculo de claves reversibles (como las variantes de claves) para crear directamente nuevas claves a partir de una clave existente.	
	7.2.c Cuando se utilice criptografía para proteger una clave, el evaluador deberá evaluar la evidencia y probar el software para confirmar que no se proporciona seguridad a ninguna clave mediante una clave de menor fuerza (por ejemplo, cifrando una clave AES de 256 bits con una clave AES de 128 bits).	
	7.2.d Cuando el sistema utilice claves públicas, el evaluador deberá evaluar la evidencia y probar el software para confirmar que se preserva la autenticidad de todas las claves públicas.	
	7.2.e Cuando las claves públicas o de caja blanca no sean únicas por instanciación del software, el evaluador deberá evaluar la evidencia para confirmar que existen métodos y procedimientos para revocar y/o sustituir dichas claves (o pares de claves).	

Objetivos de Control	Requisitos de las Pruebas	Guía
	7.2.f Cuando el software dependa de archivos externos u otros elementos de datos para el material clave, como para los certificados TLS públicos, el evaluador deberá evaluar la evidencia para confirmar que se proporciona a las partes interesadas una guía sobre cómo instalar dicho material clave, incluyendo detalles que señalen cualquier requisito de seguridad para dicho material clave, de acuerdo con el Objetivo de Control 12.1.	
	7.2.g Cuando las claves públicas se carguen manualmente o se utilicen como claves raíz, el evaluador deberá evaluar la evidencia y probar el software para confirmar que las claves se instalan y almacenan de manera que se proporcione un doble control (a un nivel que sea viable en el entorno de ejecución), evitando que un solo usuario reemplace una clave para facilitar un ataque de intermediario o permitir el descifrado no autorizado de los datos almacenados. Cuando no sea posible un doble control completo (por ejemplo, debido a una limitación del entorno de ejecución), el evaluador confirmará que los métodos aplicados sean los adecuados para proteger las claves públicas.	
	7.2.h El evaluador evaluará las evidencias para confirmar que las claves secretas y/o privadas se gestionan de forma que se garantice el conocimiento dividido sobre la clave a un nivel que sea factible dada la plataforma en la que se ejecuta el software. Cuando el conocimiento dividido absolutamente no sea factible, el evaluador deberá confirmar que los métodos implementados son razonables para proteger secretos y / o claves privadas.	

Objetivos de Control	Requisitos de las Pruebas	Guía
7.3 Todos los números aleatorios utilizados por el software se generan utilizando únicamente algoritmos o bibliotecas de generación de números aleatorios (RNG) estándar del sector. Los algoritmos o bibliotecas RNG estándar de la industria son aquellos que cumplen con los estándares de la industria para una suficiente imprevisibilidad (por ejemplo, *Publicación Especial NIST 800-22*).	**7.3.a** El evaluador deberá evaluar la evidencia y probar el software para identificar todos los generadores de números aleatorios utilizados por el software y para confirmar que todos los métodos de generación de números aleatorios: • Utilice al menos 128 bits de entropía antes de la salida de cualquier número aleatorio. • Asegúrese de que no es posible que el sistema proporcione o produzca una entropía reducida al arrancar o al entrar en otros estados predecibles del sistema.	Los números aleatorios se utilizan a menudo con la criptografía para proteger la información confidencial. Las claves de encriptación y los valores de inicialización (semillas) son ejemplos de implementaciones en las que son obligatorios números aleatorios. Diseñar e implementar un generador de números aleatorios seguro no es una tarea trivial. Los proveedores de software están obligados a utilizar únicamente algoritmos y bibliotecas de generación de números aleatorios aprobados o a proporcionar pruebas que ilustren cómo se han probado los algoritmos y bibliotecas de generación de números aleatorios para confirmar que los números aleatorios generados son suficientemente impredecibles. La implementación puede basarse en una biblioteca o módulo criptográfico validado. El proveedor de software debe tener una buena comprensión de la instalación, inicio, configuración y uso por ejemplo, la siembra inicial de la función aleatoria de los mecanismos de RNG para garantizar que la implementación pueda cumplir con la seguridad efectiva obligatoria para el uso previsto.
	7.3.b Cuando se utilicen programas, plataformas o bibliotecas de terceros para la totalidad o parte del proceso de generación de números aleatorios, el evaluador deberá evaluar la evidencia (como la bibliografía actual disponible públicamente) para confirmar que los programas de terceros no exponen ninguna vulnerabilidad que pueda comprometer su uso para generar valores aleatorios.	
	7.3.c Cuando el proveedor del software se basa en una evaluación previa del generador de números aleatorios o de la fuente de entropía inicial, el evaluador deberá evaluar la evidencia (tal como los registros de aprobación de la evaluación previa) para confirmar que este esquema y la aprobación específica incluyen las áreas correctas del software en el alcance de su evaluación, y que las afirmaciones del vendedor no exceden el alcance de la evaluación o aprobación de dicho software. Por ejemplo, algunas implementaciones criptográficas aprobadas bajo FIPS 140-2 o 140-3 requieren de la siembra de una fuente de entropía externa para producir correctamente datos aleatorios.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	7.3.d Cuando el proveedor del software no se base en una evaluación previa del generador de números aleatorios o la fuente de entropía inicial, el evaluador probará el software para obtener 128 MB de salida de datos de cada generador de números aleatorios implementado en el sistema para confirmar la falta de correlación estadística en la salida. Estos datos pueden ser generados por el evaluador directamente o suministrados por el proveedor, pero el evaluador debe confirmar que el método de generación implementado asegura que los datos se produzcan como los produciría el software durante el funcionamiento normal. ***Nota:*** *El evaluador puede utilizar el conjunto de pruebas estadísticas del NIST para identificar la correlación estadística en la implementación de generación de números aleatorios.*	
7.4 Los valores aleatorios tienen la entropía que cumple con los requisitos mínimos de fuerza efectiva de las primitivas criptográficas y claves que dependen de estos.	**7.4.a** El evaluador deberá evaluar las pruebas de los proveedores y pondrá a prueba el software para confirmar que los métodos utilizados para la generación de todas las claves criptográficas y otros materiales (como los IV, los valores "k" para las firmas digitales, etc.) tienen la entropía que cumple los requisitos mínimos de resistencia efectiva de las criptográficas primitivas y las claves.	La entropía es el grado de aleatoriedad de un generador de valores aleatorios. Cuanto mayor sea la entropía, menos predecible será el siguiente valor en un generador de números aleatorios.
	7.4.b Cuando las claves criptográficas se generan a través de procesos que requieren la interacción directa del usuario, como la entrada de una frase de contraseña o el uso de la interacción "aleatoria" del usuario con el software, el evaluador deberá evaluar la evidencia y probar el software para confirmar que estos procesos se implementen de tal manera que proporcionen suficiente entropía. Concretamente, el evaluador confirmará que: *(continúa en la página siguiente)*	Tenga en cuenta que un Generador de Números Aleatorios No Determinista (NDRG,) puede producir una cadena de salida que contiene menos entropía de lo que implica la longitud de la salida. Un Generador de Números Aleatorios Determinista (DRNG,) depende de la entropía de su valor semilla.

Objetivos de Control	Requisitos de las Pruebas	Guía
	• Los métodos utilizados para generar claves directamente a partir de una contraseña/frase de contraseña imponen un dominio de entrada que puede proporcionar suficiente entropía, de modo que las entradas totales posibles sean al menos iguales a la potencia de bits equivalente de la clave que se genera (por ejemplo, un campo de entrada de 32 dígitos hexadecimales para una clave AES128). • Las frases de contraseña se pasan través de una función de derivación de clave estándar de la industria, como PBKDF2 o bcrypt, que amplía el factor de trabajo para cualquier intento de aplicar fuerza bruta al valor de la frase de la contraseña. El evaluador confirmará que se aplica un factor de trabajo de al menos 10 000 a dicha aplicación. • Se proporciona una guía a las partes interesadas de acuerdo con el Objetivo de Control 12.1 que incluye instrucciones para que cualquier frase de contraseña utilizada deba: – Generarse aleatoriamente, utilizando un proceso aleatorio válido y seguro y no se debe utilizar un generador de números aleatorios en línea para este propósito. – Nunca sea implementa por una sola persona, de tal manera que una persona tenga una ventaja para recuperar el valor de la clave clara, violando los requisitos para el conocimiento dividido.	

Operaciones de Software Seguras

El software proporciona mecanismos para detectar y alertar sobre actividad anómala y para garantizar la responsabilidad del usuario.

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control 8: Seguimiento de la Actividad Se realiza un seguimiento de toda la actividad del software que involucra activos críticos.		
8.1 Todos los intentos de acceso y el uso de activos críticos son rastreados y trazables hasta un único usuario. ***Nota:*** *Este Estándar de Software Seguro reconoce que algunos entornos de ejecución no pueden apoyar los requisitos de registro detallados en otros estándares PCI. Por lo tanto, el término "seguimiento de actividades" se utiliza aquí para diferenciar las expectativas de este estándar con respecto al registro de requisitos similares en otros estándares PCI.*	**8.1** El evaluador deberá evaluar la evidencia y probar el software para confirmar que todos los intentos de acceso y uso de activos críticos se rastrean y rastrean hasta un individuo, sistema o entidad único.	Para garantizar la responsabilidad del usuario y apoyar la investigación forense posterior al incidente, el software de pago debe capturar y mantener registros históricos de todas las actividades de software que involucren activos críticos y garantizar que todas esas actividades puedan rastrearse hasta un usuario único (por ejemplo, una persona, sistema u otra entidad). Entre los ejemplos de actividades que el software debe registrar se incluyen: • Todos los intentos de los usuarios individuales para acceder a los datos o recursos confidenciales. • El uso o cambios de funciones confidenciales, tales como mecanismos de identificación y autenticación del software o los mecanismos de seguimiento de actividades. • La inicialización, detención o pausa de las funciones confidenciales. Este Objetivo de Control no exige el registro de cada operación de cifrado o función de procesamiento de datos confidenciales, pero sí requiere que se rastree el acceso y también se rastree cualquier método que pueda exponer datos confidenciales.

Objetivos de Control	Requisitos de las Pruebas	Guía
8.2 Toda la actividad se captura con el detalle suficiente y necesario para describir con precisión qué actividades específicas se realizaron, quién las realizó, el momento en que se realizaron y qué activos críticos se vieron afectados.	**8.2.a** El evaluador deberá evaluar la evidencia y probar el software para confirmar que el método o métodos de seguimiento implementados captan la actividad específica realizada, incluyendo: • La habilitación de cualquier modo de funcionamiento privilegiado. • La desactivación del cifrado de datos confidenciales. • La desencriptación de los datos confidenciales. • La exportación de datos confidenciales a otros sistemas o procesos. • Los intentos de autenticación erróneos. • La desactivación o eliminación de un control de seguridad o alteración las funciones de seguridad.	Al registrar los detalles en este requisito en todos los intentos de acceso o uso de activos críticos, la actividad maliciosa o el potencial de compromiso del software o datos pueden identificarse rápidamente y con suficiente detalle para saber quién realizó la actividad, si el intento tuvo éxito, cuando ocurrió la actividad, qué activos críticos se vieron afectados y el origen del evento.
	8.2.b El evaluador deberá evaluar la evidencia y probar el software para confirmar que el método o métodos de seguimiento implementados proporcionan lo siguiente: • Una identificación única para el individuo, sistema o entidad que accede o utiliza activos críticos. • Una marca de tiempo para cada evento rastreado. • Detalles sobre a qué activo crítico se ha accedido.	
	8.2.c El evaluador deberá probar el software para confirmar que los datos confidenciales no se registran directamente en los datos de seguimiento.	
8.3 El software apoya la retención segura de los registros de actividad detallados.	**8.3.a** Cuando los registros de actividad son administrados por el software, incluso solo temporalmente antes de pasar a otros sistemas, el evaluador deberá evaluar la evidencia del proveedor y probar el software para confirmar que los métodos de protección se implementan para proteger la integridad, exactitud e integridad de los registros de actividad.	Para identificar comportamientos anómalos y permitir la investigación forense ante la sospecha de un posible compromiso del software o de los datos, el software debe proporcionar la conservación de registros de actividad detallados, ya sea a través de medios nativos (dentro del propio software) o apoyando la integración con otras soluciones como servidores de registro centralizados, soluciones de registro basadas en la nube o soluciones de supervisión del servidor (*back-end*). *(continúa en la página siguiente)*

Objetivos de Control	Requisitos de las Pruebas	Guía
	8.3.b Cuando el software utilice sistemas externos o de terceros para el mantenimiento de los datos de seguimiento, tales como un servidor de registros, el evaluador deberá evaluar la evidencia para confirmar que se proporciona a las partes interesadas una guía sobre la configuración y/o integración correcta y completa del software con el sistema o sistemas externos o de terceros, de acuerdo con el Objetivo de Control 12.1.	Sin la protección adecuada de los registros de actividad, no se puede garantizar su integridad y exactitud, y se anularía cualquier confianza que de otro modo se pudiera depositar sobre ellos (tal como se hace durante una investigación forense). Cuando los registros de actividad son administrados por el software, los registros deben protegerse de acuerdo con todos los requisitos aplicables para la protección de datos confidenciales.
	8.3.c El evaluador probará el software para confirmar que se aplican métodos para asegurar la autenticidad de los datos de seguimiento durante la transmisión al sistema de almacenamiento de registros, y para confirmar que esta protección cumple los requisitos de este estándar (por ejemplo, los parámetros de autenticidad deben aplicarse utilizando criptografía robusta) y cualquier cuenta o parámetros de autenticación utilizados para el acceso a un sistema de registro externo están protegidos.	
8.4 El software gestiona los fallos en los mecanismos de seguimiento de la actividad de forma que se preserve la integridad de los registros de actividad existentes.	**8.4.a** El evaluador deberá evaluar la evidencia y probar el software para confirmar que el fallo del mecanismo o mecanismos de seguimiento de la actividad no viola la integridad de los registros existentes confirmando que: • El software no sobrescribe los datos de seguimiento existentes al reiniciar el software. Cada nuevo inicio solo se añadirá a los conjuntos de datos existentes o creará un nuevo conjunto de datos de seguimiento. • Cuando se confía en los nombres de conjuntos de datos únicos para mantener la integridad entre las instancias de ejecución, la implementación garantiza que otro software (incluyendo otra instancia del mismo software) no pueda sobrescribir o invalidar los conjuntos de datos existentes. *(continúa en la página siguiente)*	Se deben implementar controles de seguridad del software para garantizar que cuando fallan los mecanismos de seguimiento de las actividades, esas fallas se manejen de manera que se mantenga la integridad de los registros. De lo contrario, los atacantes pueden apuntar intencionadamente a los mecanismos de seguimiento de la actividad y provocar fallos que les permitan ocultar o sobrescribir la evidencia de sus actividades.

Objetivos de Control	Requisitos de las Pruebas	Guía
	8.4.a • Siempre que es posible, el software aplica privilegios de archivo adecuados para ayudar a mantener la integridad del conjunto de datos de seguimiento (como la aplicación de un control de acceso solo para añadir a un conjunto de datos una vez se haya creado). Cuando el software no aplique tales controles, el evaluador confirmará que existe una justificación razonable que describa por qué es así, por qué el comportamiento es suficiente y qué atenuaciones adicionales se aplican para mantener la integridad de los datos de seguimiento.	
	8.4.b El evaluador deberá evaluar la evidencia y probar el software para confirmar que la integridad de los registros de seguimiento de actividades se mantiene mediante: • Realizar acciones que deben rastrearse, forzar el cierre y luego reiniciar el software, y realizar otras acciones rastreadas. • Realizar acciones que se deben rastrear, encender la plataforma en la cual se ejecuta el software y, a continuación, reiniciar el software y realizar otras acciones rastreadas. • Bloquear el acceso al conjunto de datos de seguimiento y confirmar que el software gestiona la imposibilidad de acceder a este conjunto de datos de forma segura, como por ejemplo, creando un nuevo conjunto de datos o impidiendo el uso posterior del software. • Impedir la creación de nuevas entradas de conjuntos de datos impidiendo la escritura posterior en los medios en los que se encuentra el conjunto de datos (por ejemplo, mediante el uso de medios que no tienen suficiente espacio disponible). Cuando alguna de las pruebas anteriores no sea posible, el evaluador entrevistará al personal para confirmar que existe una justificación razonable para describir por qué este es el caso y deberá confirmar que se han implementado protecciones para evitar que tales escenarios afecten la integridad de los registros de seguimiento.	

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control 9: Detección de los Ataques Se detectan los ataques y se minimizan los impactos y efectos de los ataques.		
9.1 El software detecta y alerta ante la detección de comportamientos anómalos, como cambios en las configuraciones posteriores al despliegue o comportamientos de ataque evidentes.	**9.1.a** El evaluador deberá evaluar la evidencia y probar el software para confirmar que se implementan métodos para validar la integridad de los ejecutables del software y cualquier opción de configuración, archivos y conjuntos de datos en los que se base el software para su funcionamiento, de modo que se detecten cambios no autorizados posteriores a la implementación. Cuando el entorno de ejecución lo impida, el evaluador deberá evaluar la evidencia (incluida la documentación disponible públicamente sobre la plataforma y las tecnologías asociadas) para confirmar que, efectivamente, no existen métodos para validar la autenticidad y que se han implementados controles de seguridad adicionales para minimizar el riesgo asociado.	El software debe poseer una funcionalidad básica para diferenciar entre el comportamiento normal y el comportamiento anómalo del usuario. Los ejemplos de comportamientos anómalos que el software debe detectar automáticamente incluyen cambios en las configuraciones posteriores a la implementación (o posteriores a la inicialización) o comportamientos obvios de ataques automáticos, como intentos de autenticación repetidos con una frecuencia que no es factible para un usuario humano. En algunos casos, puede resultar poco práctico implementar estas capacidades directamente en el software de pago, por lo que puede ser obligatorio recurrir a herramientas o servicios de terceros. Cuando se dependa de dichas herramientas o servicios, el proveedor de software debe brindar una guía (o instrucciones sobre dónde puede obtenerse la guía adecuada) que describa cómo y en qué medida se deben configurar las herramientas y servicios de terceros para satisfacer el objetivo de control y los requisitos de prueba asociados.
	9.1.b El evaluador deberá evaluar la evidencia y probar el software para confirmar que los valores de integridad utilizados por el software y el conjunto o conjuntos de datos en los que se basa para un funcionamiento seguro se comprueban en el momento de la ejecución del software, y como mínimo cada 36 horas a partir de entonces (si el software continúa ejecutándose durante ese periodo de tiempo).	
	9.1.c Cuando se utilicen criptográficas primitivas mediante cualquier método de detección de anomalías, el evaluador deberá examinar la evidencia y probar el software para confirmar que las primitivas criptográficas están protegidas.	
	9.1.d Cuando los valores almacenados sean utilizados por cualquier método de detección de anomalías, el evaluador deberá evaluar la evidencia y probar el software para confirmar que estos valores se consideran datos confidenciales y están protegidos en consecuencia.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	9.1.e Cuando el software pueda modificar la configuración u otros valores del conjunto de datos durante la ejecución, el evaluador deberá evaluar la evidencia y probar el software para confirmar que las protecciones de integridad están implementadas para permitir esta actualización y, al mismo tiempo, garantizar que la integridad del conjunto de datos se pueda validar después de la actualización.	
	9.1.f El evaluador deberá evaluar la evidencia y probar el software para confirmar que éste implementa controles para evitar ataques de fuerza bruta en los campos de entrada de cuentas, contraseñas o claves criptográficas (por ejemplo, limitación de la velocidad de entrada).	
	9.1.g Cuando el software dependa de herramientas o servicios de terceros para proporcionar capacidades de detección de ataques, el evaluador deberá evaluar la evidencia para confirmar que se proporciona a las partes interesadas una guía sobre cómo configurar dichas herramientas y servicios para apoyar este objetivo de control, de acuerdo con el Objetivo de Control 12.1.	

