DGTIC UNAM
DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

Fallas de seguridad informática (vulnerabilidades)

CURSOS EN LÍNEA

Fallas de seguridad informática (vulnerabilidades)

Una vulnerabilidad es cualquier debilidad en los controles de seguridad de un sistema de información que pudiera ser aprovechada por una amenaza. Por lo tanto, las vulnerabilidades representan todas aquellas fallas, omisiones, o errores en la protección que permiten o facilitan la violación a la confidencialidad, integridad y disponibilidad de la información. A continuación, revisaremos los aspectos generales de las vulnerabilidades en los sistemas de información y algunas herramientas que permiten su identificación.

Vulnerabilidades en aplicaciones de software

Ya sea desde una computadora, dispositivo móvil, o bien, desde una plataforma web, existe una amplia gama de aplicaciones con las que los usuarios interactuamos día a día y por las que fluye una inmensa cantidad de información.

Sin embargo, las vulnerabilidades en aplicaciones de software son muy comunes. Se trata de defectos o errores en el código que pueden llegar a permitir a un atacante robar y manipular información de un sistema, o incluso, tomar control sobre el sistema en su totalidad. Es por esto que los desarrolladores trabajan de manera continua en la búsqueda de mejoras para crear actualizaciones y parches de seguridad para poder mitigarlas.

> Siempre mantenga sus aplicaciones y sistema operativo con las últimas actualizaciones y parches de seguridad.

OWASP TOP TEN

OWASP (*Open Worldwide Application Security Project*) es una fundación sin ánimo de lucro que trabaja en la mejora de la seguridad del software. Se trata de una comunidad dedicada a proporcionar orientación a las organizaciones para la concepción, desarrollo, adquisición, operación y mantenimiento de aplicaciones para que estas puedan ser confiables.

Entre los proyectos de OWASP se encuentra el OWASP TOP TEN que se encarga de señalar las principales vulnerabilidades en una amplia diversidad de software que puede abarcar desde aquel destinado para dispositivos móviles, hasta el software utilizado en la nube.

> OWASP es a su vez un estándar informativo para desarrolladores que señala los riesgos principales de seguridad que se presentan de manera común en las aplicaciones. Visite su página oficial para conocer más información: https://owasp.org/

Figura 1. Logotipo del proyecto OWASP.[1]

Abarcar las vulnerabilidades de aplicaciones para todo tipo de sistema informático representa una tarea que sobrepasa los objetivos del curso. Sin embargo, en este apartado nos concentraremos en el proyecto OWASP más renombrado, el OWASP TOP TEN de Aplicaciones Web y observaremos sus semejanzas y diferencias con el OWASP TOP TEN para Aplicaciones de Escritorio, dado que son dos tipos de aplicaciones que utilizamos de manera generalizada. A continuación, se muestra la lista de vulnerabilidades que esta fundación ha identificado como las más críticas para cada plataforma.

Tabla 1. **COMPARATIVA OWASP TOP TEN PARA APLICACIONES WEB Y PARA APLICACIONES DE ESCRITORIO**

OWASP TOP TEN Web[2]	OWASP TOP TEN Escritorio[3]
1. Pérdida de Control de acceso	1. Inyección
2. Fallos en la criptografía	2. Pérdida de control de acceso
3. Inyección	3. Exposición de datos sensibles
4. Diseño inseguro	4. Fallos en la criptografía
5. Configuración de seguridad incorrecta	5. Fallas en la identificación y autenticación
6. Componentes vulnerables y desactualizados	6. Configuración de seguridad incorrecta
7. Fallas en la identificación y autenticación	7. Comunicaciones no seguras
8. Fallas en el software y la integridad de los datos	8. Pobre calidad del código
9. Fallos en el registro y monitoreo	9. Componentes vulnerables y desactualizados
10. Fallos del lado del servidor	10. Fallos en el registro y monitoreo

[1] [Logotipo OWASP]. Recuperado de: https://owasp.org/
[2] OWASP. Top Ten. https://owasp.org/Top10/es/
[3] OWASP. Desktop app security Top 10. https://owasp.org/www-project-desktop-app-security-top-10/

Como se puede observar en la tabla, existen 7 vulnerabilidades principales que se consideran tanto para aplicaciones web como para aplicaciones de escritorio (destacadas en color gris). A continuación, revisaremos cada una de las vulnerabilidades de aplicaciones web para posteriormente, examinar aquellas restantes que solo se observen en aplicaciones de escritorio.

