Anda abriendo

Facultad de Telemática

Ingeniería de Software

Proyecto de Innovación

Sandoval Sarrillo Sara

Tarea 2  
Actividad: “Escucha el video”  
22 de febrero del 2021

Alumno: Victor Rojas Bayardo

6° K

**Introducción o planteamiento**

En este se libro de abordara la seguridad del software, el libro de Gary McGraw (Software Security Building Security In), muestra cómo combinar el desarrollo y las pruebas para mejorar la calidad del software. Al hacerlo, presenta un marco que los desarrolladores, evaluadores y gerentes de software harían bien en adoptar. Por esta y mas razones este libro aborda temas interesantes de los cuales en su tiempo fueron muy acertados a mi parecer, lo cual me parece muy interesante.

**Desarrollo**

Capítulo 1. Definición de una disciplina

Gary nos dice que la única forma en que veo que se resuelve el enigma de la seguridad es enfrentando el problema y no los síntomas del problema, y que se necesita diseñar y construir seguridad desde el principio. Ninguna aplicación, ningún sistema operativo, ninguna pieza de middleware debería ser lanzado que aún no ha sido diseñado para la seguridad y revisado para detectar vulnerabilidades de seguridad. Solo entonces comenzaremos a librar estas nuevas batallas con las armas y tácticas correctas que nos brinden la oportunidad de ganar.

Como nos recuerda el Dr. McGraw, romper algo es más fácil que diseñar algo que no se puede romper, aunque personalmente prefiero la fórmula terrenal de Sam Rayburn, a saber: "Cualquier idiota puede derribar un granero, pero se necesita un buen carpintero para construir uno. "

El software seguro está, por definición, diseñado teniendo en cuenta las fallas.

Cheswick y Bellovin, "cualquier programa, por inofensivo que parezca, puede albergar agujeros de seguridad"

Los defectos de software con ramificaciones de seguridad, incluidos errores de implementación, como desbordamientos de búfer y fallas de diseño, como el manejo inconsistente de errores, prometen estar con nosotros durante años.

Los agujeros de seguridad en el software son comunes.

La seguridad del software consiste en comprender los riesgos de seguridad inducidos por el software y cómo gestionarlos. Las buenas prácticas de seguridad de software aprovechan las buenas prácticas de ingeniería de software e implican pensar en la seguridad en las primeras etapas del ciclo de vida del software, conocer y comprender los problemas comunes.

La mayoría de los sistemas informáticos modernos son susceptibles a los problemas de seguridad del software, entonces, ¿por qué la seguridad del software es un problema mayor ahora que en el pasado? Tres tendencias, que juntas constituyen la trinidad de problemas, tienen una gran influencia en el crecimiento y la evolución del problema.

Conectividad: La creciente conectividad de las computadoras de todo tipo a través de Internet. El acceso a través de una red no requiere intervención humana.

La extensibilidad es el grado en que los sistemas se han vuelto extensibles. Un sistema extensible acepta actualizaciones o extensiones, a veces denominadas código móvil para que la funcionalidad del sistema pueda evolucionar en una moda incremental, Por ejemplo, la arquitectura de complementos de los navegadores web facilita la instalación de extensiones. Desde un punto de vista económico, los sistemas extensibles son atractivos porque proporcionan interfaces flexibles que se pueden adaptar a través de nuevos componentes. En el mercado actual, es fundamental que el software se despliegue lo más rápido posible para ganar cuota de mercado. Sin embargo, el mercado también exige que las aplicaciones proporcionen nuevas funciones con cada versión.

La complejidad es el crecimiento desenfrenado del tamaño y la complejidad de los sistemas de información modernos, especialmente los sistemas de software. El problema de la complejidad se ve agravado por el uso de lenguajes de programación inseguros como antes mencionados en su libro “C y C ++” que no protegen contra tipos simples de ataques, como los desbordamientos de búfer, que son problemas muy comunes en ese tiempo.

