TALENTO TECH

TIC

REGIÓN 3

CAUCA – NARIÑO

LECCIÓN 2 – UNIDAD 1

Universidad Tecnológica de Pereira

faceIT

Métricas de seguridad:

¿Cómo se puede saber si un sistema es seguro?

Es imposible desarrollar y entregar un sistema que no tenga fallas de seguridad. La seguridad va mejorando a medida que una solución se mejora y se incrementa. A pesar de que un sistema invulnerable no es posible, si se pueden tener sistemas con altos estándares de seguridad.

Para la implementación de la estrategia de devsecops es necesario una primera etapa de planificación. Como en cualquier primera etapa, habrá usuarios que no quieran discutir los aspectos de seguridad, pero es necesario incluirlos desde el inicio de un proyecto.

Para iniciar se debe responder a las siguientes preguntas:

¿Qué marco normativo en la seguridad en la información se debe cumplir?
¿Qué experiencia tiene el equipo de personas en la implementación de estrategias de seguridad en la información?
¿Cómo podemos saber si se ha producido una vulneración de seguridad?
¿Cómo se están registrando los eventos en el sistema? ¿Hay trazabilidad o reportes?

Con esta información se debe crear un modelo de amenazas.

Al tener un modelo de amenazas es necesario realizar el análisis del código, preferiblemente con herramientas que automaticen ese proceso. Las herramientas suelen ayudar a encontrar vulnerabilidades comunes reportadas, como por ejemplo mala gestión de errores, fallas que permiten la inyección SQL, bibliotecas de terceros con fallas o desactualizadas y mejoras.
Una vez encontradas las fallas, se genera una lista de las acciones para arreglarlas. Normalmente es una lista de tareas en el backlog de uno o múltiples equipos de desarrollo.

Luego de implementadas las mejoras, se hacen pruebas de seguridad y pruebas de penetración que ayuden a validar que todas las vulnerabilidades detectadas han sido reparadas.

Finalmente se incorporan las prácticas al ciclo de vida del desarrollo. De esta forma cada que un nuevo cambio se quiera incluir en una base de código, primero pasará por análisis a nivel de código, luego por pruebas de seguridad y todas las validaciones automatizadas que ayudan a mantener el código libre de vulnerabilidades conocidas.

Es recomendable constantemente evaluar el sistema frecuentemente, dando por hecho que las vulneraciones y los ataques van a ocurrir y que se quiere mitigar el riesgo de pérdida de información.

Allí es donde existen algunas métricas importantes para determinar la calidad del proceso y de la seguridad.

Tiempo medio de detección Indica cuanto tiempo tarda un equipo de seguridad en darse cuenta de que hay un ataque en curso. También sirve para determinar cuanto se tarda una organización en darse cuenta de que hubo una vulneración en la seguridad. Esta métrica debe mantenerse lo más bajo posible para actuar con prontitud ante ataques y prevenir pérdida de información y daños al sistema.

Tiempo medio de recuperación

Es el tiempo en que una corporación se tarda en recuperarse de un ataque. Por ejemplo, en un caso de un ataque de denegación de servicio distribuida (DDoS) es necesario tomar las acciones que eviten que millones de solicitudes desde muchos equipos diferentes se procesen. El tiempo que tarda la aplicación en estar disponible nuevamente para los usuarios legítimos es el tiempo medio de recuperación.
En un plan de atención de desastres siempre se mide y se evalúa como reducir este tiempo.

Rendimiento de las directrices Es recomendable realizar mediciones periódicas para validar qué tan efectivas son las directrices y qué tanto cumplimiento se está dando. Por ejemplo, se pueden establecer reglas incrementales que impidan a un desarrollador enviar cambios de software si este no pasa con cierto porcentaje de cobertura en pruebas unitarias y de seguridad.

También se pueden integrar herramientas que, al encontrar fallas de seguridad potenciales en el análisis estático de código, bloqueen la posibilidad de crear artefactos que se desplegarán.

Se pueden medir a menudo los porcentajes de cobertura y de cumplimiento en la seguridad, así como la cantidad de tareas por resolver asociadas con la seguridad a lo largo del tiempo del proyecto. Siempre se espera que sea un número que esté reduciéndose con el paso del tiempo, lo que indica que el equipo de desarrollo y despliegue está alineado con las metas de seguridad en la información.

Medición de amenazas desde los vectores de ataque

Suponiendo que una persona se haga con el control de algún vector de ataque, hay que determinar los riesgos y establecer rúbricas que permitan medir los daños que el atacante puede generar.

Algunas preguntas útiles para establecer las medidas son:

¿Puede un atacante ingresar a las redes internas?

¿En caso de ingreso a qué dispositivos puede entrar?

¿Qué información puede obtener? ¿Esta información contiene usuarios y contraseñas de equipos o bases de datos?

¿En caso de tener acceso, puede el atacante cambiar los datos? ¿Puede ejecutar código o insertar código?
¿Puede el atacante manipular el proceso de compilación y despliegue del código?
¿Puede el atacante cambiar las versiones o alterar el proceso de versionamiento de artefactos?
¿Puede un atacante llegar a los entornos de desarrollo , pruebas y producción?

Para proteger el proceso se deben tener en cuentas algunas tácticas como:

- Almacenamiento de secretos en almacenes protegidos
- Eliminación de cuentas de administrador local
- Restricción de SAMR
- Credential Guard
- Eliminación de servidores de base dual
- Suscripciones independientes
- Autenticación multifactor
- Estaciones de trabajo con privilegios de acceso
- Almacenamiento de tokens, claves y contraseñas en data vaults.
- Almacenamiento de claves de cuentas de almacenamiento y credenciales para las aplicaciones en almacenes de secretos seguros.
- Evitar que una clave se guarde en múltiples almacenes.
- Validar que solo los procesos de CI/CD tengan acceso a los almacenes de secretos.

Existen tácticas corporativas que pueden ayudar a descubrir vulnerabilidades y mejorar la seguridad y las prácticas de seguridad en la información. Una de ellas, que emplea Microsoft son los juegos de guerra.