OWASP Top Ten Aplicaciones Web

Para aplicaciones web se consideran las siguientes vulnerabilidades:

Pérdida de Control de acceso

El control de acceso son todas aquellas medidas de protección establecidas para asegurar que solo los usuarios autorizados puedan ingresar a la aplicación. Por ejemplo, si un usuario puede ingresar a una sección que se supone protegida entonces se puede decir que se ha perdido el control de acceso. Esta vulnerabilidad puede llevar a que se exponga información sensible o que los usuarios no autorizados realicen acciones que se supone no deberían hacer.

Fallos en la criptografía

Los fallos en criptografía se refieren a todas las vulnerabilidades derivadas de la falta o de la configuración inadecuada de los algoritmos de cifrado que protegen la información de la aplicación. Por ejemplo, una aplicación podría estar utilizando algún estándar de cifrado con vulnerabilidades conocidas para la transmisión de datos, de tal manera que un atacante podría capturar esa información y mediante algún proceso de conversión obtener la información tal y como un usuario autorizado la recibiría.

Figura 2. Los fallos en la criptografía exponen la información a ataques que permitan descifrarla.

La criptografía puede entenderse como el arte de crear e implementar procesos de encriptación utilizando códigos secretos y cifrado. Es, a su vez, un método de almacenamiento y transmisión de datos, de tal manera, que solo los destinatarios deseados pueden leer y procesar. Se considera como una ciencia de protección de información por medio de codificación para evitar que individuos no autorizados puedan acceder a ella.

Inyección

Las vulnerabilidades de inyección provocan que los atacantes puedan ingresar datos o sentencias especializadas para que los programas las interpreten como comandos o parámetros y realicen acciones que no deberían. Esto sucede, principalmente, porque no hay una verificación ni un filtro previo sobre el tipo de datos que puede ingresar un usuario (por ejemplo, en un formulario o campo de entrada). Así, es posible que estas vulnerabilidades puedan llegar a aprovecharse para provocar que la aplicación brinde acceso no autorizado a información o a sectores protegidos.

Diseño inseguro

Esta categoría abarca todas las vulnerabilidades inherentes a la arquitectura de la aplicación. No son errores de configuración o implementación sino en la misma concepción en la que no se consideran las amenazas de su contexto. Es decir, la falta de controles de seguridad que pueden exponer a la aplicación a una serie de ataques. Por ejemplo, si las contraseñas se almacenan en una base de datos de una aplicación sin ningún tipo de implementación de cifrado para ocultar sus valores.

Figura 3. Las fallas en el diseño pueden provocar graves consecuencias en la seguridad.

Configuración de seguridad incorrecta

Este tipo de vulnerabilidades, como su nombre señala, implica la configuración inadecuada de los controles de seguridad. Esto puede ser provocado de muchas maneras, por ejemplo, en los servicios que se presentan activos cuando debieran estar inhabilitados, o en casos cuando el manejo de errores de aplicación revela información sobre su configuración. Imagine que intenta poner una cerradura a una puerta, pero la cerradura puede abrirse con un clip, esto es un ejemplo de configuración de seguridad incorrecta.

Componentes vulnerables y desactualizados

En esta categoría se abarca el uso de programas externos (porciones de código, scripts, librerías, etc.) como complemento de una aplicación, pero en los que no se implementan los controles de seguridad correspondientes, o bien, ya no cuentan con el soporte de sus desarrolladores para la creación de actualizaciones ni cuenta con parches de seguridad adecuados.

Fallas en la identificación y autenticación

La identificación y autenticación se utilizan para verificar a un usuario autorizado que desea acceder a los recursos de una aplicación. La identificación implica declarar quiénes somos,

mientras que la autenticación se refiere a la verificación de esa identidad. Un ejemplo claro de este proceso es cuando utilizamos un nombre de usuario (identificación) y una contraseña (autenticación) para iniciar sesión en una aplicación. Las vulnerabilidades asociadas con errores en la identificación y autenticación surgen de implementaciones deficientes de este proceso, lo que podría permitir que un atacante se aproveche de ellos. Por ejemplo, si un programa no cuenta con un mecanismo para protegerse contra ataques de fuerza bruta, donde un atacante intenta adivinar la contraseña probando todas las combinaciones posibles de valores.