La desafortunada realidad es que "más líneas, más errores", esta afirmación basada en un análisis hecho por Geer, y yo estoy de acuerdo con eso, más código, más errores, más problemas de seguridad.

La seguridad del software, es decir, el proceso de diseño, construcción y prueba de software para la seguridad, llega al corazón de la seguridad informática al identificar y eliminar problemas en el propio software. De esta forma, la seguridad del software intenta crear software que pueda resistir los ataques de forma proactiva.

Defecto: tanto las vulnerabilidades de implementación como las de diseño son defectos. Un defecto es un problema que puede permanecer inactivo en el software durante años solo para emerger en un sistema de campo con importantes consecuencias.

Error: un error es un problema de software a nivel de implementación. Pueden existir errores en el código, pero nunca se ejecutarán. Los errores son problemas a nivel de implementación que se pueden descubrir y solucionar fácilmente.

Defecto: un defecto es un problema a un nivel más profundo.

Riesgo: los defectos y los errores conllevan riesgos. Los riesgos no son fallas. Los riesgos capturan la probabilidad de que una falla o error afecte el propósito del software. Las medidas de riesgo también deben tener en cuenta el daño potencial que puede ocurrir. No solo es probable que ocurra un riesgo muy alto, sino que también es probable que cause un gran daño. Los riesgos pueden gestionarse por medios técnicos y no técnicos.

El problema con la seguridad de las aplicaciones. Se ha llegado a significar la protección del software una vez que ya está construido.

las herramientas de prueba de seguridad de aplicaciones pueden decirle algo sobre la seguridad, es decir, que se encuentra en un problema muy grave. Es decir, si su software falla en alguna de las pruebas enlatadas, tiene que hacer un trabajo de seguridad serio. Las herramientas pueden ayudar a descubrir problemas conocidos. Pero si pasa todas las pruebas con gran éxito, no sabe nada más que pasó un puñado de pruebas con gran éxito.

Otra gran debilidad es que se centran solo en la entrada a una aplicación proporcionada a través del puerto 80. El único buen uso para las herramientas de seguridad de aplicaciones es probar software comercial disponible.

El material más importante en Capítulo 1 es la introducción de los tres pilares de la seguridad del software: gestión de riesgos aplicada, mejores prácticas de seguridad del software (puntos de contacto) y conocimiento. Cada uno de los tres pilares es una necesidad para la seguridad del software

Pilar I: Gestión de riesgos aplicada: no es más que una herramienta de apoyo a la toma de decisiones a nivel empresarial: una forma de recopilar los datos necesarios para tomar una decisión acertada, basada en el conocimiento de las vulnerabilidades, amenazas, impactos y probabilidades. Pilar II: Puntos de contacto de seguridad del software: es un conjunto de mejores prácticas. Pilar III: Conocimiento: El conocimiento es más que una simple lista de cosas que sabemos o una colección de hechos. La información y el conocimiento no son lo mismo y es importante comprender la diferencia. El conocimiento es información en contexto, información puesta en práctica mediante procesos y procedimientos.

**Conclusión**

El software seguro comprende de una buena gestión, conocimiento y uso de mejores prácticas principalmente, y estas son tan importantes que se consideran con los pilares de la seguridad de software. La seguridad de software no se refiere a dar seguridad a uno ya construido y realizar pruebas que nos dice el autor no servirían mas que para saber que tan mal esta en seguridad, que tanta calidad tiene ese software cuando fue construido. El software seguro comienza desde el sistema donde será instalado hasta su mantenimiento. Este libro fue muy interesante para mí y abrió más perspectivas de cómo es la seguridad de software.

**Bibliografía**

* Edgescan: Vulnerability Stats Report. (2019). Network Security, 2019(3), 3-18. https://doi.org/10.1016/s1353-4858(20)30027-1
* Wesley, A. (2006). Software Security - Building Security In [Software Security] (Vol. 1). Recuperado de https://doi.org/10.1109/ISSRE.2006.43