



UTPL
La Universidad Católica de Loja

Modalidad Abierta y a Distancia



Evaluación de la Seguridad en Sistemas de Tecnologías de la Información

Guía didáctica

Facultad de Ingenierías y Arquitectura

Departamento de Ciencias de la Computación y Electrónica

Evaluación de la Seguridad en Sistemas de Tecnologías de la Información

Guía didáctica

Carrera	PAO Nivel
▪ <i>Tecnologías de la Información</i>	IX

Autores:

Jaramillo Campoverde Byron Gustavo
Aguilar Mora Carlos Darwin



SIST_5018

Asesoría virtual
www.utpl.edu.ec

Universidad Técnica Particular de Loja

Evaluación de la Seguridad en Sistemas de Tecnologías de la Información

Guía didáctica

Jaramillo Campoverde Byron Gustavo

Aguilar Mora Carlos Darwin

Diagramación y diseño digital:

Ediloja Cía. Ltda.

Telefax: 593-7-2611418.

San Cayetano Alto s/n.

www.ediloja.com.ec

edilojacialtda@ediloja.com.ec

Loja-Ecuador

ISBN digital - 978-9942-39-515-3



**Reconocimiento-NoComercial-CompartirIgual
4.0 Internacional (CC BY-NC-SA 4.0)**

Usted acepta y acuerda estar obligado por los términos y condiciones de esta Licencia, por lo que, si existe el incumplimiento de algunas de estas condiciones, no se autoriza el uso de ningún contenido.

Los contenidos de este trabajo están sujetos a una licencia internacional Creative Commons – **Reconocimiento-NoComercial-CompartirIgual 4.0 (CC BY-NC-SA 4.0)**. Usted es libre de **Compartir – copiar y redistribuir el material en cualquier medio o formato**. **Adaptar – remezclar, transformar y construir a partir del material citando la fuente, bajo los siguientes términos:** **Reconocimiento-** debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante. **No Comercial-no puede hacer uso del material con propósitos comerciales.** **Compartir igual-Si remezcla, transforma o crea a partir del material, debe distribuir su contribución bajo la misma licencia del original.** No puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia. <https://creativecommons.org/licenses/by-nc-sa/4.0/>

Índice

1. Datos de información.....	8
1.1. Presentación de la asignatura	8
1.2. Competencias genéricas de la UTPL.....	8
1.3. Competencias específicas de la carrera.....	8
1.4. Problemática que aborda la asignatura	9
2. Metodología de aprendizaje.....	9
3. Orientaciones didácticas por resultados de aprendizaje.....	11
Primer bimestre.....	11
Resultado de aprendizaje 1.....	11
Contenidos, recursos y actividades de aprendizaje.....	12
Semana 1	12
Unidad 1. Introducción a la Seguridad en Sistemas de Información	12
1.1. Conceptos de seguridad informática	12
1.2. Amenazas, ataques y activos.....	15
Actividades de aprendizaje recomendadas	19
Semana 2	20
1.3. Requisitos funcionales de seguridad	20
1.4. Principios fundamentales del diseño de seguridad.....	21
1.5. Estrategias de seguridad informática.....	21
Actividades de aprendizaje recomendadas	24
Autoevaluación 1	25
Resultado de aprendizaje 2.....	29
Contenidos, recursos y actividades de aprendizaje.....	29
Semana 3	29
1.6. Confidencialidad con cifrado simétrico.....	30
1.7. Autenticación de mensajes y funciones hash	34
Actividades de aprendizaje recomendadas	39

Semana 4	39
1.8. Cifrado de clave pública	39
1.9. Firmas digitales y gestión de claves.....	43
Actividad de aprendizaje recomendada	46
Autoevaluación 2	47
Resultado de aprendizaje 3.....	50
Contenidos, recursos y actividades de aprendizaje.....	50
Semana 5	50
Unidad 2. Autenticación de usuarios.....	51
2.1. Principios de autenticación digital de usuarios.....	51
2.2. Autenticación basada en contraseña	56
Semana 6	61
2.3. Autenticación basada en token.....	61
2.4. Autenticación biométrica	63
2.5. Autenticación de usuario remoto.....	65
Actividad de aprendizaje recomendada	66
Autoevaluación 3	67
Resultado de aprendizaje 4.....	70
Contenidos, recursos y actividades de aprendizaje.....	70
Semana 7	70
Unidad 3. Control de acceso	70
3.1. Principios de control de acceso.....	70
3.2. Sujetos, objetos y derechos de acceso	74
3.3. Control de acceso discrecional (DAC)	75
3.4. Ejemplo: control de acceso a archivos UNIX	77
3.5. Control de acceso basado en roles (RBAC).....	78
Actividad de aprendizaje recomendada	80
Autoevaluación 4	81
Actividades finales del bimestre.....	84

Semana 8	84
Segundo bimestre	85
Resultado de aprendizaje 5.....	85
Contenidos, recursos y actividades de aprendizaje.....	85
Semana 9	85
Unidad 4. Seguridad de la base de datos y del centro de datos.....	85
4.1. La necesidad de seguridad en las bases de datos	85
4.2. Sistemas de gestión de bases de datos.....	87
Semana 10	89
4.3. Bases de datos relacionales	89
4.4. Ataques de SQL injection (SQLi)	93
Actividad de aprendizaje recomendada	97
Autoevaluación 5	98
Resultado de aprendizaje 6 y 7	101
Contenidos, recursos y actividades de aprendizaje.....	101
Semana 11	101
Unidad 5. Software malicioso	101
5.1. Tipos de software malicioso	101
5.2. Amenaza persistente avanzada.....	105
5.3. Propagación - Contenido infectado – Virus.....	106
Semana 12	108
5.4. Propagación - Explotación de vulnerabilidades – Gusanos	108
5.5. Propagación - Ingeniería social - Correo electrónico no deseado, troyanos.....	112
Actividad de aprendizaje recomendada	117
Autoevaluación 6	118
Resultado de aprendizaje 8.....	121
Contenidos, recursos y actividades de aprendizaje.....	121

Semana 13	121
 Unidad 6. Seguridad del Software	121
6.1. Problemas de seguridad del Software.....	121
6.2. Manejo de la entrada del programa.....	125
6.3. Escritura de código de programa seguro	128
Semana 14	130
6.4. Interacción con el sistema operativo y otros programas.....	130
6.5. Manejo de la salida del programa.....	132
Actividad de aprendizaje recomendada	133
Autoevaluación 7.....	135
Resultado de aprendizaje 9 y 10	138
Contenidos, recursos y actividades de aprendizaje.....	138
Semana 15	138
 Unidad 7. Análisis informático forense	138
7.1. Objetivos de la informática forense.....	138
7.2. Buenas prácticas en informática forense	140
7.3. Software usado para informática forense.....	140
Actividad de aprendizaje recomendada	141
Autoevaluación 8	142
Actividades finales del bimestre.....	145
Semana 16	145
4. Solucionario	146
5. Referencias bibliográficas	155



1. Datos de información

1.1. Presentación de la asignatura



1.2. Competencias genéricas de la UTPL

- Comportamiento ético

1.3. Competencias específicas de la carrera

- Implementar aplicaciones de baja, mediana y alta complejidad integrando diferentes herramientas y plataformas para dar solución a requerimientos de la organización.
- Asegurar la calidad tanto de los productos como de los procesos en los proyectos informáticos, utilizando buenas prácticas definidas por la industria para garantizar sistemas eficientes y negocios rentables.
- Gestionar la implementación de soluciones de negocio mediante la ejecución de proyectos de TI que cumplan adecuadamente los requisitos especificados por la organización.

- Implementar mecanismos de seguridad física y lógica en los sistemas organizacionales mediante el uso de estándares y marcos de trabajo internacionales que garanticen la correcta operación del negocio.

1.4. Problemática que aborda la asignatura

- Frente a los problemas e incidentes de seguridad que existen en los sistemas informáticos, se propone el estudio de las vulnerabilidades y las alternativas y técnicas de solución.
- Algunas organizaciones carecen o desconocen la importancia de una Política de Seguridad Informática, se realizará un estudio y planteará métodos para el desarrollo de esta; así mismo, se describe el papel del usuario en el aseguramiento de la información y cómo encaja en el plan general de seguridad de la información.
- Frente a un ataque consumado, es importante conocer los métodos de análisis que propone la ciencia forense digital.



2. Metodología de aprendizaje

Las metodologías de aprendizaje que se utilizarán en el tratamiento de la asignatura de Evaluación de la Seguridad en Sistemas de Tecnologías de la Información del primero y segundo bimestre se enfocan en el estudio de casos y en el aprendizaje basado en TIC.

A través de los estudios de casos, permitirá al estudiante adquirir capacidades de análisis y reflexión sobre la seguridad informática, para construir su propio aprendizaje en un contexto que los aproxima a su entorno actual. Para la comprensión de los conceptos de la seguridad en los sistemas y tecnologías de la información, se encuentra disponible en el texto básico algunos casos prácticos que servirán de análisis según el tema de estudio, además, existen preguntas y problemas de reflexión, que permitirán al estudiante, junto con el apoyo del tutor, reflexionar sobre situaciones reales en el contexto de la seguridad informática.

Por otro lado, se utilizará el aprendizaje basado en TIC, pues durante el desarrollo de la asignatura se emplearán algunas herramientas tecnológicas y juegos didácticos que permitirán al estudiante comprender la asignatura de forma diferente e interactiva. También se empleará para la actividad síncrona el uso de varias plataformas para retroalimentar el/los tema(s) abordado(s).



3. Orientaciones didácticas por resultados de aprendizaje



Primer bimestre

Resultado de aprendizaje 1

- Explica los problemas de seguridad clave desde una perspectiva de administración de sistemas.

En la actualidad, las distintas organizaciones están expuestas a diferentes riesgos en su operación, por lo cual es necesario contar con herramientas, infraestructura y procesos que permitan la continuidad del negocio; dentro de los mecanismos de mitigación de riesgos, la Seguridad en Sistemas y Tecnologías de la Información es la base para alcanzar ese objetivo de continuidad operacional de una empresa.

El estudio de la Unidad 1, inicia con el abordaje de aspectos generales y la relevancia de conocer de forma general los conceptos de Seguridad en Sistemas de Información. En esencia, la seguridad informática se ocupa de activos relacionados con la tecnología que están sujetos a una variedad de amenazas y, para los cuales se toman varias medidas para proteger esos activos. Por lo tanto, la sección de esta unidad proporciona una descripción general de las categorías de activos relacionados a la tecnología que, usuarios y administradores del sistema desean preservar y proteger, y una mirada a las diversas amenazas y ataques que se pueden realizar sobre dichos activos. Luego, se examinan las medidas que se pueden tomar para hacer frente a tales amenazas y ataques, esto lo hacemos desde tres puntos de vista diferentes, en los puntos 1.3 a 1.5. A continuación, exponemos en términos generales una estrategia de seguridad informática.

El enfoque de esta unidad está en dar respuesta a las siguientes preguntas:

1. ¿Qué activos necesitamos proteger?
2. ¿Cómo se ven amenazados esos bienes?
3. ¿Qué podemos hacer para contrarrestar esas amenazas?



Semana 1

Preste mucha atención al contenido de la unidad 1, qué le servirá de base para comprender las generalidades de la Seguridad en Sistemas de Información y la importancia de su enfoque para el desarrollo de sistemas.

Unidad 1. Introducción a la Seguridad en Sistemas de Información

1.1. Conceptos de seguridad informática



Seguridad Informática: Son las medidas y controles que aseguran la confidencialidad, integridad y, la disponibilidad de los activos del sistema de información, esto incluye hardware, software, firmware e información que se procesa, almacena y comunica (Stallings, 2018).

Dada la relevancia de conocer los conceptos básicos de seguridad informática, lo invito a revisar el capítulo 1 del texto básico, donde se expone de forma detallada cada uno de los conceptos que serán de apoyo para la comprensión de los temas que revisaremos.

Luego de la lectura realizada, es importante que se identifique los tres objetivos clave de la seguridad informática, le invito a revisar el siguiente recurso interactivo sobre el resumen de los objetivos clave de la seguridad informática:

Objetivos clave de la seguridad informática

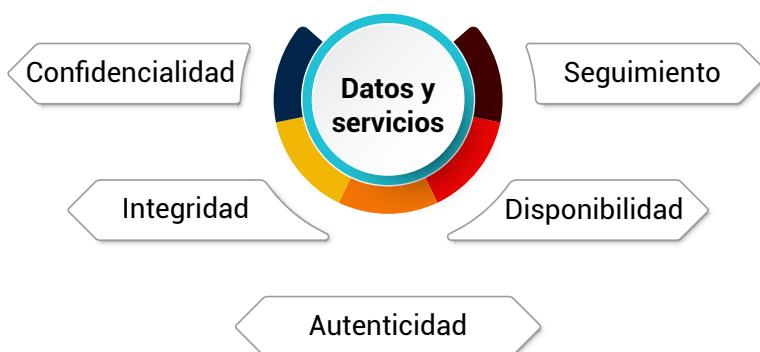
Como se puede evidenciar, estos tres conceptos forman lo que a menudo se denomina “la triada CIA” – Confidencialidad, Integridad y Disponibilidad (por sus siglas en inglés) y estos son los definidos como los objetivos fundamentales de la seguridad, tanto para los datos como para la información. Aunque el uso de la triada CIA para definir los objetivos de la seguridad está bien establecido, existen sugerencias de incluir conceptos

adicionales para presentar una imagen completa, en la figura 1, se incluye a dos de los más comúnmente mencionados:

- **Autenticidad:** La propiedad de ser único y poder ser verificado; confianza en la validez de una transmisión, un mensaje o el propio emisor. Significa verificar que los usuarios son quienes dicen ser y, que cada entrada que llega al sistema, proviene de una fuente confiable.
- **Seguimiento:** Los sistemas deben mantener registros de sus actividades para permitir el análisis forense posterior, rastrear brechas de seguridad o ayudar en disputas de transacciones.

Figura 1.

Imagen completa, objetivos clave de la seguridad informática



Dada esta revisión de conceptos generales, lo invito a reflexionar en los siguientes cuestionamientos antes de continuar con el siguiente tema:

1. ¿Cuál sería la definición de una pérdida de seguridad en cada categoría?
2. ¿Existirá alguna priorización dentro de las categorías revisadas o, todas estarán al mismo nivel?

Luego de dar respuesta a las preguntas planteadas, pasaremos a realizar un análisis más detallado de otros conceptos y, la relación que existe entre ellos.

Conceptos complementarios en la Seguridad Informática

De los conceptos expuestos en los puntos anteriores, vamos a complementar con terminología que será útil a lo largo de nuestro estudio,

la tabla 1 define los términos y, la figura 2 indica la relación entre estos términos.

Tabla 1.

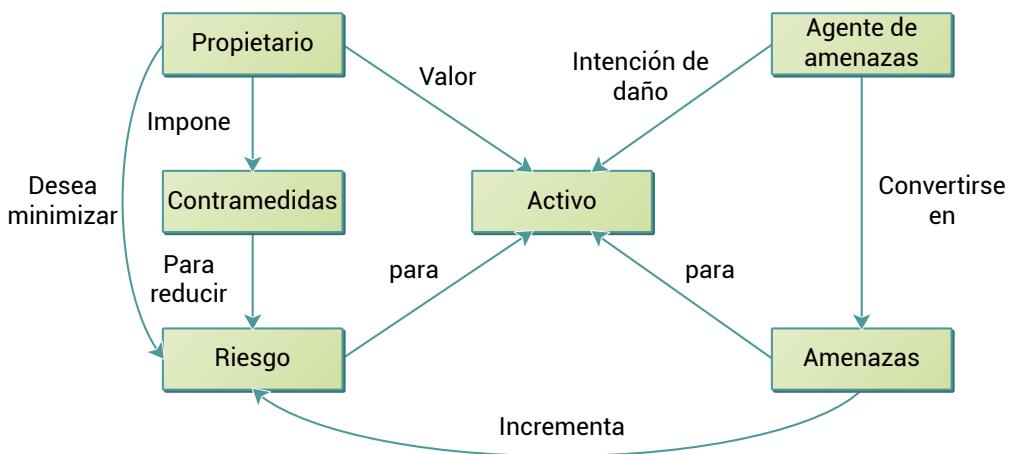
Conceptos complementarios de Seguridad Informática (Stallings, 2018).

Concepto	Definición
Adversario (agente de amenaza)	Individuo, grupo, organización o gobierno que realiza o tiene la intención de realizar actividades perjudiciales.
Ataque	Cualquier tipo de actividad maliciosa que intente recopilar, interrumpir, negar, degradar o destruir el sistema de información, los recursos o la información misma.
Riesgo	Medida en que una entidad se ve amenazada por una circunstancia o evento potencial y, por lo general, se mide en función de: <ol style="list-style-type: none">1. Los impactos adversos que surgirían si la circunstancia o evento ocurre; y2. La probabilidad de ocurrencia.
Política de seguridad	Un conjunto de criterios para la prestación de servicios de seguridad.
Amenaza	Cualquier circunstancia o evento con el potencial de afectar negativamente las operaciones de la organización (incluida la misión, las funciones, la imagen o la reputación), los activos de la organización, las personas, otras organizaciones o la Nación a través de un sistema de información a través del acceso, la destrucción o la divulgación no autorizada; modificación de información, y/o denegación de servicio.
Vulnerabilidad	Debilidad en un sistema de información, procedimientos de seguridad del sistema, controles internos o implementación que podría ser explotado por una fuente de amenaza.
Recurso del sistema (activo)	Una aplicación principal, un sistema de soporte general, un programa de alto impacto, una planta física, un sistema de misión crítica, personal, equipo o un grupo de sistemas lógicamente relacionado.

Es importante no solo comprender la terminología expuesta, sino la relación que existe entre estos conceptos, la figura 2, muestra lo indicado.

Figura 2.

Relaciones y conceptos de seguridad informática (Stallings, 2018)



La base de la seguridad de la información está definida en la relación de estos conceptos, comprenderlos le ayudará a interpretar de mejor manera la interacción de las diferentes actividades a desarrollar.

El siguiente tema que abordaremos está orientado a detallar los conceptos de amenazas, ataques y activos que necesitamos salvaguardar sobre los objetivos de seguridad revisados.

1.2. Amenazas, ataques y activos

En este punto de nuestro estudio, vamos a revisar de manera más detallada los temas relacionados con las amenazas, los ataques y activos de una institución o servicio. En primera instancia, estudiaremos los tipos de amenazas a la seguridad que deben abordarse y, a continuación, daremos algunos ejemplos de los tipos de amenazas que se aplican a las diferentes categorías de activos.

Amenazas y ataques

La tabla 2, basada en RFC 4949, describe cuatro tipos de amenazas y enumera los tipos de ataques que se desencadenan por cada amenaza presentada.

Tabla 2.Amenazas y tipos de ataques (*Adaptado RFC 4949*)

Amenaza	Tipos de ataques desencadenados
Divulgación no autorizada Circunstancia o evento por el cual una entidad obtiene acceso a los datos para los cuales la entidad no está autorizada.	Exposición: los datos confidenciales se divulgan directamente a una entidad. Intercepción: una entidad no autorizada accede directamente a datos confidenciales que viajan entre fuentes y destinos autorizados. Inferencia: Una acción de amenaza mediante la cual una entidad no autorizada accede indirectamente a datos confidenciales (pero no necesariamente a los datos contenidos en la comunicación) razonando a partir de características o subproductos de las comunicaciones. Intrusión: una entidad no autorizada obtiene acceso a datos confidenciales eludiendo las protecciones de seguridad de un sistema
Fraude Una circunstancia o evento que puede resultar en una entidad autorizada al recibir datos falsos y validarlos como correctos.	Enmascaramiento: Una entidad no autorizada obtiene acceso a un sistema o realiza un acto malicioso haciéndose pasar por una entidad autorizada. Falsificación: Los datos falsos engañan a una entidad autorizada. Repudio: Una entidad engaña a otra negando falsamente responsabilidad por un acto
Interrupción Una circunstancia o evento que interrumpe o impide la correcta operación de los servicios del sistema y sus funciones.	Incapacidad: Impide o interrumpe el funcionamiento del sistema al deshabilitar un componente del mismo. Corrupción: Altera indeseablemente el funcionamiento del sistema al modificar funciones o datos del sistema. Obstrucción: Acción de amenaza que interrumpe la entrega del sistema o sus servicios al dificultar su funcionamiento.
Usurpación Circunstancia o evento que resulta en el control de los servicios o funciones del sistema por parte de una entidad no autorizada.	Apropiación indebida: Una entidad que asume operaciones lógicas o control físico de un recurso del sistema. Mal uso: Hace que un componente del sistema realice una función o servicio que es perjudicial para la seguridad del sistema

Amenazas y activos

Los activos de un sistema informático se pueden clasificar en hardware, software, datos y líneas o redes de comunicación. La tabla 3 describe

brevemente estas cuatro categorías y se expone algunos ejemplos relacionados a los conceptos de integridad, confidencialidad y disponibilidad.

Tabla 3.

Activos y ejemplos de afectación a los conceptos claves de seguridad informática

Activo	Disponibilidad	Confidencialidad	Integridad
Hardware	El equipo es robado o inhabilitado, por lo que se niega el servicio.	Un dispositivo de memoria extraíble es robado.	
Software	Programas son borrados negando el acceso a los usuarios.	Se realizan copias no autorizadas de un software.	Un programa es modificado provocando fallas en la ejecución de las tareas.
Datos	Archivos son borrados negando el acceso a los usuarios.	Se realiza lectura no autorizada de datos.	Archivos existentes son modificados o, se crean nuevos archivos.
Líneas o redes de comunicación	Mensajes son destruidos o borrados	Patrones de tráfico son observados por entidades no autorizadas.	Los mensajes son modificados, retardados, reordenados o duplicados.
	Líneas de comunicación o redes no están disponibles.		Mensajes falsos creados.

Como se pudo apreciar, la clasificación de los activos de un sistema informático son hardware, software, datos y líneas o redes de comunicación, a continuación, una leve explicación de cada uno:

Hardware: Una gran amenaza para el hardware de los sistemas informáticos es la disponibilidad. El hardware es el más vulnerable a los ataques y el menos susceptible a los controles automatizados. Las amenazas incluyen daños accidentales y deliberados al equipo, así como hurto.

Software: El software incluye el sistema operativo, las utilidades y los programas de aplicación. Una amenaza clave para el software es un ataque a la disponibilidad. El software, especialmente el software de aplicación, suele ser fácil de eliminar. El software también puede ser alterado o dañado para dejarlo inservible. La gestión cuidadosa de la configuración del software que incluye la realización de copias de seguridad de la versión más

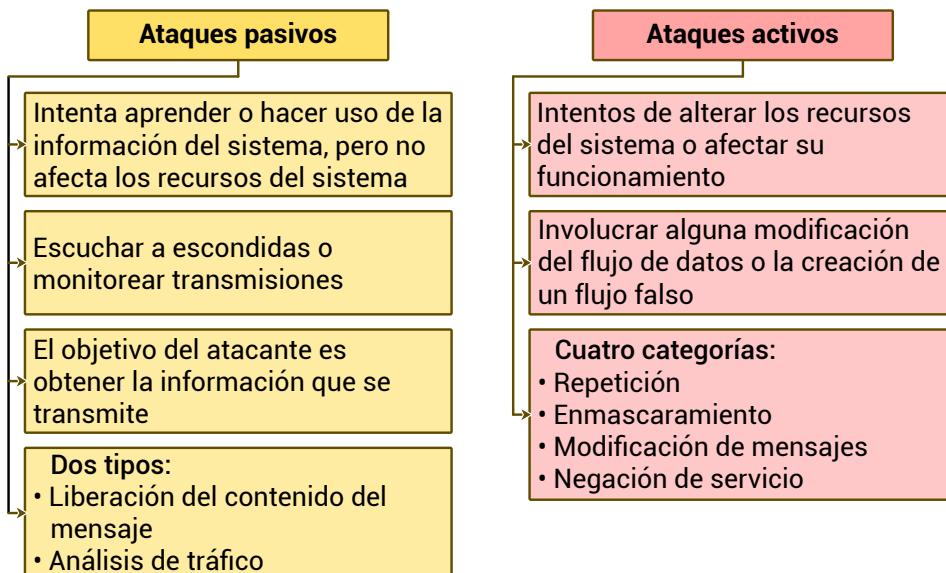
reciente, puede mantener una alta disponibilidad. Un problema más difícil de tratar es la modificación del software que da como resultado un programa que aún funciona, pero que se comporta de manera diferente que antes, lo cual es una amenaza para la integridad/autenticidad. Los virus informáticos y los ataques relacionados entran en esta categoría.

Datos: Un problema mucho más generalizado es la seguridad de los datos, que involucra archivos y otras formas de datos controlados por individuos, grupos y organizaciones empresariales. Las preocupaciones de seguridad con respecto a los datos son amplias y abarcan la disponibilidad, privacidad y la integridad. En el caso de la disponibilidad, la preocupación es la destrucción de los archivos de datos, que puede ocurrir de forma accidental o malintencionada. La preocupación obvia con la privacidad, es la lectura no autorizada de archivos de datos o bases de datos, y esta área ha sido objeto de quizás más investigación y esfuerzo que cualquier otra área de la seguridad informática. Finalmente, la integridad de los datos es una preocupación importante en la mayoría de las instalaciones. Las modificaciones a los archivos de datos pueden tener consecuencias que van desde menores hasta desastrosas.

Líneas o redes de comunicación: Los ataques a la seguridad de la red se pueden clasificar como ataques pasivos y ataques activos. Un ataque pasivo intenta aprender o hacer uso de la información del sistema, pero no afecta los recursos del sistema. Un ataque activo intenta alterar los recursos del sistema o afectar su funcionamiento. La figura 3 nos brinda un resumen de estos conceptos:

Figura 3.

Ataques activos y pasivos



Estimado/a estudiante, le animo a completar las actividades recomendadas descritas a continuación.



Actividades de aprendizaje recomendadas

Para apoyar a la comprensión del tema estudiado, lo invito a participar de la siguiente actividad denominada “*Amenazas y tipos de ataques*”, y con base en la amenaza expuesta, identificar el tipo de ataque que corresponda. El desarrollo de esta actividad le permitirá diferenciar según el tipo de amenaza, los ataques correspondientes y retroalimentar lo estudiado.

[Amenazas y tipos de ataques](#)

Visualice el siguiente recurso denominado “*Alcance de la seguridad informática*”, donde se expone de forma sintetizada las generalidades de la seguridad informática. En esta infografía se plasma un resumen del tema estudiado que facilitará su comprensión.

[Alcance de la seguridad informática](#)



En esencia, la seguridad informática se ocupa de los activos relacionados con la informática, al estar estos sujetos a una variedad de amenazas, es menester se tomen diversas medidas de protección.



Semana 2

1.3. Requisitos funcionales de seguridad

En este punto de nuestro estudio, es importante mencionar que existen algunos estándares que definen los requisitos funcionales de seguridad informática que deben cumplir las organizaciones, para nuestra referencia, revisaremos el documento federal de procesamiento de la información (Federal Information Processing Standard – FIPS); este es un estándar federal obligatorio desarrollado por el Instituto Nacional de Estándares y Tecnologías (*National Institute of Standards and Technology – NIST*) que incluye numerosos requisitos de seguridad, en este capítulo revisaremos los que presentan mayor relevancia según su categoría. Para un mayor detalle, lo invito a revisar la Tabla 1.4 del texto base (Capítulo 1, sección 1.3).

Requisitos funcionales de seguridad

Como parte de la relevancia del cumplimiento de estos estándares, es importante mencionar que algunos proveedores de servicios en la nube como, Amazon y Microsoft están optando por implementar los requisitos indicados en el NIST, e incluso han obtenido la certificación NIST 800 - 171 (Navarro, 2020).

Para fortalecer su aprendizaje sobre los requisitos funcionales de seguridad, lo invito a complementar su estudio revisando la siguiente página oficial del [Instituto Nacional de Estándares y Tecnología – NIST \(EEUU\)](#), aquí encontrará información relevante al tema revisado.

Luego de conocer sobre los requisitos funcionales de seguridad, es conveniente reflexionar sobre las siguientes preguntas:

1. ¿Cree usted que la aplicación de las recomendaciones listadas en el FIPS – 200 son aplicables a las instituciones de nuestro país?

2. ¿Puede identificar cuál de los requerimientos listados usted recomendaría implementar en la institución a la que pertenece?

1.4. Principios fundamentales del diseño de seguridad

Una vez que, con el apoyo del autor de nuestro texto base, hemos definido algunos requisitos fundamentales de seguridad, es momento de revisar los principios fundamentales del diseño de seguridad, en este punto, es importante mencionar que, a pesar de los diversos esfuerzos de investigación y desarrollo, no ha sido posible desarrollar técnicas de diseño e implementación que permitan descartar de manera sistemática los errores o defectos de seguridad sobre los sistemas, en este sentido resulta útil definir un conjunto de principios de diseño acordados de forma global, que permitan guiar la definición y desarrollo de los mecanismos de protección. La Seguridad Nacional de los Estados Unidos y el Departamento de Seguridad Nacional de los Estados Unidos, enumeran los siguientes como principios fundamentales de diseño de seguridad (Stallings, 2018).

Veamos el siguiente recurso acerca de los principios de seguridad:

[Principios fundamentales de seguridad](#)



En el capítulo 1 del texto básico, se encuentra la descripción y ejemplos de cada principio revisado, por lo que es importante revisar esta sección para su mejor comprensión.

1.5. Estrategias de seguridad informática

Concluimos esta unidad realizando una revisión general a la estrategia global de seguridad informática propuesta por Lampson (Lampson, 2004), la misma incluye tres aspectos:

- Especificación/política: ¿Qué debe hacer el esquema de seguridad?
- Implementación/mecanismos: ¿Cómo lo hace?
- Corrección/garantía: ¿Funciona realmente?

A continuación, la figura 4 representa los 4 pasos propuestos para el desarrollo de una estrategia de seguridad informática.

Figura 4.
Estrategia de seguridad informática

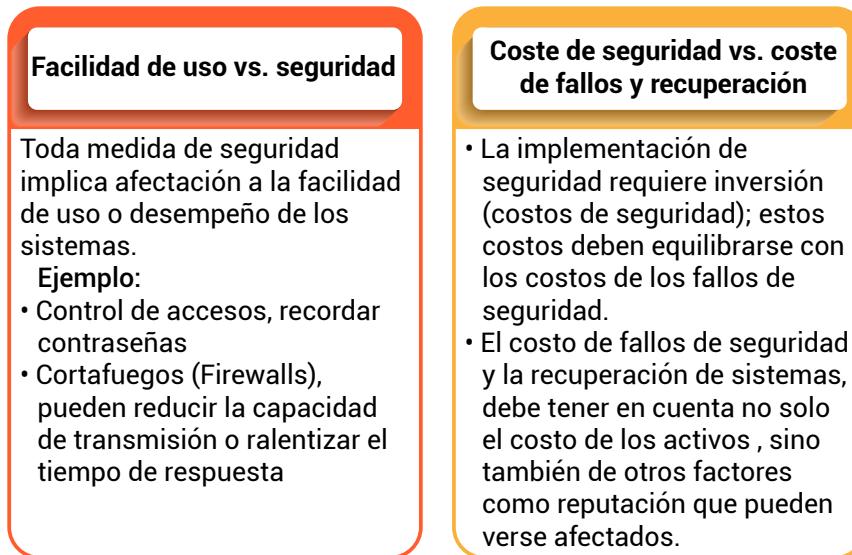


Como indica la figura 4, el primer paso para diseñar servicios y mecanismos de seguridad es desarrollar una **política de seguridad**. El área responsable de la seguridad informática utiliza el término política de seguridad de varias maneras; para una definición formal nos basaremos en el RFC 4949, el mismo que indica a una política de seguridad como una declaración formal de reglas y prácticas que especifican o regulan cómo un sistema u organización proporciona servicios de seguridad para proteger los recursos sensibles y críticos del sistema. Esta política de seguridad formal se presta a ser aplicada por los controles técnicos del sistema, así como sus controles operativos y de gestión.