Figura 4. Las fallas en la identificación y autenticación pueden provocar la concesión de privilegios a usuarios no autorizados.

Fallas en el software y la integridad de los datos

Las vulnerabilidades relacionadas al software e integridad de los datos se enfocan en errores de código de la aplicación y en la forma en la que ésta manipula los datos. Esto incluye situaciones donde el software no funciona según lo esperado y provoca errores, pérdida, o incluso, la corrupción misma de los datos. Por ejemplo, cuando un programa se cierra sin razón aparente o responde con datos incorrectos para el cómputo de operaciones.

Fallos en el registro y monitoreo

Las vulnerabilidades relacionadas con la falla en el registro y monitoreo abarcan, justamente, las debilidades o falta de mecanismos de control que permitan la detección y atribución de acciones realizadas en una sesión por un usuario. En este sentido, estas vulnerabilidades permitirían a un atacante realizar diferentes operaciones de manera desapercibida, sin que se tengan datos sobre lo que sucede o ha sucedido, o bien, sin poder adjudicar responsabilidades.

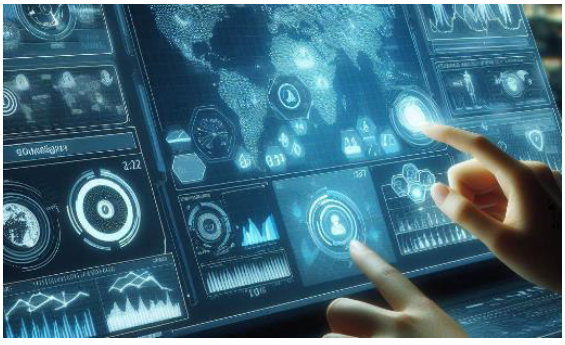

Figura 5. El monitoreo y registro de actividades es un elemento esencial para garantizar la seguridad de los sistemas.

Falsificación de solicitudes del lado del servidor

La mayoría de aplicaciones web residen en un servidor web donde utilizan y son alimentadas con la información de una base de datos que se almacena, a su vez, en otro, o incluso en el mismo servidor web.

En este tipo de vulnerabilidades se presentan errores en el servidor porque no se realiza de manera adecuada la comprobación sobre los recursos o información que recibe de una fuente externa para el procesamiento de datos de la aplicación.

OWASP Top Ten Aplicaciones de Escritorio

En ambos tipos de aplicaciones, ya sea web o de escritorio, se pueden observar numerosas similitudes. Como se ha enumerado, hay siete vulnerabilidades que aparecen tanto en una lista como en la otra. Por lo tanto, en esta sección revisaremos las vulnerabilidades señaladas por OWASP específicamente para las aplicaciones de escritorio, sin tener en cuenta las ya mencionadas en el OWASP Top Ten para aplicaciones web.

Exposición de datos sensibles

En muchas aplicaciones de escritorio se procesa y almacena información sensible. La exposición de estos datos es una vulnerabilidad importante que puede ocurrir por diferentes factores. Por ejemplo, la falta de cifrado en el código o la forma en la que se almacenan los datos, o incluso la falta de controles de acceso adecuados.

Comunicaciones no seguras

Las comunicaciones no seguras son un tipo de vulnerabilidad que aparece cuando se utilizan protocolos de transferencia que no cuentan con cifrado, autenticación o cualquier otra medida de protección. De esta manera, se expone el flujo de información que la aplicación utiliza, lo que puede llevar a un atacante a capturar, leer y modificar los mensajes entre sistemas.

Figura 6. Diferentes tipos de ataques utilizan los fallos en los protocolos de comunicación.

