Probador Certificado de ISTQB®

Programa de Estudio

Nivel Especialista

Ingeniero de Prueba de Seguridad

Versión ES V01.00

Traducción realizada por

Spanish Software Testing Qualifications Board

Traducción del Programa de Estudio

"ISTQB® Certified Tester - Security Test Engineer Syllabus, Version 1.0.1"

SSTQB
Spanish Software Testing Qualifications Board

ISTQB®
International Software
Testing Qualifications Board

Nota sobre Derechos de Propiedad Intelectual

Historial de Revisiones

Versión	Fecha	Observaciones
1.0	18/10/2024	Versión final.
Syllabus v2.0	31/01/2025	Versión actualizada y definitiva tras la revisión técnica

Historial de Revisiones – Traducción al idioma español		
Versión	**Fecha**	**Observaciones**
ES V 01.00	18/03/2025	Entrega traducción para su publicación.

Tabla de Contenidos

Agradecimientos

Este documento fue entregado formalmente por la Asamblea General del ISTQB® el 3 de mayo de 2024. Fue elaborado por Test Automation Task Force of the Specialist Working Group from the International Software Testing Qualifications Board: Dr. Frank Simon (chair), Alain Ribault, Gabriel Firmino Barjollo, Michael Pott, Beata Karpinska, Maria Kispal, Frans Dijkman.

Las siguientes personas participaron en la revisión, los comentarios y la votación de este programa de estudio:

Attila Toth, Claude Zhang, Dani Almog, Daniel Säther. Gary Mogyorodi, Haiying Liu (Sandy), Ingvar Nordström, Laura Albert, Nancy Thompson, Petr Zacek, Szilard Szell, Tal Pe'er, Tamás Gergely, Claude Zhang, Daniel van der Zwan, Dingguofu, Giancarlo Tomasig, Klaus Skafte, Markus Niehammer, Mattijs Kemmink, Meile Posthuma, Michael Stahl, Nicola Rosa, Samuel Ouko, Tauhida Parveen, Tetsu Nagata

Notas de la Versión en Idioma Español

Este “programa de estudio, de Nivel Especialista, Probador Certificado de ISTQB®, Ingeniero de Prueba de Seguridad, Versión ES V01.00” ha sido traducido por Spanish Software Testing Qualifications Board (SSTQB).

El equipo de traducción y revisión para este programa de estudio es el siguiente (por orden alfabético):

Responsable de la traducción: Gustavo Márquez Sosa (España)

El Comité Ejecutivo del SSTQB agradece toda aportación que permita mejorar esta traducción del programa de estudio.

En una siguiente versión se podrán incorporar aportaciones adicionales. El SSTQB considera conveniente mantener abierta la posibilidad de realizar cambios en el “Programa de Estudio”.

Madrid, 18 de marzo de 2025

0 Introducción

0.1 Objetivo de este programa de estudio

Este programa de estudio constituye la base para la formación como Probador Certificado del ISTQB® de Nivel Avanzado, Ingeniería de Automatización de la Prueba. El ISTQB proporciona este programa de estudio en los siguientes términos:

1. A los comités miembro, para que lo traduzcan a su idioma local y acrediten a los proveedores de formación. Los comités miembro pueden adaptar el programa de estudio a sus necesidades lingüísticas particulares y modificar las referencias para adaptarlas a sus publicaciones locales.
2. A los organismos de certificación, para que obtengan preguntas de examen en su idioma local adaptadas a los objetivos de aprendizaje de este programa de estudio.
3. A los proveedores de formación, para que elaboren material para formación y determinen los métodos de enseñanza adecuados.
4. A los candidatos a la certificación, para preparar el examen de certificación (ya sea como parte de un curso de formación o de forma independiente).
5. A la comunidad internacional de ingeniería de software y sistemas, para hacer avanzar la profesión de prueba de software y sistemas, y como fuente de libros y artículos.

0.2 Resultados de negocio

Esta sección enumera los resultados de negocio esperados de un candidato que haya obtenido la certificación de ingeniero de automatización de la prueba.

Un candidato que haya logrado la certificación de Ingeniero de automatización de la prueba puede...

STE-BO1	Comprender los paradigmas fundamentales de la seguridad y su impacto en la prueba de seguridad
STE-BO2	Utilizar y aplicar las técnicas de prueba de seguridad adecuadas y conocer sus puntos fuertes y sus limitaciones
STE-BO3	Contribuir a la planificación, el diseño y la ejecución de las pruebas de seguridad
STE-BO4	Comprender cómo pueden utilizarse los estándares de prueba de seguridad y las buenas prácticas de seguridad para probar la seguridad.
STE-BO5	Ajustar y realizar actividades de pruebas de seguridad de acuerdo con el contexto específico de la organización.
STE-BO6	Ajustar y realizar actividades de pruebas de seguridad de acuerdo con métodos de desarrollo y ciclos de vida de desarrollo del software específicos.
STE-BO7	Introducir los resultados de la prueba de seguridad en un sistema de gestión de la seguridad de la información (SGSI) para una gestión activa del riesgo de seguridad.
STE-BO8	Recopilar, evaluar y agregar los resultados de las pruebas y redactar un informe de prueba detallado que incluya todas las evidencias y hallazgos.

STE-BO9 Identificar los requisitos adecuados para las herramientas en función del enfoque de prueba de la seguridad requerido y ayudar a seleccionar las herramientas de prueba de la seguridad.

0.3 Objetivos de aprendizaje evaluables y nivel cognitivo de conocimiento

Los objetivos de aprendizaje apoyan los resultados de negocio y se utilizan para crear los exámenes Probador Certificado de ISTQB®, Nivel Ingeniero de Prueba de Seguridad.

En general, todos los contenidos de este programa de estudio son evaluables a nivel K1. Es decir, se puede pedir al candidato que reconozca, recuerde o rememore una palabra clave o un concepto mencionado en cualquiera de los nueve capítulos. Los niveles específicos de los objetivos de aprendizaje se muestran al principio de cada capítulo y se clasifican de la siguiente manera:

- K1: Recordar
- K2: Comprender
- K3: Aplicar
- K4: Analizar

En el Apéndice A se ofrecen más detalles y ejemplos de los objetivos de aprendizaje. Todos los términos enumerados como palabras clave y palabras clave específicas del dominio de la seguridad serán evaluables (K1), incluso si no se mencionen explícitamente en los objetivos de aprendizaje.

0.4 El examen del certificado de nivel avanzado de gestión de la prueba

El examen de certificación de Probador Certificado de ISTQB®, Nivel Ingeniero de Prueba de Seguridad se basará en este programa de estudio. El otro programa de estudio Analista de Prueba de Seguridad se concentra en el diseño de pruebas de seguridad que posteriormente ejecuta el Ingeniero de Prueba de Seguridad.

Las respuestas a las preguntas del examen pueden requerir el uso de material basado en más de una sección de este programa de estudio. Todas las secciones del programa de estudio son evaluables, excepto los apéndices. Se incluyen normas y libros como referencias, pero su contenido no es evaluable, más allá de lo que se resume en el propio programa de estudio a partir de dichas normas y libros.

Para más detalles, consulte el documento Estructuras y normas de examen para el ingeniero de pruebas de seguridad.

El criterio de acceso para presentarse al examen de ingeniero de pruebas de seguridad es que los candidatos tengan interés en las pruebas de software. Sin embargo, se recomienda que los candidatos también tengan al menos una formación mínima en desarrollo de software, pruebas de software o pruebas de seguridad.

Nota sobre los requisitos de acceso: Antes de presentarse al examen de certificación de Ingeniero de Pruebas de Seguridad deberá obtenerse el certificado de nivel básico ISTQB®.

La compleción de un curso de formación acreditado no es un requisito previo para el examen.

0.5 Acreditación

Un Comité Miembro de ISTQB® puede acreditar a los proveedores de formación cuyo material de curso siga este programa de estudio. Los proveedores de formación deberán obtener las directrices de acreditación del Comité Miembro u organismo que realice la acreditación. Un curso acreditado es reconocido como conforme a este programa de estudio y se le permite tener un examen ISTQB® como parte del curso.

Las directrices de acreditación para este programa de estudio son las Directrices de Acreditación genéricas publicadas por el Grupo de Trabajo de Gestión de Procesos y Conformidad del ISTQB®.

0.6 Tratamiento de los estándares

En el programa de estudios de Probador Certificado de ISTQB®, Nivel Ingeniero de Prueba de Seguridad se hace referencia a estándares (por ejemplo, NIST e ISO). El propósito de estas es proporcionar un marco de trabajo o proporcionar una fuente de información adicional si así lo desea el lector. Tenga en cuenta que el programa de estudio utiliza los estándares como referencia y que no están pensados para ser examinados. Consulte el capítulo 4 para obtener más información sobre el uso de estándares, buenas prácticas y normas.

0.7 Nivel de detalle

El nivel de detalle de este programa de estudio permite que los cursos y los exámenes sean consistentes a escala internacional. Para lograr este objetivo, el programa de estudio consta de:

- Objetivos generales de instrucción que describen la intención del programa de estudio de gestión de la prueba de nivel avanzado.
- Una lista de palabras clave que los alumnos deben ser capaces de recordar.
- Objetivos de aprendizaje para cada área de conocimiento, que describen el resultado cognitivo de aprendizaje que debe alcanzarse.
- Una descripción de los conceptos clave, incluyendo referencias a fuentes como la bibliografía aceptada o los estándares.

El contenido del programa de estudio no es una descripción de toda el área de conocimiento de la prueba de seguridad; refleja el nivel de detalle que debe cubrirse en los cursos de formación de ingeniero de prueba de seguridad a nivel de especialista. Se concentra en conceptos y técnicas de prueba que pueden aplicarse a todos los proyectos de software independientemente del ciclo de vida de desarrollo del software empleado.

0.8 Cómo está organizado este programa de estudio

Hay nueve capítulos con contenido evaluable. El encabezamiento de nivel superior de cada capítulo especifica el tiempo de formación del capítulo; no se proporciona cronología por debajo del nivel del capítulo. Para los cursos de formación acreditados, el programa de estudio exige un mínimo de 1290 minutos (unas 22 horas) lectivos, distribuidos en los nueve capítulos del siguiente modo:

1. Paradigmas de seguridad - 135 minutos
2. Técnicas de prueba de seguridad - 150 minutos

3. El proceso de prueba de seguridad - 120 minutos
4. Estándares y buenas prácticas de la prueba de seguridad - 195 minutos
5. Adaptación de la prueba de seguridad al contexto de la organización - 195 minutos
6. Adaptación de la prueba de seguridad a los modelos de ciclo de vida del desarrollo del software - 165 minutos
7. Prueba de seguridad como parte de un sistema de gestión de la seguridad de la información - 105 minutos
8. Suministro de información sobre los resultados de las pruebas de seguridad - 135 minutos
9. Herramientas de Prueba de Seguridad - 90 minutos

1 Paradigmas de seguridad - (K3)

Duración: 135 minutos

Palabras clave[1]

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB**: identificarán palabras clave del proceso de prueba
- **ESPDOM**: identificarán palabras clave **esp**ecíficas del **dom**inio: **SEGURIDAD**

Tipo Palabra Clave	Español	Inglés
ISTQB	disponibilidad	availability
ISTQB	confidencialidad	confidentiality
ISTQB	integridad	integrity
ESPDOM	software de código abierto	open-source software
ISTQB	prueba de seguridad	security testing
ISTQB	vulnerabilidad	vulnerability
ESPDOM	confianza cero	zero-trust

[1] Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.

Objetivos de aprendizaje para "Capítulo 1"

1.1. - Niveles de seguridad de activos

STE - 1.1.1 (K2) Explicar los distintos niveles de seguridad de los activos y su correspondiente nivel de protección.

STE - 1.1.2 (K2) Explicar la relación entre la sensibilidad de la información y la prueba de seguridad.

1.2. - Auditorías de seguridad

STE - 1.2.1 (K2) Describir el rol de la prueba de seguridad en el contexto de las auditorías de seguridad.

1.3. - El concepto de confianza cero

STE - 1.3.1 (K2) Explicar el concepto de confianza cero.

STE - 1.3.2 (K3) Aplicar la confianza cero en la prueba de seguridad.

1.4. - Software de código abierto (SCA)

STE - 1.4.1 (K2) Aportar un ejemplo del concepto de reutilización de software de código abierto en el desarrollo de software y su impacto en la prueba de seguridad

1.1 Niveles de seguridad de activos

Un activo es todo aquello que tiene valor en una organización y que, por lo tanto, requiere protección. Los activos permiten a las organizaciones operar procesos de negocio y tomar decisiones. Todos los activos de información y datos son vulnerables y, por lo tanto, deben protegerse. Los activos son objetos de la seguridad de la información. Los activos pueden ser personas, información, software, hardware, funciones, procesos e instalaciones de reputación corporativa [Chapple 2021]. Algunos ejemplos:

- activos software: sistema operativo, aplicaciones y bases de datos.
- activos de información: planes de negocio, documentación, inventos, fotografías y registros personales.
- activos hardware: sistemas informáticos, almacenamiento de datos y dispositivos de comunicación de datos.
- activos físicos: instalaciones.

1.1.1 Activos y su nivel de protección correspondiente

El valor de un activo viene determinado por tres pilares de la seguridad de la información (denominados la tríada CID): confidencialidad, integridad y disponibilidad [Stallings18].

La **confidencialidad** tiene como objetivo evitar la divulgación no autorizada de la información. Se produce una pérdida de confidencialidad cuando la información se revela a una parte o un sistema no autorizados [Stallings18].

La **integridad** tiene como objetivo evitar que los datos sean modificados o borrados por una parte no autorizada. [Stallings18].

La **disponibilidad** asegura que la información esté disponible cuando se necesite. [Stallings18].

Los tres pilares de la seguridad de la información pueden clasificarse en los siguientes niveles: alto, medio y bajo.

Los tres pilares de la seguridad de la información pueden clasificarse en los siguientes niveles: alto, medio y bajo. Cuanto más alto sea el nivel de seguridad, mayor será el requisito de que se despliegue el nivel de protección (salvaguardas). Las salvaguardas son funciones y restricciones de seguridad, seguridad del personal y seguridad de las estructuras, áreas y dispositivos físicos.

La pérdida de confidencialidad, integridad o disponibilidad puede repercutir en la operación de la organización, en los activos de la organización, en las personas, en los clientes o incluso en los países.

La clasificación de los activos define los niveles de sensibilidad y confidencialidad de estos. Esto ayuda a las organizaciones a implementar controles de seguridad y niveles de protección adecuados.

Si el activo tiene una sensibilidad baja, el nivel de protección podría establecerse en un nivel de seguridad bajo, y la prueba de seguridad podría no ser necesaria. El pilar adecuado para ello es la disponibilidad.

Un ejemplo de baja sensibilidad puede ser cuando la disponibilidad de la información es de mayor importancia para un conjunto de personas, como la información sobre el tráfico.

Si el activo tiene una sensibilidad media, el nivel de protección se fijará en un nivel de seguridad medio y será necesario realizar pruebas de seguridad. El pilar adecuado para ello es la integridad.

Un ejemplo de sensibilidad media puede ser cuando las personas necesitan información precisa de fuentes de confianza, como las autoridades.

Si el activo tiene una sensibilidad alta, el nivel de protección se fijará en un nivel de seguridad alto, y será necesario realizar pruebas de seguridad. El pilar adecuado para ello es la confidencialidad. Un ejemplo de alta sensibilidad puede ser la información personal sobre los empleados.

1.1.2 Sensibilidad de la información y la prueba de seguridad

La sensibilidad de la información representa el grado en que la información requiere protección para asegurar su confidencialidad, integridad y disponibilidad [Glosario del NIST]. Dado que las consecuencias de comprometer información sensible pueden ir de menores a desastrosas, debe realizarse una prueba de seguridad antes de que pueda tener lugar cualquier forma de compromiso de información.

El propósito de la prueba de seguridad es verificar que una implementación protege los datos y mantiene la funcionalidad según lo previsto. Cuanta más protección necesite un activo, más acciones de seguridad se necesitarán. La prueba de seguridad ayuda a identificar los puntos débiles y las vulnerabilidades, y comprueba si se han implementado los controles de seguridad adecuados.

La prueba de seguridad no puede garantizar que un sistema o una organización estén libres de vulnerabilidades. Tales pasos incluyen la realización de la prueba de seguridad para lograr los siguientes objetivos:

- Evaluar la efectividad de los controles de seguridad existentes.
- Descubrir debilidades y vulnerabilidades
- Establecer una estrategia de prueba de seguridad que incluya pruebas de confirmación para seguir el avance de los parches de software y las actualizaciones a largo plazo.

Para el ingeniero de prueba de seguridad (IPS), una evaluación del riesgo de seguridad realizada desde una perspectiva de sensibilidad de la información puede ser una importante fuente de información a partir de la cual se pueden planificar y diseñar pruebas de seguridad. Además, una evaluación del riesgo de seguridad puede utilizarse para priorizar la prueba de seguridad, de modo que se puedan determinar los riesgos y las prioridades, y las que tengan los niveles de riesgo más altos se sometan a prueba más rigurosas. La evaluación de riesgos es solo una instantánea en el momento actual y puede basarse en información limitada.

1.2 Auditorías de seguridad

Una auditoría de seguridad es una revisión y examen independiente de la seguridad del sistema de información de una organización, controlando su conformidad con un conjunto de criterios establecidos. Las auditorías tienen por objeto determinar la adecuación de los controles del sistema, asegurar la conformidad con una política y procedimientos de seguridad establecidos. Pero también, detectar brechas en los servicios de seguridad y recomendar los cambios que estén indicados para aplicar contramedidas [Glosario del NIST].

1.2.1 Auditorías de seguridad y prueba de seguridad

Las auditorías de seguridad tienen las siguientes características:

- Pueden ser realizadas por auditores internos o externos.
- Se centran en aspectos de los procesos y controles de seguridad de una organización, tales como:
    - Componentes físicos del sistema de información y el entorno en el que se almacena la información.
    - Aplicaciones y software, incluidos los parches y configuraciones de seguridad.
    - Controles de derechos y privilegios de los usuarios.

- Vulnerabilidades de la red, incluidas evaluaciones de la conexión entre diferentes puntos dentro y fuera de la red de la organización.
- Cómo los empleados recopilan, comparten y almacenan información.
- Mecanismos de detección de intrusiones.
- Planes de respuesta en caso de una brecha.

- Son un tipo de prueba estática (ver sección 2.1.2) que implica el examen manual de los productos de trabajo o revisiones automatizadas con herramientas de auditoría de seguridad.
- Investigan aspectos de las políticas de seguridad, procedimientos y controles de una organización que son difíciles de probar dinámicamente.
- Verifican la efectividad de los controles de seguridad instalados e identifican dónde se han alcanzado o no los criterios establecidos por la organización en un momento determinado.
- No garantizan el hallazgo de todas las vulnerabilidades, pero aseguran que se identifican las áreas problemáticas e indican dónde se necesitan medidas correctivas.

Las auditorías de seguridad deben formar parte de la rutina habitual. Pueden realizarse como:

- Evaluaciones puntuales para circunstancias especiales, como cuando una organización introduce una nueva plataforma de software o una nueva integración. Debe realizarse una prueba y auditoría de seguridad para descubrir nuevos riesgos y/o defectos.
- Auditorías programadas regularmente para verificar que se siguen los procesos y procedimientos de seguridad y que son adecuados para el clima y las necesidades actuales del negocio.

En algunos enfoques de auditoría de seguridad, las pruebas de seguridad se realizan como parte del proceso de auditoría para determinar si los controles de seguridad están realmente implementados y funcionan de manera efectiva. Sin embargo, el alcance de una auditoría de seguridad es mucho mayor que el de la prueba de seguridad.

La prueba de seguridad y la auditoría operan conjuntamente. La auditoría identifica deficiencias y áreas de importancia para probar. La prueba de seguridad es el medio por el cual se prueba o refuta que los controles de seguridad están realmente implementados y funcionando de forma efectiva.

1.3 El concepto de confianza cero

1.3.1 ¿Qué es confianza cero?

La confianza cero es un modelo de seguridad creado a partir de una colección de conceptos e ideas diseñados para minimizar la incertidumbre en la aplicación de un acceso preciso y de privilegio mínimo por solicitud. Se basa en el principio de controles de acceso estrictos y de no confiar en nadie por defecto, incluso si todos ya están dentro del perímetro de la red.

El modelo de confianza cero encarna el principio de “no confiar en nada, verificarlo todo, la parte visible de una red se considera comprometida”.

El aumento del uso de servicios en línea ha dado lugar a un aumento correspondiente de las vulnerabilidades y los ataques. La información se distribuye a menudo entre los proveedores de la nube, lo que ha dado lugar a que los conceptos de seguridad basados en el perímetro sean menos efectivos a la hora de proporcionar seguridad a las organizaciones, los empleadores, los usuarios y los clientes. La seguridad tradicional basada en el perímetro confía en cualquier persona y cualquier cosa dentro de la red. El problema de este enfoque es que, una vez que un atacante obtiene acceso a la red, tiene acceso a todos los activos que hay en su interior.

Al utilizar el modelo de confianza cero, los sistemas y servicios de información operan bajo el supuesto de que sus redes ya están comprometidas y de que la red no tiene ningún espacio de confianza. La

perspectiva de la confianza cero hace que la práctica que traslada las defensas de seguridad de los perímetros estáticos basados en la red se concentre en los usuarios, los activos y los recursos [Glosario del NIST].

Beneficios de la confianza cero [Cloudflare]:

- Reduce la superficie de ataque de una organización
- Minimiza el daño de un ataque al restringir la brecha a un área pequeña y reduce el costo de recuperación
- Reduce el impacto del robo de credenciales de usuario y los ataques de suplantación de identidad al requerir múltiples factores de autenticación

1.3.2 Confianza cero en la prueba de seguridad

Las redes actuales no tienen perímetro, y están migrando de despliegues planos a entornos dinámicos, distribuidos e híbridos. Las organizaciones se adaptan a la creciente complejidad de sus entornos, que abarcan lugares de trabajo híbridos, trabajadores remotos, interacciones con otras organizaciones y proveedores, y necesitan proteger a las personas, los dispositivos, las aplicaciones, las redes y los datos dondequiera que se encuentren. El modelo de confianza cero, con la premisa de "no confiar en nada, verificarlo todo", impulsa la necesidad de un cambio de paradigma completo en la seguridad de TI, en el que las personas, los dispositivos, las aplicaciones, las redes y los datos deben someterse a estrictas validaciones antes de obtener acceso a una aplicación o recurso solicitado.

Los principios de confianza cero [Micro22] [Cloudflare] incluyen:

- Monitorización y verificación continuas: siempre se verifica el acceso para todos los recursos. Los inicios de sesión y las conexiones se agotan periódicamente, lo que obliga a los usuarios y a los dispositivos a una continua reverificación.
- Principio de "acceso con menos privilegios": conceder a los usuarios sólo el acceso que necesiten.
- Autenticación multifactor: requiere más de una pieza de evidencia para autenticar a un usuario. No basta con introducir una contraseña para permitir el acceso.
- Monitorización de la seguridad de los puntos finales de los dispositivos que acceden a los datos, como ordenadores portátiles, estaciones de trabajo, tabletas, dispositivos móviles: cada punto final remoto puede ser el punto de entrada de un ataque.
- Microsegmentación: dividir los perímetros de seguridad en pequeñas zonas para lograr un acceso independiente para partes separadas de la red.
- Ningún movimiento lateral dentro de las redes sin una validación continua: el acceso de confianza cero está segmentado y debe restablecerse periódicamente. Un atacante no puede desplazarse a otros microsegmentos dentro de la red. Una vez detectada la presencia del atacante, el segmento o la cuenta de usuario comprometidos pueden ponerse en cuarentena y se les puede interrumpir el acceso.
- Protección de los datos a través de los archivos y el contenido: cifrado y restricciones de acceso basadas en políticas de la organización.

El modelo de confianza cero afecta a la forma en que se debería gestionar la prueba de seguridad. Esto significa centrar los casos de prueba en los mecanismos de seguridad de confianza cero. La prueba de seguridad contribuye a identificar las siguientes posibles debilidades:

- Red: Microsegmentos de confianza cero, reduciendo los daños y destacando las violaciones.
    - Probar el tráfico entre segmentos con automatización y con microsegmentos definidos por el usuario.
- Datos: La confianza cero exige que los datos se transmitan de forma cifrada y segura.

- Probar los puntos finales que exponen o almacenan datos utilizando métodos no encriptados.

- Identidad: las personas, los dispositivos y los procesos sólo pueden hacer lo que les está permitido.
  - Probar si las personas, los dispositivos y/o los procesos tienen más permisos de acceso de los necesarios que puedan comprometer los activos de la red.
  - Comprobar si los usuarios no autorizados pueden acceder a segmentos de los recursos de red.
- Los dispositivos deberían ejecutar sólo software seguro y ser monitorizados y gestionados de forma centralizada.
  - Probar si el dispositivo está protegido con software de seguridad y realizar la prueba de penetración.
- Limitación del radio de acción:
  - Las pruebas se realizan de forma regular para mostrar las debilidades con el fin de mitigar el impacto del riesgo para limitar el radio de acción en caso de que se produzca una brecha externa o interna.

1.4 Software de código abierto (SCA)

El SCA se desarrolla y mantiene mediante una colaboración abierta y está disponible, normalmente sin coste alguno, para que cualquiera pueda utilizarlo, modificarlo y redistribuirlo como desee.

1.4.1 El concepto de SCA y su impacto en la prueba de seguridad

El SCA se construye sobre código abierto para que cualquiera pueda utilizarlo, modificarlo y compartirlo libremente. El SCA se distribuye a menudo bajo licencias que se ajustan a la definición de código abierto, según la Iniciativa de Código Abierto (SCA) [en inglés, open-source software (OSS)].

La principal ventaja de utilizar SCA es la transparencia del código y la suposición de que muchos desarrolladores voluntarios han comprobado el código para detectar y resolver defectos. Se supone que esta apertura hará que haya más personas involucradas en la rápida identificación de vulnerabilidades y la solución de defectos.

Sin embargo, el hecho de que estas aplicaciones, librerías y otros objetos reutilizables estén disponibles en todo el mundo supone un reto, ya que cualquiera puede actualizar el código e introducir potencialmente vulnerabilidades y vectores de ataque [Shacklett]. Por lo general, el SCA tiene más vectores de ataque que el software cerrado (propietario), porque cualquiera puede añadir puertas traseras intencionadas, propagar vulnerabilidades a través de la reutilización y explotar vulnerabilidades y errores humanos divulgados públicamente. Una vez publicada una vulnerabilidad, los usuarios de esa versión del software corren el riesgo de sufrir un ataque.

El uso de SCA significa que cada elemento explotable publicado para un componente específico podría afectar potencialmente a miles de sistemas. Cuando el código fuente está disponible en versiones ejecutables, la observación, la ingeniería inversa, las revisiones de código, el desmontaje y las pruebas exploratorias pueden ser capaces de encontrar vulnerabilidades [Stallings18].

El ingeniero de prueba de seguridad (IPS) realiza las siguientes tareas:

- Identificación de vulnerabilidades del código abierto.
- Realizar revisiones del código como parte del desplazamiento a la izquierda, que tiene como objetivo la detección de defectos en fases tempranas del proceso de desarrollo.

Cuando se trata de probar SCA para la seguridad de las aplicaciones, el IPS piensa como un atacante. Los casos de prueba captan cómo se comporta una aplicación en diferentes escenarios de uso y uso inadecuado y permiten a los desarrolladores poner en marcha mitigaciones del riesgo adecuadas.

Los desarrolladores y los IPS realizan revisiones de código, tanto del lado del productor como del consumidor, para identificar código no seguro.

El Open Web Application Security Project (OWASP) ha puesto a disposición de los proyectos de código abierto herramientas automatizadas de detección de vulnerabilidades que son gratuitas [OWASP Top 10]. El NIST ofrece orientación para la seguridad del software de código abierto (SCA) [en inglés, open-source software (OSS)].

2 Técnicas de prueba de seguridad - (K3)

Duración: 150 minutos

Palabras clave[2]

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB**: identificarán palabras clave del proceso de prueba
- **ESPDOM**: identificarán palabras clave específicas de dominio: **SEGURIDAD**

Tipo Palabra Clave	Español	Inglés
ISTQB	prueba destructiva	destructive testing
ISTQB	prueba aleatoria	fuzz testing
ISTQB	escaneo de software malicioso	malware scanning
ISTQB	escaneo de vulnerabilidad	vulnerability scanning
ESPDOM	autenticación	authentication
ESPDOM	autorización	authorization
ESPDOM	encriptado	encryption
ESPDOM	cortafuegos	firewall

[2] Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.

Objetivos de aprendizaje para "Capítulo 2"

2.1 Uso de tipos de prueba de seguridad en función de un contexto de prueba

STE - 2.1.1 (K2) Aportar ejemplos de tipos de prueba de seguridad en función de un contexto de seguridad de caja negra, caja blanca y caja gris.

STE - 2.1.2 (K2) Aportar ejemplos de tipos de prueba de seguridad en función de una prueba de seguridad estática o una prueba de seguridad dinámica.

2.2 Uso de tipos de prueba de seguridad en función de un proyecto y un contexto técnico

STE - 2.2.1 (K3) Aplicar casos de prueba de seguridad, basados en un enfoque de prueba de seguridad dado, junto con riesgos de seguridad funcionales y estructurales identificados.

STE - 2.2.2 (K2) Describir cómo probar la reconciliación y recertificación de identidades y permisos.

STE - 2.2.3 (K2) Describir cómo probar el control de gestión de identidades y accesos.

STE - 2.2.4 (K2) Describir cómo probar el control de protección de datos.

STE - 2.2.5 (K2) Describir cómo probar tecnologías de protección.

2.1 Aplicación de tipos de prueba de seguridad en función de un contexto de prueba

2.1.1 Prueba de caja negra, prueba de caja blanca y prueba de caja gris

Los tipos de prueba para la base de prueba se clasifican como prueba de caja negra, prueba de caja gris y prueba de caja blanca [ISTQB FL]. El Glosario del ISTQB [Glosario del ISTQB] define las pruebas de caja negra y de caja blanca.