Una vez iniciado este proceso de desarrollo, es importante tomar en cuenta las consideraciones importantes de una política de seguridad; la figura 5 representa lo indicado.

Figura 5.

Consideraciones de una Política de Seguridad



Con base en la figura 5 y lo relacionado con la Política de Seguridad, podemos indicar que esta es una decisión empresarial, posiblemente influenciada por requisitos legales.

Luego de la Política de Seguridad, pasamos a la siguiente etapa, **Implementación de la Seguridad**; esta fase describe cuatro líneas de actuación complementarias.

- a. Prevención: Un esquema de seguridad ideal es aquel en el que ningún ataque tiene éxito; aunque esto no es práctico en todos los casos, hay una amplia gama de amenazas en las que la prevención es un objetivo razonablemente alcanzable.
- b. Detección: En algunos casos, la protección absoluta no es factible, pero, si lo es detectar los ataques. Por ejemplo, existen sistemas de detección de intrusos diseñados para detectar la presencia de personas no autorizadas en un sistema.
- c. Respuesta: Si los mecanismos de seguridad detectan un ataque en curso, como un ataque de denegación de servicio, el sistema puede ser capaz de responder de tal manera que detenga el ataque y evitar más daños.

- d. Recuperación: Un ejemplo de recuperación es el uso de sistemas de copia de seguridad, de modo que, si la integridad de los datos se ve comprometida, se puede volver a cargar una copia anterior y correcta de los datos.

Como siguiente paso para el desarrollo de una estrategia de seguridad informática, tenemos el **Aseguramiento y Evaluación**; el aseguramiento es un atributo de un sistema de información que proporciona motivos para confiar en que el sistema funciona de manera que se aplica la política de seguridad. Esto abarca tanto el diseño como la implementación del sistema. Por lo tanto, el aseguramiento se ocupa de las preguntas:

- “¿El diseño del sistema de seguridad cumple sus requisitos?”
- “¿La implementación del sistema de seguridad cumple sus especificaciones?”

Por otro lado, la evaluación es el proceso de examinar un producto o sistema informático con respecto a ciertos criterios. La evaluación implica la realización de pruebas y también puede implicar técnicas analíticas o matemáticas formales. El eje central de los trabajos en este ámbito es el desarrollo de criterios de evaluación que puedan aplicarse a cualquier sistema de seguridad (abarcando servicios y mecanismos de seguridad) y que están ampliamente respaldados para realizar comparaciones de productos.

¡Excelente! Hemos finalizado el estudio de la primera unidad, avancemos a la unidad 2, pero antes, lo invito a desarrollar la actividad propuesta.



Actividades de aprendizaje recomendadas

Recuerde, en esta semana hemos revisado los Requisitos Funcionales de Seguridad, luego avanzamos con los Principios fundamentales del diseño de seguridad, para concluir con una estrategia de seguridad informática; sin embargo, con el objetivo de revisar cada uno de estos conceptos, le invito a desarrollar el siguiente recurso interactivo denominado *“Requisitos, fundamentos y estrategias de Seguridad Informática”*.

[Requisitos, fundamentos y estrategias de Seguridad Informática](#)

Estimado/a estudiante, le invito a que realice la autoevaluación para comprobar sus conocimientos.



Autoevaluación 1

Lea atentamente las preguntas propuestas con relación a los conceptos de seguridad informática y seleccione la opción de respuesta correcta.

1. ¿Qué entiende por la tríada CIA (siglas en inglés)?
 - a. Confidencialidad, Integridad y Disponibilidad.
 - b. Corrección, Integridad y Disponibilidad.
 - c. Computación, Inteligencia y Aplicaciones.
 - d. Correlación, Integridad y Aplicabilidad.
2. ¿Qué es la seguridad informática?
 - a. Son las medidas y controles que aseguran la confidencialidad, integridad y la disponibilidad de los activos del sistema de información, esto incluye *hardware*, *software*, *firmware* e información que se procesa, almacena y comunica.
 - b. Son las medidas y controles que aseguran la comunicación del sistema de información, esto incluye, técnicos, *hardware*, *software*, *firmware*.
 - c. La seguridad informática es el proceso mediante el cual se define una política de seguridad con el objetivo de prevenir la fuga de información de una empresa.
 - d. Son los controles y actualizaciones que los técnicos de sistemas realizan de manera organizada, planificada y controlada a los sistemas críticos de la organización.

3. ¿Cuál de las siguientes son consideradas como amenazas a los activos de una empresa? (escoja cuatro).
 - a. Divulgación no autorizada.
 - b. Fraude.
 - c. Interrupción.
 - d. Usurpación.
 - e. Virus.
 - f. Intrusión.
 - g. Denegación de servicio
4. ¿Cuál de los siguientes tipos de ataques es considerado como ataque pasivo?
 - a. Intento de alterar los recursos del sistema o afectar su funcionamiento.
 - b. Involucrar alguna modificación del flujo de datos o la creación de un flujo falso.
 - c. Enmascaramiento de información.
 - d. Escuchar a escondidas o monitorear transmisiones.
5. Del siguiente listado, seleccione 3 requisitos funcionales de seguridad de la información.
 - a. Control de acceso.
 - b. Concienciación y formación.
 - c. Identificación y autenticación.
 - d. Implementación y mecanismos.
 - e. Corrección/garantía.

6. Dentro de las etapas de implementación de una estrategia de seguridad, la fase de aseguramiento tiene relación con:
 - a. Proceso de validación de un producto o sistema informático con respecto a determinados criterios.
 - b. La realización de pruebas y también puede implicar técnicas analíticas o matemáticas formales.
 - c. Declaración formal de normas y prácticas que especifican o regulan cómo un sistema u organización proporciona servicios de seguridad.
 - d. Atributo de un sistema de información que proporciona motivos para confiar en que el sistema funciona de manera que la política de seguridad del sistema se cumple.
7. La política de seguridad definida por la organización, ¿es aplicable únicamente al área de sistemas?
 - a. Sí.
 - b. No.
8. Dentro de los requisitos funcionales de seguridad, el control de accesos se refiere a:
 - a. Proporcionar capacitación de sensibilización sobre la seguridad para reconocer y reportar posibles indicadores de información privilegiada amenaza.
 - b. Seguimiento, revisión, aprobación/desaprobación y auditoría de cambios en los sistemas de información organizacional.
 - c. Realizar mantenimiento en los sistemas de información de la organización.
 - d. Emplear el principio de los privilegios mínimos, incluso para funciones de seguridad específicas y cuentas.

9. Indique, verdadero o falso, ¿la integridad de los datos asegura que la información y los programas se cambien solo de una manera especificada y autorizada?
- Verdadero.
 - Falso.
10. _____ asegura que las personas controlan o influyen en qué información relacionada con ellos se puede recopilar y almacenar, y por quién y a quién se puede divulgar esa información.
- Disponibilidad.
 - Integridad del sistema.
 - Privacidad.
 - Integridad de datos.

[Ir al solucionario](#)

 *Si al contestar la autoevaluación su respuesta tuvo resultados positivos ¡FELICITACIONES, SIGA ADELANTE!, caso contrario revise nuevamente el contenido de los ítems errados, para reforzar su aprendizaje. Recuerde que en caso de tener alguna inquietud consulte con el profesor tutor*

Resultado de aprendizaje 2

- Describe los métodos para ocultar información en un sistema de archivos.

El uso de los algoritmos criptográficos son un elemento importante en muchos servicios y aplicaciones de seguridad informática. Esta unidad ofrece una visión general de los distintos tipos de algoritmos, además, se realizará una discusión de su aplicabilidad. Para cada tipo de algoritmo, presentaremos los algoritmos estandarizados más importantes de uso común.

El estudio de la Unidad 2, inicia con la revisión del cifrado simétrico, el cual, se utiliza en la mayor variedad de contextos, principalmente para proporcionar confidencialidad; a continuación, examinamos las funciones hash seguras y discutimos su uso en la autenticación de mensajes. En la sección 2.3 examinaremos el cifrado de clave pública, también conocido como cifrado asimétrico, para luego finalizar esta unidad con el análisis de las dos aplicaciones más relevantes del cifrado de clave pública, la firma digital y la gestión de claves. En el caso de las firmas digitales, el cifrado asimétrico y las funciones hash seguras, se combinan para producir una herramienta extremadamente útil.

Por último y, como parte de nuestra revisión práctica revisaremos un Caso de estudio de un área de aplicación de los algoritmos criptográficos, al examinar el cifrado de datos almacenados.

Preste mucha atención al contenido de la unidad 2, "Herramientas Criptográficas", le servirá para comprender las generalidades de la criptografía, base de la Seguridad en Sistemas de Información.

Contenidos, recursos y actividades de aprendizaje



Semana 3

1.6. Confidencialidad con cifrado simétrico

La técnica universal para brindar confidencialidad (acceso a la información solo por el personal autorizado) a los datos transmitidos o almacenados es el cifrado simétrico, esta sección presenta el concepto básico del cifrado simétrico, para luego realizar una descripción general de los dos algoritmos de cifrado simétrico más importantes: el Estándar de cifrado de datos (DES) y el Estándar de cifrado avanzado (AES), que son algoritmos de cifrado de bloques. Finalmente, esta sección introduce el concepto de algoritmos de cifrado de flujo simétrico.

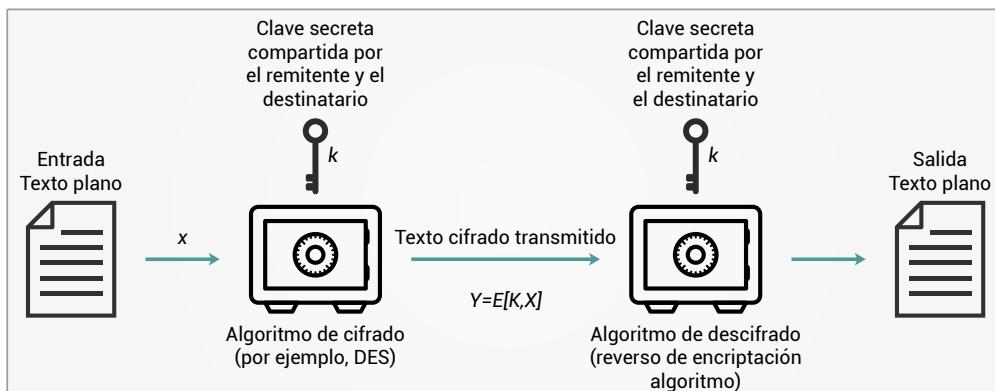
Cifrado simétrico

El cifrado simétrico, también conocido como cifrado convencional o de clave única, era el único tipo de cifrado utilizado antes del advenimiento del cifrado de clave pública a fines de la década de 1970. Innumerables personas y grupos, desde Julio César hasta la fuerza submarina alemana y los usuarios diplomáticos, militares y comerciales de hoy en día han utilizado el cifrado simétrico para las comunicaciones secretas. Todavía se usa más ampliamente de los dos tipos de cifrado.

La figura 6 indica los cinco elementos requeridos para la ejecución de un cifrado simétrico.

Figura 6.

Elementos del cifrado simétrico (Stallings, 2018).



Como se indica, en la figura 6 podemos apreciar los elementos principales del cifrado simétrico, en resumen:

- a. *Texto plano*, este es el mensaje original o los datos que se introducen en el algoritmo como entrada al proceso de cifrado.
- b. *Algoritmo de cifrado*, el algoritmo de cifrado realiza varias sustituciones y transformaciones en el texto plano.
- c. *Clave secreta*, la clave secreta también se ingresa en el algoritmo de cifrado. Las sustituciones y transformaciones exactas efectuadas por el algoritmo dependen de la clave.
- d. *Texto cifrado*, este es el mensaje codificado producido como salida. Depende del texto sin formato y de la clave secreta. Para un mensaje dado, dos claves diferentes producirán dos textos cifrados diferentes.
- e. *Algoritmo de descifrado*, este es esencialmente el algoritmo de cifrado ejecutado a la inversa. Toma el texto cifrado y la clave secreta y produce el texto original.

Antes de avanzar, es importante que se plantee la siguiente pregunta.

¿Cuál sería un proceso de ataque a un esquema de clave simétrica?

Algoritmos de cifrado de bloques simétricos

El algoritmo de cifrado simétrico más utilizado es el cifrado de bloques, su funcionamiento se basa en que procesa la entrada de texto plano en bloques de tamaño fijo y, genera un bloque de texto cifrado del mismo tamaño para cada bloque de texto plano ingresado. El algoritmo trata grandes cantidades de texto más sencillo como una serie de bloques de tamaño fijo; los algoritmos simétricos más importantes, todos los cuales pertenecen al cifrado de bloques son, el Estándar de cifrado de datos (DES), triple DES y el Estándar de cifrado avanzado (AES); la tabla 4 presenta una comparación de estos algoritmos.

Tabla 4.

Comparación entre los tres algoritmos de cifrado simétrico más populares.

Característica	DES	Triple DES	AES
Tamaño de bloque de texto plano (bits)	64	64	128
Tamaño del bloque de texto cifrado (bits)	64	64	128

Característica	DES	Triple DES	AES
Tamaño de clave (bits)	56	112 o 168	128, 192 o 256

DES = Data Encryption Standard

AES = Advanced Encryption Standard

A continuación, una breve descripción y características de cada uno de estos algoritmos:

Estándar de cifrado de datos (DES). - Hasta hace poco, el esquema de cifrado más usado; está basado en el Estándar de cifrado de datos (DES) adoptado en 1977 por la Oficina Nacional de Estándares. El algoritmo en sí se conoce como algoritmo de cifrado de datos (DEA). DES toma un bloque de texto sin formato de 64 bits y una clave de 56 bits para producir un bloque de texto cifrado de 64 bits.

Las preocupaciones sobre la fuerza de DES se dividen en dos categorías:

- Preocupaciones sobre el algoritmo en sí; se refiere a la posibilidad de que el criptoanálisis sea posible explotando las características del algoritmo DES. A lo largo de los años, han existido numerosos intentos de encontrar y explotar las debilidades del algoritmo, lo que convierte a DES en el algoritmo de cifrado más estudiado que existe. A pesar de numerosos enfoques, hasta ahora nadie ha informado de una debilidad importante en DES (Stallings, 2018).
- Una preocupación más seria es la longitud de la clave; como revisamos en la tabla 5, DES propone una longitud de clave de 56 bits, es decir, hay claves posibles, que son aproximadamente claves en total; aunque esto parezca un número elevado de combinaciones, dada la velocidad de los procesadores comerciales estándar, esta longitud de clave es lamentablemente inadecuada.

Con la información que hemos revisado y, tomando en cuenta el tamaño de la clave, la tabla 5 muestra cuánto tiempo se requiere para un ataque de fuerza bruta.

Tabla 5.

Tiempo promedio para la búsqueda de claves, según su tamaño y algoritmo.

Tamaño de clave (bits)	Cifrado	Número de claves generadas	Tiempo requerido	Tiempo requerido
56	DES		1,125 años	1 hora
128	AES		años	años
168	Triple DES		años	años
192	AES		años	años
256	AES		años	años

Como se puede apreciar en la tabla 5, una sola PC puede romper DES en aproximadamente un año; si varias PC trabajan en paralelo, el tiempo se reduce drásticamente. Los tamaños de clave de 128 bits o más son efectivamente irrompibles utilizando simplemente un enfoque de fuerza bruta. Incluso si consiguiéramos acelerar el sistema de ataque por un factor de 1 billón (), se necesitarían más de 100 000 años para descifrar un código utilizando una clave de 128 bits.

Afortunadamente, hay una serie de alternativas a DES, las más importantes son triple DES y AES, discutidas a continuación.

Triple DES. - La vida de DES se amplió con el uso de triple DES (3DES), que consiste en repetir el algoritmo DES básico tres veces, utilizando dos o tres claves únicas, con un tamaño de clave de 112 o 168 bits. 3DES se estandarizó por primera vez para su uso en aplicaciones financieras en el estándar en 1985 y, 3DES se incorporó como parte del Estándar de cifrado de datos en 1999 (Stallings, 2018).

El algoritmo 3DES tiene dos ventajas que lo avalan en la continuidad de los servicios de cifrado, la primera, su clave de longitud de 168 bits, la cual supera la vulnerabilidad al ataque de fuerza bruta de DES. En segundo lugar, el algoritmo base de 3DES es el mismo que DES y, este proceso ha sido tema de escrutinio que cualquier otro algoritmo de cifrado por un largo periodo y, no se ha encontrado ningún ataque criptoanalítico efectivo que atente contra él. Ahora, es importante mencionar que existen algunos inconvenientes respecto a 3DES, uno de los principales es la lentitud en su ejecución mediante software; esta característica está relacionada con su diseño original fue desarrollado para hardware (1970). 3DES requiere tres veces el cálculo respecto a DES, por lo tanto, es más lento. Adicional,

podemos mencionar el tamaño de sus bloques, 64 bits; por razones de seguridad, es deseable tener bloques con mayor tamaño.

Estándar de Cifrado Avanzado (AES). - Debido a los inconvenientes analizados a 3DES, este no es un candidato razonable para el uso a largo plazo; como reemplazo, en 1997 se emitió una convocatoria de propuestas para un nuevo Estándar de Cifrado Avanzado (AES), que debería tener una seguridad igual o mejor que 3DES y una eficiencia significativamente mejorada.

Además de estos requisitos generales, se especificó que AES debe ser un cifrado de bloque simétrico con una longitud de bloque de 128 bits y soporte para longitudes de clave de 128, 192 y 256 bits. Los criterios de evaluación incluyeron seguridad, eficiencia computacional, requisitos de memoria, idoneidad de hardware y software y flexibilidad.

En una primera ronda de evaluación, se aceptaron 15 algoritmos propuestos, para luego reducir el campo a 5 algoritmos. Una vez completado el proceso de evaluación, se publicó el estándar final como *Advanced Encryption Standard*, noviembre de 2001 - AES. AES ahora está ampliamente disponible en productos comerciales.

1.7. Autenticación de mensajes y funciones hash

En este punto de nuestro estudio, es importante enfatizar en algo, el cifrado de los datos estudiados en el apartado anterior, protege a los activos de los ataques pasivos (escuchas); un requisito diferente y lógico, es la protección contra ataques activos (falsificación de datos y transacciones); la protección contra este tipo de ataques, se conoce como autenticación de mensajes o datos. Se define a un mensaje, archivo, documento u otra recopilación de datos como auténtico, cuando es genuino y proviene de su supuesta fuente autorizada. La autenticación de mensajes o datos es un procedimiento que permite a las partes que se comunican, verificar que los mensajes recibidos o almacenados son auténticos. Existen dos aspectos importantes de la autenticación de datos, así:

- a. Verificar que el contenido del mensaje no haya sido alterado y,
- b. Verificar la fuente de emisión del mensaje.

Adicional y, como validación complementaria, es deseable verificar la puntualidad de un mensaje (no se ha retrasado ni producido de forma

arbitraria) y, la secuencia en relación con los otros mensajes parte del flujo de la comunicación. Este tipo de características pertenecen a la autenticación de los datos, que, como recordará, revisamos en la unidad 1.

Autenticación mediante cifrado simétrico

Reflexionemos un poco sobre la autenticación y la integridad; nuestra hipótesis será que, mediante el cifrado simétrico (estudiado en el punto anterior) se puede brindar autenticación a los mensajes; pues bien, si asumimos que solo el remitente y el receptor comparten una clave (que es como debería ser), entonces solo el remitente genuino podría cifrar un mensaje con éxito para el otro participante, siempre que el receptor pueda reconocer un mensaje válido (autenticación). Además, si el mensaje incluye un código de detección de errores y un número de secuencia, el receptor tiene la seguridad de que no se han realizado alteraciones y que la secuencia es correcta (integridad). Si el mensaje también incluye una marca de tiempo, el receptor tiene la seguridad de que el mensaje no se ha retrasado más de lo que normalmente se espera para tránsito de la red (autenticación).

Aunque este análisis pareciera completo y aceptaría como válida a la hipótesis planteada, veremos que, el cifrado simétrico por sí solo no es una herramienta adecuada para la autenticación de datos. Para demostración, simplemente analicemos el modo de cifrado por bloques, si un atacante reordena los bloques de texto cifrado, entonces cada bloque se descifrará con éxito (integridad). Sin embargo, el reordenamiento puede alterar el significado de la secuencia de datos general. Aunque los números de secuencia pueden usarse en algún nivel (por ejemplo, cada paquete IP), normalmente no se asocia un número de secuencia separado con cada bloque de b-bit de texto sin formato. Por lo tanto, el reordenamiento de bloques es una amenaza (autenticación).

Autenticación de mensajes sin cifrado de mensajes

A continuación, examinamos varios enfoques para la autenticación de mensajes que no se basan en el cifrado de mensajes. En estos enfoques, se genera una etiqueta de autenticación y se agrega a cada mensaje para su transmisión. El mensaje en sí no está encriptado y se puede leer en el destino independientemente de la función de autenticación que este gestione.

Debido a que los enfoques discutidos en esta sección no cifran el mensaje, no se proporciona confidencialidad del mensaje. Como se mencionó, el cifrado de mensajes por sí solo no proporciona una forma segura de autenticación. Sin embargo, es posible combinar autenticación y confidencialidad en un solo algoritmo cifrando un mensaje, más su etiqueta de autenticación. Sin embargo, normalmente, la autenticación de mensajes se proporciona como una función separada del cifrado de mensajes. Se sugiere tres situaciones en las que es preferible la autenticación de mensajes sin confidencialidad (Stallings, 2018):

1. Asumamos que existe una serie de aplicaciones en las que se transmite el mismo mensaje a varios destinos. Es óptimo y confiable tener un solo destino responsable de monitorear la autenticidad. Por lo tanto, el mensaje se transmitirá en texto plano con una etiqueta de autenticación de mensaje asociada. El sistema responsable realiza la autenticación y, si ocurre una violación, los otros sistemas de destino son alertados por una alarma general.
2. Otro escenario posible es un intercambio en el que un lado tiene una carga pesada y no puede asignarse recursos para descifrar todos los mensajes entrantes. La autenticación se lleva a cabo de forma selectiva y, los mensajes se eligen al azar para su verificación.
3. Finalmente, el escenario en el que la autenticación de un programa informático en texto plano es un servicio confiable. El programa de computadora se puede ejecutar sin tener que descifrarlo cada vez, lo que supondría un desperdicio de recursos del procesador. Sin embargo, si se adjunta una etiqueta de autenticación de mensaje al programa, podría verificarse siempre que se requiera garantía de la integridad del programa.

Por el análisis realizado, podemos concluir que existe un lugar tanto para la autenticación como para el cifrado en el cumplimiento de los requisitos de seguridad. Ahora, le invito a conocer más sobre las técnicas de autenticación de mensajes

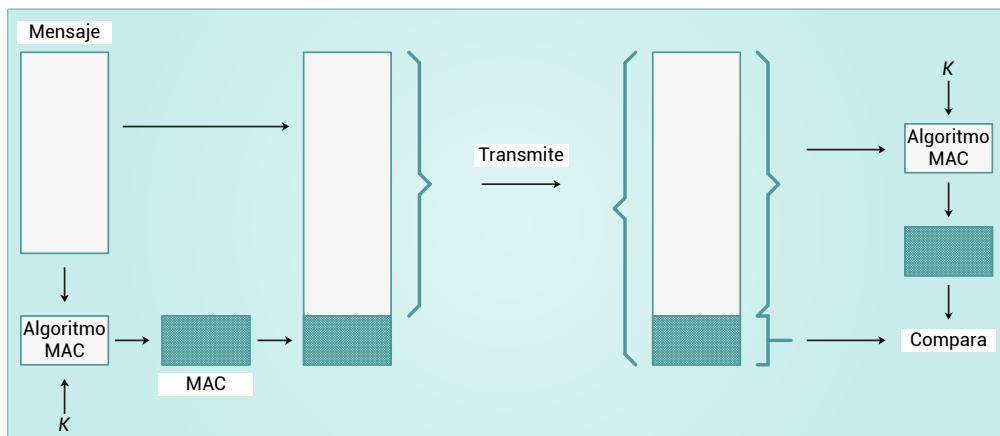
Técnicas de autenticación de mensajes

Código de autenticación de mensajes (MAC). – MAC es una técnica de autenticación de mensajes que implica el uso de una clave secreta para generar un pequeño bloque de datos, conocido como código de

autenticación de mensajes (MAC), este se adjunta al mensaje. La figura 7 describe el proceso y la interpretación de la técnica basada en MAC. Bajo este contexto, si asumimos que solo el receptor y el remitente conocen la identidad de la clave secreta y, si el código recibido coincide con el código calculado, entonces podemos concluir:

- a. El receptor tiene la seguridad de que el mensaje no ha sido alterado. Si un atacante altera el mensaje, pero no altera el código, entonces el cálculo del código por parte del receptor diferirá del código recibido, esto debido a que se supone que el atacante no conoce la clave secreta, el atacante no puede alterar el código para que se corresponda con las alteraciones en el mensaje.
- b. El receptor tiene la seguridad de que el mensaje es del presunto remitente; como nadie más conoce la clave secreta, nadie más podría preparar un mensaje con un código válido.
- c. Si el mensaje incluye un número de secuencia (como el que se usa en los protocolos X.25, HDLC y TCP), entonces, el receptor puede estar seguro de la secuencia correcta, porque un atacante no puede alterar con éxito el número de secuencia.

Figura 7.
Proceso de autenticación mediante MAC



La figura 7 supone que dos partes que se comunican, digamos A y B, comparten una clave secreta común . Cuando A tiene un mensaje para enviar a B, calcula el código de autenticación del mensaje como una función compleja del mensaje y la clave . El mensaje más el código se transmiten al

destinatario previsto. El destinatario realiza el mismo cálculo en el mensaje recibido, utilizando la misma clave secreta, para generar un nuevo código de autenticación de mensajes.

Funciones hash seguras

La función hash unidireccional, o función hash segura, es importante no solo en la autenticación de mensajes sino también en las firmas digitales. El propósito principal de una función hash es obtener un código de un archivo, mensaje u otro bloque de datos, a este código, le podemos llamar "huella digital". Para cumplir con los objetivos de la autenticación de mensajes, una función hash debe tener las propiedades indicadas en la figura 8.

Figura 8.

Propiedades de una función HASH.

Se puede aplicar a un bloque de datos de cualquier tamaño.

Produce una salida de longitud fija

$H(x)$ es relativamente fácil de calcular para cualquier x dada

Unidireccional o resistente a la imagen previa
No es factible computacionalmente encontrar x tal que $H(x) = h$

Computacionalmente inviable encontrar $y \neq x$ tal que $H(y) = H(x)$

Resistente a colisiones o fuerte resistencia a colisiones
No es factible computacionalmente encontrar cualquier par (x,y) tal que $H(x) = H(y)$

Algoritmos de función hash segura

En la actualidad, la función hash más utilizada ha sido el algoritmo hash seguro (SHA). SHA fue desarrollado por el Instituto Nacional de Estándares y Tecnología (NIST) y publicado como un estándar federal de procesamiento de información (FIPS 180) en 1993. Cuando se descubrieron debilidades en SHA, se emitió una versión revisada en 1995 y es generalmente conocida como SHA-1, produciendo un valor hash de 160 bits. En 2002, NIST produjo una versión revisada del estándar que definía tres nuevas versiones de SHA, con longitudes de valor hash de 256, 384 y 512 bits, conocidas

como SHA-256, SHA-384, y SHA-512. Estas nuevas versiones, conocidas colectivamente como SHA-2, tienen la misma estructura subyacente y utilizan los mismos tipos de operaciones binarias lógicas y aritméticas modulares que SHA-1. SHA-2, particularmente la versión de 512 bits, parecería proporcionar una seguridad inexpugnable. Sin embargo, debido a la similitud estructural de SHA-2 con SHA-1, NIST decidió estandarizar una nueva función hash que es muy diferente de SHA-2 y SHA-1; esta nueva función hash, conocida como SHA-3, se publicó en 2015 y ahora está disponible como alternativa a SHA-2.



En el capítulo 2 del texto básico, se encuentra la descripción y ejemplos de los algoritmos estudiados, por lo que es importante revisar esta sección para su mejor comprensión.

Continuemos con el aprendizaje mediante su participación en la actividad que se describe a continuación:



Actividades de aprendizaje recomendadas

Luego del estudio realizado, es importante que usted pueda diferenciar las alternativas de algoritmos para procesos de integridad de datos, frente a las alternativas de autenticación de datos; le invito a desarrollar el siguiente recurso interactivo denominado *“Algoritmos para confidencialidad y autenticación de datos”*.

[Algoritmos para confidencialidad y autenticación de datos](#)



Semana 4

1.8. Cifrado de clave pública

Avanzando con nuestro estudio, iniciaremos la revisión del cifrado de clave pública, el cual es aplicable en la autenticación de mensajes y la distribución de claves. El cifrado de clave pública, propuesto públicamente por primera vez por Diffie y Hellman en 1976 (Stallings, 2018), es el primer avance verdaderamente revolucionario en el cifrado, literalmente

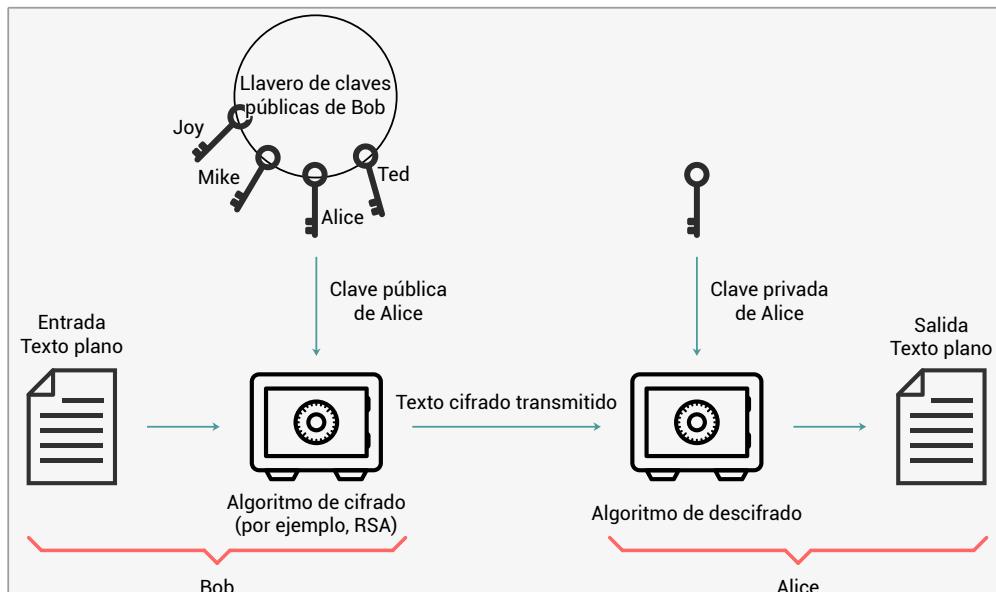
en miles de años. Los algoritmos de clave pública se basan en funciones matemáticas y no en operaciones simples sobre patrones de bits, como los que se utilizan en los algoritmos de cifrado simétrico. Más importante aún, la criptografía de clave pública es asimétrica, lo que implica el uso de dos claves separadas, en contraste con el cifrado simétrico, que usa solo una clave. La utilización de dos claves tiene profundas consecuencias en las áreas de confidencialidad, distribución de claves y autenticación.