Gestión Segura del Ciclo de Vida del Software

El software se mantiene utilizando prácticas de gestión del ciclo de vida del software seguro.

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control 10: Gestión de Amenazas y Vulnerabilidades Las amenazas y vulnerabilidades del software de pago se identifican, evalúan y gestionan adecuadamente.		
10.1 Se identifican, evalúan y abordan las amenazas y vulnerabilidades del software.	**10.1.a** Utilizando la información obtenida en el Requisito de Prueba 4.1.a, el evaluador deberá evaluar la evidencia para confirmar que se han identificados los métodos de ataque comunes contra el software. Esto puede incluir ataques a nivel de plataforma, protocolo o nivel de idioma.	Para determinar cómo proteger y defender eficazmente el software contra los ataques es necesario conocer a fondo las amenazas específicas y las posibles vulnerabilidades aplicables al software del proveedor. Por lo general, esto implica comprender lo siguiente: • Los tipos de información recopilada, almacenada, procesada o transmitida por el software. • Las motivaciones que un atacante puede tener para atacar el software. • Los métodos que un atacante podría utilizar o las vulnerabilidades que un atacante podría tratar de explotar durante un ataque. • La explotabilidad de cualquier vulnerabilidad identificada. • El impacto de un ataque exitoso. Las amenazas y vulnerabilidades identificadas deben rastrearse, asignarse al personal responsable y corregirse o mitigarse antes del lanzamiento del software de pago. Para obtener una guía sobre el análisis de amenazas y los principios de diseño de resiliencia cibernética, consulte los estándares y guías de la industria, como la versión actual de la *Publicación Especial NIST 800-160.*
	10.1.b El evaluador deberá evaluar las pruebas para confirmar que los ataques identificados son válidos para el software y señalará dónde no se incluyen los métodos de ataque comunes detallados en las referencias estándar de la industria, como las listas OWASP y CWE.	
	10.1.c El evaluador evaluará la evidencia para confirmar que se implementan mitigaciones contra cada ataque identificado y que el proceso de lanzamiento del software incluye la validación continua de la existencia de estas mitigaciones.	

Objetivos de Control	Requisitos de las Pruebas	Guía
10.2 Las vulnerabilidades en el software y los componentes de terceros se prueban y corrigen antes de su lanzamiento.	**10.2.a** El evaluador deberá evaluar la evidencia para confirmar que se utilizan procesos de prueba robustos a lo largo del ciclo de vida del software para gestionar las vulnerabilidades del software y para verificar que las mitigaciones utilizadas para proteger el software contra los ataques se mantengan y sean efectivas.	La mayoría de las vulnerabilidades del software se introducen como resultado de errores de codificación, un mal diseño, una implementación inadecuada de la funcionalidad del software o el uso de componentes vulnerables. El software debe desarrollarse y probarse de manera que minimice la existencia de vulnerabilidades y detecte las que surgen a lo largo del tiempo, de modo que las vulnerabilidades puedan abordarse antes de que se publique o actualice el software. Las técnicas para evitar la introducción de vulnerabilidades durante el desarrollo incluyen el uso de prácticas de codificación de seguridad, la comprobación del código durante cada fase del ciclo de vida del desarrollo utilizando herramientas automatizadas (como herramientas de análisis estático/dinámico y herramientas de pruebas de seguridad interactivas), y el uso de componentes seguros conocidos (por ejemplo, código común que ya ha sido objeto de una investigación de seguridad significativa). Para minimizar la introducción de vulnerabilidades del software de componentes de terceros, esos componentes también deben evaluarse. Idealmente, deberían estar sujetos a los mismos procesos seguros de desarrollo y prueba del software creado por el proveedor. El personal de los proveedores debidamente calificados o terceros deben realizar las pruebas de seguridad. Además, el personal encargado de las pruebas de seguridad debe poder realizarlas de forma objetiva y estar autorizado para comunicar cualquier vulnerabilidad identificada al personal de gestión o de desarrollo adecuado para que pueda tratarse adecuadamente.
	10.2.b El evaluador evaluará la evidencia, incluidos los procesos de la prueba documentados y los resultados de varias instancias de la prueba para confirmar que el proceso de prueba: • Incluye, como mínimo, el uso de herramientas automatizadas capaces de detectar vulnerabilidades tanto en el código del software como durante su ejecución. • Incluye el uso de herramientas de pruebas de seguridad adecuadas para la arquitectura de software, lenguajes de desarrollo y marcos utilizados en el desarrollo del software. • Contabiliza toda la base del código, incluyendo la detección de vulnerabilidades en componentes y bibliotecas de terceros, de código abierto o compartidos. • Contabiliza las vulnerabilidades comunes y métodos de ataque. • Demuestra un historial de búsqueda de vulnerabilidades de software y su reparación antes del lanzamiento del software.	
	10.2.c Cuando la evidencia evaluada en el Requisito de Prueba 10.2.b muestre el lanzamiento del software con vulnerabilidades conocidas, el evaluador deberá evaluar la evidencia adicional para confirmar que: • Se utiliza un sistema de clasificación de vulnerabilidades estándar de la industria (como CVSS) para clasificar/categorizar las vulnerabilidades. • Se mantiene un plan de remediación para todas las vulnerabilidades detectadas que asegura que las vulnerabilidades no permanezcan sin mitigar por un período indefinido.	

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control 11: Actualizaciones Seguras del Software Las versiones y actualizaciones de software para abordar las vulnerabilidades se proporcionan de manera segura y oportuna.		
11.1 Las actualizaciones del software para corregir vulnerabilidades conocidas se ponen a disposición de las partes interesadas de manera oportuna.	**11.1.a** El evaluador deberá evaluar la evidencia de para confirmar que: • Existen criterios razonables para publicar actualizaciones del software para corregir vulnerabilidades de seguridad. • Las actualizaciones de seguridad se ponen a disposición de las partes interesadas de acuerdo con los criterios definidos.	Las vulnerabilidades de los programas informáticos deben corregirse lo antes posible para que los usuarios de los mismos y otras partes interesadas puedan hacer frente a cualquier riesgo antes de que los atacantes puedan explotar las vulnerabilidades de sus sistemas de pago y programas informáticos. Las vulnerabilidades deben abordarse de una manera acorde con el riesgo que suponen para los usuarios del software u otras partes interesadas. Las vulnerabilidades más críticas o graves (es decir, las que tienen mayor posibilidad de ser explotadas y el mayor impacto potencial para las partes interesadas) deben ser parcheadas inmediatamente, seguidas de las que tienen una capacidad de explotación o un impacto potencial de moderado a bajo. Deben definirse y seguirse los criterios para determinar cómo y cuándo poner los parches a disposición de las partes interesadas.
	11.1.b El evaluador deberá evaluar las evidencias, incluidos los resultados y detalles de las pruebas de seguridad específicas de las actualizaciones, para confirmar que las actualizaciones de seguridad se ponen a disposición de las partes interesadas de acuerdo con los criterios definidos. Cuando no se proporcionen actualizaciones de acuerdo con los criterios definidos, el evaluador confirmará que tales casos están justificados y son razonables.	
11.2 Las versiones y actualizaciones del software se entregan de manera segura, lo cual garantiza la integridad del software y su código.	**11.2.a** El evaluador deberá evaluar la evidencia para confirmar que los métodos por los cuales el proveedor publica las actualizaciones de software mantienen la integridad del código de software durante la transmisión e instalación.	Las actualizaciones de seguridad deben incluir un mecanismo dentro del proceso de actualización para verificar que el código de actualización no haya sido reemplazado o manipulado. Los ejemplos de controles de integridad incluyen, entre otros, sumas de control y certificados firmados digitalmente (si se implementan correctamente). La verificación podría implementarse dentro del propio software o a través de la orientación que se proporciona a las partes interesadas para orientarlos sobre la verificación manual de las actualizaciones de software. Además, el proceso de distribución de actualizaciones y parches debe evitar que personas malintencionadas intercepten las actualizaciones en tránsito, las modifiquen y luego se las redistribuyan a clientes desprevenidos.
	11.2.b Cuando sea obligatoria la participación o interacción del usuario para validar la integridad del código del software, el evaluador deberá evaluar la evidencia para confirmar que se proporciona una guía sobre este proceso a las partes interesadas de acuerdo con el Objetivo de Control 12.1.	
	11.2.c Cuando el método de integridad implementado no sea criptográficamente seguro, el evaluador deberá evaluar la evidencia para confirmar que el método de la distribución de software proporciona una cadena de confianza, como por ejemplo mediante el uso de una conexión TLS que proporciona implementaciones de conjunto ‑de cifrado conformes.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	11.2.d El evaluador deberá evaluar la evidencia del vendedor para confirmar que se notifica a los interesados las actualizaciones del software y que se les proporciona una guía sobre cómo pueden obtenerlas e instalarlas, de acuerdo con el Objetivo de Control 12.1.	
	11.2.e El evaluador deberá evaluar la evidencia para confirmar que se notifica a las partes interesadas cuando se detectan vulnerabilidades conocidas en software que aún no se ha actualizado con una corrección. Esto incluye las vulnerabilidades que pueden existir en el software de terceros y bibliotecas y en las bibliotecas utilizadas por el software. El evaluador confirmará que este proceso incluye el suministro a los usuarios de sugerencias para mitigar dichas vulnerabilidades.	
	11.2.f El evaluador deberá evaluar la evidencia para confirmar que los mecanismos de actualización del software cubran todo el software, los archivos de configuración y otros metadatos que el software pueda utilizar con fines de seguridad o que puedan afectar de algún modo a la seguridad del software.	

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control 12: Guía de Implementación del Proveedor de Software El proveedor del software proporciona a las partes interesadas una guía clara y completa sobre la implementación, configuración y funcionamiento seguros del software.		
12.1 El proveedor del software proporciona a las partes interesadas una orientación clara y completa sobre la implementación, configuración y funcionamiento seguros de su software de pago.	**12.1.a** El evaluador deberá evaluar la evidencia para confirmar que el proveedor crea y proporciona a las partes interesadas una guía clara y suficiente para permitir la instalación, configuración y uso seguros del software. **12.1.b** El evaluador evaluará la evidencia para confirmar que la guía: • Incluye detalles sobre cómo instalar de forma segura y correcta cualquier software de terceros que sea obligatorio para el funcionamiento del software del proveedor. • Proporciona instrucciones sobre la configuración correcta de las plataformas en las cuales se va a ejecutar el software, incluyendo la configuración de parámetros de seguridad y la instalación de cualquier elemento de datos (como certificados). • Incluye instrucciones para la gestión de las claves (por ejemplo, el uso de las claves, cómo se distribuyen, cargan, retiran, cambian, destruyen, etc.) • No indica al usuario que deshabilite la configuración o los parámetros de seguridad dentro del entorno instalado, como software anti- malware o firewall u otros sistemas de protección a nivel de la red. • No indica al usuario que ejecute el software en un modo privilegiado superior al obligatorio por el software. • Proporciona detalles sobre cómo validar la versión del software e indican claramente para qué versiones del software está escrita la guía. *(continúa en la página siguiente)*	Cuando se siguen las guías de implementación del proveedor de software, se garantiza que el software y los parches se pueden instalar, configurar y mantener de forma segura en el entorno del cliente, y que todas las funciones de seguridad deseadas están activas y funcionando según lo previsto. La guía debe cubrir todas las opciones y funcionalidades disponibles para los usuarios del software que podrían afectar la seguridad del software o los datos con los cuales interactúa. La guía también debe incluir opciones de configuración segura para cualquier componente proporcionado con el software o apoyado por el software, tales como software externo y plataformas subyacentes. Entre los ejemplos de las opciones configurables se incluyen: • Cambio de credenciales y contraseñas predeterminadas. • Activación y desactivación de las cuentas, servicios y funciones de la aplicación. • Cambios en los permisos de acceso a los recursos. • Integración con bibliotecas criptográficas de terceros, generadores de números aleatorios, etc. La guía proporcionada debería dar como resultado una configuración segura en todas las plataformas apoyadas y todas las opciones configurables.

Objetivos de Control	Requisitos de las Pruebas	Guía
	12.1.b • Proporcionan una justificación para cualquier requisito de este estándar que deba evaluarse como no aplicable. Para cada uno de ellos, el evaluador confirmará que existe una justificación razonable de por qué este es el caso y confirmará que está de acuerdo con su comprensión y los resultados de sus pruebas del software.	

Módulo A – Requisitos de Protección de Datos de Tarjetahabientes

Nombre del Módulo	Descripción General	Objetivos de Control
Módulo A: Requisitos de Protección de Datos de Tarjetahabientes	Requisitos de seguridad para el software que almacena, procesa o transmite los datos de tarjetahabientes.	**A.1**: Datos de Autenticación Confidenciales **A.2**: Protección de Datos de Tarjetahabientes

Propósito y Alcance

Esta sección (en adelante denominada "Módulo de Protección de los Datos de Tarjetahabientes") define los requisitos de seguridad y los procedimientos de evaluación para el software que almacena, procesa o transmite los Datos de Tarjetahabientes. Para los propósitos de este módulo, los datos de tarjetahabientes se definen de la siguiente manera:

Datos de Tarjetahabientes	
Los Datos de Tarjetahabientes incluyen:	**Los Datos de Autenticación Confidenciales incluyen:**
▪ Número de la Cuenta Principal (PAN,) ▪ Nombre del Tarjetahabiente ▪ Fecha de Expiración ▪ Código de Servicio	▪ Datos de la pista completos (datos de la banda magnética o equivalentes en un chip) ▪ CAV2/CVC2/CVV2/CID ▪ Pines y bloques de pines

El número de la cuenta principal (PAN) es el factor que define los datos de tarjetahabientes. Si el PAN se almacena, procesa o transmite o está presente de otro modo, los requisitos de este módulo se aplican además de los Requisitos Básicos del Software Seguro.

La tabla de la página siguiente ilustra los elementos de uso común de los datos de tarjetahabientes y los datos de autenticación confidenciales, si el almacenamiento de esos datos está permitido o prohibido, y si es necesario proteger estos datos. Esta tabla no pretende ser exhaustiva, sino que se presenta para ilustrar los diferentes tipos de requisitos que se aplican a cada elemento de datos.

		Elementos de Datos	Almacenamiento Permitido	Hacer Ilegibles los Datos Almacenados por Objetivo de Control A.2.3
Datos de Tarjetahabientes	**Datos de Tarjetahabientes**	Número de la Cuenta Principal (PAN,)	Sí	Sí
		Nombre del Tarjetahabiente	Sí	No
		Código de Servicio	Sí	No
		Fecha de Expiración	Sí	No
	Datos de Autenticación Confidenciales[2]	Datos de la Pista Completa[3]	No	No se puede almacenar según el Objetivo de Control A.1.1
		CAV2/CVC2/CVV2/CID[4]	No	No se puede almacenar según el Objetivo de Control A.1.1
		Bloqueo PIN/PIN[5]	No	No se puede almacenar según el Objetivo de Control A.1.1

Los Objetivos de Control A.2.2 y A.2.3 se aplican solo al PAN. Si el PAN se almacena con otros elementos de los datos de tarjetahabientes, solo el PAN debe ser ilegible de acuerdo con el objetivo de Control A.2.3. Los datos de autenticación confidenciales no deben almacenarse después de la autorización, incluso si están de forma encriptada, a menos que el software esté destinado únicamente a ser utilizado por los emisores o las organizaciones que apoyan los servicios de emisión. Solo en esos casos se pueden almacenar datos de autenticación confidenciales después de la autorización.

2 *Los datos de autenticación confidenciales no deben almacenarse después de la autorización (incluso si están cifrados).*

3 *Datos de la pista completa de la banda magnética, datos equivalentes en el chip o en otro lugar.*

4 *El valor de tres o cuatro dígitos impreso en el anverso o reverso de una tarjeta de pago.*

5 *Número de identificación personal introducido por el tarjetahabiente durante una transacción de presentación de la tarjeta, y/o bloque PIN cifrado presente en el mensaje de transacción.*

Requisitos de Seguridad

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control A.1: Datos de Autenticación Confidenciales Los Datos de Autenticación Confidenciales (SAD) no se conservan después de la autorización.		
A.1.1 El software no almacena datos de autenticación confidenciales después de la autorización (incluso si están encriptados) a menos que el software esté diseñado para ser utilizado únicamente por emisores u organizaciones que apoyan los servicios de emisión.	**A.1.1** Usando la información obtenida en el Requisito de Prueba 1.1.a en la sección de Requisitos Básicos, el evaluador deberá examinar la evidencia y probar el software para identificar todas las ubicaciones de almacenamiento potenciales para los Datos de Autenticación Confidenciales y para confirmar que el software no almacena dichos datos después de que complete la autorización de la transacción. Esto incluye el almacenamiento de SAD en almacenamiento temporal (como memoria volátil), almacenamiento semipermanente (como discos RAM) y almacenamiento no volátil (como medios de almacenamiento magnéticos y flash). Cuando los Datos de Autenticación Confidenciales se almacenen después de la autorización, el evaluador deberá evaluar la evidencia para confirmar que el software está diseñado explícitamente para fines de emisión o para que lo utilicen los emisores u organizaciones que apoyan los servicios de emisión.	Los datos de autenticación confidenciales consisten en los datos completos de la pista, el código o valor de validación de la tarjeta y los datos del PIN. Se prohíbe el almacenamiento de los datos de autenticación confidenciales después de la autorización. Estos datos son valiosos para las personas malintencionadas, ya que les permiten generar tarjetas de pago falsificadas y crear transacciones fraudulentas. Las pruebas deben incluir al menos los siguientes tipos de archivos, así como cualquier otro resultado generado por el software de pago: • Datos de las transacciones entrantes • Todos los registros (por ejemplo, transacción, historial, depuración, error) • Archivos de historial • Archivos de seguimiento • Archivos de audio e imagen (por ejemplo, voz digital y biometría) • Memoria no volátil (incluyendo la caché no volátil) • Esquemas de los bases de datos • Contenido de la base de datos

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control A.2: Protección de Datos de Tarjetahabientes Los datos almacenados de tarjetahabientes están protegidos.		
A.2.1 El proveedor del software proporciona una guía a los clientes sobre la eliminación segura de los datos de tarjetahabientes después de la expiración del período de retención definido por el cliente.	**A.2.1** El evaluador deberá evaluar la evidencia para confirmar que se proporciona una guía a las partes interesadas de acuerdo con el Objetivo de Control 12.1 que detalla: • Una lista de todas las ubicaciones donde el software almacena los datos de tarjetahabientes. • Cómo eliminar de forma segura los datos de tarjetahabientes almacenados por el software de pago, incluidos los datos de tarjetahabientes almacenados en software o sistemas subyacentes (como en archivos del sistema operativo o en bases de datos). • Cómo configurar el software o los sistemas subyacentes para evitar la captura o retención involuntaria de los datos de tarjetahabientes (por ejemplo, mediante copias de seguridad del sistema o puntos de restauración).	El proveedor del software debe proporcionar detalles de todas las ubicaciones donde el software puede almacenar los datos de tarjetahabientes, incluso en cualquier software o sistema subyacente (como SO, bases de datos, etc.), así como instrucciones para eliminar de forma segura los datos de estas ubicaciones una vez que los datos hayan excedido el valor definido por el cliente como período de retención. A las partes interesadas también se les debe proporcionar detalles de configuración para los sistemas subyacentes en los que se ejecuta el software para garantizar que estos sistemas subyacentes no capturen datos de tarjetahabientes sin el conocimiento de la parte interesada. Las partes interesadas deben saber cómo los sistemas subyacentes podrían estar capturando datos del software para que puedan evitar que se capturen o asegurarse de que los datos estén debidamente protegidos.