La prueba de seguridad de caja gris se define en [NIST] como “una metodología de prueba que asume cierto conocimiento de la estructura interna y los detalles de implementación del objeto de evaluación”. El ingeniero de prueba de seguridad (IPS) tiene acceso a cierta información sobre el sistema sujeto a prueba (SSP) (por ejemplo, información parcial de un mapa de direccionamiento de red, información parcial de la documentación de la arquitectura, acceso de usuario y acceso a una máquina interna).

El ingeniero de prueba de seguridad (IPS) [en inglés, security test engineer (STE)] tiene acceso a toda la información necesaria sobre el sistema sujeto a prueba (SSP) [en inglés, system under test (SUT)] durante las pruebas de seguridad:

- La arquitectura del SSP.
- El código fuente.
- Los flujos de datos en el SSP.
- El diseño de la red y la estructura de zonas.
- Los requisitos de contraseña.
- Las reglas del cortafuegos.
- La autenticación.
- El almacenamiento de registros y la información de gestión.

La elección de las técnicas de prueba se basa en los objetivos del enfoque de la prueba de seguridad, así como en la disponibilidad del código. El ingeniero de prueba de seguridad (IPS) [en inglés, security test engineer (STE)] debe decidir el nivel de profundidad de las pruebas y si centrarse en las amenazas externas o internas.

2.1.2 Prueba de seguridad estática y dinámica

Tanto la prueba estática como la prueba dinámica se utilizan en la prueba de seguridad para proteger el sistema sujeto a prueba (SSP) [en inglés, system under test (SUT)] a lo largo de todo su ciclo de vida.

- **Prueba de Seguridad Estática**

  Desde la perspectiva del ingeniero de pruebas de seguridad (IPS) [en inglés security test engineer (STE)], entre los productos de trabajo que pueden revisarse durante la prueba de seguridad estática se encuentran:

  - Documentación de análisis de riesgos de seguridad
  - Requisitos de seguridad

  Al buscar vacíos en los requisitos, deben tenerse en cuenta los siguientes mecanismos de seguridad:

  - Gestión de usuarios

- Autenticación
- Autorización
- Confidencialidad
- Integridad
- Responsabilidad
- Gestión de sesiones
- Seguridad en el transporte
- Segregación de sistemas por capa
- Conformidad legislativa y normativa, incluidas las normas de privacidad, gubernamentales y del sector
- Documentación técnica de seguridad arquitectónica
- Código fuente
- Configuración e infraestructura/configuración operativa

- **Prueba de Seguridad Dinámica**

  En comparación con la prueba de seguridad estática, el objetivo de la prueba de seguridad dinámica es comprobar que el sistema sujeto a prueba (SSP) implementa y utiliza correctamente las funciones de seguridad según lo requerido o especificado y que estas funciones de seguridad no pueden ser eludidas durante el uso del sistema o la aplicación.

  La prueba de seguridad dinámica puede realizarse mediante:

  - Uso de la prueba de caja negra para evaluar las funciones de seguridad comprobando que los resultados de la prueba son los esperados cuando se envían entradas particulares.
  - Prueba de penetración: Intenta hallar vulnerabilidades que podrían ser elementos explotables (fragmentos de código).

  Se recomienda ejecutar la prueba de seguridad dinámica de la siguiente manera en los siguientes entornos, si fuera posible:

  - El entorno de desarrollo
    - Esta es la primera oportunidad para que el ingeniero de prueba de seguridad (IPS) [en inglés security test engineer (STE)] ejecute pruebas de seguridad. En este entorno, el ingeniero de prueba de seguridad (IPS) [en inglés security test engineer (STE)] verifica que la implementación de la seguridad se ajusta a los requisitos de seguridad (por ejemplo, el desarrollo correcto de una contraseña controlando el tamaño mínimo, el tamaño máximo y los tipos de caracteres obligatorios).
  - El entorno de prueba y el entorno de preproducción
    - A menudo conocido como entorno de prueba de aceptación o de ensayo, este entorno debe ser lo más similar posible al entorno de producción y debe contener todas las medidas de seguridad previstas que se aplicarán en el entorno de producción.
  - El entorno de producción
    - Este es el entorno más crítico. El ingeniero de prueba de seguridad (IPS) [en inglés, security test engineer (STE)] debería tener cuidado de no desactivar el sistema sujeto a prueba (SSP) [en inglés, system under test (SUT)] y debería realizar auditorías de seguridad para mantener el sistema seguro. El objetivo de las pruebas de seguridad en el entorno de producción es comprobar que se han solucionado las vulnerabilidades recién descubiertas y conocidas.

  El uso de una herramienta dinámica de prueba de seguridad de las aplicaciones (PDSA) permite detectar condiciones que podrían dar lugar a vulnerabilidades (véase el capítulo 9). La prueba dinámica de seguridad de las aplicaciones (PDSA) [en inglés, dynamic application security testing

(DAST)] puede incluirse en una canalización de integración continua/entrega continua (CI/CD) en las siguientes fases:

- Durante la prueba posterior a una construcción, funciona como un escáner de seguridad dinámico para detectar defectos de seguridad.
- Durante la producción, funciona como un escáner de vulnerabilidades.

2.2 Aplicación de la prueba de seguridad

El diseño de la prueba de seguridad puede basarse en las siguientes fuentes:

- Un análisis de riesgo completo.
- Modelos de amenazas disponibles.
- Una categorización ad-hoc de los riesgos de seguridad (véase [ISTQB_ATTA_SYL]).
- Un enfoque de la prueba de seguridad.
- Requisitos de seguridad de las funciones y mecanismos de seguridad.
- Sistemas y productos en el ciclo de vida de desarrollo de software (CVDS).
- La experiencia y las habilidades de prueba del ingeniero de pruebas de seguridad (STE).
- Incidentes anteriores en los que la seguridad fue (casi) violada.
- Una guía de pruebas de referencia, como la [Guía de Prueba OWASP].

Los siguientes son atributos de una prueba de seguridad que deben tenerse en cuenta durante el diseño de la prueba de seguridad:

- Normativas o leyes requeridas (obligatorias).
- Riesgos de seguridad identificados y modelos de amenaza priorizados por el enfoque de prueba de seguridad.
- Trazado según los requisitos de seguridad definidos
- Definido según los probadores previstos (por ejemplo, desarrolladores, probadores funcionales y ETS)
- Definido según los perfiles de los defectos de seguridad y las vulnerabilidades conocidas Diseñado para ser automatizado, si procede Prueba destructiva o prueba no destructiva
- Intrusivo o no intrusivo (por ejemplo. el objetivo es romper un sistema desde dentro o hacer caer un sistema mediante una denegación de servicio distribuida)
- El flujo de trabajo básico del diseño de pruebas de seguridad es:
- Análisis del enfoque de las pruebas de seguridad (a nivel de proyecto)
- Análisis de los riesgos de seguridad, modelos de amenazas y requisitos (a nivel de proyecto)
- Aplicación de técnicas de pruebas de seguridad

En la mayoría de los casos, un diseño eficaz de pruebas de seguridad se basa en una mezcla de las fuentes anteriores. Dependiendo del tipo de proyecto, es importante asegurar que se realice una prueba de seguridad en cada fase del ciclo de vida de desarrollo de software (CVDS) del sistema sujeto a prueba (SSP) [en inglés, system under test (SUT)].

2.2.1 Afrontar los riesgos de seguridad en el diseño de pruebas

Un principio clave es que el ingeniero de prueba de seguridad (IPS) debería ser capaz de crear e implementar casos de prueba de seguridad basados en cualquier riesgo de seguridad, requisito de seguridad, amenaza y experiencia identificados.

La prueba de seguridad puede basarse en riesgos de seguridad externos en el entorno de producción (amenazas sobre un sistema o producto), en un enfoque de prueba de seguridad y en otras fuentes como los modelos de amenazas. Los riesgos de seguridad también pueden considerarse de naturaleza funcional y estructural (es decir, riesgos debidos a la falta de seguridad por diseño).

Durante el diseño del caso de prueba de seguridad, el ingeniero de prueba de seguridad debe identificar si una prueba es destructiva o no destructiva. Si una prueba se identifica como destructiva debe asegurar que no tiene ningún impacto negativo en otras actividades de prueba, entornos o el negocio.

A continuación, se describen los riesgos y vulnerabilidades de seguridad más comunes en los niveles funcional y estructural, junto con las técnicas de prueba de seguridad apropiadas.

- **Controles de seguridad funcional, riesgos o vulnerabilidades**

  Las pruebas se diseñan para verificar y validar que los controles están instalados, que funcionan correctamente y que son eficaces para detectar y prevenir acciones no autorizadas.

  Las técnicas de prueba de seguridad se basan en los requisitos de los controles de seguridad funcional y los controles de accesibilidad funcional.

- **Controles de acceso estructural, riesgos o vulnerabilidades**

  Las pruebas de estos controles se basan en cómo se han establecido los derechos de usuario para el acceso a los datos, el acceso funcional y los niveles de privacidad. Los controles de acceso estructurales suelen ser aplicados por un administrador del sistema, un administrador de seguridad o un administrador de la base de datos. En algunos casos, los derechos de acceso son una opción de configuración en una aplicación. En otros casos, los derechos de acceso se aplican a nivel de la infraestructura de un sistema.

- **Riesgos o vulnerabilidades de acceso al sistema operativo**

  Una vez obtenido el acceso al sistema operativo, un atacante puede controlar los procesos, los datos y el acceso a la red, lo que puede posibilitar la inserción de software malicioso.

- **Riesgos o vulnerabilidades de la plataforma**

  Las técnicas de prueba de seguridad se basan en la experiencia del ingeniero de prueba de seguridad ((IPS) [en inglés security test engineer (STE)] (por ejemplo, en el tratamiento de vulnerabilidades) y en la prueba de procedimientos de seguridad (por ejemplo, la prueba del mantenimiento de las condiciones de seguridad).

- **Amenazas externas e internas**

  Algunas amenazas, como la explotación de vulnerabilidades de la aplicación o del lenguaje de programación, pueden detectarse, probarse y limitar su impacto. Las amenazas internas son ejecutadas por empleados internos. Las amenazas externas son ejecutadas por personas externas (por ejemplo, atacantes).

  Las técnicas de prueba de seguridad se basan en pruebas exploratorias (por ejemplo, para encontrar objetivos potenciales y puntos de inyección/vectores de ataque útiles) y en la experiencia del ingeniero de prueba de seguridad (IPS) [en inglés security test engineer (STE)].

2.2.2 Prueba de reconciliación y pruebas de recertificación

2.2.2.1 Comprender los asuntos de interés sobre la gestión de la identidad y el acceso

Los servicios de negocio que prestan las organizaciones son cada vez más complejos. Se despliegan como un sistema de sistemas y se alojan en entornos híbridos compuestos por elementos propios, suministrados por socios, suministrados por clientes y en la nube. En este entorno distribuido, la gestión de la seguridad de las cuentas de usuario y de los derechos de usuario es crítica. Por ejemplo:

- El usuario debe tener los privilegios adecuados, y ninguno más.
- Los derechos deben ser revocados después de que un empleado haya abandonado la organización.
- La gestión de derechos debe cumplir la normativa, por ejemplo, el Reglamento General de Protección de Datos (RGPD).

La gestión de la identidad y el acceso (GIA) es una disciplina para gestionar y mantener las cuentas y los derechos de los usuarios mediante la definición de quién (identidad) está dispuesto a acceder a qué (función) para un recurso específico. La gestión de la identidad y el acceso (GIA) enumera dos subprocesos, que están directamente relacionados con la prueba de seguridad:

- Reconciliación:
    - Comparación y actualización del acceso de los usuarios, derechos y cuentas privilegiadas a través de solicitudes de cambio y cadenas de aprobación a una base de datos autorizada y fiable de gestión de identidades.
- Recertificación:
    - Revisiones periódicas de las cuentas y los derechos y privilegios relacionados para verificar si siguen siendo necesarios.

La reconciliación es necesaria para asegurar que todos los accesos a las aplicaciones estén sincronizados con la misma “fuente de confianza”. Los niveles dentro del proceso de reconciliación son:

- Completa (full):
    - Comparación de todas las cuentas y atributos de acceso de los usuarios con el sistema de gestión de identidades y accesos, con el fin de identificar cualquier diferencia.
- Incremental (incremental):
    - Sólo se comparan los cambios en las cuentas y derechos creados, actualizados o eliminados.
- Automática (automatic):
    - Cuando se utilizan aplicaciones que pueden establecer un calendario de comparaciones automáticas de cualquier cambio realizado relevante para la seguridad.

La recertificación consiste en auditar una cuenta de usuario y los privilegios de acceso para determinar si siguen estando justificados, son consistentes con las políticas internas de la organización y cumplen la normativa. Esto implica la realización de una auditoría continua para asegurar que los usuarios sólo tienen acceso a lo que necesitan y para lo que tienen permiso. La evaluación puede ser:

- Manual
    - Extraer y recopilar información contable.
    - Presentar la información.
    - Revisión por parte de los revisores.
- Automatizada
    - Se envían mensajes a los gestores para que emitan solicitudes de recertificación.
    - Esto tiene la ventaja de poder planificar auditorías de forma regular.

Dentro de una gran organización, con sistemas alojados en una amplia gama de entornos, la gestión de la identidad y el acceso (GIA) se vuelve compleja debido a la necesidad de gestionar varias aplicaciones, cada una con accesos que deben concederse y revocarse en función de los movimientos de un usuario (por ejemplo, llegada, salida y traslado). En algunas organizaciones, las cuentas y los privilegios de los usuarios se gestionan al margen de un proceso formal de GIA para ahorrar tiempo. Se trata de una dificultad crítica desde el punto de vista de la seguridad porque las cuentas huérfanas y no utilizadas pueden dar lugar a que un atacante las explote.

2.2.2.2 Cómo realizar pruebas de reconciliación y pruebas de recertificación para los mecanismos de identidades y permisos

Deben realizarse pruebas de reconciliación y pruebas de recertificación para evitar incoherencias en las cuentas de usuario de todas las aplicaciones a las que tiene acceso el usuario. Esto incluye aspectos como las credenciales de inicio de sesión y los privilegios.

- **Serán de aplicación las siguientes condiciones de prueba:**
  - Nueva gestión de cuentas.
  - Modificación de las credenciales y privilegios de la cuenta.
  - Reubicación de una cuenta, incluida la eliminación del acceso a aplicaciones y la inclusión del acceso a nuevas aplicaciones.
  - Revisión de todas las cuentas.
- **Los objetivos de prueba pueden ser:**
  - Intentar intercambiar, cambiar o acceder a otro rol.
  - Revisar la granularidad de los roles y las necesidades detrás de los permisos otorgados.
  - Verificar que los requisitos de identidad para el registro de usuarios se adecuen a los requisitos de negocio y de seguridad.
- **Las técnicas de prueba de seguridad para la reconciliación y la recertificación incluyen:**
  - Revisar la documentación de los procesos de reconciliación y recertificación.
  - Comprobar si los registros han sido examinados por una persona antes de la provisión, o si se conceden automáticamente cuando se cumplen determinados criterios
  - Verificar, examinar y autorizar las solicitudes de desaprovisionamientro.
  - Comprobar que las modificaciones de las cuentas son efectivas
  - Realizar pruebas aleatorias de los posibles roles para asegurarse de que el sistema rechaza los roles falsificados.
  - Revisar los permisos de rol después de recopilar todas las modificaciones aplicadas

Se debería tener en cuenta que OWASP [Open Web Application Security Project (OWASP)] proporciona una lista de objetivos de prueba de seguridad y técnicas de prueba relacionadas con la prueba de la gestión de la identidad y el acceso (GIA) [en inglés, identity and access management (IAM)].

2.2.3 Prueba de identificación, autenticación y autorización

2.2.3.1 Comprender la relación entre autenticación y autorización

Los activos sensibles de una organización deben estar protegidos y sólo deben ser accesibles a una persona autorizada que haya sido previamente autenticada. La identificación es el primer paso para obtener acceso a un recurso. Es el proceso de aserción de una identidad.

La autenticación se basa en la verificación de un identificador de usuario y un testigo ("token") para responder a las siguientes preguntas:

- ¿Quién es el usuario? (identificador de usuario).
- ¿Realmente es el usuario que alega ser? (testigo ("token"), como una contraseña o un certificado).

Se pueden utilizar diferentes implementaciones de mecanismos de autenticación en función del nivel de protección que se quiera ofrecer contra ataques para apropiarse de una autenticación o robar el testigo ("token").

La autorización se utiliza para los siguientes fines:

- Verificar si el usuario autenticado tiene derechos para realizar una acción.
- Determinar qué nivel de acceso debería permitirse a los recursos del sistema.

Existe una estrecha relación entre autenticación y autorización basada en el principio de que un usuario no autenticado no debe tener privilegios o tenerlos restringidos en el sistema.

La autenticación, autorización y responsabilidad del sujeto dan lugar a la abreviatura AAR (en inglés, AAA), que es un marco de trabajo que ayuda a realizar el control y seguimiento del acceso dentro de una red informática. La responsabilidad es la "R" del marco AAR. del marco de trabajo AAR, que tiene en cuenta el registro, seguimiento y actividades de un usuario.

MARCO	Inglés		Español	MARCO
A	**Authentication**		**Autenticación**	**A**
A	**Authorization**		**Autorización**	**A**
A	**Accounting**		**Responsabilidad**	**R**

2.2.3.2 Cómo probar los mecanismos de autenticación y autorización

- **Las técnicas de prueba de seguridad (pruebas de penetración) para los mecanismos de autenticación podrían incluir:**
  - Prueba de credenciales por defecto.
  - Prueba de la debilidad de la política de contraseñas.
  - Búsqueda de información filtrada utilizando inteligencia de fuentes abiertas [en inglés, open source intelligence (OSINT)].
  - Pruebas de fuerza bruta utilizando diccionarios y tablas arcoíris (es decir, tablas precalculadas de valores resumen inversos de contraseñas) para realizar ataques que intenten descubrir las contraseñas de los usuarios. Los primeros pasos pueden ser, por ejemplo, probar con "123456", "111111", la fecha de nacimiento o el nombre de una mascota.

- **Las técnicas de prueba de seguridad (pruebas de penetración) para los mecanismos de autorización podrían incluir:**
    - Explotar una falta de filtrado de entrada, como inyectar solicitudes SQL para autenticarse sin ningún nombre de usuario/contraseña conocidos o provocar un desbordamiento de memoria intermedia de la entrada para obtener acceso de administración a una sesión de intérprete de comandos.
    - Introducir un Identificador Uniforme de Recursos no autorizado (por ejemplo, ../../ en una cuenta de Protocolo de Transferencia de Archivos) o un Localizador Uniforme de Recursos (URL) (por ejemplo, dirección del sitio/admin) para intentar acceder a datos confidenciales.
    - Prueba de violaciones de escalado de privilegios horizontales y verticales.

Se debería tener en cuenta que la Guía de Prueba OWASP proporciona una lista de técnicas de prueba de seguridad para probar la autenticación y la autorización.

2.2.4 Cifrado

2.2.4.1 Comprender el cifrado

Se puede utilizar un mecanismo de cifrado para evitar la divulgación de datos sensibles, incluso si se puede acceder a ellos cuando se almacenan en algún lugar o se intercambian entre un cliente y un servidor. El cifrado es un proceso de codificación de datos, (por ejemplo, texto plano), en datos cifrados (por ejemplo, texto cifrado), utilizando un algoritmo criptográfico y claves secretas. El secreto es compartido y sólo lo conocen los usuarios autorizados. El objetivo es utilizar un cifrado lo suficientemente fuerte como para impedir que un atacante, que haya conseguido robar los datos cifrados, recupere el texto sin cifrar. El uso de algoritmos criptográficos ayuda a asegurar la confidencialidad y la integridad de los activos sensibles y a evitar que sean manipulados.

- **Los tipos de protocolos criptográficos principales y más utilizados son:**
    - Cifrado simétrico: basado en el uso de una clave secreta compartida.
    - Cifrado asimétrico: basado en el uso de una clave privada, pública y certificados, gestionados a través de una infraestructura de clave pública.

2.2.4.2 Cómo probar los mecanismos de cifrado más comunes

Se sabe que algunos mecanismos criptográficos son débiles, especialmente debido a la corta longitud de las claves secretas. Otros mecanismos pueden ser vulnerables porque no se implementan siguiendo las buenas prácticas o porque incorporan defectos de codificación (por ejemplo, desbordamiento de memoria intermedia).

- Las pruebas de los mecanismos de cifrado deberían incluir:
    - Pruebas de conformidad (por ejemplo, pruebas basadas en requisitos de seguridad) de implementaciones del mecanismo de cifrado.
    - Pruebas de vulnerabilidad “por diseño”.
    - Pruebas de vulnerabilidad “por construcción”.
    - Pruebas de vulnerabilidades “por configuración”.

Se debería tener en cuenta que la Guía de Prueba OWASP proporciona una lista de pruebas de seguridad para comprobar implementaciones criptográficas débiles.

2.2.5 Prueba de tecnologías de protección

Los ingenieros de prueba de seguridad (IPS) deben comprender los matices de las diferentes líneas de defensa para poder diseñar pruebas adecuadas que verifiquen y validen su efectividad.

2.2.5.1 Cómo probar la fortificación de sistema

La efectividad de la fortificación de sistemas se puede probar de varias maneras. La fortificación de sistemas restringe el acceso al sistema a los roles correctos, abre solo los servicios necesarios y monitoriza las actualizaciones de las aplicaciones. Por lo tanto, para probar la efectividad de la fortificación de sistema, las pruebas deben diseñarse para detectar si las medidas de fortificación de sistema están funcionando, aplicadas en los lugares correctos y de la manera correcta. También es relevante probar las protecciones de fortificación de sistema que son demasiado restrictivas y podrían ser excesivas en comparación con los riesgos de seguridad.

Algunas pruebas de fortificación del sistema pueden basarse en una revisión o una auditoría, mientras que otras pueden basarse en la capacidad de ciertos grupos de usuarios para realizar ciertas acciones o acceder a ciertos datos.

2.2.5.2 Cómo probar cortafuegos

Debido a la cantidad de protocolos, sus diferentes opciones y la complejidad de las redes a proteger, es difícil configurar un cortafuegos de manera eficiente y consistente. Las pruebas de efectividad del cortafuegos deberían incluir:

- Realizar una auditoría para verificar la configuración del cortafuegos.
- Escaneo de puertos para verificar si la política de seguridad está bien implementada.
- Uso de paquetes de red formateados de forma incorrecta y prueba aleatoria de red para encontrar elementos explotables (fragmentos de código) que generen comportamientos inesperados.
- Ataques de fragmentación para eludir las prestaciones de filtrado con el objetivo de llevar a cabo un ataque detrás del cortafuegos.
- Atacar el cortafuegos de aplicaciones web codificando y comprimiendo datos u ofuscándolos para ocultar la información maliciosa que representa el ataque. Los criterios de evaluación del cortafuegos de aplicaciones web [WAFEC] pueden utilizarse para probar la efectividad de un cortafuegos de aplicaciones web.

2.2.5.3 Cómo realizar la detección de intrusiones

La detección basada en escenarios se basa en un escenario conocido o "patrón". Es fácil de eludir porque sólo se detectan los ataques conocidos. Las pruebas podrían incluir las siguientes técnicas de evasión:

- Codificación de caracteres o modificación de datos (por ejemplo, añadir espacios en blanco e indicadores de fin de línea).
- Fragmentación del protocolo de Internet (IP), segmentación del protocolo de control de transmisión (TCP).
- Cifrado u ofuscación.
- Codificación del Localizador Uniforme de Recursos (URL).

La detección basada en el comportamiento se basa en un modelo de comportamiento del sistema y genera un gran número de resultados de falsos positivos y resultados de falsos negativos. Un resultado de falso negativo es cualquier alerta de seguridad que debería haber sido informada pero no lo fue. Los resultados de falso negativo pueden producirse cuando se desarrolla un nuevo ataque del que un sistema de detección de intrusión (SDI) no es consciente, o quizás una regla podría estar escrita de tal forma que detectara algunos ataques, pero pasara por alto aquellos que no están especificados en el modelo.

La exactitud de este método de detección debe mantenerse. Es posible que un atacante se desvíe del comportamiento normal de un IDS, lo que da lugar a una nueva especificación del comportamiento intrusivo. Las pruebas complementarias deben utilizar el tráfico malicioso para añadir nuevas especificaciones de intrusión que se tengan en cuenta como tráfico autorizado.

2.2.5.4 Cómo escanear software malicioso

Los desarrolladores de software malicioso utilizan diferentes técnicas para proteger su código contra la ingeniería inversa y la detección por parte de software contra software malicioso. Algunas de estas técnicas incluyen:

- Explotación de las funciones de la librería del sistema utilizadas por el software malicioso.
- Ofuscación de cadenas para impedir la comprensión del comportamiento del código malicioso.
- Permutación de código.
- Inserción de código no utilizado.
- Carga dinámica de funciones y librerías (por ejemplo, para limitar el análisis del código malicioso).
- Actualización automática de las aplicaciones.

Desde la perspectiva de las pruebas de idoneidad funcional, las herramientas antimalware basadas en firmas podrían utilizarse para probar la eficacia de las medidas contra el software malicioso sin desarrollar piezas de código malicioso reales. Deben probarse otros tipos de archivos maliciosos en relación con el tipo de aplicaciones.

La prueba del producto contra software malicioso basada en el comportamiento es difícil porque no existe una comprensión y definición claras de lo que es un comportamiento malicioso. Las ideas para las pruebas pueden beneficiarse de las técnicas utilizadas por los desarrolladores de software malicioso:

- Archivos de ejecución sin firmar que intentan utilizar llamadas al sistema para realizar cambios en el sistema.
- Intentar lanzar procesos inusuales con derechos concedidos.
- Intentar copiar archivos de ejecución en ubicaciones no autorizadas.
- Intentar llamar a IPA's del sistema inusuales.

Una consideración importante a la hora de implementar una nueva medida contra software malicioso (ya sea basada en un patrón o en el comportamiento) o de actualizar la existente es probar la implementación en una plataforma representativa antes de desplegarla en toda la organización.

2.2.5.5 Probar la ofuscación de datos

Se necesita un estricto control de la configuración entre los datos ofuscados y las claves utilizadas para la ofuscación para asegurar que se utilizan las versiones correctas de las claves. De lo contrario, los datos no podrán ser comprensibles para su uso.

Dado que en algunas pruebas podrían estar implicados datos privados, la ofuscación de datos puede utilizarse con fines de prueba para convertir en anónimos los datos de producción utilizados en un entorno de prueba del sistema. Los datos sensibles, como la información de usuarios utilizada por un sistema de

información sanitaria, no deben divulgarse a los probadores. Las pruebas podrían incluir ataques de fuerza bruta o de diccionario para intentar obtener datos en claro a partir de datos ofuscados.

Las pruebas para verificar la ofuscación del código podrían incluir:

- Ingeniería inversa del código.
- Ataques de fuerza bruta, porque algunos mecanismos de ofuscación son vulnerables.

3 El proceso de prueba de seguridad - (K3)

Duración: 120 minutos

Palabras clave[3]

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB**: identificarán palabras clave del proceso de prueba
- **ESPDOM**: identificarán palabras clave específicas de dominio: **SEGURIDAD**

Tipo Palabra Clave	Español	Inglés
ISTQB	riesgo	risk
ISTQB	entorno de prueba	test environment
ISTQB	plan de prueba	test plan
ISTQB	proceso de prueba	test process

[3] Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.

Objetivos de aprendizaje para "Capítulo 3"

3.1 El proceso de prueba de seguridad

STE - 3.1.1 **(K2)** Explicar las diferentes actividades, tareas y responsabilidades dentro de un proceso de prueba de seguridad

STE - 3.1.2 **(K2)** Comprender los elementos clave y las características de un entorno de prueba de seguridad.

3.2 Diseño de pruebas de seguridad para niveles de prueba

STE - 3.2.1 **(K2)** Aportar ejemplos de pruebas de seguridad en el nivel de prueba de componente basado en un código determinado

STE - 3.2.2 **(K2)** Aportar ejemplos de pruebas de seguridad en el nivel de integración de componentes basadas en una especificación de diseño determinada.

STE - 3.2.3 **(K3)** Implemente una prueba de seguridad de extremo a extremo que valide uno o más requisitos de seguridad relacionados con uno o más procesos de negocio.

3.1 El proceso de prueba de seguridad

La prueba de seguridad es un proceso dentro de un CVDS. El proceso de prueba de seguridad está dedicado al alcance de la seguridad y debe estar alineado con el proceso de desarrollo para que se realicen las actividades de prueba adecuadas cuando sea necesario.

Los riesgos de seguridad y las necesidades de cada organización son únicos debido a la naturaleza de la organización, los entornos técnicos, el ciclo de vida de desarrollo de software (CVDS) y los riesgos de negocio. Por lo tanto, el proceso de prueba de seguridad debe definirse e implementarse en el contexto de estos factores.

3.1.1 Proceso de prueba de seguridad de ISTQB

La tabla 3.1 muestra cómo tener en cuenta y tratar las actividades de prueba de seguridad dentro del Proceso de Prueba Básico de ISTQB.

Proceso de Prueba de Seguridad de ISTQB	Tareas de Prueba de Seguridad	Responsabilidades
Planificación de la prueba de seguridad: El objetivo es definir un alcance adecuado de la prueba de seguridad que se corresponda con los riesgos de seguridad.	• Tener en cuenta los requisitos relacionados con la seguridad. • Definir los objetivos de la prueba de seguridad. • Definir el alcance de la prueba de seguridad. • Identificar los recursos para la prueba de seguridad. • Definir las estimaciones y los calendarios de la prueba de seguridad. • Definir las métricas de la prueba de seguridad, los criterios de entrada y los criterios de salida.	El responsable de esta tarea es un analista de prueba de seguridad (APS). El ingeniero de prueba de seguridad (IPS) contribuye a la planificación estimando la carga de trabajo de las pruebas y los recursos de hardware y software necesarios.