Previo a nuestro avance en el estudio de este tema, lo invito a reflexionar sobre la siguiente pregunta, ¿Ud. cree que el cifrado de clave pública es más seguro para el criptoanálisis que los sistemas de clave privada?

Para nuestra referencia, revisemos la figura 9, donde se describen los seis componentes de un esquema de cifrado con clave pública.

Figura 9.
Componentes de un cifrado de clave pública.



La figura 9 indica los componentes del cifrado de clave pública:

- Text plano, es el mensaje original, legible, el mismo que es la entrada al algoritmo.

- b. Algoritmo de cifrado, es el proceso que realiza varias transformaciones al texto original.
- c. Clave pública y privada, conjunto de claves que se han seleccionado con el objetivo de cifrar y descifrar los datos.
- d. Texto cifrado, es el mensaje codificado, producto del uso del algoritmo de cifrado. Depende del texto original y la clave; para un mismo texto, dos claves diferentes, producirán dos salidas diferentes.
- e. Algoritmo de descifrado, es el algoritmo que acepta el texto cifrado y, con la clave correcta, produce el texto original.

Como sugieren los nombres y, de manera intuitiva, podemos indicar que la clave pública (del par de claves a usar, una pública y una privada) se hace pública para que otros la usen en el cifrado de datos, mientras que la clave privada solo la conoce su propietario y, es utilizada para descifrar los datos.



Recuerde, un algoritmo criptográfico de clave pública de propósito general, se basa en una clave para el cifrado y una clave diferente pero relacionada para el descifrado.

En el siguiente recurso interactivo denominado “*Proceso de cifrado y descifrado con clave pública*” revisaremos la interacción de los componentes utilizados en el proceso de cifrado con clave pública.

[Proceso de cifrado y descifrado con clave pública](#)



En el capítulo 2, sección 2.3 del texto básico, encontrará información complementaria a lo estudiado, por lo que es importante y, le sugiero revisar este apartado para complementar su estudio.

Ahora, profundicemos su aprendizaje acerca de las aplicaciones para criptosistemas

Aplicaciones para criptosistemas de clave pública

Como hemos revisado, los sistemas de clave pública se caracterizan por el uso de un tipo de algoritmo criptográfico con dos claves, una privada y otra disponible públicamente. Según la aplicación, el remitente usa la clave

privada del remitente o la clave pública del destinatario, o ambas, para realizar algún tipo de función criptográfica. En términos generales, podemos clasificar el uso de criptosistemas de clave pública en tres categorías:

- a. Firma digital
- a. Distribución de claves simétricas y,
- b. Cifrado de claves secretas.

Estas aplicaciones se discutirán en la siguiente sección, pero podemos adelantar indicando que, algunos algoritmos son adecuados para las tres aplicaciones, mientras que otros pueden usarse sólo para una o dos de estas aplicaciones. La tabla 6 indica las aplicaciones soportadas por los algoritmos discutidos en esta sección.

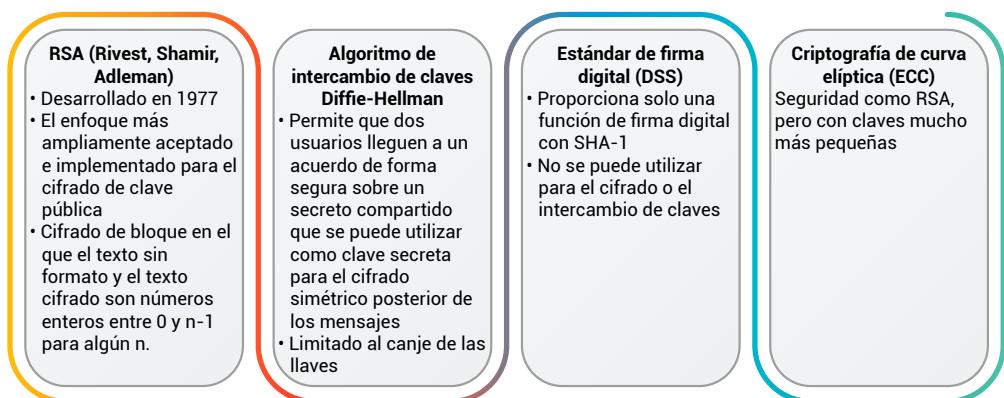
Tabla 6.
Aplicaciones para criptosistemas de clave pública

Algoritmo	Firmas digitales	Distribución de claves simétricas	Cifrado de claves secretas
RSA	Si	Si	Si
Diffie-Hellman	No	Si	No
DSS	Si	No	No
Elliptic Curve	Si	Si	Si

Algoritmos de cifrado asimétrico

La figura 10 presenta de forma general los algoritmos de clave asimétrico más utilizados.

Figura 10.
Algoritmos de cifrado asimétrico



1.9. Firmas digitales y gestión de claves

Como acabamos de revisar, existen algunas aplicaciones del cifrado de clave pública o cifrado asimétrico, de forma general, dichas aplicaciones se dividen en dos, firmas digitales y técnicas relacionadas con la gestión y distribución de claves.

Si revisamos las técnicas relacionadas con la gestión y distribución de claves, tenemos al menos tres áreas sobre las cuales el cifrado de clave pública tiene aplicación, por ejemplo:

- Distribución segura de claves públicas.
- El uso de cifrado de claves públicas para distribuir claves secretas.
- El uso de cifrado de claves públicas para crear claves temporales para cifrado de mensajes.

Con esta leve introducción, iniciaremos nuestro estudio de las firmas digitales y los diversos tipos de administración y distribución de claves.

Firmas digitales



Una firma digital es un patrón de bits dependiente de los datos, generado por un agente en función de un archivo, mensaje u otra forma de bloque de datos (Stallings, 2018).

El cifrado de clave pública se puede utilizar para la autenticación con una técnica conocida como firma digital. El NIST (julio de 2013) define una firma digital como el resultado de una transformación criptográfica de datos que, cuando se implementa correctamente, proporciona un mecanismo para verificar la autenticación de origen, la integridad de los datos y el firmante (no repudio).

Para la generación de las firmas digitales, se define tres algoritmos, así:

- a. Algoritmo de firma digital (DSA), el algoritmo original aprobado por NIST, que se basa en la dificultad de calcular logaritmos discretos.
- b. Algoritmo de firma digital RSA, basado en el algoritmo de clave pública RSA.
- c. Algoritmo de Firma Digital de Curva Elíptica (ECDSA), basado en criptografía de curva elíptica.

Como punto importante, podemos indicar que, con el uso de la firma digital Otro agente puede acceder al bloque de datos y su firma asociada y verificar (1) que el presunto firmante haya firmado el bloque de datos y (2) que el bloque de datos no haya sido alterado desde la firma. Además, el firmante no puede repudiar la firma.

Certificados de clave pública

De forma general, el objetivo del cifrado de clave pública es que la clave pública es pública, por lo tanto, si existe algún algoritmo de clave pública ampliamente aceptado, como RSA, cualquier participante puede enviar su clave pública a cualquier otro participante o, transmitir la clave a la comunidad en general. Aunque este enfoque es conveniente, tiene una gran debilidad; cualquiera puede falsificar tal anuncio público. Es decir, algún usuario podría hacerse pasar por el emisor y, enviar una clave pública a otro participante o transmitir dicha clave pública. Hasta el momento en que el emisor descubra la falsificación y alerte a otros participantes, el falsificador puede leer todos los mensajes cifrados destinados al emisor y puede usar las claves falsificadas para la autenticación.

La solución a este problema es el **certificado de clave pública**. En esencia, un certificado consta de:

- a. Una clave pública
- b. Un ID de usuario del propietario de la clave
- c. Todo el bloque firmado por un tercero de confianza.

El certificado también incluye información sobre el tercero, más una indicación del período de validez del certificado. Normalmente, el tercero es una autoridad de certificación (CA) en la que confía la comunidad de usuarios, como una agencia gubernamental o una institución financiera. Un usuario puede presentar su clave pública a la autoridad de forma segura y obtener un certificado firmado. A continuación, el usuario puede publicar el certificado. Cualquiera que necesite la clave pública de este usuario puede obtener el certificado y comprobar que es válido mediante la firma de confianza adjunta.



Para complementar el estudio de este tema, le invito a revisar la sección 2.4 del texto básico, donde encontrará la descripción de un modelo genérico del proceso de creación y uso de firmas digitales y, los pasos resumidos para la creación y validación de certificados digitales.

Intercambio de claves simétricas mediante cifrado de clave pública

Si recordamos lo revisado en la semana anterior, un requisito fundamental en el cifrado simétrico es que las partes comparten una clave segura. Si suponemos que un emisor desea permitir el acceso a un sistema únicamente con un corresponsal autorizado, el emisor debe idear una forma de compartir la clave secreta única que nadie más conozca; si el emisor y el corresponsal están físicamente cercanos, podemos pensar en un intercambio físico de un documento que contenga la clave, pero, si ambos están separados geográficamente (por ejemplo, en otro continente), ¿cómo sería el proceso de entrega de la clave? Una solución a la situación expuesta, es el uso del intercambio de claves Diffie-Hellman, esta alternativa es ampliamente utilizada, sin embargo, no proporciona autenticación entre los dos interlocutores que se comunican (existen variaciones de Diffie-Hellman que superan este problema).

Sobres digitales

Otra aplicación en la que se emplea el cifrado de clave pública para proteger una clave simétrica es el sobre digital, que se puede utilizar para proteger un mensaje sin necesidad de disponer primero que el remitente y el receptor tengan la misma clave secreta. La técnica se conoce como sobre digital, que es el equivalente a un sobre sellado que contiene una carta sin firmar.

El enfoque general lo describimos en la figura 11; supongamos que un emisor desea enviar un mensaje confidencial a un corresponsal, pero no comparten una clave secreta simétrica. El emisor hace lo siguiente:

Figura 11.
Proceso de envío de clave única



Como explica la figura 11, únicamente el corresponsal es capaz de descifrar la clave de único uso, por lo tanto, podrá recuperar el mensaje original. Si el emisor obtuvo la clave pública del corresponsal, por medio del certificado de clave pública, entonces el emisor tiene la seguridad de que una clave válida.

¡Vamos avanzando bien! Hemos finalizado el estudio de la segunda unidad, pasemos a la unidad 3, pero antes, lo invito a desarrollar la siguiente actividad.



Actividad de aprendizaje recomendada

Luego del estudio realizado a los procesos de cifrado de clave pública, firmas digitales y gestión de claves es importante que revise la sección 2.6 del texto básico, donde se expone un caso de estudio relacionado con el cifrado del almacenamiento de datos.



Autoevaluación 2

Lea atentamente las preguntas propuestas con relación a los conceptos de seguridad informática y seleccione la opción de respuesta correcta.

1. ¿Cuántas claves utiliza el cifrado simétrico?
 - a. Una.
 - b. Dos.
 - c. Una pública y una privada.
 - d. No usa claves, sino funciones hash.

2. En el cifrado simétrico por bloques, el tamaño de los bloques puede ser:
 - a. Variable, cada bloque puede ser del tamaño según el texto a enviar.
 - b. Definido previamente, al inicio, se define el valor a 64 *bits* y se trabaja todos los bloques con ese tamaño.
 - c. Fijo, el tamaño del bloque de texto cifrado es del mismo tamaño para cada bloque de texto plano ingresado.
 - d. Variable y ajustable, según el tamaño de la clave 64, 128 o 256 *bits*, los bloques se van ajustando.

3. ¿Cuál es el tamaño máximo en *bits* de la clave que usa DES?
 - a. 168 *bits*.
 - b. 256 *bits*.
 - c. 52 *bits*.
 - d. 56 *bits*.

4. ¿En qué consiste el algoritmo 3DES?
 - a. Consiste en repetir el algoritmo DES básico tres veces, utilizando dos o tres claves únicas, con un tamaño de clave de 112 o 168 *bits*.
 - b. Consiste en tomar un bloque de texto sin formato de 64 *bits* y una clave de 56 *bits* para producir un bloque de texto cifrado de 64 *bits*.
 - c. Consiste en el cifrado de bloque simétrico con una longitud de bloque de 128 *bits* y soporte para longitudes de clave de 128, 192 y 256 *bits*.
5. ¿Cuál de los siguientes algoritmos/procesos, son considerados como técnicas de autenticación de mensajes?
 - a. MAC.
 - b. DES.
 - c. 3DES.
 - d. AES.
6. Del siguiente listado, indique aquellos que aplican como algoritmos de funciones HASH.
 - a. SHA2.
 - b. SHA3.
 - c. DES.
 - d. 3DES.
7. En un algoritmo de cifrado de clave pública, ¿cuántas claves interactúan?
 - a. Dos, la pública y la privada.
 - b. Una; la pública que usa tanto el emisor como el receptor.
 - c. Tres, la pública, la privada y la de la entidad autenticadora.
 - d. Ninguna, todo se autentica contra una entidad externa.

8. De los siguientes algoritmos de clave pública, ¿cuáles son aplicables a las firmas digitales?
- a. RSA.
 - b. Diffie-Helman.
 - c. Elliptic.Curve.
 - d. DSS.
9. Indique verdadero o falso, ¿el cifrado simétrico se utiliza principalmente para proporcionar confidencialidad?
- a. Verdadero.
 - b. Falso.
10. El mensaje original o los datos que se introducen en el algoritmo de cifrado es _____.
- a. Algoritmo de cifrado.
 - b. Clave secreta.
 - c. Algoritmo de descifrado.
 - d. Texto plano.

[Ir al solucionario](#)



Si al contestar la autoevaluación su respuesta tuvo resultados positivos ¡FELICITACIONES, SIGA ADELANTE!, caso contrario revise nuevamente el contenido de los ítems errados, para reforzar su aprendizaje. Recuerde que en caso de tener alguna inquietud consulte con el profesor tutor.

Resultado de aprendizaje 3

- Describe el papel del usuario en el aseguramiento de la información y cómo encaja en un plan general de seguridad de la información para una organización.

Una vez que ha adquirido conocimientos básicos sobre seguridad en sistemas de información y herramientas criptográficos, en la unidad 3 plantea los medios generales de autenticación de la identidad de un usuario, el uso de contraseñas codificadas mediante hash, autenticación de usuarios basada en tokens, así como el enfoque para la autenticación de usuarios remotos.

Para alcanzar el resultado de aprendizaje propuesto, al término de la unidad de estudio se plantea una autoevaluación con preguntas específicas para el análisis y comprensión de contenidos, que le permitirán al estudiante retroalimentar lo estudiado.

El estudiante también realizará actividades autónomas con el acompañamiento permanente de los docentes y tutores, quienes, mediante los encuentros sincrónicos y herramientas como foros, chats académicos y tutorías aportan al proceso formativo.

Los contenidos se complementan con la guía didáctica virtualizada (GDV) y el texto básico para asegurar y garantizar la calidad de su aprendizaje. Le invito a complementar los temas de cada semana apoyándose en la lectura comprensiva de la bibliografía básica y complementaria impresa o digital disponible en la web.

Contenidos, recursos y actividades de aprendizaje



Semana 5

Unidad 2. Autenticación de usuarios

2.1. Principios de autenticación digital de usuarios

La autenticación digital de usuarios se define como el proceso de establecer la confianza en las identidades de los usuarios que se presentan electrónicamente a un sistema de información. Los sistemas pueden utilizar la identidad autenticada para determinar si el individuo autenticado está autorizado a realizar funciones particulares, como transacciones de bases de datos o acceso a recursos del sistema. En muchos casos, la autenticación y la transacción, u otra función autorizada, tienen lugar a través de una red abierta como Internet. Igualmente, la autenticación y la posterior autorización pueden tener lugar localmente, por ejemplo, a través de una red de área local, (NIST SP 800-63-3, Digital Authentication Guidelines, October 2016).

La tabla 7, extraída del NIST SP 800-171 (Protecting Controlled Unclassified Information in Nonfederal Information Systems and Organizations, diciembre de 2016), proporciona una lista útil de requisitos de seguridad para los servicios de identificación y autenticación.

Tabla 7.

Requisitos de seguridad para la identificación y la autenticación

Requisitos de seguridad	Descripción
Básicos	<ol style="list-style-type: none">Identificar los usuarios del sistema de información, los procesos que actúan en nombre de los usuarios o los dispositivos.Autenticar (o verificar) las identidades de esos usuarios, procesos o dispositivos, como requisito previo para permitir acceso a los sistemas de información de la organización.

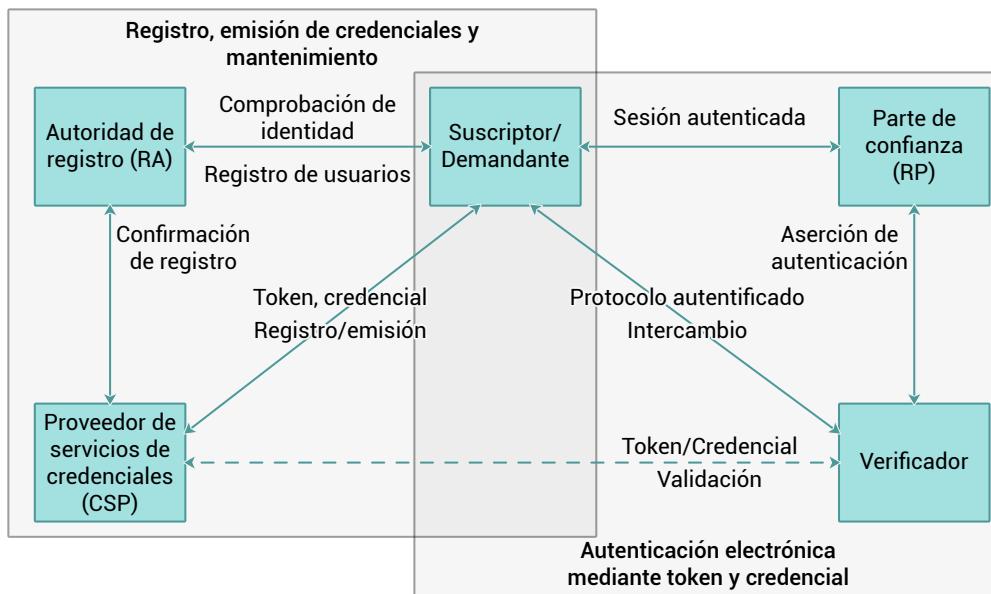
Requisitos de seguridad	Descripción
Derivados	<p>3. Utilizar la autenticación multifactor para el acceso local y de red a las cuentas con privilegios y para el acceso de red a las cuentas no privilegiadas.</p> <p>4. Emplear mecanismos de autenticación resistentes a la repetición para el acceso en red a las cuentas con y sin privilegios.</p> <p>5. Impedir la reutilización de identificadores durante un período definido.</p> <p>6. Desactivar los identificadores después de un período definido de inactividad.</p> <p>7. Imponer una complejidad mínima de la contraseña y el cambio de caracteres cuando se crean nuevas contraseñas.</p> <p>8. Prohibir la reutilización de contraseñas durante un número determinado de generaciones.</p> <p>9. Permitir el uso de contraseñas temporales para el inicio de sesión del sistema con un cambio inmediato a una contraseña permanente.</p> <p>10. Almacenar y transmitir sólo contraseñas protegidas criptográficamente.</p> <p>11. Obstruir la retroalimentación de la información de autenticación.</p>

Ahora, estimado/a estudiante le invito a profundizar sus conocimientos acerca de los principios de autenticación digital de usuarios

Modelo para la autenticación digital de usuarios

El NIST SP 800-63-3 (Digital Authentication Guidelines, October 2016) define un modelo general para la autenticación de usuarios que incluye una serie de entidades y procedimientos. Este modelo se presenta a continuación en la figura 12

Figura 12.
Modelo para autenticación de usuarios



El requisito inicial para realizar la autenticación de usuarios es que el usuario debe estar registrado en el sistema. La siguiente es una secuencia típica de registro:

- Un solicitante se dirige a una autoridad de registro (AR) para convertirse en suscriptor de un proveedor de servicios de credenciales (CSP).
- En este modelo, la AR es una entidad de confianza que establece y avala la identidad de un solicitante ante un CSP.
- A continuación, el CSP realiza un intercambio con el suscriptor.
- Dependiendo de los detalles del sistema general de autenticación, el CSP emite algún tipo de credencial electrónica al suscriptor. La credencial es una estructura de datos que vincula de forma autorizada una identidad y atributos adicionales a un token que posee un suscriptor, y que puede ser verificado cuando se presenta al verificador en una transacción de autenticación. El token puede ser una clave de cifrado o una contraseña cifrada que identifique al suscriptor. El token puede ser emitido por el CSP, generado directamente por el suscriptor, o proporcionado por un tercero. El

token y la credencial pueden utilizarse en posteriores eventos de autenticación.

Una vez que un usuario está registrado como abonado, el proceso de autenticación propiamente dicho puede tener lugar entre el abonado y uno o varios sistemas que realizan la autenticación y, posteriormente, la autorización. La parte que va a ser autenticada se llama demandante, y la parte que verifica esa identidad se llama verificador. Cuando un demandante demuestra con éxito la posesión y el control de un token a un verificador a través de un protocolo de autenticación, el verificador puede verificar que el demandante es el suscriptor nombrado en la credencial correspondiente. El verificador transmite una afirmación sobre la identidad del suscriptor a una entidad denominada parte de confianza (RP). Esa afirmación incluye información de identidad sobre el abonado, como el nombre del abonado, un identificador asignado en el registro u otros atributos del abonado que se verificaron en el proceso de registro. El RP puede utilizar la información autenticada proporcionada por el verificador para tomar decisiones de control de acceso o de autorización.

Un sistema de autenticación implementado será diferente o más complejo que este modelo simplificado, pero el modelo ilustra los roles y funciones clave necesarias para un sistema de autenticación seguro.

Medios de autenticación

Hay cuatro medios generales para autenticar la identidad de un usuario, que pueden utilizarse solos o combinados, estos se muestran a continuación:

Tabla 8.

Medios de autenticación de usuarios

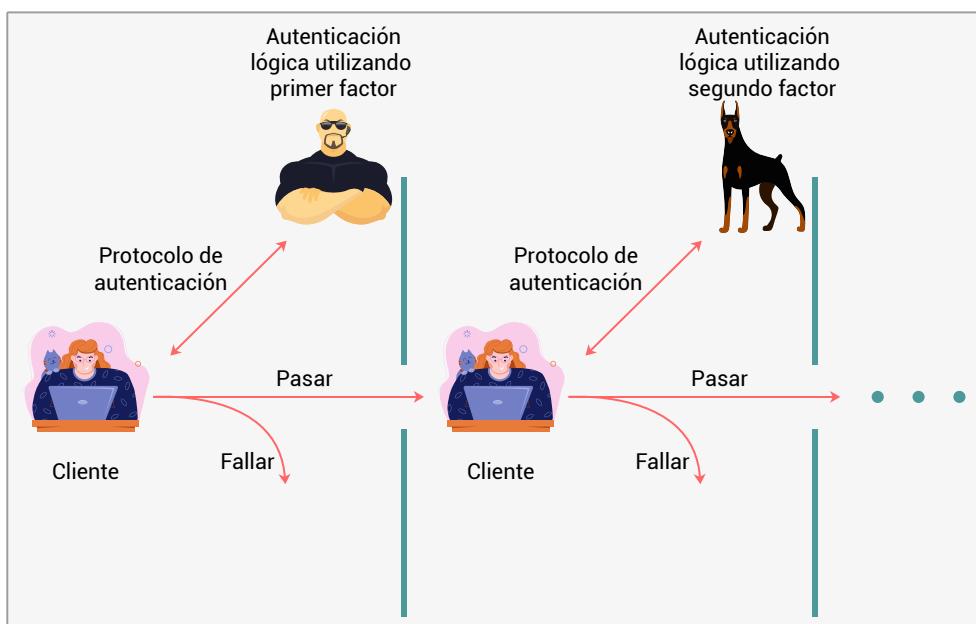
Algo que el individuo conoce	Algo que el individuo posee	Algo que el individuo es (biometría estática)	Algo que el individuo hace (biometría dinámica)
<ul style="list-style-type: none">▪ Contraseña▪ PIN▪ Respuestas a un conjunto preestablecido de preguntas	<ul style="list-style-type: none">▪ Tarjetas electrónicas▪ Tarjetas inteligentes▪ Llaves físicas	<ul style="list-style-type: none">▪ Huella dactilar▪ Retina▪ Rostro	<ul style="list-style-type: none">▪ Patrón de voz▪ Características de escritura a mano▪ Ritmo de tecleo

Dada esta revisión de conceptos, ¿cree usted que estos cuatro medios de autenticación son suficientes?

De hecho, correctamente implementados y utilizados, pueden proporcionar una autenticación segura del usuario. Sin embargo, cada método tiene problemas. Un adversario puede ser capaz de adivinar o robar una contraseña. Del mismo modo, un adversario puede ser capaz de falsificar o robar un token.

Un usuario puede olvidar una contraseña o perder un token. Además, la gestión de la información de las contraseñas y los tokens en los sistemas y la protección de dicha información en los sistemas conlleva una importante carga administrativa. Con respecto a los autenticadores biométricos, hay una serie de problemas, incluyendo el tratamiento de falsos positivos y falsos negativos, la aceptación del usuario, el coste y la comodidad. La autenticación multifactor se refiere al uso de más de uno de los medios de autenticación mostrados en la figura 13.

Figura 13.
Autenticación multifactor



La fuerza de los sistemas de autenticación viene determinada en gran medida por el número de factores que incorpora el sistema. Las implementaciones que utilizan dos factores se consideran más fuertes que las que solo utilizan un factor; los sistemas que incorporan tres factores

son más fuertes que los sistemas que solamente incorporan dos de los factores, y así sucesivamente.

Evaluación de riesgos para la autenticación de usuarios

La evaluación de riesgos de seguridad involucra tres conceptos distintos que se relacionan entre sí: nivel de seguridad, impacto potencial y áreas de riesgo.

Figura 14.

Evaluación de riesgos, autenticación de usuarios

Nivel de seguridad	Impacto potencial	Áreas de riesgo
Nivel 1 Poca o ninguna confianza en la validez de la identidad declarada.	Bajo Se puede esperar que un error de autenticación tenga un efecto adverso limitado en las operaciones de la organización, los activos o los individuos.	Baja En el peor de los casos, una pérdida financiera irrecuperable insignificante o sin consecuencias para cualquier parte, o en el peor de los casos, una responsabilidad de la organización insignificante o sin consecuencias.
Nivel 2 Cierta confianza en la validez de la identidad declarada.	Moderado Un error de autenticación podría tener un efecto adverso grave.	Moderada En el peor de los casos, una pérdida financiera grave e irrecuperable para cualquiera de las partes, o una responsabilidad grave de la organización.
Nivel 3 Alta confianza en la validez de la identidad declarada.	Alto Un error de autenticación podría tener un efecto adverso grave o catastrófico.	Alta Pérdida financiera grave o catastrófica irrecuperable para cualquiera de las partes; o responsabilidad grave o catastrófica de la organización.
Nivel 4 Confianza muy alta en la validez de la identidad declarada.		
Alta Pérdida financiera grave o catastrófica irrecuperable para cualquiera de las partes; o responsabilidad grave o catastrófica de la organización.		

2.2. Autenticación basada en contraseña

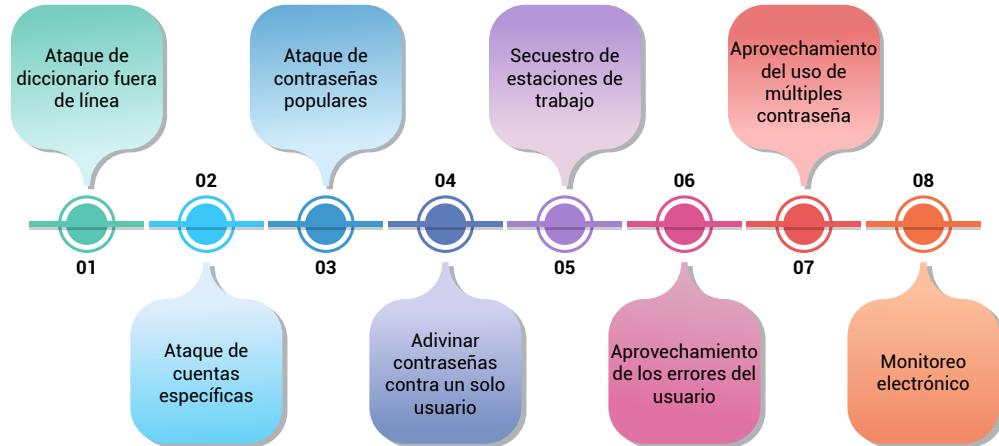
Una línea de defensa muy utilizada contra los intrusos es el sistema de contraseñas. Prácticamente, todos los sistemas multiusuario, los servidores basados en la red, los sitios de comercio electrónico basados en la web y otros servicios similares requieren que un usuario proporcione no solo un nombre o identificador (ID) sino también una contraseña. El sistema compara la contraseña con una contraseña previamente almacenada para ese ID de usuario, mantenida en un archivo de contraseñas del sistema. La contraseña sirve para autenticar el ID de la persona que se conecta al sistema. A su vez, el ID proporciona seguridad de las siguientes maneras:

- El ID determina si el usuario está autorizado a acceder a un sistema. En algunos sistemas, solamente se permite el acceso a aquellos que ya tienen un ID registrado en el sistema.
- El ID determina los privilegios concedidos al usuario. Algunos usuarios pueden tener el estatus de administrador o "superusuario" que les permite leer archivos y realizar funciones especialmente protegidas por el sistema operativo. Algunos sistemas tienen cuentas de invitado o anónimas, y los usuarios de estas cuentas tienen privilegios más limitados que otros.
- El ID se utiliza en lo que se denomina control de acceso discrecional. Por ejemplo, al enumerar los ID de los demás usuarios, un usuario puede concederles permiso para leer los archivos que le pertenecen a ese usuario.

La vulnerabilidad de las contraseñas

A continuación, en la figura 15 puede ver las principales formas de ataque contra la autenticación basada en contraseñas.