Pobre calidad del código

La pobre calidad del código es una vulnerabilidad que ocurre cuando el código de una aplicación no sigue las mejores prácticas de desarrollo. Esto puede manifestarse a través de omisiones, errores o inconsistencias en el código que afectan negativamente su funcionamiento y, lo que es

más preocupante, su seguridad. La mala calidad del código puede originarse por la falta de revisión y pruebas exhaustivas, la falta de experiencia técnica del equipo de desarrollo, limitaciones de presupuesto o la ausencia de estándares de desarrollo sólidos. Es importante entender que un código de baja calidad no solo puede causar problemas de funcionamiento, sino que también puede introducir vulnerabilidades de seguridad en la aplicación, lo que la hace más susceptible a ataques por parte de personas malintencionadas. Por lo tanto, es crucial para los desarrolladores dedicar tiempo y recursos a garantizar que el código de sus aplicaciones sea robusto, seguro y confiable.

Las vulnerabilidades en software pueden desencadenar problemas severos en la seguridad de la información. Por lo tanto, es preciso que se tomen medidas de mitigación. Es recomendable mantener las aplicaciones actualizadas y que cuenten con los últimos parches de seguridad. Además, se recomienda descargar e instalar software autorizado de compañías que lo respalden.

Detección de vulnerabilidades

El análisis de vulnerabilidades puede entenderse como un conjunto de técnicas que se aplican para identificar y valorar las posibles vulnerabilidades que se presentan en un sistema. Es decir, este análisis permitirá conocer las deficiencias y las áreas de oportunidad donde se pueden fortalecer los controles de seguridad.

Para realizar un análisis de vulnerabilidades de los sistemas de información será necesario:

- Identificar los activos con los que se cuenta.
- Priorizar los activos con base en su valor.
- Identificar las vulnerabilidades a nivel de aplicaciones, red y sistema operativo.

Actualmente, existen diferentes herramientas que permiten la detección y análisis automatizado de vulnerabilidades en los sistemas de información y aplicaciones. Estas herramientas son conocidas como escaners de vulnerabilidades y facilitan enormemente la identificación de debilidades en la configuración de los sistemas. Sin embargo, es recomendable que estos análisis los realice personal experto que cuente con las capacidades técnicas para su ejecución y, sobre todo, para la interpretación de sus resultados.

Existen diferentes soluciones y opciones de software que permiten realizar el escaneo de vulnerabilidades, tanto en la red, los sistemas operativos, o incluso en las aplicaciones. A continuación, se mencionan algunas de las herramientas más populares.

NMAP

NMAP (*Network Mapper*) es una herramienta de código abierto que permite analizar las vulnerabilidades de un sistema por medio del escaneo de puertos y servicios. Es una poderosa herramienta que facilita la comprensión de la infraestructura de red y permite automatizar procesos relevantes de análisis de seguridad evitando el error humano para revelar posibles fallos en la seguridad de la información y de los equipos.

Cabe destacar que es una herramienta que se puede utilizar tanto para conocer y mitigar las vulnerabilidades de un sistema, pero también para explotarlas, de hecho, es una herramienta muy utilizada en diferentes fases de las pruebas de penetración realizadas a los sistemas.

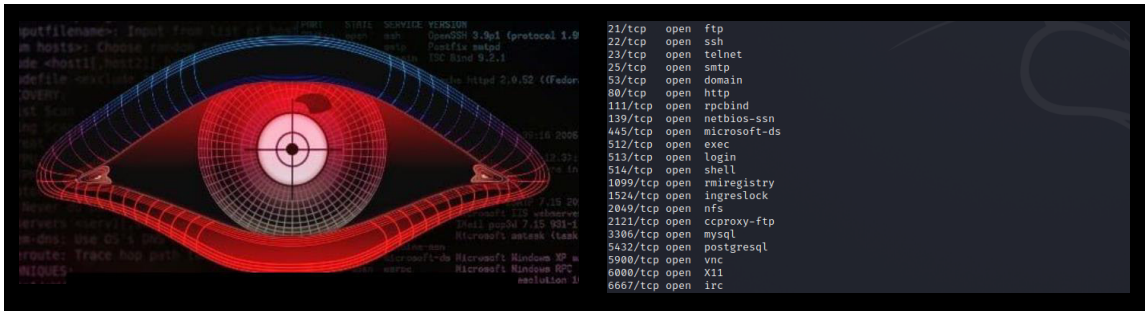

Figura 7. Composición de imágenes logotipo de Nmap y ejemplo de resultados en línea de comando.

