Unidad 4: 1. Consideraciones de prueba de seguridad para validar que el software cumpla con los requisitos y especificaciones de seguridad establecidos (y no establecidos)

El tema "Consideraciones de prueba de seguridad para validar que el software cumpla con los requisitos y especificaciones de seguridad establecidos (y no establecidos)" trata de las **técnicas y estrategias que se utilizan para evaluar la seguridad del software**. Las pruebas de seguridad son un componente fundamental del proceso de desarrollo de software seguro, ya que ayudan a identificar y corregir vulnerabilidades de seguridad antes de que el software se ponga en producción.

Las pruebas de seguridad pueden centrarse en los requisitos y especificaciones de seguridad establecidos, o también pueden buscar vulnerabilidades que no se hayan especificado explícitamente. Las pruebas de requisitos y especificaciones se basan en la documentación de seguridad del software, como el plan de seguridad, la especificación de seguridad y los requisitos de seguridad. Las pruebas no establecidas, por otro lado, se basan en la experiencia del equipo de pruebas y en el conocimiento de las vulnerabilidades comunes de seguridad.

Algunos de los aspectos más importantes que hay que tener en cuenta al realizar pruebas de seguridad son los siguientes:

- **El alcance de las pruebas**: Es importante definir claramente el alcance de las pruebas, es decir, qué partes del software se van a probar. El alcance debe incluir todas las funciones y datos sensibles del software.
- **El tipo de pruebas**: Hay muchos tipos diferentes de pruebas de seguridad, como pruebas de penetración, pruebas de auditoría, pruebas de fuzzing (introducir datos aleatorios, inválidos o inesperados) y pruebas de análisis estático. Es importante elegir el tipo de pruebas adecuado para el software y las vulnerabilidades que se quieren identificar.
- **Los recursos**: Las pruebas de seguridad pueden ser complejas y requieren recursos, como tiempo, personal y herramientas. Es importante planificar los recursos necesarios para las pruebas.

En cuanto a sitios web que traten el tema, te recomiendo los siguientes:

- OWASP: https://owasp.org/: La Open Web Application Security Project es una organización sin ánimo de lucro que proporciona recursos y herramientas para la seguridad de aplicaciones web.

- NIST: https://csrc.nist.gov/: El Instituto Nacional de Estándares y Tecnología de Estados Unidos proporciona estándares y directrices para la seguridad de la información.
- ISO/IEC 27001: https://www.iso.org/isoiec-27001-information-security.html: La norma ISO/IEC 27001 es un estándar internacional para la gestión de la seguridad de la información.

Estos sitios web proporcionan información sobre los diferentes tipos de pruebas de seguridad, las técnicas que se utilizan y las herramientas que están disponibles. También proporcionan recursos para ayudar a los equipos de desarrollo a planificar y ejecutar pruebas de seguridad efectivas.

Existen muchas herramientas disponibles para ayudar a los equipos de desarrollo a realizar pruebas de seguridad. Las herramientas se pueden clasificar en tres categorías principales:

- **Herramientas de análisis estático**: Estas herramientas **analizan** el código fuente del software en busca de vulnerabilidades de seguridad. Las herramientas de análisis estático pueden ser muy efectivas para identificar vulnerabilidades comunes, como las vulnerabilidades de inyección de código.

- **Herramientas de análisis dinámico**: Estas herramientas **ejecutan** el software en un entorno controlado para buscar vulnerabilidades de seguridad. Las herramientas de análisis dinámico pueden ser más efectivas para identificar vulnerabilidades que no se pueden encontrar con el análisis estático, como las vulnerabilidades de desbordamiento de búfer.

- **Herramientas de pruebas de penetración**: Estas herramientas simulan ataques de piratas informáticos para probar la seguridad del software. Las herramientas de pruebas de penetración pueden ser muy efectivas para identificar vulnerabilidades que podrían ser explotadas por un atacante real.