Proceso de Prueba de Seguridad de ISTQB	Tareas de Prueba de Seguridad	Responsabilidades
Análisis de prueba de seguridad: El objetivo es comprender todas las condiciones de prueba de seguridad y determinar qué probar.	• Revisar la base de prueba para la prueba de seguridad, como las evaluaciones del riesgo de seguridad, los requisitos relacionados con la seguridad, la arquitectura basada en la seguridad y las políticas de seguridad. • Definir las condiciones de prueba de seguridad en función de: ○ Objetivos de seguridad. ○ Riesgos de seguridad ○ Estándares de seguridad y vulnerabilidades conocidas. ○ Defensas implementadas para asegurar el sistema y sus datos. ○ Alcance de la prueba de seguridad.	Un analista de prueba de seguridad (APS) define el alcance de la prueba de seguridad.
Diseño de prueba de seguridad:	• El objetivo es identificar casos de prueba de alto nivel, es decir, cómo probar • Diseñar casos de prueba de seguridad y conjuntos de pruebas • Dar prioridad a los casos de prueba y a las suites de prueba • Identificar los datos de prueba necesarios, ya sea para la evaluación de la vulnerabilidad o para las pruebas de penetración • Diseñar el entorno de prueba de la seguridad (es decir, la infraestructura y las herramientas) • Establecer la trazabilidad entre la base de prueba y los casos de prueba	El ingeniero de prueba de seguridad (IPS) diseña y prioriza los casos de prueba de seguridad Un STA revisa los productos del trabajo del ingeniero de pruebas de seguridad (IPS).

Proceso de Prueba de Seguridad de ISTQB	Tareas de Prueba de Seguridad	Responsabilidades
Implementación de prueba de seguridad:	• Organizar los casos de pruebas de seguridad en procedimientos de prueba o guiones de prueba. • Establecer un entorno de prueba para realizar la prueba de seguridad.	Un ingeniero de pruebas de seguridad (IPS) implementa los casos de prueba de seguridad.
Ejecución de prueba de seguridad:	• Realizar pruebas de seguridad de adecuación funcional. • Realizar pruebas de penetración • Determinar vulnerabilidades específicas • Informar de forma detallada los resultados provisionales de las pruebas de seguridad a la dirección.	El ingeniero de pruebas de seguridad (IPS) ejecuta las pruebas de seguridad, genera resultados detallados de las pruebas y comunica las vulnerabilidades identificadas lo antes posible.
Monitorización y control de la prueba de seguridad:	• Monitorizar el avance de las pruebas de seguridad y los resultados de las pruebas. • Tomar las medidas correctivas necesarias en respuesta a la información recopilada	El responsable de esta tarea es el analista de pruebas de seguridad (APS).
Compleción de la prueba de seguridad:	• Asegurar que se han realizado todas las pruebas de seguridad planificadas. • Analizar los resultados de las pruebas de seguridad para evaluar los riesgos residuales. • Analizar los resultados de las pruebas de seguridad para mejorar el desarrollo del software en términos de seguridad. • Informar de los resultados finales de las pruebas de seguridad a la dirección y a otras partes interesadas autorizadas. • Determinar si se han entregado los entregables de pruebas de seguridad (es decir, los informes de prueba). • Archivar los resultados de pruebas, datos de pruebas y otra información sensible en lugares seguros.	El analista de prueba de seguridad (APS) recopila toda la información producida durante la ejecución de la prueba de seguridad y elabora un informe de prueba de alto nivel para la dirección.

Cuando se han realizado pruebas exploratorias, el diseño de la prueba de seguridad, la implementación de la prueba de seguridad y la ejecución de la prueba de seguridad se basan en los resultados de pruebas anteriores utilizando técnicas estándar de exploración como husmeadores (o monitores de red), escáneres, ataques de fuerza bruta y bots. El diseño de prueba, la implementación de prueba y la ejecución de prueba se realizan de forma continua.

3.1.2 El entorno de prueba de seguridad

Aunque muchos tipos de prueba pueden utilizar un entorno de prueba situado en el mismo servidor(es) y red(es) con otros sistemas, la prueba de seguridad tiene riesgos específicos, incluso si está virtualizada o basada en contenedores. Por ejemplo, la realización de pruebas de seguridad destructivas, la contaminación del sistema sujeto a prueba (SSP) y la corrupción o divulgación de datos requieren un enfoque segregado para la construcción de un entorno de prueba para la prueba de seguridad. Además, en la mayoría de los dominios empresariales, las normativas exigen que se utilicen distintos entornos para el desarrollo, la prueba y la producción. Por ejemplo, el estándar de seguridad de datos del sector de las tarjetas de pago (PCI DSS), requisito 6.4.2, establece que es necesaria la separación de funciones entre los entornos de desarrollo, prueba y producción. Del mismo modo, el requisito 6.4.3 de la PCI DSS establece que los datos de producción (es decir, las redes de área personal) no se utilizarán para prueba o desarrollo [véase el capítulo 6, PCI DSS].

El entorno de pruebas de seguridad debe contener todas las funciones necesarias con las que realizar las pruebas. Entre ellas se incluyen herramientas de gestión de la prueba, herramientas de prueba de seguridad y herramientas de automatización de la prueba. Éstas se necesitan para permitir el descubrimiento del mayor número posible de vulnerabilidades en el tiempo asignado, y con el menor número posible de resultados falsos positivos y falsos negativos.

Por lo tanto, podría ser necesario identificar, especificar y establecer un entorno de prueba de seguridad independiente y eficaz para proteger otros entornos como el de desarrollo, prueba de componentes, prueba de integración de componentes, prueba de sistema, prueba de aceptación y producción. Esta efectividad debe cubrir tanto la tolerancia a defectos respecto a las pruebas de seguridad destructivas como proporcionar protección contra otros sistemas sujetos a prueba y para la productividad de las pruebas de seguridad.

El ingeniero de pruebas de seguridad (IPS) debe analizar y estimar la arquitectura necesaria, las IPA (en inglés, Application Programming Interface - API) y el comportamiento del sistema sujeto a prueba (SSP) para apreciar el impacto de las pruebas de seguridad y definir el entorno de pruebas más eficaz.

Las principales características de un entorno de prueba de seguridad incluyen:

1. Aislamiento al nivel adecuado (si es necesario): En función de los riesgos, el sistema sujeto a prueba (SSP) se aísla mediante comunicaciones filtradas, o bien el SSP y todos los demás sistemas dependientes se aíslan de otros entornos (por ejemplo, un sitio web comercial necesita un servicio de gestión de pagos independiente).
2. Entorno objetivo representativo: Para obtener el comportamiento correcto del SSP, el entorno total debe reflejar el entorno de producción en cuanto a versión y configuración exactas.
3. Productivo:

   Contiene todas las herramientas necesarias para planificar, preparar, ejecutar e informar sobre las pruebas de seguridad, ya sea de forma manual o (cuando sea posible) automatizada. La ejecución de pruebas de seguridad necesita herramientas de prueba específicas, como se describe en el capítulo 9.
4. Recuperable:

   Para repetir las pruebas según se necesite y para recuperarse de corrupción en caso de que se produjera.

3.2 Diseño de pruebas de seguridad para niveles de prueba

El modelado de amenazas es una actividad repetitiva en la que cada nivel de prueba de seguridad debe ajustarse en función de los resultados de prueba de los últimos resultados del modelado de amenazas.

Dependiendo del tipo de proyecto, es importante asegurar que hay una prueba de seguridad planificada en cada fase aplicable del CVDS.

3.2.1 Diseño de prueba de seguridad a nivel de prueba de componente

3.2.1.1 Base de prueba para el diseño de pruebas de seguridad para el nivel de prueba de componente

- Ejemplos de productos de trabajo que pueden utilizarse para diseñar pruebas de seguridad son:
  - Análisis del riesgo.
  - Requisitos de las funciones y mecanismos de seguridad.
  - Diseño detallado de funciones y mecanismos de seguridad (por ejemplo, IPA y algoritmos).
  - Modelos de datos.
  - Reglas de compilación o construcción.
  - Información del compilador.

3.2.1.2 Objetos de prueba para el diseño de pruebas de seguridad para el nivel de prueba de componente

- Objetos de prueba típicos para la prueba de componente de seguridad son:
  - Componentes.
  - Dependencias (por ejemplo, bibliotecas de terceros).
  - Código fuente.
  - Módulos de base de datos.

3.2.1.3 Defectos y fallos de seguridad típicos en el nivel de prueba de componente

- Ejemplos de defectos y fallos de seguridad típicos que se pueden encontrar en el nivel de prueba de componentes son:
  - Código y lógica incorrectos.
  - Comportamiento incorrecto.
  - Debilidades en el filtrado de entradas.
  - Problemas de flujo de datos.
  - Problemas de flujo de llamadas.
  - Código inalcanzable (muerto).
  - Código malicioso insertado deliberadamente (es decir, bombas software).

3.2.1.4 Tipos de pruebas de seguridad en el nivel de prueba de componente

En el nivel de prueba de componente pueden aplicarse pruebas estáticas y pruebas dinámicas.

En función de los objetivos de la prueba de seguridad, la base de prueba, los objetos de prueba y los tipos de prueba, pueden utilizarse diferentes enfoques de diseño y técnicas en el nivel de prueba de componente basado en el código dado:

- Verificar que la implementación de las funciones y los mecanismos de seguridad se comportan según lo esperado por los requisitos de seguridad.

  El diseño de las pruebas de seguridad se basa en requisitos detallados (por ejemplo, especificaciones detalladas y diseño detallado del sistema sujeto a prueba (SSP) [en inglés, system under test (SUT)]. Deben utilizarse técnicas de prueba conocidas basadas en las especificaciones, como el análisis del valor frontera y la partición de equivalencia [ISTQB FL]. El ingeniero de prueba de seguridad (IPS) debe trazar los casos de prueba de seguridad hasta las especificaciones detalladas.

- Construir confianza en la calidad del código de seguridad (es decir, codificación segura)

  Los casos de prueba de seguridad deben concentrarse en la aplicación de reglas de codificación segura. El ingeniero de prueba de seguridad (IPS) [en inglés security test engineer (STE)] también debe verificar que el equipo de desarrollo no utiliza instrucciones de código peligrosas y evita las debilidades de los lenguajes de programación y los compiladores. Normalmente, los equipos de desarrollo o las organizaciones definen sus propias buenas prácticas de codificación segura basándose en referencias conocidas, que pueden ser internas a la organización o externas como la fundación OWASP. El ingeniero de pruebas de seguridad (IPS) puede diseñar casos de prueba basados en estas normas que pueden considerarse requisitos no funcionales (por ejemplo, la mantenibilidad y otras características de calidad no funcionales). Estos casos de prueba de la seguridad pueden procesarse automáticamente utilizando herramientas de análisis estático.

  Las pruebas para cualquier componente deben incluir la evaluación de posibles violaciones de la siguiente lista de comprobación de buenas prácticas: [CERT1]

  - Validar las entradas.
  - Prestar atención a las advertencias del compilador.
  - Crear arquitectura y diseñar políticas de seguridad.
  - “Principio de “mantener la sencillez.
  - Negación por defecto, que define una lista de “permitidos”.
  - Adhesión al principio del menor privilegio.
  - Sanear los datos enviados a otros sistemas.
  - Practicar la defensa en profundidad.
  - Utilizar técnicas eficaces de control de la calidad.
  - Adoptar un estándar de codificación seguro

    Las pruebas realizadas con estas listas de buenas prácticas deben incluir evaluaciones de posibles violaciones de estas prácticas basadas en un análisis del riesgo bien documentado que incorpore una modelización realista de las amenazas.

- Detectar vulnerabilidades en componentes

  Tras verificar la correcta implementación de las funciones y mecanismos de seguridad y que se han seguido las mejores prácticas de codificación segura, el ingeniero de prueba de seguridad (IPS) debe diseñar pruebas de seguridad con el objetivo de encontrar vulnerabilidades en los componentes desarrollados (por ejemplo, pruebas aleatorias de la IPA (en inglés, Application Programming Interface - API) de un componente).

El IPS puede utilizar herramientas de prueba estática de seguridad de las aplicaciones (PESA) o de prueba dinámica de seguridad de las aplicaciones (PDSA) para ayudar a encontrar vulnerabilidades.

- Mitigar los riesgos de seguridad

  Todas las pruebas de seguridad descritas anteriormente ayudan a mitigar los riesgos de seguridad de la aplicación o el sistema desarrollado.

3.2.1.5 Análisis del diseño de pruebas de seguridad en el nivel de prueba de componente

Una medida clave de la adecuación del diseño de la prueba de seguridad implica la evaluación de la cobertura. Varias medidas de cobertura proceden de las pruebas realizadas.

La cobertura puede medirse como cualquiera de las siguientes:

- Porcentaje del número total de requisitos de seguridad probados.
- Porcentaje de casos de uso / abuso de seguridad especificados que se han probado.
- Porcentaje de funciones, escenarios o hilos de misión de seguridad críticos probados.
- Porcentaje de cobertura de código fuente (por ejemplo, para identificar código muerto o puertas traseras).
- Porcentaje de cobertura de particiones de equivalencia de datos (por ejemplo, para detectar malas capturas de excepciones).
- Número de hallazgos de seguridad.
- Eficiencia de las herramientas de seguridad utilizadas (por ejemplo, número de resultados de falso positivo y resultados de falso negativo)

3.2.2 Diseño de pruebas de seguridad en el nivel de integración de componentes

El programa de estudio de nivel básico [ISTQB FL] caracteriza dos niveles diferentes de integración software: prueba de integración de componentes y prueba de integración de sistemas. Los componentes y/o subsistemas que deben integrarse pueden proceder de distintas fuentes, como otro equipo de la misma organización, un subcontratista, un producto comercial disponible en el mercado, un servicio ya disponible en la nube o un servicio de código abierto. Durante estas actividades de integración para construir finalmente el sistema de producción completo, las posibilidades de que se produzcan brechas de seguridad no son simplemente la suma de las vulnerabilidades de cada uno de los componentes. En su lugar, se hacen posibles nuevos vectores de ataque debido a las interacciones entre componentes dentro del sistema más amplio y a causa de elementos organizativos.

Sin embargo, algunas interacciones entre componentes pueden mitigar o bloquear las posibles secuencias que conducen a brechas de seguridad.

La prueba de integración de componentes puede demostrar la complejidad del diseño de un sistema y la estabilidad de su comportamiento. El enfoque de prueba de integración de componentes (por ejemplo, descendente o ascendente) puede afectar al momento en que se revelen los problemas de seguridad o la necesidad de pruebas adicionales específicas para la seguridad.

Al igual que con la prueba de componente, la prueba de integración de componentes debe probarse sobre la base de un análisis del riesgo bien documentado que incorpore un modelado realista de las amenazas. A medida que se integran componentes separados, tenga en cuenta que puede ser necesario el andamiaje o la simulación en forma de stubs y controladores para probar caminos incompletos a través de un sistema durante la integración. A medida que se añaden más componentes implementados al sistema, el andamiaje/la simulación se elimina de forma incremental, lo que permite una evaluación más completa de la adecuación funcional, así como la apertura de nuevos caminos a las vulnerabilidades que podrían ser explotadas.

Según el nivel de confianza en los componentes / subsistemas que se van a integrar, el diseño de prueba de seguridad en el nivel de integración de componentes debe incluir:

- Pruebas de seguridad de la arquitectura de seguridad global basadas en la documentación técnica de la arquitectura.
- Pruebas de seguridad de los flujos integrados configurados (por ejemplo, autorizados o no y el nivel de confidencialidad, integridad y disponibilidad).
- Pruebas de seguridad de las IPA (en inglés, Application Programming Interface - API) integradas (por ejemplo, para detectar dificultades de seguridad en las IPA debidas a la falta de controles o al desconocimiento de dichas IPA).
- Pruebas de seguridad relativas a la configuración de seguridad de los componentes integrados (por ejemplo, filtrar el acceso de un componente por otro, ya que los componentes no firmados deben tener un acceso limitado).
- Verificación de que los componentes integrados que son software externo, de código abierto o de código cerrado están libres de vulnerabilidades.

En el nivel de integración de los componentes, la cobertura puede medirse como cualquiera de los siguientes elementos:

- Porcentaje de IPA's utilizadas/probadas (en inglés, Application Programming Interface - API).
- Porcentaje de interacciones probadas entre componentes/subsistemas basadas en la documentación de la arquitectura técnica.
- Número de hallazgos de seguridad en el nivel de integración de componentes.
- Número de hallazgos de seguridad que deberían haberse encontrado en el nivel de prueba de componentes.
- Eficiencia de las herramientas de seguridad utilizadas (por ejemplo, número de resultados de falso positivo y resultados de falso negativo).

3.2.3 Prueba de seguridad en la prueba de sistema y prueba de aceptación

3.2.3.1 Prueba de sistema de la seguridad

Es el nivel de prueba durante el cual se prueba la implementación de los requisitos de seguridad para asegurar que funcionan como se espera. Las actividades de prueba de seguridad del sistema incluyen la realización de pruebas de seguridad en alguna aproximación del entorno de producción, lo que requiere que se produzca una transición fuera del entorno de desarrollo en el que han tenido lugar las actividades de implementación e integración precedentes.

3.2.3.2 El rol de la prueba de seguridad en la prueba de sistema

La prueba de seguridad del sistema es la primera oportunidad para practicar la funcionalidad de extremo a extremo de los componentes totalmente integrados. Aunque normalmente se realiza en un entorno de prueba, debe revelar propiedades emergentes del sistema que no se habrían observado antes de que se completara la integración. Los requisitos de seguridad suelen tenerse en cuenta junto con los requisitos funcionales.

El objetivo de la prueba de seguridad en la prueba de sistema es probar:

- todos los requisitos de seguridad implementados en las funciones de seguridad en un entorno de prueba que represente el entorno de producción.
- que la configuración de la operación es segura.

3.2.3.3 Prueba de aceptación de la seguridad

Se trata del nivel final de prueba durante el cual los usuarios, o sus representantes, adquieren la confianza de que el sistema es capaz de ofrecer las capacidades necesarias en el entorno de producción de forma segura. Los objetivos de la prueba de aceptación de seguridad incluyen la prueba de seguridad frente a los criterios de aceptación relacionados con la seguridad establecidos para el sistema. Normalmente, los criterios de aceptación relacionados con la seguridad se concentran en los controles y procesos de seguridad funcionales. Las actividades de la prueba de aceptación de seguridad pueden incluir:

- Instalar el sistema en un entorno de preproducción.
- Realizar pruebas de seguridad basadas en los criterios de aceptación.
- Determinar la aceptación basándose en los resultados de las pruebas de seguridad.

3.2.3.4 El rol de la prueba de seguridad en la prueba de aceptación

La prueba de aceptación se distingue de la prueba de sistema en que se realiza en un entorno parecido al de producción. Por último, sitúa el sistema en el entorno en el que los agentes de amenazas externas tratarían de encontrar debilidades en el día a día. Estas pruebas permiten una evaluación razonable de la eficiencia de desempeño y otros comportamientos basados en interacciones a través de interfaces externas.

Lo ideal sería que las pruebas de aceptación validaran que se han alcanzado los objetivos iniciales del proyecto y que se cumplen los criterios de aceptación de seguridad documentados. Esto se consigue diseñando y realizando pruebas para validar procesos / escenarios de seguridad como el control de derechos, la gestión de autorizaciones y el filtrado de cortafuegos.

El mejor momento para definir y documentar los criterios de aceptación es antes del desarrollo o de la adquisición del sistema. En el contexto de la prueba de seguridad, los criterios de aceptación pueden ser de naturaleza global. Por ejemplo, podría haber criterios de aceptación que especifiquen lo que es aceptable en términos de seguridad global del sistema. Esto incluiría criterios que se aplican a todas las funciones del sistema, como la autenticación del usuario, los derechos de usuario, los niveles de cifrado y los rastros de auditoría.

3.2.3.5 Análisis del diseño de la prueba de seguridad en el nivel de prueba de aceptación

En el nivel de prueba de aceptación, la cobertura puede medirse de la siguiente manera:

- Porcentaje de procesos / escenarios de seguridad probados.
- Número de hallazgos de seguridad que deberían haberse encontrado en los niveles de prueba previos con su severidad.
- Eficiencia de las herramientas de seguridad utilizadas (por ejemplo, número de resultados de falso positivo y resultados de falso negativo).
- Porcentaje de requisitos de seguridad probados.
- Porcentaje de elementos de la configuración de seguridad operativa probados.


SSTQB
Spanish Software Testing Qualifications Board

4 Estándares y buenas prácticas de la prueba de seguridad - (K3)

Duración: 195 minutos

Palabras clave[4]

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB**: identificarán palabras clave del proceso de prueba
- **ESPDOM**: identificarán palabras clave específicas de dominio: **SEGURIDAD**

Tipo Palabra Clave	Español	Inglés
ISTQB	Enumeración y Clasificación de Patrones de Ataque Comunes [en inglés, Common Attack Pattern Enumeration and Classification (CAPEC)]	Common Attack Pattern Enumeration and Classification (CAPEC)
ISTQB	Vulnerabilidades y Exposiciones Comunes [en inglés, Common Vulnerabilities and Exposures (CVE)]	Common Vulnerabilities and Exposures (CVE)
ISTQB	Sistema de Puntuación de Vulnerabilidades Comunes [en inglés, Common Vulnerability Scoring System (CVSS)]	Common Vulnerability Scoring System (CVSS)
ISTQB	Enumeración de Debilidades Comunes [en inglés, Common Weakness Enumeration (CWE)]	Common Weakness Enumeration (CWE)
ISTQB	Sistema de Puntuación de Debilidades Comunes [en inglés, Common Weakness Scoring System (CWSS)]	Common Weakness Scoring System (CWSS)
ISTQB	vulnerabilidad	vulnerability
ISTQB	debilidad	weakness

[4] Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.

Objetivos de aprendizaje para "Capítulo 4"

4.1 Introducción a los estándares y buenas prácticas de seguridad

STE - 4.1.1 (K3) Explicar las diferentes fuentes de estándares, buenas prácticas y su aplicabilidad.

4.2 Aplicación de estándares importantes y buenas prácticas en la prueba de seguridad

STE - 4.2.1 (K3) Aplicar el concepto del proyecto abierto de seguridad de las aplicaciones web, la enumeración de vulnerabilidades comunes, la enumeración de debilidades comunes, el sistema de puntuación de vulnerabilidades comunes y el sistema de puntuación de debilidades comunes y cómo aprovecharlos en la prueba de seguridad.

4.3 Aprovechamiento de estándares y buenas prácticas de la prueba de seguridad

STE - 4.3.1 (K2) Explicar las ventajas y desventajas de los oráculos de prueba utilizados para la prueba de seguridad.

STE - 4.3.2 (K3) Comprender las ventajas y desventajas de utilizar los mejores estándares y las mejores prácticas de seguridad

4.1 Introducción a los estándares y buenas prácticas de seguridad

Las normas y las buenas prácticas de diversos tipos proporcionan visibilidad al consenso profesional y a las obligaciones reglamentarias. Un estándar basado en el consenso representa la opinión ponderada de un conjunto de expertos bien informados.

Aunque a menudo se utilicen como sinónimos, existen grandes diferencias entre los estándares y las buenas prácticas, que se explican en las siguientes subsecciones. Las diferencias tienen un impacto significativo en el proceso de selección y en los posibles casos de uso para utilizarlas.

4.1.1 Estándares y buenas prácticas

4.1.1.1 Estándares

Los estándares se definen como «un documento, establecido por consenso de expertos en la materia y aprobado por un organismo reconocido, que proporciona orientación sobre el diseño, el uso o el rendimiento de materiales, productos, procesos, servicios, sistemas o personas». ([ISO_Web_21], y Apéndice D).

Existen varios niveles de «organismo reconocido», lo que permite distinguir entre distintos tipos de estándares. Uno de los niveles más altos de reconocimiento de estándares está representado en todo el mundo por la Organización Internacional de Normalización (ISO). Normalmente, cada país que forma parte de la Organización Mundial del Comercio (OMC), tiene su propia representación local. Los estándares tienen el mayor nivel de reconocimiento para un ingeniero de prueba de seguridad (IPS) debido a su alto nivel de madurez. Sin embargo, esta madurez tiene la desventaja de que lleva mucho tiempo completarlas y a menudo da como resultado estándares con un carácter muy reactivo.

Los organismos reconocidos pueden crear sus propios estándares. Éstos pueden clasificarse en normas industriales, normas de facto y normas específicas de los fabricantes:

- Estándares industriales:
  - Se han establecido a lo largo de años de aplicación en muchos contextos y han demostrado algún valor añadido al resolver un problema concreto. El «Internet Engineering Task Force (IETF)» es un actor importante en la creación de estándares a este nivel. Elaboran estándares basándose en el juicio combinado de ingeniería de sus participantes y en su experiencia en el mundo real a la hora de implementar y desplegar su especificación [IETF23].
- Estándares de facto:
  - Suelen tener sus raíces en los estándares de la industria. Dado que su cobertura y aceptación son elevadas, incluso cumplen muchos de los criterios para tener en cuenta el nivel más alto de los estándares. Un buen ejemplo es el protocolo TCP, que se estableció como estándar de la industria pero que hoy se tiene en cuenta como estándar de facto [IETF23].
- Estándares específicos del fabricante:
  - Algunos clientes/organizaciones han aprendido que hay un valor añadido en seguir las especificaciones propias de un fabricante específico.

En la vida real, esta clasificación clara puede tener muchos solapamientos difusos, y no siempre es sencillo hacer una clasificación clara de un estándar determinado.

4.1.1.2 Buenas prácticas

Las buenas prácticas y su organismo reconocido están organizados de manera menos formal. El Glosario de Gartner [Gart21], define las buenas prácticas como un «grupo de tareas que optimiza la eficiencia (coste y riesgo) o la efectividad (nivel de servicio) de la disciplina o el proceso de negocio al que contribuye. Debe poder implementarse, reproducirse, transferirse y adaptarse a todos los sectores». A este nivel, cada grupo de expertos, aunque trabaje en el mismo contexto, puede crear su propio conjunto de buenas prácticas.

4.2 Aplicación de estándares importantes y buenas prácticas en la prueba de seguridad

Existen varios estándares y buenas prácticas para la disciplina de prueba de seguridad. Debido al alto nivel de requisitos que deben cumplirse para ser considerados un estándar, su creación y mantenimiento es mucho más lento que en el caso de las buenas prácticas. Esto permite un profundo reconocimiento dentro de la industria e incluye muchos bucles de retroalimentación para la mejora. Sin embargo, esto impide la rápida adaptación a las nuevas tendencias y riesgos. En comparación, las buenas prácticas tienen una alta eficiencia de desempeño general, pero es más difícil que lleguen a ser bien conocidas, que alcancen un alto nivel de cobertura y que estén empoderadas por evidencias prácticas.

4.2.1 Estándares industriales para la prueba de seguridad

El estándar más establecido en seguridad informática es la serie de normas ISO 27000. El estándar [ISO 27001] está aceptado internacionalmente y se titula «Tecnología de la información - Técnicas de seguridad - Sistemas de gestión de la seguridad de la información - Visión general y vocabulario». Este estándar se concentra en la gestión de la seguridad de la información, es decir, en identificar los riesgos, evaluarlos y gestionarlos mediante controles de seguridad de la información. Todas estas actividades se combinan en un sistema de gestión de la seguridad de la información (SGSI) que es el núcleo general de la norma ISO 27000. El estándar es de amplio alcance y se concentra en la forma general en que una organización debe evaluar sus riesgos, contrastarlos con sus necesidades específicas y ocuparse de los riesgos más relevantes. El estándar básico puede aplicarse a todas las organizaciones.

La serie ISO 27000 consta de más de 40 estándares individuales, que pueden clasificarse en los siguientes:

- Estándar principal: Visión general e introducción a un SGSI (empezando por la ISO 27000 hasta la ISO 27005)
- Estándares de temas específicos para cubrir temas concretos como la gestión de servicios ISO 27013 y el proveedor de nube pública ISO 27017 (véase [ITGOV23a])
- Estándares específicos de dominio para concentrarse en dominios concretos como los proveedores de telecomunicaciones ISO 27011 y la industria financiera ISO 27015 (véase [ITGOV23a])

Los estándares más utilizados que se aplican al contexto de la prueba de seguridad y cubren los objetos de prueba y las condiciones de prueba relevantes que el ingeniero de prueba de seguridad (IPS) debe tener en cuenta, son los siguientes:

- ISO 27000: En esta parte se explica la estructura general de la serie ISO 27000 y se presenta un SGSI y el rol que pueden desempeñar las pruebas de seguridad.
- ISO 27001: Es el estándar más utilizado, ya que enumera un amplio conjunto de recomendaciones y controles de seguridad para estructurar y construir un SGSI individual. Se concentra en establecer una visión global de los activos relevantes dentro de una organización, sus riesgos expuestos y las posibles mitigaciones. [ISO 27001]
- ISO 27001, Apéndice A: La parte más importante de la ISO 27001 para un ingeniero de prueba de seguridad (IPS) se presenta en este apéndice. Enumera los controles de seguridad para diferentes aspectos como el control de acceso, la recuperación de desastres y la seguridad de red. Cada uno de estos controles, si se aplican en un contexto específico, son entradas importantes para un STE, ya que es su tarea medir la efectividad de un control de seguridad. [Cald11]
- ISO 27002: Este estándar toma los controles de seguridad genéricos de la norma ISO 27001 y ofrece más orientación sobre cómo aplicarlos en la práctica y cómo especificarlos con más detalle para un contexto específico. [Cald11]
- ISO 27003: Este estándar ayuda a una organización a crear un plan para establecer un SGSI basado en la norma ISO 27001.

4.2.1.1 Estándares de facto para la prueba de seguridad

Hay muchos estándares de facto que pueden ser aprovechados por un ingeniero de prueba de seguridad (IPS) [en inglés security test engineer (STE)]. Una de las series más importantes de estándares de facto procede de la corporación MITRE, aunque su negocio principal no sea generar estándares. MITRE es una organización privada sin ánimo de lucro que proporciona orientación técnica y de ingeniería al gobierno federal estadounidense. Los patrocinadores más importantes de MITRE son el Departamento de Defensa, la Administración Federal de Aviación y el Departamento de Seguridad Nacional [MITRE21].