Objetivos de Control	Requisitos de las Pruebas	Guía
A.2.2 El software proporciona funciones para restringir o enmascarar todas las presentaciones del PAN al número mínimo de dígitos obligatorios.	**A.2.2.a** El evaluador deberá evaluar la evidencia para confirmar que el software proporciona funciones que permiten a las partes responsables restringir o enmascarar la visualización del PAN al número mínimo de dígitos obligatorio para satisfacer una necesidad empresarial definida.	La visualización del PAN completo en elementos como las pantallas del ordenador, los recibos de tarjetas de pago, registros, faxes o informes en papel puede dar lugar a que personas no autorizadas usen los datos los obtenidos y que se utilicen fraudulentamente. El enfoque de enmascaramiento siempre debe garantizar que solo se muestre el número mínimo de dígitos según sea necesario para realizar una función comercial específica. Por ejemplo, si solo se necesitan los últimos cuatro dígitos para realizar una función comercial, el software debe proporcionar funciones para enmascarar el PAN de modo que las personas que realizan esa función puedan ver solo los últimos cuatro dígitos.
	A.2.2.b El evaluador deberá examinar la evidencia para confirmar que todas las visualizaciones del PAN están completamente enmascaradas por defecto, y que es obligatoria la autorización explícita para mostrar cualquier dígito del PAN.	
	A.2.2.c Cuando sea obligatoria la participación o interacción del usuario para configurar las funciones y opciones de enmascaramiento del PAN, el evaluador deberá evaluar la evidencia para confirmar que se proporciona una guía sobre cómo configurar estas funciones/opciones a las partes interesadas de acuerdo con el Objetivo de Control 12.1.	
	A.2.2.d El evaluador deberá examinar la evidencia para confirmar que todas las visualizaciones del PAN están completamente enmascaradas por defecto, y que es obligatoria una autorización explícita para mostrar cualquier dígito del PAN.	

Objetivos de Control	Requisitos de las Pruebas	Guía
A.2.3 El PAN se vuelve ilegible en cualquier lugar donde esté almacenado (incluidos los datos en medios digitales portátiles, medios de copia de seguridad y registros) mediante cualquiera de los siguientes enfoques: • Truncamiento (los hashes no pueden utilizarse para reemplazar el segmento truncado del PAN). • Tokens de índice y pads (los pads deben guardarse de forma segura). • Criptografía robusta con procesos y procedimientos de gestión de claves asociados.	**A.2.3.a** El evaluador deberá evaluar la evidencia y probar el software para confirmar que se han implementado métodos para hacer ilegible el PAN en cualquier lugar donde se almacene utilizando los siguientes métodos: • Truncamiento. • Tokens de índice y pads, con los pads almacenados de forma segura. • Criptografía robusta, con procesos y procedimientos de gestión de claves asociados. ***Nota**: El evaluador deberá evaluar varias tablas, archivos, archivos de registro y cualquier otro recurso creado o generado por el software para verificar que el PAN sea ilegible.* **A.2.3.b** Cuando sea obligatoria la participación o interacción del usuario para configurar métodos que hagan que el PAN sea ilegible cuando se almacene, el evaluador deberá evaluar la evidencia para confirmar que se proporciona orientación sobre la configuración de estas opciones a las partes interesadas de acuerdo con el Objetivo de Control 12.1 y que la guía incluye lo siguiente: • Detalles de cualquier opción configurable de cada método utilizado para hacer que los datos de tarjetahabientes sean ilegibles e instrucciones sobre cómo configurar cada método para todas las ubicaciones donde se almacenan los datos de tarjetahabientes. • Una lista de todas las instancias en las que los datos de tarjetahabientes pueden salir para ser almacenados fuera de la aplicación de pago e instrucciones de que la entidad que lo implementa es responsable de hacer que el PAN sea ilegible en todas esas instancias. *(continúa en la página siguiente)*	La falta de protección de los PAN puede permitir que personas malintencionadas vean o descarguen estos datos. La intención del truncamiento es que sólo se almacene una parte (que no supere los seis primeros y los cuatro últimos dígitos) del PAN. La intención de la criptografía robusta es que el cifrado se base en un algoritmo probado y aceptado en la industria (no en un algoritmo propietario o "creado en casa"), con claves criptográficas robustas.

Objetivos de Control	Requisitos de las Pruebas	Guía
	• Instrucción de que si alguna vez se habilitan los registros de depuración (para solucionar problemas) y ellos contienen PAN, estos deben protegerse; que dicha depuración debe desactivarse tan pronto como se complete la solución de problemas y que los registros de depuración deben eliminarse de forma segura cuando ya no se necesiten.	
	A.2.3.c Cuando el software crea versiones tokenizadas y truncadas del mismo PAN, el evaluador deberá evaluar la evidencia y probar el software para confirmar que las versiones tokenizadas y truncadas no se pueden correlacionar para reconstruir el PAN original.	
	A.2.3.d Cuando el software cree o genere archivos para su uso fuera del software (por ejemplo, archivos generados para la exportación o la realización de copias de seguridad), incluido el almacenamiento en soportes extraíbles, el evaluador deberá evaluar las evidencias y probar el software para confirmar que el PAN resulta ilegible.	
	A.2.3.e Si el proveedor del software almacena PAN por cualquier motivo (por ejemplo, porque se reciben archivos de registro, archivos de depuración y otras fuentes de datos de los clientes con fines de depuración o solución de problemas), el evaluador deberá evaluar la evidencia y probar el software para confirmar que el PAN se hace ilegible de acuerdo con este objetivo de control.	

Módulo B – Requisitos del Software del Terminal

Nombre del Módulo	Descripción General	Objetivos de Control
Módulo B: Requisitos del Software del Terminal	Requisitos de seguridad para el software destinado a la implementación y ejecución en dispositivos POI aprobados por PCI.	**B.1**: Documentación del Software del Terminal **B.2**: Diseño del Software del Terminal **B.3**: Mitigación de los Ataques del Software del Terminal **B.4**: Pruebas de Seguridad del Software del Terminal **B.5**: Guía de Implementación del Software del Terminal

Propósito y Alcance

Esta sección (en lo sucesivo denominada "Módulo de Software del Terminal" o "este módulo") define los requisitos de seguridad y los procedimientos de evaluación para el software de pago y las aplicaciones que dependen de las características de seguridad de los dispositivos de POI aprobados por PCI para proteger los datos de pago. Las aplicaciones de software desarrolladas explícitamente para su implementación y ejecución en dispositivos POI aprobados por PCI que no cumplen con la definición de Firmware según se define en los *Requisitos de Seguridad Modular del Punto de Interacción (POI) de Seguridad de Transacciones PIN (PTS) de PCI* (en lo sucesivo, como el "Estándar PCI PTS POI") están dentro del alcance de los requisitos de este módulo.

Antecedentes

Los dispositivos de POI aprobados por PCI proporcionan un alto grado de confidencialidad y protección de la integridad de los datos de pago y las transacciones de pago mediante la aplicación de estrictos mecanismos de protección física y lógica. El software que se implementa y ejecuta en dispositivos POI aprobados por PCI no debe degradar ni afectar negativamente los mecanismos de protección proporcionados por el dispositivo. Además, el software no debe proporcionar características o funciones que puedan facilitar o permitir que esos mecanismos de protección se eludan o se vuelvan ineficaces.

Los requisitos y procedimientos de evaluación definidos en el Módulo de Software del Terminal se han desarrollado para ayudar a garantizar que el software del terminal proteja los datos de pago y no introduzca características, funciones o debilidades que le permitan al atacante eludir o hacer ineficaces los mecanismos de protección proporcionados por los dispositivos de POI aprobados por PCI subyacentes en los que se pretende implementar el software.

Consideraciones

Algunos procedimientos de evaluación de este módulo requieren del examen de la documentación que describe las características y funciones de seguridad del terminal de pago subyacente. El proveedor del software del terminal debe trabajar con sus evaluadores, así como con los respectivos proveedores de los terminales de pago para cada uno de los dispositivos que se incluirán como parte de la evaluación del software del terminal, para identificar y compilar toda la documentación del dispositivo necesaria para la evaluación del software del terminal. Para obtener más información sobre la preparación y las actividades de la evaluación de Software Seguro, consulte la *Guía del Programa de Software Seguro.*

Requisitos de Seguridad

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control B.1: Documentación del Software del Terminal La arquitectura del software está documentada e incluye diagramas que describen todos los componentes y servicios del software en uso y cómo interactúan.		
B.1.1 El proveedor del software mantiene documentación que describe todos los componentes del software, interfaces y servicios proporcionados o utilizados por el software.	**B.1.1.** El evaluador deberá evaluar la evidencia para confirmar que se mantiene la documentación que describe todos los flujos de datos confidenciales, incluidos, entre otros, los siguientes: • Todos los componentes de código abierto y de terceros, servicios externos e Interfaces de Programación de Aplicaciones (API,) utilizados por el software. • Todas las Interfaces de Usuario (UI,) y las API proporcionadas o hechas accesibles por el software.	Los proveedores del software también deben mantener la documentación detallada que describa clara y eficazmente el diseño y la función general de su software, incluyendo todos los servicios (internos y externos), componentes y funciones utilizados y proporcionados por el software, y cómo esos servicios, componentes y funciones que interactúan.
B.1.2 El proveedor del software mantiene la documentación que describe todos los flujos de datos y funciones que involucran los datos confidenciales. ***Nota:*** *Este objetivo de control es una extensión de los Objetivos de Control 1.1 y 1.2. La validación de estos objetivos de control debe realizarse al mismo tiempo.*	**B.1.2.a** El evaluador deberá evaluar la evidencia para confirmar que se mantiene la documentación que describe todos los flujos de datos confidenciales, incluidos, entre otros, los siguientes: • Todos los datos confidenciales almacenados, procesados o transmitidos por el software. • Todas las ubicaciones donde se almacenan los datos confidenciales, incluyendo las ubicaciones de almacenamiento temporales y persistentes. • Cómo se eliminan de forma segura los datos confidenciales del almacenamiento (tanto temporales como persistentes) cuando ya no se necesitan.	Además de identificar los componentes, interfaces y servicios expuestos por el software, el proveedor del software también debe mantener la documentación que identifique y describa claramente los tipos de datos almacenados, procesados y transmitidos por el software y cómo se comparten esos datos entre los componentes y los mecanismos de protección implementados o en los que el software confía para proteger esos datos. Este tipo de documentación aclara cómo se almacenan, procesan o transmiten los datos por el software, con quién se comparten los datos y cómo se puede atacar el software para acceder a los activos críticos del mismo.

Objetivos de Control	Requisitos de las Pruebas	Guía
	B.1.2.b El evaluador deberá evaluar la evidencia para confirmar que se mantenga la documentación que describa todas las funciones que manejan datos confidenciales, incluidas, entre otras, las siguientes: • Todas las entradas, salidas y posibles condiciones de error para cada función que maneja los datos confidenciales. • Todos los algoritmos criptográficos, modos de operación y prácticas de administración de claves asociadas para todas las funciones que emplean criptografía para la protección de los datos confidenciales.	
B.1.3 El proveedor del software mantiene la documentación que describe todas las opciones configurables que pueden afectar a la seguridad de los datos confidenciales.	**B.1.3** El evaluador deberá evaluar la evidencia para confirmar que se mantiene la documentación que describe todas las opciones configurables proporcionadas o puestas a disposición por el software que pueden afectar la seguridad de los datos confidenciales, incluidos, entre otros, los siguientes: • Todas las opciones configurables que podrían permitir el acceso a los datos confidenciales. • Todas las opciones configurables que podrían permitir la modificación de cualquier mecanismo utilizado para proteger los datos confidenciales. • Todas las características, funciones y parámetros de acceso remoto proporcionados o puestos a disposición por el software. • Todas las características, funciones y parámetros de actualización remota proporcionados o puestos a disposición por el software. • La configuración predeterminada para cada opción configurable.	Los proveedores del software deben identificar todas las opciones configurables disponibles dentro de su software, especialmente las que controlan las funciones y características de seguridad. Las características configurables deben ser consideradas como potenciales vías de ataque al software. Cuando las opciones configurables permiten el control sobre las funciones y las características de seguridad, deben implementarse controles de seguridad robustos para proteger las características de seguridad configurables del mal uso. Además, todas las opciones configurables deben estar configuradas en sus valores más seguros por defecto, de acuerdo con el Objetivo de Control 2.

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control B.2: Diseño del Software del Terminal El software no implementa ningún aspecto que permita que las características, funciones y características de seguridad del terminal de pago sean eludidas o ineficaces.		
B.2.1 El software está destinado a ser implantado y operado en los terminales de pago (dispositivos POI aprobados por PCI).	**B.2.1** El evaluador deberá evaluar la evidencia para determinar los terminales de pago en los que se utilizará el software. Para cada uno de los terminales de pago identificados e incluidos en la evaluación del software, el evaluador deberá evaluar las características del dispositivo del terminal de pago y compararlas con las siguientes características especificadas en *la Lista de Dispositivos PTS aprobados del PCI SSC* para confirmar que coinciden: • Nombre y número del modelo • Número de aprobación de la Seguridad de la PTS • Número de la versión del hardware • Número(s) de la versión del Firmware	Los terminales de pago proporcionan un alto grado de confidencialidad e integridad de los datos de pago y las transacciones de pago mediante la aplicación de mecanismos estrictos de protección física y lógica. Los programas informáticos que se despliegan y ejecutan en estos terminales de pago deben utilizar las características y funciones aprobadas que proporciona el terminal de pago, en lugar de implementar sus propias características o funciones equivalentes, para evitar exponer vulnerabilidades u otros puntos débiles que podrían permitirle a un atacante eludir o hacer ineficaces las características de seguridad del terminal de pago.
B.2.2 El software solo utiliza los métodos de comunicación externa incluidos en la evaluación del dispositivo de la PTS del terminal de pago. ***Nota:*** *El terminal de pago puede proporcionar una pila IP aprobada por el módulo PTS de protocolos abiertos, o el dispositivo puede proporcionar los puertos seriales o módem aprobados por la evaluación de la PTS para comunicar los datos de transacción cifrados por sus funciones PCI PTS SRED. El uso de los métodos de comunicación externos no incluidos en la evaluación del dispositivo POI aprobado por PCI invalida la aprobación de PTS, y dicho uso está prohibido para el software del terminal.*	**B.2.2.a** El evaluador deberá evaluar la evidencia (incluido el código fuente) para determinar si el software apoya las comunicaciones externas. **B.2.2.b** Cuando el software apoye las comunicaciones externas, el evaluador deberá evaluar toda la documentación pertinente del terminal de pago (incluida la guía/política de seguridad del proveedor del terminal de pago) para determinar qué métodos de comunicación externa se incluyeron en la evaluación del dispositivo STP del terminal de pago. **B.2.2.c** El evaluador deberá evaluar la evidencia (incluido el código fuente) para confirmar que el software utiliza únicamente los métodos de comunicación externa incluidos en la evaluación del dispositivo STP del terminal de pago y no implementa sus propios métodos de comunicación externa o pila IP.	Para garantizar que el software no degrade o anule los mecanismos de seguridad proporcionados por el terminal de pago subyacente, el software debe utilizar las características y funciones de seguridad proporcionadas por el dispositivo de acuerdo con la guía o política de seguridad del proveedor del terminal de pago. Esto es particularmente cierto para los métodos de comunicación externos. En ningún caso el software debe proporcionar sus propios métodos de comunicación (por ejemplo, VMs, pila IP, lenguajes de script , etc.) para controlar las interfaces a nivel del dispositivo. La introducción de cualquier función de este tipo por parte del software podría introducir nuevas vulnerabilidades o debilidades que permitirían a las entidades maliciosas eludir las protecciones de seguridad proporcionadas por el terminal de pago y degradar las características de seguridad generales tanto del software como del dispositivo subyacente.