Algunos ejemplos de herramientas de pruebas de seguridad son los siguientes:

- Herramientas de análisis estático:
    - Fortify SCA
    - AppScan Source Edition
    - SonarQube
    - **Lynix**
- Herramientas de análisis dinámico:
    - OWASP ZAP

    - Burp Suite
    - Nessus
- Herramientas de pruebas de penetración:
    - **Metasploit**
    - Kali Linux
    - **Nmap**

Las herramientas de pruebas de seguridad pueden ser una herramienta valiosa para ayudar a los equipos de desarrollo a identificar y corregir vulnerabilidades de seguridad. Sin embargo, es importante tener en cuenta que las herramientas de pruebas de seguridad no son infalibles. Las herramientas pueden pasar por alto vulnerabilidades, y los resultados de las pruebas deben ser verificados por un equipo de expertos.

Vulnerabilidades de software

Las vulnerabilidades de software suelen ser provocadas por errores en el sistema operativo o en el código de la aplicación.

Un ejemplo es la vulnerabilidad SynFul Knock descubierta en el sistema operativo de interconexión de redes (IOS) de Cisco en 2015.

La vulnerabilidad SynFUL Knock permitió a los atacantes obtener el control de los enrutadores de nivel empresarial, como los enrutadores ISR de Cisco, desde los cuales podían monitorear todas las comunicaciones de red e infectar otros dispositivos de red.

Esta vulnerabilidad se introdujo en el sistema cuando una versión alterada de IOS se instaló en los routers. Para evitar esto, verifique siempre la integridad de la imagen de IOS descargada y limite el acceso físico al equipo solo al personal autorizado.

Categorización de vulnerabilidades de software

La mayoría de las vulnerabilidades de seguridad del software se dividen en varias categorías principales.

Desbordamiento de Búfer: Los búferes son áreas de memoria asignadas a una aplicación. Se produce una vulnerabilidad cuando los datos se escriben más allá de los límites de un búfer. Al cambiar los datos más allá de los límites de un búfer, la aplicación puede acceder a la memoria asignada a otros procesos. Esto puede provocar un bloqueo del sistema o comprometer los datos, o proporcionar una escalada de privilegios.

Entrada No Validada: Los programas a menudo requieren la entrada de datos, pero estos pueden tener contenido malicioso, diseñado para obligar al programa a comportarse de forma no deseada.

Por ejemplo, considere un programa que recibe una imagen para procesarla. Un usuario malintencionado podría crear un archivo de imagen con dimensiones de imagen no válidas. Las dimensiones creadas maliciosamente podrían forzar al programa a asignar búferes de tamaños incorrectos e imprevistos.

Condiciones de Carrera: Esta vulnerabilidad describe una situación en la que la salida de un evento depende de salidas ordenadas o programadas. Una condición de carrera se convierte en una fuente de vulnerabilidad cuando los eventos ordenados o cronometrados requeridos no ocurren en el orden correcto o en el momento adecuado.

Debilidad en las Practicas de Seguridad: Los sistemas y los datos confidenciales se pueden proteger mediante técnicas como la autenticación, la autorización y el cifrado. Los desarrolladores deben ceñirse al uso de técnicas de seguridad y bibliotecas que ya hayan sido creadas, probadas y verificadas y no deben intentar crear sus propios algoritmos de seguridad. Es probable que solo introduzcan nuevas vulnerabilidades.

Problemas de Control de Acceso: El control de acceso es el proceso de controlar quién hace qué y va desde administrar el acceso físico al equipo hasta dictar quién tiene acceso a un recurso, como un archivo, y qué pueden hacer con él, como leer o cambiar el archivo. Muchas vulnerabilidades de seguridad se generan por el uso incorrecto de los controles de acceso.

Casi todos los controles de acceso y las prácticas de seguridad pueden superarse si un atacante tiene acceso físico al equipo objetivo. Por ejemplo, independientemente de la configuración de permisos de un archivo, un hacker puede omitir el sistema operativo y leer los datos directamente del disco. Por lo tanto, para proteger la máquina y los datos que contiene, se debe restringir el acceso físico y se deben utilizar técnicas de cifrado para proteger los datos contra robos o corrupción.

Actualizaciones de software

El objetivo de las actualizaciones de software es mantenerse actualizado y evitar el aprovechamiento de vulnerabilidades. Microsoft, Apple y otros productores de sistemas

operativos lanzan parches y actualizaciones casi todos los días y las empresas u organizaciones responsables de las aplicaciones, como los navegadores web, las aplicaciones móviles y los servidores web, suelen actualizarlas.