En el ámbito de las pruebas de seguridad, MITRE acoge y mantiene los siguientes estándares bien conocidos que proporcionan un valor añadido para el ingeniero de pruebas de seguridad (IPS) [en inglés security test engineer (STE)]:

- **Common Attack Pattern Enumeration and Classification (CAPEC™) [Enumeración y Clasificación de Patrones de Ataque Comunes]**

  CAPEC™ proporciona un catálogo de patrones de ataque comunes a disposición del público. La idea es conseguir una mejor comprensión de cómo los atacantes explotan las debilidades de las aplicaciones y otras capacidades cibernéticas. Los patrones de ataque se basan en patrones de diseño de software para atacantes. Dos patrones de ataque de entrada típicos son la inyección SQL (CAPEC-66) y el cruce de caminos relativos (CAPEC-139) [CAPEC21].

  CAPEC ofrece diferentes puntos de vista sobre sus conjuntos de datos. Las más relevantes son:

  - Dominio del ataque, como software, ingeniería social y seguridad física. En el nivel más alto, CAPEC enumera nueve dominios de ataque.
  - Mecanismos de ataque, como inyectar elementos inesperados y manipular recursos del sistema. En el nivel más alto, CAPEC enumera seis mecanismos de ataque.

  Cada objeto de prueba que prueba un ingeniero de prueba de seguridad (IPS) debe ubicarse dentro de este catálogo. A menudo, CAPEC es el punto de partida para obtener una visión general inicial de los posibles ataques que podrían ser relevantes para un sistema determinado.

Los siguientes estándares del MITRE se utilizan para refinar con el propósito de lograr pruebas de seguridad eficaces:

- **Enumeración de Debilidades Comunes [en inglés, Common Weakness Enumeration (CWE)]**

  CWE es una lista de debilidades de software/hardware. Normalmente, cada patrón de ataque común tiene una o más debilidades que son utilizables para aprovechar un patrón de ataque CAPAC [CWE21]. CWE utiliza el concepto de vistas, las más utilizadas de las cuales son:

  - Desarrollo de software, como una IPA (en inglés, Application Programming Interface - API), malas prácticas de código y dificultades con los permisos. En el nivel más alto, CWE enumera 40 activos de desarrollo de software.
  - Diseño de hardware, como dificultades con la memoria y el almacenamiento, problemas con el núcleo y los periféricos, interconexión integrada en chip y problemas de E/S de interfaz. En el nivel más alto, CWE enumera 12 activos de diseño de hardware.

  Cada debilidad común es un punto de partida eficaz para que un ingeniero de prueba de seguridad (IPS) pruebe si el patrón de ataque subyacente puede ser explotado.

- **Open Web Application Security Project (OWASP)**

  Es importante darse cuenta de que el CWE y el OWASP [OWASP21] se solapan y ambos enumeran debilidades comunes. OWASP es bien conocido por publicar su clasificación OWASP Top 10.

- **Sistema de Puntuación de Debilidades Comunes [en inglés, Common Weakness Scoring System (CWSS)]**

  Cuanto más común se vuelve una debilidad, más importante es disponer de un esquema de priorización. El CWSS proporciona un mecanismo para priorizar las debilidades de forma consistente, flexible y abierta [CWSS21]. La priorización se calcula mediante tres grupos de métricas:

  - Grupo de métricas de hallazgos base:

    Se calcula el riesgo inherente de una debilidad, la confianza en la exactitud del hallazgo y la solidez de los controles. Una métrica típica es el «impacto técnico», que va desde el control total sobre un sistema hasta la ausencia de impacto técnico.

  - Grupo de métricas de superficie de ataque:

    Calcula las barreras que debe superar un atacante para explotar la debilidad. Una métrica típica es el privilegio requerido, que va desde ningún privilegio requerido hasta privilegios de administrador.

  - Grupo de métricas de entorno:

    Calcula las características de las debilidades que son específicas de un entorno o contexto operativo concreto. Una métrica típica es el impacto en el negocio, que va desde que el negocio podría fallar hasta ningún impacto.

  Mediante el uso de factores de ponderación específicos y predefinidos, todas estas métricas pueden agregarse en un valor CWSS global para una debilidad específica. El CWSS puede tratar métricas desconocidas mediante valores predeterminados o definiendo/concentrándose en un subconjunto de métricas individuales. Además, muchas métricas del grupo de métricas de hallazgos base pueden calcularse automáticamente mediante una herramienta de análisis estático.

- **Sistema de Puntuación de Vulnerabilidades Comunes [en inglés, Common Vulnerability Scoring System (CVSS)]**

  Un mecanismo de priorización similar a CWSS es CVSS [CVSS21], que sigue un enfoque similar, pero asume una vulnerabilidad existente y desplegada (véase CVE más adelante). Tanto CVSS como CWSS son sistemas de puntuación para la seguridad informática: CVSS es un enfoque reactivo porque las vulnerabilidades ya existen antes de la clasificación. CWSS es un enfoque proactivo, ya que se trabaja con el software antes de entregarlo a producción. Ambos enfoques se utilizan a menudo juntos, aunque no sean totalmente compatibles (cf. [SecJour21]).

- **Vulnerabilidades y Exposiciones Comunes [en inglés, Common Vulnerabilities and Exposures (CVE)]**

  CVE es una base de datos de información divulgada públicamente sobre dificultades de seguridad [CVE21]. Un número CVE identifica de forma exclusiva una vulnerabilidad concreta de la lista. CVE es útil porque proporciona un identificador estándar para una vulnerabilidad determinada dentro de un sistema específico. Si un sistema está afectado por un CVE específico, esta vulnerabilidad es una instancia específica de una debilidad común (CWE) que puede utilizarse para realizar un ataque específico (CAPEC). Las nuevas entradas en el repositorio de Vulnerabilidades y Exposiciones Comunes (CVE) suelen tener su origen en el trabajo diario de los IPS. Si identifican una nueva vulnerabilidad desconocida para CVE, pueden publicarla en CVE para involucrar a la comunidad de seguridad en la identificación de contramedidas.

4.2.1.2 Buenas prácticas para la prueba de seguridad

Las buenas prácticas sólo necesitan alcanzar un criterio formal bajo para ser consideradas como tales. Muchas buenas prácticas pueden fallar al cabo de un tiempo si no sirven de nada. Algunas dejarán de utilizarse por falta de publicación/comercialización, pero unas pocas podrían mejorar su madurez en su camino a ser tenidas en cuenta para un estándar.

Una buena práctica madura típica que se sigue utilizando hoy en día es el modelo STRIDE, inventado por Microsoft [Micro09]. STRIDE permite el modelado sistemático de amenazas desde la perspectiva del atacante. El término en sí

es un acrónimo de seis categorías de amenazas, que clasifica las amenazas potenciales: suplantación de identidad, manipulación, repudio, revelación de información, denegación de servicio y elevación de privilegios.

Suplantación de identidad, es decir, pretender dentro de un sistema ser una persona o un sistema que no se es.	**S**	Spoofing identity
Manipulación de datos, es decir, la modificación maliciosa de datos.	**T**	Tampering with data
Repudio, es decir, amenazas que tienen como objetivo la auditoría y el rastreo, asegurando que los malos comportamientos no puedan ser demostrados	**R**	Repudiation
Revelación de información, es decir, la exposición de información a personas que se supone que no tienen acceso a ella.	**I**	Information disclosure
Denegación de servicio, es decir, denegar el servicio a usuarios válidos.	**D**	Denial of service
Elevación de privilegios, es decir, que un usuario sin privilegios obtenga acceso privilegiado.	**E**	Elevation of privilege

En general, el STRIDE se utiliza para ayudar a los desarrolladores a tener en cuenta las amenazas durante el diseño y a cerrar los vacíos identificados. El ingeniero de pruebas de seguridad (IPS) puede utilizar el mismo enfoque para concentrarse en la prueba.

4.3 Aprovechamiento de estándares y buenas prácticas de prueba de seguridad

Hay muchos casos de uso posibles para aprovechar los estándares y las buenas prácticas. En general, estos casos de uso pueden dividirse en aplicaciones obligatorias y aplicaciones voluntarias.

4.3.1 Aplicación obligatoria de estándares y buenas prácticas

En este tipo de caso de uso, los estándares y las buenas prácticas son obligatorios para un tercero:

- **Requisitos de seguridad en contratos:**
  - Las buenas prácticas son una forma eficaz de especificar los requisitos de seguridad para el desarrollo de software, especialmente aquellos que se delegan en un tercero. En lugar de enumerar todos los requisitos específicos, sólo se exige el cumplimiento de un estándar concreto, lo que implica cumplir todos los consejos y requisitos de seguridad contenidos.
- **Requisitos de seguridad como normativa:**
  - Incluso las instituciones reguladoras (por ejemplo, en el dominio bancario) suelen utilizar estándares y buenas prácticas, que son fáciles de gestionar.

4.3.2 Aplicación voluntaria de estándares y buenas prácticas

En este tipo de caso de uso, la aplicación de estándares específicos y buenas prácticas es una decisión de la dirección para generar el siguiente valor añadido:

- Establecimiento de un alto nivel de seguridad mediante la reutilización de los conocimientos establecidos en materia de seguridad almacenados en los estándares y las buenas prácticas existentes.
- Evidencia bien conocida para demostrar la concienciación por la seguridad
- Propósito general de marketing y creación de puntos de venta únicos en un área de negocio competitiva.

4.3.3 Oráculos de prueba extraídos de estándares y buenas prácticas

Un caso de uso general para aprovechar los estándares y las buenas prácticas que es independiente de ser obligatorio o voluntario es la noción de utilizar potentes oráculos de prueba. A nivel de aplicación, el oráculo de prueba suele ser la sección de requisitos de seguridad de la especificación. En niveles inferiores, por ejemplo, las bibliotecas incluidas, el sistema operativo subyacente, el tráfico de red, los estándares y las buenas prácticas pueden utilizarse fácilmente. Especialmente el tipo más volátil de buenas prácticas podría enumerar muchas vulnerabilidades conocidas para un sistema dado y determinar los resultados esperados que se utilizarán como prueba de que es seguro o no. Se trata de una poderosa herramienta para los IPS, ya que sólo deben definir los correspondientes casos de prueba de bajo nivel, ejecutarlos y, a continuación, comparar los resultados de la prueba con los que figuran en la lista de buenas prácticas.

4.3.4 Ventajas y desventajas de aprovechar los estándares y las buenas prácticas para la prueba de seguridad

Aprovechar los estándares y las buenas prácticas para las pruebas de seguridad tiene muchas ventajas, pero hay algunos aspectos negativos que deben tenerse en cuenta cuidadosamente para un contexto específico. En general, las siguientes ventajas se aplican al aprovechamiento de los estándares y las buenas prácticas:

- **Terminología consistente:**
  - En TI existen muchos términos de marketing, sinónimos y frases sin una definición o distinción clara entre ellos. Los estándares y a veces incluso las buenas prácticas pueden contribuir a aclarar la terminología.
- **Reutilización del conocimiento experto:**
  - Definir los estándares y las buenas prácticas puede ser una tarea que lleve mucho tiempo y que suelen realizar los expertos en seguridad. Sus conocimientos pueden captarse y reutilizarse en estándares y buenas prácticas.
- **Doble comprobación comparativa y de completitud:**
  - Si una empresa utiliza su propio marco de trabajo específico para las pruebas de seguridad, las normas existentes y las mejores prácticas pueden utilizarse como punto de referencia para comprobar la completitud de sus soluciones.
- **Mayor compromiso entre proveedor y cliente:**
  - Cuanto más establecido y reconocido esté un estándar o una buena práctica, más eficazmente podrá utilizarse como base para el compromiso entre el consumidor (lo que quiere tener) y el proveedor (lo que debe hacer).
- **Facilitar la comunicación sobre el nivel de seguridad alcanzado:**
  - Si una organización utiliza su propio conjunto de pruebas de seguridad sin ninguna referencia externa, puede resultar difícil demostrar su efectividad. El uso de estándares y buenas prácticas ayuda a conseguir una actitud general positiva y simplifica la comunicación de forma espectacular.

Sin embargo, es posible que se presenten algunos aspectos negativos cuando se utilizan estándares y buenas prácticas para la prueba de seguridad:

- **Selección errónea:**
  - Hay muchos estándares y buenas prácticas disponibles, cada uno con su propio enfoque y las precondiciones necesarias para ser aplicable. Aprovechar la fuente equivocada reduce el impacto de lograr una mejor seguridad y podría incluso disminuir los recursos disponibles para dedicar a la seguridad.
- **Buenas prácticas en un contexto específico erróneo:**
  - Mientras que la mayoría de los estándares han alcanzado una gran calidad debido a sus largos procesos de creación y a sus largos ciclos de retroalimentación, las buenas prácticas pueden aparecer y desaparecer a corto plazo. Especialmente cuando se proponen inicialmente, su corrección y su valor añadido no siempre están claros y pueden no tener un vínculo sólido con el contexto específico. El uso de una buena práctica nueva y patentada que aún no ha demostrado crear ningún valor añadido podría incluso disminuir el nivel de seguridad.
- **Falta de adaptación:**
  - Con frecuencia, los estándares y las buenas prácticas definen ciertos parámetros que deben cumplirse para ser aplicables en un contexto específico. Si esto se omite o no se hace correctamente, su aplicación podría aportar un valor añadido limitado.
- **Consideraciones de producto básico:**
  - Cuanto más establecido y popular se hace un estándar o una buena práctica, menos puede utilizarse para crear singularidad en comparación con otros productos de la competencia (si es necesario).
- **Ceguera operativa:**
  - Los desperfectos o el surgimiento de nuevos hilos conductores podrían reducir la atención si las normas o las buenas prácticas no se adoptan a tiempo.

5 Adaptación de la prueba de seguridad al contexto de la organización - (K4)

Duración: 195 minutos

Palabras clave[5]

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB**: identificarán palabras clave del proceso de prueba
- **ESPDOM**: identificarán palabras clave específicas de dominio: **SEGURIDAD**

Tipo Palabra Clave	Español	Inglés
ISTQB	debilidad	weakness
ESPDOM	rootkit	rootkit

[5] Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.

Objetivos de aprendizaje para "Capítulo 5"

5.1 Impacto de las estructuras de una organización en el contexto de la prueba de seguridad

STE - 5.1.1 (K3) Analizar un contexto dado de una organización y determinar qué aspectos específicos hay que tener en cuenta para la prueba de seguridad.

5.2 Impacto de normativas en las políticas de seguridad y cómo probarlas

STE - 5.2.1 (K3) Analizar el impacto de las normativas en las políticas de seguridad y cómo probarlas.

5.3 Análisis de un escenario de ataque

STE - 5.3.1 (K4) Analizar un escenario de ataque e identificar las posibles fuentes y motivaciones del ataque.

5.1 Impacto de las estructuras de una organización en el contexto de la prueba de seguridad

La seguridad de la información no puede lograrse únicamente protegiendo la infraestructura y confiando en medidas implementadas tecnológicamente. También deben tenerse en cuenta las personas y los procesos de una organización.

5.1.1 Analizar el contexto de una organización específica y determinar qué aspectos concretos deben tenerse en cuenta en la prueba de seguridad

Las personas suelen ser víctimas de ataques de ingeniería social y procesos importantes, como una respuesta de emergencia definida, pueden faltar o no estar correctamente implementados. Por lo tanto, un ingeniero de prueba de seguridad (IPS) [en inglés security test engineer (STE)] debe cubrir estos aspectos durante la prueba de seguridad, ya que ambos afectan a la seguridad de la información de una organización. Las personas tienen un rol definido. Dependiendo de su rol, participan en diferentes procesos. Los roles y procesos suelen depender en gran medida de la estructura organizativa.

Las estructuras organizativas pueden clasificarse en tres tipos:

- **Estructura funcional:**
  - organizada por funciones comunes, como producción, marketing, recursos humanos, TI y contabilidad.
- **Estructura por divisiones:**
  - organizada como una colección de funciones que producen un producto.
- **Estructura matricial:**
  - los empleados se agrupan simultáneamente por función y producto.

A continuación, se tiene en cuenta la forma en que estas estructuras organizativas afectan al flujo de información y a la implementación de decisiones administrativas:

- En una organización estructurada funcionalmente, la información necesita ser intercambiada entre los diferentes departamentos de la organización y las decisiones administrativas se implementan directamente en un enfoque descendente desde la dirección a toda la organización.
- Las organizaciones estructuradas por divisiones añaden una capa administrativa en la parte superior de cada división, y por lo tanto las decisiones sólo pueden afectar a una sola división. Además, el flujo de información entre divisiones se reduce ligeramente, aunque las divisiones a menudo contienen departamentos similares, como el de desarrollo.
- Las estructuras matriciales intentan unir ambos aspectos. Están separadas funcionalmente, pero también adoptan un enfoque basado en el producto sin añadir la capa administrativa separada que se encuentra en las estructuras divisionales. Sin embargo, existe un mayor riesgo de conflictos, ya que las decisiones pueden tomarse tanto desde una perspectiva de producto como desde una perspectiva puramente administrativa.

Llevando esto al contexto de la prueba de seguridad, un ingeniero de prueba de seguridad (IPS) [en inglés security test engineer (STE)] debe conocer la estructura organizativa por las siguientes razones:

1. Los resultados de una prueba de seguridad estarán bajo la influencia del departamento que haya ordenado y planificado la prueba.
2. Dependiendo de la estructura general de la organización, un ingeniero de pruebas de seguridad (IPS) [en inglés security test engineer (STE)] puede aprovechar las debilidades en el flujo de información.

La primera de estas dos razones es el resultado del hecho de que los departamentos de TI y seguridad suelen estar autorizados a implementar y garantizar la seguridad, pero otros departamentos también pueden tener conocimiento de ello. Por ejemplo, saber que un empleado del equipo de pruebas de penetración está visitando otro departamento aumentará la probabilidad de que las personas de ese departamento cumplan con las políticas de seguridad, al menos durante el tiempo de su estancia.

La segunda de las dos razones anteriores resulta de la probabilidad de que existan debilidades particulares para cada estructura organizativa. En una estructura funcional, un ingeniero de prueba de seguridad (IPS) [en inglés security test engineer (STE)] podría aprovechar esto de la siguiente manera:

- Los empleados de diferentes departamentos podrían no conocer a personas de otros departamentos, en particular a los responsables de la administración de TI.
- La concienciación sobre seguridad y la aceptación de ciertas medidas de seguridad podrían ser significativamente menores en departamentos donde solo trabaja personal no técnico.
- La información sobre una incidencia puede permanecer en un solo departamento durante un período de tiempo, lo que, en el peor de los casos, puede retrasar el tiempo de respuesta.

En las organizaciones estructuradas por divisiones, se puede tener en cuenta una debilidad potencial similar, aunque esta puede transferirse a las divisiones:

- Los empleados de diferentes departamentos pueden no conocer a personas de otros departamentos. Dependiendo de cómo se mantenga la infraestructura de TI de la organización, es posible que no sepan quién es el responsable de la administración de TI.
- La información sobre una incidencia puede residir en un solo departamento durante un período de tiempo, lo que resulta en retrasos en el tiempo de reacción.
- Dependiendo del tamaño de una división, los equipos también pueden subdividirse por funcionalidad en departamentos más pequeños dentro de la división, lo que hace que el punto anterior sobre la concienciación en materia de seguridad y la aceptación de ciertas medidas de seguridad también sea válido para las estructuras por divisiones.

Aunque las organizaciones con una estructura por divisiones distribuyen funcionalidades similares entre diferentes divisiones, algunos servicios, como la administración de TI, suelen residir en un departamento central.

En las organizaciones matriciales, una persona podría aprovecharse de posibles conflictos entre la gestión administrativa y la gestión de producto. Sin embargo, este no es siempre el caso y, como ya se ha mencionado para las organizaciones estructuradas por divisiones, algunos servicios pueden estar centralizados.

Consideraciones

La información anterior adopta una perspectiva de alto nivel de las estructuras organizativas y sus posibles debilidades. En la práctica, muchas organizaciones no tienen estructuras puramente funcionales, por divisiones o matriciales. En particular, la función de gestión de la seguridad suele estar a cargo de un departamento central que dicta medidas de seguridad, como una política de seguridad para toda la organización.

Una política de seguridad puede definirse como «un documento de alto nivel que describe los principios, el enfoque y los principales objetivos de la organización en materia de seguridad» [Glosario del ISTQB]. Un servicio de seguridad, según el NIST, se define como «una capacidad que respalda uno o varios de los objetivos de seguridad. Ejemplos de servicios de seguridad son la gestión de claves, el control de acceso y la autenticación» [NIST02].

El estudio de las políticas de seguridad de la organización puede revelar posibles vectores de ataque al dar a conocer las restricciones impuestas al comportamiento de sus miembros, así como las restricciones impuestas a los adversarios por los mecanismos de seguridad. Es posible que las políticas de seguridad de la empresa no estén disponibles públicamente. Algunas políticas organizativas pueden ser accesibles únicamente para los empleados o incluso solo para ciertos miembros del personal.

Un aspecto importante a tener en cuenta en el contexto organizativo es la forma en que la organización subcontrata partes de su producción o servicios. También se debe considerar a los socios relevantes como posibles objetivos para una prueba de seguridad. Esto depende de la naturaleza y el contenido del contrato entre las dos organizaciones que definen las obligaciones legales. A los socios externos a menudo se les da acceso (limitado) a la Red Privada Virtual (RPV), trabajan en el mismo repositorio de códigos o tienen un token de acceso para una oficina local. Aunque a menudo tienen cuentas restringidas, podría ser el primer paso hacia un ataque exitoso. Otro aspecto importante en relación con los socios en el contexto de la prueba de seguridad es el análisis de la cadena de suministro, ya que los ataques en este ámbito pueden tener graves consecuencias (por ejemplo, [WIRED21]).

Los aspectos generales tratados en esta sección podrían aplicarse a casi cualquier organización, pero no tienen en cuenta dificultades industriales específicas, como el tipo de producto o servicio que ofrece una organización, así como el sector industrial en el que opera la organización. Ofrecer un servicio web para música podría tener requisitos de seguridad diferentes en comparación con un dispositivo médico utilizado en un hospital. Precisamente por esa razón, existen regulaciones para ciertos sectores que prescriben requisitos para procesos, seguridad, medidas de protección u otros aspectos específicos de dominio. (véase la sección 5.2). Esto podría afectar a las organizaciones que desarrollan sus propios productos mucho más que a las organizaciones que, por ejemplo, venden productos comerciales.

5.2 Impacto de normativas en las políticas de seguridad y cómo probarlas

Las regulaciones de seguridad impulsan el contenido de las políticas de seguridad, que a su vez impulsan el marco de trabajo de control de seguridad de la información para las pruebas de seguridad. El analista de prueba de seguridad (APS) [en inglés, security test analyst (STA)] desarrolla este marco de trabajo de control. Con conocimiento del marco de trabajo, un ingeniero de prueba de seguridad (IPS) [en inglés, security test engineer (STE)] desarrolla y utiliza casos de prueba para poner a prueba los controles.

5.2.1 El impacto de la normativa oficial en las normas de seguridad

Debido a la fuerte interconectividad de la mayoría de los sectores industriales, los ataques a la ciberseguridad pueden tener un profundo impacto en una sola organización y, en el peor de los casos, en la infraestructura general de todo un país. Como reacción a esto, los gobiernos han definido normativas para obligar a las organizaciones críticas a adaptar su nivel de seguridad al menos a un mínimo. Las organizaciones que no cumplen las normas pueden ser multadas o cerradas temporalmente hasta que se implementen las medidas de seguridad exigidas. Por lo tanto, las organizaciones afectadas por las normativas tienen un incentivo para mejorar sus medidas y políticas de seguridad hasta alcanzar al menos el nivel de seguridad exigido. El NIST define las leyes y normativas en el contexto de la seguridad informática como «leyes, normativas, políticas, directrices, estándares y procedimientos específicos de las organizaciones y de todo el gobierno federal que imponen requisitos para la gestión y protección de los recursos de las tecnologías de la información». Aunque esto puede ser cierto en algunos casos, las normativas generales pueden definirse a nivel global, sindical o nacional, como por ejemplo:

- Normativas globales definidas, por ejemplo, por la Organización Mundial de Comercio [en inglés, World Trade Organization (WTO)].
- Reglamentos específicos de la Unión, como el GDPR y los Reglamentos sobre Redes y Sistemas de Información (NIS) definidos por la Unión Europea (UE)
- Normativas nacionales como la Ley de Intercambio de Seguridad Cibernética de los Estados Unidos de América (EE.UU.)

Además, pueden aplicarse otras normativas a sectores industriales específicos, como:

- Ley de portabilidad y responsabilidad del seguro médico [HIPAA].
    - (“Health Insurance portability and accountability Act [HIPAA]”)
- UNECE WP.29 para el sector de la automoción [UNECE20]
    - (“UNECE WP.29 for automotive sector [UNECE20]”)
- Reglamento de implementación (DVO) (UE) 2019/1583 para la seguridad de la aviación [BSI21]
    - (“Implementing Regulation (DVO) (EU) 2019/1583 for aviation security [BSI21]”)
- PCI DSS [PCI22]

La formulación de la normativa corre a cargo de instituciones específicas como la Agencia de Ciberseguridad y Seguridad de las Infraestructuras (“Cybersecurity and Infrastructure Security Agency”) en EE.UU., o la Oficina Federal de Seguridad de la Información (“Federal Office for Information Security”) en Alemania. En la UE, la Agencia Europea de Seguridad de las Redes y de la Información (“European Network and Information Security Agency”) definió la directiva NIS, que entró en vigor en 2016. Su objetivo es aumentar y estandarizar el nivel de ciberseguridad en todos los Estados miembros. Las mismas instituciones publican a menudo recomendaciones (por ejemplo, [TR02021]), buenas prácticas o, al menos, referencias a otras publicaciones.

Esto es importante porque las normativas pueden ser muy poco específicas sobre qué tecnología real utilizar en la práctica. Debido a que la tecnología cambia rápidamente con el tiempo, se necesitaría una adaptación continua de las leyes definidas. Sin embargo, existe un entendimiento común sobre la tecnología punta que publican, por ejemplo, instituciones como TeleTrust [TELETRUST] o el NIST y que se va adaptando con el tiempo.

Las normativas suelen ser poco específicas porque pretenden abarcar un amplio alcance de sectores industriales. Ceñirse a un conjunto fijo de tecnologías de seguridad informática podría causar problemas, ya que algunas tecnologías podrían resultar inaplicables para determinadas organizaciones.

Aunque el uso de la tecnología actual es una parte de la normativa, también es necesario tener en cuenta tres elementos fundamentales:

- Recursos (por ejemplo, hardware, software y tecnología punta).
- Personal.
- Procesos de seguridad de la información.

En cuanto al personal, hay que tener en cuenta cuatro aspectos principales:

- El personal debe ser consciente de la importancia de la seguridad de la información y debe entender que es responsable de asegurar la seguridad.
- Deben tener los conocimientos necesarios para implementar y aplicar las medidas de seguridad definidas, teniendo en cuenta que el tipo y la especificidad pueden variar según los distintos roles. Se necesitan habilidades y conocimientos sobre las políticas y procedimientos de seguridad definidos.
- Deben aceptar y aplicar los procesos de seguridad de la información definidos (véase la sección siguiente).
- Parte del personal requiere habilitaciones especiales

Los procesos de seguridad de la información incluyen aspectos como:

- La definición de responsabilidades. En el contexto de las normativas, esto afecta a los roles y responsabilidades dentro de una organización, y también a las instituciones que son responsables de monitorizar la conformidad y de informar sobre incidentes, como la Respuesta Nacional a Ciber Emergencias. A menudo se designa a una persona como único punto de contacto dentro de una organización. Tienen que definir cuándo y a quién necesitan informar.
- Cómo tratar a los empleados nuevos o que se van, al personal que cambia de departamento o a los empleados externos.

- En caso de alerta de seguridad, quién necesita ser informado, quién será responsable de tomar decisiones y cuándo tiene que informar la organización del incidente a una institución (de ámbito estatal).
- Cómo tratar con socios comerciales y proveedores.
- Revisiones y reevaluaciones periódicas de los procesos y medidas actualmente definidos.
- Procedimientos de auditoría, incluidos los preparativos y las correcciones tras la realización de una auditoría.
- Los aspectos descritos anteriormente forman parte integrante de un SGSI operativo. Algunas normativas pretenden establecer y preservar un SGSI esencial (véase el capítulo 8).

Cómo probar políticas de seguridad

Los IPS tienen un alto nivel de responsabilidad a la hora de probar las políticas de seguridad de las organizaciones afectadas por la normativa. Las principales razones para ello son:

1. Desde la perspectiva de una organización, es importante cumplir la normativa. De lo contrario, la organización puede tener que pagar multas o arriesgarse a que su negocio cierre temporalmente.
2. Dado que las incidencias en los sectores industriales regulados pueden tener consecuencias mucho más graves que en otros sectores, la prueba debe hacerse muy a fondo para asegurar un alto nivel de seguridad.

El siguiente paso consiste en evaluar las buenas prácticas y las tecnologías más avanzadas, ya que es posible que las normativas sólo hagan referencia a ellas. Los IPS necesitan validar que las medidas de seguridad dadas para un sistema sujeto a prueba (SSP) siguen siendo suficientes. Por ejemplo, necesitan comprobar si los datos están cifrados y qué algoritmo se utiliza, ya que es posible que esto ya no sea seguro.

A efectos de prueba, los resultados de auditorías realizadas con anterioridad pueden tenerse en cuenta como parte de la base de prueba. Sin embargo, los hallazgos de una auditoría anterior que ahora se han implementado necesitan ser probados mediante confirmación. El ingeniero de pruebas de seguridad (IPS) no puede asumir que la implementación se ha realizado correctamente. Además, como la normativa y los objetivos clave de una política de seguridad incluyen a las personas y los procesos, las actividades de prueba también deben incluir estos aspectos.

La prueba de los aspectos relacionados con el personal puede realizarse aplicando pruebas como fingir un ataque de ingeniería social o intentar saltarse el control de acceso (físico). Este tipo de prueba depende del departamento de la organización que haya solicitado la prueba de seguridad.

La prueba de los aspectos relacionados con el personal puede realizarse aplicando pruebas como fingir un ataque de ingeniería social o intentar saltarse el control de acceso (físico). Este tipo de prueba depende del departamento de la organización que haya solicitado la prueba de seguridad.