Objetivos de Control	Requisitos de las Pruebas	Guía
B.2.2.1 Cuando el software se basa en la función de Protocolos Abiertos del terminal de pago, el software se desarrolla de acuerdo con la guía y política de seguridad del proveedor del terminal de pago.	**B.2.2.1** El evaluador deberá evaluar toda la documentación pertinente del terminal de pago (incluida la guía/política de seguridad del proveedor del terminal de pago) y toda la documentación pertinente del proceso del vendedor del software y la documentación del diseño del software para confirmar que el software se ha desarrollado de acuerdo con la guía/política de seguridad del proveedor del terminal de pago.	La guía y política de seguridad del proveedor de terminales de pago está destinada a los desarrolladores de aplicaciones, integradores de sistemas y usuarios finales de la plataforma para cumplir con los requisitos del Protocolo Abierto de PTS POI de PCI (así como otros PTS) como parte de una evaluación de dispositivos POI aprobada por PCI.
B.2.2.2 El software no elude, omite ni agrega servicios o protocolos adicionales a los Protocolos Abiertos del terminal de pago según lo aprobado y documentado en la guía y política de seguridad del proveedor del terminal de pago. Esto incluye el uso de: • Protocolos Capa del Enlace • Protocolos IP • Protocolos de seguridad • Servicios IP	**B.2.2.2** El evaluador deberá evaluar la evidencia (incluido el código fuente) para confirmar que el software no elude, evita o añade servicios o protocolos adicionales a los Protocolos Abiertos del terminal de pago, tal y como se aprobó y documentó en la guía/política de seguridad del vendedor del terminal de pago. Esto incluye el uso de: • Protocolos Capa del Enlace • Protocolos IP • Protocolos de seguridad • Servicios IP	Los requisitos del protocolo abierto en el *estándar PCI PTS POI* aseguran que los protocolos y servicios abiertos en los terminales de pago no tengan vulnerabilidades que puedan explotarse de forma remota y brinden acceso a los datos confidenciales o recursos confidenciales en el terminal de pago. El proveedor del terminal de pago define qué protocolos y servicios apoyan el terminal de pago y proporciona la guía para su uso. Agregar o habilitar servicios o protocolos adicionales o no seguir la guía/política de seguridad del proveedor del terminal de pago emitido, invalida el estado de aprobación de ese dispositivo para dicha implementación.
B.2.3 El software no evita ni hace ineficaz ningún método de encriptación o de seguridad de los datos de tarjetahabientes implementado por el terminal de pago de acuerdo con la guía y política de seguridad del proveedor del terminal de pago.	**B.2.3.a** El evaluador deberá evaluar la evidencia (incluido el código fuente) para determinar si el software proporciona cifrado de datos confidenciales. Cuando el software proporcione dicha función, el evaluador deberá confirmar que el software no omite ni hace ineficaz ningún método de cifrado o método de seguridad de datos de tarjetahabientes implementado por el terminal de pago de la siguiente manera:	Los terminales de pago están diseñados para proporcionar sólidas funciones criptográficas y de gestión de claves. Por ejemplo, se ha verificado que los dispositivos aprobados por la PCI PTS POI cumplen con los estrictos requisitos de carga, gestión y protección de claves criptográficas. El software que proporciona sus propios métodos de encriptación de datos no debe incluir métodos que permitan a un atacante omitir o hacer ineficaces los métodos de encriptación implementados por el terminal de pago y obligatorios por la guía/ política de seguridad del proveedor del terminal de pago.

Objetivos de Control	Requisitos de las Pruebas	Guía
	B.2.3.b El evaluador deberá evaluar toda la documentación pertinente del terminal de pago (incluida la guía/política de seguridad del proveedor del terminal de pago) para determinar qué métodos de cifrado proporciona el terminal de pago.	
	B.2.3.c El evaluador deberá evaluar la evidencia (incluido el código fuente) para confirmar que el software no elude ni hace ineficaz ningún método de encriptación proporcionado por el terminal de pago de acuerdo con la guía/política de seguridad del proveedor del terminal de pago.	
	B.2.3.d Cuando el software proporciona cifrado de datos confidenciales, pero no es obligatorio que el terminal de pago proporcione métodos de encriptación aprobados (según el *Estándar PCI PTS POI*), el evaluador deberá evaluar la evidencia (incluido el código fuente) para confirmar que los métodos de encriptación utilizados o implementados por el software para encriptar datos confidenciales proporcionan una "criptografía robusta" y se implementan de acuerdo con los Objetivos de Control 7.1 y 7.2.	
B.2.4 El software utiliza únicamente la(s) función(es) de generación de números aleatorios incluida(s) en la evaluación del dispositivo STP del terminal de pago en todas las operaciones criptográficas que involucran datos confidenciales o funciones confidenciales en las que son obligatorios valores aleatorios y no implementa su(s) propia(s) función(es) de generación de números aleatorios.	**B.2.4.a** El evaluador deberá evaluar la evidencia (incluido el código fuente) para determinar si el software requiere que se generen valores aleatorios para cualquier operación criptográfica que involucre datos confidenciales o funciones confidenciales.	El hecho de que los números aleatorios sean imprevisibles es de vital importancia para garantizar la eficacia de las operaciones criptográficas. Diseñar e implementar un generador de números aleatorios seguro no es una tarea trivial. Por esta razón, el software del terminal solo debe usar la función o funciones de generación de números aleatorios implementadas por el terminal de pago para todas las operaciones criptográficas que involucren datos confidenciales o funciones confidenciales donde sean obligatorios valores aleatorios.
	B.2.4.b Cuando el software requiera valores aleatorios para operaciones criptográficas que impliquen datos confidenciales o funciones confidenciales, el evaluador deberá evaluar toda la documentación pertinente del terminal de pago (incluidas la guía/políticas de seguridad del proveedor del terminal de pago) para determinar todas las funciones de generación de números aleatorios incluidas en la evaluación del dispositivo PTS del terminal de pago.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	B.2.4.c El evaluador deberá evaluar la evidencia (incluido el código fuente) para confirmar que el software utiliza únicamente la(s) función(es) de generación de números aleatorios incluida(s) en la evaluación del dispositivo PTS del terminal de pago en todas las funciones criptográficas que involucran datos confidenciales o funciones confidenciales en las que son obligatorios valores aleatorios, y no implementa su(s) propia(s) función(es) de generación de números aleatorios.	
B.2.5 El software no facilita, a través de sus propias interfaces lógicas, el intercambio de datos de tarjetahabientes en texto claro directamente con otro software. ***Nota:*** *El software puede compartir los datos de tarjetahabientes en texto claro directamente con el firmware del terminal de pago.*	**B.2.5.a** El evaluador deberá evaluar la evidencia (incluido el código fuente) para determinar todas las interfaces lógicas del software, lo que incluye: • Todas las interfaces lógicas y el propósito y la función de cada uno. • Las interfaces lógicas destinadas a compartir datos de tarjetahabientes en texto claro como las que se utilizan para devolver datos de tarjetahabientes en texto claro al firmware aprobado del terminal de pago. • Las interfaces lógicas que no están destinadas a compartir los datos de tarjetahabientes en texto claro, como las destinadas a la comunicación con otro software.	Muchos terminales de pago ofrecen mecanismos para la Lectura y el Intercambio Seguro de Datos (SRED). Estos mecanismos se prueban rigurosamente como parte de la evaluación del dispositivo PTS del terminal de pago para confirmar que se mantiene la confidencialidad y la integridad de los datos de tarjetahabientes en texto claro durante el intercambio de información con el firmware del terminal de pago. El software que proporciona sus propios mecanismos para compartir datos de tarjetahabientes en texto claro directamente con otro software tiene más probabilidades de ser propenso a ataques y a la divulgación no intencionada o no autorizada de datos de tarjetahabientes en texto claro que el software que utiliza las funciones SRED (o similares) proporcionadas por el terminal de pago.
	B.2.5.b El evaluador deberá evaluar la evidencia (incluido el código fuente) para confirmar que el software no permite compartir datos de tarjetahabientes en texto claro directamente con otro software a través de sus propias interfaces lógicas.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	B.2.5.c El evaluador deberá instalar y configurar el software de acuerdo con la guía obligatoria en los Objetivos de Control 12.1 and B.5.1. Utilizando una "plataforma de prueba" apropiada y herramientas y/o métodos forenses adecuados, el evaluador probará el software utilizando todas las funciones de software que manejan datos de tarjetahabientes para confirmar que el software no permite compartir datos de tarjetahabientes en texto claro directamente con otro software a través de sus propias interfaces lógicas.	
B.2.6 El software utiliza y/o integra todos los recursos compartidos de forma segura y de acuerdo con las directrices y políticas de seguridad del proveedor del terminal de pago.	**B.2.6.a** El evaluador deberá evaluar la evidencia (incluido el código fuente) para determinar si el software se conecta y/o utiliza los recursos compartidos proporcionados por el terminal de pago y cómo lo hace, y para confirmar que: • La guía obligatoria de los Objetivos de Control 12.1 y B.5.1 incluye instrucciones detalladas sobre cómo configurar el software para garantizar una integración segura con los recursos compartidos. • La guía obligatoria para la integración segura con recursos compartidos está de acuerdo con la guía/política de seguridad del proveedor de la terminal de pago.	Cuando el software utilice o integre los recursos compartidos proporcionados por el terminal de pago, el software deberá utilizar o integrar recursos de acuerdo con la dirección y política del proveedor de terminales de pago. Si no se utilizan estos recursos compartidos de acuerdo con las directrices de los terminales de pago, los datos confidenciales compartidos con dichos recursos corren un mayor riesgo de ser revelados sin autorización.
	B.2.6.b El evaluador deberá instalar y configurar el software de acuerdo con la guía obligatoria en los Objetivos de Control 12.1 y B.5.1. Usando una "plataforma de prueba" apropiada y herramientas y/o métodos forenses adecuados (herramientas comerciales, scripts, etc.), el evaluador deberá probar el software usando todas las funciones del software que usan o integran recursos compartidos para confirmar que cualquier conexión o uso de recursos compartidos los recursos se manejan de forma segura.	

Objetivos de Control	Requisitos de las Pruebas	Guía
B.2.7 El software no eludirá ni hace ineficaz ninguna segregación de aplicaciones aplicada por el terminal de pago.	**B.2.7.a** El evaluador deberá evaluar toda la documentación relevante del terminal de pago (incluida las directrices y políticas de seguridad del proveedor del terminal de pago) necesaria para determinar si el terminal de pago aplica la segregación de aplicaciones y cómo lo hace. **B.2.7.b** El evaluador deberá examinar la evidencia (incluido el código fuente) para confirmar que el software no presenta ninguna función que le permita eludir o anular cualquier control de segregación de aplicaciones a nivel de dispositivo.	Muchos terminales de pago imponen la separación lógica entre las aplicaciones del software. En el contexto de este módulo, las aplicaciones del software son entidades lógicas que no se ajustan a la definición del "firmware" de la PTS. Los controles de segmentación de aplicaciones lógicas están destinados a evitar que una aplicación en el terminal de pago interfiera o altere otras aplicaciones. Sin embargo, estos controles de segregación lógica no están destinados a impedir que las aplicaciones compartan datos. Su objetivo principal es evitar que las aplicaciones modifiquen la estructura o el funcionamiento de otras aplicaciones o del Firmware del terminal de pago. Para preservar la integridad de los controles de segregación de las aplicaciones de los terminales de pago, todos los programas informáticos de los terminales deben adherirse a esos controles de segregación y no incluir o introducir ninguna función o funciones que permita utilizar el software (intencional o involuntariamente) para eludir o derrotar la aplicación de segregación a nivel del dispositivo.
B.2.8 Todos los archivos de software están firmados criptográficamente para permitir la autenticación criptográfica de los archivos de software por parte del firmware del terminal de pago.	**B.2.8.a** El evaluador deberá evaluar la guía obligatoria en los Objetivos de Control 12.1 y B.5.1 para confirmar que incluye instrucciones detalladas sobre cómo firmar criptográficamente los archivos de software de manera que permita la autenticación criptográfica de todos esos archivos por parte del terminal de pago.	Para apoyar la autenticación criptográfica de los archivos del software por parte del terminal de pago, los proveedores del software deben "firmar" de forma criptográfica todos los archivos del software (incluyendo todos los binarios, bibliotecas y archivos de configuración) utilizando los certificados digitales en los cuales el proveedor del terminal de pago esté incluido en la cadena de certificados. Además, el proceso de la firma criptográfica debe incorporar el uso de un Dispositivo Criptográfico Seguro (SCD), generalmente proporcionado por el proveedor del terminal de pago. La firma criptográfica también debe realizarse bajo un control dual para proteger la integridad de todas las claves criptográficas, los archivos del software y el proceso de la firma criptográfica en general.

Objetivos de Control	Requisitos de las Pruebas	Guía
	B.2.8.b El evaluador deberá instalar y configurar el software de acuerdo con la guía obligatoria en los Objetivos de Control 12.1 y B.5.1. Usando una "plataforma de prueba" apropiada y herramientas y/o métodos forenses adecuados, el evaluador deberá confirmar que todos los archivos de software estén firmados criptográficamente de manera que permita la autenticación criptográfica de todos los archivos de software.	
	B.2.8.c Cuando el software admita la carga de archivos fuera de los paquetes de software base, el evaluador deberá evaluar la evidencia y probar el software para determinar si cada uno de esos archivos está firmado criptográficamente de manera que permita la autenticación criptográfica de esos archivos por parte del terminal de pago. En el caso de los archivos que no puedan firmarse de forma criptográfica, el evaluador deberá justificar por qué la imposibilidad de firmar de forma criptográfica cada uno de esos archivos no afecta negativamente a la seguridad del software o del terminal de pago subyacente.	
	B.2.8.d El evaluador deberá evaluar la evidencia (incluido el código fuente) para determinar si el software es compatible con las transacciones de pago EMV® y de qué manera. Cuando las transacciones de pago EMV estén apoyadas por el software, el evaluador deberá instalar y configurar el software de acuerdo con la guía obligatoria en los Objetivos de Control 12.1 y B.5.1. Utilizando una "plataforma de prueba" apropiada y herramientas y/o métodos forenses adecuados, el evaluador deberá confirmar que todas las claves públicas de la autoridad de certificación EMV estén firmadas criptográficamente de manera que permita la autenticación criptográfica de esos archivos por parte del terminal de pago.	Cuando el software del terminal admita transacciones de pago EMV, las claves públicas de la Autoridad de Certificación EMV también deben firmarse y autenticarse de forma criptográfica utilizando los mismos métodos y procedimientos que los archivos del software del terminal.

Objetivos de Control	Requisitos de las Pruebas	Guía
B.2.9 La integridad de los archivos de la solicitud del software está protegida de acuerdo con el Objetivo de Control B.2.8.	**B.2.9.a** El evaluador deberá evaluar la evidencia (incluido el código fuente) para determinar si el software apoya el uso de avisos de entrada de datos y/o archivos de solicitud. Cuando el software admita dichas funciones, el evaluador deberá confirmar que el software protege la integridad de esas indicaciones, tal como se define en los requisitos de prueba de B.2.9.b a B.2.9.c.	Los datos confidenciales (incluidos el PIN y otros datos de tarjetahabientes) capturados y manejados por el software y el terminal de pago subyacente a menudo se controlan mediante archivos de solicitud. Los archivos de solicitud son archivos de configuración que controlan las solicitudes de visualización del software. Para preservar la integridad de las solicitudes, los archivos de solicitudes deben almacenarse y administrarse de forma segura. En cualquier lugar donde el software permita la entrada de datos en texto claro, se deben implementar controles de solicitud.
	B.2.9.b El evaluador deberá evaluar la guía obligatoria en los Objetivos de Control 12.1 y B.5.1 para confirmar que incluye las instrucciones detalladas para que las partes interesadas firmen criptográficamente todos los archivos de solicitud de una manera que permita la autenticación criptográfica de todos esos archivos de acuerdo con B.2.8.	Muchos de los archivos de solicitud se almacenan dentro de un límite seguro del dispositivo, como un chip seguro o un elemento seguro o dentro de un Entorno de Ejecución de Confianza. Cuando los archivos de solicitud deben mantenerse en las ubicaciones de almacenamiento compartido, los archivos deben firmarse criptográficamente y autenticados por el terminal de pago antes de la instalación o ejecución.
	B.2.9.c El evaluador deberá instalar y configurar el software de acuerdo con la guía obligatoria en los Objetivos de Control 12.1 y B.5.1. Utilizando una "plataforma de prueba" apropiada y herramientas y/o métodos forenses adecuados, el evaluador confirmará que todos los archivos de solicitud están firmados criptográficamente de manera que permita la autenticación criptográfica de esos archivos por parte del terminal de pago de acuerdo con B.2.8.	

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control B.3: Mitigación de los Ataques del Software del Terminal Los controles de seguridad del software se implementan para mitigar los ataques al software.		
B.3.1 El software valida todos los usuarios y otras entradas externas. ***Nota:*** *Los Objetivos de Control B.3.1 a B.3.3 son extensiones del Objetivo de Control 4.2. La validación de estos objetivos de control debe realizarse al mismo tiempo.*	**B.3.1.a** El evaluador deberá evaluar la evidencia (incluyendo el código fuente) para identificar todos los lugares donde el software acepta datos de entrada de fuentes no confiables. Para cada instancia, el evaluador deberá confirmar que los datos de entrada están obligados a ajustarse a una lista de características esperadas y que todas las entradas que no se ajustan a la lista de características esperadas son rechazadas por el software o manejadas de otra manera de forma segura.	Cualquier función de software de terminal que acepte datos suministrados externamente (directa o indirectamente) es un vector de ataque potencial, particularmente cuando los datos son procesados por un intérprete. Los ataques de inyección son comunes en casi todos los tipos de software y están destinados a manipular los datos de entrada de una manera que haga que el software se comporte inesperada o involuntariamente. Por ejemplo, el software que acepta información suministrada externamente, como un nombre de archivo o una ruta del archivo para construir un comando de búsqueda, puede ser fácilmente manipulado para revelar información sobre archivos y recursos confidenciales a los que nunca se pretendió acceder a través de la interfaz del software. Para protegerse contra este y otros tipos de ataques de inyección, todos los datos de entrada deben validarse, filtrarse y/o desinfectarse antes de enviar la información a cualquier intérprete. Las entradas para el software del terminal tienden a involucrar los comandos y datos simples. Por lo tanto, todos los datos de entrada del software del terminal deben validarse con un conjunto definido y restringido de valores aceptables antes de pasar los datos a cualquier intérprete de comandos. Cualquier dato que no se identifique explícitamente como un valor aceptable o un rango de valores aceptable debe rechazarse.
	B.3.1.b El evaluador deberá instalar y configurar el software de acuerdo con la guía obligatoria en los Objetivos de Control 12.1 y B.5.1. Utilizando una “plataforma de prueba” apropiada y herramientas y/o métodos forenses adecuados, el evaluador deberá probar el software intentando proporcionar a cada usuario u otra entrada externa características no válidas o inesperadas para confirmar que el software valida todas las entradas y rechaza o maneja de forma segura todas las características inesperadas.	