A pesar de que las organizaciones se esfuerzan mucho en encontrar y reparar vulnerabilidades de software, se descubren nuevas vulnerabilidades con regularidad. Es por eso que algunas organizaciones utilizan investigadores de seguridad de terceros que se especializan en encontrar vulnerabilidades en el software, o realmente invierten en sus propios equipos de pruebas de penetración dedicados a buscar, encontrar y parchear las vulnerabilidades del software antes de que puedan ser explotadas.

Project Zero de Google es un gran ejemplo de esta práctica. Después de descubrir una serie de vulnerabilidades en varios software utilizados por los usuarios finales, Google formó un equipo permanente dedicado a encontrar vulnerabilidades de software.

Unidad 4: 2. Hacking Ético

El hacking ético, también conocido como seguridad informática ofensiva, es la práctica de evaluar la seguridad de un sistema o red informática mediante la simulación de ataques de piratas informáticos. El objetivo del hacking ético es identificar y corregir vulnerabilidades de seguridad antes de que puedan ser explotadas por piratas informáticos malintencionados.

Los hackers éticos suelen ser profesionales con experiencia en seguridad informática. Utilizan una variedad de técnicas y herramientas para evaluar la seguridad de los sistemas y redes. Estas técnicas pueden incluir:

- **Análisis estático**: Análisis del código fuente del software en busca de vulnerabilidades.
- **Análisis dinámico**: Ejecución del software en un entorno controlado para buscar vulnerabilidades.
- **Pruebas de penetración**: Simulación de ataques de piratas informáticos para probar la seguridad del sistema.
- **Pruebas de fuzzing**: Introducción de datos aleatorios, inválidos o inesperados en el sistema para buscar vulnerabilidades.

Los hackers éticos suelen trabajar con organizaciones para evaluar la seguridad de sus sistemas y redes. Las organizaciones pueden contratar a hackers éticos para realizar auditorías de seguridad, realizar pruebas de penetración o proporcionar servicios de consultoría de seguridad.

El hacking ético es una práctica importante para la seguridad informática. Ayuda a las organizaciones a identificar y corregir vulnerabilidades de seguridad antes de que puedan ser explotadas por piratas informáticos malintencionados.

Pruebas de Penetración

Las pruebas de penetración, son el acto de evaluar un sistema informático, una red o una organización en busca de vulnerabilidades de seguridad. Una prueba de penetración busca violar los sistemas, las personas, los procesos y el código para descubrir vulnerabilidades que podrían explotarse. Esta información se utiliza para mejorar las defensas del sistema y garantizar que pueda resistir mejor los ciberataques en el futuro.

Paso 1: Planificación

El Pentester recopila la mayor cantidad de información posible sobre un sistema o red de destino, sus posibles vulnerabilidades y exploits para usarlo en su contra. Esto implica llevar a cabo un reconocimiento pasivo o activo (huella) e investigación de vulnerabilidad.

Paso 2: Escaneo

La prueba de penetración lleva a cabo un reconocimiento activo para sondear un sistema o red objetivo e identificar posibles debilidades que, si se explotan, podrían dar acceso a un atacante. El reconocimiento activo puede incluir:

- escaneo de puertos para identificar puntos de acceso potenciales en un sistema de destino
- análisis de vulnerabilidades para identificar posibles vulnerabilidades explotables de un objetivo en particular
- establecer una conexión activa con un destino (enumeración) para identificar la cuenta de usuario, la cuenta del sistema y la cuenta de administrador.

Paso 3: Obtener Acceso

La prueba de penetración intentará obtener acceso a un sistema de destino y rastreará el tráfico de la red, utilizando varios métodos para explotar el sistema, que incluyen:

- lanzar un exploit con una carga útil en el sistema
- traspasar las barreras físicas a los activos
- ingeniería social
- explotar las vulnerabilidades del sitio web
- explotar vulnerabilidades o configuraciones incorrectas de software y hardware

- violación de los controles de acceso de seguridad
- descifrar Wi-Fi encriptado débil.

Paso 4: Mantener el Acceso

La prueba mantendrá el acceso al objetivo para averiguar qué datos y sistemas son vulnerables a la explotación. Es importante que permanezcan sin ser detectados, por lo general, utilizando puertas traseras, caballos de Troya, rootkits y otros canales encubiertos para ocultar su presencia.

Cuando esta infraestructura esté en su lugar, la prueba procederá a recopilar los datos que considere valiosos.