La prueba de seguridad de los procesos incluye la copia de seguridad y restauración, así como la respuesta a emergencias y la elaboración de informes. Estas pruebas pueden ser muy elaboradas y costosas, ya que puede haber muchas personas involucradas durante la prueba.

5.3 Análisis de un escenario de ataque

5.3.1 Escenarios de ataque comunes

Las incidencias de seguridad varían mucho en función de las técnicas y herramientas de ataque aplicadas, el tipo de atacante y la motivación para realizar un ataque. Como resultado, es difícil dar una descripción

genérica de un ataque. Sin embargo, ciertos pasos son comunes a casi todos los ataques. Estos pasos pueden definirse como:

1. Recopilar información.
2. Explotar/obtener acceso.
3. Persistir/mantener acceso.
4. Limpiar rastros.

A modo de comparación, el tratamiento de incidentes de seguridad se define según el NIST [NIST03] mediante los siguientes pasos:

1. Preparación.
2. Detección y análisis.
3. Contención, erradicación y recuperación.
4. Actividad postincidente.

Aunque ambas enumeraciones describen una secuencia común de ataque y respuesta, factores como la motivación, los recursos, las habilidades de un atacante y el enfoque utilizado tienen un fuerte impacto tanto en el éxito de un ataque como en las consecuencias para la parte atacada.

5.3.1.1 Clasificación de atacantes y su motivación

La palabra «jáquer» se utiliza en esta sección como sinónimo de atacante. Sin embargo, el término jáquer se refiere generalmente a una persona con una gran habilidad técnica.

Los atacantes pueden dividirse en diferentes tipos, dependiendo de sus habilidades técnicas y de los recursos de que dispongan:

Tipo de atacante		Descripción
ES	**EN**	
niños de guion	**script kiddies**	Se trata de personas con un bajo nivel de conocimientos técnicos, que utilizan las herramientas y guionizados existentes sin entenderlos del todo y que disponen de recursos muy limitados.
estafadores	**scammers**	Estas personas utilizan técnicas sencillas como la suplantación de identidad, pero suelen dirigirse a muchas personas, lo que aumenta sus posibilidades
jáqueres privados	**private hackers**	Estas personas tienen una sólida formación técnica y están interesadas en la seguridad informática o son muy curiosas
hackers profesionales	**professional hackers**	Estas personas tienen una formación técnica muy elevada y son capaces de realizar ataques muy sofisticados para ganarse la vida
gobiernos	**governments**	Estas organizaciones pagan a equipos completos para el espionaje, el pirateo o el sabotaje y disponen de muchos más recursos que las personas que actúan en solitario o los grupos pequeños.

Aunque se trata de una categorización gruesa que depende de la habilidad y los recursos, otro aspecto importante es la motivación de un atacante. Es posible que los niños de guion sólo quieran jugar con algo que acaban de encontrar en línea o impresionar a sus amigos, mientras que los jáqueres profesionales ganan su dinero desarrollando actividades de pirateo informático y, por lo tanto, también quieren tener una buena reputación. A continuación, se indican las categorías de motivación:

- Motivaciones personales (por ejemplo, fama, venganza, celos y curiosidad).
- Motivaciones políticas (por ejemplo, jactivismo ("hacktivism"), guerra y espionaje).
- Motivaciones profesionales (por ejemplo, dinero, reputación y espionaje industrial).

Dependiendo del nivel de motivación, un ataque puede realizarse contra una sola entidad o contra varias. Por ejemplo, el software de secuestro de datos («ransomware») es un software malicioso que suele cifrar los datos y bloquear el uso del sistema infectado hasta que la víctima paga una determinada cantidad de dinero. Dado que infectar más sistemas aumentará las posibilidades de que el atacante gane dinero, el software de secuestro de datos («ransomware») suele escribirse para infectar tantos sistemas como sea posible.

En contraste con esto, los gusanos informáticos como Stuxnet [WIKI02], se desarrollan principalmente para el caso especial de infectar sistemas de control de supervisión y adquisición de datos. Éstos se han utilizado para sabotear los programas nucleares de países enteros.

5.3.1.2 Enfoque común de un atacante

5.3.1.2.1 Recopilar información

La primera fase de un ataque es la recopilación de información, también conocida como reconocimiento. Un atacante busca información sobre el objetivo e intenta encontrar debilidades en las siguientes áreas:

- Infraestructura informática (por ejemplo, una vulnerabilidad de software conocida).
- Infraestructura física y los mecanismos de control de acceso relacionados (por ejemplo, irrumpir en una oficina, que puede tener un sistema de alarma deficiente y permite el acceso a información sensible).
- Empleados, que pueden tener una escasa conciencia de la seguridad.
- Procesos dentro de una organización que pudieran ser explotables.

La recopilación de información puede ser pasiva o activa. La recopilación pasiva de información puede realizarse buscando en la web mediante consultas de búsqueda especializadas, como Google, que pueden revelar una cantidad sorprendentemente grande de información. Esta práctica se conoce como «Google hacking» o «Google dorking» [WIKI01]. Además, las plataformas de medios sociales son una fuente importante de información sobre los empleados, especialmente en lo que respecta a sus números de teléfono, direcciones de correo electrónico y otros datos personales que pueden utilizarse para la ingeniería social. El uso de los datos de una persona permite a los atacantes personalizar sus ataques, como en el caso de la suplantación de identidad dirigida, o el envío de correos personalizados con archivos adjuntos maliciosos. Éstos pueden utilizar, por ejemplo, el nombre, la dirección o la fecha de nacimiento correctos de una persona para crear correos electrónicos con contenidos que suenen muy familiares o íntimos para la víctima, aumentando así la probabilidad de que ésta abra el archivo adjunto malicioso o haga clic en un enlace y visite un sitio web malicioso.

La recopilación activa de información incluye el uso de herramientas y técnicas que interactúan con el objetivo pero aumentan el riesgo de ser reconocido como atacante. La recopilación de información puede realizarse de diferentes maneras:

- Intentar ponerse en contacto con el personal por correo electrónico o por teléfono (suplantación de identidad con la voz).
- Buscar en la basura de la víctima información útil, como direcciones y números de teléfono [SENG22]. Esto se conoce como «buscar entre la basura».
- Escanear puertos
- Huella digital del sistema operativo
- Enumeración (DNS) (véase más adelante)
- Escanear vulnerabilidades
- Colección de información útil de acceso público (por ejemplo, inteligencia de fuentes abiertas [OSINT]).

Técnicas como la enumeración de DNS pueden pasar desapercibidas, mientras que otras técnicas como el escaneo de puertos o el escaneo de vulnerabilidades a menudo pueden identificarse fácilmente analizando los archivos de registro de servidores o cortafuegos. Los sistemas de detección de intrusiones en la red (NIDS), que suelen implementarse como medida de seguridad en la infraestructura de red, analizan el tráfico entrante en busca de patrones sospechosos y emiten una alerta de seguridad si reconocen un posible tráfico malicioso. En particular, los escáneres de vulnerabilidad tienen una huella digital de red fácilmente detectable. Aunque el uso de escáneres puede revelar información útil al atacante muy rápidamente, su uso también aumenta el riesgo de ser detectado.

La duración de la fase de recopilación de información puede variar considerablemente y dependerá de la motivación del atacante. Un niño de guion puede aburrirse tras unos minutos u horas porque sólo quiere jugar con una herramienta que ha encontrado en Internet, mientras que los atacantes con motivaciones políticas pueden recopilar información durante meses antes de lanzar realmente un ataque. Los métodos activos de recopilación de información siempre dejarán rastros, pero a veces sólo se detectarán mediante investigaciones forenses después de que un ataque se haya realizado con éxito.

Aparte de la motivación, el tipo de ataque también influye en la fase de recopilación de información. Un atacante que quiera realizar un ataque de denegación de servicio que sólo afecte a la disponibilidad puede no necesitar inmiscuirse en una red privada y, por tanto, puede ignorar cierta información. Sin embargo, una mayor información siempre aumenta las posibilidades de éxito de un ataque.

La identificación de un sistema vulnerable puede hacerse enumerando todos los servicios disponibles y sus versiones y realizando después una búsqueda en las bases de datos de elementos explotables disponibles públicamente.

Hay que tener en cuenta que los servicios públicos legales pueden automatizar el escaneado constante activo en todo Internet, actualizando sus bases de datos constantemente. Este tipo de servicios permiten buscar dispositivos no seguros que utilicen contraseñas por defecto y servicios vulnerables. Estos servicios pueden ser utilizados tanto por organizaciones como por particulares, lo que facilita a cualquiera la búsqueda de sistemas vulnerables. Permiten buscar direcciones IP, dominios y versiones de servidores web específicos, lo que la convierte en otra herramienta útil para la recopilación de información.

Sin embargo, la recopilación de información también puede hacerse sin conexión, buscando entre la basura, observando, haciéndose pasar por cliente o infiltrándose en una organización como empleado. Esto aumenta enormemente la posibilidad de ser detectado y, por lo tanto, se evita en la mayoría de los casos. Sólo si un atacante tiene una motivación muy alta o cuando el aprendizaje sobre un objetivo puede haber fallado ya, podría tenerse en cuenta como enfoque para la recopilación de información.

Por último, la recopilación de información es una tarea recurrente que se realiza durante un ataque, porque una vez que un atacante ha conseguido acceder a una infraestructura que no está a disposición del público, necesita seguir aprendiendo sobre ella.

5.3.1.2.2 Explotar/obtener acceso

Una vez que los atacantes han obtenido un conocimiento razonable sobre su objetivo, pasarán al ataque propiamente dicho. «Conocimiento razonable» en este contexto significa que realmente han encontrado al menos una posibilidad de ataque que tendrá una alta probabilidad de éxito. Un ataque con éxito puede deberse a:

- Vulnerabilidades de software conocidas en software sin parchear.
- Mala configuración (por ejemplo, falta de configuración o configuración incorrecta).
- Elementos explotables de día cero (fragmento de código).
- Contraseñas débiles.
- Ingeniería social.

Si un ataque tiene éxito, lo normal es que el atacante no disponga de una cuenta de administrador. Esto podría representar un obstáculo para ellos, ya que querrían alterar el sistema para sus fines (por ejemplo, deteniendo o modificando el software antivirus). Por lo tanto, a menudo se requiere la adquisición de niveles de privilegio superiores tras el acceso inicial. La escalada de privilegios puede tener éxito por las mismas razones que el ataque inicial, como una vulnerabilidad del software o una mala configuración. La mala configuración es una seria amenaza para la escalada de privilegios. En los sistemas UNIX, por ejemplo, muchos programas permiten el acceso a un intérprete de comandos raíz y, si permiten el uso de SUDO, (acrónimo de super user do), pueden utilizarlo para ejecutar programas como superusuario u otro usuario [GTFO22].

Aunque se trata de ataques que pueden ejecutarse de forma remota, también es posible que un atacante obtenga acceso directo desde la red interna de una organización. Un atacante podría ser un empleado que quiere perjudicar a la organización. Además, un atacante podría obtener acceso físico a una oficina debido a un control de acceso insuficiente y desde allí podría ser capaz de conectarse a la red interna.

La ingeniería social es una amenaza que puede llevar al atacante a obtener acceso directo. Según el NIST, se define como «el acto de engañar a un individuo para que revele información sensible obtenga acceso no autorizado o cometa fraude asociándose con él para ganarse su confianza». [NIST05].

5.3.1.2.3 Persistir/mantener acceso

Para obtener acceso no autorizado a un sistema, a menudo se utilizan elementos explotables que pueden ser difíciles de aplicar, tienen una alta probabilidad de fallo o sólo pueden aplicarse en determinados momentos. Por lo tanto, los atacantes necesitan mantener el acceso al sistema comprometido hasta que hayan logrado sus objetivos. Una vez más, esto depende de la motivación de los atacantes, ya que algunos sólo pueden estar interesados en un ataque exitoso, pero no quieren ir más allá.

El acceso persistente suele lograrse a través de rootkit, creados específicamente para este fin. Los rootkits intentan mantener el acceso, incluso si se reinicia un sistema. Dado que el software contra software malicioso intenta detectarlo, los rootkits se construyen para ocultarse a sí mismos y a software malicioso adicional, como registradores de claves y husmeadores de red en el sistema. También pueden habilitar el control remoto automatizado de sistemas comprometidos, permitiendo a los atacantes crear redes de bots para realizar ataques distribuidos de denegación de servicio contra otros sistemas, o utilizarlos indebidamente como servidor de spam.

5.3.1.2.4 Limpiar rastros

Dado que los ataques pueden tener graves consecuencias legales, los atacantes desean permanecer en el anonimato o, al menos, no quieren ser identificados en persona. Como resultado, tienen una gran motivación para eliminar todos los rastros de sus actividades precedentes después de haber logrado su objetivo. Esto incluye eliminar todos los programas y archivos que el atacante ha copiado en el sistema o

sistemas comprometidos, borrar o eliminar los archivos de registro, los historiales de comandos y quizás destruir el hardware que utilizaron para el ataque.

Aunque se trata de tareas que un atacante realiza al completar un ataque, los atacantes también utilizarán técnicas como el encadenamiento de intermediarios («proxy chaining»), una red privada virtual (RPN) o servidores de salto ya comprometidos durante su acceso remoto para ocultar sus rastros.

Un atacante debe ser consciente de que cada actividad que realice contra un sistema puede quedar potencialmente registrada y, en la mayoría de los casos, será incapaz de borrar por completo todos sus rastros.

En el contexto de las pruebas de seguridad y penetración, es necesario actuar de forma similar, ya que el sistema sujeto a prueba puede ser un sistema de detección de intrusión (SDI) [en inglés, intrusion detection system (IDS)] o un proceso de respuesta a emergencias.

5.3.1.3 Respuesta a incidentes y análisis posterior al incidente

En las secciones anteriores se ha descrito un escenario de ataque desde la perspectiva de un atacante. Por otro lado, las organizaciones invierten en medidas de seguridad para detectar y resolver incidentes de seguridad, tarea que se conoce como respuesta a incidentes. Hay que tener en cuenta que la respuesta a incidentes y las fases de ataque descritas anteriormente suelen tener lugar al mismo tiempo. En muchos casos, un atacante ya ha sido detectado en una fase anterior (por ejemplo, la fase de persistencia), y no justo después de que haya intentado borrar sus rastros o huellas.

5.3.1.3.1 Preparación

Aunque la preparación forma parte de la gestión de incidencias, no forma parte de la respuesta a incidencias, sino que construye los cimientos de un procedimiento operativo de respuesta a incidencias que tendrá lugar en caso de que se produzca un incidente de seguridad.

La respuesta a incidentes tiene como objetivo lo siguiente:

- Identificar una incidencia y analizar la situación.
- Contención, por ejemplo, aislar los sistemas comprometidos y cerrar los servicios
- Erradicación, por ejemplo, eliminar las cuentas de usuario comprometidas, eliminar el software malicioso y aplicar un parche a un sistema
- Recuperación, por ejemplo, volver a poner en línea los servicios y restaurar los datos

Una vez resuelto un incidente, se debería realizar una revisión para identificar las debilidades de la infraestructura y de los procesos de respuesta a incidentes.

5.3.1.3.2 Detección y análisis

La detección de una incidencia puede ser intencionada o no por parte del atacante. En el caso de la desfiguración de un sitio web, por ejemplo, la intención de los atacantes es que se reconozca el ataque. En otros casos, pueden dejar un mensaje a un administrador del sistema.

Se pueden utilizar diferentes herramientas para ayudar a detectar actividades sospechosas. Entre ellas se encuentran los sistemas de detección de intrusiones en la red o en el dispositivo anfitrión (NIDS/HIDS), los escáneres de software malicioso y los analizadores de registros. En el mejor de los casos, se puede identificar un ataque antes de que el atacante haya tenido éxito. Este puede ser el caso si un atacante es demasiado llamativo durante el escaneado, o si fallan sus primeros intentos con un elemento explotable (fragmento de código). Asimismo, muchos intentos fallidos de inicio de sesión, o intentos de registro de usuarios que no existen o que no suelen iniciar sesión de forma remota pueden ser indicadores de un posible ataque. Si esto puede detectarse fuera de la red de la organización, hay muchas posibilidades de derrotar el ataque.

Si se detectan actividades sospechosas dentro de la red de la organización, hay que analizar qué sistemas están ya comprometidos y cómo consiguió el atacante el acceso. Esto puede hacerlo el departamento de TI, pero lo más habitual es que un equipo forense comience a analizar la incidencia. La respuesta a una incidencia es crítica para el tiempo y las acciones deben llevarse a cabo lo antes posible para evitar daños mayores.

5.3.1.3.3 Contención, erradicación y recuperación

Si se tiene una idea clara de qué sistemas están comprometidos, el siguiente paso es contener estos sistemas para evitar que un atacante comprometa más sistemas. Esto puede lograrse, por ejemplo, cerrando los servicios temporalmente, trasladándolos a otra red o bloqueando las cuentas de los usuarios.

Un aspecto importante a tener en cuenta durante esta fase es que algunas acciones podrían borrar pruebas forenses, que podrían ser útiles para el análisis posterior a la incidencia. Por ejemplo, apagar un sistema borrará la memoria principal, que podría contener datos útiles como el elemento explotable (fragmento de código) inicial que se utilizó. De nuevo, es vital reaccionar con rapidez, ya que un atacante podría comprometer otros sistemas. Las acciones que se lleven a cabo deben decidirse en función del nivel de riesgo.

La contención y el análisis de la situación se alternarán en un momento determinado, ya que habrá que volver a evaluar si se han identificado correctamente todos los sistemas y el atacante no es capaz de obtener más accesos. Si se tiene la certeza suficiente de que todos los sistemas afectados han sido identificados y contenidos, se puede proceder a la erradicación. Un aspecto importante durante esta fase es proporcionar pruebas para su posterior análisis. La erradicación puede incluir la eliminación de todo un sistema y su posterior recreación a partir de una copia de seguridad. También es posible eliminar sólo componentes parciales del sistema y sustituirlos durante la fase de recuperación por un nuevo componente. En ambos casos, hay que asegurarse de que la copia de seguridad o el componente utilizados no contengan rastros de la incidencia resuelta anteriormente.

5.3.1.3.4 Actividad postincidente

Después de resolver una incidencia, deben darse varios pasos para evaluar y mejorar las rutinas de seguridad actuales. Esto incluye:

- Investigaciones forenses, que en el mejor de los casos pueden identificar al atacante.
- El cierre de las vulnerabilidades que pudieran haberse revelado a través del ataque.
- Reevaluar la infraestructura actual.
- Aumentar la concienciación de los empleados en materia de seguridad.
- Reevaluar y quizás adaptar las políticas de seguridad.
- Perfeccionando los procesos de respuesta a incidentes.
- Haciendo anuncios a clientes o consumidores, especialmente si la organización tiene obligaciones de suministro de información. Esto incluye informar a la institución correspondiente o a la administración pública.

6 Adaptación de la prueba de seguridad a los modelos de ciclo de vida del desarrollo del software - (K4)

Duración: 165 minutos

Palabras Clave[6]

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB**: identificarán palabras clave del proceso de prueba
- **ESPDOM**: identificarán palabras clave específicas de dominio: **SEGURIDAD**

Tipo Palabra Clave	Español	Inglés
ISTQB	ciclo de vida del desarrollo de software	software development lifecycle

[6] Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.

Objetivos de Aprendizaje para "Capítulo 6"

6.1 Efectos de los diferentes modelos de ciclo de vida del desarrollo de software en la prueba de seguridad

STE - 6.1.1 (K2) Resumir por qué las actividades de prueba de seguridad deben cubrir el ciclo de vida de desarrollo del software.

STE - 6.1.2 (K4) Analizar cómo se ven afectadas las actividades de prueba de seguridad por los diferentes modelos de ciclo de vida de desarrollo de software.

6.2 Prueba de seguridad durante el mantenimiento

STE - 6.2.1 (K3) Definir y realizar pruebas de regresión de seguridad y pruebas de confirmación basadas en un cambio en un sistema.

STE - 6.2.2 (K2) Analizar los resultados de las pruebas de seguridad para determinar la naturaleza de una vulnerabilidad y su impacto técnico potencial.

6.1 Efectos de los diferentes modelos de ciclo de vida del desarrollo de software en la prueba de seguridad

El ciclo de vida de la aplicación o del sistema puede describirse como un modelo con diferentes CVDS. Las fases de CVDS más utilizadas son la planificación, el análisis, el diseño, el desarrollo, la prueba, la implementación, el mantenimiento y la terminación.

Las actividades y tareas planificadas para cada una de estas fases pueden diferir en función de la aplicación, el sistema, el proyecto o la organización. Éstas se definen en el modelo de ciclo de vida de desarrollo de software (CVDS), que puede implementarse utilizando un enfoque de desarrollo secuencial o de desarrollo ágil.

Con un enfoque de desarrollo secuencial es más fácil reconocer las distintas fases del CVDS [ciclo de vida de desarrollo de software (SDLC)]. Con el enfoque ágil puede que no esté tan claro cuándo y qué actividad o tarea de qué proceso del ciclo de vida se realiza. Las actividades y tareas pueden repetirse con frecuencia, añadiendo valor a la aplicación o sistema con cada iteración.

En el programa de estudio de nivel básico del ISTQB Certified Tester [ISTQB FL] se mencionan varios modelos de desarrollo, incluido el modelo de cascada y los modelos de desarrollo ágil de software (por ejemplo, Rational Unified Process, Scrum, Kanban y Spiral). Se menciona que la prueba de seguridad debería adaptarse a estos modelos para ser más eficaz. Como es de esperar, los enfoques de la prueba de seguridad también necesitan adaptarse a los diferentes modelos de CVDS [ciclo de vida de desarrollo de software (CVDS)]. En este programa de estudio se habla de DevOps además de los modelos descritos anteriormente.

El ingeniero de prueba de seguridad (IPS) debería conocer las características más importantes de estos modelos de CVDS y cómo pueden afectar a su capacidad para realizar la prueba de seguridad.

Al comparar estas categorías, se pueden observar diferencias en cuanto a los siguientes atributos. Un cambio en cualquiera de estos atributos también tendrá un impacto en cómo se realiza la prueba de seguridad.

Atributo	Descripción
Duración del desarrollo	Tiempo necesario desde la formulación de un requisito hasta su despliegue • La duración también afectará al tiempo disponible para ejecutar la prueba de seguridad durante el ciclo de vida de desarrollo de software (CVDS). • Cuanto menos tiempo haya disponible, más difícil será tomar decisiones y establecer prioridades.
Tamaño del despliegue	Mayores lotes de funcionalidad o una única prestación por despliegue • A menudo en relación directa con la duración del desarrollo. • Cuanto menor sea el tamaño de un despliegue, más específico puede (y debe) ser el enfoque de la prueba de seguridad. • Con despliegues más grandes, la superficie de ataque puede aumentar considerablemente en una iteración. La necesidad de pruebas de regresión aumenta en los modelos de desarrollo incremental.
Tiempo disponible para la prueba	Cantidad de recursos de tiempo reservados para realizar la prueba • En la mayoría de los casos, la prueba funcional puede planificarse, pero la prueba no funcional (incluida la prueba de seguridad) no suele planificarse en los ciclos de vida del proyecto. • Si el tiempo lo permite, puede crear oportunidades para realizar pruebas no funcionales más extensas, incluidas las pruebas de seguridad.

Atributo	Descripción
Independencia del equipo	Nivel de decisiones en materia de seguridad que puede tomar el equipo • Si fuera necesario, puede permitir que el equipo asigne o contrate una competencia autónoma para la prueba de seguridad.
Transversalidad del equipo	Disponibilidad de las habilidades necesarias (para el desarrollo) en el equipo • El equipo puede disponer de competencias en prueba de seguridad diferentes y complementarias.
Nivel de automatización	Proporción del proceso de desarrollo que se encuentra automatizada • Gran parte de las pruebas de seguridad pueden realizarse con la automatización de la prueba mediante actividades relacionadas con la prueba estática de seguridad de aplicaciones ((PESA) y la prueba dinámica de seguridad de aplicaciones (PDSA).
Principios de gestión	¿La organización es de línea, de proyecto u orientada al producto? Los principios de gestión del equipo pueden influir en la organización de la prueba de seguridad. Pueden ser realizadas con equipos facilitadores, concentrándose continuamente en la seguridad, sólo durante la fase de proyecto o sólo durante el mantenimiento.
Entorno de prueba	Se puede establecer un entorno de prueba separado y dedicado a la prueba de seguridad destructiva.

6.1.1 Modelos de desarrollo secuencial

A menudo, el ciclo de vida de desarrollo de software (CVDS) completo está especificado por todos los procesos necesarios para desarrollar, mantener y dar soporte al sistema desde su inicio hasta su retirada/eliminación. Un estándar muy utilizado que describe todos estos procesos y sus relaciones es [ISO 15288].

El modelo de desarrollo de sistemas describe la implementación de (partes de) el ciclo de vida de desarrollo de software (CVDS) [software development lifecycle (SDLC)] necesario para desarrollar e implementar un sistema. Esta implementación no incluye necesariamente todos los procesos del SDLC y debe ajustarse a la organización (véase el capítulo 5) y al contexto del proyecto.

La seguridad de la información y la verificación de la seguridad deben ser un aspecto integrado cubierto en todos los procesos del ciclo de vida de desarrollo de software (CVDS) [software development lifecycle (SDLC)] utilizados en la organización. Sólo entonces podrá lograrse un enfoque verdaderamente holístico. Esto también apoyará o garantizará que las actividades requeridas durante los procesos de desarrollo puedan llevarse a cabo de forma consistente.

En [NIST 800-160] se describe la seguridad del sistema como un problema de diseño. Señala que «se requiere una combinación de salvaguardas de hardware, software, comunicaciones, físicas, de personal y administrativo-procedimentales para una seguridad integral. Las salvaguardas de software por sí solas no son suficientes».

Este estándar del NIST presenta consideraciones y enfoques que abarcan todos los procesos del ciclo de vida de desarrollo de software (CVDS) sobre cómo abordar las actividades de seguridad de la información.

El modelo de desarrollo secuencial en cascada o su implementación como modelo V sigue siendo un modelo muy utilizado. El modelo V se refiere a actividades de prueba genéricas para cada una de sus fases con el objetivo del desplazamiento a la izquierda. Estos modelos están más establecidos en organizaciones con una separación clara entre los distintos equipos y fases de desarrollo. En general, cabe

esperar que el ingeniero de prueba de seguridad (IPS) disponga de tiempo para planificar, preparar y realizar la prueba de seguridad cuando utilice este modelo de desarrollo de software. Las fases del modelo se establecen en secuencia, pero pueden solaparse entre sí.

En estos modelos, el ingeniero de pruebas de seguridad (IPS) debería tener en cuenta lo siguiente:

- Los requisitos y riesgos de seguridad se definen al principio del proyecto y deben documentarse en las especificaciones de los requisitos software.
- Los requisitos de seguridad pueden cambiar en el transcurso del proyecto a medida que se descubren nuevas amenazas, pero es posible que no se reflejen en los requisitos actualizados del software. Por lo tanto, la prueba de seguridad puede parecer muy específica y completa, pero, en realidad, puede no estar completa o actualizada debido a los riesgos de un proyecto tardío.
- La prueba de seguridad puede realizarse en cualquier momento o en cualquier fase de desarrollo, pero es habitual que se realice tarde en el proyecto.
- Puede resultar difícil abordar los resultados de la prueba de seguridad y las correcciones al final de un proyecto de modelo de desarrollo secuencial, ya que los plazos se fijarán en la mayoría de los casos en una fase temprana del proyecto.

6.1.2 Desarrollo Ágil de Software

El Desarrollo Ágil de Software promueve la compleción del trabajo a realizar por equipos autoorganizados y transversales en iteraciones cortas. Los equipos facilitadores que prestan servicios específicos al proyecto pueden estar a disposición de estos equipos de Desarrollo Ágil de software para ayudar con competencias de dominios específicos, como la prueba de seguridad.

Lo genérico del desarrollo ágil de software es que los incrementos (de sistema y software) se entregan en una serie de iteraciones. Cada una de estas iteraciones puede durar desde días hasta algunas semanas. Los modelos de desarrollo ágil de software suelen utilizarse principalmente en la fase de desarrollo de aplicaciones/sistemas, aunque algunos modelos, como Kanban, también pueden aplicarse durante la fase de operación.

El marco de trabajo Scrum, en diversas implementaciones, es el más utilizado en el desarrollo ágil de software. Todo el análisis, el diseño, la codificación y las pruebas se realizan durante cada iteración, incluida la prueba de seguridad.

Las listas de trabajo acumulado de producto actúan, al menos en parte, como una especificación de requisitos. Se espera que tanto los requisitos de seguridad como otros requisitos no funcionales formen parte de la lista de trabajo acumulado del producto. Las épicas se dividen en varias historias de usuario y tareas que son seleccionadas por el equipo o equipos para ser desarrolladas o entregadas en uno de los esprints.

La funcionalidad desarrollada puede modificarse o incluso eliminarse en futuros esprint. La funcionalidad «creciente» y los futuros cambios o incluso eliminaciones hacen que la base de prueba con la que trabajar sea inestable. El Desarrollo Ágil de Software puede verse como una mezcla de desarrollo (nueva funcionalidad) y mantenimiento (plataforma y funcionalidad existente) durante la fase de proyecto. Por lo tanto, es esencial repetir la prueba de seguridad automatizada.

Se pueden adoptar varios enfoques para realizar las actividades de la prueba de seguridad. Algunos ejemplos son:

- Realizar una parte de la prueba de seguridad y, a continuación, concentrarse en los objetos de prueba funcionales, técnicos o relacionados con la plataforma en cada esprint diferente.
- Realizar una parte de la prueba de seguridad en la mayoría de los esprints y llevar a cabo una prueba de seguridad completa en un esprint específico.
- Realice todas las pruebas de seguridad en un solo esprint (tardío) que se asemeje al modelo de desarrollo secuencial.

El equipo de desarrollo ágil puede involucrar a un equipo facilitador o contratar recursos para realizar las pruebas de seguridad, ya que estas competencias no suelen residir en el equipo.

A medida que la solución cambia con cada esprint debe realizarse una prueba de regresión. La seguridad no se introduce a través de una prueba o un parche en una aplicación ya construida. Más bien, se consigue mediante un diseño orientado a la seguridad (es decir, seguridad por diseño) y la verificación a lo largo de todo el proceso de construcción.