Figura 15.
Vulnerabilidad de las contraseñas, formas de ataque



Para comprender las formas de ataque y conocer las estrategias de contramedidas, lo invito a revisar este tema, en el capítulo 3 del texto básico.

A pesar de las numerosas vulnerabilidades de seguridad de las contraseñas, estas siguen siendo la técnica de autenticación de usuarios

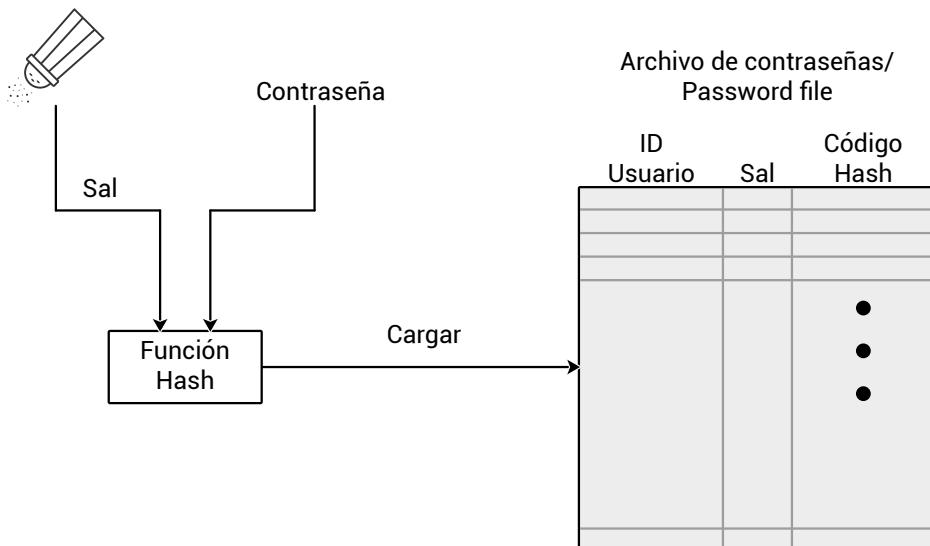
más utilizada, y es poco probable que esto cambie en un futuro próximo. (referenciar)

Uso de contraseñas codificadas

Una técnica de seguridad de contraseñas muy utilizada es el uso de contraseñas con hash, método por el cual se agrega una secuencia aleatoria de bits (denominada valor de sal), a la contraseña, para lo cual se emplea el procedimiento mostrado en la figura 16, generalmente utilizado por sistemas UNIX.

Figura 16.

Cargar una nueva contraseña



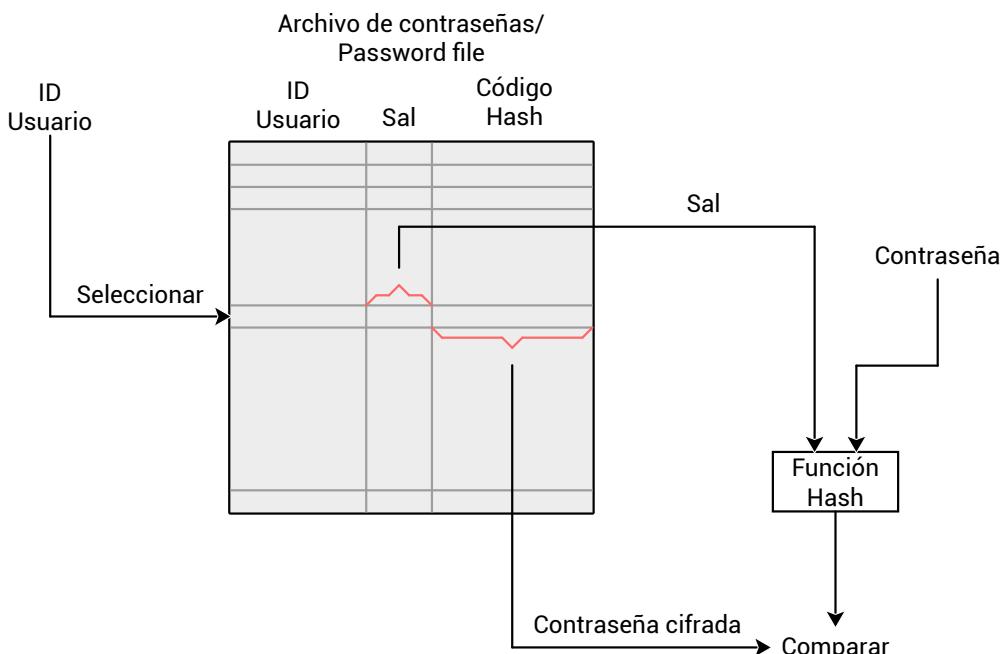
Para cargar una nueva contraseña en el sistema, el usuario selecciona o se le asigna una contraseña, la cual se combina con un valor de sal de longitud fija; la contraseña y la sal sirven como entradas a un algoritmo de hash para producir un código de hash de longitud fija; finalmente la contraseña cifrada se almacena, junto con una copia en texto plano de la sal, en el archivo de contraseñas para el correspondiente identificador (ID) de usuario. El algoritmo hash está diseñado para que se ejecute lentamente con el fin de frustrar los ataques.

Cuando un usuario intenta entrar en un sistema, proporciona un ID y una contraseña, el sistema operativo utiliza el ID para indexar el archivo de contraseñas y recuperar el valor de sal en texto plano y la contraseña

cifrada, el valor de sal y la contraseña proporcionada por el usuario se utilizan como entrada a la rutina de encriptación. Si el resultado coincide con el valor almacenado, se acepta la contraseña. Ver figura 17.

Figura 17.

Verificar una contraseña



Propósitos del valor de sal:

- Evita que las contraseñas duplicadas sean visibles en el archivo de contraseñas.
- Dificultar los ataques de diccionario fuera de línea, aumentando la complejidad de adivinar una contraseña en un ataque de diccionario.
- Resulta casi imposible averiguar si una persona con contraseñas en dos o más sistemas ha usado la misma contraseña en todos ellos.

Descifrado de contraseñas elegidas por el usuario

El enfoque tradicional para adivinar contraseñas, o también llamado cracking de contraseñas, es desarrollar un gran diccionario de posibles contraseñas y probar cada una de ellas contra el archivo de contraseñas. Esto significa que cada contraseña debe ser convertida en hash usando cada valor de sal disponible y luego comparada con los valores de hash

almacenados. Si no se encuentra ninguna coincidencia, el programa de cracking intenta hacer variaciones de todas las palabras de su diccionario de contraseñas probables. Dichas variaciones incluyen la ortografía inversa de las palabras, números adicionales o caracteres especiales, o una secuencia de caracteres.

Control de acceso al archivo de contraseñas

Una forma de frustrar un ataque con contraseña es negar al oponente el acceso al archivo de contraseñas. Si la parte de la contraseña cifrada del archivo solo es accesible por un usuario con privilegios, el adversario no podrá leerla sin conocer la contraseña de un usuario con privilegios. A menudo, las contraseñas cifradas se guardan en un archivo separado de los ID de usuario, denominado archivo de contraseñas en la sombra (shadow password file), el mismo que debe estar protegido del acceso no autorizado.

Aunque la protección del archivo de contraseñas merece la pena, sigue habiendo vulnerabilidades, como se detalla en la figura 18.

Figura 18.
Archivo de contraseñas y vulnerabilidades

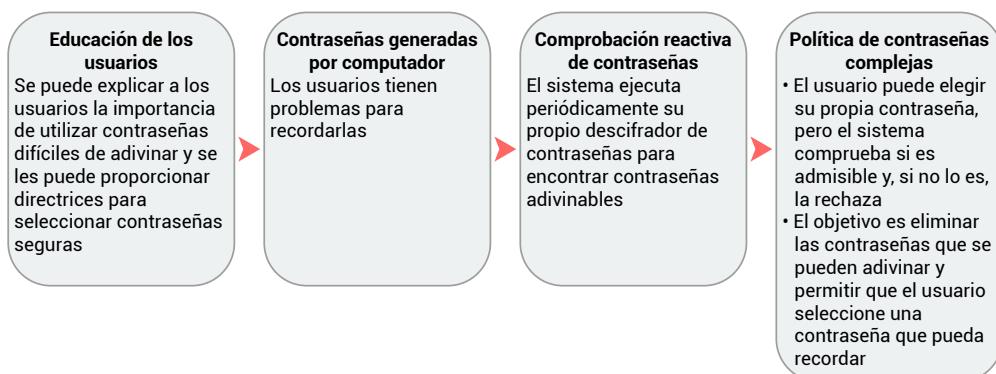
Bloquear los ataques de adivinación fuera de línea negando el acceso al archivo de contraseñas y cifradas					
Habilitar acceso sólo a los usuarios con privilegios	Vulnerabilidades				
	Debilidad en el sistema operativo que permite el acceso al archivo	Accidente con los permisos que lo hacen legible	Usuarios con la misma contraseña en otro sistema	Acceso desde un medio de respaldo	Captura de contraseñas en el tráfico de red
Uso del archivo shadow password					

Estrategias de selección de contraseñas

Cuando no hay restricciones, muchos usuarios eligen una contraseña demasiado corta o demasiado fácil de adivinar. En el otro extremo, si a los usuarios se les asignan contraseñas formadas por ocho caracteres imprimibles seleccionados al azar, el descifrado de la contraseña es efectivamente imposible, pero sería casi igual de imposible para la mayoría de los usuarios recordar sus contraseñas.

Afortunadamente, incluso si se limita el universo de contraseñas a cadenas de caracteres que sean razonablemente memorables, el tamaño del universo sigue siendo demasiado grande para permitir el descifrado práctico. El objetivo, por tanto, es eliminar las contraseñas que se pueden adivinar y permitir al mismo tiempo que el usuario seleccione una contraseña que sea memorable. Para ello se utilizan cuatro técnicas básicas descritas en la figura 19.

Figura 19.
Selección de contraseñas



Recuerde, la autenticación del usuario abarca dos funciones:

1. El usuario se identifica ante el sistema presentando una credencial, como su ID de usuario.
2. El sistema verifica al usuario mediante el intercambio de información de autenticación.



Para profundizar sobre las estrategias de selección de contraseñas, lo invito a revisar este tema, en el capítulo 3 del texto básico.



Semana 6

2.3. Autenticación basada en token

¿Se ha preguntado qué es un token?

Los objetos que posee un usuario con el fin de autenticarse se denominan tokens, generalmente implementados mediante tarjetas.

Tabla 9.

Tipos de tarjetas utilizadas como tokens

Tipo de tarjeta	Característica	Ejemplo
En relieve	Sólo caracteres en relieve, en el anverso	Tarjeta de crédito antigua
Banda magnética	Barra magnética en el reverso, caracteres en el anverso	Tarjeta bancaria
Memoria	Memoria electrónica en el interior	Tarjeta telefónica prepago
Inteligente	Memoria electrónica y procesador en el interior	Tarjeta de identificación
Con contacto	Contactos eléctricos expuestos en la superficie	
Sin contacto	Antena de radio incrustada en el interior	remota

Lo invito a conocer los tipos de tokens más utilizados: tarjeta de memoria (figura 20) y tarjeta inteligente (figura 21).

Figura 20.

Tarjeta de memoria

01

Pueden almacenar pero no procesar datos

02

La más común es la tarjeta de banda magnética

03

Puede incluir una memoria electrónica interna

04

Puede utilizarse solo para el acceso físico (habitación e hotel, cajero automático)

05

Proporcionan mayor seguridad en combinación con una contraseña o un PIN

06

Inconvenientes de las tarjetas de memoria

- Requiere un lector especial
- Pérdida del token
- Insatisfacción del usuario

Figura 21.
Tarjeta inteligente



2.4. Autenticación biométrica

Un sistema de autentificación biométrica intenta autenticar a un individuo basándose en sus características físicas únicas. Estas incluyen características estáticas, como las huellas dactilares, la geometría de la mano, las características faciales y los patrones de la retina y el iris; y características dinámicas, como la huella vocal y la firma. En esencia, la biometría se basa en el reconocimiento de patrones. En comparación con las contraseñas y los tokens, la autenticación biométrica es técnicamente más compleja y costosa.

La figura 22 detalla las características utilizadas en aplicaciones biométricas.

Figura 22.

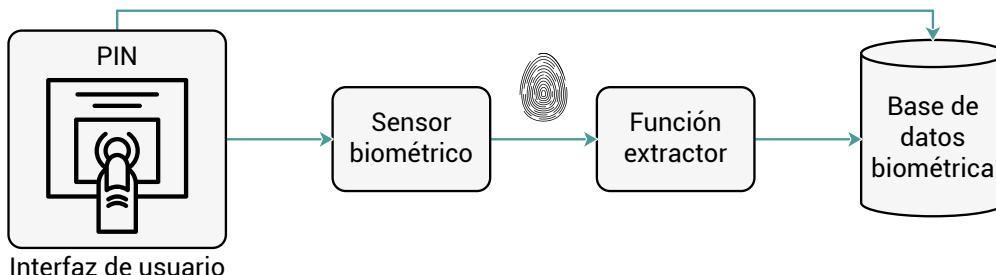
Características físicas utilizadas en las aplicaciones biométricas

Faciales	<ul style="list-style-type: none">Basado en la ubicación y forma relativa de los rasgos faciales clave (ojos, cejas, nariz, labios y barbilla)Mediante cámaras de infrarrojos se puede producir un termograma facial que se correlacione con el sistema vascular en el rostro humano
Huellas dactilares	<ul style="list-style-type: none">Patrón de crestas y surcos en la superficie de la yema del dedoUsado durante siglosSistematizado y automatizado para fines policialesSe cree que las huellas dactilares son únicas en toda la población humana
Geometría de la mano	Identifican la características de la mano, incluyendo la forma, longitud y anchura de los dedos
Patrón retinario	<ul style="list-style-type: none">El patrón formado por las venas bajo la superficie de la retina es únicoObtiene una imagen digital del patrón retiniano proyectando un haz de luz visual o infrarroja de baja intensidad en el ojo.
Iris	La estructura detallada del iris constituye una característica física única
Firma	<ul style="list-style-type: none">Estilo único de escritura de un individuo, reflejado especialmente en la firmaVarias muestras de firmas de un mismo individuo pueden no ser idénticas, lo que complica la tarea de desarrollar una representación informática de la firma que pueda compararse con futuras muestras
Voz	<ul style="list-style-type: none">Patrones de voz estrechamente ligados a las características físicas y anatómicas del hablanteVariación de una muestra a otra a lo largo del tiempo de un mismo hablante, lo que complica la tarea de reconocimiento biométrico.

En un sistema biométrico cada persona que vaya a ser incluida en la base de datos de usuarios autorizados debe ser primero registrada en el sistema, este proceso se ilustra en la figura 23.

Figura 23.

Sistema biométrico genérico, registro de usuario



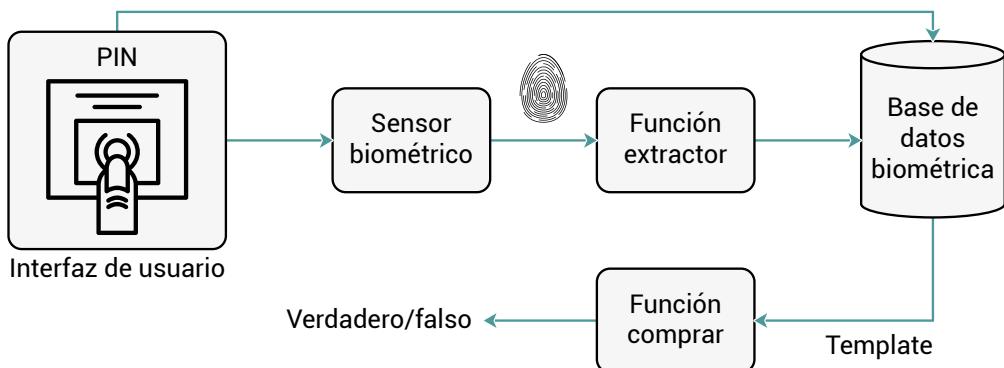
En un sistema biométrico, el usuario presenta al sistema un nombre y, normalmente, algún tipo de contraseña o PIN. Al mismo tiempo, el sistema detecta alguna característica biométrica de este usuario (por ejemplo, la

huella dactilar del dedo índice derecho). El sistema digitaliza la entrada y extrae un conjunto de características que pueden almacenarse como un número o conjunto de números que representan esta característica biométrica única; este conjunto de números se denomina plantilla (template) del usuario. El usuario queda registrado en el sistema, que mantiene para el usuario un nombre (ID), tal vez un PIN o contraseña, y el valor biométrico.

La autenticación del usuario en un sistema biométrico implica el proceso de verificación, en la que, el usuario introduce un PIN y también utiliza un sensor biométrico, el sistema extrae la característica correspondiente y la compara con la plantilla (template) almacenada para este usuario. Si hay una coincidencia, el sistema autentifica a este usuario. El proceso de verificación se ilustra en la figura 24.

Figura 24.

Sistema biométrico generico, verificación de usuario



2.5. Autenticación de usuario remoto

La forma más sencilla de autenticación de usuarios es la autenticación local, en la que un usuario intenta acceder a un sistema que está presente localmente, como un PC de oficina independiente o un cajero automático. El caso más complejo es el de la autenticación de usuario remota, que tiene lugar a través de Internet, una red o un enlace de comunicaciones. La autenticación remota de usuarios plantea amenazas de seguridad adicionales, como que un espía pueda capturar una contraseña o que un adversario reproduzca una secuencia de autenticación que haya sido observada.

Para contrarrestar las amenazas a la autenticación remota de usuarios, los sistemas suelen basarse en algún tipo de protocolo de desafío-respuesta, mostrados en la figura 25.

Figura 25.

Tipos de protocolos de desafío-respuesta

Contraseñas

- El host no almacena la contraseña, sino un código hash de la misma
- No se transmite directamente el hash de la contraseña, sino una función en la que el hash de la contraseña es uno de los argumentos

Tokens

- Un usuario transmite primero su identidad al host remoto
- El token almacena un código de acceso estático o genera un código de acceso aleatorio de una sola vez

Biométrico

- Estático
- El usuario transmite un ID al host, que responde con un número aleatorio y un identificador de cifrado
- Dinámico
- El host proporciona una secuencia aleatoria así como un número aleatorio como reto.

¡Excelente! Hemos finalizado el estudio de la tercera unidad, lo invito a desarrollar las actividades de aprendizaje, con el fin de evaluar los conocimientos adquiridos hasta el momento.



Actividad de aprendizaje recomendada

Desarrolle la autoevaluación de la unidad 3 “Autenticación de usuarios”, considere que las preguntas planteadas constituyen una estrategia de aprendizaje y tienen como finalidad conocer el grado de asimilación de los contenidos estudiados. En caso de que tenga dificultad para responder alguna pregunta, le recomiendo volver a revisar los contenidos en el texto básico y la guía didáctica virtualizada.



Autoevaluación 3

Lea atentamente las preguntas propuestas en relación con los conceptos de autenticación de usuarios y seleccione la opción de respuesta correcta.

1. Hay cuatro medios generales para autenticar la identidad de un usuario, uno de ellos es:
 - a. Propiedades del individuo.
 - b. Algo que el individuo conoce.
 - c. Cuenta bancaria del individuo.
 - d. Referencias del individuo.
2. ¿El método por el cual se agrega una secuencia aleatoria de *bits* a la contraseña se denomina?
 - a. *Token*.
 - b. PIN.
 - c. Hash.
 - d. NIST.
3. La evaluación de riesgos de seguridad involucra tres conceptos distintos que se relacionan entre sí. Elija la opción correcta.
 - a. Nivel de riesgo, impacto discrecional y áreas de seguridad.
 - b. Nivel de impacto, riesgo potencial y áreas de seguridad.
 - c. Nivel de seguridad, impacto discrecional y áreas de riesgo.
 - d. Nivel de seguridad, impacto potencial y áreas de riesgo.
4. Respecto a la autenticación biométrica, ¿cuál de las siguientes afirmaciones es correcta?
 - a. Es técnicamente más compleja y costosa.
 - b. A pesar de que es más compleja su costo es bajo.
 - c. Es técnicamente sencilla por lo que su costo es mínimo.
 - d. Su complejidad no implica que sea costosa.

5. La presentación o la generación de información de autenticación que corrobora la vinculación entre la entidad y el identificador es la _____.
- etapa de identificación.
 - etapa de verificación.
 - etapa de autenticación.
 - etapa de corroboración.
6. El reconocimiento por huella dactilar, retina y rostro son ejemplos de _____.
- reconocimiento facial.
 - biometría dinámica.
 - biometría estática.
 - autenticación por *token*.
7. Un _____ es un programa para adivinar contraseñas.
- hash de contraseñas.
 - cracker de contraseñas.
 - biométrico de contraseñas.
 - sal de contraseñas.
8. La estrategia _____ es cuando se explica a los usuarios la importancia de utilizar contraseñas difíciles de adivinar y se les proporcionan directrices para seleccionar contraseñas seguras.
- comprobación reactiva de contraseñas.
 - comprobación proactiva de contraseñas.
 - contraseña generada por ordenador.
 - educación del usuario.
9. Cada persona que vaya a ser incluida en la base de datos de usuarios autorizados debe ser primero _____ en el sistema.
- verificado.
 - autenticado.
 - identificado.
 - registrado.

10. _____, sistema que identifica las características de la mano, incluyendo la forma y la longitud y anchura de los dedos.
- a. Firma.
 - b. Geometría de la mano.
 - c. Huella dactilar.
 - d. Huella de la palma de la mano.

[Ir al solucionario](#)

 *Si al contestar la autoevaluación su respuesta tuvo resultados positivos ¡FELICITACIONES, SIGA ADELANTE!, caso contrario revise nuevamente el contenido de los ítems errados, para reforzar su aprendizaje. Recuerde que en caso de tener alguna inquietud puede consultar al profesor tutor.*

Resultado de aprendizaje 4

- Enumera y explica las amenazas y las vulnerabilidades típicas para la red de una organización.

Ahora que domina los conceptos básicos de autenticación de usuarios, puede avanzar con la unidad 4, en la que conocerá los principales conceptos del control de acceso, en un contexto más amplio que incluye autenticación, autorización y auditoría.

Para alcanzar el resultado de aprendizaje propuesto, al término de la unidad de estudio se plantea una autoevaluación con preguntas específicas para el análisis y comprensión de contenidos, que le permitirán al estudiante retroalimentar lo estudiado.

El estudiante también realizará actividades autónomas con el acompañamiento permanente de los docentes y tutores, quienes, mediante los encuentros sincrónicos y herramientas como foros, chats académicos y tutorías aportan al proceso formativo.

Los contenidos se complementan con la guía didáctica virtualizada (GDV) y el texto básico para asegurar y garantizar la calidad de su aprendizaje. Le invito a complementar los temas de cada semana apoyándose en la lectura comprensiva de la bibliografía básica y complementaria impresa o digital disponible en la web.

Contenidos, recursos y actividades de aprendizaje



Semana 7

Unidad 3. Control de acceso

3.1. Principios de control de acceso

¿A qué se refiere el control de acceso?

Dos definiciones de control de acceso son útiles para entender su alcance.

El NISTIR 7298 (Glosario de términos clave de seguridad de la información, mayo de 2013), define el control de acceso como el proceso de concesión o denegación de solicitudes específicas para:

1. obtener y utilizar información y servicios de procesamiento de información relacionados; y
2. entrar en instalaciones físicas específicas.

El RFC 4949, Internet Security Glossary, define el control de acceso como un proceso por el cual el uso de los recursos del sistema se regula de acuerdo con una política de seguridad y solamente se permite a las entidades autorizadas (usuarios, programas, procesos u otros sistemas) de acuerdo con esa política.

En un sentido amplio, toda la seguridad informática está relacionada con el control de acceso. De hecho, el RFC 4949 define la seguridad informática de la siguiente manera: medidas que implementan y garantizan los servicios de seguridad en un sistema informático, en particular los que aseguran el servicio de control de acceso. El control de acceso implementa una política de seguridad que especifica quién o quiénes pueden tener acceso a cada recurso específico del sistema, y el tipo de acceso que se permite en cada caso. Veamos los requisitos en la siguiente tabla:

Tabla 10.
Requisitos de seguridad para control de acceso

Requisitos de seguridad básicos
1. Limitar el acceso al sistema de información a los usuarios autorizados, a los procesos que actúan en nombre de los usuarios autorizados o a los dispositivos.
2. Limitar el acceso al sistema de información a los tipos de transacciones y funciones que los usuarios autorizados están autorizados.
3. Requisitos de seguridad derivados
4. Controlar el flujo de CUI de acuerdo con las autorizaciones aprobadas.
5. Separar las funciones de los individuos para reducir el riesgo de actividad maliciosa sin colusión.
6. Emplear el principio de mínimo privilegio, incluso para funciones de seguridad específicas y cuentas privilegiadas.
7. Utilizar cuentas o roles no privilegiados cuando se acceda a funciones no relacionadas con la seguridad.
8. Evitar que los usuarios sin privilegios ejecuten funciones privilegiadas y auditar la ejecución de dichas funciones.
9. Limitar los intentos fallidos de inicio de sesión.

Requisitos de seguridad básicos

10. Proporcionar avisos de privacidad y seguridad coherentes con las normas CUI aplicables.
11. Utilizar el bloqueo de la sesión con pantallas de ocultación de patrones para impedir el acceso y la visualización de datos tras un período de inactividad.
12. Terminar (automáticamente) una sesión de usuario después de una condición definida.
13. Supervisar y controlar las sesiones de acceso remoto.
14. Emplear mecanismos criptográficos para proteger la confidencialidad de las sesiones de acceso remoto.
15. Enrutar el acceso remoto a través de puntos de control de acceso gestionados.
16. Autorizar la ejecución remota de comandos privilegiados y el acceso remoto a información relevante para la seguridad.
17. Autorizar el acceso inalámbrico antes de permitir dichas conexiones.
18. Proteger el acceso inalámbrico mediante autenticación y cifrado.
19. Controlar la conexión de los dispositivos móviles.
20. Cifrar la CUI en los dispositivos móviles.
21. Verificar y controlar/limitar las conexiones y el uso de sistemas de información externos.
22. Limitar el uso de dispositivos de almacenamiento portátiles de la organización en sistemas de información externos.
23. Controlar la CUI publicada o procesada en sistemas de información de acceso público.

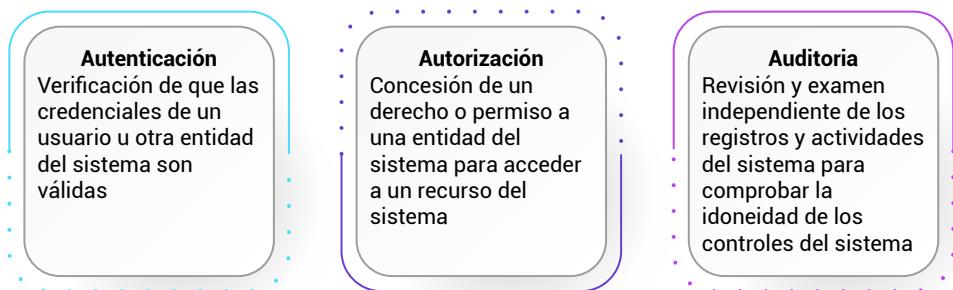
Nota: CUI = controlled unclassified information (información no clasificada controlada)

Un mecanismo de control de acceso media entre un usuario (o un proceso que se ejecuta en nombre de un usuario) y los recursos del sistema, como aplicaciones, sistemas operativos cortafuegos, routers, archivos y bases de datos.

El control de acceso incluye las entidades y funciones descritas en la figura 26.

Figura 26.

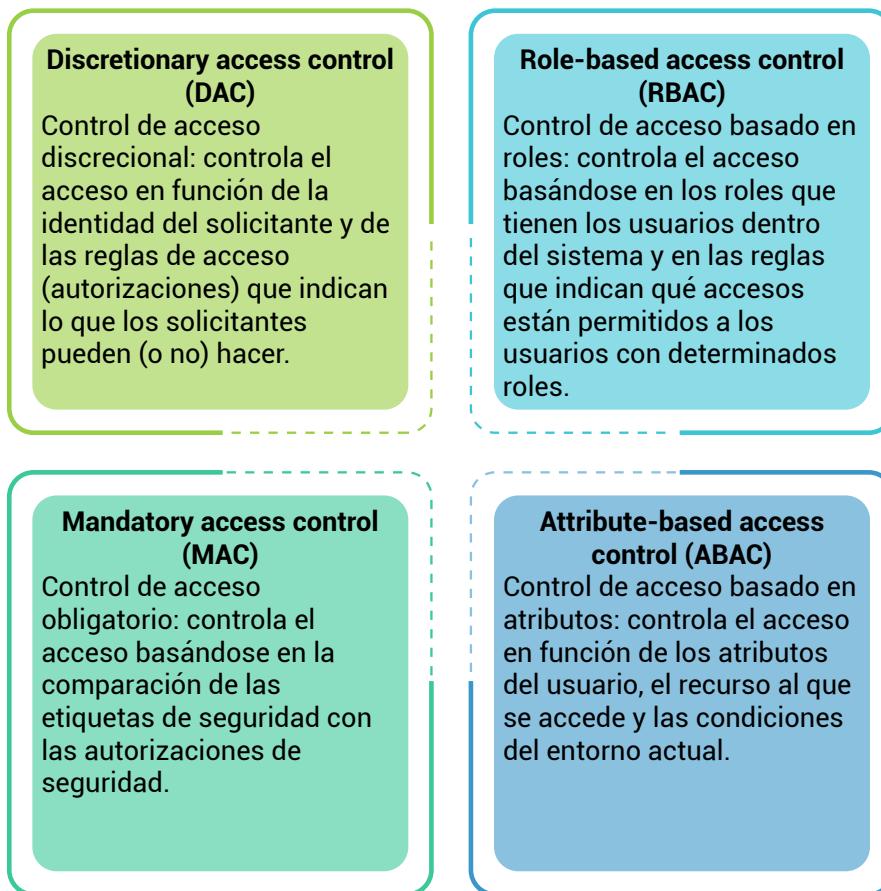
Contexto del control de acceso



Una política de control de acceso, que puede plasmarse en una base de datos de autorización, dicta qué tipos de acceso se permiten, en qué circunstancias y por quién. Las políticas de control de acceso se agrupan generalmente en cuatro categorías, las mismas que constan en la figura 27.

Figura 27.