Objetivos de Control	Requisitos de las Pruebas	Guía
B.3.1.1 Todos los valores de la cadena son validados por el software.	**B.3.1.1.a** El evaluador deberá evaluar la evidencia (incluido el código fuente) para identificar todas las funciones de software del terminal en las que se pasan valores de cadena como entradas, y para confirmar que todas las cadenas se verifican en busca de texto o datos que puedan interpretarse errónea o maliciosamente como un comando.	Las entradas suministradas externamente que pueden interpretarse como comandos son particularmente susceptibles a los ataques de inyección. Incluso si los insumos suministrados externamente se procesan o transforman de alguna manera (por ejemplo, se aumentan con datos adicionales o se desinfectan), aún pueden ser susceptibles.
	B.3.1.1.b El evaluador deberá instalar y configurar el software de acuerdo con la guía obligatoria en los Objetivos de Control 12.1 y B.5.1. Utilizando una "plataforma de prueba" apropiada y herramientas y/o métodos forenses adecuados, el evaluador deberá probar el software intentando suministrar a cada una de las funciones identificadas datos que incluyan comandos para confirmar que el software rechaza dichas entradas o las maneja de otro modo seguro.	Por lo tanto, todas las entradas que puedan interpretarse como comandos deben manejarse de forma segura para que la ejecución de cualquier comando construido se controle, en lugar de ejecutar ciegamente cualquier comando incluido en la cadena.
B.3.1.2 El software comprueba las entradas y rechaza o maneja de forma segura cualquier entrada que infrinja el tamaño del búfer u otros umbrales de asignación de memoria.	**B.3.1.2.a** El evaluador deberá evaluar la evidencia (incluido el código fuente) para identificar todas las funciones de software que gestionen los búferes y procesen los datos suministrados a partir de fuentes no fiables. Para cada una de las funciones indicadas, el evaluador deberá confirmar que cada una de las funciones identificadas: • Utiliza solo variables sin signo para definir el tamaño de los búferes. • Realiza verificaciones que confirman que los búferes tienen el tamaño adecuado para los datos que deben manejar, incluyendo la consideración de sub desbordamientos y desbordamientos. • Rechaza o gestiona de forma segura cualquier entrada que infrinja el tamaño del búfer u otros umbrales de asignación de memoria.	Los terminales de pago y el software de los terminales suelen utilizar lenguajes de programación de bajo nivel, como C y C++. Estos lenguajes permiten que el software manipule directamente las características y funciones a nivel del sistema operativo o a nivel del hardware. El uso de lenguajes de programación de bajo nivel ofrece muchas ventajas, pero también tiene varios inconvenientes. Los lenguajes de programación de bajo nivel son susceptibles a ataques que utilizan las características de bajo nivel para manipular el software o el hardware subyacente. Los desbordamientos y sub desbordamientos del búfer son ejemplos de este tipo de ataques. *(continúa en la página siguiente)*

Objetivos de Control	Requisitos de las Pruebas	Guía
	B.3.1.2.b El evaluador deberá instalar y configurar el software de acuerdo con la guía obligatoria en los Objetivos de control 12.1 y B.5.1. Utilizando una "plataforma de prueba" adecuada y herramientas y/o métodos forenses apropiados (herramientas comerciales, scripts, etc.), el evaluador probará el software intentando suministrar a cada función señalada con entradas que violen los umbrales de tamaño del búfer para confirmar que el software rechaza o maneja con seguridad todos esos intentos.	Para protegerse contra los ataques de desbordamiento del búfer todas las funciones del software del terminal que definen o controlan el tamaño de los búferes deben comparar la cantidad de datos destinados a esos búferes con el tamaño del búfer. Los datos que violen los umbrales de tamaño del búfer (desbordamientos y sub desbordamientos) deben rechazarse o manejarse de forma segura.
B.3.2 Los valores devueltos se verifican y las condiciones de error se manejan de forma segura.	**B.3.2.a** Utilizando la información obtenida en el Requisito de Prueba 1.2.a, el evaluador deberá evaluar la evidencia (incluido el código fuente) para identificar todas las funciones de software que manejan datos confidenciales. Para cada una de las funciones del software señaladas, el evaluador confirmará que cada función: • Comprueba los valores de retorno para detectar la presencia de datos confidenciales. Procesa los valores retorno de manera que no "filtre" inadvertidamente los datos confidenciales.	Otra técnica común utilizada por los atacantes para comprometer los datos confidenciales almacenados, procesados o transmitidos por el software es manipular el software de una manera que genera excepciones no manejadas. Las excepciones no controladas son condiciones de error que el proveedor del software no ha anticipado y, por lo tanto, no ha tenido en cuenta en el diseño del software. Si el atacante puede manipular una función de software que se sabe que maneja datos confidenciales de una manera que genera una condición que el software no maneja correctamente, es posible que el software genere un error que incluya datos confidenciales. Para protegerse contra los ataques que involucran excepciones no manejadas, todas las funciones del software del terminal que manejan datos confidenciales deben incluir procesos o rutinas que instruyan al software sobre cómo tratar las excepciones desconocidas. Estos procesos deben determinar qué información incluir en los códigos o valores de error. Debe evitarse la divulgación de datos confidenciales mediante condiciones de error o informes de errores, ya sea intencional o accidental.
	B.3.2.b El evaluador deberá instalar y configurar el software de acuerdo con la guía obligatoria en los Objetivos de Control 12.1 y B.5.1. El evaluador, utilizando una "plataforma de prueba" adecuada y herramientas y/o métodos forenses apropiados, deberá probar cada función del software que maneja datos confidenciales al tratar de manipular el software de manera que se genera una excepción no manejada para confirmar que las condiciones de error no exponen los datos confidenciales.	

Objetivos de Control	Requisitos de las Pruebas	Guía
B.3.3 Se evitan las condiciones de carrera.	**B.3.3.a** El evaluador deberá evaluar la evidencia (incluido el código fuente) para identificar todas las funciones de software que dependen del procesamiento síncrono. Para cada una de las funciones señaladas, el evaluador deberá confirmar que se han implementado los mecanismos de protección en el software para mitigar las condiciones de carrera.	Las condiciones de carrera pueden surgir cuando el software requiere el procesamiento secuencial de los datos para realizar alguna función del software. Por ejemplo, existe una "condición de carrera del tiempo de uso y tiempo de comprobación" cuando un archivo se comprueba en un punto y se utiliza inmediatamente después, con el supuesto de que la comprobación anterior sigue siendo válida. Esta suposición puede no ser correcta si el sistema permite que el archivo se modifique en el medio. Si un atacante puede identificar y manipular el software para aprovechar una condición de carrera, puede llegar a ejecutar un código arbitrario o generar otras condiciones que el atacante podría explotar aún más.
	B.3.3.b El evaluador deberá instalar y configurar el software de acuerdo con la guía obligatoria en los Objetivos de Control 12.1 y B.5.1. Utilizando una "plataforma de prueba" adecuada y herramientas y/o métodos forenses apropiados (herramientas comerciales, scripts, etc.), el evaluador deberá probar cada función del software que se base en el procesamiento síncrono intentando generar una condición de carrera (por ejemplo, mediante ataques especialmente diseñados para explotar el tiempo de los eventos síncronos) para confirmar que el software es resistente a tales ataques.	Para protegerse contra las condiciones de carrera, el software del terminal debe implementar mecanismos de protección para controlar más estrictamente el procesamiento secuencial. Utilizando el ejemplo descrito anteriormente, se podría utilizar un mecanismo de "bloqueo" para evitar las actualizaciones del archivo hasta que el archivo se pueda procesar por completo. Independientemente de los métodos utilizados, cualquier software del terminal que requiera el procesamiento secuencial de datos para su funcionamiento debe implementar las protecciones para evitar condiciones de carrera.

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control B.4: Pruebas de Seguridad del Software del Terminal El software se prueba rigurosamente para detectar vulnerabilidades antes de cada lanzamiento.		
B.4.1 Se mantiene y sigue un proceso documentado para probar el software en busca de vulnerabilidades antes de cada actualización o lanzamiento. ***Nota:*** *Este objetivo de control es una extensión del Objetivo de Control 10.2. La validación de estos objetivos de control debe realizarse al mismo tiempo.*	**B.4.1.a** El evaluador deberá evaluar la evidencia para confirmar que el proveedor de software mantiene un proceso documentado de acuerdo con el Objetivo de Control 10.2 para probar el software en busca de vulnerabilidades antes de cada actualización o lanzamiento, y que el proceso documentado incluye descripciones detalladas de cómo el proveedor prueba lo siguiente: • La presencia o uso de puertos y protocolos innecesarios. • El almacenamiento, transmisión o la salida no intencionados de cualquier dato de tarjetahabiente en texto claro. • La presencia de cualquier cuenta de usuario por defecto con credenciales de acceso por defecto o estáticas. • La presencia de cualquier credencial de autenticación codificada en código o en archivos de configuración. • La presencia de cualquier dato de prueba o cuentas de prueba. • La presencia de cualquier control de seguridad del software defectuoso o ineficaz.	Muchas vulnerabilidades del software son el resultado de que el proveedor del software no eliminó las funciones o los datos de prueba. Estas funciones y datos persistentes pueden proporcionarle al atacante una ruta para poner en peligro el software. Antes de que el software se haga público, debe probarse para confirmar que las funciones y los datos de prueba no están incluidos en la versión de lanzamiento. Entre los ejemplos de estas funciones y datos que deben eliminarse explícitamente antes de la liberación se incluyen: • Cualquier puerto o protocolo de comunicación que no sea absolutamente obligatorio para el funcionamiento del software. • Cualquier función que permita el almacenamiento, la transmisión o la salida no intencionados de cualquier dato de tarjetahabiente en texto claro. • Cualquier credencial de autenticación codificada en código o archivos de configuración. • Cualquier dato de prueba o cuentas de usuario de prueba. • Cualquier control de seguridad del software y mecanismos de protección defectuosos o ineficaces.
	B.4.1.b El evaluador deberá evaluar la evidencia para confirmar que el software se somete a pruebas para detectar vulnerabilidades antes de cada autorización y que las pruebas cubren lo siguiente: • La presencia o uso de puertos y protocolos innecesarios. • El almacenamiento, transmisión o la salida no intencionados de cualquier dato de tarjetahabiente en texto claro. *(continúa en la página siguiente)*	

Objetivos de Control	Requisitos de las Pruebas	Guía
	B.4.1.b • La presencia de cualquier cuenta de usuario predeterminada con credenciales de acceso estáticas. • La presencia de cualquier credencial de autenticación codificada en código o en archivos de configuración. • La presencia de cualquier dato de prueba o cuentas de prueba. • La presencia de cualquier control de seguridad del software defectuoso o ineficaz.	

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control B.5: Guía de Implementación del Software del Terminal El proveedor del software le proporciona a las partes interesadas una guía clara y exhaustiva sobre la implementación, configuración y funcionamiento seguros del software en los terminales de pago aplicables.		
B.5.1 El proveedor del software proporciona una guía de implementación sobre cómo implementar y operar el software de forma segura para los terminales de pago en los que se implementará. ***Nota**: Este objetivo de control es una extensión del Objetivo de Control 12.1. La validación de estos objetivos de control debe realizarse al mismo tiempo.*	**B.5.1** El evaluador deberá evaluar la evidencia para confirmar que se proporciona a las partes interesadas una guía sobre cómo implementar y operar de forma segura el software para todos los terminales de pago aplicables, de acuerdo con el Objetivo de Control 12.1.	Dado que muchas de las características de seguridad utilizadas por el software del terminal son proporcionadas por el terminal de pago subyacente, el proveedor de software del terminal debe incluir instrucciones en su guía de implementación sobre cómo configurar todas las características de seguridad disponibles tanto del software del terminal como del terminal de pago subyacente, cuando sea aplicable.
B.5.1.1 La guía de implementación incluye instrucciones detalladas sobre cómo configurar todas las opciones y los parámetros de seguridad disponibles del software.	**B.5.1.1** El evaluador deberá evaluar las evidencias para confirmar que la guía obligatoria incluye instrucciones detalladas sobre cómo configurar todas las opciones y parámetros de seguridad disponibles del software de acuerdo con el Objetivo de Control B.1.3.	
B.5.1.2 La guía de implementación incluye instrucciones detalladas sobre cómo configurar de forma segura el software para utilizar las características y funciones de seguridad del terminal de pago, cuando proceda.	**B.5.1.2** El evaluador deberá evaluar la evidencia para confirmar que la guía obligatoria incluye instrucciones detalladas sobre cómo configurar de forma segura el software para utilizar las características y funciones de seguridad del terminal de pago cuando proceda.	

Objetivos de Control	Requisitos de las Pruebas	Guía
B.5.1.3 La guía de implementación incluye instrucciones detalladas sobre cómo configurar el software para integrar o utilizar de forma segura cualquier recurso compartido proporcionado por el terminal de pago.	**B.5.1.3** El evaluador deberá evaluar la evidencia para confirmar que la guía obligatoria incluye instrucciones detalladas sobre cómo configurar el software para integrar o utilizar de forma segura cualquier recurso compartido proporcionado por el terminal de pago de acuerdo con el Objetivo de Control B.2.6.	
B.5.1.4 La guía de implementación incluye instrucciones detalladas sobre cómo firmar criptográficamente los archivos del software de forma que facilite la autenticación criptográfica de todos estos archivos por parte del terminal de pago.	**B.5.1.4** El evaluador deberá evaluar la evidencia para confirmar que la guía obligatoria incluye instrucciones detalladas sobre cómo firmar criptográficamente los archivos de software de forma que permita la autenticación criptográfica de todos esos archivos por parte del terminal de pago de acuerdo con el Objetivo de Control B.2.8.	
B.5.1.5 La guía de implementación incluye instrucciones para que las partes interesadas firmen criptográficamente todos los archivos de solicitud.	**B.5.1.5** El evaluador deberá evaluar la evidencia para confirmar que la guía obligatoria incluye instrucciones detalladas para que las partes interesadas firmen criptográficamente todos los archivos de solicitud de acuerdo con el Objetivo de Control B.2.9.	
B.5.2 La guía de implementación se adhiere a la guía del proveedor del terminal de pago sobre la configuración segura del terminal de pago.	**B.5.2** El evaluador deberá evaluar la evidencia (incluyendo la guía/política de seguridad del vendedor del terminal de pago y la guía obligatoria en el Objetivo de Control B.5.1) para confirmar que la guía se alinea con la guía/política de seguridad del vendedor del terminal de pago.	La guía de implementación del software debe excluir las instrucciones que entren en conflicto con la guía y las recomendaciones del proveedor del terminal de pago. La guía de implementación del software debe alinearse con la guía/política de seguridad del proveedor del terminal de pago. De lo contrario, los usuarios del software que confían en el proveedor del software para recibir instrucciones pueden, sin saberlo, configurar incorrectamente el software y / o el terminal de pago subyacente.

Módulo C – Requisitos del Software Web

Nombre del Módulo	Descripción General	Objetivos de Control
Módulo C: Requisitos del Software Web	Requisitos de seguridad adicionales para el software de pago que utiliza tecnologías, protocolos e idiomas de Internet para iniciar o apoyar transacciones de pago electrónico.	**C.1**: Servicios y Componentes del Software Web **C.2**: Controles de Acceso al Software Web **C.3**: Mitigación de Ataques del Software Web **C.4**: Comunicaciones de Software Web

Propósito y Alcance

Esta sección (denominada en lo sucesivo "módulo de software web" o "este módulo") define los requisitos de seguridad y los procedimientos de evaluación para el software y las aplicaciones de pago que utilizan tecnologías, protocolos y lenguajes de Internet con el fin de iniciar o apoyar operaciones de pago electrónico. Esto incluye aplicaciones de pago tanto tradicionales (monolíticas) como nativas de la nube, API, servicios web, microservicios, funciones sin servidor, GRPC y cualquier otro método utilizado para hacer accesibles las funciones de pago o para realizar transacciones de pago electrónico a través de Internet. Cualquier característica o función basada en software que maneje solicitudes de "clientes" de Internet y genere respuestas para iniciar o apoyar una transacción de pago electrónico está en el alcance de los requisitos de este módulo.

Consideraciones

Las arquitecturas de software web pueden ser extremadamente complejas e involucrar características y funciones que son proporcionadas por diferentes entidades y que pueden estar distribuidas en diferentes ubicaciones geográficas. Los aspectos de seguridad que afectan al software web pueden variar significativamente. Los requisitos de seguridad definidos en este módulo web no abordan todos los riesgos que afectan al software de pago basado en la web. Pretenden ser un conjunto mínimo de características de seguridad, controles, funciones y capacidades que el software de pago basado en la web debe poseer para defenderse de los ataques más comunes al software web.

Aunque muchos de los objetivos de control definidos en este estándar protegen contra ataques nuevos y/o novedosos más allá de las técnicas más comunes, inevitablemente evolucionarán o surgirán ataques que requerirán nuevos métodos o enfoques para mitigarlos. En última instancia, es responsabilidad de los vendedores, proveedores, desarrolladores y suministradores de software de pago mantenerse al corriente de la evolución de las técnicas de ataque y aplicar los controles de seguridad adecuados para que su software pueda defenderse de dichos ataques.

Requisitos de Seguridad

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control C.1: Servicios y Componentes del Software Web Todos los componentes y servicios utilizados por el software se identifican y mantienen de forma que se minimice la exposición a vulnerabilidades.		
C.1.1 Todos los componentes y servicios de software están documentados o catalogados de otro modo en una lista de materiales de software (SBOM).	**C.1.1** El evaluador deberá evaluar la evidencia para confirmar que se mantiene la información que describe todos los componentes de software y servicios que comprenden la solución de software, incluyendo: • Todas las bibliotecas, paquetes, módulos y/o código de software propietario empaquetados de forma que puedan ser rastreados como una unidad independiente de software. • Todos los marcos, bibliotecas y código de terceros y de código abierto incrustados en el software o utilizados por el software durante la operación. • Todas las dependencias de software de terceros, API y servicios llamados por el software durante su funcionamiento.	El software moderno rara vez se crea íntegramente de forma interna y suele estar compuesto por varios segmentos de código a medida integrados con numerosos componentes como marcos comerciales y/o de código abierto, bibliotecas, API y servicios. Cualquier parte de este código puede tener o desarrollar vulnerabilidades con el tiempo que requerirán parches o mitigación. Conocer todos los componentes que integran una aplicación o servicio de software, de dónde proceden y cómo se actualizan y mantienen es fundamental para minimizar y gestionar las vulnerabilidades de las aplicaciones de software. Sin esta información, sería extremadamente difícil identificar y rastrear las vulnerabilidades en los componentes de software que podrían exponer la aplicación de software de incrustación a ataques. Una Lista de Materiales de Software o "SBOM" cumple este propósito al documentar la información sobre los componentes y versiones de software utilizados para crear un producto de software, sus proveedores y cualquier código de terceros que también pueda estar incrustado en estos componentes. NIST se refiere a esta información como "datos de procedencia" y existen numerosos estándares y marcos disponibles, como CycloneDX, SPDX y SWID, que describen cómo se debe estructurar esta información. Para más información, consulte dichos estándares y marcos.