[Synopsys] ha descrito cómo se puede aplicar la prueba de seguridad en el desarrollo ágil de software aplicando los cuatro principios siguientes definidos en el manifiesto ágil:

- Desarrolladores y probadores por encima de especialistas en seguridad.
- Asegurar mientras se trabaja por encima de asegurar después de terminar.
- Implementar las prestaciones de forma segura por encima de añadir prestaciones de seguridad.
- Mitigar los riesgos por encima de arreglar los defectos.

6.1.2.1 Desarrolladores y probadores por encima de especialistas en seguridad

Los especialistas en seguridad experimentados son recursos valiosos. Los equipos ágiles rara vez pueden permitirse el lujo de contar con sus propios especialistas en seguridad. Esto significa que, la mayoría de las veces, los equipos ágiles deben responsabilizarse de su propia seguridad y no pueden esperar a una revisión de seguridad externa antes de que el código pase a la siguiente fase de desarrollo. La seguridad debe integrarse en el desarrollo y las pruebas del código. Los equipos deben apropiarse de la seguridad del mismo modo que se apropian de la experiencia del usuario, la fiabilidad, la eficiencia de desempeño y otros requisitos no funcionales.

6.1.2.2 Asegurar mientras se trabaja por encima de asegurar después de terminar

La aplicación de metodologías y prácticas seguras a la hora de crear, entregar y mantener software funcional es imprescindible. Al mismo tiempo, las actividades de seguridad no deben obligar a los desarrolladores a dejar lo que están haciendo, ir a otra herramienta para remediarlo y luego volver a lo que estaban haciendo. La alternativa es integrar la retroalimentación y la información sobre seguridad en las herramientas del desarrollador. Las tareas de seguridad se presentan (por ejemplo, en pizarras) y se establecen prioridades para que sean visibles junto a otras tareas.

6.1.2.3 Implementar las prestaciones de forma segura por encima de añadir prestaciones de seguridad

Hay que concentrarse en cumplir la misión de negocio del software, pero siempre hay que tener en cuenta la integración de la seguridad. Esto significa que arquitectos, desarrolladores, probadores y otros implicados deben tener en cuenta los aspectos de seguridad y trabajar juntos para definir y construir sistemas más seguros. Los sistemas seguros deben diseñarse y construirse desde el principio.

6.1.2.4 Mitigar los riesgos por encima de arreglar los defectos

Hay que tener en cuenta los riesgos específicos del negocio, los usuarios, los datos y el software. La gestión del riesgo tiene en cuenta la forma correcta de afrontar un riesgo. Esto puede lograrse adoptando una visión de alto nivel de lo que podría salir mal en lugar de reducir la seguridad a una larga lista de defectos individuales que necesitan ser resueltos. Aunque el modelado de amenazas es más difícil que la simple localización y corrección de defectos, es un enfoque eficaz para detectar problemas en una fase

temprana del ciclo de vida de desarrollo de software (CVDS) [software development lifecycle (SDLC)]. Es definitivamente más barato cuando los problemas pueden resolverse antes de entregar el software.

6.1.3 La metodología DevOps

La mayor parte del desarrollo ágil de software abarca la entrega del sistema o software al departamento de operaciones. DevOps va más allá al incluir el desarrollo y las operaciones. El objetivo principal con DevOps es entregar (pequeños) cambios rápidamente. Además, la cultura de equipo tiene un impacto mucho mayor en el éxito del equipo. Los equipos DevOps suelen ser más autónomos y estar más orientados al producto que otros equipos.

DevSecOps abarca todo el ciclo de vida de desarrollo de software (SDLC), incluido el desarrollo, la seguridad y las operaciones. Durante el desarrollo, la seguridad se concentra en identificar y prevenir las vulnerabilidades, mientras que en las operaciones, monitorizar y defenderse de los ataques son los objetivos principales.

En general, las iteraciones de DevOps pueden ser tan cortas como una hora y las entregas suelen ser de una sola prestación/tarea de desarrollo o pequeñas ramas. El objetivo del equipo de DevOps es que los resultados de las pruebas estén disponibles casi inmediatamente después de realizar un cambio en cada paso del proceso de desarrollo. Esto ejerce una gran presión sobre la prueba y las metodologías en uso, lo que a su vez tiene un efecto importante sobre las posibilidades de realizar la prueba de seguridad.

Para lograr iteraciones de desarrollo DevOps cortas se automatizan muchas de las tareas repetitivas y que consumen muchos recursos. Esto se hace en forma de una canalización compuesta por una serie de fases de canalización, cada una de las cuales puede contener uno o más trabajos relacionados con la realización de tareas específicas en el proceso de construcción y despliegue. Una fase de canalización puede denominarse prueba de sistema y puede tener un trabajo que ejecute pruebas de regresión automatizadas. A continuación, esta canalización se ejecuta para cada prestación que deba entregarse.

DevOps viene en diferentes sabores. Los dos enfoques más utilizados son:

- Mediante el uso de una rama principal o maestra. Los cambios se desarrollan y prueban en una rama de prestaciones de corta duración o troncal y se despliegan directamente en producción tras su aprobación. Mediante una rama de desarrollo separada, los cambios se entregan continuamente en un entorno de prueba y se despliegan juntos en una rama principal.
- Mediante el uso de una rama de desarrollo independiente, los cambios se entregan en un entorno de prueba de forma continua y se despliegan juntos en un pequeño lote tras un breve periodo. Esto también se conoce como desarrollo basado en prestaciones.

Las actividades de prueba de seguridad requieren tiempo. Una pregunta común a la que hay que responder implica decidir si se realizan pruebas de seguridad para cada canalización o si se establecen calendarios durante la noche después de ejecutar algunas canalizaciones.

DevSecOps es un concepto que indica la importancia que se da a la prueba de seguridad en DevOps. Se suele tener en cuenta como varios trabajos de prueba de seguridad ejecutados automáticamente en la canalización e incluye tanto el análisis estático (es decir, el desplazamiento a la izquierda) como la concentración en la monitorización y la prevención relacionadas con la seguridad, como la formación en seguridad y la escritura de código seguro.

Se hace énfasis en la seguridad como responsabilidad del equipo, teniendo en cuenta la seguridad como parte de todas las actividades de desarrollo durante todas las fases para todos los miembros del equipo.

Los retos comunes en la implementación de la prueba de seguridad utilizando la metodología DevOps son:

- La prueba de seguridad se sigue teniendo en cuenta como una tarea especializada que deben realizar recursos específicos. Esto inhibe la tan necesaria integración de la prueba de seguridad dentro del equipo DevOps.

- Es posible que se dé demasiada prioridad a la seguridad, lo que hace que se descuiden otras características de calidad, como la eficiencia de desempeño y la usabilidad. La seguridad es importante, pero necesita una implementación equilibrada.
- La seguridad puede llevar a inhibir la creatividad de los desarrolladores, la autonomía del equipo y la posibilidad de experimentar. Estos atributos se tienen en cuenta como esenciales para el éxito de la implementación de DevOps.

El proyecto Phoenix [TechTarget] describe las siguientes prácticas necesarias para una implementación satisfactoria de DevOps. Éstas también deberían aplicarse a la prueba de seguridad:

- Crear y mantener un flujo de tareas
    - La prueba de seguridad suele tenerse en cuenta como una tarea grande o única (por ejemplo, la prueba de la aplicación, el escaneado de la red y la revisión de la arquitectura). Al aplicar DevOps, es necesario planificar, preparar y realizar la prueba de seguridad en tareas más pequeñas. Éstas deben avanzar (fluir) del mismo modo que otras tareas de desarrollo aplicando conceptos como hacer visibles las tareas, limitar el trabajo en curso, asignar tareas individuales a personas individuales y automatizar cuando sea posible.
- Asegurar una retroalimentación instantánea
    - Los resultados de las pruebas deben estar disponibles lo antes posible. Deben ser comprensibles y resolubles. La capacidad de hacer esto se apoya en el flujo antes mencionado y en el uso de tareas más pequeñas. Esto también ayuda a reducir la deuda técnica.
- Fomentar una cultura de seguridad DevOps
    - El equipo debe ser abierto y transparente en lo que respecta a las dificultades de seguridad. Deben estar motivados para informar de las dificultades relacionadas con la seguridad y tener en cuenta la seguridad como una responsabilidad del equipo.

Conceptualmente, DevOps permite fallar, aprender y mejorar. Al introducir DevOps, se tiene en cuenta que es mejor empezar con algunas pequeñas mejoras y poner en marcha el flujo.

Para ser eficiente, el ingeniero de prueba de seguridad debe integrar todas las tareas necesarias relacionadas con la seguridad en la canalización DevOps (CI/CD) descrita anteriormente. Esto significa no sólo concentrarse en la prueba de seguridad, sino también formular y mejorar las épicas, las historias de usuario y las tareas durante la fase de planificación.

6.2 Prueba de seguridad durante el mantenimiento

6.2.1 Prueba de regresión y prueba de confirmación de seguridad

La prueba de seguridad continúa después de la puesta en producción de un sistema. Pueden producirse cambios en el entorno técnico, en los sistemas externos y en las integraciones del sistema sujeto a prueba (SSP) [en inglés, system under test (SUT)]. Los cambios pueden deberse a actualizaciones periódicas de seguridad o a otros cambios en el middleware, el firmware y el hardware. Además, el sistema sujeto a prueba (SSP) estará sujeto a cambios planificados y no planificados que podrían abrir nuevas vulnerabilidades y posibles ataques.

Todos estos cambios requieren al menos la realización periódica de pruebas de regresión de seguridad. En función de la magnitud de los cambios, es posible que se necesite una nueva prueba de seguridad.

Las pruebas de seguridad pueden consistir en comprobar que el sistema sigue resistiendo con éxito los intentos de anular los controles de seguridad establecidos. Las mejoras de la usabilidad o de la eficiencia de desempeño son especialmente propensas a afectar negativamente a los controles de seguridad.

La prueba de regresión de la seguridad debe concentrarse en confirmar que se satisfacen todos los requisitos de seguridad y en probar las nuevas vulnerabilidades que puedan haberse introducido durante las actividades de mantenimiento.

La prueba de regresión suele aplicarse con una colección de casos de prueba que se basan en probar funciones individuales. Sin embargo, para la prueba de seguridad, a menudo es insuficiente para detectar defectos de regresión con un impacto en la seguridad. Los escenarios de prueba de regresión extremo a extremo son más robustos y proporcionan un mayor nivel de confianza en que se pueden realizar transacciones completas de forma segura. Para este tipo de prueba de regresión, debe definirse un conjunto de condiciones de prueba de seguridad y probarse cada vez que se realice un cambio en el sistema. Los defectos de regresión pueden aparecer por cambios en todos los parámetros relevantes del sistema. Algunos de los defectos de regresión pueden tener repercusiones en la seguridad.

Después de que un sistema se haya puesto en producción, puede ser necesario un esfuerzo de desarrollo adicional para corregir defectos en la versión entregada (es decir, mantenimiento correctivo), para ajustarse a otros cambios en el entorno operativo (es decir, mantenimiento adaptativo) o para ampliar o mejorar prestaciones (es decir, mantenimiento perfectivo).

La perspectiva de las pruebas de seguridad para el mantenimiento del sistema se concentra en probar los cambios realizados para corregir los defectos y la funcionalidad básica. El propósito de esto es:

- asegurar que no se han introducido nuevas vulnerabilidades en el sistema sujeto a prueba (SSP) [en inglés, system under test (SUT)].
- verificar que las defensas de seguridad existentes siguen siendo efectivas tras un cambio.

Parte del proceso de mantenimiento consiste en mantener actualizados los cortafuegos y otras tecnologías de seguridad. La monitorización continua del sistema puede detectar actividades sospechosas que pueden necesitar una intervención inmediata.

7 Prueba de seguridad como parte de un sistema de gestión de la seguridad de la información - (K3)

Duración: 105 minutos

Palabras Clave[7]

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB**: identificarán palabras clave del proceso de prueba
- **ESPDOM**: identificarán palabras clave específicas de dominio: **SEGURIDAD**

Tipo Palabra Clave	Español	Inglés
ESPDOM	Sistema de gestión de la seguridad de la información (SGSI)	Information security management system (ISMS)

[7] Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.

Objetivos de aprendizaje para "Capítulo 7"

7.1 Criterios de aceptación para la prueba de seguridad

STE - 7.1.1 (K2) Comprender los criterios de aceptación de la prueba de seguridad y cómo influyen en la selección de enfoques de prueba de seguridad y técnicas de prueba.

7.2 Entradas para un sistema de gestión de la seguridad de la información

STE - 7.2.1 (K2) Comprender el rol de la prueba de seguridad para la efectividad de un sistema de gestión de la seguridad de la información.

7.3 Mejora de un sistema de gestión de la seguridad de la información mediante el ajuste de la prueba de seguridad

STE - 7.3.1 (K3) Evaluar la madurez de un sistema de gestión de la seguridad de la información mediante la incorporación de diferentes enfoques de prueba, nuevos objetos de prueba o una cobertura mejorada.

STE - 7.3.2 (K2) Comprender la capacidad de ser medido dentro de un sistema de gestión de la seguridad de la información.

7.1 Criterios de aceptación para la prueba de seguridad

La prueba de seguridad puede aplicarse como una actividad puntual ad hoc para un sistema antes de entrar en producción o como un proceso continuo y sistemático en desarrollo. Ambos tipos generarán resultados de prueba, pero su capacidad para proporcionar evidencias de riesgos de seguridad significativos varía mucho. Lo mismo ocurre con las distintas técnicas de prueba de seguridad. Por ejemplo, la prueba de caja blanca de seguridad generará resultados de prueba y podría identificar otras vulnerabilidades que las generadas mediante el uso de la prueba de caja negra de seguridad.

Al igual que ocurre con la ingeniería de software en general, los requisitos para la prueba de seguridad rara vez están claramente definidos, son completos, precisos o consistentes. Comenzar una prueba de seguridad basada en esos requisitos mal definidos podría generar algunos resultados de prueba, pero su valor depende de la calidad de los requisitos, que no está predefinida. Cualquier acción basada en tales requisitos será, probablemente, arriesgada por las siguientes razones:

- Técnicas de prueba cuestionables:
    - La prueba de seguridad utiliza una técnica de prueba específica o una combinación de diferentes técnicas, cada una de las cuales tiene puntos fuertes y débiles. La mejor técnica de prueba no existe de por sí, pero existen algunas preferencias para alcanzar un objetivo de prueba determinado. Sin objetivos de prueba, todas las técnicas de prueba pueden responder a las expectativas, pero sin ninguna garantía de crear valor añadido.
- Cobertura cuestionable:
    - La mayoría de las técnicas de prueba no tienen una definición predefinida de compleción, sino que necesitan métricas bien definidas que tienen que cumplirse para que la prueba pueda considerarse hecha. El tipo de métrica y sus umbrales dependen de los objetivos de la prueba de seguridad. Sin ellos, el ingeniero de pruebas de seguridad (IPS) puede alcanzar un nivel de cobertura que podría no cumplir los objetivos de la prueba.

Para evitar estos escollos, es esencial definir los criterios de aceptación antes de cualquier prueba de seguridad. Estos deben cumplirse antes de utilizar los resultados de prueba como base para identificar cualquier desviación o elemento de acción.

La palabra aceptación es clave. ([WaCh90]) afirma que esto significa «que los productos software provisionales y finales se examinan para determinar si cumplen unos criterios específicos. Si lo hacen, entonces han pasado la aceptación». Naturalmente, los requisitos de seguridad deben tener sus propios criterios de aceptación (cf. ([WaCh90]), que respaldan la decisión de aceptación de rechazar, aceptar parcialmente o aceptar.

La prueba de seguridad puede ser muy adecuada para controlar los criterios de aceptación de seguridad definidos para un sistema sujeto a prueba (SSP) [en inglés, system under test (SUT)]. Los resultados de las pruebas, que suelen agregarse en un informe de prueba, deben contener toda la información necesaria para posibilitar una decisión de aceptación. Para respaldar esta decisión, el enfoque de prueba de seguridad seleccionado debe basarse en los criterios de aceptación específicos. Normalmente esto se hace en los siguientes pasos:

- Leer detenidamente los criterios de aceptación de seguridad.
- Hacer una lista de las posibles técnicas de prueba de seguridad (véase el capítulo 2) que pueden utilizarse para apoyar la decisión de aceptación. Se debe tener en cuenta que algunas técnicas de prueba de seguridad pueden abarcar de cero a muchos criterios de aceptación.
- Crear un conjunto específico de pruebas de seguridad para estos criterios de aceptación concretos. Los principios rectores para esto son:
    - Posibilidad de aplicación:

        - ¿Es posible una técnica de prueba específica para el sistema sujeto a prueba (SSP) [en inglés, system under test (SUT)]? (por ejemplo, ¿se dispone de las herramientas correspondientes?)
    - Optimización del coste, tiempo y calidad:
        - El reto consiste en definir un juego de pruebas y unas herramientas específicas que generen resultados de prueba significativos, que puedan aplicarse en un plazo determinado y que sean rentables.

Tras seleccionar las mejores técnicas y herramientas de prueba de seguridad, hay que aplicar la prueba de seguridad, analizar los resultados de la prueba e informar de ellos en el informe de prueba. El propio informe de prueba debe reflejar los criterios de aceptación y constituir la base de la decisión de aceptación de la seguridad.

7.2 Entradas para un sistema de gestión de la seguridad de la información

La prueba en sí no mejora la calidad. La prueba aporta información sobre la calidad alcanzada para una característica de calidad específica, como la seguridad. Un informe de prueba no mejora el nivel de seguridad. Si se analizan los hallazgos del informe de prueba y se resuelve la mayoría de ellos, una prueba de confirmación podría ser apropiada para demostrar un aumento de la seguridad. Además de estas acciones de mitigación del riesgo específicas del sistema, algunos de los hallazgos podrían motivar una política de seguridad específica para evitar dichas vulnerabilidades en el futuro. Por otro lado, algunos de estos hallazgos podrían tener su origen en una falta de concienciación sobre la seguridad o en el uso de técnicas inmaduras/no sistemáticas.

Para aprovechar la prueba de seguridad de forma eficaz y eficiente, debe integrarse en un proceso de seguridad general. Debe tratar de minimizar el riesgo y asegurar la continuidad del negocio limitando de forma proactiva el impacto de una brecha de seguridad. Este es precisamente el objetivo de un SGSI. [ITGov23b] define un SGSI de la siguiente manera:

- Un SGSI adopta un enfoque sistemático para garantizar la confidencialidad, integridad y disponibilidad de los activos de información corporativa.
- Un SGSI ISO 27001 consta de políticas, procedimientos y otros controles que implican a personas, procesos y tecnología.
- Un SGSI es una forma eficiente de mantener la seguridad de los activos de información, basada en evaluaciones periódicas de los riesgos y en enfoques neutrales con respecto a la tecnología y los proveedores.

Un SGSI adopta una visión holística de la seguridad y asegura la interacción eficaz de los tres atributos clave de la seguridad de la información:

- Proceso.
- Tecnología.
- Comportamiento dentro de la organización [Cald11].

La prueba de seguridad de aplicaciones o sistemas está directamente relacionada con la tecnología. Sin embargo, cada vulnerabilidad identificada por las pruebas de seguridad podría tener sus raíces en el proceso y/o el comportamiento y podría mitigarse en el futuro cambiando los procesos y/o el comportamiento.

Existen al menos las siguientes razones para que una organización implemente un SGSI que se apoyan directamente en la prueba de seguridad [Cald11]:

- Estratégicas, para gestionar mejor la seguridad de la información en el contexto de los riesgos generales del negocio.

- Confianza del cliente, para demostrar que una organización cumple con las buenas prácticas de gestión de la seguridad de la información.
- Normativa, para cumplir diversos requisitos reglamentarios.
- Efectividad interna, para gestionar tácticamente la información de forma más eficaz dentro de una organización.

La prueba de seguridad tiene un rol importante en el establecimiento de un SGSI. Puesto que está relacionado con la prueba, demuestra el estado actual de un sistema. Esto puede entenderse de la siguiente manera:

- Como evidencia de un objetivo, que se planea alcanzar.
- Como evidencia de un punto de partida, que motiva nuevas acciones de seguridad.

Un bucle de retroalimentación bien conocido para modelar el objetivo y el punto de partida es el ciclo planificar-hacer-comprobar-actuar (PHCA) [en inglés, plan-do-check-act (PDCA)]. ISO 27001 «adopta el modelo de proceso planificar-hacer-comprobar-actuar (PHCA), que se aplica para estructurar todos los procesos del SGSI» [Cald11]. Ambos aspectos, objetivo y punto de partida, son visibles dentro de este modelo:

- Objetivo:
    - Después de los pasos Planificar y Hacer, el paso Comprobar establece si se ha alcanzado el objetivo planificado.
- Punto de partida:
    - El resultado del paso Comprobar, que se apoya en actividades de prueba de seguridad, se analiza dentro del paso Actuar para mejorar el proceso global. Sus desviaciones son la base para el siguiente ciclo PHCA.

La prueba de seguridad proporciona el mayor valor añadido dentro de una organización si tiene en cuenta ambos aspectos (es decir, el objetivo y el punto de partida).

Para medir la consecución de los pasos Planificar y Hacer en términos de mejora de la seguridad, el enfoque de prueba de seguridad debe ajustarse con precisión, de modo que la prueba de seguridad coincida exactamente con el objetivo previsto. Es una buena práctica definir los criterios de aceptación de las pruebas de seguridad para el paso Comprobar con antelación al definir el plan de seguridad.

Para aprovechar la prueba de seguridad como punto de partida para el siguiente ciclo PHCA, la técnica de prueba, las herramientas utilizadas, los juegos de prueba ejecutados y todos los resultados de prueba (tanto los positivos como los negativos) deben figurar en el informe de prueba.

Es posible aprovechar el mismo enfoque de prueba de seguridad para establecer un nuevo punto de partida y medir la efectividad del siguiente ciclo PHCA. La prueba de seguridad debe llevarse a cabo con un enfoque muy sistemático. Todos los parámetros relevantes deben almacenarse para permitir una prueba de seguridad repetible y objetiva.

7.3 Mejora de un sistema de gestión de la seguridad de la información mediante el ajuste de la prueba de seguridad

La efectividad de un SGSI definido para proporcionar una mayor seguridad depende en gran medida de las acciones proactivas introducidas por un SGSI. Esto significa que un SGSI debe derivar controles de seguridad basados en el estado actual de la seguridad, y una evaluación o una situación actual del negocio (por ejemplo, una incidencia).

Además de cualquier acción directa de mitigación del riesgo que se realice para corregir las dificultades identificadas dentro de las operaciones normales, el SGSI intenta obtener controles que eviten que se

produzcan tales dificultades en un desarrollo futuro. Cuantas más iteraciones PHCA haya experimentado un SGSI específico, mejor será el conjunto de controles de seguridad derivados y mejor será el nivel de seguridad de todas las aplicaciones desarrolladas.

7.3.1 Mejorar la visión holística de un SGSI

Para mejorar un SGSI aumentando su visión holística la prueba de seguridad debe asumir una nueva responsabilidad, además de establecer la línea base para el paso planificar-hacer-comprobar-actuar (PHCA) [en inglés, plan-do-check-act (PDCA)]. Si el objetivo es mejorar la madurez del SGSI, el ingeniero de prueba de seguridad (IPS) debe aportar aspectos completamente nuevos que aún no se habían planificado como parte del ciclo PHCA. Estas nuevas pruebas de seguridad podrían generar conocimientos adicionales sobre el SSP, que luego pueden utilizarse para mejorar aún más la madurez del SGSI obteniendo controles de seguridad adicionales.

Las dimensiones típicas que un ingeniero de prueba de seguridad (IPS) puede utilizar para mejorar el alcance del SGSI son:

- Objetos de prueba adicionales:
    - Cada sistema que debe alcanzar un nivel específico de seguridad tiene que ser tenido en cuenta dentro de su entorno habitual cuando está en producción. El sistema puede estar situado detrás de un cortafuegos, o conectado a una base de datos central, o tener una interfaz IPA con un sistema/aplicaciones externas, o estar controlado por un proceso de respaldo diario, o tener una conexión con un sistema de gestión de cuentas privilegiadas. Cuantos más sistemas estén conectados al sistema sujeto a prueba (SSP), más amplia será la superficie de ataque. Cada sistema puede ser atacado, y si sufre un fallo existe una alta probabilidad de que la red global de sistemas falle también.
- Un SGSI debe abarcar tantos aspectos como sea posible para gestionar la seguridad de la información de la forma más holística posible:
    - Cada aspecto debe reflejar los posibles vectores de ataque. Para ello, es esencial que un ingeniero de prueba de seguridad (IPS) se concentre en tantos objetos de prueba como sea posible, porque cualquiera de ellos podría ser un componente de riesgo en una red global. Si una de las pruebas de seguridad identifica debilidades adicionales en un componente que aún no forma parte del SGSI global, puede mejorar la madurez del SGSI basándose en esta nueva información. No es tarea del ingeniero de pruebas de seguridad (IPS) definir contramedidas (por ejemplo, políticas de seguridad o ajustes de los procesos), pero su tarea sí incluye tener una mentalidad abierta con respecto a que los componentes adicionales del SSP sean útiles para madurar el SGSI.
- Enfoques de prueba adicionales:
    - Otra posibilidad para que un ingeniero de pruebas de seguridad (IPS) aporte información adicional a un sistema es utilizar diferentes tipos de prueba. Incluso si la acción de comprobación como parte del ciclo PHCA se concentra en una prueba dinámica de caja negra del SSP, podría ser beneficioso realizar también una prueba estática. Esto podría mostrar vulnerabilidades adicionales que no se han identificado hasta ahora y que también podrían ser utilizadas por los atacantes. Estos nuevos conocimientos deberían aprovecharse como entrada a un SGSI para mejorar aún más su madurez.
- Mejora de la cobertura:
    - Incluso manteniendo el objeto de prueba y el enfoque de prueba existentes, el ingeniero de pruebas de seguridad (IPS) puede generar información valiosa para la madurez del SGSI. Simplemente añadiendo algunos casos de prueba más se podrían generar perspectivas completamente nuevas. Esto puede hacerse fácilmente utilizando herramientas de prueba aleatorias estructuradas o tablas arcoíris.

Otra forma de mejorar la cobertura podría ser aumentar el número de casos de prueba ejecutados por unidad de tiempo (por ejemplo, automatizando algunos conjuntos de pruebas) o aumentar el número de ciclos de prueba realizados para imponer un comportamiento inusual que pueda utilizarse para ataques. Si estas medidas de cobertura mejoradas identifican vulnerabilidades adicionales, ayudará a mejorar la madurez del SGSI.

7.3.2 Mejorar la capacidad de medición dentro de un SGSI

El ingeniero de prueba de seguridad (IPS) puede mejorar la madurez del SGSI introduciendo un bucle de retroalimentación basado en métricas. Estas métricas suelen denominarse indicadores clave de rendimiento y apoyan la mejora continua mediante ciclos PHCA. La idea fundamental del ciclo PHCA subyacente es comprobar la efectividad de las acciones precedentes de planificar-hacer-comprobar-actuar (PHCA) [en inglés, plan-do-check-act (PDCA)]. Cuanto más objetiva sea esta comprobación, más objetivo será el ciclo de retroalimentación. Las políticas derivadas del SGSI incluyen métricas de cobertura para las técnicas de prueba y el comportamiento. Si una política de la cartera global de políticas de una organización incluye el uso directo de una técnica de prueba, entonces las pruebas de seguridad pueden comprobar directamente si esa política tiene éxito.

Incluso si una política sólo afecta indirectamente a los procesos utilizados, el ingeniero de pruebas de seguridad (IPS) puede apoyar la medición de su efectividad. Por ejemplo, podría ser difícil medir directamente la efectividad de alguna formación impartida sobre convenciones de codificación segura. Una forma podría ser realizar evaluaciones al final de cada sesión de formación. Otra forma sería establecer una prueba de seguridad que comprobara con precisión la aparición de los anti-patrones de seguridad que se habían tratado en la formación. Cuanto más frecuentemente sigan produciéndose estos anti-patrones después de la formación, menor será el valor añadido de la formación en términos de seguridad. Lo mismo podría decirse a la hora de evaluar la efectividad de un comportamiento no deseado. Por ejemplo, la prueba de seguridad podría definir una simulación de suplantación de identidad que cuente el número de clics no deseados realizados dentro de un correo electrónico. Cuando se tiene en cuenta la suplantación de identidad, los porcentajes de clics más elevados suelen considerarse malos porque significan que los usuarios no notifican que el correo electrónico es de suplantación de identidad, mientras que los porcentajes de clics bajos suelen considerarse buenos. Sin embargo, para medir la efectividad de una iniciativa de concienciación, esta prueba de seguridad debe repetirse para habilitar la medición de un cambio antes y después. [StGrTh20]

La prueba de seguridad puede mejorar la madurez del SGSI porque los bucles de retroalimentación se basan en los hechos concretos generados por la prueba de seguridad. Las mejoras se producen más rápidamente y son más fiables que las basadas en subjetividades o sentimientos.

8 Suministro de información sobre los resultados de las pruebas de seguridad - (K3)

Duración: 135 minutos

Palabras clave[8]

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB**: identificarán palabras clave del proceso de prueba
- **ESPDOM**: identificarán palabras clave específicas de dominio: **SEGURIDAD**

Tipo Palabra Clave	Español	Inglés
ISTQB	jáquer ético	ethical hacker
ISTQB	mitigación de riesgo	risk mitigation
ISTQB	informe de prueba	test report

[8] Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.

Objetivos de aprendizaje para "Capítulo 8"

8.1 Suministro de información sobre prueba de seguridad

STE - 8.1.1 (K2) Comprender el carácter crítico de los resultados de la prueba de seguridad y cómo esto afecta a su tratamiento y comunicación.

8.2 Identificación y análisis de vulnerabilidades

STE - 8.2.1 (K3) Evaluar los resultados de una prueba de seguridad determinada para identificar vulnerabilidades.

8.3 Cierre de vulnerabilidades

STE - 8.3.1 (K3) Evaluar diferentes técnicas para cerrar vulnerabilidades identificadas.