Políticas de control de acceso

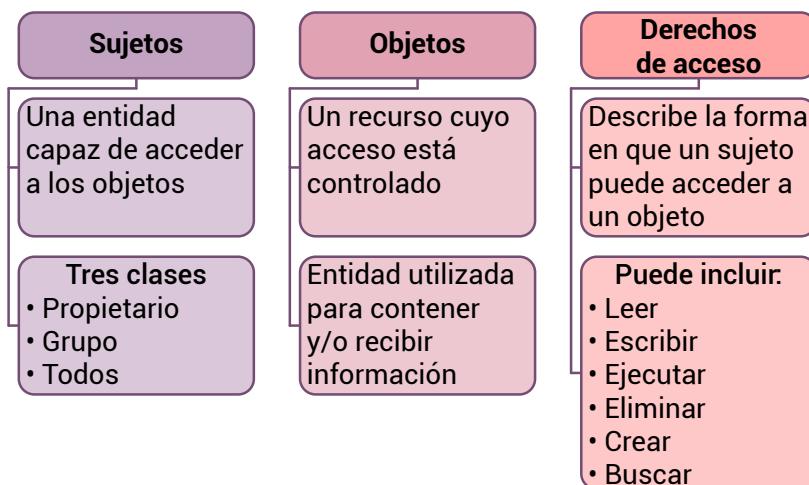


3.2. Sujetos, objetos y derechos de acceso

Los elementos básicos del control de acceso son: sujeto, objeto y derecho de acceso. Un sujeto es una entidad capaz de acceder a los objetos. En general, el concepto de sujeto equivale al de proceso. Cualquier usuario o aplicación accede realmente a un objeto por medio de un proceso que representa a ese usuario o aplicación. El proceso asume los atributos del usuario, como los derechos de acceso.

Figura 28.

Sujetos, objetos y derechos de acceso



3.3. Control de acceso discrecional (DAC)

Como ya se explicó en la sección principios de control de acceso, un esquema de control de acceso discrecional es aquel en el que se pueden conceder a una entidad derechos de acceso que le permitan, por su propia voluntad, permitir a otra entidad acceder a algún recurso. Un enfoque general del DAC, tal y como lo lleva a cabo un sistema operativo o un sistema de gestión de bases de datos, es el de una matriz de acceso.

Figura 29.

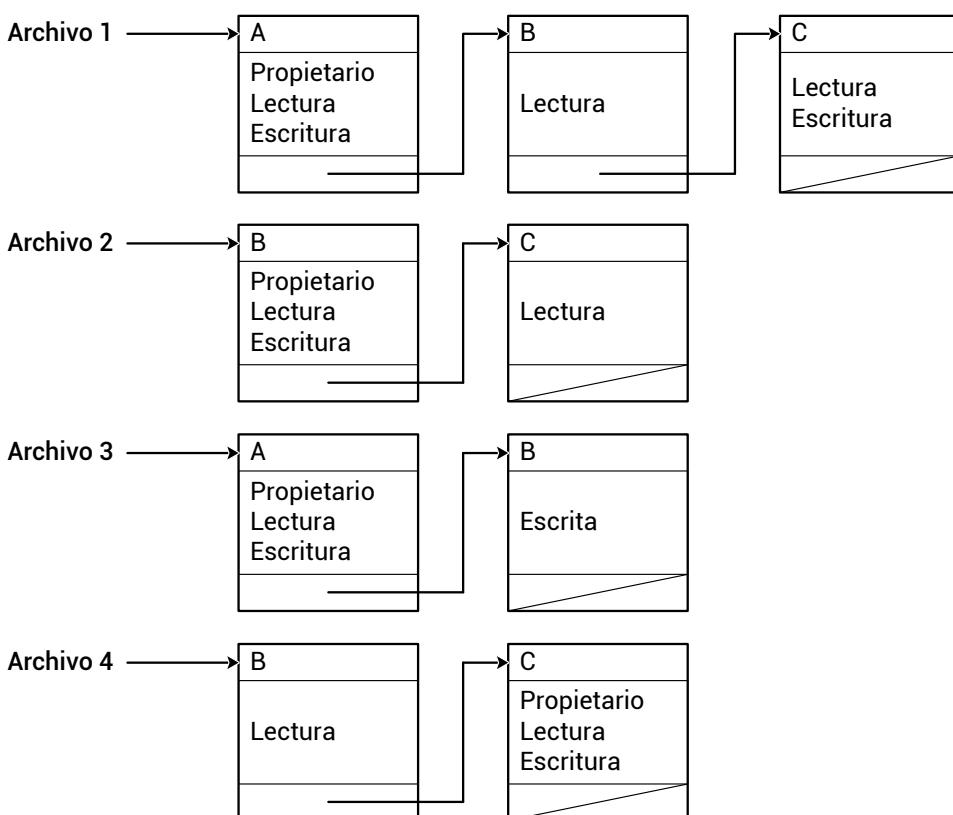
Matriz de acceso

		Objetos			
		Archivo 1	Archivo 2	Archivo 3	Archivo 4
Sujetos	Usuario A	Propietario Lectura Escritura		Propietario Lectura Escritura	
	Usuario B	Lectura	Propietario Lectura Escritura	Escritura	Lectura
	Usuario C	Lectura Escritura	Lectura		Propietario Lectura Escritura

En la práctica, una matriz de acceso se implementa mediante listas de control de acceso (ACL), para cada objeto, una ACL enumera los usuarios y sus derechos de acceso permitidos, la ACL puede contener una entrada por defecto, o pública, esto permite que los usuarios que no están explícitamente listados con derechos especiales tengan un conjunto de derechos por defecto. El conjunto de derechos por defecto debe seguir siempre la regla del menor privilegio o acceso de solo lectura, según sea aplicable. Los elementos de la lista pueden incluir usuarios individuales, así como grupos de usuarios.

Figura 30.

Listas de control de acceso para los archivos de la figura 29



Cuando se desea determinar qué sujetos tienen qué derechos de acceso a un recurso en particular, las ACL son convenientes, porque cada ACL proporciona la información para un recurso determinado. Sin embargo, esta estructura de datos no es conveniente para determinar los derechos de acceso disponibles para un usuario específico.

3.4. Ejemplo: control de acceso a archivos UNIX

Para abordar el control de acceso a archivos en UNIX, es necesario conocer conceptos básicos relativos a los archivos y directorios UNIX.

Figura 31.

Control de acceso a archivos UNIX tradicional

Los archivos UNIX se administran mediante inodos (nodos de índice)

- Estructuras de control con información clave necesaria para un archivo en particular
- Varios nombres de archivo pueden estar asociados a un solo inodo
- Un inodo activo está asociado exactamente a un archivo
- Los atributos de los archivos, los permisos y la información de control se ordenan en el inodo
- En el disco hay una tabla de inodos, o lista de inodos, que contiene los inodos de todos los archivos del sistema de archivos
- Cuando se abre un archivo, su inodo se lleva a la memoria principal y se almacena en una tabla de inodos residente en memoria

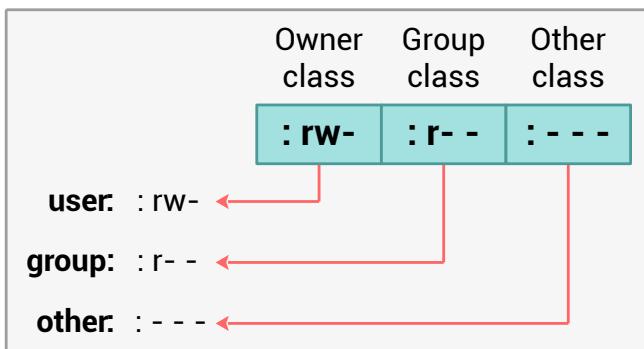
Los directorios se estructuran en un árbol jerárquico

- Pueden contener archivos y/u otros directorios
- Contiene los nombres de los archivos y los punteros a los inodos asociados

La mayoría de los sistemas UNIX dependen, o al menos se basan, en el esquema de control de acceso a archivos introducido con las primeras versiones de UNIX. La figura 32, muestra el enfoque de UNIX para el control de acceso.

Figura 32.

Enfoque tradicional de UNIX (lista de control de acceso mínima)



El control de acceso a archivos implica para el usuario:

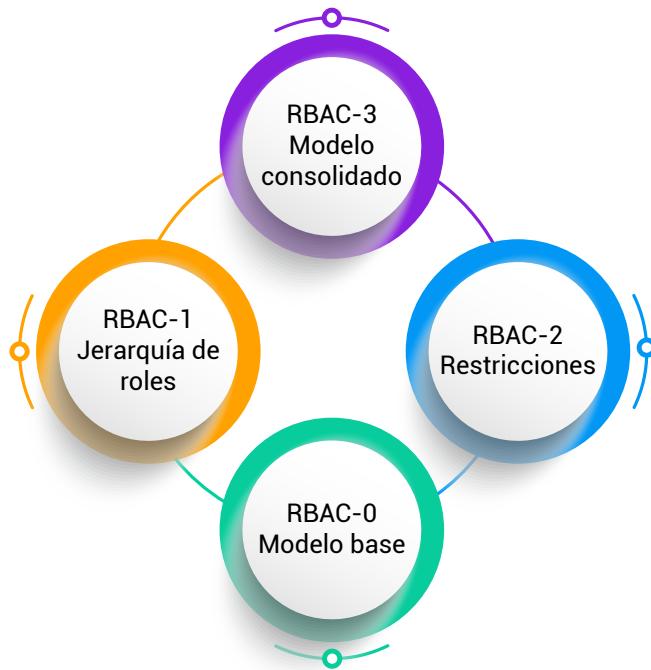
- Número único de identificación de usuario (ID de usuario)
- Miembro de un grupo primario identificado por un ID de grupo
- Pertenece a un grupo específico
- 12 bits de protección
 - Especifican el permiso de lectura, escritura y ejecución para el propietario del archivo, los miembros del grupo y todos los demás usuarios.
- El ID del propietario, el ID del grupo y los bits de protección forman parte del nodo del archivo

3.5. Control de acceso basado en roles (RBAC)

Los sistemas DAC tradicionales definen los derechos de acceso de usuarios individuales y grupos de usuarios. En cambio, el RBAC se basa en los roles que los usuarios asumen en un sistema y no en la identidad del usuario. Normalmente, los modelos RBAC definen un rol como una función de trabajo dentro de una organización. Los sistemas RBAC asignan derechos de acceso a los roles en lugar de a los usuarios individuales. A su vez, los usuarios son asignados a diferentes roles, de forma estática o dinámica, según sus responsabilidades.

RBAC presenta algunos modelos de referencia, con una variedad de funciones y servicios que pueden ser incluidos bajo el enfoque general de RBAC, se presentan cuatro modelos que están relacionados entre sí.

Figura 33.
Modelos de referencia RBAC



RBAC0 contiene la funcionalidad mínima para un sistema RBAC. RBAC1 incluye la funcionalidad de RBAC0 y añade jerarquías de roles, que permiten que un rol herede los permisos de otro rol. RBAC2 incluye RBAC0 y añade restricciones, que restringen las formas en que los componentes de un sistema RBAC pueden ser configurados. RBAC3 contiene la funcionalidad de RBAC0, RBAC1 y RBAC2.

Las restricciones proporcionan un medio para adaptar RBAC a las específicas de las políticas administrativas y de seguridad de una organización. Una restricción es una relación definida entre roles o una condición relacionada con los roles. Se presentan los siguientes tipos de restricciones: roles mutuamente excluyentes, cardinalidad y roles prerequisito

Figura 34.

Tipos de restricciones

Roles mutuamente excluyentes

- Un usuario sólo puede ser asignado a un rol en el conjunto (ya sea durante una sesión o estáticamente)
- Cualquier permiso (derecho de acceso) sólo puede concederse a un rol del conjunto

Cardinalidad

Establecimiento de un número máximo con respecto a los roles

Roles prerrequisito

Dicta que un usuario sólo puede ser asignado a un rol particular si ya está asignado a algún otro rol específico

¡Excelente! Hemos finalizado el estudio de la cuarta unidad, lo invito a desarrollar las actividades de aprendizaje, con el fin de evaluar los conocimientos adquiridos hasta el momento.



Actividad de aprendizaje recomendada

Desarrolle la autoevaluación de la unidad 4 “Control de acceso”, considere que las preguntas planteadas constituyen una estrategia de aprendizaje y tienen como finalidad conocer el grado de asimilación de los contenidos estudiados. En caso de que tenga dificultad para responder alguna pregunta, le recomiendo volver a revisar los contenidos en el texto básico y la guía didáctica virtualizada.



Autoevaluación 4

Lea atentamente las preguntas propuestas en relación con los conceptos de control de acceso y seleccione la opción de respuesta correcta.

1. Elija la opción que corresponda a los tres elementos básicos del control de acceso:
 - a. Sujeto, proceso y derecho de acceso.
 - b. Sujeto, objeto y derecho de acceso.
 - c. Entidad, proceso y control de acceso.
 - d. Entidad, objeto y control de acceso.
2. Existen tres tipos de sujetos, uno de ellos es:
 - a. Grupo.
 - b. Sociedad.
 - c. Personal.
 - d. Particular.
3. ¿Qué tipo de control implementa una política de seguridad que especifica quién o qué puede tener acceso a cada recurso específico del sistema, y el tipo de acceso que se permite en cada instancia?
 - a. Control de auditoría.
 - b. Control de recursos.
 - c. Control del sistema.
 - d. Control de acceso.
4. ¿Qué término define la verificación de que las credenciales de un usuario u otra entidad del sistema son válidas?
 - a. Adecuación.
 - b. Autenticación.
 - c. Autorización.
 - d. Auditoría.

5. La concesión de un derecho o permiso a una entidad del sistema para acceder a un recurso del sistema se denomina:
- Autorización.
 - Autenticación.
 - Control.
 - Supervisión.
6. _____ es el método tradicional para implementar el control de acceso.
- MAC.
 - RBAC.
 - DAC.
 - MBAC.
7. Un _____ es una entidad capaz de acceder a objetos.
- grupo.
 - objeto.
 - sujeto.
 - propietario.
8. Un _____ es un recurso cuyo acceso está controlado.
- objeto.
 - propietario.
 - mundo.
 - sujeto.
9. _____ se basa en los roles que los usuarios asumen en un sistema y no en la identidad del usuario.
- DAC.
 - RBAC.
 - MAC.
 - URAC.

10. _____ proporciona un medio para adaptar el RBAC a las especificidades de las políticas administrativas y de seguridad de una organización.
- a. Restricciones.
 - b. Roles mutuamente excluyentes.
 - c. Cardinalidad.
 - d. Prerrequisitos.

[Ir al solucionario](#)



Si al contestar la autoevaluación su respuesta tuvo resultados positivos ¡FELICITACIONES, SIGA ADELANTE!, caso contrario revise nuevamente el contenido de los ítems errados, para reforzar su aprendizaje. Recuerde que en caso de tener alguna inquietud puede consultar al profesor tutor.



Actividades finales del bimestre



Semana 8

1. Se recomienda revisar nuevamente en el texto básico y la guía didáctica virtualizada, los temas relacionados con la unidad 1, 2, 3 y 4, para ello se recomienda retomar sus apuntes del primer bimestre y prepararse para la evaluación correspondiente.
2. Esquematice el proceso de registro y verificación de usuario en un sistema biométrico; además, revise los objetivos claves de la seguridad informática.
3. Visualice los REA expuestos en el plan docente de los temas abordados en el primer bimestre.
4. Recuerde que, si no alcanzó a participar de la actividad síncrona, está a tiempo de recuperar la misma desarrollando la actividad suplementaria.
5. Espero que haya desarrollado las actividades que se han planificado durante este Primer Bimestre, las cuales son formativas, sumativas (calificadas) y aportan a lo largo de su preparación y autoaprendizaje. Dichas actividades planificadas son Foro, Chat y Cuestionarios que se han configurado en la plataforma académica virtual. Así mismo, sugiero que en la medida de lo posible desarrolle los ejercicios y autoevaluaciones propuestos en el presente curso, que son formativos y sirven como preparación para la evaluación final de bimestre.



¡Felicitaciones! Ha conseguido grandes aprendizajes en el primer bimestre en la asignatura de evaluación de seguridad en sistemas y tecnologías de información, le invito a seguir estudiando con el mismo entusiasmo y dedicación en el segundo bimestre.



Segundo bimestre

Resultado de aprendizaje 5

- Analiza un sistema de almacenamiento de datos propuesto describiendo todos los objetos de información y sus estados durante todo el ciclo de vida del objeto.

Al iniciar el segundo bimestre es necesario tener claro los conceptos básicos de seguridad estudiados en el primer bimestre, como paso previo para estudiar conceptos más avanzados de seguridad.

Contenidos, recursos y actividades de aprendizaje



Semana 9

Unidad 4. Seguridad de la base de datos y del centro de datos

El estudio de la unidad 5, involucra el análisis de los problemas de seguridad exclusivos de las bases de datos, con enfoque en los sistemas de gestión de bases de datos relacionales (RDBMS). Comenzamos con una visión general de la necesidad de técnicas de seguridad específicas para las bases de datos. A continuación, ofrecemos una breve introducción a la gestión de bases de datos, seguida de una visión general de las bases de datos relacionales.

4.1. La necesidad de seguridad en las bases de datos

Según (Stallings, 2018), para muchas empresas y otras organizaciones, es importante poder proporcionar a los clientes, socios y empleados, acceso a esta información. Pero esta información puede ser objeto de amenazas internas y externas de uso indebido o cambio no autorizado. En consecuencia, la seguridad específicamente adaptada a las bases de datos es un componente cada vez más importante de una estrategia global de seguridad de la organización.

Las bases de datos organizacionales tienden a concentrar la información sensible en un único sistema lógico. Algunos ejemplos se muestran en la figura 35.

Figura 35.

Ejemplos de información sensible

-
- 01** Datos financieros de la empresa
 - 02** Registros telefónicos confidenciales
 - 03** Información de clientes y empleados, como el nombre, el número de la Seguridad Social, la información de la cuenta bancaria y la información de la tarjeta de crédito
 - 04** Información sobre productos patentados
 - 05** Información sanitaria e historiales médicos

A pesar de la sensibilidad de la información de las organizaciones, la seguridad de las bases de datos no ha seguido el ritmo de la creciente dependencia de estas, (Stallings, 2018) cita algunas razones, según se detalla a continuación:

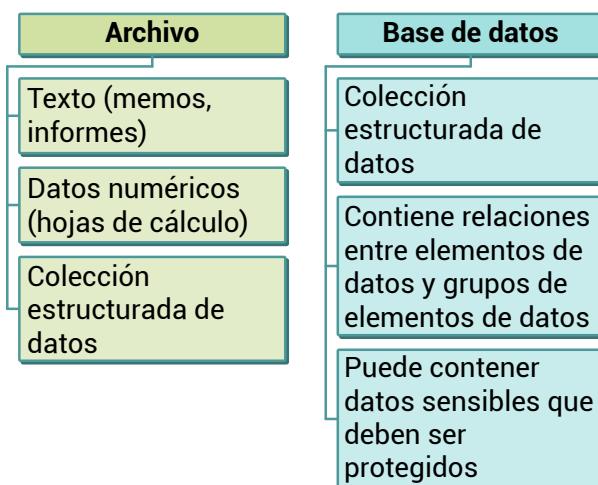
- Existe un dramático desequilibrio entre la complejidad de los modernos sistemas de gestión de bases de datos (DBMS) y la técnica de seguridad utilizada para proteger estos sistemas críticos.
- Las bases de datos tienen un sofisticado protocolo de interacción, el lenguaje de consulta estructurado (SQL), que es complejo.
- Una seguridad eficaz de las bases de datos requiere una estrategia basada en un conocimiento completo de las vulnerabilidades de seguridad de SQL.
- La organización típica carece de personal de seguridad de bases de datos a tiempo completo.

- La mayoría de los entornos empresariales consisten en una mezcla heterogénea de plataformas de bases de datos, plataformas empresariales y plataformas de sistemas operativos, lo que crea un obstáculo de complejidad adicional para el personal de seguridad.
- La creciente dependencia de la tecnología en la nube para alojar parte o la totalidad de la base de datos corporativa.

4.2. Sistemas de gestión de bases de datos

En ocasiones, una organización puede funcionar con una colección relativamente sencilla de archivos de datos, o con archivos más elaborados que consisten en un conjunto de registros; sin embargo, para una organización de cualquier tamaño, es necesario una estructura más compleja conocida como base de datos. La figura 36 muestra la conceptualización de un archivo, así como de una base de datos.

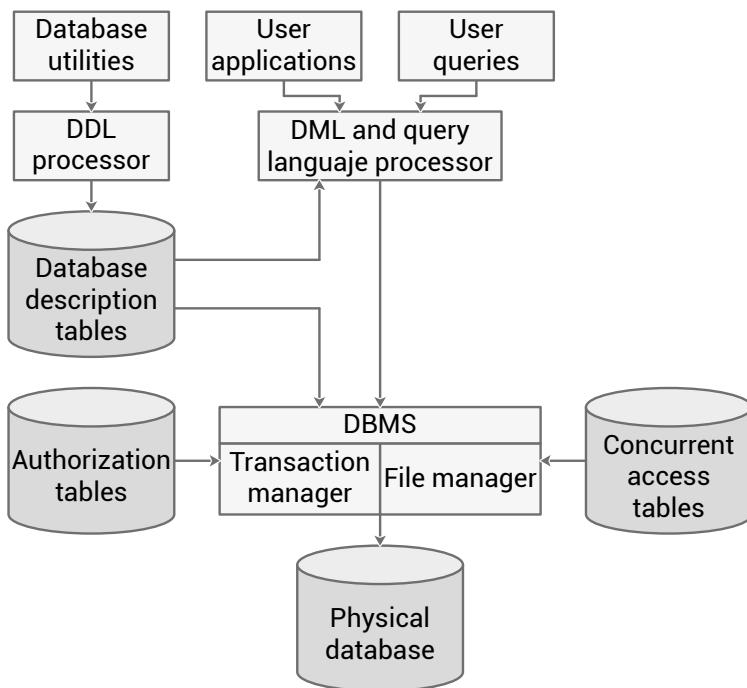
Figura 36.
Conceptualización de archivo y base de datos



Una base de datos requiere de un sistema de gestión de base de datos (DBMS, por sus siglas en inglés), que es un conjunto de programas para construir y mantener la base de datos y para ofrecer facilidades de consulta a múltiples usuarios y aplicaciones. La figura 37, muestra la arquitectura simplificada de un DBMS.

Figura 37.

Arquitectura de un sistema de gestión de base de datos (DBMS)



Nota: Tomado de (Stallings, 2018); DDL = data definition language; DML = data manipulation language

Los sistemas de bases de datos proporcionan un acceso eficiente a grandes volúmenes de datos y son vitales para el funcionamiento de muchas organizaciones. Debido a su complejidad y criticidad los sistemas de bases generan requisitos de seguridad que van más allá de la capacidad de los mecanismos de seguridad típicos basados en el sistema operativo o los paquetes de seguridad independientes.

Los mecanismos de seguridad del sistema operativo suelen controlar el acceso de lectura y escritura a archivos completos, sin embargo, no podrían utilizarse para limitar el acceso a registros o campos específicos de ese archivo. Por el contrario, un BDMS suele permitir especificar este tipo de control de acceso más detallado, además suele permitir que se especifiquen controles de acceso a una gama más amplia de comandos, como seleccionar, insertar, actualizar o eliminar elementos especificados en la base de datos. Por lo tanto, se necesitan servicios y mecanismos de seguridad diseñados específicamente para los sistemas de bases de datos e integrados en ellos.

Profundice los conceptos de un DBMS mediante la lectura comprensiva de la sección 5.2 del texto base.



Semana 10

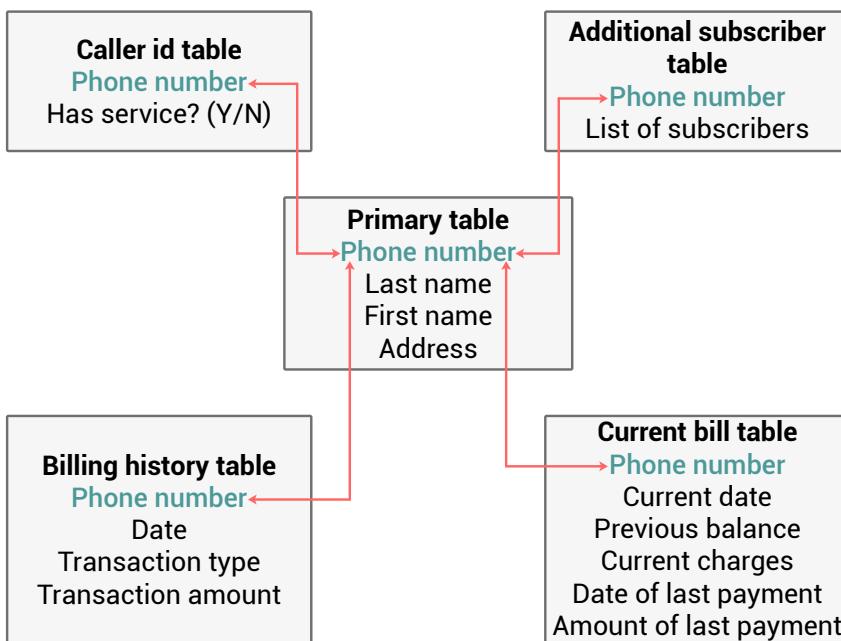
4.3. Bases de datos relacionales

Una base de datos relacional es una tabla de datos, formada por filas y columnas, similar a una hoja de cálculo, cada columna contiene un tipo concreto de datos, mientras que cada fila contiene un valor específico para cada columna. Lo ideal es que la tabla tenga al menos una columna en la que cada valor sea único, sirviendo así de identificador para una entrada determinada. Este tipo de tabla se denomina archivo plano porque es un único archivo bidimensional (filas y columnas). En un archivo plano, todos los datos se almacenan en una sola tabla.

El inconveniente de utilizar una sola tabla es que algunas de las posiciones de las columnas de una fila pueden estar en blanco (no se utilizan). Además, cada vez que se incorpora un nuevo servicio o un nuevo tipo de información, hay que añadir más columnas y rediseñar la base de datos y el software que la acompaña deben ser rediseñados y reconstruidos. Un ejemplo típico es una base de datos telefónica, la figura 38 muestra cómo pueden añadirse nuevos servicios y funciones a la base de datos telefónica sin reconstruir la tabla principal. En este ejemplo, hay una tabla primaria con información básica para cada número de teléfono. El número de teléfono sirve de clave primaria. El administrador de la base de datos puede definir una nueva tabla con una columna para la clave primaria y otras columnas para otra información.

Figura 38.

Ejemplo de base de datos relacional



Nota: Adaptado de (Stalling, 2018)

En el lenguaje de las bases de datos relacionales, el bloque básico de construcción es una relación, que es una tabla plana, las filas son referenciadas como tuplas y las columnas como atributos. En la tabla 11, se detalla la terminología básica utilizada en bases de datos relacionales.

Tabla 11.

Terminología básica en bases de datos relacionales

Nombre formal	Nombre común	También conocido como
Relación	Tabla	Archivo
Tupla	Fila	Registro
Atributo	Columna	Campo

Nota: Adaptado de (Stallings, 2018)

Además, se identificaron algunos elementos de las bases de datos relacionales: primary key (clave primaria), foreign key (clave foránea) y vista o tabla virtual, como se describe en la figura 39.

Figura 39.

Elementos de una base de datos relacional



La figura 40 muestra un ejemplo para clave primaria (primary key) y clave foránea (foreign key), en la que, para la tabla "Department Table", el ID de departamento (Did) es la clave primaria; para la tabla "Employee Table", el ID de empleado es la clave primaria, mientras que el ID de departamento (Did) se constituye en la clave foránea. Note que "Did" que es clave primaria en la tabla "Department Table", es clave foránea en la tabla "Employee Table", esto es lo que permite establecer una relación entre dos tablas.

Figura 40.

Ejemplo de primary key y foreign key

Department Table			Employee Table				
Did	Dname	Dacctno	Ename	Did	Salarycode	Eid	Ephone
4	human resources	528221	Robin	15	23	2345	6127092485
8	education	202035	Neil	13	12	5088	6127092246
9	accounts	709257	Jasmine	4	26	7712	6127099348
13	public relations	755827	Cody	15	22	9664	6127093148
15	services	223945	Holly	8	23	3054	6127092729

Primary key

Foreign key

Primary key

Nota: Tomado de (Stalling 2018)

Una vista es una tabla virtual, como se mencionó en la figura 38, una vista es el resultado de una consulta que devuelve filas y columnas seleccionadas de una o varias tablas, en la figura 39 se presenta una vista que incluye el nombre (Ename), ID (Eid) y número de teléfono (Ephone) del empleado de la tabla "Employee Table" y el nombre del departamento

correspondiente de la tabla “Department table”, tablas mostradas en la figura 41.

Figura 41.

Ejemplo de vista/tabla virtual

Dname	Ename	Eid	Ephone
human resources	Jasmine	7712	6127099348
education	Holly	3054	6127092729
education	Robin	2976	6127091945
accounts	Smith	4490	6127099380
public relations	Neil	5088	6127092246
services	Robin	2345	6127092485
services	Cody	9664	6127093148

Nota: Tomado de (Stalling 2018)

Finalmente, es importante conocer el lenguaje de consulta estructurado (SQL), que es un lenguaje estandarizado que puede utilizarse para definir el esquema, manipular y consultar datos en una base de datos relacional. Existen varias versiones de la norma ANSI/ISO y una variedad de implementaciones diferentes, pero todas siguen la misma sintaxis y semántica básicas.

A continuación, en la figura 42 se presenta el código SQL utilizado para crear las tablas de la figura 40 y la vista de la figura 41.

Figura 42.

Ejemplo de sentencias SQL

Crear tabla "department"

```
CREATE TABLE department (
    Did INTEGER PRIMARY KEY,
    Dname CHAR (30),
    Dacctno CHAR (6))
```

Crear tabla "employee"

```
CREATE TABLE employee (
    Ename CHAR (30),
    Did INTEGER,
    SalaryCode INTEGER,
    Eid INTEGER PRIMARY KEY,
    Ephone CHAR (10),
    FOREIGN KEY (Did) REFERENCES department (Did))
```

Crear vista

```
CREATE VIEW newtable (Dname, Ename, Eid, Ephone)
AS SELECT D.Dname E.Ename, E.Eid, E.Ephone
FROM Department D Employee E
WHERE E.Did = D.Did
```

Nota: Adaptado de (Stalling 2018)

4.4. Ataques de SQL injection (SQLi)

Estimado/a estudiante, le invito a profundizar acerca de los ataques de SQL injection (SQLi)

SQL injection es uno de los ataques de seguridad más frecuentes, que consiste infiltrar código a las bases de datos, aprovechándose de las vulnerabilidades de los sistemas informáticos, a continuación, se presentan algunas características:

- Es una de las amenazas de seguridad basadas en la red más frecuentes y peligrosas.
- Diseñado para explotar la naturaleza de las aplicaciones web.
- Envía comandos SQL maliciosos al servidor de la base de datos.
- El objetivo más común del ataque es la extracción masiva de datos.
- Dependiendo del entorno, SQL injection también puede ser explotada para:

- Modificar o eliminar datos.
- Ejecutar comandos arbitrarios del sistema operativo.
- Lanzar ataques de denegación de servicio (DoS).

Un ataque de SQL injection generalmente sigue la secuencia mostrada en la siguiente figura

Figura 43.

Secuencia de un ataque SQL injection

El hacker encuentra una vulnerabilidad en una aplicación web personalizada e inyecta un comando SQL a una base de datos enviando el comando al servidor web. El comando se inyecta en el tráfico que será aceptado por el firewall.

El servidor web recibe el código malicioso y lo envía al servidor de la aplicación web.

