Ciclos de desarrollo de software seguro

Para minimizar los riesgos de que se produzcan vulnerabilidades, se deben abordar todos los aspectos relacionados con la seguridad en los primeros ciclos del desarrollo. El coste de encontrar vulnerabilidades en las primeras fases es muy inferior al de detectarlas en una aplicación en producción. Según los estudios del NIST, el coste de un problema de seguridad puede llegar a multiplicarse por 25 desde la fase de diseño. En algunos casos el impacto puede llegar a resultar en pérdidas económicas y reputacionales. Uno de los procesos de desarrollo más conocidos es el Secure Development Lifecycle (SDL) de Microsoft, que estandariza un conjunto de buenas prácticas a aplicar durante el desarrollo de un producto. También existe una versión ágil de la metodología SDL que define los elementos que se deberían ejecutar en cada sprint o iteración.

Formación

En una primera fase se debe impartir formación en seguridad a todos los miembros del equipo.

- Codificación segura
- Pruebas
- Privacidad

Implementación

Se definirá un documento de buenas prácticas de implementación relacionado con aspectos de seguridad.

SAST

- Diseñadas para analizar el código fuente o el código compilado para encontrar problemas de seguridad
- Son muy útiles para detectar algunos tipos de vulnerabilidades como Inyección SQL o desbordamiento de buffer
- Su principal inconveniente es que no son capaces de detectar muchos tipos de vulnerabilidades como problemas de autenticación.
- Una de estas herramientas es SonarQube que detecta problemas en el código y general. Puede ser itegrado con netbeans, inteliJ y Eclipse.

Pruebas

- Revisión de código por parte de otros miembros
- Realizar pruebas de caja negra
- Uso de herramientas DAST (Dynamic Application Security Testing) que realizan pruebas automáticas de caja negra sobre aplicaciones. (ZAP sería un ejemplo)

Revisión de código

Se debe poner atención a:

- Autenticación
- Autorización
- Excepciones