8.1 Suministro de información sobre prueba de seguridad

Cada prueba de seguridad termina en un informe de prueba [Glosario ISTQB]. Sin el suministro de información de prueba, ésta carece de evidencias que puedan utilizarse para determinar acciones o decisiones basadas en el resultado de la prueba.

La siguiente información estándar es importante a la hora de informar sobre un caso de prueba de seguridad fallido:

- Entorno de prueba utilizado: Suele incluir direcciones IP específicas, listas blancas de IP aplicadas y cuentas/contraseñas utilizadas.
- Precondiciones de las pruebas ejecutadas. Esto incluye todas las actividades de preparación que deben aplicarse antes de que pueda ejecutarse el juego de pruebas preparado. Esto puede incluir actividades como detalles de registro, ajustes específicos del archivo de configuración o configuraciones específicas del perímetro.
- Datos de prueba utilizados.
- Procedimiento de ejecución de prueba.
- Resultados esperados y resultados reales.

Una prueba de seguridad fallida significa que una prueba específica ha detectado la violación de al menos un aspecto de seguridad de la tríada CID (véase el capítulo 1). Un buen informe de prueba incluye un nivel de detalle suficiente para posibilitar la repetición de la prueba. Las herramientas utilizadas para ejecutar las pruebas podrían identificarse en el informe de prueba y podrían incluirse capturas de pantalla para respaldar los resultados de la prueba con evidencias.

En general, los informes de prueba de seguridad deben tratarse con un alto nivel de confidencialidad. Si este tipo de información se filtra fuera de la organización, podría reducir drásticamente la reputación de ésta. Peor aún, la información podría utilizarse para atacar cualquier sistema que incluya esta vulnerabilidad.

Cuantas más pruebas fallidas contenga un informe de prueba de seguridad, más críticos y sensibles serán el informe de prueba y su comunicación. En general, todo informe de prueba de seguridad debe comunicarse con cuidado dentro de la organización. Esto incluye las comunicaciones internas dentro de la organización que produce el sistema sujeto a prueba (SSP), ya que los atacantes podrían proceder del interior de una organización (cf. [SwissCybInst20]). Por otro lado, los informes de prueba de seguridad pueden ser importantes para muchas personas dentro de una organización. Esta paradoja influye directamente en las actividades de información del ingeniero de prueba de seguridad (IPS) y suele resolverse creando diferentes versiones del mismo informe de prueba, cada una de las cuales contiene distintos niveles de detalle. Cada versión del informe de prueba debe seguir el concepto de «necesidad de conocer».

La sensibilidad de un informe de prueba de seguridad puede modificarse en función de las vulnerabilidades identificadas. Esto es esencial cuando los jáquer éticos identifican una vulnerabilidad en un sistema sujeto a prueba (SSP) y desean informar sólo al desarrollador para darle la oportunidad de mitigar este riesgo antes de que el informe de prueba se ponga a disposición del público. Esta divulgación responsable es una de las características de un jáquer ético, especialmente para los hackers de sombrero blanco [Huneidy21]. Los hackers de sombrero gris suelen utilizar la publicación de informes de prueba para aumentar la presión sobre una organización para que trabaje en parches [Huneidy21].

8.2 Identificación y análisis de vulnerabilidades

Es importante señalar que la ausencia de un conjunto de pruebas fallidas no significa que el sistema carezca de defectos. Incluso los juegos de prueba pasados no significan necesariamente que el vector de ataque examinado no pueda ser explotado. Simplemente significa que con los juegos de prueba utilizados no es posible ser explotado por un vector de ataque analizado.

Si una prueba de seguridad falla, se identifica una vulnerabilidad potencial. El informe de prueba debe aportar todas las evidencias necesarias para repetir el caso de prueba fallido. Un informe de prueba de seguridad puede demostrar muchas vulnerabilidades. Antes de emprender cualquier acción correctiva, deben seguirse los siguientes pasos:

- Demarcación de la vulnerabilidad:
    - Normalmente, una prueba fallida representa un único caso de prueba fallido y representa una vulnerabilidad. Sin embargo, pueden existir otros casos de prueba que muestren la misma vulnerabilidad. Por ejemplo, si se puede utilizar un parámetro de entrada vacío para hacerse con el control de una aplicación, puede que se consiga el mismo comportamiento al utilizar un archivo de 100 MB como parámetro de entrada. Durante la fase de demarcación de la vulnerabilidad, se ejecutan pruebas diferentes pero similares para demarcar la vulnerabilidad identificada. Esto es importante para la posterior evaluación del riesgo y debe contar con el apoyo del ingeniero de prueba de seguridad (IPS) [en inglés security test engineer (STE)].
- Ajuste de la probabilidad del riesgo:
    - Este paso se realiza para comprobar la probabilidad del riesgo de poder identificar una vulnerabilidad en producción. Normalmente, una prueba de seguridad no se realiza en un sistema en producción. Aunque el entorno de prueba sea similar, nunca será completamente idéntico al entorno de producción. En particular, algunos controles de seguridad podrían desactivarse explícitamente para permitir la realización de una prueba de seguridad. Si un sistema sujeto a prueba (SSP) demuestra una vulnerabilidad, ésta podría estar ofuscada por otros parámetros establecidos para la producción. Si este es el caso, la vulnerabilidad sigue existiendo pero no puede explotarse directamente debido a otros parámetros. Este ajuste podría cambiar el nivel de riesgo sugerido por la prueba de seguridad y, por lo tanto, podría cambiar la necesidad general de planificar acciones de mitigación del riesgo. El ingeniero de prueba de seguridad (IPS) tiene la tarea de tener en cuenta este ajuste del riesgo y tomar las medidas correspondientes.
- Ajuste del impacto del riesgo:
    - Este paso se realiza para volver a comprobar el posible impacto del riesgo derivado de la exposición a una vulnerabilidad. Normalmente, el ingeniero de prueba de seguridad (IPS) se concentra en los aspectos técnicos, lo que dificulta el cálculo del impacto en el negocio. Especialmente si el ingeniero de prueba de seguridad (IPS) reutiliza evaluaciones de impacto para vulnerabilidades identificadas de una fuente externa (véase el capítulo 4, CVSS), éstas pueden ser imprecisas para un contexto específico. Si se identifica una vulnerabilidad, los implicados del negocio refinan el posible impacto del riesgo. Es posible que se estime que la vulnerabilidad no tiene ningún impacto desde el punto de vista del negocio (por ejemplo, si el componente afectado se utiliza raramente), o puede que se tenga en cuenta que tiene un alto nivel de criticidad para el negocio. Este ajuste podría modificar el nivel de riesgo sugerido por el ingeniero de prueba de seguridad (IPS) y justificar una actualización de las acciones de mitigación del riesgo previstas

Si se aplican los tres pasos anteriores, se puede obtener una visión clara de la vulnerabilidad identificada y de su riesgo identificado. El ajuste de la probabilidad del riesgo y del impacto del riesgo debe tener en cuenta algunos parámetros importantes:

- Una estrecha comunicación con los implicados del negocio, ya que ellos tienen la última palabra a la hora de discutir el posible impacto del riesgo.
- Estrecha comunicación con el equipo de operaciones, ya que tienen la última palabra a la hora de considerar los parámetros de producción y cómo difieren de los utilizados en las pruebas de seguridad.
- Incluso si la existencia de la vulnerabilidad en el sistema sujeto a prueba (SSP) está clara, los implicados pueden intentar influir activamente en el paso de ajuste del riesgo (ajustando la probabilidad o el impacto)

Si se tiene en cuenta que el nivel de riesgo restante es demasiado alto para entrar en producción o permanecer en producción, debe crearse un plan de mitigación del riesgo. La dirección es responsable de decidir sobre la urgencia de dichos planes de mitigación del riesgo. La decisión puede situarse entre los siguientes niveles de urgencia:

- Detener la operación inmediatamente o detener actividades adicionales de puesta en marcha:
    - Si se considera que el nivel de riesgo es demasiado alto y no puede aceptarse, sólo puede evitarse no ejecutando el sistema sujeto a prueba (SSP) [en inglés, system under test (SUT)]. El nivel de riesgo (por ejemplo, alto, medio o bajo) que influye en esta decisión depende del apetito de riesgo, que se define como «la cantidad de riesgo que una organización está dispuesta a aceptar para alcanzar sus objetivos» [CARM22].
- Continuar operando el sistema con una monitorización intensiva:
    - Si el sistema es demasiado crítico o el riesgo es demasiado crítico, el sistema sujeto a prueba (SSP) puede seguir ejecutándose, pero se monitoriza intensivamente.
- Añadir acciones de mitigación del riesgo al plan de entrega normal:
    - Si se puede tratar el riesgo o el sistema tiene muchas restricciones estrictas de entrega, las acciones de mitigación del riesgo se analizan y se llevan a cabo, pero no se aplican directamente al sistema sujeto a prueba (SSP) [en inglés, system under test (SUT)]. En su lugar, los componentes parcheados se añaden al ciclo de entrega normal para asegurar que la próxima entrega prevista contiene los parches de seguridad requeridos.

El ingeniero de prueba de seguridad (IPS) debe asegurar que se realizan pruebas de confirmación y pruebas de regresión para cada acción de mitigación del riesgo. La prueba de confirmación debe tener en cuenta las evidencias proporcionadas en el informe de prueba y también debe utilizar las lecciones aprendidas de la etapa de demarcación de vulnerabilidades.

8.3 Cierre de vulnerabilidades

Si se conoce la demarcación de una vulnerabilidad identificada y se decide mitigar el riesgo, se dispone de al menos dos alternativas de alto nivel para mitigar las vulnerabilidades identificadas:

- Ocultar la vulnerabilidad reduciendo el riesgo previsto.
- Evitar la vulnerabilidad aplicando parches al sistema afectado.

Ambas alternativas pueden combinarse. Ocultar la vulnerabilidad puede proporcionar una solución a corto plazo y el parche adecuado puede aplicarse después.

8.3.1 Ocultación de una vulnerabilidad

La idea de ocultar vulnerabilidades es similar al enfoque que aplica un probador, cuando diferencia entre un defecto que no causa un fallo y un defecto que causa un fallo, (es decir, que posiblemente pueda afectar a un cliente).

Incluso si los pasos de ajuste del riesgo han demostrado que la vulnerabilidad identificada está causada por un fallo, (es decir, que expone un alto riesgo), el propio sistema podría modificarse de tal manera que el defecto no se revele al cliente, aunque éste permanezca inalterado. En términos de mitigación del riesgo, este tipo de acción tiene como objetivo la reducción del impacto del riesgo. El defecto sigue existiendo en el sistema, pero ya no puede ser explotado. Los enfoques típicos para ocultar vulnerabilidades de este modo son:

- Bloqueo de tráfico:
  - Los cortafuegos modernos permiten análisis y mecanismos de bloqueo muy sofisticados. Si se conoce bien el tráfico necesario para explotar la vulnerabilidad, muchos cortafuegos pueden configurarse para bloquear dichos patrones de tráfico. Al hacerlo, la vulnerabilidad permanece inalterada, pero ya no puede ser explotada.
- Parcheo virtual:
  - El parcheo virtual no bloquea necesariamente el tráfico, pero lo convierte de forma que la vulnerabilidad no pueda ser explotada. [OWASP11] lo define como «una capa de aplicación de la política de seguridad que impide la explotación de una vulnerabilidad conocida. El parche virtual funciona ya que la capa de aplicación de la seguridad analiza las transacciones e intercepta los ataques en tránsito, por lo que el tráfico malicioso nunca llega a la aplicación. El resultado de un parche virtual es que, aunque el código fuente real de la propia aplicación no se ha modificado, el intento de explotación no tiene éxito.»
- Desactivar o reconfigurar funciones específicas del sistema:
  - Si la vulnerabilidad tiene un alcance muy limitado dentro del sistema, podría ser posible desconectar la funcionalidad afectada por la vulnerabilidad identificada.
- Reducir el alcance de las vulnerabilidades:
  - Podría ser posible aceptar el riesgo asociado a una vulnerabilidad reduciendo su alcance. Por ejemplo, podría ser posible establecer filtros IP para que sólo las máquinas conocidas con direcciones IP dedicadas puedan conectarse a la máquina vulnerable. Además, puede ser posible inhabilitar la máquina vulnerable para el acceso externo y permitir únicamente el acceso interno.

El ingeniero de prueba de seguridad (IPS) es responsable, en las pruebas de confirmación, de aplicar todos los enfoques de vulnerabilidad oculta para asegurar que la vulnerabilidad específica ya no puede ser explotada.

8.3.2 Evitar una vulnerabilidad

Por lo general, evitar la vulnerabilidad es una acción que requiere mucho tiempo y es costosa. Para evitar una vulnerabilidad se deben realizar los siguientes pasos:

Paso	Descripción
Localizar la vulnerabilidad	• Dado que la acción de mitigación del riesgo debe llevarse a cabo en el nivel utilizado para implementar la funcionalidad (por ejemplo, código, modelos y configuraciones) puede llevar tiempo identificar el componente afectado y, dentro de ese componente, el área afectada basándose en una vulnerabilidad identificada a nivel del sistema.
Comprender la vulnerabilidad	• Antes de realizar una reparación, debe obtenerse una comprensión completa de la vulnerabilidad mediante el análisis de la vulnerabilidad en el área afectada (por ejemplo, fragmento de código).

Paso	Descripción
Identificar la acción de mitigación del riesgo	• Debe desarrollarse un enfoque para la mitigación del riesgo. • La acción de mitigación puede consistir en un algoritmo completamente nuevo, un componente nuevo, un pequeño cambio de configuración o sólo algunos ajustes menores del código (por ejemplo, incluir un comportamiento de excepción específico).
Ejecución de la acción de mitigación del riesgo	• Se aplica la acción de mitigación del riesgo identificada.
Prueba de confirmación	• Realizar una prueba de confirmación en el sistema para probar si se ha eliminado la vulnerabilidad.
Prueba de regresión	• La prueba es una parte fundamental para evitar vulnerabilidades y no debe concentrarse únicamente en el código modificado. Es esencial que se ejecute el juego de regresión completo para asegurarse de que el sistema sigue funcionando correctamente y de que la acción de mitigación no ha tenido efectos secundarios no deseados.
Despliegue	• Es habitual que, tras el despliegue del sistema, se realice una monitorización minuciosa durante un tiempo para asegurarse de que el sistema funciona correctamente.

9 Herramientas de Prueba de Seguridad - (K3)

Duración: 90 minutos

Palabras clave[9]

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB**: identificarán palabras clave del proceso de prueba
- **ESPDOM**: identificarán palabras clave específicas de dominio: **SEGURIDAD**

Tipo Palabra Clave	Español	Inglés
ISTQB	aplicación dinámica	dynamic application
ISTQB	prueba de seguridad de aplicación interactiva	interactive application security testing
ESPDOM	software de código abierto	open-source software
ESPDOM	análisis de la composición del software (ACS)	software composition analysis (SCA)
ISTQB	prueba de seguridad de aplicación estática	static application security testing
ISTQB	escaneo de vulnerabilidad	vulnerability scanning

[9] Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.

Objetivos de aprendizaje para "Capítulo 9"

9.1 Clasificación de las herramientas de prueba de la seguridad

STE - 9.1.1 (K3) Analizar diferentes casos de uso y aplicar la clasificación de herramientas para pruebas de seguridad

9.2 Selección de herramientas de prueba de seguridad

STE - 9.2.1 (K2) Comprender el uso y los conceptos de las herramientas de prueba de seguridad dinámica.

STE - 9.2.2 (K2) Comprender el uso y los conceptos de las herramientas de prueba de seguridad estática.

9.1 Clasificación de las herramientas de prueba de la seguridad

Existen varias formas de clasificar por categorías las herramientas de prueba de la seguridad. Una selección de estas categorías incluye:

- La actividad del proceso de prueba de seguridad en la que se puede utilizar la herramienta.
- Herramientas de prueba de seguridad de código abierto frente a las de código cerrado.
- Herramientas de análisis estático frente a herramientas de prueba dinámica.
- Plataforma / infraestructura frente a la aplicación.
- Ejecución de prueba de seguridad frente a gestión de prueba de seguridad.
- Prueba de caja negra frente a prueba de caja blanca frente a prueba de caja gris.

Cada una de estas categorías tiene sus ventajas e inconvenientes a la hora de que el ingeniero de pruebas de seguridad (IPS) seleccione la herramienta de prueba de la seguridad más adecuada. Este programa de estudio recoge tres categorías principales:

- Prueba de caja negra frente a prueba de caja blanca.
- Prueba estática de seguridad frente a prueba dinámica de seguridad.
- Herramientas de prueba de seguridad de código abierto frente a herramientas de prueba de seguridad de código cerrado.

Se espera que los ingenieros de prueba de seguridad (IPS) establezcan su propia librería de herramientas para su dominio y sus contextos. No obstante, pueden adoptar las categorías sugeridas.

9.1.1 Herramientas de prueba de caja blanca de seguridad

Si el ingeniero de prueba de seguridad (IPS) tiene acceso a nivel de código, las herramientas de prueba de caja blanca de seguridad le proporcionan conocimientos relacionados con el código, la información de configuración, las librerías utilizadas, las aplicaciones, el sistema y la plataforma, la arquitectura y los detalles de inicio de sesión. Un prerrequisito importante es que el ingeniero de pruebas de seguridad (IPS) tenga permiso para realizar el análisis, ya que intentará identificar y evaluar las vulnerabilidades del sistema y de los sistemas integrados.

9.1.2 Herramientas de prueba de caja negra de seguridad

Un prerrequisito para realizar pruebas de caja negra es tener acceso a una aplicación o sistema en ejecución en un entorno similar al de producción. Esto es necesario para poder ejecutar las pruebas utilizando herramientas de prueba de caja negra de seguridad, que consideran el sistema sujeto a prueba (SSP) como una caja negra y no necesitan ningún conocimiento interno del software. Las herramientas de prueba de caja negra de seguridad concentran su atención en la identificación de vulnerabilidades durante la ejecución de la aplicación / sistema.

9.1.3 Herramientas de prueba de caja gris de seguridad

La realización de prueba de caja gris de seguridad proporciona al ingeniero de pruebas de seguridad información limitada sobre el funcionamiento interno de la aplicación o el sistema y acceso a una versión en ejecución. Las herramientas de prueba de seguridad de caja gris pueden considerarse una mezcla de las herramientas de prueba de caja blanca y de las herramientas de prueba de caja negra. Necesitan cierta

información interna, así como un sistema en funcionamiento. Se concentran en identificar vulnerabilidades ejecutando la aplicación / el sistema y ejecutando pruebas que tengan en cuenta detalles internos.

9.1.4 Herramientas de prueba estática de seguridad

Una dimensión importante para la categorización de las herramientas de prueba de seguridad se basa en la diferencia entre la prueba estática de seguridad y la prueba dinámica de seguridad. Las definiciones, diferencias y descripciones de las pruebas de seguridad estáticas y dinámicas se tratan en la sección 2.1.2.

Dentro de esta categoría, las herramientas pueden tenerse en cuenta según su uso. Entre ellas se incluyen la prueba de redes, la prueba de sistemas operativos, la prueba de bases de datos y la prueba de aplicaciones.

La prueba estática de seguridad tiene mucho en común con la prueba de caja blanca de seguridad. La principal diferencia es que las herramientas de prueba estática de la seguridad no necesitan que la aplicación o el sistema estén en ejecución. La herramienta accede al código, las librerías o los archivos de configuración en su alcance y los analiza con respecto al repositorio interno de la herramienta de vulnerabilidades conocidas de sintaxis, semántica o estándar de codificación para el lenguaje concreto en uso.

Existen varias formas de realizar pruebas de seguridad con herramientas. En el caso de las herramientas de prueba de seguridad estática, éstas incluyen, entre otras, la prueba estática de seguridad de las aplicaciones (PESA) [en inglés, static application security testing (SAST)] y el análisis de la composición del software (SCA) [en inglés, Software Composition Analysis (SCA)]. A continuación se explican con más detalle.

- Prueba estática de seguridad de las aplicaciones (PESA) [en inglés, static application security testing (SAST)]
    - PESA es una actividad típica de pruebas estáticas de seguridad que suele incluirse en una canalización de Dev(Sec)Ops, como se explica en el capítulo 6.1 y en [NIST DevSecOps]. Dentro de estas canalizaciones se ejecuta automáticamente cada vez que algún código ha cambiado y se comprueba entrada. Utilizar PESA de esta forma proporciona una retroalimentación inmediata. PESA se concentra principalmente en la aplicación y, en la mayoría de los casos, no cubre ningún componente de la plataforma o de la infraestructura. Además, estas herramientas proporcionan buenas métricas de cobertura de código.

      Existe una fuerte dependencia entre la prueba de seguridad estática y el atributo de herramienta del software de código abierto, ya que las herramientas de código abierto entregan por definición el código fuente. Esto significa que la prueba estática de seguridad de las aplicaciones (PESA) [en inglés,static application security testing (SAST)] puede realizarse para todos los sistemas de código abierto. No es el caso de las aplicaciones de código cerrado. Podría ser posible realizar algunas pruebas estáticas (por ejemplo, identificar algunas librerías reutilizadas), sin embargo, este tipo de análisis no es posible debido a la ofuscación o por acuerdos de licencia especiales que prohíben los análisis estáticos.
- Análisis de la composición del software (ACS) [en inglés, software composition analysis (SCA)]
    - El análisis de la composición del software (ACS) tiene muchos puntos de contacto con la seguridad. Las herramientas de ACS analizan las vulnerabilidades del código, incluidas las dependencias y los componentes de código abierto utilizados por la aplicación. Estos componentes son bien conocidos y pueden tener varias vulnerabilidades. Las herramientas de ACS pueden sugerir soluciones a las vulnerabilidades de seguridad basándose en los componentes identificados. Para ello, casi todas las herramientas de ACS utilizan la base de datos Common Vulnerabilities and Exposures [CVE21] de vulnerabilidades de dominio público. (véase la sección 4.2.2)

9.1.5 Herramientas de prueba dinámica de seguridad

Las herramientas de prueba dinámica de la seguridad interactúan con el sistema sujeto a prueba (SSP) mientras se está ejecutando. Tanto las pruebas de caja negra como las pruebas de caja gris están estrechamente relacionadas con las pruebas dinámicas de seguridad. Estas herramientas pueden tenerse en cuenta en las categorizaciones de prueba dinámica de seguridad de las aplicaciones (PDSA) [en inglés, dynamic application security testing (DAST)] y prueba de seguridad de aplicación interactiva (PSAI) [en inglés, Interactive Application Security Testing (IAST)].

- Prueba dinámica de seguridad de las aplicaciones (PDSA) [en inglés, dynamic application security testing (DAST)].
    - Al igual que con PESA, la prueba dinámica de seguridad de las aplicaciones (PDSA) se utiliza comúnmente en un contexto DevSecOps [NIST DevSecOps]. La actividad de prueba se realiza automáticamente en la canalización utilizando una herramienta de prueba de caja negra de seguridad configurable. Esta analiza la aplicación o simula a un atacante mientras el software se está ejecutando, buscando vulnerabilidades como la validación de entradas, pruebas aleatorias, autenticación y autorización, configuración y despliegue, gestión de sesiones, tratamiento de errores y criptografía.

      La prueba dinámica de seguridad de las aplicaciones (PDSA) [en inglés, dynamic application security testing (DAST)] escanea y simula diferentes ataques en tiempo real contra el sistema sujeto a prueba (SSP) de forma automatizada para identificar cualquier vulnerabilidad en la aplicación. Las técnicas de prueba se utilizan en un sistema sujeto a prueba (SSP) en ejecución cuando se realizan la prueba dinámica. Como resultado, éstas consumen más tiempo y rendimiento en comparación con la prueba estática. Aunque la prueba dinámica de seguridad de las aplicaciones (PDSA) [en inglés, dynamic application security testing (DAST)] se concentra en la aplicación, las vulnerabilidades identificadas suelen estar relacionadas con los componentes de la infraestructura que son necesarios para ejecutar la aplicación

      PDSA no cubre de forma fiable todas las dificultades de la lista de las 10 principales de OWASP [OWASP Top 10] ni de la lista de las 25 principales de SANS CWE [CWE21]. Muchas herramientas pueden cubrir aspectos específicos de cada una de las clases de vulnerabilidad, pero el uso de estas herramientas puede generar una falsa sensación de seguridad.
- Prueba de seguridad de aplicación interactiva (PSAI) [en inglés, Interactive Application Security Testing (IAST)].
    - La prueba de seguridad de aplicación interactiva (PSAI) es un enfoque de prueba híbrido que aprovecha tanto la prueba estática como la prueba dinámica de seguridad. Las herramientas se utilizan para determinar si las vulnerabilidades conocidas en el código fuente son explotables durante el tiempo de ejecución.

      En el entorno en el que se ejecuta la aplicación se instala un agente que monitoriza la aplicación e identifica cualquier vulnerabilidad en la aplicación mientras el ingeniero de prueba de seguridad (IPS) (o la herramienta de prueba de seguridad dinámica) interactúa con el SSP.

9.1.6 Consideraciones para la selección de herramientas de prueba de seguridad

Los ingenieros de prueba de seguridad (IPS) deben saber qué herramienta seleccionar para cada contexto. Por lo tanto, deben conocer los diferentes esquemas de categorización y las herramientas pertenecientes a cada categoría.

Los catálogos de herramientas de prueba de seguridad pueden ayudar a seleccionar la herramienta adecuada. Se pueden encontrar algunos ejemplos en: [KALI], [OWASP], [SANS] y [NIST]. Hay que tener en cuenta, sin embargo, que las herramientas pueden estar categorizadas de forma diferente en algunos de estos catálogos y pueden estar presentes en uno y faltar en otro.

No es necesario que el ingeniero de prueba de seguridad (IPS) conozca y sea capaz de utilizar todas las herramientas de prueba de la seguridad disponibles. Los IPS que llevan más tiempo trabajando en un dominio/contexto concreto suelen construir sus propias librerías específicas.

Al seleccionar una herramienta o construir una librería de herramientas, se recomienda hacer lo siguiente:

- Concentrarse en lo que se necesita verificar.
- No depender de un único proveedor para obtener los resultados de prueba requeridos. Utilizar varios proveedores para la misma funcionalidad de la herramienta.
- Escanear periódicamente el mercado en busca de nuevas herramientas emergentes.

9.1.6.1 Código abierto frente a código cerrado

La diferencia entre herramientas de prueba de código abierto y de código cerrado (con licencia) puede ser un aspecto importante en la selección de herramientas.

Cualquiera puede participar en el desarrollo de aplicaciones o herramientas de código abierto. Esto ayuda a eliminar las vulnerabilidades de seguridad lo antes posible, al menos si el proyecto de código abierto cuenta con una comunidad de desarrollo activa. Además, las herramientas de código abierto pueden probarse, (al menos teóricamente), de modo que las puertas traseras serían visibles en el código y pueden excluirse.

El software de código abierto puede personalizarse y utilizarse para contextos específicos, lo que le confiere una clara ventaja.

Las siguientes características podrían considerarse desventajas del uso de herramientas de código abierto:

- Falta de apoyo profesional, especialmente cuando no existe una comunidad activa que respalde un producto específico. Sin embargo, algunas organizaciones se especializan en ofrecer soporte para aplicaciones de código abierto (SCA) [en inglés, open-source software (OSS)], que es un aspecto importante para el entorno empresarial.
- Dificultades con las licencias (por ejemplo, cuando se utilizan licencias públicas de GNU No Unix).
- Es necesario disponer de las habilidades de desarrollo adecuadas (por ejemplo, para ajustarse a contextos específicos).
- Si existe una vulnerabilidad en un sistema de código abierto, existe el riesgo de que alguien la identifique y cree elementos explotables.
- El desarrollo posterior de las aplicaciones de código abierto es incierto, ya que en su mayoría están impulsadas por una comunidad.

A diferencia de las aplicaciones de SCA, las de código cerrado utilizan código propietario que no está disponible para el usuario. Esto ofrece la ventaja de que el proveedor ofrece contratos de soporte.

Las siguientes características podrían considerarse desventajas del uso de software de código cerrado:

- Hay que pagar derechos de licencia.
- No hay garantías contra las puertas traseras.
- Cualquier vulnerabilidad de seguridad podría permanecer desconocida durante mucho tiempo.
- Puede producirse una fuerte dependencia del proveedor cuando un cliente se vuelve muy dependiente de los productos o servicios de un proveedor específico, lo que dificulta el conmutador («switch») a un proveedor diferente.
- En general, sólo existe una capacidad limitada para adaptar la herramienta a un contexto específico (por ejemplo, no se puede modificar el código).

.

9.2 Uso de herramientas de prueba de seguridad

9.2.1 Comprender el uso y los conceptos de las herramientas de prueba estática de seguridad

Los siguientes aspectos describen algunas de las principales características de las herramientas de prueba estática de seguridad:

- Son más eficaces si existe un conocimiento del sistema sujeto a prueba (SSP) [en inglés, system under test (SUT)].
- Pueden utilizarse incluso si la propia aplicación puede estar incompleta y contener defectos.
- Están muy relacionadas con la prueba de caja blanca.
- Son muy adecuadas para encontrar código inseguro o configuraciones erróneas al principio del ciclo de vida de desarrollo de software (CVDS), ya que sólo requieren información estática del código fuente.
- Cubren la totalidad del código fuente y las configuraciones y, por tanto, requieren acceso de lectura a los mismos.
- Son capaces de leer el objeto dado, como el código fuente de una aplicación sin compilar, y compararlo con conjuntos de datos predefinidos de buenas prácticas y vulnerabilidades conocidas (por ejemplo, comandos y cadenas no seguros dentro del código fuente).
- La automatización los hace rentables
- A menudo proporcionan resultados de falso positivo, ya que no son conscientes del contexto. Esto significa que desconocen los casos de uso, la pila de llamadas y la composición de varias líneas de código. Como resultado, podrían identificar vulnerabilidades en código que nunca se ejecutarán en la vida real y que, por lo tanto, no tienen ningún impacto negativo en la seguridad.
- Utilizando, por ejemplo, el escaneado de la red, la herramienta SAST evaluaría los archivos de configuración para encontrar configuraciones no seguras.