El servidor de aplicaciones web recibe el código malicioso del servidor web y lo envía al servidor de la base de datos.

El servidor de la base de datos ejecuta el código malicioso en la base de datos. La base de datos devuelve los datos de la tabla de tarjetas de crédito.

El servidor de aplicaciones web genera dinámicamente una página con datos que incluyen datos de la tarjeta de crédito de la base de datos.

El servidor web envía los datos de la tarjeta de crédito al hacker.

Los ataques de SQL injection se pueden clasificar por vía de ataque y por tipo de ataque. Las principales vías de ataque se detallan en la figura 44.

Figura 44.

Vías de ataque SQL injection

Entrada de usuario

Los atacantes inyectan comandos SQL proporcionando una entrada de usuario convenientemente manipulada.

Variables del servidor

Los atacantes pueden falsificar los valores que se colocan en las cabeceras HTTP y de red, y explotar esta vulnerabilidad colocando datos directamente en las cabeceras.

Inyección de segundo orden

Un usuario malintencionado podría basarse en datos ya presentes en el sistema o en la base de datos para desencadenar un ataque de SQL injection, de modo que cuando se produce el ataque, la entrada que modifica la consulta para provocar un ataque no procede del usuario, sino del propio sistema.

Cookies

Un atacante podría alterar las cookies de manera que cuando el servidor de aplicaciones construya una consulta SQL basada en el contenido de la cookie, la estructura y la función de la consulta sean modificadas.

Entrada física del usuario

Es posible mediante el suministro de entrada de usuario que construye un ataque fuera del ámbito de las solicitudes web. Esta entrada del usuario podría tomar forma de códigos de barras convencionales, etiquetas RFID, o incluso formularios de papel que son escaneados mediante el reconocimiento óptico de caracteres y transmitidos a un sistema de gestión de bases de datos.

Los ataques SQL injection se pueden clasificar por tipo, en: ataque dentro de banda (inband attack), ataque inferencial y ataque fuera de banda (out-of-band attack). La figura 45 resume los ataques dentro de banda.

Figura 45.

Ataque dentro de banda (inband attack)

Tautología

Esta forma de ataque inyecta código en una o más sentencias condicionales para que siempre se evalúen como verdaderas.

Comentario de fin de línea

Tras inyectar código en un campo concreto, el código legítimo que le sigue se anula mediante el uso de comentarios de fin de línea.

Consulta complementaria

El atacante añade consultas adicionales más allá de la consulta prevista, añadiendo el ataque a una solicitud legítima.

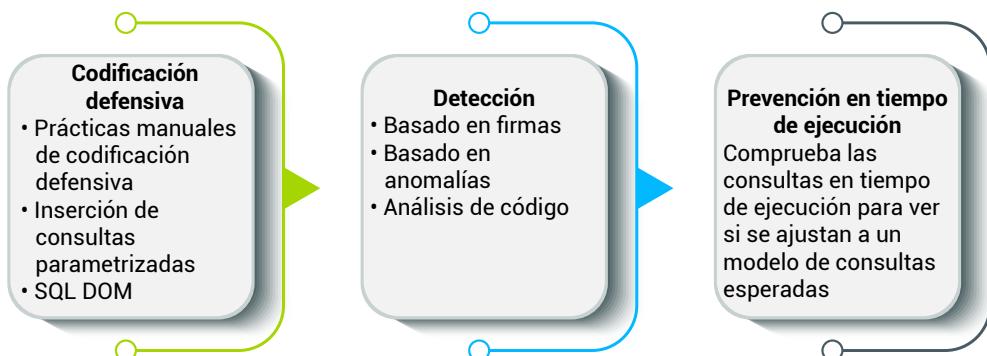
Para el ataque inferencial, a continuación, se describen algunas características importantes:

- No hay una transferencia real de datos, pero el atacante es capaz de reconstruir la información enviando peticiones particulares y observando el comportamiento resultante del servidor del sitio web/base de datos Incluir:
 - Consultas ilegales/lógicamente incorrectas
 - Este ataque permite a un atacante reunir información importante sobre el tipo y la estructura de la base de datos backend de una aplicación web.
 - El ataque se considera un paso preliminar de recopilación de información para otros ataques
 - Inyección SQL ciega
 - Permite a los atacantes inferir los datos presentes en un sistema de base de datos incluso cuando el sistema es lo suficientemente seguro como para no mostrar ninguna información errónea al atacante.

En el caso de un ataque fuera de banda (Out-of-Band Attack), los datos se recuperan utilizando un canal diferente. Se puede usar cuando hay limitaciones en la recuperación de información, cuando la conectividad de salida desde el servidor de la base de datos es deficiente.

Debido a que los ataques SQLi son tan frecuentes, dañinos y variados tanto por vía de ataque como por el tipo de ataque, una sola contramedida es insuficiente. Más bien es necesario un conjunto integrado de técnicas o contramedidas (Stallings, 2018). Estas contramedidas pueden clasificarse en tres tipos: codificación defensiva, detección y prevención en tiempo de ejecución. En la figura 46 se describen estos tres tipos de contramedidas.

Figura 46.
SQLi contramedidas



¡Excelente! Hemos finalizado el estudio de la quinta unidad, lo invito a desarrollar las actividades de aprendizaje, con el fin de evaluar los conocimientos adquiridos hasta el momento.



Actividad de aprendizaje recomendada

Desarrolle la autoevaluación de la unidad 5 “Seguridad de la base de datos”, considere que las preguntas planteadas constituyen una estrategia de aprendizaje y tienen como finalidad conocer el grado de asimilación de los contenidos estudiados. En caso de que tenga dificultad para responder alguna pregunta, le recomiendo volver a revisar los contenidos en el texto básico y la guía didáctica virtualizada.



Autoevaluación 5

Lea atentamente las preguntas propuestas en relación con los conceptos de autenticación de usuarios y seleccione la opción de respuesta correcta.

1. Una _____ es una colección estructurada de datos almacenados para ser utilizados por una o varias aplicaciones.
 - a. atributo.
 - b. base de datos.
 - c. tupla.
 - d. inferencia.

2. Una _____ es una tabla de datos, compuesta por filas y columnas, similar a una hoja de cálculo.
 - a. base de datos relacional.
 - b. conjunto de consultas.
 - c. DBMS.
 - d. perturbación.

3. En el lenguaje de las bases de datos relacionales el bloque de construcción básico es una _____, que es una tabla plana.
 - a. fila.
 - b. tupla.
 - c. clave primaria.
 - d. relación.

4. En una base de datos relacional, las filas se denominan _____.
 - a. relaciones.
 - b. atributos.
 - c. vistas.
 - d. tuplas.

5. Una _____ se define como una porción de una fila que se utiliza para identificar de forma única una fila en una tabla.
- a. clave foránea.
 - b. consulta.
 - c. clave primaria.
 - d. perturbación de datos.
6. Una _____ es una tabla virtual.
- a. tupla.
 - b. consulta.
 - c. vista.
 - d. DBMS.
7. En una base de datos relacional las columnas se denominan _____.
- a. relaciones.
 - b. atributos.
 - c. vistas.
 - d. tuplas.
8. Una _____ es el resultado de una consulta que devuelve filas y columnas seleccionadas de una o varias tablas.
- a. tupla.
 - b. relación.
 - c. clave foránea.
 - d. vista.
9. _____ es un lenguaje estandarizado que puede utilizarse para definir esquemas, manipular y consultar datos en una base de datos relacional.
- a. SQL.
 - b. DBMS.
 - c. DB.
 - d. RBAC.

10. ¿Cuáles son las tres categorías para agrupar los diferentes tipos de ataques SQLi?
- a. *Inband, inferential, out-of-band.*
 - b. *Inband, perturbación, out-of-band.*
 - c. *Command line, inferential, console attack.*
 - d. *Comand line, perturbación, console attack.*

[Ir al solucionario](#)



Si al contestar la autoevaluación su respuesta tuvo resultados positivos ¡FELICITACIONES, SIGA ADELANTE!, caso contrario revise nuevamente el contenido de los ítems errados, para reforzar su aprendizaje. Recuerde que en caso de tener alguna inquietud puede consultar al profesor tutor.

Resultado de aprendizaje 6 y 7

- Compara y contrasta diversas metodologías de ataque y diferencia entre ataques internos y externos.
- Explica cómo los diferentes tipos de ataques afectan a los recursos tecnológicos de una organización.

Contenidos, recursos y actividades de aprendizaje



Semana 11

Unidad 5. Software malicioso

La unidad 6 abarca el estudio de los distintos tipos de software malicioso ampliamente conocido como malware, además su clasificación, sus estrategias de propagación, así como sus objetivos, y finalmente las contramedidas para y de ser el caso recuperación ante un ataque.

5.1. Tipos de software malicioso

El software malicioso o también conocido como malware, constituye posiblemente una de las categorías más importantes de amenazas para los sistemas informáticos.

Pero ¿Qué es el malware?

"Un programa que se inserta en un sistema, normalmente de forma encubierta, con la intención de comprometer la confidencialidad, integridad o disponibilidad de los datos, aplicaciones o el sistema operativo de la víctima o molestarla o perturbarla de alguna manera", (NIST SP 800-83, 2013).

La terminología de software malicioso es amplia, en la tabla 12 se presentan algunos términos y su descripción.

Tabla 12.*Terminología de software malicioso (malware)*

Nombre	Descripción
Advanced Persistent Threat (APT)	Ciberdelincuencia dirigida a objetivos empresariales y políticos, que utiliza una amplia variedad de tecnologías de intrusión y malware, aplicada de forma persistente y eficaz a objetivos específicos durante un período prolongado, a menudo atribuido a organizaciones patrocinadas por un estado.
Adware	Publicidad que se integra en el software. Puede dar lugar a anuncios emergentes o a la redirección de un navegador a un sitio comercial.
Attack kit	Conjunto de herramientas para generar nuevos programas maliciosos de forma automática utilizando una variedad de mecanismos de propagación y carga útil
Auto-router	Herramientas maliciosas de hackers utilizadas para entrar en nuevas máquinas de forma remota
Backdoor (trapdoor)	Cualquier mecanismo que evite una comprobación de seguridad normal; puede permitir el acceso no autorizado a la funcionalidad de un programa, o a la de los demás, o a un sistema comprometido
Downloaders	Código que instala otros elementos en una máquina que está siendo atacada. Normalmente está incluido en el código de malware que se inserta primero en un sistema comprometido para luego importar un paquete de malware más grande.
Drive-by-download	Un ataque que utiliza código en un sitio web comprometido que aprovecha una vulnerabilidad del navegador para atacar un sistema cliente cuando se visualiza el sitio.
Exploit	Código específico para una única vulnerabilidad o conjunto de vulnerabilidades.
Flooders (DoS client)	Utilizados para generar un gran volumen de datos para atacar sistemas informáticos en red, llevando a cabo alguna manera de ataque de denegación de servicio (DoS).
Keyloggers	Captura las pulsaciones del teclado en un sistema comprometido
Logic bomb	Código insertado en el malware por un intruso. Una bomba lógica permanece inactiva hasta que se cumple una condición predefinida; entonces el código activa alguna carga útil
Macro virus	Tipo de virus que utiliza un código de macro o de scripting, normalmente incrustado en un documento o plantilla de documento, y que se activa cuando se visualiza o edita el documento, para ejecutarse y replicarse en otros documentos de este tipo.
Mobile code	Software (por ejemplo, script y macro) que puede enviarse sin cambios a una colección heterogénea de plataformas y ejecutarse con idéntica semántica.
Rootkit	Conjunto de herramientas de hacker utilizadas después de que el atacante haya entrado en un sistema informático y haya obtenido acceso a nivel de raíz.

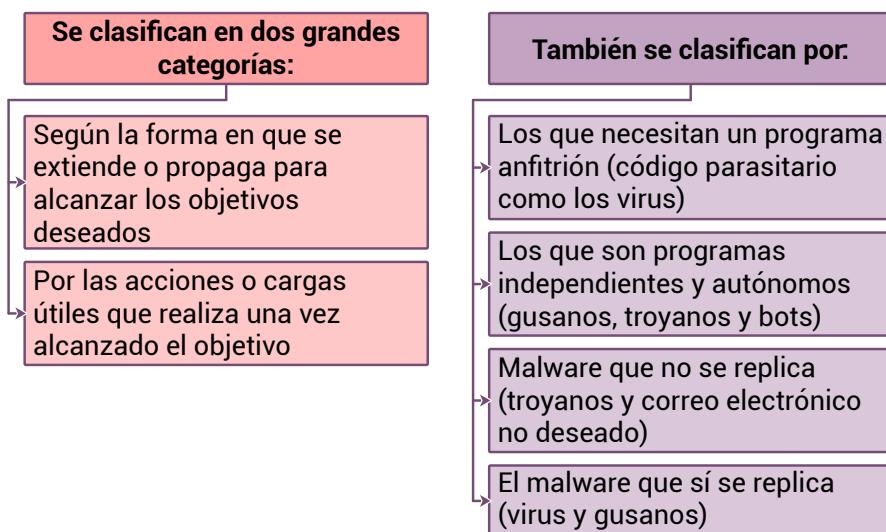
Nombre	Descripción
Spammer programs	Utilizados para enviar grandes volúmenes de correo electrónico no deseado.
Spyware	Software que recoge información de un ordenador y la transmite a otro sistema mediante la monitorización de las pulsaciones del teclado, los datos de la pantalla y/o el tráfico de la red; o mediante el escaneo de los archivos del sistema en busca de información sensible
Trojan horse	Caballo de Troya, es un programa informático que parece tener una función útil, pero que también tiene una función oculta y potencialmente maliciosa que evade los mecanismos de seguridad, a veces explotando las autorizaciones legítimas de una entidad del sistema que lo invoca.
Virus	Malware que, cuando se ejecuta, intenta replicarse en otro código ejecutable de la máquina o script; cuando lo consigue, se dice que el código está infectado. Cuando se ejecuta el código infectado, el virus también se ejecuta
Worm	Gusano, programa informático que puede ejecutarse de forma independiente y puede propagar una versión completa de sí mismo en otros hosts de una red, explotando vulnerabilidades de software en el sistema de destino o utilizando credenciales de autorización capturadas.
Zombie, bot	Programa instalado en una máquina infectada que se activa para lanzar ataques a otras máquinas.

Nota: adaptado de (Stallings, 2018)

El malware supone una amenaza para los programas de aplicación, para los programas de utilidad como editores y compiladores, sitios web y servidores comprometidos o maliciosos, correos electrónicos de spam especialmente diseñados, cuyo objetivo es engañar a los usuarios para que revelen información personal sensible. Esto conlleva a una amplia y variada clasificación de malware por parte de varios autores, sin embargo, en el texto base se muestra una clasificación con un enfoque práctico, tal como se detalla en la figura 47.

Figura 47.

Clasificación de software malicioso (malware)



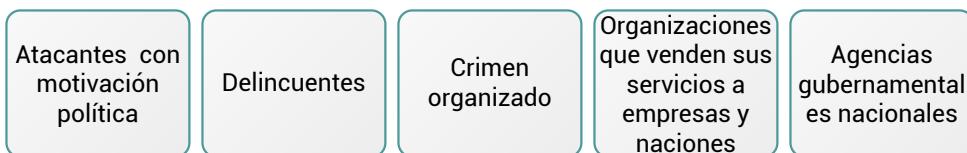
Dada la clasificación de software malicioso, continuaremos revisando algunos de ellos.

Attack kits (kits de ataque)

Aunque al principio, el desarrollo y despliegue de programas maliciosos requería una considerable habilidad técnica por parte de los autores de software, el desarrollo de kits de herramientas para la creación de virus a principios de la década de 1990 y, posteriormente, de kits de ataque más generales en la década de 2000, ha contribuido en gran medida al desarrollo y despliegue de programas maliciosos. Los kits de herramientas se conocen a menudo como "crimeware".

Otra evolución significativa del malware es el cambio de los atacantes, que han pasado de ser individuos a menudo motivados para demostrar su competencia técnica a sus compañeros, a fuentes de ataque más organizadas y peligrosas, como:

Figura 48.
Fuentes de ataque



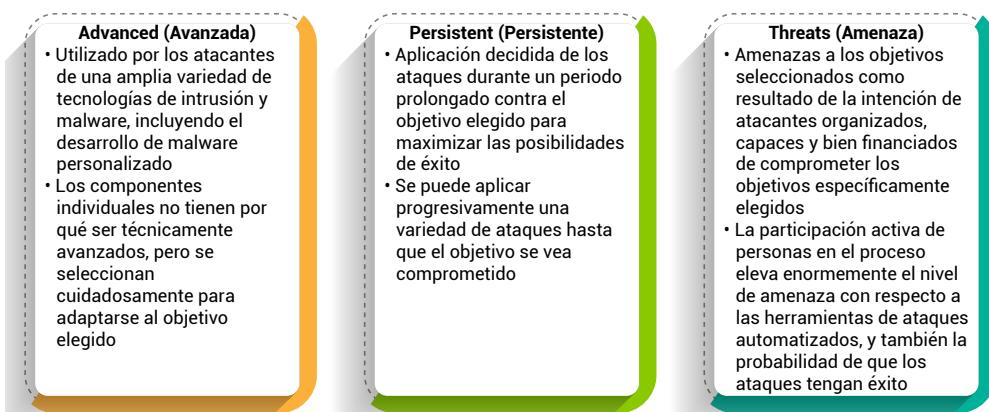
En la sección 6.2 seguiremos revisando algunos tipos de ataque que amenazan a los sistemas informáticos.

5.2. Amenaza persistente avanzada

Las Amenazas Persistentes Avanzadas (APT por sus siglas en inglés) se basa en la aplicación persistente y con recursos de una amplia variedad de tecnologías de intrusión y malware a objetivos seleccionados, normalmente empresariales o políticos. Las APT se atribuyen normalmente a organizaciones patrocinadas por el Estado, algunos ataques probablemente también provengan de empresas criminales.

Las APT se diferencian de otros tipos de ataques por su cuidadosa selección de objetivos y por sus esfuerzos de intrusión persistentes, a menudo sigilosos, durante largos períodos. En la figura 49 se describen las características de las APT.

Figura 49.
Características APT



Nota: Adaptado de (Stallings, 2018)

5.3. Propagación - Contenido infectado – Virus

Las infecciones por virus informáticos constituyeron la mayor parte del malware visto en la primera época de los computadores. El término “virus informático” todavía se utiliza a menudo para referirse al malware en general, más que a los virus informáticos específicamente.

Entonces, ¿Qué es un virus informático?

Un virus informático es una pieza de software que puede “infectar” otros programas, o de hecho cualquier tipo de contenido ejecutable, modificándolo. La modificación incluye la inyección del código original con una rutina para hacer copias del código del virus, que luego puede infectar otros contenidos.

Según (Stallings, 2018), un virus tiene los siguientes componentes: mecanismo de infección, disparador (trigger) y carga útil (payload), más detalle se analiza en la figura 48; además, durante su vida, un virus típico pasa por las cuatro fases descritas en la figura 40.

Figura 50.

Componentes de un virus

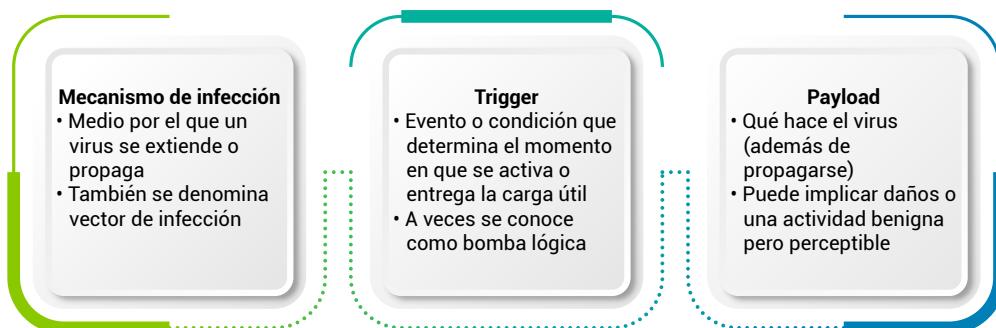
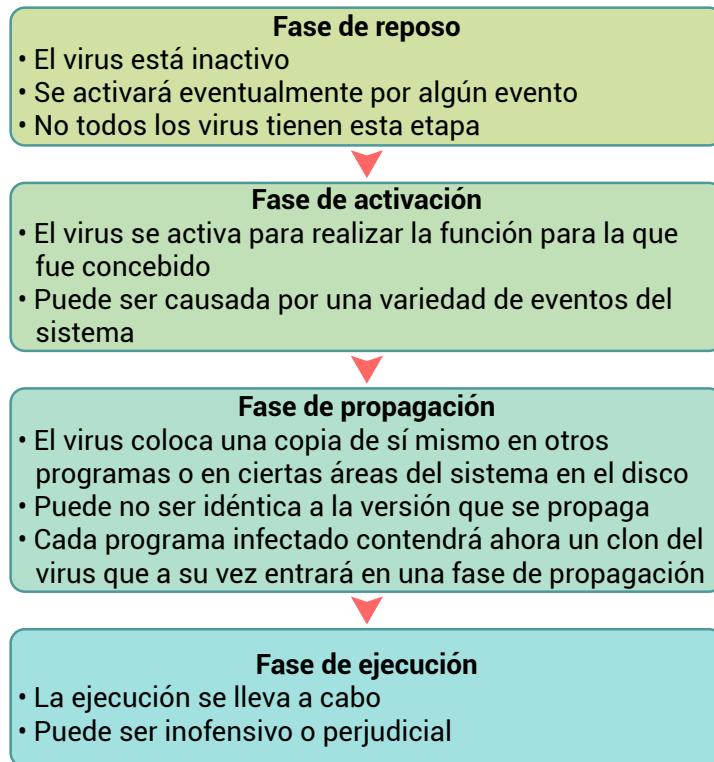


Figura 51.

Fases de un virus



Finalmente, para clasificar los virus, no existe un esquema de clasificación simple o universalmente acordado para los virus. (Stallings, 2018), clasifica los virus según dos ejes: el tipo de objetivo que el virus intenta infectar, y la estrategia que utiliza el virus para ocultarse de la detección de los usuarios y del software antivirus. Esta clasificación se muestra en la tabla 13.

Tabla 13.

Clasificación de los virus

Por objetivo	Por estrategia de ocultación
Infector del sector de arranque Infecta un registro de arranque maestro o registro de arranque y se propaga cuando un sistema se inicia desde el disco que contiene el virus.	Virus encriptado Una parte del virus crea una clave de cifrado aleatoria y cifra el resto del virus.
Infector de archivos Infecta archivos que el sistema operativo o el shell consideran ejecutables.	Virus furtivo Una forma de virus diseñada explícitamente para ocultarse de la detección del software antivirus.

Por objetivo	Por estrategia de ocultación
Macro virus Infecta archivos con código de macros o scripts que son interpretados por una aplicación.	Virus polimórfico Un virus que muta con cada infección.
Virus multipartito Infecta archivos de múltiples maneras.	Virus metamórfico Un virus que muta y se reescribe completamente en cada iteración y puede cambiar tanto su comportamiento como su apariencia.

"Desde que aparecieron los virus, ha habido una continua competencia entre los creadores de virus y los creadores de software antivirus. A medida que se desarrollan contramedidas eficaces para tipos de virus existentes, se desarrollan otros virus nuevos. "



Semana 12

5.4. Propagación - Explotación de vulnerabilidades – Gusanos

Es momento de hablar de los gusanos informáticos. Un gusano es un programa que busca activamente más máquinas para infectar, y luego cada máquina infectada sirve como plataforma de lanzamiento automatizada para los ataques a otras máquinas. Los programas de gusanos aprovechan las vulnerabilidades del software en los programas cliente o servidor para acceder a cada nuevo sistema. Pueden utilizar las conexiones de red para propagarse de un sistema a otro. También pueden propagarse a través de medios compartidos, como unidades USB o discos de datos de CD y DVD. Los gusanos de correo electrónico pueden propagarse en códigos de macros o scripts incluidos en documentos adjuntos al correo electrónico o en transferencias de archivos de mensajería instantánea. Tras su activación, el gusano puede replicarse y propagarse de nuevo.

Figura 52.
Replicación de gusanos

Correo electrónico o mensajería instantánea	• El gusano envía por correo electrónico una copia de sí mismo a otros sistemas • Se envía a sí mismo como un archivo adjunto a través de un servicio de mensajería instantánea
Compartir archivos	Crea una copia de sí mismo o infecta un archivo como virus en un medio extraíble
Capacidad de ejecución remota	El gusano ejecuta una copia de sí mismo en otro sistema
Capacidad de acceso o transferencia remota de archivos	El gusano utiliza un servicio de acceso o transferencia remota de archivos para copiarse a sí mismo de un sistema a otro
Capacidad de inicio de sesión remoto	El gusano se registra en un sistema remoto como usuario y luego utiliza comandos para copiarse a sí mismo de un sistema a otro

Los gusanos informáticos por años se han convertido en una amenaza para las organizaciones, unos han logrado acceso a los sistemas informáticos, comprometiendo la información sensible de las organizaciones. A continuación, se detalla la historia y evolución de los gusanos informáticos.

[Historia de los gusanos informáticos](#)

Uno de los gusanos más poderosos y populares de los últimos tiempos, es ransomware, este es un gusano del tipo WannaCry, que ha causado grandes pérdidas económicas a diferentes organizaciones. La figura 53 muestra un resumen de las principales características de este gusano.

Figura 53.

Ransomware (WannaCry)

Ataque de ransomware en mayo de 2017 que se extendió con extrema rapidez durante un periodo de horas a días, infectando cientos de miles de sistemas pertenecientes a organizaciones tanto públicas como privadas en más de 150 países.

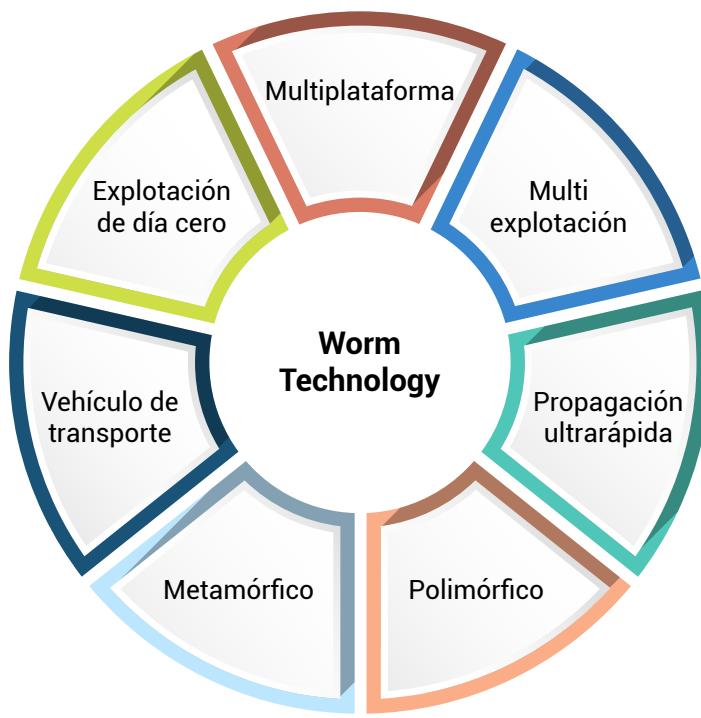
Se propagó como un gusano mediante el escaneo agresivo tanto de redes locales como de redes remotas aleatorias, intentando explotar una vulnerabilidad en el servicio de intercambio de archivos SMB en sistemas Windows no parcheados.

Esta rápida propagación sólo se vio frenada por la activación accidental de un dominio "kill-switch" por parte de un investigador de seguridad del Reino Unido.

Una vez instalado en los sistemas infectados, también encriptaba los archivos, exigiendo el pago de un rescate para recuperarlos.

A lo largo del tiempo los gusanos informáticos han aprovechado la tecnología para mejorar su estrategia de ocultación y lograr sus objetivos antes de ser detectados, en la figura 54 se muestran algunos elementos tecnológicos adoptados por los gusanos.

Figura 54.
Tecnología utilizada por gusanos



- Multiplataforma: Los gusanos más recientes no se limitan a las máquinas Windows, sino que pueden atacar una variedad de plataformas, especialmente las variedades populares de UNIX; o explotar lenguajes de macros o scripts soportados en tipos de documentos populares.
- Multi explotación: Los nuevos gusanos penetran en los sistemas de diversas maneras, utilizando explotaciones contra servidores web, navegadores, correo electrónico, intercambio de archivos y otras aplicaciones o a través de medios compartidos.
- Propagación ultrarrápida: Explotan diversas técnicas para optimizar la velocidad de propagación de un gusano para maximizar su probabilidad de localizar el mayor número de máquinas vulnerables como sea posible en un corto período de tiempo.
- Polimórfico: Para evadir la detección, saltarse los filtros y frustrar el análisis en tiempo real, los gusanos adoptan técnicas polimórficas de virus. Cada copia del gusano tiene un nuevo código generado sobre

la marcha utilizando instrucciones funcionalmente equivalentes y técnicas de encriptación.

- Metamórfico: Además de cambiar su apariencia, los gusanos metamórficos tienen un repertorio de patrones de comportamiento que se desencadenan en diferentes etapas de propagación.
- Vehículo de transporte: Como los gusanos pueden comprometer rápidamente un gran número de sistemas, son ideales para propagar una amplia variedad de cargas útiles maliciosas, como bots de denegación de servicio distribuidos, rootkits, generadores de spam y spyware.
- Explotación de día cero: Para lograr la máxima sorpresa y distribución, un gusano debe explotar una vulnerabilidad desconocida que solo es descubierta por la comunidad de la red en general cuando se lanza el gusano.



Le invito a conocer otros tipos de gusanos informáticos como: mobile code, mobile phone, drive-by-download, watering-hole, malvertising y clickjacking, tanto en el texto base, capítulo 6, como en bibliografía complementaria.

5.5. Propagación - Ingeniería social - Correo electrónico no deseado, troyanos

La última categoría de propagación de malware implica la ingeniería social, "engaño" a los usuarios para que ayuden a comprometer sus propios sistemas o información personal. Esto puede ocurrir cuando un usuario ve y responde a algún correo SPAM, o permite la instalación y ejecución de algún programa troyano o código de scripting (Stallings, 2018). La figura X muestra las categorías de malware que buscan lograr su objetivo a través de ingeniería social.

Figura 55.

Categorización de propagación de malware mediante ingeniería social



Correo electrónico spam (no solicitado)

Con el crecimiento explosivo de Internet en las últimas décadas, el uso generalizado del correo electrónico y el bajísimo coste necesario para enviar grandes volúmenes de correos electrónicos se ha producido el aumento del correo electrónico masivo no solicitado, conocido comúnmente como spam. Según (Stallings, 2018) señala que más de la mitad del tráfico de correo electrónico empresarial entrante sigue siendo spam, a pesar de haber disminuido gradualmente en los últimos años. Esto impone costes significativos tanto a la infraestructura de red necesaria para retransmitir este tráfico, como a los usuarios que necesitan filtrar sus correos electrónicos legítimos de esta avalancha de correos. En respuesta a este crecimiento explosivo, se ha producido un crecimiento igualmente rápido de la industria antispam que ofrece productos para detectar y filtrar los correos electrónicos basura. Esto ha dado lugar a una competencia entre los remitentes de spam que idearon técnicas para colar sus contenidos y los defensores, que se esfuerzan por bloquearlos mediante el uso de antispam.