Objetivos de Control	Requisitos de las Pruebas	Guía
C.1.2 La SBOM describe cada uno de los componentes y servicios primarios en uso, así como sus relaciones y dependencias de componentes secundarios transitivos en la mayor medida posible.	**C.1.2.a** El evaluador deberá evaluar la evidencia para confirmar que la SBOM describe todos los componentes y servicios primarios (de nivel superior) en uso y todas sus relaciones y dependencias transitivas secundarias. **C.1.2.b** El evaluador deberá probar el software para confirmar que la información proporcionada en la SBOM refleja con precisión los componentes y servicios de software en uso durante el funcionamiento del software, incluyendo tanto los componentes y servicios primarios como sus relaciones y dependencias de componentes secundarios transitivos. Cuando dichas dependencias y relaciones no estén identificadas y descritas en el SBOM, el evaluador deberá confirmar que la ausencia de dicha información está justificada y es razonable.	Los componentes y servicios de software pueden tener muchas relaciones y dependencias anidadas con otros componentes y servicios de software que son propiedad o están mantenidos por múltiples entidades diferentes. Identificar todas estas relaciones diferentes puede ser todo un reto cuando hay muchos componentes diferentes de terceros anidados en el código del software. Afortunadamente, muchos marcos de desarrollo de software y compiladores proporcionan la capacidad de identificar y asignar dependencias anidadas y transitorias. A efectos de este estándar, se espera que la SBOM identifique, como mínimo, el código obtenido de los terceros, así como sus relaciones y dependencias de componentes transitivos secundarios (es decir, el código incrustado en código de terceros). Si existen circunstancias que complican o impiden la identificación de las relaciones y dependencias de los componentes transitivos secundarios, dichas circunstancias deberán documentarse y deberá mantenerse una justificación razonable para explicar por qué estas dependencias no se reflejan con exactitud en la SBOM. Ejemplos de tales circunstancias pueden incluir API de terceros, donde la transparencia en los componentes anidados de terceros llamados por o integrados en esas API no es proporcionada por el proveedor de API.

Objetivos de Control	Requisitos de las Pruebas	Guía
C.1.3 Cuando el software se proporciona "como servicio", la SBOM incluye información que describe las dependencias del software presentes en el entorno de ejecución del software de producción en la mayor medida posible.	**C.1.3.a** El evaluador deberá evaluar la evidencia para confirmar que la SBOM describe todas las dependencias presentes en el entorno de ejecución del software de producción del que depende el software para su funcionamiento o para satisfacer los requisitos de seguridad de este estándar.	El software que se proporciona "como servicio" suele involucrar el uso de componentes y servicios residentes en el entorno de producción que son exclusivos de ese entorno. Para garantizar que se identifican y rastrean estas dependencias y relaciones y se identifican y mitigan las vulnerabilidades de estos componentes y servicios, estos componentes y servicios deben incluirse también en la SBOM.
	C.1.3.b El evaluador deberá evaluar la evidencia y probar el software (en la medida de lo posible) para confirmar que la información proporcionada en la SBOM refleja con exactitud las dependencias del software presentes en el entorno de ejecución del software de producción. Cuando dichas dependencias no estén identificadas y descritas en el SBOM, el evaluador deberá confirmar que la ausencia de dicha información está justificada y es razonable.	Algunos ejemplos de este tipo de componentes son, entre otros, los servidores de bases de datos, los servidores web, los servidores de aplicaciones, las plataformas de ejecución, los servidores/servicios de autenticación, los "plugins" y cualquier otro componente o servicio presente en el entorno de producción.

Objetivos de Control	Requisitos de las Pruebas	Guía
C.1.4 La SBOM incluye información suficiente sobre cada componente o servicio para permitir el seguimiento de cada componente o servicio a lo largo de la cadena de suministro del software.	**C.1.4.a** El evaluador deberá evaluar la evidencia para confirmar que se mantiene la información en la SBOM que describe lo siguiente para cada componente y servicio en uso, incluyendo las relaciones y dependencias de los componentes secundarios: • La fuente/proveedor original del componente o servicio. • El nombre del componente o servicio tal y como lo definió el proveedor original. • Una descripción de la(s) relación(es) entre el componente y el servicio y otros componentes/servicios integrados en el software o utilizados por éste. • La versión del componente o servicio definida por el proveedor original para diferenciarlo de las versiones anteriores o de otro tipo. • El nombre del autor que diseñó/desarrolló el componente o servicio. • Cualquier otro identificador proporcionado por el proveedor original para identificar de forma única el componente o servicio. **C.1.4.b** El evaluador deberá evaluar la evidencia y probar el software para confirmar que la información proporcionada en la SBOM es una representación exacta de los componentes de software y servicios presentes en y/o en uso por el software.	El objetivo principal de una SBOM es permitir el seguimiento de los componentes de software a lo largo de la cadena de suministro de software y asignarlos a repositorios que contengan información sobre vulnerabilidades de estos componentes. Para facilitar el seguimiento de los componentes con estos fines, debe incluirse información en la SBOM que permita a los interesados en el software: • Identifique de forma exclusiva cada uno de los componentes y servicios utilizados por el software. • Identificar de forma única las diferentes versiones de los mismos componentes de software y servicios que pueden ser utilizados por el software, y para diferenciarlos de otras versiones de los mismos componentes de software y servicios puestos a disposición por el proveedor o proveedores. • Localice las fuentes de estos componentes y servicios para poder descargar, instalar y/o hacer referencia a las versiones actualizadas cuando proceda. Sin esta información básica, el seguimiento de las vulnerabilidades y los parches disponibles en estos componentes y servicios puede resultar extremadamente difícil, si no imposible.

Objetivos de Control	Requisitos de las Pruebas	Guía
C.1.5 Se crea o genera una nueva SBOM cada vez que se actualiza el software.	**C.1.5** El evaluador deberá evaluar la evidencia para confirmar que se crea o se genera una nueva SBOM para cada nueva versión del software.	Para permitir el seguimiento de las vulnerabilidades a través de las diferentes versiones de un software de pago, es imprescindible que cada versión del software tenga una SBOM generada que refleje con precisión los componentes y servicios en uso por esa versión. Dado que muchas versiones diferentes de un software de pago pueden estar disponibles (o activas) en un momento dado y pueden incluir múltiples versiones de numerosos componentes y servicios de terceros, cada versión del software de pago debe ser rastreada independientemente de otras versiones. Si no se identifican y describen los componentes y servicios exclusivos de una versión determinada del software de pago, podrían introducirse vulnerabilidades sin que el proveedor del software lo supiera, si desconoce que se está utilizando una versión vulnerable de un componente o servicio del software.
C.1.6 Las vulnerabilidades de los componentes y servicios de terceros se supervisan y gestionan de acuerdo con el Objetivo de Control 10.	**C.1.6.a** El evaluador deberá evaluar las evidencias para confirmar que los componentes y servicios de terceros presentes en y/o en uso por el software son monitoreados regularmente en busca de vulnerabilidades de acuerdo con el Objetivo de Control 10.1.	Las vulnerabilidades en los componentes y servicios de terceros deben ser manejadas de la misma manera que las vulnerabilidades en el código controlado por el proveedor. Deben supervisarse en busca de vulnerabilidades mediante pruebas y/o el seguimiento de los repositorios de divulgación de vulnerabilidades disponibles públicamente y gestionarse de forma que cualquier vulnerabilidad conocida en esos componentes y servicios se parchee, o se mitigue de otro modo, lo antes posible. No parchear o mitigar una vulnerabilidad en los componentes o servicios de terceros puede tener las mismas ramificaciones que no parchear o mitigar una vulnerabilidad en el propio código del proveedor del software de pago.
	C.1.6.b El evaluador deberá evaluar la evidencia para confirmar que las vulnerabilidades en los componentes y servicios de terceros se identifican y se parchean o se mitigan de otra manera de manera oportuna de acuerdo con el Objetivo de Control 10.2.	

Objetivos de Control	Requisitos de las Pruebas	Guía
C.1.7 Cuando los componentes y/o recursos de software se alojan o mantienen en sistemas de terceros, como las redes de distribución de contenidos (CDN), la autenticidad de dichos componentes y recursos se verifica cada vez que se obtienen.	**C.1.7.a** Cuando los componentes y recursos de software se obtengan de repositorios externos y/o de terceros, el evaluador deberá evaluar la evidencia para confirmar que la autenticidad del componente de software se verifica cada vez que se obtiene el componente.	Es una técnica de diseño arquitectónico común en las aplicaciones web modernas para descargar u "obtener" componentes y recursos de terceros (por ejemplo, archivos, scripts, hojas de estilo, paquetes y bibliotecas) que se encuentran en repositorios de código disponibles públicamente (como archivos públicos) redes de entrega de contenido) en el momento en que se necesitan, en lugar de incorporar y mantener esos componentes y recursos en repositorios de códigos locales. Esta técnica proporciona muchos beneficios, entre ellos la capacidad de desplegar automáticamente actualizaciones de los componentes y recursos de terceros sin tener que volver a compilar necesariamente el código. Desafortunadamente, hay algunos inconvenientes significativos en este enfoque. Los repositorios de código de terceros son el objetivo principal de los atacantes porque les permite comprometer potencialmente numerosas aplicaciones y entidades al comprometer un solo paquete, biblioteca, secuencia de comandos o función. Por ejemplo, si una persona maliciosa pudiera comprometer estos repositorios o reemplazar una biblioteca de JavaScript ampliamente utilizada con una versión modificada, entonces esa biblioteca malintencionada podría propagarse automáticamente a todos los usuarios de esa biblioteca sin su conocimiento. Para mitigar el riesgo de obtener versiones maliciosas de código de repositorios de terceros, los proveedores de software de pago deben validar la autenticidad de dichos componentes antes de que sean recuperados (y/o cargados) por la aplicación de llamada. Hay numerosas formas de conseguirlo, entre ellas, el método más común de verificar la autenticidad de un componente es mediante criptografía robusta y firmas digitales.
	C.1.7.b El evaluador deberá probar el software para confirmar que la autenticidad de todos los componentes y recursos de software obtenidos de sistemas o repositorios de terceros se verifica cada vez que son obtenidos por el software.	

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control C.2: Controles de Acceso al Software Web Los controles de seguridad del software se implementan para restringir el acceso a las interfaces, funciones y recursos accesibles por Internet únicamente a los usuarios explícitamente autorizados.		
C.2.1 Se autentica el acceso de los usuarios a las funciones confidenciales y recursos confidenciales expuestos a través de interfaces accesibles por Internet.	**C.2.1** Usando la información obtenida en los Requisitos de Prueba 1.2.a y 2.1.a en los Requisitos Básicos, el evaluador deberá evaluar la evidencia para identificar todas las funciones sensibles y los recursos sensibles expuestos a través de interfaces accesibles por Internet.	Escribir funciones de autenticación personalizadas no es un asunto trivial. Existen numerosas cuestiones y consideraciones que deben tenerse en cuenta en el diseño y la implementación de dichas funciones, entre las que se incluye el hecho de que son un objetivo importante para los atacantes. Las funciones de autenticación deben estar libres de debilidades en su diseño y deben ser resistentes a ataques específicos. Dada la importancia y la gran dependencia de dichas funciones para fines de seguridad (y los de este estándar), se recomienda que las entidades utilicen funciones, módulos, bibliotecas, servicios de autenticación de terceros, etc., que ya se utilicen ampliamente en el sector y que hayan sido sometidos a pruebas y escrutinios de seguridad exhaustivos. Cuando el uso de estos mecanismos no sea factible, podrán utilizarse métodos personalizados. Sin embargo, los métodos personalizados deben diseñarse e implementarse en estricta conformidad con los estándares aplicables de la industria o las mejores prácticas para la autenticación. No hacerlo podría exponer vulnerabilidades o debilidades de diseño en los métodos de autenticación personalizados a entidades malintencionadas que podrían explotar esas vulnerabilidades para manipular o eludir de otro modo los mecanismos de autenticación personalizados.

Objetivos de Control	Requisitos de las Pruebas	Guía
C.2.1.1 Los métodos implementados para autenticar el acceso de los usuarios a funciones confidenciales y recursos confidenciales utilizan los mecanismos estándar de la industria**.**	**C.2.1.1.a** El evaluador deberá evaluar la evidencia para identificar todos los métodos implementados por el software para autenticar el acceso a funciones confidenciales y recursos confidenciales.	Al igual que el desarrollo de mecanismos de autorización propios, el desarrollo de mecanismos de autenticación personalizados puede ser una empresa bastante compleja. Gran parte de la seguridad de una aplicación depende de la fuerza y robustez de sus mecanismos de autenticación y autorización. Existen numerosas cuestiones y consideraciones que deben tenerse en cuenta en el diseño y la implementación de tales funciones, incluido, entre otros, el hecho de que son un objetivo importante para los atacantes. Las funciones de autenticación deben estar libres de puntos débiles y ser capaces de resistir ataques específicos. Por esta razón, sólo deben utilizarse mecanismos bien diseñados y probados. En general, se entiende que los mecanismos de autenticación proporcionados por proveedores aceptados por la industria y ampliamente adoptados por ésta han sido sometidos a pruebas y validaciones sustanciales a lo largo de dicha adopción. Por lo tanto, se recomienda encarecidamente que estas entidades utilicen estos mecanismos en lugar de escribir e implementar sus propios mecanismos. Cuando el uso de mecanismos de terceros no sea factible, podrán utilizarse métodos personalizados. Sin embargo, los métodos personalizados deben diseñarse e implementarse en estricta conformidad con los estándares aplicables de la industria o las mejores prácticas para la autenticación. No hacerlo podría exponer vulnerabilidades o debilidades de diseño en los métodos de autenticación personalizados a entidades malintencionadas que podrían explotar esas vulnerabilidades para manipular, o eludir de otro modo, los mecanismos de autenticación personalizados, inutilizando de hecho todas esas funciones.
	C.2.1.1.b El evaluador deberá evaluar la evidencia para confirmar que los métodos implementados utilizan mecanismos estándar de la industria que son: • Proporcionados por proveedores terceros conocidos y aceptados por la industria; o bien • Diseñados e implementados de acuerdo con los estándares aplicables de la industria o las mejores prácticas.	
	C.2.1.1.c Cuando las sesiones se utilicen para autenticar el acceso del usuario a funciones y recursos confidenciales, el evaluador deberá evaluar la evidencia para confirmar que las sesiones se manejen de acuerdo con los estándares reconocidos por la industria y las mejores prácticas para la gestión de sesiones seguras.	
	C.2.1.1.d Cuando se utilicen tokens (por ejemplo, tokens de acceso y tokens de actualización) para autenticar el acceso de los usuarios a funciones y recursos confidenciales, el evaluador deberá evaluar la evidencia para confirmar que los tokens se manejen de acuerdo con los estándares reconocidos por la industria y las mejores prácticas para la gestión segura de tokens.	

Objetivos de Control	Requisitos de las Pruebas	Guía
C.2.1.2 Los métodos implementados para autenticar el acceso de los usuarios a funciones sensibles y recursos sensibles a través de interfaces accesibles por Internet son lo suficientemente fuertes y robustos para proteger las credenciales de autenticación de acuerdo con el Objetivo de Control 5.3.	**C.2.1.2** Usando la información obtenida en el Requisito de Prueba C.2.1.1.a, el evaluador deberá evaluar la evidencia para confirmar que los métodos de autenticación implementados son lo suficientemente fuertes y robustos para proteger las credenciales de autenticación de acuerdo con el Objetivo de Control 5.3 en la sección de los Requisitos Básicos.	Los métodos de autenticación fuertes y robustos son aquellos resistentes a los ataques habituales. Ejemplos de tales métodos incluyen, entre otros, la autenticación multifactor y/o los métodos de autenticación que emplean criptografía robusta (como firmas digitales o certificados).
C.2.1.3 Las decisiones de autenticación se aplican dentro de un área segura del software.	**C.2.1.3.a** El evaluador deberá evaluar la evidencia para identificar en qué parte de la arquitectura del software se aplican las decisiones de autenticación.	Al igual que las decisiones de autorización, las de autenticación deben aplicarse dentro de un área segura del software. Los métodos de autenticación nunca deben depender únicamente de scripts o datos obtenidos del cliente o del navegador. Dicho esto, está permitido utilizar scripts y datos del lado del cliente cuando se combinan con métodos del lado del servidor para reforzar las capacidades de autenticación.
	C.2.1.3.b El evaluador deberá evaluar la evidencia para confirmar que todas las decisiones de autenticación se aplican dentro de un área segura de la arquitectura del software.	
	C.2.1.3.c El evaluador deberá evaluar la evidencia y probar el software para confirmar que las funciones, los scripts y los datos del lado del cliente o basados en el navegador nunca se basen únicamente en ellos para fines de autenticación.	
C.2.2 El acceso a todas las interfaces accesibles por Internet está restringido únicamente a los usuarios explícitamente autorizados.	**C.2.2.a** Usando la información obtenida en el Requisito de Prueba 2.1.a en la sección de Requisitos Básicos, el evaluador deberá evaluar la evidencia para identificar todas las interfaces de software que están expuestas a Internet o que pueden configurarse de manera que las expongan a Internet.	Las aplicaciones web modernas, en particular las que dependen en gran medida de API, microservicios y entornos sin servidor, requieren capacidades de control de acceso de grano fino para manejar las relaciones cada vez más complejas entre usuarios de software, interfaces, funciones y recursos.
	C.2.2.b El evaluador deberá evaluar la evidencia para identificar todos los métodos utilizados para autorizar el acceso a las interfaces accesibles por Internet.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	C.2.2.c El evaluador deberá evaluar la evidencia y probar el software para confirmar que cada uno de estos métodos es: • implementado correctamente; • apropiado para los tipos de usuarios que se espera que utilicen la interfaz; y • no expone vulnerabilidades conocidas.	Una diferencia clave entre las aplicaciones web "monolíticas" tradicionales y las aplicaciones web modernas es el grado en que una aplicación está expuesta (o potencialmente expuesta) a Internet. Mientras que las aplicaciones web monolíticas tienden a mantener las interacciones entre los componentes de software confinadas a un único contexto de seguridad (como un sistema o red internos o aislados), las aplicaciones web modernas suelen estar segmentadas en muchas funciones de software distintas y/o independientes que luego se exponen a Internet a través de API para que puedan ser accesibles a otras aplicaciones o usuarios, independientemente de dónde residan. Desafortunadamente, cada interfaz accesible por Internet (y las funciones y recursos que proporciona) es un vector de ataque potencial. Para mitigar los riesgos asociados a la exposición de tantas funciones de software a Internet, cada interfaz debe implementar mecanismos de control de acceso para garantizar que sólo los sistemas y usuarios autorizados puedan acceder a la interfaz, y a las funciones y recursos expuestos a través de esas interfaces.
	C.2.2.d Cuando los métodos utilizados para autorizar el acceso a las interfaces accesibles por Internet sean configurables por el usuario, o requieran la participación o interacción del usuario, el evaluador deberá evaluar la evidencia para confirmar que una guía adecuada esté disponible para las partes interesadas de acuerdo con el Objetivo de control 12.1 que describe las opciones configurables disponibles y cómo configurar cada método de forma segura.	
	C.2.2.e Cuando los métodos utilizados para autorizar el acceso a las interfaces accesibles a través de Internet estén configurados y controlados por la entidad evaluada, el evaluador deberá evaluar la evidencia para confirmar que el acceso a las interfaces accesibles a través de Internet está restringido a un conjunto apropiado de usuarios (o entidades) explícitamente autorizados.	
	C.2.2.f El evaluador deberá evaluar la evidencia y probará el software para confirmar que el acceso a todas las interfaces accesibles por Internet está restringido únicamente a los usuarios explícitamente autorizados.	