Sitio web oficial de Nmap: **https://nmap.org/**

NESSUS

Nessus es un escáner de vulnerabilidades que ofrece diferentes opciones para el análisis de vulnerabilidades en sistemas operativos, aplicaciones, servicios en la nube y recursos de red. Esta herramienta permite la identificación de fallas en el software, falta de parches de seguridad, malware, el uso de contraseñas predeterminadas, así como errores de configuración en los sistemas informáticos. Es tal su capacidad y amplitud que se ha convertido en uno de los programas preferidos en la industria.

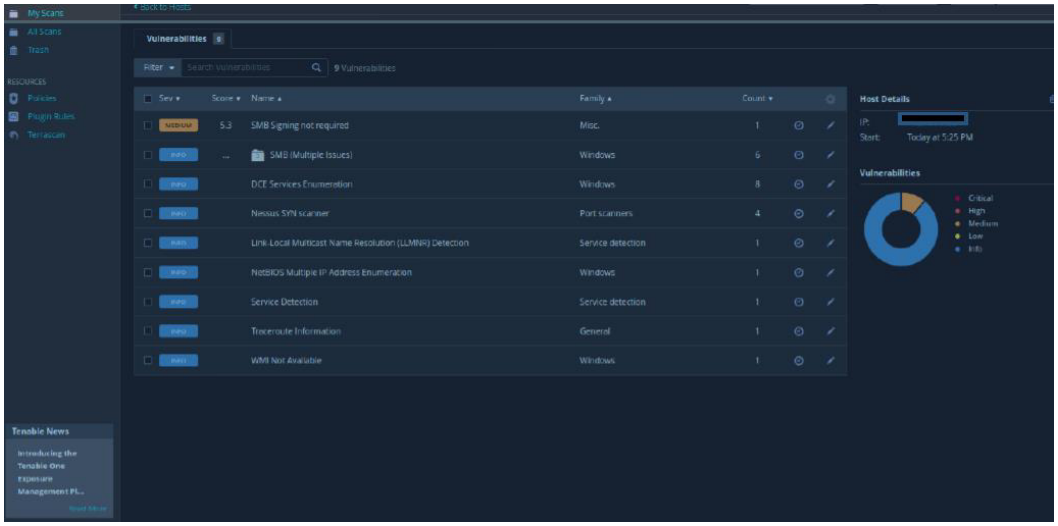

Figura 8. Captura de pantalla de la interfaz de NESSUS.

Sitio web oficial de NESSUS: https://es-la.tenable.com/products/nessus

OPENVAS

OPENVAS (*Open Vulnerability Assessment System*) es otro ejemplo de escáner de vulnerabilidades que incluye una amplia gama de pruebas automatizadas que permite la revisión y monitoreo de redes, sistemas, y aplicaciones en busca de posibles debilidades en la infraestructura. Una de las características principales de OPENVAS es su extensa base de datos de vulnerabilidades, que se actualiza regularmente para incluir las últimas amenazas y debilidades conocidas. OPENVAS es una herramienta de código abierto (es gratuita y de libre acceso) lo que la hace una opción popular para la comunidad de seguridad, aunque es preciso señalar que también cuenta con una versión de pago especializada en la valoración de vulnerabilidades en un ambiente empresarial.

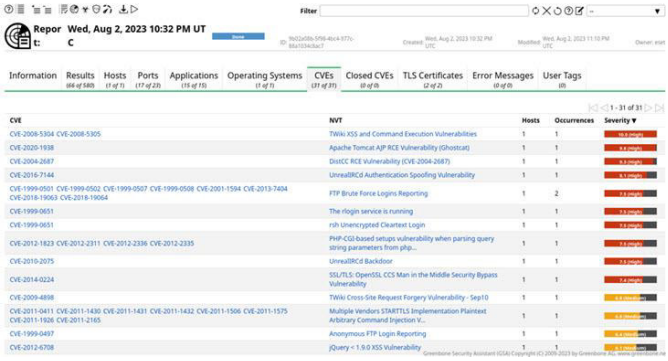

Figura 9. Interfaz OPENVAS.[4]

Sitio web oficial OPENVAS: https://www.openvas.org/

ACUNETIX

Es un escáner de vulnerabilidades especializado en auditorías de seguridad de aplicaciones web. Esta herramienta facilita la detección e identificación de errores de código y en la configuración de una aplicación web incluyendo vulnerabilidades comunes como inyección y desbordamientos de búfer entre otros. Además de su capacidad para identificar vulnerabilidades, Acunetix ofrece características avanzadas, como análisis de archivos presentes en los directorios de la aplicación, chequeos sobre la validación de datos y la capacidad de simular ataques automatizados para evaluar la resistencia de la aplicación a posibles amenazas.