Caballos de Troya

Un caballo de Troya es un programa o utilidad útil, o aparentemente útil, que contiene código oculto que, al ser invocado, realiza alguna función no deseada o dañina. Los programas troyanos pueden utilizarse para efectuar funciones indirectamente que el atacante no podría hacer directamente. Por ejemplo, para obtener acceso a información personal sensible almacenada en los archivos de un usuario, un atacante podría crear un programa troyano que, al ejecutarse, escaneé los archivos del usuario en busca de la información sensible deseada y envíe una copia de esta al atacante a través de un formulario web o de un correo electrónico o mensaje de texto.

El autor podría entonces incitar a los usuarios a ejecutar el programa incorporándolo a un juego o a un programa de utilidad, y poniéndolo a disposición a través de un sitio de distribución de software conocido o de una tienda de aplicaciones. Este enfoque se ha usado recientemente con utilidades que "dicen" ser el último escáner antivirus, o la última actualización de seguridad, para los sistemas, pero que en realidad son troyanos maliciosos, a menudo con cargas útiles como programas espía que buscan credenciales bancarias. Por ello, los usuarios deben tomar precauciones para validar el origen de cualquier software que instalen.

Los troyanos se ajustan a uno de estos tres modelos:

- Continúan realizando la función del programa original y además realizan una actividad maliciosa independiente.
- Continúan efectuando la función del programa original, pero modificando la función para realizar una actividad maliciosa o para disfrazar otra actividad maliciosa.
- Realizar una función maliciosa que sustituya completamente la función del programa original.

Troyanos para teléfonos móviles

Los troyanos para teléfonos móviles también aparecieron por primera vez en 2004 con el descubrimiento de Skaller. Al igual que los gusanos para móviles, el objetivo es el smartphone, y los primeros troyanos para móviles tenían como objetivo los teléfonos Symbian. Más recientemente, se ha detectado un número importante de troyanos dirigidos a teléfonos Android y a iPhones de Apple. Estos troyanos suelen distribuirse a través de uno o varios mercados de aplicaciones para el sistema operativo del teléfono objetivo.

El rápido crecimiento de las ventas y el uso de los teléfonos inteligentes, que contienen cada vez más información personal valiosa, los convierten en un objetivo atractivo para los delincuentes y otros atacantes. Según (Stallings, 2018), cinco de cada seis teléfonos nuevos funcionan con Android, siendo estos su objetivo clave. El número de vulnerabilidades descubiertas y las familias de malware dirigidas a estos teléfonos han aumentado constantemente en los últimos años. Algunos ejemplos recientes son un troyano de phishing que engaña al usuario para que introduzca sus datos

bancarios y un ransomware que imita el estilo de diseño de Google para parecer más legítimo e intimidante.

Los controles más estrictos que Apple impone en su tienda de aplicaciones hacen que muchos troyanos para iPhone se dirijan a teléfonos "rooteados" y se distribuyan a través de sitios no oficiales. Sin embargo, varias versiones del sistema operativo del iPhone contenían algún tipo de vulnerabilidad gráfica o PDF. De hecho, estas vulnerabilidades eran el principal medio utilizado para "liberar" los teléfonos. Pero también proporcionaban una vía que el malware podría emplear para atacar los teléfonos. Aunque Apple ha corregido varias de estas vulnerabilidades, se han seguido descubriendo nuevas variantes.

Enfoques de contramedidas contra el malware

La solución ideal a la amenaza del malware es la prevención.

Figura 56.

Prevención de malware

Elementos principales de prevención
<ul style="list-style-type: none">• Políticas• Concienciación• Mitigación de la vulnerabilidad• Mitigación de amenazas

Si la prevención falla, se pueden utilizar mecanismos técnicos para apoyar las siguientes opciones de mitigación de amenazas:

- Detección: Una vez producida la infección, determinar que se ha producido y localizar el malware.
- Identificación: Una vez realizada la detección, identificar el malware específico que ha infectado el sistema.
- Eliminación: Una vez identificado el malware específico, eliminar todos los rastros del virus malicioso de todos los sistemas infectados para que no pueda seguir propagándose.

Si la detección tiene éxito, pero no es posible la identificación o la eliminación, la alternativa es descartar los archivos infectados o maliciosos y volver a cargar una versión de copia de seguridad limpia. En el caso de algunas infecciones particularmente desagradables, esto puede requerir un

borrado completo del almacenamiento y reconstruir el sistema infectado a partir de medios limpios conocidos.

Finalmente, otra de las contramedidas de prevención es el uso de software antivirus, su uso en los computadores personales está ahora muy extendido, en parte por el crecimiento explosivo del volumen y la actividad del malware. En la figura 57, se esquematiza la evolución del software antivirus a través de sus distintas generaciones.

Figura 57.
Evolución de software antivirus

Primera generación: escáneres simples

- Requiere una firma de malware para identificar el malware
- Limitado a la detección de malware conocido

Segunda generación: escáneres heurísticos

- Utiliza reglas heurísticas para buscar probables casos de malware
- Otro enfoque es la comprobación de la integridad

Tercera generación: trampas de actividad

Programas residentes en memoria que identifican el malware por sus acciones y no por su estructura en un programa infectado

Cuarta generación: protección completa

- Paquetes que consisten en una variedad de técnicas antivirus utilizadas en conjunto
- Incluyen componentes de escaneo y trampas de actividad y capacidad de control de acceso

Otras contramedidas incluyen: análisis sandbox (caja de arena), análisis dinámico de malware basado en el host, detección y eliminación de spyware, y escaneo perimetral. Le invito a profundizar estos conceptos en la sección 6.10 del texto base.

¡Excelente! Hemos finalizado el estudio de la sexta unidad, lo invito a desarrollar las actividades de aprendizaje, con el fin de evaluar los conocimientos adquiridos hasta el momento.



Actividad de aprendizaje recomendada

Desarrolle la autoevaluación de la unidad 6 "Software malicioso", considere que las preguntas planteadas constituyen una estrategia de aprendizaje y tienen como finalidad conocer el grado de asimilación de los contenidos estudiados. En caso de que tenga dificultad para responder alguna pregunta, le recomiendo volver a revisar los contenidos en el texto básico y la guía didáctica virtualizada.



Autoevaluación 6

Lea atentamente las preguntas propuestas en relación con los conceptos de autenticación de usuarios y seleccione la opción de respuesta correcta.

1. Un programa que se inserta de forma encubierta en un sistema, con la intención de comprometer la integridad o la confidencialidad de los datos de la víctima se denomina _____.
 - a. *Bomber man.*
 - b. Animoto.
 - c. Malware.
 - d. Prezi.

2. Un _____ es un conjunto de programas que se instalan en un sistema para mantener un acceso encubierto a ese sistema, con privilegios de administrador (*root*) mientras se ocultan las pruebas de su presencia.
 - a. *Rootkit.*
 - b. *Toolkit.*
 - c. *Bootkit.*
 - d. *Coolkit.*

3. _____ es un código insertado en un *malware* que permanece inactivo hasta que se cumple una condición predefinida que desencadena un acto no autorizado.
 - a. Bomba lógica.
 - b. Trampilla.
 - c. Gusano.
 - d. Troyano.

4. _____ es el componente de un virus informático que ejecuta una actividad maliciosa.
 - a. Mecanismo de infección.
 - b. Desencadenante.
 - c. Bomba lógica D.
 - d. Carga útil.

5. La _____ es cuando se realiza la función del virus.
- fase de latencia.
 - fase de propagación.
 - fase de activación.
 - fase de ejecución.
6. Durante la _____ el virus está inactivo.
- fase de reposo.
 - fase de propagación.
 - fase de activación.
 - fase de ejecución.
7. El correo electrónico masivo no solicitado se denomina _____.
- spam*.
 - propagación.
 - phishing*.
 - crimeware*.
8. _____ es un *malware* que encripta los datos del usuario y exige un pago para acceder a la clave necesaria para recuperar la información.
- Caballo de Troya.
 - Ransomware*.
 - Crimeware*.
 - Polimórfico.
9. Un ataque _____ es un ataque de *bots* a un sistema informático o a una red que causa una pérdida de servicio a los usuarios.
- spam*.
 - phishing*.
 - DDoS.
 - sniff*.

10. La solución ideal a la amenaza del *malware* es _____.

- a. la identificación.
- b. la eliminación.
- c. detección.
- d. prevención.

[Ir al solucionario](#)



Si al contestar la autoevaluación su respuesta tuvo resultados positivos ¡FELICITACIONES, SIGA ADELANTE!, caso contrario revise nuevamente el contenido de los ítems errados, para reforzar su aprendizaje. Recuerde que en caso de tener alguna inquietud puede consultar al profesor tutor.

Resultado de aprendizaje 8

- Discute las políticas y prácticas que pueden ser aplicadas a la integración de sistemas y arquitecturas para garantizar un funcionamiento seguro del sistema y el aseguramiento de la información.

Cuando se escribe programas informáticos, existen muchas deficiencias que dan como resultado la subversión de los mecanismos de seguridad y, permiten el acceso y uso no autorizado de los datos y recursos informáticos; por ejemplo, los desbordamientos de buffer, que es una de las vulnerabilidades de software más comunes y ampliamente explotadas, es por ello que, los programas deben escribirse de forma segura para evitar que ocurran tales vulnerabilidades.

En la presente unidad exploraremos el tema general de la seguridad del software. Abordaremos un modelo simple de un programa informático que ayuda a identificar dónde pueden ocurrir problemas de seguridad. Luego, exploramos el tema clave de cómo manejar correctamente la entrada del programa para prevenir muchos tipos de vulnerabilidades y, de manera más general, cómo escribir código de programa seguro y administrar las interacciones con otros programas y el sistema operativo.

Contenidos, recursos y actividades de aprendizaje



Semana 13

Unidad 6. Seguridad del Software

6.1. Problemas de seguridad del Software



La programación defensiva o segura, es el proceso de diseñar e implementar software para que continúe funcionando incluso cuando está bajo ataque (Stallings, 2018).

La mayoría de vulnerabilidades de seguridad informática resultan de malas prácticas de programación, que según el Informe sobre el estado de la seguridad del software de Veracode (Veracode, 2016) son mucho más frecuentes de lo que podamos imaginar. La lista de los 25 errores de software más peligrosos según la "Enumeración de debilidades comunes CWE" - Common Weakness Enumeration, detalla la opinión de consenso sobre las malas prácticas de programación que son la causa de la mayoría de los ataques ciberneticos. Estos errores se agrupan en tres categorías: interacción insegura entre componentes, gestión de recursos arriesgada y defensas porosas, el contenido resumido se detalla en la Tabla 15 (Stallings, 2018).

Estimado/a estudiante, le invito a revisar el recurso acerca de categorías de errores sobre las malas prácticas de programación.

Categorías de errores sobre las malas prácticas de programación

Continuemos con el aprendizaje conociendo más sobre OWAS

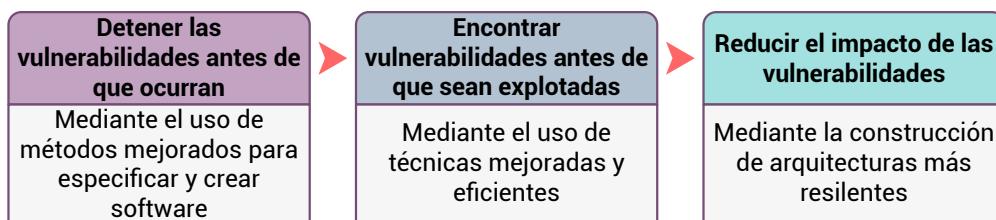
El Proyecto de Seguridad de Aplicaciones Web Abiertas (OWAS) presenta la lista Top Ten (OWAS, 2013) de fallas críticas de seguridad de aplicaciones web incluye cinco relacionadas con código de software inseguro. Estos incluyen:

- a. Entrada no validada
- b. Secuencias de comandos entre sitios
- c. Desbordamiento de búfer
- d. Fallas de inyección
- e. Manejo inadecuado de errores

Estas fallas ocurren como consecuencia de una verificación y validación insuficientes de los datos y códigos de error en los programas. El conocimiento de estos problemas, es un paso inicial y crítico para escribir un código de programa más seguro. Ambas fuentes enfatizan en que los desarrolladores de software aborden estas áreas de preocupación conocidas y brindan orientación sobre cómo mitigarlas. El informe "Reducción drástica de vulnerabilidades de software" (Black, 2016) presenta una variedad de enfoques con el objetivo de reducir drásticamente la cantidad de vulnerabilidades de software; la figura 58 resume las recomendaciones.

Figura 58.

Recomendaciones para reducir la cantidad de vulnerabilidades en el software



La seguridad del software está estrechamente relacionada con la calidad y la confiabilidad del software, pero con sutiles diferencias. **La calidad y la confiabilidad del software se relacionan con la falla accidental de un programa** como resultado de alguna entrada teóricamente aleatoria, no anticipada, interacción del sistema o uso de código incorrecto. Se espera que estos fracasos sigan algún tipo de distribución de probabilidad. El enfoque habitual para mejorar la calidad del software es utilizar alguna forma de diseño y prueba estructurados para identificar y eliminar tantos errores como sea razonablemente posible de un programa. La prueba generalmente involucra variaciones de entradas probables y errores comunes, con la intención de minimizar la cantidad de errores que se verían en el uso general. La preocupación no es la cantidad total de errores en un programa, sino la frecuencia con la que se activan, lo que ocasiona una falla del programa.

La seguridad del software difiere en que el atacante elige la distribución de probabilidad, enfocándose en errores específicos que resultan en una falla que puede ser aprovechada por el atacante. Estos errores a menudo pueden desencadenarse por entradas que difieren drásticamente de lo que generalmente se espera y, por lo tanto, es poco probable que se identifiquen mediante enfoques de prueba comunes. Escribir código seguro requiere prestar atención a todos los aspectos de cómo se ejecuta un programa, el entorno en el que se ejecuta y el tipo de datos que procesa, no se puede suponer nada y todos los errores potenciales deben verificarse.



Recuerde: La regla clave en la programación defensiva es nunca asumir nada, sino verificar todas las suposiciones y manejar cualquier posible estado de error.

Ahora bien, reflexionemos un poco respecto a la “programación defensiva”; cuando escribimos un programa, generalmente nos enfocamos en la resolución de un problema, en este sentido, nuestra atención y esfuerzos técnicos, están enfocados en desarrollar los pasos necesarios para que el programa en progreso, tenga el éxito que buscamos; si bien hacemos suposiciones o validaciones sobre el tipo de entradas que va a recibir nuestro software, estas validaciones buscan anticipar, verificar y manejar de manera correcta los posibles errores o situaciones que se pudieran presentar, pero, esta dinámica de enfocarnos “fuera del objetivo” de nuestro programa, requerirá incrementar la cantidad de código y tiempo necesario para la entrega de un proyecto, lo cual, lógicamente, puede entrar en conflicto con presiones comerciales de la empresa.

Adicional, cuando se requieren cambios en un programa, el técnico se enfoca en la solución necesaria para cubrir los nuevos requerimientos y, la “programación defensiva” puede quedar de lado o, en un segundo plano. Bajo este enfoque analítico general, la “programación defensiva” requiere un cambio de mentalidad respecto a las prácticas de programación tradicionales, esto significa que el programador necesita ser consciente de las consecuencias de las fallas en su código y, las técnicas que usan normalmente los atacantes. La necesidad de que la seguridad y la confiabilidad sean objetivos de diseño desde el inicio de un proyecto ha sido reconocida por la mayoría de las disciplinas de ingeniería, para muestra, tenemos como base una serie de estándares de calidad y desarrollo de software como ISO 12207 (Procesos del ciclo de vida del software, 1997); como complemento, podemos citar al Foro para aseguramiento de la excelencia en el código (SAFECode, 2011); estas recomendaciones, son parte del estudio de esta unidad.

En este punto de nuestro análisis, debemos convenir en algo, nuestro estudio se centrará en los problemas observados en las aplicaciones web; si tomamos en cuenta que, muchas veces, el rápido desarrollo de este tipo de aplicaciones, por lo general por parte de desarrolladores sin mayor conocimiento de temas de seguridad, da como resultado aplicaciones particularmente vulnerables, sin embargo, los principios que discutiremos en esta unidad, son aplicables a todo tipo de programas, siempre es posible que una utilidad sencilla, diseñada para uso local, pueda ser extrapolada a aplicaciones más grandes, con preocupaciones de seguridad, en el fondo, diferentes.

6.2. Manejo de la entrada del programa

La entrada del programa hace referencia a cualquier fuente de datos que se origina fuera del programa y, cuyo valor no es conocido explícitamente por el programador cuando escribió el código, si bien esta definición incluye datos leídos en el programa desde la entrada de teclado o mouse del usuario, archivos o conexión de red, también incluye los datos proporcionados al programa en el momento de la ejecución. Las claves de preocupación sobre las cuales debe poner atención el programador con:

- a. El tamaño de la entrada
- b. El significado e interpretación de la entrada

El tamaño de la entrada

Por lo general, al leer o copiar entradas de una fuente, el programador hace suposiciones del tamaño máximo esperado de la entrada; a menudo se supone que esta entrada no excederá unas pocas líneas de tamaño, como consecuencia, el programador asigna un búfer de memoria (por lo general, 512 o 1024 bytes) para almacenar esta entrada y, así mismo, por lo general, no verifica para confirmar no es mayor al tamaño esperado. Si esta situación se presenta, se produce un desbordamiento de búfer.

Escribir código que sea seguro contra desbordamientos de búfer requiere una mentalidad que considere cualquier entrada como peligrosa y la procese de una manera que no exponga el programa al peligro. **La alternativa al tamaño de la entrada, sería usar un búfer de tamaño dinámico para garantizar que haya suficiente espacio disponible o procesar la entrada en bloques del tamaño de un búfer.** Incluso si se utilizan búferes de tamaño dinámico, es necesario tener cuidado para asegurarse de que el espacio solicitado no excede la memoria disponible. Si esto ocurre, el programa debe manejar este error correctamente, por ejemplo, aplicar el procesamiento de la entrada en bloques, descartar el exceso de entrada, terminar el programa o cualquier otra acción que sea razonable en respuesta a tal situación anormal. Estas comprobaciones deben aplicarse siempre que datos cuyo valor se desconozca o sean manipulados por el programa. También deben aplicarse a todas las fuentes potenciales de insumos.

Interpretación de la entrada del programa

La clave de esta verificación, es el significado e interpretación de la entrada en un programa; comúnmente, los programas procesan datos textuales como entrada. Para ilustrar los problemas con la interpretación de los datos de entrada textuales, analizaremos la clase general de **ataques de inyección**, quienes aprovechan la falla para validar la interpretación de la entrada.

Ataques de inyección. – Este término se refiere a una amplia variedad de fallas en el programa relacionadas con el manejo no válido de los datos de entrada. Puntualmente, este problema ocurre cuando los datos de entrada de un programa influyen de manera accidental o deliberadamente en el flujo de ejecución de un programa, por ejemplo, cuando los datos se pasan como un parámetro a otro programa auxiliar en el sistema, cuya salida luego es procesada y utilizada por el programa original.

A continuación, un ejemplo de un ataque de inyección (Stallings, 2018); considere la secuencia de comandos perl CGI de ejemplo que se muestra en la figura 59, que está diseñada para devolver algunos detalles básicos sobre el usuario especificado utilizando el comando finger de UNIX. Este script se colocaría en una ubicación adecuada en el servidor web y se invocaría en respuesta a un formulario simple, como el que se muestra en la figura 60. El script recupera la información deseada ejecutando un programa en el sistema del servidor y devolviendo el resultado de ese programa, reformateado adecuadamente si es necesario, en una página web HTML.

Figura 59.

Script inseguro – Perl finger CGI

```
1 #!/usr/bin/perl
2 # finger.cgi - finger CGI script using Perl5 CGI module
3
4 use CGI;
5 use CGI::Carp qw(fatalsToBrowser);
6 $q = new CGI; # create query object
7
8 # display HTML header
9 print $q->header,
10 $q->start_html('Finger User'),
11 $q->h1('Finger User');
12 print "<pre>";
13
14 # get name of user and display their finger details
15 $user = $q->param("user");
16 print `/usr/bin/finger -sh $user`;
17
18 # display HTML footer
19 print "</pre>";
20 print $q->end_html;
```

Este tipo de formulario simple y el controlador asociado fueron ampliamente vistos y, a menudo, se presentaron como ejemplos simples de cómo escribir y usar scripts CGI. Desafortunadamente, este script contiene una vulnerabilidad crítica. El valor del usuario se pasa directamente al programa *finger* como parámetro. Si se proporciona el identificador de un usuario legítimo, por ejemplo, "lpb", la salida será la información sobre ese usuario, como se muestra a continuación:

Figura 60.

Respuestas CGI finger esperadas y subvertidas

Finger User

Login Name TTY Idle Login Time Where

lpb Lawrie Brown p0 Sat 15:24 ppp41.grapevine

Finger User

attack success

-rwxr-xr-x 1 lpb staff 537 Oct 21 16:19 finger.cgi

-rw-r--r-- 1 lpb staff 251 Oct 21 16:14 finger.html

Sin embargo, si un atacante proporciona un valor que incluye meta caracteres de shell, por ejemplo, "xxx; echo attack success; ls -l finger*", entonces el resultado se muestra en la figura 60. El atacante puede ejecutar cualquier programa en el sistema con los privilegios del servidor web. En este ejemplo, los comandos adicionales eran solo para mostrar un mensaje y enumerar algunos archivos en el directorio web. Pero se podría usar cualquier comando.

Como contraparte, la figura 61 muestra una extensión adecuada del script CGI de *finger* vulnerable. Esto agrega una prueba que garantiza que la entrada del usuario contiene solamente caracteres alfanuméricos. De lo contrario, la secuencia de comandos finaliza con un mensaje de error que especifica que la entrada suministrada contenía caracteres ilegales. Tenga en cuenta que, aunque este ejemplo usa Perl, el mismo tipo de error puede ocurrir en un programa CGI escrito en cualquier idioma. Si bien los detalles de la solución difieren, todos implican verificar que la entrada coincida con las suposiciones sobre su forma.

Figura 61.

Extensión segura para Perl finger CGI script

```
14 # get name of user and display their finger details
15 $user = $q->param("user");
16 die "The specified user contains illegal characters!"
17 unless ($user =~ /\w+/);
18 print `/usr/bin/finger -sh $user`;
```

Adicional al ejemplo expuesto, existen otros que lo invito a revisar en el capítulo 11 del texto base, seguro le servirán de insumo para comprender de mejor manera el tema revisado.

6.3. Escritura de código de programa seguro

Un segundo componente (aparte del manejo de la entrada del programa) es el procesamiento de los datos de entrada según el algoritmo. Es importante recordar que, los lenguajes de alto nivel generalmente se compilan y vinculan en código de máquina, que luego es ejecutado directamente en el procesador de destino; aunque, existen algunas variaciones de estos procesos, en todos los casos, la ejecución de un programa implica

la ejecución de instrucciones en lenguaje de máquina por parte de un procesador para implementar el algoritmo deseado; estas instrucciones manipularán los datos almacenados en varias regiones de la memoria y en los registros del procesador (Stallings, 2018).

Si tomamos en cuenta nuestro tema de estudio, la seguridad en el software, las cuestiones clave son:

- a. *Si el algoritmo implementado resuelve correctamente el problema especificado*, es decir, es posible que el algoritmo no implemente correctamente todos los casos o variantes del problema. Esto podría permitir que alguna entrada de programa aparentemente legítima desencadene un comportamiento del programa que no estaba previsto, proporcionando al atacante capacidades adicionales.
- b. *Si las instrucciones ejecutadas correctamente representan la especificación del algoritmo de alto nivel*, este segundo problema se refiere a la correspondencia entre el algoritmo especificado en algún lenguaje de programación y las instrucciones de la máquina que se ejecutan para implementarlo. La suposición es que el compilador o intérprete realmente genera o ejecuta código que implementa válidamente las declaraciones del lenguaje. Cuando se considera esto, el problema suele ser uno de eficiencia, generalmente abordado especificando el nivel requerido de indicadores de optimización para el compilador.
- c. Si la manipulación de valores de datos en variables, tal como se almacenan en registros de máquina o memoria, es válida y significativa.



Recuerde, el software escrito usando los procesos descritos en esta unidad puede detectar condiciones erróneas resultantes de algún ataque y, continuar ejecutándose de manera segura o fallar sin problemas mayores.



6.4. Interacción con el sistema operativo y otros programas

El sistema operativo genera un entorno de ejecución para un proceso cuando se ejecuta un programa, además del código y los datos para el programa, el proceso incluye información proporcionada por el sistema operativo. Estos incluyen variables de entorno, que se pueden usar para adaptar la operación del programa, y cualquier argumento de línea de comando especificado para el programa. Todos estos datos deben considerarse entradas externas al programa cuyos valores necesitan validación antes de su uso. Desde la perspectiva de la seguridad del software, los programas necesitan acceso a los diversos recursos, como archivos y dispositivos, que utilizan; a menos que se otorgue el acceso adecuado, es probable que estos programas fallen. Sin embargo, los niveles excesivos de acceso también son peligrosos porque cualquier error en el programa podría comprometer una mayor parte del sistema.

También existen preocupaciones cuando varios programas acceden a recursos compartidos, como un archivo común, entonces, se necesitan mecanismos de sincronización apropiados. Ahora discutimos cada uno de estos temas con más detalle.

Variables de entorno

Las variables de entorno son una colección de valores de cadena heredados por cada proceso de su padre que pueden afectar la forma en que se comporta un proceso en ejecución. El sistema operativo los incluye en la memoria del proceso cuando se construye. De manera predeterminada, son una copia de las variables de entorno de los padres. Sin embargo, la solicitud para ejecutar un nuevo programa puede especificar una nueva colección de valores para usar en su lugar. Un programa puede modificar las variables de entorno en su proceso en cualquier momento, y estas a su vez, se pasarán a sus hijos.

La preocupación de seguridad para un programa es que estos proporcionan otra ruta para que los datos no confiables ingresen a un programa y, por lo tanto, deben validarse. El uso más común de estas variables en un ataque es por parte de un usuario local que intenta obtener mayores privilegios

en el sistema. El objetivo es subvertir un programa que otorga privilegios de superusuario o administrador, obligándolo a ejecutar el código de la selección del atacante con estos privilegios superiores.

Lo invito a revisar algunos ejemplos de ataques usando las variables de entorno del sistema que están descritos en la sección 11.4 del texto base, serán de mucho apoyo para comprender desde una vista práctica la importancia del cuidado para identificar la forma en que un programa interactúa con el sistema en el que se ejecuta y para considerar cuidadosamente las implicaciones de seguridad de estas suposiciones.

Uso de privilegios mínimos apropiados

La consecuencia de muchas de las fallas del programa que analizamos en este capítulo es que el atacante puede ejecutar código con los privilegios y derechos de acceso del programa o servicio comprometido. Si estos privilegios son mayores que los que ya tiene el atacante, esto da como resultado una avalancha de privilegios, una etapa importante en el proceso general de ataque. El uso de los niveles más altos de privilegios puede permitir que el atacante realice cambios en el sistema, lo que garantiza el uso futuro de estas mayores capacidades. Esto sugiere fuertemente que los programas deben ejecutarse con la menor cantidad de privilegios necesarios para completar su función. Esto se conoce como el principio de privilegios mínimos y es ampliamente reconocido como una característica deseable en un programa seguro.

Otra preocupación es garantizar que cualquier programa privilegiado pueda modificar solo los archivos y directorios necesarios. Una deficiencia común que se encuentra con muchos programas privilegiados es que tienen la propiedad de todos los archivos y directorios asociados. Si el programa se ve comprometido, el atacante tiene más posibilidades de modificar y corromper el sistema. Esto viola el principio de privilegio mínimo.

La mayor preocupación con los programas privilegiados (o, de superusuario) ocurren cuando dichos programas se ejecutan con privilegios de raíz o administrador. Estos proporcionan niveles muy altos de acceso y control al sistema. La adquisición de tales privilegios suele ser el principal objetivo de un atacante en cualquier sistema. Por lo tanto, cualquier programa privilegiado es un objetivo clave. El principio de privilegio mínimo indica que dicho acceso debe otorgarse tan rara vez y tan brevemente como sea posible. Lamentablemente, debido al diseño de los sistemas operativos y

la necesidad de restringir el acceso a los recursos del sistema subyacente, existen circunstancias en las que se debe otorgar dicho acceso. Los ejemplos clásicos incluyen los programas utilizados para permitir que un usuario inicie sesión o cambie las contraseñas en un sistema; dichos programas solo son accesibles para el usuario root.

Técnicas de mitigación

La figura 62 representa algunas técnicas que permiten cumplir con el desarrollo de software seguro tomando en cuenta las vulnerabilidades descritas en esta unidad.

Figura 62.

Técnicas de diseño de programas defensivos – acceso a variables del sistema.



6.5. Manejo de la salida del programa

El componente final de nuestro modelo de programas informáticos es la generación de una salida, como resultado del procesamiento de la entrada y otras interacciones. Esta salida puede:

- Almacenarse para uso futuro (por ejemplo, en archivos o una base de datos)
- Transmitirse a través de una conexión de red
- Estar destinada a mostrarse a algún usuario

Al igual que con la entrada del programa, los datos de salida pueden clasificarse como binarios o textuales. En todos los casos, es importante desde la perspectiva de la seguridad del programa que la salida realmente se ajuste a la forma e interpretación esperada. Si se dirige a un usuario,

será interpretado y mostrado por algún programa o dispositivo apropiado. Si esta salida incluye contenido inesperado, entonces puede ocasionar un comportamiento anómalo, con efectos perjudiciales para el usuario. Una cuestión crítica aquí es la suposición de un origen común. Si un usuario está interactuando con un programa, se supone que todo el resultado visto fue creado por, o al menos validado por, ese programa. Sin embargo, un programa puede aceptar la entrada de un usuario, guardarla y luego mostrarla a otro usuario. Cualquier programa que recopile y confíe en datos de terceros debe ser responsable de garantizar que cualquier uso posterior de dichos datos sea seguro y no viole las suposiciones del usuario. Estos programas deben identificar qué contenido de salida está permitido y filtrar cualquier dato posiblemente no confiable para asegurarse de que solo se muestre la salida válida.

Con el objetivo de afianzar los conocimientos de este tema, le invito a revisar algunos ejemplos detallados en el apartado 11.5 del texto guía, estos ejemplos de fallas de seguridad que resultan de la salida del programa, no era el programa que generaba la salida, sino el programa o dispositivo utilizado para mostrar la salida. Se podría argumentar que esta no es la preocupación del programador, ya que su programa no está haciendo algo fuera de lo normal, sin embargo, si el programa actúa como conducto para un ataque, la reputación del programador se verá empañada y es posible que los usuarios estén menos dispuestos a utilizar el programa.