Paso 5: Análisis y Reporte

La prueba de penetración proporcionará comentarios a través de un informe que recomienda actualizaciones de los productos, las políticas y la capacitación para mejorar la seguridad de la organización.

Aquí hay algunos ejemplos específicos de vulnerabilidades que han sido encontradas mediante hacking ético:

- **Vulnerabilidades de inyección de código**: Estas vulnerabilidades permiten a un atacante inyectar código malicioso en un sistema o aplicación. Por ejemplo, una vulnerabilidad de inyección de código podría permitir a un atacante tomar el control de una aplicación web o robar datos confidenciales.
- **Vulnerabilidades de desbordamiento de búfer**: Estas vulnerabilidades permiten a un atacante escribir datos más allá del límite de un búfer de memoria. Esto puede provocar una pérdida de datos o la ejecución de código malicioso.
- **Vulnerabilidades de fuerza bruta**: Estas vulnerabilidades permiten a un atacante adivinar contraseñas o claves secretas mediante un proceso de prueba y error.
- **Vulnerabilidades de seguridad de la aplicación web**: Estas vulnerabilidades afectan a las aplicaciones web y pueden permitir a un atacante robar datos confidenciales, tomar el control de una aplicación web o realizar ataques de denegación de servicio.
- **Vulnerabilidades de seguridad de la red**: Estas vulnerabilidades afectan a las redes y pueden permitir a un atacante acceder a datos confidenciales, tomar el control de una red o realizar ataques de denegación de servicio.

Hacker vs Actor de Amenazas

Como sabemos, “hacker” es un término comúnmente usado para describir a un atacante. Sin embargo, el término "hacker" tiene una variedad de significados, como los siguientes:

- Un programador inteligente capaz de desarrollar nuevos programas y cambios de código en los programas existentes para hacerlos más eficientes.
- Una red profesional que utiliza habilidades de programación sofisticadas para asegurar que las redes no sean vulnerables a los ataques.
- Una persona que trata de obtener acceso no autorizado a los dispositivos en Internet.
- Una persona que ejecuta programas para prevenir o retardar el acceso a la red de un gran número de usuarios, o para dañar o eliminar los datos en los servidores.

Un vector de ataque es una ruta por la cual un atacante puede obtener acceso a un servidor, host o red. Los vectores de ataque se originan dentro o fuera de la red corporativa. Por ejemplo, las amenazas pueden apuntar a una red a través de Internet, para interrumpir las operaciones de la red y crear un ataque de denegación de servicio (DoS).

Como se muestra en la figura, se suelen usar los términos hacker de sombrero blanco, hacker de sombrero negro y hacker de sombrero gris para describir a los hackers.

1. **Los hackers de sombrero blanco** son hackers éticos que utilizan sus habilidades de programación con fines buenos, éticos y legales. Pueden realizar pruebas de penetración de redes con la finalidad de poner en riesgo los sistemas y las redes usando sus conocimientos sobre sistemas de seguridad informática con el fin de detectar las vulnerabilidades de la red. Las vulnerabilidades de seguridad son reportadas a los desarrolladores y al personal de seguridad quienes intentaran arreglar la vulnerabilidad antes que pueda ser explotada. Algunas organizaciones otorgan premios o recompensan a los hackers de sombrero blanco cuando proveen información que ayuda a identificar vulnerabilidades.

2. **Los hackers de sombrero gris** son personas que cometen delitos y realizan acciones probablemente poco éticas, pero no para beneficio personal ni para causar daños. Un ejemplo sería alguien que pone en riesgo una red sin permiso y luego divulga la vulnerabilidad públicamente. Un hacker de sombrero gris puede divulgar una vulnerabilidad a la organización afectada después de haber puesto en peligro la red. Esto permite que la organización solucione el problema.

3. **Los hackers de sombrero negro** son delincuentes poco éticos que violan la seguridad de una computadora y una red para beneficio personal o por motivos maliciosos, como ataques a la red. Los hackers de Sombrero Negro atacan las vulnerabilidades para comprometer la computadora y los sistemas de red.

Bueno o malo, el hacking es un aspecto importante de la seguridad de la red. En este curso, el término "**agente de amenaza**" se utiliza cuando se hace referencia a aquellos individuos o grupos que podrían clasificarse como hackers de **sombrero gris o negro**.