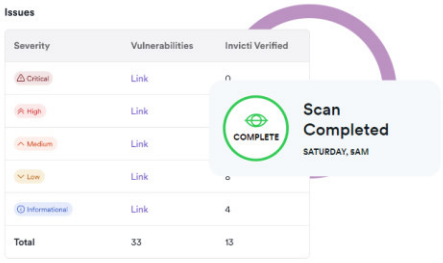

Figura 10. Interfaz Acunetix.

[4] [Interfaz OPENVAS]. Recuperado de: https://www.welivesecurity.com/es/recursos-herramientas/evaluacion-vulnerabilidades-openvas/

BURP SUITE

Debido a su versatilidad y potencia en la detección y explotación de fallos de seguridad, Burp Suite se ha convertido en el estándar de la industria para el análisis de vulnerabilidades en aplicaciones web y móviles. Una de las principales fortalezas de Burp Suite es su amplia gama de herramientas entre las que se incluyen un escáner de vulnerabilidades, un proxy HTTP/HTTPS, un intrusor, un secuenciador, un descifrador de contraseñas, entre otros. Estas herramientas permiten realizar un análisis exhaustivo de posibles fallos de seguridad en las aplicaciones web. Existen diferentes ediciones de esta herramienta: la Community Edition es gratuita y las versiones Professional y Enterprise que son versiones de pago, pero que cuentan con características adicionales.

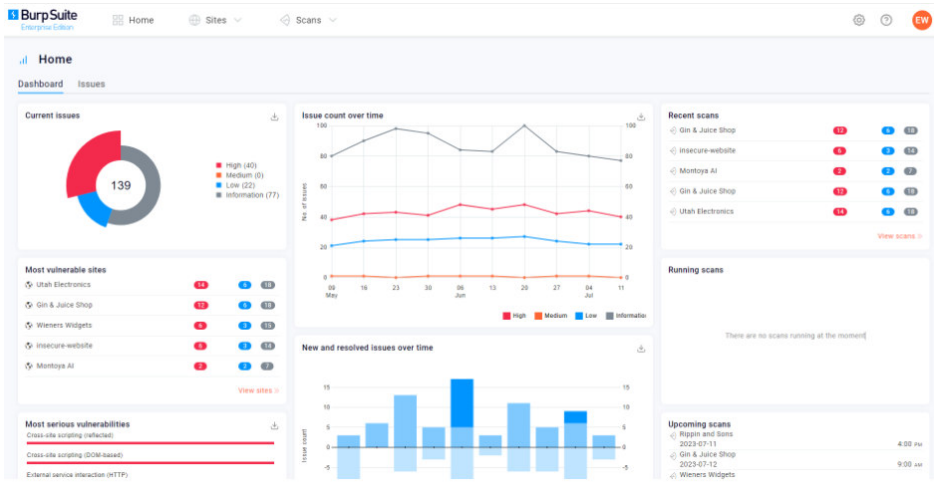

Figura 11. Captura de pantalla interfaz Burp Suite.

Página oficial Burp Suite: https://portswigger.net/burp

El proceso de detección y valoración de vulnerabilidades es un paso muy importante que permitirá su evaluación y determinar cuáles representan un mayor riesgo a la seguridad de la información. Por lo tanto, es preciso identificar estos errores antes que los intrusos para mejorar los controles de seguridad y hacer más difícil que puedan ser aprovechadas.

Por otro lado, el análisis de vulnerabilidades en nuestros sistemas se debe realizar de manera periódica, con base en el contexto específico de una organización y sus necesidades de seguridad, para determinar las mejores acciones para su mitigación.

AUTOR

ELIER EMANUEL LÓPEZ BASILIO