¡Felicitaciones! Hemos terminado el estudio de la penúltima unidad, avancemos a la unidad final, pero antes, lo invito a desarrollar la actividad propuesta.



Actividad de aprendizaje recomendada

Recuerde, en esta semana hemos revisado los temas relacionados con la seguridad del software, puntualmente, la importancia de conocer y controlar la interacción de nuestros sistemas, con el sistema operativo; además, revisamos la necesidad de gestionar la salida que nuestro código genera e incluso, si nuestro programa es utilizado para mostrar esos resultados. Con el objetivo de revisar la temática expuesta, lo invito a desarrollar el siguiente recurso interactivo denominado *“Seguridad en el software, identificar estrategias de mitigación”*

Seguridad en el software, identificar estrategias de mitigación

Estimado/a estudiante, le invito a que realice la autoevaluación para comprobar sus conocimientos.



Autoevaluación 7

Lea atentamente las preguntas propuestas en relación a los conceptos de seguridad informática y seleccione la opción de respuesta correcta.

1. _____ es el proceso de diseñar e implementar *software* para que continúe funcionando incluso cuando esté bajo ataque.
 - a. Programación asegurada.
 - b. Programación orientada a objetos.
 - c. Programación con respaldos.
 - d. Programación defensiva o segura.
2. Las vulnerabilidades de seguridad informática son el resultado de:
 - a. Malas prácticas de programación.
 - b. Infección de virus en los sistemas.
 - c. *Malware*.
 - d. Usuarios que no ejecutan bien los programas.
3. Del siguiente listado, ¿cuáles son consideradas como fallas críticas de seguridad de aplicaciones web relacionadas con código de *software* inseguro? (elija dos).
 - a. Entrada no válida.
 - b. Desbordamiento de búfer.
 - c. Asignación errónea de memoria.
 - d. Capacidad insuficiente de almacenamiento.
 - e. Errores de escritura.
4. Cuando analizamos la seguridad en el *software* esta se relaciona con:
 - a. La falla accidental de un programa.
 - b. La forma de diseño y prueba estructurada para identificar y eliminar errores en un programa.
 - c. El enfoque en errores específicos que resultan en una falla que puede ser aprovechada por un atacante.
 - d. La cantidad total de errores en un programa.

5. La entrada del programa hace referencia a _____.
_____.
- a. cualquier fuente de datos que se origina fuera del programa.
 - b. el código inicial del programa.
 - c. las librerías con la entrada de código externo.
 - d. el usuario y clave previos a la ejecución de un programa.
6. ¿Una de las alternativas para controlar el problema de desbordamiento de búfer es trabajar con un búfer de tamaño dinámico?
- a. Verdadero.
 - b. Falso.
7. Un ataque de inyección está relacionado con _____.
_____.
- a. fallas en el programa relacionadas con el manejo no válido de los datos de entrada.
 - b. el ingreso de código no validado previamente.
 - c. la relación que existe entre el código inicial y el desarrollo final.
 - d. fallas en el código relacionadas con la mala programación inicial.
8. ¿El uso de privilegios mínimos indica que los programas deben ejecutarse con la mayor cantidad de privilegios necesarios para completar su función?
- a. Verdadero.
 - b. Falso.
9. La modularización de programas tiene relación con:
- a. Limitar la vista de un programa al sistema de archivos.
 - b. Uso de contenedores.
 - c. Abstracción de recursos computacionales.
 - d. Mayor grado de aislamiento de componentes.

10. Cualquier programa que recopile y confíe en datos de terceros no puede ser responsable de garantizar el uso posterior de dichos datos y que no se violen las suposiciones del usuario.

- a. Verdadero.
- b. Falso.

[Ir al solucionario](#)



Si al contestar la autoevaluación su respuesta tuvo resultados positivos ¡FELICITACIONES, SIGA ADELANTE!, caso contrario revise nuevamente el contenido de los ítems errados, para reforzar su aprendizaje. Recuerde que en caso de tener alguna inquietud puede consultar al profesor tutor.

Resultado de aprendizaje 9 y 10

- Describe cómo la ciencia forense digital, encaja con las otras disciplinas forenses.
- Realiza análisis de medios informáticos utilizando herramientas forenses.

En la actualidad, el 85% de infracciones de seguridad cibernetica son causadas por errores humanos (Verizon, 2021), como consecuencia, se puede habilitar muchos delitos desde robos, estafas masivas utilizando equipo tecnológico. Esta unidad tiene como objetivo analizar las diversas técnicas y métodos empleados para detectar los diferentes delitos informáticos, enfocándonos en el conocimiento de los objetivos de la informática forense, las estrategias recomendadas a ejecutar (buenas prácticas) y el software usado para ejecutar los procesos de informática forense.

Le recomiendo poner mucha atención a los contenidos y desarrollar las actividades recomendadas que servirán de apoyo para el proceso de aprendizaje.

Contenidos, recursos y actividades de aprendizaje



Semana 15

Unidad 7. Análisis informático forense

7.1. Objetivos de la informática forense

La informática forense tiene como objetivo generar evidencias legales que sirvan de base a procedimientos judiciales. Por definición, la evidencia es el conjunto de recursos y datos a los que ha tenido acceso un perito para extraerlos, analizarlos, verificar su autenticidad y, poder así responder a las cuestiones técnicas planteadas por una contraparte o por un tribunal (Romero, 2020); bajo este contexto podemos concluir que la informática forense permite brindar solución a los problemas relacionados con la

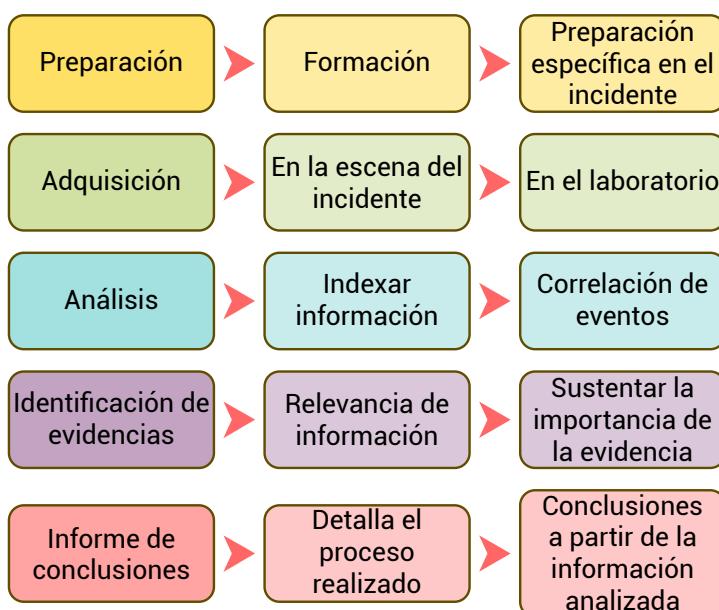
seguridad informática, con el objetivo de que, en caso de que exista un delito/afectación, la información quede a buen recaudo y permita ser base o apoye a un proceso de investigación.

Fases de un peritaje

Las fases pueden ser diversas y, según el autor o el enfoque, se puede definir varias etapas; en general, la figura 63 presenta un enfoque general a las fases de un peritaje.

Figura 63.

Fases generales de un peritaje informático



El análisis forense tiene que ser objetivo, científico y no subjetivo; se debe dar respuesta a las preguntas planteadas en el inicio del proceso de investigación; al final, la ciencia forense es la aplicación de métodos, conocimientos y técnicas de investigación que permitan validar la autenticidad de los hechos en cuestión. En informática forense, gran parte de los objetivos es verificar que la información proporcionada sea verídica, no alterada, aplica tanto a los datos, programas, como equipos informáticos.

7.2. Buenas prácticas en informática forense

Uno de los puntos clave en la ejecución práctica de informática forense, es establecer la cadena de custodia, que no es más que el mecanismo que va a permitir conservar las garantías de la veracidad de una evidencia, ya que, de lo contrario, dicha evidencia pondrá en duda las conclusiones que se generen a partir de esta. **La cadena de custodia** debe estar formada por algunas etapas, a continuación, el detalle:

1. La identificación, extracción y registro de la evidencia
2. La preservación y almacenamiento de la evidencia
3. Los traslados a los que va a ser sometida la evidencia

El proceso de cambio de responsables de la posesión de la evidencia, es lo que conocemos como la cadena custodia, por ejemplo, si se requiere pasar un disco duro para ser almacenado de un lugar X a Y, este proceso debe ser registrado cuándo y dónde tuvieron lugar, quien entrega y quien recibe la evidencia. Como artefacto de registro, se puede trabajar con formularios que pueden ser creados tomando como bases formularios de registro de evidencia de otros procesos judiciales.

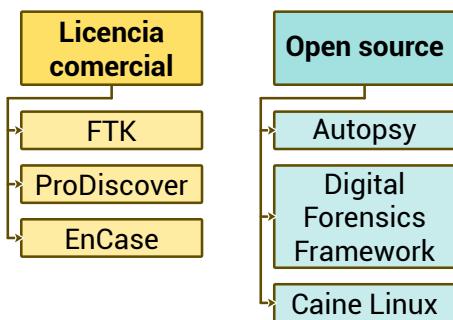
7.3. Software usado para informática forense

Dentro de los objetivos de nuestro estudio, está preparar al perito informático para realizar el análisis de investigación forense; bajo este contexto, el conocimiento de herramientas permitirá realizar un análisis eficiente.

El perito informático debe estar preparado para todo lo que se pueda encontrar dentro de su investigación; bajo este contexto, clasificaremos al software recomendado en dos categorías, la figura 64 detalla la clasificación propuesta de software informático forense.

Figura 64.

Clasificación de software informático forense



Seguramente a la fecha de su estudio, encontrará nuevas herramientas destinadas al propósito de informática forense, todas son válidas y, existen herramientas especializadas que se centran en una tarea específica de investigación, por ejemplo, FTK Imager o DiskExplorer.



En este punto de nuestro estudio, permítame extenderle mi más cordial felicitación, hemos finalizado el desarrollo de nuestra asignatura; ahora, solo nos queda hacer un repaso de los temas clave y reforzar los nuevos conocimientos adquiridos.



Actividad de aprendizaje recomendada

Desarrolle la autoevaluación de la unidad 8 "Análisis informático forense", considere que las preguntas planteadas constituyen una estrategia de aprendizaje y tienen como finalidad conocer el grado de asimilación de los contenidos estudiados. En caso de que tenga dificultad para responder alguna pregunta, le recomiendo volver a revisar los contenidos en el texto básico y la guía didáctica virtualizada.



Autoevaluación 8

Lea atentamente las preguntas propuestas en relación con los conceptos de control de acceso y seleccione la opción de respuesta correcta.

1. ¿Cuál es el objetivo de la informática forense?
 - a. Generar evidencias legales que sirvan de base a procedimientos judiciales.
 - b. Generar datos validados por otros usuarios.
 - c. Generar evidencias que creen dudas en los procesos judiciales
 - d. Generar datos que sean validados por otros expertos en un juicio.
2. Dentro de las fases del peritaje tenemos la fase de análisis esta se relaciona con _____.
 - a. tener preparación específica en el incidente.
 - b. encontrar información relevante.
 - c. brindar una conclusión de la investigación realizada.
 - d. correlación e indexación de información.
3. ¿El análisis forense, dependiendo de su complejidad, puede ser subjetivo?
 - a. Verdadero.
 - b. Falso.
4. La cadena de custodia es _____.
 - a. mecanismo que permitirá conservar las garantías de la veracidad de una evidencia.
 - b. proceso mediante el cual se lleva la información de un lugar a otro.
 - c. fase del peritaje relacionada con la movilidad de la evidencia.
 - d. el sitio de almacenamiento de la evidencia.

5. ¿Un artefacto de registro de la cadena de custodia son los formularios?
 - a. Verdadero.
 - b. Falso.
6. ¿Cuál de los siguientes softwares informáticos forenses es Open Source?
 - a. KaliLinux.
 - b. ProDiscover.
 - c. CaineLinux.
 - d. EnCase.
7. Dentro de las fases de un peritaje informático, la etapa de preparación tiene relación con _____.
 - a. formación en el área a investigar.
 - b. detallar el proceso realizado en la investigación.
 - c. indexar información.
 - d. recolectar información inicial.
8. ¿Cuál de las siguientes NO es una etapa de la cadena de custodia?
 - a. Identificación de la evidencia.
 - b. Preservación de la evidencia.
 - c. Duplicar la evidencia como *backup*.
 - d. Traslados a los que se someterá la evidencia.
9. ¿Cuál es el problema si una evidencia es alterada?
 - a. La conclusión que tomó como base esa evidencia es dudosa.
 - b. La informática forense debe indicar el momento en que se hizo la alteración.
 - c. Significa que el proceso de identificación de evidencia ha sido alterado.
 - d. El perito ha modificado la información y toda la evidencia no es válida.

10. La cadena de custodia es un proceso que permite registrar todos los cambios que afectaron a una evidencia y conservar las garantías de veracidad de dicha evidencia.
- a. Verdadero.
 - b. Falso.

[Ir al solucionario](#)



Si al contestar la autoevaluación su respuesta tuvo resultados positivos ¡FELICITACIONES, SIGA ADELANTE!, caso contrario revise nuevamente el contenido de los ítems errados, para reforzar su aprendizaje. Recuerde que en caso de tener alguna inquietud puede consultar al profesor tutor.



Actividades finales del bimestre



Semana 16

1. Se recomienda revisar nuevamente en el texto básico y la guía didáctica virtualizada, los temas relacionados con la unidad 5, 6, 7 y 8, para ello se recomienda retomar sus apuntes del segundo bimestre y prepararse para la evaluación correspondiente.
2. Revise los ejemplos de información sensible en un único sistema lógico.
3. Visualice los REA expuestos en el plan docente de los temas abordados en el segundo bimestre.
4. Recuerde que, si no alcanzó a participar de la actividad síncrona, está a tiempo de recuperar la misma desarrollando la actividad complementaria.
5. Espero que haya desarrollado las actividades que se han planificado durante este Segundo Bimestre, las cuales son formativas, sumativas (calificadas) y aportan a lo largo de su preparación y autoaprendizaje. Dichas actividades planificadas son Foro, Chat y Cuestionarios que se han configurado en la plataforma académica virtual. Así mismo, sugiero que en la medida de lo posible desarrolle los ejercicios y autoevaluaciones propuestos en el presente curso, que son formativos y sirven como preparación para la evaluación final de bimestre.



¡Felicitaciones! Ha conseguido grandes aprendizajes en el segundo bimestre en la asignatura de evaluación de seguridad en sistemas y tecnologías de información, le invito a continuar con su preparación, es importante mantener un estado de autoeducación continua.



4. Solucionario

Autoevaluación 1		
Pregunta	Respuesta	Retroalimentación
1	a	"La tríada CIA" – Confidencialidad, Integridad y Disponibilidad (por sus siglas en inglés) son los conceptos definidos como los objetivos fundamentales de la seguridad, tanto para los datos como para la información.
2	a	La seguridad informática está relacionada a las medidas y controles que aseguran la confidencialidad, integridad y la disponibilidad de los activos del sistema de información, esto incluye <i>hardware</i> , <i>software</i> , <i>firmware</i> e información que se procesa, almacena y comunica.
3	a, b, c, d	Con base en el RFC 4949, se describen cuatro tipos de amenaza como son la divulgación no autorizada, el fraude, interrupción y la usurpación.
4	d	Un ataque pasivo intenta aprender o hacer uso de la información del sistema, pero no afecta los recursos del sistema.
5	a, b, c	Los requisitos funcionales de la seguridad de la información son: control de accesos, concienciación y formación, auditoría y rendición de cuentas, gestión de la configuración, identificación y autenticación, respuesta al incidente, mantenimiento, evaluación de riesgos y protección del sistema y de las comunicaciones.
6	d	Al abarcar tanto el diseño como la implementación del sistema, el aseguramiento es un atributo de un sistema de información que proporciona motivos para confiar en que el sistema funciona, de manera que la política de seguridad del sistema se cumpla.
7	b	Una política de seguridad es como una declaración formal de reglas y prácticas que especifican o regulan cómo un sistema u organización proporciona servicios de seguridad para proteger los recursos sensibles y críticos del sistema. Esta política de seguridad formal se presta a ser aplicada por los controles técnicos del sistema, así como sus controles operativos y de gestión.
8	d	El principio del mínimo privilegio hace referencia a un concepto de seguridad de la información en que se da a un usuario los niveles (o permisos) de acceso mínimos necesarios para desempeñar sus funciones laborales.
9	a	La integridad de datos es un término usado para referirse a la exactitud y fiabilidad de los datos; cualquier cambio sobre estos debe ser un cambio autorizado y registrado.

Autoevaluación 1

Pregunta Respuesta Retroalimentación

10 c La privacidad digital es el derecho de los usuarios a proteger sus datos y a decidir qué información puede ser visible o compartida.

Ir a la
autoevaluación

Autoevaluación 2		
Pregunta	Respuesta	Retroalimentación
1	a	El cifrado simétrico, también conocido como cifrado convencional o de clave única, era el único tipo de cifrado utilizado antes del advenimiento del cifrado de clave pública, a finales de la década de 1970.
2	c	El algoritmo de cifrado simétrico más utilizado es el cifrado de bloques, su funcionamiento se basa en que procesa la entrada de texto plano en bloques de tamaño fijo, y genera un bloque de texto cifrado del mismo tamaño para cada bloque de texto plano ingresado.
3	d	El tamaño máximo usado por el algoritmo DES es de 56 bits.
4	a	La vida de DES se amplió con el uso de triple DES (3DES), que consiste en repetir el algoritmo DES básico tres veces, utilizando dos o tres claves únicas, con un tamaño de clave de 112 o 168 bits.
5	a	MAC es una técnica de autenticación de mensajes que implica el uso de una clave secreta para generar un pequeño bloque de datos, conocido como código de autenticación de mensajes (MAC), este se adjunta al mensaje y se realiza el proceso de envío.
6	a, b	En la actualidad, la función hash más utilizada ha sido el algoritmo hash seguro (SHA), en sus diferentes versiones.
7	a	La criptografía de clave pública es asimétrica, lo que implica el uso de dos claves separadas, en contraste con el cifrado simétrico que usa solo una clave.
8	a, d	Algunos algoritmos son adecuados para el uso en aplicaciones, mientras que otros pueden usarse solo en una o dos; en el caso de las firmas digitales, se trabaja con RSA y DSS.
9	a	El cifrado simétrico es utilizado con el objetivo de que las comunicaciones sean secretas, es decir, brindar confidencialidad a la información enviada.
10	d	El mensaje original, en cualquier algoritmo de cifrado, es introducido en forma legible natural, es decir, en texto plano.

**Ir a la
autoevaluación**

Autoevaluación 3		
Pregunta	Respuesta	Retroalimentación
1	b	Uno de los medios generales para autenticar la identidad de un usuario es “algo que el individuo conoce”, esto puede ser: contraseña, PIN, respuesta a un conjunto preestablecido de preguntas.
2	c	Hash es un método por el cual se agrega una secuencia aleatoria de <i>bits</i> (denominada valor de sal) a la contraseña, el resultado es un código hash.
3	d	La evaluación de riesgos de seguridad involucra tres conceptos distintos que se relacionan entre sí: nivel de seguridad, impacto potencial y áreas de riesgo.
4	a	La biometría se basa en el reconocimiento de patrones. En comparación con las contraseñas y los <i>tokens</i> , la autenticación biométrica es técnicamente más compleja y costosa.
5	b	La etapa de verificación consiste en la presentación o la generación de información de autenticación que corrobora la vinculación entre la entidad y el identificador.
6	c	El reconocimiento por huella dactilar, retina y rostro son ejemplos de biometría estática, la biometría dinámica utiliza entre otros el patrón de voz, características de escritura a mano, ritmo de tecleo.
7	b	Un programa para adivinar contraseñas se denomina <i>cracker de contraseñas</i> .
8	d	La estrategia de educación del usuario es cuando se explica a los usuarios la importancia de utilizar contraseñas difíciles de adivinar y se les proporcionan directrices para seleccionar contraseñas seguras.
9	d	Cada persona que vaya a ser incluida en la base de datos de usuarios autorizados debe ser primero registrada en el sistema.
10	b	La geometría de la mano es utilizada en aplicaciones biométricas para identificar las características de la mano, incluyendo la forma, y la longitud y anchura de los dedos.

[Ir a la autoevaluación](#)

Autoevaluación 4		
Pregunta	Respuesta	Retroalimentación
1	b	Sujetos, objetos y derechos de acceso constituyen los tres elementos básicos del control de acceso.
2	a	Generalmente, los sujetos son de tres clases: propietario, grupo, todos.
3	d	El control de acceso especifica quién o qué puede tener acceso a cada recurso específico del sistema y el tipo de acceso que se permite en cada instancia.
4	b	La autenticación es el proceso que verifica que las credenciales de un usuario u otra entidad del sistema son válidas.
5	a	La autorización es la concesión de un derecho o permiso a una entidad del sistema, para acceder a un recurso del sistema.
6	c	El método tradicional para implementar el control de acceso se denomina DAC o control de acceso discrecional.
7	c	El sujeto es una entidad capaz de acceder a objetos.
8	a	Un objeto es un recurso cuyo acceso está controlado.
9	b	RBAC o control de acceso basado en roles, se basa en los roles que los usuarios asumen en un sistema y no en la identidad del usuario.
10	a	Las restricciones proporcionan un medio para adaptar el RBAC a las especificidades de las políticas administrativas y de seguridad de una organización.

[Ir a la
autoevaluación](#)

Autoevaluación 5		
Pregunta	Respuesta	Retroalimentación
1	b	Una colección estructurada de datos almacenados es lo que se conoce como base de datos. Si bien las demás respuestas están relacionadas con la base de datos, no son correctas para esta pregunta.
2	a	Una base de datos relacional es una tabla de datos formada por filas y columnas, similar a una hoja de cálculo, cada columna contiene un tipo concreto de datos, mientras que cada fila contiene un valor específico para cada columna.
3	d	En el lenguaje de las bases de datos relacionales, el bloque básico de construcción es una relación, que es una tabla plana, las filas son referenciadas como tuplas y las columnas como atributos.
4	d	En el lenguaje de las bases de datos relacionales, el bloque básico de construcción es una relación, que es una tabla plana, las filas son referenciadas como tuplas y las columnas como atributos.
5	c	Una clave primaria es la que identifica de forma exclusiva una fila, consta de uno o varios nombres de columna.
6	c	Una vista es el resultado de una consulta que devuelve filas y columnas seleccionadas de una o varias tablas, también es conocida como tabla virtual.
7	b	En el lenguaje de las bases de datos relacionales, el bloque básico de construcción es una relación, que es una tabla plana, las filas son referenciadas como tuplas y las columnas como atributos.
8	d	El resultado de una consulta, que devuelve filas y columnas seleccionadas de una o varias tablas, se conoce como vista.
9	a	El lenguaje de consulta estructurado (SQL) es un lenguaje estandarizado que puede utilizarse para definir el esquema, manipular y consultar datos en una base de datos relacional.
10	a	Los ataques SQL <i>injection</i> se pueden clasificar por tipo, en: ataque dentro de banda (<i>inband attack</i>), ataque inferencial y ataque fuera de banda (<i>out-of-band attack</i>).

Ir a la
autoevaluación

Autoevaluación 6		
Pregunta	Respuesta	Retroalimentación
1	c	El <i>malware</i> o código malicioso tiene como estrategia el encubrimiento u ocultación para obtener información de la víctima.
2	a	Un <i>rootkit</i> es un tipo de <i>malware</i> , que, a más de permanecer encubierto, contiene un conjunto de programas que se instalan con privilegios de administrador (<i>root</i>), cualquier otra respuesta es incorrecta.
3	a	El <i>malware</i> que espera el cumplimiento de una condición para su ejecución se denomina bomba lógica, su comportamiento es diferente al de un gusano o un troyano. La respuesta correcta es bomba lógica.
4	d	En temas de virus informáticos la carga útil es lo que hace el virus, es decir, el contenido del virus que ejecuta la actividad maliciosa.
5	d	Un virus pasa por cuatro fases: fase de reposo, fase activación, fase de propagación y fase de ejecución. En la fase de ejecución es cuando el virus realiza su función.
6	a	Un virus pasa por cuatro fases: fase de reposo, fase activación, fase de propagación y fase de ejecución. Solamente en la fase de reposo el virus permanece inactivo.
7	a	El <i>spam</i> se refiere a todo el correo electrónico masivo no solicitado, por lo general correos de publicidad, promociones, posibles estafas y ataques, entre otros están categorizados como <i>spam</i> .
8	b	El <i>ransomware</i> es un tipo de <i>malware</i> que se caracteriza por pedir un pago o rescate a cambio de la clave, necesaria para descriptar la información que previamente fue encriptada (secuestrada).
9	c	DDoS es la respuesta adecuada para un ataque distribuido de denegación de servicio, cuya estrategia se basa en <i>bots</i> que comprometen el servicio en la red, dejándolos inaccesibles para los usuarios.
10	d	Contar con un esquema de seguridad apropiado que cuente con políticas, con un plan de concientización, mitigación de vulnerabilidades y de amenazas es vital para una organización, esto se denomina prevención, cuyo objetivo es que ningún ataque tenga éxito.

**Ir a la
autoevaluación**

Autoevaluación 7		
Pregunta	Respuesta	Retroalimentación
1	d	La programación defensiva o segura, es el proceso de diseñar e implementar <i>software</i> para que continúe funcionando, incluso cuando esté bajo ataque.
2	a	Las malas prácticas de programación son un punto clave que puede determinar vulnerabilidades de seguridad informática.
3	a, b	Fallas críticas de seguridad de aplicaciones web son entrada no válida, desbordamiento de búfer, secuencia de comandos entre sitios, fallas de inyección y manejo inadecuado de errores.
4	c	La seguridad en el <i>software</i> está relacionada a los errores específicos que resultan en una falla que puede ser aprovechada por un atacante.
5	a	La entrada del programa hace referencia a cualquier fuente de datos que se origina fuera del programa, y cuyo valor no es conocido explícitamente por el programador cuando escribió el código.
6	a	La alternativa al tamaño de la entrada sería usar un búfer de tamaño dinámico, para garantizar que haya suficiente espacio disponible o procesar la entrada en bloques del tamaño de un búfer.
7	a	Se refiere a una amplia variedad de fallas en el programa, relacionadas con el manejo no válido de los datos de entrada.
8	b.	El uso de privilegios mínimos indica que los programas deben ejecutarse con la menor cantidad de privilegios necesarios para completar su función.
9	d	La modularización de programas tiene relación con la división de programas en módulos pequeños, generando un mayor grado de aislamiento de componentes.
10	b	Un programa que recopile y confíe en datos de terceros, debe ser responsable de garantizar el uso posterior de dichos datos y no viole las suposiciones del usuario.

[Ir a la autoevaluación](#)

Autoevaluación 8		
Pregunta	Respuesta	Retroalimentación
1	a	La informática forense busca generar evidencias legales que sirvan de base a procedimientos judiciales.
2	d	La correlación e indexación de información es parte de los objetivos de la fase de análisis en el peritaje informático.
3	b	El análisis forense debe ser objetivo y basado en evidencias.
4	a	La cadena de custodia es el mecanismo que permitirá conservar las garantías de la veracidad de una evidencia.
5	a	Como artefacto de registro se puede trabajar con formularios, que pueden ser creados tomando como bases formularios de registro de evidencia de otros procesos judiciales.
6	c	Software informático Open Source citamos a Autopsy, Digital Forensics Framework y CaineLinux.
7	a	La fase de preparación en el peritaje informático tiene que ver con la preparación del perito en el área que va a investigar.
8	c.	Las etapas de una cadena de custodia incluyen la identificación de la evidencia, la preservación y los traslados a los que será sometida la evidencia.
9	a	Si una evidencia es alterada o no es confiable, toda conclusión relacionada a esa evidencia también será dudosa.
10	a	La cadena de custodia es un proceso que permite registrar todos los cambios que afectaron a una evidencia y conservar las garantías de veracidad de dicha evidencia.

[Ir a la autoevaluación](#)



5. Referencias bibliográficas

Bibliografía Básica

Jaramillo, B & Aguilar, C (2022); Evaluación de la Seguridad en Sistemas y Tecnologías de la Información. Guía Didáctica: EdiLoja.

El propósito de este texto – guía es presentar una recopilación de temas que involucran conceptos de Seguridad en el Software. La información presentada guiará a los estudiantes en la revisión esquematizada de los contenidos y complementará la bibliografía proporcionada.

Bibliografía Complementaria

State of Software Security Volume 12. (2016). Veracode. Recuperado 3 de marzo de 2022, de [enlace web](#)

Simpson S. (2014) SAFECode Whitepaper: Fundamental Practices for Secure Software Development 2nd Edition. In: Reimer H., Pohlmann N., Schneider W. (eds) ISSE 2014 Securing Electronic Business Processes. Springer Vieweg, Wiesbaden. [enlace web](#)

OWASP Foundation / Open Source Foundation for Application Security. (2013) OWASP. Recuperado 3 de marzo de 2022, de [enlace web](#)

2021 DBIR Master's Guide. (2021, 1 enero). Verizon Business. Recuperado 3 de marzo de 2022, de [enlace web](#)

Navarro, F. A. C., Calvo, C., & Matarrita, J. C. (2020). Cumplimiento en Regulación de Seguridad NIST 800-171. *Tecnología Vital*, 4(7).

National Institute of Standards and Technology. (2000, 3 marzo). NIST. Recuperado 3 de marzo de 2022, de [enlace web](#)

Lampson, B. W. (2004). Computer security in the real world. *Computer*, 37(6), 37-46.

Shirey, R. (2007, agosto). RFC 4949 - Internet Security Glossary, Version 2. Internet Security Glossary. Recuperado 3 de marzo de 2020, de [enlace web](#)

NIST Laboratory Information Systems Team - Problem processing request. (2016). NIST Laboratory Information Systems Team. Recuperado 3 de marzo de 2022, de [enlace web](#)

NIST Laboratory Information Systems Team - Problem processing request. (s.f.). NIST. Recuperado 3 de marzo de 2022, de [enlace web](#)

Souppaya, M. (2013, 22 julio). SP 800-83 Rev. 1, Malware Incident Prevention and Handling: Desktops and Laptops | CSRC. NIST. Recuperado 3 de marzo de 2022, de [enlace web](#)

A. (2015, 29 enero). ISO 12207. Normas y Estándares en Proyectos de T.I. Recuperado 2022, de [enlace web](#).

Santander Universidades. (2022, 14 enero). Seguridad activa y pasiva informática | Blog. Becas Santander. Recuperado 2022, de [enlace web](#)

La informática forense desde un enfoque práctico. (2020). 3Ciencias. Recuperado 2022, de [enlace web](#)

Recursos Educativos Abiertos (REA's)

Título del REA	Enlace
Bimestre 1: Criptografía y seguridad informática	enlace web
Bimestre 1: Top 10 Web Application Security Risks	enlace web
Bimestre 1 y/o 2: Seguridad Informática y Competencias Profesionales	enlace web
Bimestre 2: OWASP Security Knowledge Framework	enlace web