9.2.2 Comprender el uso y los conceptos de las herramientas de prueba dinámica de seguridad

La aplicación continua de la seguridad es importante en el contexto del enfoque de integración y entrega continua que se encuentra en el desarrollo ágil de software. La seguridad necesita ser verificada continua y repetidamente en cada incremento. La motivación para utilizar herramientas de prueba dinámicas es hacer que se ejecuten de forma regular y aplicar la automatización para asegurar que se encuentran y se informan inmediatamente las últimas vulnerabilidades conocidas. La monitorización y mejora continua de la seguridad también se denomina DevSevOps, (véase el capítulo 6 y [NIST DevSecOps]).

La prueba dinámica de seguridad de las aplicaciones (DAST) suele concentrarse en las 10 principales vulnerabilidades de OWASP, como: inyecciones (por ejemplo, inyección SQL, guionizado entre sitios e inyección de comandos), controles de acceso fallidos, falsificación de solicitud entre sitios, condiciones de secuencia, errores de lógica de negocio, fugas de memoria y vulnerabilidades conocidas.

También es importante mencionar que las herramientas de prueba dinámica de seguridad de las aplicaciones (PDSA) [en inglés, dynamic application security testing (DAST)] automatizadas sólo escanean un conjunto de vectores de ataque predefinidos. Por lo tanto, siguen siendo obligatorias otras pruebas y comprobaciones de seguridad.

Utilizando el ejemplo del escaneado de la red, una herramienta de prueba dinámica escanea primero la red en busca de puertos abiertos y, a continuación, ejecuta los servicios bajo esos puertos.

10 Referencias

[Bittau] Cryptographic protection of TCP Streams (tcpcrypt) https://tools.ietf.org/html/draft-bittau-tcp-crypt-04

[BSI21] https://www.bsi.bund.de/EN/Themen/KRITIS-und-regulierte-Unternehmen/Weitere_regulierte_Unternehmen/LuSi/Luftsicherheit_node.html, accessed on November 7, 2022

[Bullock2017] Jessey Bullock: “Wireshark for Security Professionals: Using Wireshark and the Metasploit”, Wiley 2017, online available via https://computerscience.unicam.it/marcantoni/reti/laboratorio_wireshark/Wireshark%20for%20Security%20Professionals%20-%20Using%20Wireshark%20and%20the%20Metasploit%20Framework.pdf

[BURP22] https://portswigger.net/burp, last accessed on 1. December 2022

[Cald11] Alan Calder, Jan van Bon: “Implementing Information Security based on ISO 27001/ISO 27002 – a management guide”, Van Haren Publishing, 2011

[CAPEC21] CAPEC: “Common Attack Pattern Enumeration and Classification: A Community Resource for Identifying and Understanding Attacks”, online available via https://capec.mitre.org/

[CARM22] Mary Carmichael, CRISC, CISA, CPA, Member of ISACA Emerging Trends Working Group: “Risk Appetite vs. Risk Tolerance: What is the Difference?”, online available at https://www.isaca.org/resources/news-and-trends/isaca-now-blog/2022/risk-appetite-vs-risk-tolerance-what-is-the-difference

[CERT]: https://wiki.sei.cmu.edu/confluence/pages/viewpage.action?pageId=88487865

[CERT1]: https://wiki.sei.cmu.edu/confluence/display/seccode/Top+10+Secure+Coding+Practices

[CIS22] https://www.cisecurity.org/cis-benchmarks/ , last accessed on 1. December 2022

[CLAIR22] https://github.com/quay/clair , last accessed on 1. December 2022

[Cloudflare] https://www.cloudflare.com/learning/security/glossary/what-is-zero-trust/ , last accessed on 1. December 2023

[CVE21] CVE® Program Mission: “Identify, define, and catalog publicly disclosed cybersecurity vulnerabilities.”, online available via cve.org

[CVSS21] First, Improving Security together: “Common Vulnerability Scoring System SIG”, online available via https://www.first.org/cvss/

[CWE21] CWE: “Common Weakness Enumeration: A Community-Developed List of Software & Hardware Weakness Types”, online available via https://cwe.mitre.org

[CWSS21] CWE: “Scoring CWEs: Common Weakness Scoring System CWSS”, online available via https://cwe.mitre.org/cwss/cwss_v1.0.1.html

[FUZZDB22] https://github.com/fuzzdb-project/fuzzdb , last accessed on 1. December 2022

[Gart21] Gartner Information Technology: “Glossary“, online available via https://www.gartner.com/en/information-technology/glossary

[GITLAB22] https://docs.gitlab.com/ee/user/application_security/sast/ last accessed on 1. December 2022

[GTFO22] https://gtfobins.github.io/, accessed on November 13, 2022

[Huneidy21] Mariangel Huneidy: “The Ultimate Guide to Ethical Hacking “, Sept 21, online available via The Ultimate Guide to Ethical Hacking (0x1.gitlab.io)

[IETF23] Internet Engineering Task Force: “Introduction to the IETF”, online available via https://www.ietf.org/about/introduction/

[ISO 15288] ISO/IEC 15288 – System Life Cycle Processes.

[ISO 25010] ISO/IEC 25010:2011

[ISO 27001] ISO International Standards Organization: “ISO/IEC 27001:2013; Information technology — Security techniques — information security management system— Requirements”

[ISO 31000], ISO 31000:2018 -Risk management

[ISO/IEC/IEEE 29119-3], Software and systems engineering — Software testing – Part 3 – Test Documentation

[ISO/IEC/IEEE 29119-4], Software and systems engineering — Software testing – Part 4 – Test Techniques

[ISO_Web_21] ISO International Standards Organization: “Consumers and standards: Partnership for a better world”, online available via

https://www.iso.org/sites/ConsumersStandards/1_standards.html

[ISTQB FL]: ISTQB Certified Tester Foundation Level Syllabus v4.0 Certified Tester Foundation Level (CTFL) v4.0 [NEW!] (istqb.org)

[ISTQB Glossary]: International Software Test Qualification Board: Glossary

https://glossary.istqb.org/en/search/

[ISTQB_ATTA_SYL]: ISTQB Certified Tester Advanced Level Technical Test Analyst (CTAL-TTA) Syllabus Technical Test Analyst (istqb.org)

[ISTQB_ATA_SYL]: ISTQB Certified Tester Advanced Level Test Analyst (CTAL-TA) Syllabus Test Analyst (istqb.org)

[ITGOV23a] IT-Governance: “ISO 27000 Series of Standards”, available via https://www.itgovernance.co.uk/iso27000-family

[ITGov23b]: “ISO/IEC 27001 – Information Security Management: The international standard for information security”, in IT-Governance, available via https://www.itgovernance.co.uk/iso27001

[KALI] https://www.kali.org/tools/ , last accessed 2. December 2022

[KUBEAUDIT22] https://github.com/Shopify/kubeaudit, last accessed on 1. December 2022

[METASPLOIT22] https://www.metasploit.com/ , last accessed on 1. December 2022

[Micro09] Microsoft: “The STRIDE Threat Model”, 12/2009, via The STRIDE Threat Model | Microsoft Docs

[Micro22] Microsoft: Zero Trust Essentials eBook, via Zero Trust Essentials eBook (microsoft.com)

[MITRE21] The MITRE Corporation: “Media FAQs”, 2018, via media-FAQs-2018.pdf (mitre.org)

[NESSUS22] https://tenable.com/products/nessus , last accessed on 1. December 2022

[NIST DevSecOps] https://csrc.nist.gov/Projects/devsecops

[NIST Glossary] https://csrc.nist.gov/glossary

[NIST 800-160] https://csrc.nist.gov/pubs/sp/800/160/v2/r1/final

[NIST SP 800-161] Supply Chain Risk Management Practices for Federal Information Systems and Organizations

[NIST02] https://csrc.nist.gov/glossary/term/security_service, accessed on November 6, 2022

[NIST03] NIST Special Publication 800-61, Revision 2, Computer Security Incident Handling Guide

[NIST05] https://csrc.nist.gov/glossary/term/social_engineering, accessed on November 13, 2022

[NIST06] NIST SP 800-137, Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations, accessed on September 2011

[NISTIR 7007]: An Overview of Issues in Test Intrusion Detection Systems

[NMAP22] https://https://nmap.org/ , last accessed on 1. December 2022

[OPENVAS22] https://www.openvas.org/ , last accessed on 1. December 2022

[OSI] https://opensource.org/definition/

[OWASP byte code obfuscation]: https://owasp.org/www-community/controls/Bytecode_obfuscation

[OWASP Test Guide]: https://owasp.org/www-project-web-security-test-guide

[OWASP Top 10] https://www.owasp.org

[OWASP] https://owasp.org/www-project-web-security-test-guide/v41/6-Appendix/A-Test_Tools_Resource, last accessed 2. December 2022

[OWASP11] Ryan Barnett, Dan Cornell, Achim Hoffmann Martin Knobloch “Virtual patching Best Practices”, 2011, shared by OWASP, online available at https://owasp.org/www-community/Virtual_Patching_Best_Practices

[OWASP21] OWASP: “The Open Web Application Security Project”, via https://owasp.org/

[OWASP24] OWASP Dependency-Check. https://owasp.org/www-project-dependency-check/

[PCI DSS Chapter 6]: https://www.pcidssguide.com/pci-dss-requirement-6

[PCI22] https://listings.pcisecuritystandards.org/documents/PCI-DSS-v4_0.pdf, accessed on November 7, 2022

[RFC4765] The Intrusion Detection Message Exchange Format (IDMEF), Network Working Group

[SANS] https://www.sans.org/tools/, last accessed 2. December 2022

[SecJour21] Security Journey: “Why vulnerability list methodologies matter”. Available via https://www.securityjourney.com/post/why-vulnerability-list-methodologies-matter-and-why-we-trust-cwe-owasp

[SENG22] https://www.social-engineer.org/framework/information-gathering/dumpster-diving/, accessed on November 10, 2022

[Shacklett] Shacklett, M. E., What is an attack vector? https://www.techtarget.com/searchsecurity/definition/attack-vector

[SNYK22] https://snyk.io/ , last accessed on 1. December 2022

[SONAR22] https://www.sonarqube.org/, last accessed on 1. December 2022

[SQLMAP22] https://sqlmap.org , last accessed on 1. December 2022

[SSLSCAN22] https://github.com/rbsec/sslscan , last accessed on 1. December 2022

[SSLYZE22] https://github.com/nabla-c0d3/sslyze , last accessed on 1. December 2022

[Stallings18] Stallings William, Brown Lawrie, 2018, Computer Security: Principles and Practice, ISBN 9781292220611

[StGrTh20] Michelle P. Steves, Kristen K. Greene and Mary F. Theofanos: “Categorizing Human Phishing Detection Difficulty: A Phish Scale”, Journal of Cybersecurity. Published online Sept. 14, 2020. Available via https://academic.oup.com/cybersecurity/article/6/1/tyaa009/5905453

[SwissCybInst20] Swiss Cyber Institute: “41 Insider Threat Statistics You Should Care About”, Sept 21, online available via 41 Insider Threat Statistics You Should Care About -Swiss Cyber Institute

[SYNOPSIS] https://go.synopsys.com/software-integrity-agile-security-manifesto.html

[TechTarget] TechTarget: “The three ways: The Phoenix project”, available via

https://www.techtarget.com/whatis/definition/The-Three-Ways

[TELETRUST] https://www.stand-der-technik-security.de/startseite/

[TERRASCAN22] https://runterrascan.io/, last accessed on 1. December 2022

[TR02021] https://www.bsi.bund.de/EN/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/TR-nach-Thema-sortiert/tr02102/tr02102_node.html

[UNECE20] https://unece.org/sustainable-development/press/un-regulations-cybersecurity-and-software-updates-pave-way-mass-roll, accessed on November 7, 2022

[HIPAA] https://www.cdc.gov/phlp/publications/topic/hipaa.html, last accessed on 22 August 2023 [WaCh90] Dolores R. Wallace, John C. Cherniavsky: “Guide to Software Acceptance”, NIST Special Publication 500-180, 1990, online available via

https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication500-180.pdf

[WAFEC]: https://owasp.org/www-project-wafec/

[WIKI01] https://en.wikipedia.org/wiki/Google_hacking, accessed on November 10, 2022

[WIKI02] https://en.wikipedia.org/wiki/Stuxnet, accessed on November 10, 2022

[WIRED21] https://www.wired.com/story/solarwinds-hack-supply-chain-threats-improvements/, accessed on November 6, 2022

[ZAP22] https://www.zaproxy.org/ , last accessed on 1. December 2022

Anexo A - Objetivos de aprendizaje/nivel cognitivo de conocimiento

Los siguientes objetivos de aprendizaje se definen como aplicables a este programa de estudio. Cada tema del programa de estudio se examinará de acuerdo con el objetivo de aprendizaje correspondiente.

Los objetivos de aprendizaje comienzan con un verbo de acción correspondiente a su nivel cognitivo de conocimiento, tal y como se indica a continuación.

Nivel 1: Recordar (K1)

El candidato recordará y reconocerá un término o concepto.

Verbos de acción: Recordar, reconocer.

Ejemplos	Notas
Recordar los conceptos de la pirámide de prueba.	No hay notas.
Reconocer los objetivos comunes de la prueba.	No hay notas.

Nivel 2: Comprender (K2)

El candidato puede seleccionar las razones o explicaciones de los enunciados relacionados con el tema y puede resumir, comparar, clasificar y dar ejemplos para el concepto de prueba.

Verbos de acción: clasificar, comparar, diferenciar, distinguir, explicar, dar ejemplos, interpretar, resumir.

Ejemplos	Notas
Clasificar herramientas de prueba en función de su finalidad y de las actividades de prueba a las que dan soporte.	No hay notas.
Comparar los distintos niveles de prueba.	Puede utilizarse para buscar similitudes, diferencias o ambas.
Diferencie la prueba de la depuración.	Busca diferencias entre conceptos.
Distinga entre riesgos de proyecto y de producto.	Permite clasificar por separado dos (o más) conceptos.
Explicar el impacto del contexto en el proceso de prueba.	No hay notas.
Aportar ejemplos de por qué es necesaria la prueba.	No hay notas.
Inferir la causa raíz de defectos a partir de un perfil dado de fallos.	No hay notas.
Resumir las actividades del proceso de revisión del producto de trabajo.	No hay notas.

Nivel 3: Aplicar (K3)

El candidato puede llevar a cabo un procedimiento cuando se enfrenta a una tarea conocida, o seleccionar el procedimiento correcto y aplicarlo a un contexto determinado.

Verbos de acción: aplicar, implementar, preparar, utilizar.

Ejemplos	Notas
Aplicar el análisis del valor frontera para obtener casos de prueba a partir de unos requisitos dados.	Debería referirse a un procedimiento / técnica / proceso, etc.
Implementar métodos para la recopilación de métricas que respalden los requisitos técnicos y de gestión.	No hay notas.
Preparar pruebas de instalabilidad para aplicaciones móviles.	No hay notas.
Utilizar la trazabilidad para monitorizar el avance de la prueba en cuanto a compleción y consistencia con los objetivos de prueba, la estrategia de prueba y el plan de prueba.	Podría utilizarse en un Objetivo de Aprendizaje que quiera que el candidato sea capaz de utilizar una técnica o procedimiento. Similar a “aplicar”.

Nivel 4: Analizar (K4)

El candidato puede separar la información relacionada con un procedimiento o técnica en sus partes constituyentes para una mejor comprensión y puede distinguir entre hechos e inferencias. Una aplicación típica es analizar la situación de un documento, software o proyecto y proponer acciones adecuadas para resolver un problema o tarea.

Verbos de acción: Analizar, deconstruir, esquematizar, priorizar, seleccionar.

Ejemplos	Notas
Analizar una situación concreta de un proyecto para determinar qué técnicas de prueba de caja negra o basadas en la experiencia deben aplicarse para alcanzar unos objetivos específicos.	Evaluable sólo en combinación con un objetivo medible del análisis. Debe tener la forma 'Analizar xxxx a xxxx' (o similar).
Dar prioridad a los casos de prueba de un determinado juego de prueba para su ejecución basándose en los riesgos de producto relacionados.	No hay notas.
Seleccionar los niveles de prueba y los tipos de prueba adecuados para verificar un conjunto de requisitos dado.	Necesario cuando la selección necesita análisis.

Referencias para los niveles cognitivos de los objetivos de aprendizaje

- Anderson, L. W. and Krathwohl, D. R. (eds) (2001) A Taxonomy for Learning, Teaching
- Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Allyn & Bacon

Anexo B - Matriz de trazabilidad de los resultados de negocio con respecto a los objetivos de aprendizaje

Esta sección enumera el número de objetivos de aprendizaje de nivel básico relacionados con los resultados de negocio y la trazabilidad entre los resultados de negocio de nivel básico y los objetivos de aprendizaje de nivel básico.

STE-BO1	Comprender los paradigmas fundamentales de la seguridad y su impacto en la prueba de seguridad
STE-BO2	Utilizar y aplicar las técnicas de prueba de seguridad adecuadas y conocer sus puntos fuertes y sus limitaciones
STE-BO3	Contribuir a la planificación, el diseño y la ejecución de las pruebas de seguridad
STE-BO4	Comprender cómo pueden utilizarse los estándares de prueba de seguridad y las buenas prácticas de seguridad para probar la seguridad.
STE-BO5	Ajustar y realizar actividades de pruebas de seguridad de acuerdo con el contexto específico de la organización.
STE-BO6	Ajustar y realizar actividades de pruebas de seguridad de acuerdo con métodos de desarrollo y ciclos de vida de desarrollo del software específicos.
STE-BO7	Introducir los resultados de la prueba de seguridad en un sistema de gestión de la seguridad de la información (SGSI) para una gestión activa del riesgo de seguridad.
STE-BO8	Recopilar, evaluar y agregar los resultados de las pruebas y redactar un informe de prueba detallado que incluya todas las evidencias y hallazgos.
STE-BO9	Identificar los requisitos adecuados para las herramientas en función del enfoque de prueba de la seguridad requerido y ayudar a seleccionar las herramientas de prueba de la seguridad.

Capítulo/ sección/ subsección	Objetivos de Aprendizaje	Nivel - K	B01	B02	B03	B04	B05	B06	B07	B08	B09
1	**Paradigmas de seguridad**										
1.1	**Niveles de seguridad de activos**										
STE - 1.1.1	Explicar los distintos niveles de seguridad de los activos y su correspondiente nivel de protección.	**2**	X								
STE - 1.1.2	Explicar la relación entre la sensibilidad de la información y la prueba de seguridad.	**2**	X								
1.2	**Auditorías de seguridad**										
STE - 1.2.1	**Describir el rol de la prueba de seguridad en el contexto de las auditorías de seguridad.**	**2**	**x**								
1.3	**El concepto de confianza cero**										
STE - 1.3.1	Explicar el concepto de confianza cero.	**2**	**x**								
STE - 1.3.2	Aplicar la confianza cero en la prueba de seguridad.	**3**	**x**								
1.4	**Software de código abierto (SCA)**										
STE - 1.4.1	Aportar un ejemplo del concepto de reutilización de software de código abierto en el desarrollo de software y su impacto en la prueba de seguridad	**2**	**x**								

Capítulo/ sección/ subsección	Objetivos de Aprendizaje	Nivel - K	B01	B02	B03	B04	B05	B06	B07	B08	B09
2	**Técnicas de prueba de seguridad**										
2.1	**Uso de tipos de prueba de seguridad en función de un contexto de prueba**										
STE - 2.1.1	Aportar ejemplos de tipos de prueba de seguridad en función de un contexto de seguridad de caja negra, caja blanca y caja gris.	**2**		**x**							
STE - 2.1.2	Aportar ejemplos de tipos de prueba de seguridad en función de una prueba de seguridad estática o una prueba de seguridad dinámica.	**2**		**x**							
2.2	**Uso de tipos de prueba de seguridad en función de un proyecto y un contexto técnico**										
STE - 2.2.1	Aplicar casos de prueba de seguridad, basados en un enfoque de prueba de seguridad dado, junto con riesgos de seguridad funcionales y estructurales identificados.	**3**		**x**							
STE - 2.2.2	Describir cómo probar la reconciliación y recertificación de identidades y permisos.	**2**		**x**							
STE - 2.2.3	Describir cómo probar el control de gestión de identidades y accesos.	**2**		**x**							
STE - 2.2.4	Describir cómo probar el control de protección de datos.	**2**		**x**							
STE - 2.2.5	Describir cómo probar tecnologías de protección.	**2**		**x**							
3	**El proceso de prueba de seguridad**										

Capítulo/ sección/ subsección	Objetivos de Aprendizaje	Nivel - K	B01	B02	B03	B04	B05	B06	B07	B08	B09
3.1	**El proceso de prueba de seguridad**										
STE - 3.1.1	Explicar las diferentes actividades, tareas y responsabilidades dentro de un proceso de prueba de seguridad	**2**			**x**						
STE - 3.1.2	Comprender los elementos clave y las características de un entorno de prueba de seguridad.	**2**			**x**						
3.2	**Diseño de pruebas de seguridad para niveles de prueba**										
STE - 3.2.1	Aportar ejemplos de pruebas de seguridad en el nivel de prueba de componente basado en un código determinado	**2**			**x**						
STE - 3.2.2	Aportar ejemplos de pruebas de seguridad en el nivel de integración de componentes basadas en una especificación de diseño determinada.	**2**			**x**						
STE - 3.2.3	Implemente una prueba de seguridad de extremo a extremo que valide uno o más requisitos de seguridad relacionados con uno o más procesos de negocio.	**3**			**x**						
4	**Estándares y buenas prácticas de la prueba de seguridad**										
4.1	**Introducción a los estándares y buenas prácticas de seguridad**										
STE - 4.1.1	Explicar las diferentes fuentes de estándares, buenas prácticas y su aplicabilidad.	**3**				**x**					
4.2	**Aplicación de estándares importantes y buenas prácticas en la prueba de seguridad**										

Capítulo/ sección/ subsección	Objetivos de Aprendizaje	Nivel - K	B01	B02	B03	B04	B05	B06	B07	B08	B09
STE - 4.2.1	**Aplicar el concepto del proyecto abierto de seguridad de las aplicaciones web, la enumeración de vulnerabilidades comunes, la enumeración de debilidades comunes, el sistema de puntuación de vulnerabilidades comunes y el sistema de puntuación de debilidades comunes y cómo aprovecharlos en la prueba de seguridad.**	**3**				**x**					
4.3	**Aprovechamiento de estándares y buenas prácticas de la prueba de seguridad**										
STE - 4.3.1	**Explicar las ventajas y desventajas de los oráculos de prueba utilizados para la prueba de seguridad.**	**2**				**x**					
STE - 4.3.2	Comprender las ventajas y desventajas de utilizar los mejores estándares y las mejores prácticas de seguridad	**3**				**x**					
5	**Adaptación de la prueba de seguridad al contexto de la organización**										
5.1	**Impacto de las estructuras de una organización en el contexto de la prueba de seguridad**										
STE - 5.1.1	**Analizar un contexto dado de una organización y determinar qué aspectos específicos hay que tener en cuenta para la prueba de seguridad.**	**3**					**x**				
5.2	**Impacto de normativas en las políticas de seguridad y cómo probarlas**										
STE - 5.2.1	**Analizar el impacto de las normativas en las políticas de seguridad y cómo probarlas.**	**3**					**x**				
5.3	**Análisis de un escenario de ataque**										

Capítulo/ sección/ subsección	Objetivos de Aprendizaje	Nivel - K	B01	B02	B03	B04	B05	B06	B07	B08	B09
STE - 5.3.1	**Analizar un escenario de ataque e identificar las posibles fuentes y motivaciones del ataque.**	**4**					**x**				
6	**Adaptación de la prueba de seguridad a los modelos de ciclo de vida del desarrollo del software**										
6.1	**Efectos de los diferentes modelos de ciclo de vida del desarrollo de software en la prueba de seguridad**										
STE - 6.1.1	Resumir por qué las actividades de prueba de seguridad deben cubrir el ciclo de vida de desarrollo del software.	**2**						**x**			
STE - 6.1.2	Analizar cómo se ven afectadas las actividades de prueba de seguridad por los diferentes modelos de ciclo de vida de desarrollo de software.	**4**						**x**			
6.2	**Prueba de seguridad durante el mantenimiento**										
STE - 6.2.1	Definir y realizar pruebas de regresión de seguridad y pruebas de confirmación basadas en un cambio en un sistema.	**3**						**x**			
STE - 6.2.2	Analizar los resultados de las pruebas de seguridad para determinar la naturaleza de una vulnerabilidad y su impacto técnico potencial.	**2**						**x**			
7	**Prueba de seguridad como parte de un sistema de gestión de la seguridad de la información**										
7.1	**Criterios de aceptación para la prueba de seguridad**										
STE - 7.1.1	**Comprender los criterios de aceptación de la prueba de seguridad y cómo influyen en la selección de enfoques de prueba de seguridad y técnicas de prueba.**	**2**							**x**		

Capítulo/ sección/ subsección	Objetivos de Aprendizaje	Nivel - K	B01	B02	B03	B04	B05	B06	B07	B08	B09
7.2	**Entradas para un sistema de gestión de la seguridad de la información**										
STE - 7.2.1	**Comprender el rol de la prueba de seguridad para la efectividad de un sistema de gestión de la seguridad de la información.**	**2**							**x**		
7.3	**Mejora de un sistema de gestión de la seguridad de la información mediante el ajuste de la prueba de seguridad**										
STE - 7.3.1	Evaluar la madurez de un sistema de gestión de la seguridad de la información mediante la incorporación de diferentes enfoques de prueba, nuevos objetos de prueba o una cobertura mejorada.	**3**							**x**		
STE - 7.3.2	Comprender la capacidad de ser medido dentro de un sistema de gestión de la seguridad de la información.	**2**							**x**		
8	**Suministro de información sobre los resultados de las pruebas de seguridad**										
8.1	**Suministro de información sobre prueba de seguridad**										
STE - 8.1.1	**Comprender el carácter crítico de los resultados de la prueba de seguridad y cómo esto afecta a su tratamiento y comunicación.**	**2**								**x**	
8.2	**Identificación y análisis de vulnerabilidades**										
STE - 8.2.1	**Evaluar los resultados de una prueba de seguridad determinada para identificar vulnerabilidades.**	**3**								**x**	
8.3	**Cierre de vulnerabilidades**										

Capítulo/ sección/ subsección	Objetivos de Aprendizaje	Nivel - K	B01	B02	B03	B04	B05	B06	B07	B08	B09
STE - 8.3.1	**Evaluar diferentes técnicas para cerrar vulnerabilidades identificadas.**	**3**								**x**	
9	**Herramientas de Prueba de Seguridad**										
9.1	**Clasificación de las herramientas de prueba de la seguridad**										
STE - 9.1.1	**Analizar diferentes casos de uso y aplicar la clasificación de herramientas para pruebas de seguridad**	**3**									**x**
9.2	**Selección de herramientas de prueba de seguridad**										
STE - 9.2.1	Comprender el uso y los conceptos de las herramientas de prueba de seguridad dinámica.	**2**									**x**
STE - 9.2.2	Comprender el uso y los conceptos de las herramientas de prueba de seguridad estática.	**2**									**x**

Apéndice C - Notas de la entrega

ISTQB® Ingeniero de Prueba de Seguridad es una nueva entrega. Por esta razón, no hay notas de la publicación detalladas por capítulo y sección.

Apéndice D - Términos específicos del dominio

Término en español	Término en inglés	Definición
autenticación	authentication	Procedimiento que determina si una persona o un proceso es, de hecho, quien o lo que declara ser.
autorización	authorization	Permiso otorgado a un usuario o proceso para acceder a recursos.
cifrado	encryption	Proceso de codificación de la información para que sólo las partes autorizadas puedan recuperar la información original, normalmente mediante una clave o proceso de descifrado específico.
cortafuegos	firewall	Un componente o conjunto de componentes que controla el tráfico entrante y saliente de la red basándose en reglas de seguridad predeterminadas.
sistema de gestión de la seguridad de la información (SGSI)	information security management system (ISMS)	Parte de un sistema de gestión global, basado en un enfoque de riesgos de negocio, para establecer, implementar, operar, monitorizar, revisar, mantener y mejorar la seguridad de la información.
sensibilidad de la información	information sensitivity	Una medida de la importancia de proteger la información asignada por su propietario (Según el NIST).
Open Web Application Security Project (OWASP)	Open Web Application Security Project (OWASP)	OWASP, el Open Web Application Security Project, enumera las debilidades más comunes y es bastante famoso por publicar su clasificación OWASP Top 10.
software de código abierto	open-source software	Software al que cualquiera puede acceder, utilizar, modificar y compartir.
escalada de privilegios	privilege escalation	La explotación de un elemento explotable (fragmento de código) o defecto que permite un nivel de privilegio superior al que normalmente estaría permitido.
rootkit	rootkit	Conjunto de herramientas utilizadas por un atacante para obtener y mantener el acceso de nivel raíz a un dispositivo anfitrión con el fin de ocultar las actividades del atacante a través de medios encubiertos. (Después del NIST)
política de seguridad	security policy	Documento de alto nivel que describe los principios, el enfoque y los principales objetivos de la organización en materia de seguridad.
servicio de seguridad	security service	Una capacidad que respalda uno o varios objetivos de seguridad. (Después del NIST)
ingeniería social	social engineering	Un intento de engañar a alguien para que revele información (por ejemplo, una contraseña) que pueda utilizarse para atacar sistemas o redes.
análisis de la composición del software (ACS)	software composition analysis (SCA)	Una práctica para identificar los componentes de código abierto y de código cerrado en uso en una aplicación, sus vulnerabilidades de seguridad conocidas y las

Término en español	Término en inglés	Definición
		restricciones de licencia de los adversarios (Después del NIST).
STRIDE	STRIDE	Acrónimo de seis categorías de amenazas (es decir, suplantación de identidad, manipulación, repudio, revelación de información, denegación de servicio y elevación de privilegios) utilizadas para modelar las amenazas potenciales a un sistema.
confianza cero	zero-trust	Un modelo diseñado para minimizar la incertidumbre a la hora de aplicar decisiones de acceso precisas y con los mínimos privilegios por solicitud en un componente, sistema y servicios si se considera que una red está comprometida. (Glosario del NIST)