Objetivos de Control	Requisitos de las Pruebas	Guía
C.2.3 El acceso a todas las funciones y recursos del software expuestos a través de interfaces accesibles por Internet está restringido únicamente a usuarios explícitamente autorizados.	**C.2.3** Utilizando la información obtenida en el Requisito de Prueba C.2.2.a, el evaluador deberá evaluar la evidencia para identificar todas las funciones y recursos de software que están expuestos, o que pueden ser configurados de manera que se expongan, a través de interfaces accesibles por Internet.	Además de controlar el acceso a nivel de interfaz, también debe controlarse el acceso a las funciones y recursos individuales proporcionados a través de cada interfaz accesible por Internet. Las necesidades de acceso a las diferentes funciones y recursos dentro de una interfaz dada pueden ser bastante complejas dependiendo de los tipos de usuarios y sistemas que necesiten utilizar una interfaz determinada y de las diferentes capacidades y datos accesibles a través de esas funciones y recursos.
C.2.3.1 El software garantiza la aplicación de las normas de control de acceso tanto a nivel de función como de recurso con capacidades de control de acceso de grano fino.	**C.2.3.1.a** Utilizando la información obtenida en el Requisito de Prueba C.2.3, el evaluador deberá evaluar la evidencia para determinar cómo el software controla el acceso a las funciones y recursos individuales expuestos (o potencialmente expuestos) a través de interfaces accesibles por Internet.	Para apoyar las necesidades de control de acceso de grano fino de las modernas arquitecturas de aplicaciones web y garantizar que los usuarios sólo puedan acceder a las funciones y recursos de software que están autorizados a utilizar, el software debe apoyar la capacidad de definir y aplicar normas de control de acceso en distintos "niveles" dentro de la jerarquía de la interfaz, incluido el nivel o niveles de funciones y recursos individuales. Dependiendo de los tipos de funciones y recursos expuestos en una determinada interfaz de software, los métodos utilizados para autorizar el acceso a nivel de interfaz pueden no ser apropiados para proporcionar acceso a funciones y recursos individuales expuestos a través de dichas interfaces.
	C.2.3.1.b El evaluador deberá luego evaluar la evidencia para identificar los métodos utilizados para restringir el acceso a las funciones y recursos expuestos (o potencialmente expuestos) a través de interfaces accesibles por Internet y para confirmar que cada uno de estos métodos es: • implementado correctamente; • apropiado para el tipo de función(es) y recurso(s) provisto(s); y • no expone vulnerabilidades conocidas.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	C.2.3.1.c Cuando los métodos utilizados para autorizar el acceso a las funciones y los recursos expuestos (o potencialmente expuestos) a través de interfaces accesibles a través de Internet sean configurables por el usuario o requieran de otro modo su intervención o interacción, el evaluador deberá evaluar la evidencia para confirmar que se pone a disposición de las partes interesadas una guía, de conformidad con el Objetivo de Control 12.1, que describa los mecanismos y las opciones configurables disponibles para restringir el acceso a las funciones y los recursos expuestos a través de estas interfaces, y cómo configurar dichos mecanismos.	Por ejemplo, las claves API se utilizan a menudo para autorizar el acceso a una API para una entidad particular (también denominada autorización a nivel de proyecto o de la entidad). Aunque las claves API pueden ser adecuadas para autorizar este nivel de acceso a una API, es posible que no lo sean para autorizar el acceso de usuarios individuales a funciones o recursos específicos expuestos (o potencialmente expuestos) a través de la API. Cuando sea necesario un control de acceso de grano fino, los métodos implementados para controlar el acceso a todas las funciones y recursos de software expuestos a través de interfaces accesibles por Internet deben ser apropiados para los tipos de autorización(es) obligatorios (por ejemplo, el usuario versus la entidad) y las funciones y recursos involucrados (funciones y recursos confidenciales versus los no confidenciales). Siempre que sea obligatorio que los usuarios finales configuren las autorizaciones y permisos de control de acceso para funciones y recursos individuales expuestos a través de interfaces accesibles por Internet, el proveedor de software debe proporcionar una guía (o hacer accesible de otro modo una guía) a los usuarios y otras partes interesadas para explicarles cómo configurar dichos permisos y alertarles sobre consideraciones de seguridad importantes a la hora de configurar las opciones y parámetros disponibles.
	C.2.3.1.d Cuando los métodos utilizados para autorizar el acceso a las funciones y recursos expuestos (o potencialmente expuestos) a través de interfaces accesibles por Internet estén configurados y controlados por la entidad evaluada, el evaluador deberá evaluar la evidencia para confirmar que el acceso a las funciones y recursos está restringido a un conjunto apropiado de usuarios explícitamente autorizados.	
	C.2.3.1.e El evaluador deberá evaluar la evidencia y probará el software para confirmar que los métodos utilizados para restringir el acceso a las funciones y recursos expuestos (o potencialmente expuestos) a través de interfaces accesibles por Internet requieren que los usuarios estén explícitamente autorizados antes de que se les conceda dicho acceso.	

Objetivos de Control	Requisitos de las Pruebas	Guía
C.2.3.2 Se aplican normas de autorización a cada solicitud de usuario para acceder a funciones y recursos del software a través de interfaces accesibles por Internet**.**	**C.2.3.2.a** Utilizando la información obtenida en el Requisito de Prueba C.2.3.1.a, el evaluador deberá evaluar la evidencia para confirmar que se realizan verificaciones de autorización cada vez que los usuarios solicitan acceso a una función o recurso expuesto (o potencialmente expuesto) a través de interfaces accesibles por Internet para verificar que están autorizados para la función, el recurso y el tipo de acceso solicitado.	La mayoría de las aplicaciones web modernas, en particular las que utilizan API, microservicios y arquitecturas sin servidor, funcionan sobre la base de solicitud/respuesta. Cada vez que un usuario desea realizar una función o acceder a los datos de una aplicación, envía una solicitud a la aplicación (normalmente a través de una API o similar) para acceder a una función o recurso concreto. A continuación, el software procesa esa solicitud y, si está autorizado, ejecuta la función solicitada y/o devuelve los datos solicitados. A menudo es trivial para los atacantes obtener las credenciales de acceso de los usuarios autorizados. Una estrategia de defensa en profundidad es esencial para garantizar que sólo los usuarios autorizados puedan acceder a las funciones y recursos protegidos. Cuando se combinan con otros controles de seguridad, como la caducidad de las sesiones o tokens después de un período de tiempo relativamente corto, las comprobaciones de autorización pueden limitar significativamente lo que un atacante puede hacer si es capaz de comprometer las credenciales de un usuario autorizado.
	C.2.3.2.b El evaluador deberá evaluar la evidencia y probar el software para confirmar que las normas de control de acceso se aplican cada vez que un usuario intenta acceder a una función o recurso expuesto (o potencialmente expuesto) a través de interfaces accesibles por Internet.	
C.2.3.3 Las decisiones de control de acceso se aplican dentro de un área segura de la arquitectura del software.	**C.2.3.3.a** El evaluador deberá evaluar la evidencia para identificar en qué parte de la arquitectura del software se aplican las decisiones de autorización y control de acceso.	El software de pago nunca debe depender de servicios y funciones desconocidos o inseguros para fines relacionados con la seguridad. Las áreas o sistemas seguros son aquellos dentro de la arquitectura del software en los que se garantiza la integridad de los servicios y datos disponibles y, por lo tanto, se puede confiar en ellos.
	C.2.3.3.b El evaluador deberá evaluar la evidencia para confirmar que todas las decisiones de control de acceso se aplican dentro de un área segura de la arquitectura del software.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	C.2.3.3.c El evaluador deberá evaluar la evidencia y probar el software para confirmar que las funciones, los script y los datos del lado del cliente o basados en el navegador nunca se basan únicamente en ellos para fines de control de acceso.	Históricamente, las arquitecturas de las aplicaciones web consistían en componentes "del lado del cliente" y componentes "del lado del servidor". Las funciones del lado del cliente son las que suele realizar un navegador web común. Las funciones del lado del servidor son las que suelen realizar los servidores web, de aplicaciones y/o de bases de datos. Dada la naturaleza abierta y el diseño de los navegadores web más comunes y el hecho de que son mantenidos por los usuarios finales, las funciones del lado del servidor se consideran normalmente más seguras dada la capacidad de un proveedor de software/servicios para controlar y asegurar esos aspectos de la arquitectura del software. Sin embargo, las arquitecturas modernas de software web se han vuelto cada vez más complejas, con componentes de software a menudo desplegados en múltiples ubicaciones geográficas y gestionados por múltiples entidades. En estas circunstancias, la distinción entre funciones "del lado del cliente" o "del lado del servidor" puede ser cada vez más ambigua. El término "área segura" es una referencia a las funciones tradicionales del "lado del servidor" sin entrar en especificaciones arquitectónicos. Ejemplos de un área segura incluyen un entorno de servidor seguro, un microservicio o una API sin servidor.

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control C.3: Mitigación de Ataques del Software Web Se implementan controles de seguridad del software para mitigar los ataques habituales a las aplicaciones web.		
C.3.1 El software aplica o apoya el uso de los Encabezados de Seguridad HTTP más recientes para proteger las interfaces accesibles de Internet de los ataques.	**C.3.1.a** El evaluador deberá evaluar la evidencia para confirmar que el software apoya el uso de los Encabezados de Seguridad HTTP más recientes, y para determinar las opciones disponibles y cómo se configuran dichos ajustes.	Los Encabezados de Seguridad HTTP son un conjunto de opciones de configuración relacionadas con la seguridad disponibles en los servidores web más comunes. Algunos ejemplos son el Encabezado X-Frame-Options, el encabezado HTTP-Strict-Transport-Security y el encabezado Content-Security-Policy. El uso de estas opciones puede proteger contra una variedad de diferentes tipos de ataques incluyendo cross-site scripting, clickjacking, y ataques de falsificación de petición de cross-site. Aunque el apoyo a los Encabezados de Seguridad HTTP específicos puede diferir dependiendo de la plataforma subyacente o de la tecnología de software, estas opciones están ampliamente disponibles y deben ser habilitadas y configuradas con la configuración más segura posible para una implementación dada.
	C.3.1.b Cuando los Encabezados de Seguridad HTTP estén configurados y controlados por el proveedor del software, el evaluador deberá evaluar la evidencia para confirmar que el software está configurado para utilizar los últimos Encabezados de Seguridad HTTP disponibles y que los ajustes de configuración son razonables y están justificados.	
	C.3.1.c Cuando sea obligatoria la participación o interacción del usuario para configurar los Encabezados de Seguridad HTTP, el evaluador deberá evaluar la evidencia para confirmar que se pone a disposición de las partes interesadas una guía de acuerdo con el Objetivo de Control 12.1 que describa los Encabezados de Seguridad HTTP apoyados por el software y cómo configurar dichos ajustes.	
C.3.2 Nunca se confía en los datos de entrada procedentes de fuentes no fiables y se implementan controles de seguridad del software para mitigar la explotación de vulnerabilidades a través de la manipulación de los datos de entrada.	**C.3.2.a** Utilizando la información obtenida en el Requisito de Prueba C.2.1.a, el evaluador deberá evaluar la evidencia para identificar todas las interfaces que aceptan la entrada de datos de fuentes no fiables.	Muchas vulnerabilidades en el software y los sistemas se exponen cuando los datos de entrada suministrados por una fuente no confiable son inherentemente confiables para el software y se procesan sin garantizar primero la seguridad de los datos. Las fuentes no fiables son aquellas que residen en un contexto de seguridad diferente al de la API o el sistema que recibe y procesa los datos de entrada. Ejemplos de una fuente no fiable podrían incluir un sistema, API o microservicio que resida en un entorno diferente, un sistema interno que resida en la misma red pero que se mantenga bajo una clasificación de seguridad inferior, o el navegador de un usuario. *(continúa en la página siguiente)*
	C.3.2.b Cuando el software acepte entradas de fuentes no fiables, el evaluador deberá evaluar la evidencia para identificar el formato o formatos de datos esperados por el software para cada campo de entrada y los analizadores sintácticos e intérpretes implicados en el procesamiento de los datos de entrada.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	C.3.2.c Usando la información obtenida en el Requisito de Prueba 4.1.a en la sección de Requisitos Básicos, el evaluador deberá evaluar la evidencia para determinar si los ataques que tienen como objetivo todos los analizadores sintácticos e intérpretes se reconocen en el modelo de amenazas.	Dos de los tipos de ataques más comunes, Inyección (SQL, XML, Código, Cadena, etc.) y Cross-Site Scripting (XSS), explotan la confianza del software en los datos de entrada proporcionados por fuentes no confiables para ejecutar el código malicioso o para forzar el software se comporte de manera no deseadas. Para protegerse contra estos y otros tipos de ataques relacionados, nunca se debe confiar en los datos de entrada y se deben implementar controles de seguridad del software para garantizar que los datos de entrada se validan, se convierten en seguros o se manejan de otro modo de forma que se mitigue la probabilidad y/o los impactos de la ejecución de datos de entrada maliciosos.
	C.3.2.d Cuando se reconozcan tales ataques y se use la información obtenida en el Requisito de Prueba 4.2.a en la sección de Requisitos Básicos, el evaluador deberá evaluar la evidencia para confirmar que los controles de seguridad del software están definidos e implementados para mitigar los ataques que intentan explotar las vulnerabilidades a través de la manipulación de los datos de entrada.	
	C.3.2.e Cuando la implementación de los controles de seguridad del software sea configurable o requiera de otro modo la intervención o interacción del usuario, el evaluador deberá a evaluar la evidencia para confirmar que la guía está disponible para las partes interesadas de acuerdo con el Objetivo de Control 12.1 que describe cómo configurar adecuadamente dichos controles de seguridad.	
C.3.2.1 Se utilizan métodos estándar de la industria para proteger las entradas de software de ataques que intentan explotar vulnerabilidades a través de la manipulación de los datos de entrada.	**C.3.2.1.a** Usando la información obtenida en el Requisito de Prueba 4.2.a en la sección de Requisitos Básicos, el evaluador deberá evaluar la evidencia para identificar todos los controles de seguridad del software implementados para mitigar los ataques que intentan explotar las vulnerabilidades mediante la manipulación de los datos de entrada.	Existe una gran variedad de métodos y técnicas que pueden utilizarse para proteger las entradas de software contra la inyección y otros tipos de ataques similares. El método más a menudo asociado a estas protecciones es la "validación de entradas." Sin embargo, la validación de entrada es difícil de implementar correctamente, especialmente cuando se trata de datos de entrada complejos, como direcciones URL, XML, JSON, objetos serializados, etc. Por lo tanto, la validación de entradas no es apropiada como defensa principal contra los ataques de manipulación de entradas. *(continúa en la página siguiente)*

Objetivos de Control	Requisitos de las Pruebas	Guía
	C.3.2.1.b El evaluador evaluará las evidencias para confirmar que los métodos implementados para protegerse contra dichos ataques utilizan mecanismos y/o técnicas estándar de la industria que son: • Proporcionados por proveedores terceros conocidos y aceptados por la industria; o bien • Diseñados e implementados de acuerdo con los estándares aplicables de la industria o las mejores prácticas.	Otros métodos, como la parametrización y el escape de salida, son más adecuados como mecanismos de defensa primarios. Aunque el tipo y la complejidad de los datos de entrada y la forma en que se espera utilizarlos dictan a menudo los métodos más apropiados para una entrada determinada, debe recurrirse a la parametrización siempre que sea posible. El escape de salida puede utilizarse como alternativa si la parametrización no es factible. El uso de la validación de entrada puede utilizarse como defensa secundaria, cuando proceda, para proporcionar una defensa en profundidad. Al igual que ocurre con otras funciones críticas como la autenticación y la autorización, los métodos de protección de las entradas deben aprovechar los mecanismos de terceros aceptados por la industria siempre que sea posible. Si el uso de tales mecanismos no es factible, podrán utilizarse métodos personalizados si se diseñan e implementan de acuerdo con los estándares aplicables del sector o las mejores prácticas.
	C.3.2.1.c El evaluador deberá evaluar la evidencia y probar el software para confirmar que los métodos implementados: • Se apliquen correctamente de acuerdo con las guías disponibles, y • No exponga ninguna vulnerabilidad.	
C.3.2.2 Los analizadores sintácticos y los intérpretes se configuran con la configuración más restrictiva posible.	**C.3.2.2.a** Utilizando la información obtenida en el Requisito de Prueba C.3.2.b, el evaluador deberá evaluar la evidencia para identificar las configuraciones para cada analizador sintáctico o intérprete utilizado para procesar datos de entrada no confiables.	En algunos casos, puede que no sea factible aislar (parametrizar) o modificar (escapar, codificar, etc.) los datos de entrada antes de procesarlos. En tales casos, el único método viable para protegerse contra los ataques de manipulación de entrada es usar un analizador sintáctico o intérprete que se haya reforzado para evitar tales ataques. Por ejemplo, en el momento de esta publicación la única forma viable de protegerse contra un ataque de Entidad Externa XML es configurar el analizador sintáctico XML para desactivar la función de Definición del Tipo de Documento (DTD), también conocida como función de Entidades Externas. *(continúa en la página siguiente)*

