Seguridad en aplicaciones Web

Javier Albert Segui

Curso 2020/21

Universidad de Alcalá

Seguridad Web

- La seguridad web es la acción/práctica de proteger sitios web del acceso, uso, modificación, destrucción o interrupción, no autorizados.
- La seguridad de sitios web eficaz requiere de esfuerzos de diseño a lo largo de la totalidad del sitio web: en tu aplicación web, en la configuración del servidor web, en tus políticas para crear y renovar contraseñas, y en el código del lado cliente.
- Debemos seguir siempre la siguiente premisa

Planear – Hacer – Verificar – Actuar

Aplicaciones seguras

- Las aplicaciones seguras no se dan por si mismas, son el resultaado de un esfuerzo en todos los implicados en las aplicaciones.
- Para conseguir una aplicación segura necesitamos:
  - Una organización que abogue por la seguridad
  - Políticas de seguridad documentadas y basadas en estándares
  - Metodologías de desarrollo con puntos de control y actividades de seguridad
  - Gestión segura de versiones y configuraciones.

Metodologías de desarrollo

- La elección de una u otra metodología no es tan importante como el hecho de poseer una.
- Debemos buscar las siguientes características:
  - Fuerte aceptación de las fases de diseño, testeo y documentación.
  - Espacios donde poder insertar controles de seguridad (Análisis de riesgos, amenazas, revisiones de código....)
  - Que sea funcional para el tamaño de la organización
  - Tenga potencial para reducir la tasa de errores y mejorar la productividad de los desarrolladores

Estándares de codificación

- Una metodología no es un estándar de codificación
- Deberíamos considerar:
  - Orientación de la arquitectura
  - Niveles mínimos de documentación
  - Requisitos de testeo obligatorios
  - Niveles mínimos de comentarios
  - Uso de los bloques de control
  - Método de nombrado de variables, métodos, clases y tablas.
  - Código lo mas legible posible antes que código complejo e "ideas felices"
  - Control del código fuente

Seguridad de la información

- La seguridad de la información se basa en 3 pilares que son:
  - **Confidencialidad:** permitir acceso únicamente a los datos a los cuales el usuario tiene permitido
  - **Integridad:** Asegurar que los datos no se falsifican o alteran por usuarios no autorizados
  - **Disponibilidad:** Asegurar que los sistemas y los datos están disponibles para los usuarios autorizados cuando lo necesitan

Principios de la codificación segura

- Minimizar la superficie de ataque
- Seguridad por defecto
- Principio del mínimo privilegio
- Principio de la defensa en profundidad
- Fallos de manera segura
- Separación de funciones
- Simplicidad
- Parches de seguridad

Pruebas

- Las pruebas o testing son un proceso de comparación del estado de algo ante un conjunto de criterios.
- Muchas veces este conjunto de criterios es difuso y no está estandarizado o escrito en un documento de pruebas.
- Las pruebas no son una fase adicional al desarrollo de software, sino que deben estar inmersas durante todo el ciclo de vida del mismo.
- Utiliza las herramientas adecuadas.
- Desarrolla métricas
- Por tanto, debemos tener la máxima de "Prueba pronto, prueba a menudo"

Técnicas de pruebas

- Inspecciones y revisiones manuales
  - Las inspecciones manuales son revisiones realizadas por personas, que por lo general comprueban las implicaciones de seguridad de personas, políticas y procesos, aunque pueden incluir la inspección de decisiones tecnológicas, como puede ser los diseños de la arquitectura escogidos
- Modelado de amenazas
  - es una técnica popular para ayudar a los diseñadores de sistemas acerca de las amenazas de seguridad a las que se enfrentan sus sistemas. Les permite desarrollar estrategias de mitigación para vulnerabilidades potenciales. El modelado de amenazas ayuda a las personas a concentrar sus, inevitablemente, limitados recursos y atención en aquellas partes del sistema que más lo necesitan

Técnicas de pruebas

- Revisión de código
  - es el proceso de comprobar manualmente el código fuente de una aplicación web en busca de incidencias de seguridad.
- Pruebas de intrusión
  - Las pruebas de intrusión son esencialmente el "arte" de comprobar una aplicación en ejecución remota, sin saber el funcionamiento interno de la aplicación, para encontrar vulnerabilidades de seguridad

Top 10 de Riesgos en las aplicaciones Web

- **Inyección**: Las fallas de inyección, como SQL, NoSQL, OS o LDAP ocurren cuando se envían datos no confiables a un intérprete, como parte de un comando o consulta. Los datos dañinos del atacante pueden engañar al intérprete para que ejecute comandos involuntarios o acceda a los datos sin la debida autorización.
- **Perdida de autenticación**: Las funciones de la aplicación relacionadas a autenticación y gestión de sesiones son implementadas incorrectamente, permitiendo a los atacantes comprometer usuarios y contraseñas, token de sesiones, o explotar otras fallas de implementación para asumir la identidad de otros usuarios (temporal o permanentemente).
- **Exposición de datos sensibles**: Muchas aplicaciones web y APIs no protegen adecuadamente datos sensibles, tales como información financiera, de salud o Información Personalmente Identificable (PII).Los datos sensibles requieren métodos de protección adicionales, como el cifrado en almacenamiento y tránsito

Top 10 de Riesgos en las aplicaciones Web

- **Entidades Externas XML**: Muchos procesadores XML antiguos o mal configurados evalúan referencias a entidades externas en documentos XML.
- **Perdida de control de acceso**: Las restricciones sobre lo que los usuarios autenticados pueden hacer no se aplican correctamente. Los atacantes pueden explotar estos defectos para acceder, de forma no autorizada, a funcionalidades y/o datos.
- **Configuración de Seguridad Incorrecta**: La configuración de seguridad incorrecta es un problema muy común y se debe en parte a establecer la configuración de forma manual, ad hoc o por omisión (o directamente por la falta de configuración).

Top 10 de Riesgos en las aplicaciones Web

- **Cross Site Scripting (XSS):** es un tipo de vulnerabilidad informática o agujero de seguridad típico de las aplicaciones Web, que puede permitir a una tercera persona inyectar en páginas web visitadas por el usuario código JavaScript o en otro lenguaje similar
- **Deserialización Insegura:** Estos defectos ocurren cuando una aplicación recibe objetos serializados dañinos y estos objetos pueden ser manipulados o borrados por el atacante para realizar ataques de repetición, inyecciones o elevar sus privilegios de ejecución

Top 10 de Riesgos en las aplicaciones Web

- **Componentes con vulnerabilidades conocidas:** Los componentes como bibliotecas, frameworks y otros módulos se ejecutan con los mismos privilegios que la aplicación.
- **Registro y Monitorización Insuficientes:** El registro y monitoreo insuficiente, junto a la falta de respuesta ante incidentes permiten a los atacantes mantener el ataque en el tiempo, pivotear a otros sistemas y manipular, extraer o destruir datos.

Recursos

- OWASP
  - https://owasp.org/
  - Top Ten: https://owasp.org/www-project-top-ten/
  - Testing Guide: https://owasp.org/www-project-web-security-testing-guide/
  - Development Guide: https://owasp.org/www-pdf-archive/Owasp_Dev_Guide.pdf