Objetivos de Control	Requisitos de las Pruebas	Guía
	C.3.2.2.b Para cada uno de los analizadores sintácticos/intérpretes y las configuraciones identificadas, el evaluador deberá evaluar la evidencia para confirmar que los analizadores sintácticos y los intérpretes están configurados con el conjunto más restrictivo de capacidades factible y que las configuraciones están justificadas y son razonables. Cuando ciertas características del analizador sintáctico/intérprete no puedan configurarse de forma segura, el evaluador deberá evaluar la evidencia para confirmar que se han implementado otros métodos para mitigar la falta de configuraciones seguras y para proteger aún más contra la ejecución de comandos maliciosos.	La configuración específica que debe desactivarse/activarse para protegerse contra determinados ataques depende de los analizadores sintácticos y los intérpretes. Para más información, consulte la guía de seguridad disponible sobre los analizadores sintácticos/intérpretes específicos en uso. Cuando ciertas características de los analizadores sintácticos o los intérpretes no puedan configurarse con los ajustes más seguros posibles, el procesamiento de los datos de entrada no fiables deberá utilizar técnicas como el sandboxing para evitar (o mitigar de otro modo los efectos de) la ejecución de código malicioso.
C.3.3 Los controles de seguridad del software se implementan para proteger las interfaces de software de los ataques por agotamiento de recursos.	**C.3.3.a** Utilizando la información obtenida en los Requisitos de Prueba C.2.1.a y C.2.2, el evaluador deberá evaluar la evidencia para identificar todas las interfaces accesibles por Internet y las funciones y recursos expuestos (o potencialmente expuestos) a través de dichas interfaces para identificar dónde dichas interfaces, funciones y recursos pueden ser susceptibles a ataques de agotamiento de recursos.	Mientras que la meta de muchos ataques es exponer datos confidenciales y funciones confidenciales (directa o indirectamente) a usuarios no autorizados, otros ataques pretenden impedir el uso o el acceso de una aplicación a recursos informáticos importantes. Estos ataques tienen como objetivo abrumar al software/sistema con peticiones o llenar todos los recursos disponibles del sistema, como el tiempo de procesamiento o la memoria, privando así al software/sistema de los recursos que requiere para su funcionamiento normal e inutilizable para otros usuarios. En otros casos, estos ataques pretenden forzar al software a comportarse de formas no deseadas que podrían, a su vez, permitir a un atacante ejecutar código arbitrario en el sistema específico o exponer datos confidenciales a través de mensajes de error. *(continúa en la página siguiente)*
	C.3.3.b Cuando dichas interfaces, funciones y recursos sean potencialmente susceptibles a ataques de agotamiento de recursos, el evaluador deberá evaluar la evidencia para confirmarlo: • La amenaza de tales ataques se documenta de acuerdo con el Objetivo de Control 4.1, y • Los controles de seguridad del software para mitigar dichos ataques están documentados y se aplican de acuerdo con el Objetivo de Control 4.2.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	C.3.3.c El evaluador deberá evaluar la evidencia para confirmar que los controles de seguridad del software implementados para mitigar la falta de recursos y otros ataques similares en las interfaces accesibles por Internet están diseñados e implementados de acuerdo con los estándares y las mejores prácticas aplicables de la industria.	
	C.3.3.d Cuando la aplicación de los controles de seguridad del software sea configurable por el usuario o requiera de otro modo la participación o interacción del usuario, el evaluador deberá evaluar la evidencia para confirmar que está disponible a las partes interesadas una guía de acuerdo con el Objetivo de Control 12.1 que describa cómo configurar dichos mecanismos.	Algunos ejemplos de métodos utilizados para mitigar el riesgo de este tipo de ataques son la limitación de la tasa en el número de solicitudes que pueden enviarse en un periodo de tiempo determinado (limitación de tasa). Otros métodos para evitar estos ataques pueden incluir la definición de otros límites, como el número de usuarios y/o sistemas que pueden enviar solicitudes, la autenticación mutua de dichos usuarios y sistemas, o el uso de CAPTCHA u otras técnicas anti-automatización que puedan evitar que se envíen grandes volúmenes de solicitudes a las interfaces de software en un breve periodo de tiempo. En el caso del SaaS u otros entornos similares, también se pueden utilizar controles apropiados basados en la red para hacer frente a este tipo de ataques.

Objetivos de Control	Requisitos de las Pruebas	Guía
C.3.4 Los controles de seguridad del software se implementan para proteger las interfaces accesibles por Internet del contenido de archivos maliciosos.	**C.3.4.a** Utilizando la información obtenida en el Requisito de Prueba C.2.1.a, el evaluador deberá evaluar la evidencia para identificar todas las interfaces accesibles por Internet que aceptan la carga de archivos y los tipos de archivos permitidos.	Las cargas de archivos pueden utilizarse para proporcionar conjuntos de datos más grandes a un software. Sin embargo, estas cargas deben gestionarse de forma segura para evitar el uso indebido de esta función. Los archivos que no se gestionen correctamente pueden acabar siendo ejecutables en el sistema anfitrión o utilizarse como vector para infectar o subvertir el software u otros sistemas (por ejemplo, creando o sobrescribiendo archivos de configuración maliciosos). Las interfaces de carga de archivos también pueden proporcionar acceso no deseado al sistema host subyacente o al software. Los distintos tipos de archivos pueden estar provistos de diferentes permisos o funciones dentro de un sistema anfitrión, y cualquier sistema de carga de archivos debe garantizar que sólo se acepten para su carga los tipos de archivos esperados. Sin embargo, hay que tener cuidado de que este proceso añadido no exponga por sí mismo vulnerabilidades que puedan ser explotadas. Muchos formatos de archivo permiten incrustar otros archivos o datos que pueden "expandirse" al analizar el archivo fuente. En algunos escenarios, esto puede utilizarse para obtener privilegios o explotar vulnerabilidades en la plataforma anfitriona que de otro modo no sería posible. Los archivos cargados deben gestionarse de forma que se evite la explotación de ataques de análisis o expansión de archivos. *(continúa en la página siguiente)*
	C.3.4.b Cuando el programa acepte la carga de archivos a través de interfaces accesibles por Internet, el evaluador deberá evaluar la evidencia para confirmar que: • La amenaza de ataques a los mecanismos de carga de archivos está documentada de acuerdo con el Objetivo de Control 4.1, y • Los controles de seguridad del software para mitigar dichos ataques están documentados y se aplican de acuerdo con el Objetivo de Control 4.2.	
	C.3.4.c El evaluador evaluará la evidencia para confirmar que los controles de seguridad del software implementados para mitigar los ataques a los mecanismos de carga de archivos se implementan de acuerdo con los estándares aplicables de la industria y las mejores prácticas.	
	C.3.4.d El evaluador examinará las evidencias para confirmar que los controles de seguridad del software implementados para mitigar los ataques a los mecanismos de carga de archivos incluyen métodos para restringir los tipos de archivos permitidos por los mecanismos de carga de archivos.	
	C.3.4.e El evaluador deberá evaluar la evidencia para confirmar que los controles de seguridad del software implementados para mitigar los ataques a los mecanismos de carga de archivos incluyen métodos para restringir el número y tamaño máximos de archivos permitidos para la carga.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	C.3.4.f El evaluador deberá evaluar la evidencia para confirmar que los controles de seguridad del software implementados para mitigar los ataques a los mecanismos de carga de archivos tienen en cuenta el uso de formatos de archivo complejos o comprimidos que se utilizan a menudo para sobrecargar o explotar de otro modo los mecanismos de análisis de archivos.	Para evitar la explotación de los sistemas de carga de archivos, no se pueden asignar privilegios de escritura o ejecución a los archivos que se carguen. Los archivos que son obligatoriamente escribibles deben copiarse en un archivo separado administrado solo por el software para evitar que un usuario malintencionado explote el archivo entre la carga y el uso. Para una defensa en profundidad, algunos lenguajes y marcos de desarrollo de software incluyen la capacidad de realizar llamadas a los sistemas anti-malware para escanear estos archivos al cargarlos. Para obtener más información, consulte la documentación de terceros correspondiente. Los archivos y los archivos tipo analizadores sintácticos son fuentes notorias de exploits. Estos analizadores sintácticos no deben tomar decisiones de seguridad basadas en los nombres o las extensiones de los archivos. Los tipos de archivo aceptables deben tener una estructura básica que permita al software determinar el tipo de archivo sin utilizar nombres o extensiones de archivo.
	C.3.4.g El evaluador deberá evaluar la evidencia para confirmar que los controles de seguridad del software implementados para mitigar los ataques a los mecanismos de carga de archivos incluyen métodos para restringir el número y tamaño máximos de archivos permitidos para la carga.	
	C.3.4.h El evaluador deberá evaluar la evidencia para confirmar que el uso de mecanismos de análisis de archivos no depende de los nombres o extensiones de los archivos por motivos de seguridad.	
	C.3.4.i Cuando la aplicación de los controles de seguridad del software sea configurable por el usuario o requiera de otro modo la participación o interacción del usuario, el evaluador deberá evaluar la evidencia para confirmar que está disponible a las partes interesadas una guía de acuerdo con el Objetivo de Control 12.1 que describa cómo configurar dichos mecanismos.	

Objetivos de Control	Requisitos de las Pruebas	Guía
C.3.5 Los controles de seguridad del software se implementan para proteger las interfaces accesibles por Internet de la creación de objetos hostiles y la manipulación de datos.	**C.3.5.a** Utilizando la información obtenida en los Requisitos de Prueba C.2.1.a y C.2.2, el evaluador deberá evaluar la evidencia para identificar todas las funciones de software expuestas a través de interfaces accesibles por Internet que aceptan y procesan objetos de datos como entradas.	Algunas API de software aceptan objetos de datos serializados (por ejemplo, matrices, cookies, tokens, etc.) para pasar desde otros sistemas. Sin embargo, si no se colocan los métodos adecuados para restringir la deserialización y creación de objetos, las personas malintencionadas podrían utilizar estas API para lanzar ataques de denegación de servicio, comprometer los mecanismos de control de acceso o inyectar y ejecutar código malicioso en los sistemas subyacentes. Existen numerosos métodos para protegerse contra los ataques de serialización (y deserialización). Algunos lenguajes de programación, bibliotecas y API proporcionan características y funciones resistentes a los ataques de serialización. Otros métodos incluyen el uso de mecanismos de deserialización que sólo apoyan formatos de datos puros como JSON o XML, la limitación de los tipos de datos permitidos durante la creación de objetos, el cifrado de las comunicaciones y la autenticación de los clientes de la API. Los métodos apropiados para protegerse contra los ataques de serialización dependen de la implementación de la API. Consulte fuentes del sector, como el Proyecto Abierto de Seguridad de Aplicaciones Web (OWASP,) para obtener más información. Por las mismas razones explicadas en el último párrafo de la guía del Objetivo de Control C.3.4, los mecanismos de análisis de archivos no deben tomar decisiones de seguridad basadas en los nombres o las extensiones de los archivos. Los tipos de archivo aceptables deben tener una estructura básica que permita al software determinar el tipo de archivo sin utilizar nombres o extensiones de archivo. *(continúa en la página siguiente)*
	C.3.5.b Cuando el software acepte y procese objetos de datos como entradas, el evaluador deberá evaluar la evidencia para confirmar que: • La amenaza de creación de objetos hostiles y ataques de manipulación de datos se documenta de acuerdo con el Objetivo de Control 4.1, y • Los controles de seguridad del software para mitigar dichos ataques están documentados y se aplican de acuerdo con el Objetivo de Control 4.2.	
	C.3.5.c El evaluador deberá evaluar la evidencia para confirmar que los controles de seguridad del software implementados para mitigar la creación de objetos hostiles y los ataques de manipulación de datos se implementan de acuerdo con los estándares aplicables de la industria y las mejores prácticas.	
	C.3.5.d El evaluador deberá evaluar la evidencia para confirmar que los controles de seguridad del software implementados para mitigar los ataques de creación de objetos hostiles y de manipulación de datos incluyen métodos que restringen los formatos de archivo permitidos por los mecanismos de análisis sintácticos de archivos.	
	C.3.5.e El evaluador deberá evaluar la evidencia para confirmar que el uso de mecanismos de análisis sintácticos de archivos no depende de los nombres o extensiones de los archivos con fines de seguridad.	
	C.3.5.f El evaluador deberá evaluar la evidencia para confirmar que el uso de mecanismos de análisis sintáctico de archivos no expone otras vulnerabilidades.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	C.3.5.g Cuando el software acepte objetos serializados como entradas, el evaluador deberá evaluar la evidencia para confirmar que los controles de seguridad del software están implementados para proteger contra los ataques de deserialización y que dichos controles de seguridad se adhieren a los estándares aplicables de la industria y las mejores prácticas.	Algunos mecanismos de análisis sintáctico de archivos son inherentemente susceptibles a ciertas vulnerabilidades. Por ejemplo, los analizadores sintácticos XML suelen ser vulnerables a los ataques de la Entidad Externa. Del mismo modo, los analizadores sintácticos JSON son vulnerables a ataques en los que comandos inseguros, como eval(), pueden permitir la ejecución de código malicioso. Para mitigar los ataques que intentan explotar las vulnerabilidades de los mecanismos de análisis sintáctico de archivos, puede ser necesario que las entidades implementen controles de seguridad de software adicionales. Ejemplos de estos controles incluyen, entre otros, configurar los mecanismos de análisis sintáctico de archivos para que utilicen la configuración más restrictiva posible, evitar o escapar de ciertos comandos que son problemas conocidos para los mecanismos de análisis sintáctico de archivos, o aislar y ejecutar los comandos de análisis sintáctico de archivos en una caja de arena. Los métodos utilizados para mitigar aún más estos ataques deben tener en cuenta los analizadores sintácticos e intérpretes específicos en uso y aplicarse de forma adecuada para cada analizador sintáctico e intérprete.
	C.3.5.h Cuando los controles de seguridad del software implementados para proteger contra la creación de objetos hostiles y la manipulación de datos sean configurables por el usuario o requieran de otro modo la entrada o interacción del usuario, el evaluador deberá evaluar la evidencia para confirmar que se pone a disposición de las partes interesadas una guía de acuerdo con el Objetivo de Control 12.1 que describa cómo configurar dichos mecanismos.	

Objetivos de Control	Requisitos de las Pruebas	Guía
C.3.6 Se implementan controles de seguridad del software para proteger las interfaces accesibles por Internet de ataques que explotan el uso compartido de recursos de múltiples orígenes.	**C.3.6.a** El evaluador deberá evaluar la evidencia para determinar si y/o cómo el software apoya el acceso de origen cruzado a interfaces accesibles a través de Internet, y para confirmar que el acceso a las API y los recursos del software desde los scripts basados en el navegador está deshabilitado de manera predeterminada.	Puede ser obligatorio que el software permita el acceso a recursos o interfaces API de otros dominios u orígenes de Internet. Esta práctica puede dar lugar a vulnerabilidades que expongan los datos o funciones confidenciales a ataques. Cuando no sea obligatorio, deberá deshabilitarse el uso compartido de recursos de origen cruzado. Cuando sea necesario compartir recursos de origen cruzado debido a un propósito comercial legítimo, dicho acceso debe estar habilitado sólo para los dominios y orígenes obligatorios para que el software realice su(s) función(es) prevista(s). El uso de listas de permisos u otras configuraciones puede ser adecuado para identificar los orígenes permitidos, pero dichas configuraciones también deben estar protegidas contra modificaciones por parte de personas malintencionadas.
	C.3.6.b Cuando se habilite el acceso de origen cruzado, el evaluador deberá evaluar la evidencia para confirmar que las razones para habilitar el acceso de origen cruzado son razonables y están justificadas, y que el acceso está restringido al mínimo número de orígenes factible.	
	C.3.6.c El evaluador deberá probar el software para confirmar que las afirmaciones hechas por la entidad evaluada sobre el acceso de origen cruzado son válidas. Como mínimo, se espera que las pruebas incluyan pruebas funcionales utilizando herramientas/técnicas forenses.	
	C.3.6.d Cuando deshabilitar o restringir el acceso de origen cruzado a las API del software requiere la participación o la interacción del usuario, el evaluador deberá evaluar la evidencia para confirmar que se brinde la guía adecuada sobre este proceso a las partes interesadas de acuerdo con el Objetivo de control 12.1.	

Objetivos de Control	Requisitos de las Pruebas	Guía
Objetivo de Control C.4: Comunicaciones de Software Web Las transmisiones de datos confidenciales están aseguradas de acuerdo con el Objetivo de Control 6.		
C.4.1 Las transmisiones de datos confidenciales se cifran de acuerdo con los Objetivos de Control 6.2 y 6.3.	**C.4.1.a** Utilizando la información obtenida en el Requisito de Prueba 6.2.a, el evaluador deberá evaluar la evidencia para determinar cómo el software maneja las comunicaciones, incluidas las que se producen entre sistemas separados en la arquitectura general del software.	Los tipos de datos que pueden considerarse confidenciales pueden variar en función de las distintas implementaciones. Véase el Objetivo de Control 1.1 para más información sobre la identificación de datos confidenciales. Por lo tanto, es importante que cualquier conexión que transmita datos confidenciales esté cifrada utilizando una criptografía robusta. Los métodos comunes para conseguirlo incluirán el uso de TLS utilizando conjuntos de cifrado apropiados. Si bien las conexiones que no transmiten datos confidenciales no requieren explícitamente el uso de encriptación, se observa que el uso de criptografía robusta para asegurar todas las conexiones se considera una mejor práctica y debe implementarse para todas las comunicaciones a menos que existan restricciones comerciales o tecnológicas legítimas que hagan tal enfoque inviable. En la mayoría de los casos, sin embargo, las comunicaciones entre los componentes de la aplicación web incluyen la transmisión de información de autenticación (credenciales de usuario o información de sesión) que se consideran datos confidenciales por definición y, por lo tanto, deben cifrarse utilizando criptografía robusta.
	C.4.1.b Cuando el software permita o apoye de otro modo la transmisión de datos confidenciales entre usuarios y sistemas en diferentes contextos de seguridad, el evaluador deberá evaluar la evidencia para confirmar que todas esas comunicaciones están cifradas mediante criptografía robusta de acuerdo con los Objetivos de Control 6.2 y 6.3.	

Objetivos de Control	Requisitos de las Pruebas	Guía
	C.4.1.c Cuando se transmitan datos confidenciales mediante comunicaciones de servidor a servidor (por ejemplo, utilizando API), el evaluador deberá evaluar la evidencia para confirmar que el software aplica o apoya de otro modo la autenticación mutua entre sistemas.	Cuando se transmiten datos confidenciales entre sistemas que operan dentro de diferentes contextos de seguridad y/o diferentes entornos, es importante que dichas comunicaciones se restrinjan a una lista de sistemas explícitamente aprobada, y que los sistemas involucrados se autentiquen mutuamente de manera que los intentos de interceptar o comprometer dichos las comunicaciones se mitiguen adecuadamente. Cuando el proveedor del software controle la configuración de dichas comunicaciones, se debe hacer cumplir la autenticación mutua. De lo contrario, el proveedor de software debe proporcionar funciones para apoyar la autenticación mutua de sistemas dispares para que la entidad que implementa pueda configurar dichas funciones en consecuencia.
	C.4.1.d Cuando se utilicen certificados generados internamente o autofirmados para proteger las transmisiones de datos confidenciales, el evaluador deberá evaluar la evidencia para confirmar que: • El uso de certificados generados internamente o autofirmados es razonable y está justificado. • El software está configurado para aceptar el número mínimo factible de certificados generados internamente o autofirmados.	Muchas organizaciones que optan por utilizar certificados generados internamente y/o autofirmados lo hacen por los beneficios que ofrecen sin tener en cuenta el sobrecoste adicional necesaria para gestionarlos de forma segura. Como resultado, los procesos de seguridad críticos, como la revocación de certificados y la gestión de claves, no se implementan ni mantienen apropiadamente. Por esta razón, el uso de certificados generados internamente y/o autofirmados debe reducirse al mínimo absoluto. Cuando su uso sea obligatorio, estos casos deberán documentarse y justificarse.