GRUPO DE SEGURIDAD INFORMÁTICA

Taller de Seguridad Informática

Seguridad en aplicaciones
ClickJacking

Contenido

- Introducción
- Problemas de seguridad en el software
- Mejores prácticas
- Principales ataques basados en navegadores
- Click Jacking

Introducción

- El software es seguro si puede manejar entradas malformadas intencionalmente
- El software es seguro si puede proteger la integridad del sistema en ejecución

Gollmann

Introducción

- Software seguro es el que está diseñado para soportar ataques maliciosos.

- Software seguro NO es el que implementa funciones de seguridad.

Problemas de seguridad

G. McGraw

- **Defecto:** Vulnerabilidades de diseño e implementación. Puede permanecer latente por años y sale a la superficie con consecuencias importantes.

Tipos de defecto

- **Bug:** Errores simples de implementación. Pueden existir en el código y nunca ser ejecutados. Pueden ser descubiertos y corregidos rápidamente. Ej: Validación incorrecta de entradas
- **Falla:** Instanciados en el código pero presentes en el diseño. Ej: Manejo incorrecto de errores (fail to open), Microsoft Bob.

Análisis de Riesgo

- **Riesgo:** Probabilidad de que una falla o un bug impacten en el propósito de un software.
  - Riesgo = probabilidad x impacto

Resolviendo el problema

- Tres pilares fundamentales:
  - **Gestión del riesgo:** identificar, clasificar, hacer seguimiento y entender los riesgos
  - **Conjunto de buenas prácticas:** aplicadas durante el desarrollo
  - **Conocimiento:** recolectar, encapsular y compartir el conocimiento en seguridad que puede ser usado para construir bases sólidas de buenas prácticas

Mejores prácticas

- Prevención:
  - Análisis de riesgos en diseño y especificación
  - Tipos seguros (Type safe)
- Detección:
  - Inspección del código
  - Testing

Mejores prácticas

- Mitigar:
  - Mínimo privilegio
- Reacción:
  - Actualización

Ataques basados en navegadores

- Cross-Site Scripting (XSS) (17 años)
- Cross-Site Request Forgery (CSRF) (13 años)
- ClickJacking (5 años)

XSS

- Inyecta código JavaScript en aplicaciones Web

  https://mymail.com/search?foo<script>doBadStuff()</script>

- Le da a un atacante control de la sesión de usuario y los datos
- Ejecuta comandos o inyecta datos

CSRF

http://mybank.com/transfer?amt=10000&acct=badguy

- Permite a un atacante tomar acciones como un usuario
- Sólo puede escribir; no lee los resultados
- Se soluciona agregando tokens randómicos a los requests

http://mybank.com/transfer&amt=50&acct=friend&token=e43d2af7ecb

ClickJacking

- Descubierta en 2008 por Robert Hansen & Jeremiah Grossman
- Es una técnica de ataque para engañar a usuarios Web, haciendo que realicen acciones en sitios web sin saberlo
- Permite realizar acciones maliciosas en los sitios a los que el usuario está logueado

ClickJacking

- Básicamente consiste en enmarcar la página victima en un *iframe* transparente, que se coloca encima de lo que parece ser una página normal
- Cuando el usuario interactúa con la página normal está involuntariamente interactuando con la página victima
- Permite a los atacantes realizar acciones como si fueran los usuarios

Ejemplo gráfico

(Stallings. Cryptography and network security. Fig. 16.5)

Diferentes técnicas Básicas

- Visible completamente

Borrar

- Invisible con un botón

Presione aquí

- Invisible y siguiendo el ratón

Borrar

Diferentes técnicas Next Generation

- Inyección de texto
- Extracción de contenido
- Extracción de código fuente HTML
- Otras

GRUPO DE SEGURIDAD INFORMÁTICA

Bibliografía y Referencias

- **D. Gollmann**, *Computer Security*, 2006.
- **G. McGraw**, *Software Security: Building Security In*, 2006
- **R. Hansen, J. Grossman,** *Click Jacking*, http://www.sectheory.com/clickjacking.htm, **2008.**
- **C. Jackson et all,** *Busting Frame Busting: a Study of Click Jacking Vulnerabilities on Popular Sites*, 2010.

Bibliografía y Referencias

- **Paul Stone,** *New attacks against framed web pages, http://www.contextis.co.uk/resources/white-papers/clickjacking/Context-Clickjacking_white_paper.pdf*
- **World Wide Web Consortium**, *Inline frames: The IFRAME element, http://www.w3.org/TR/REC-html40/present/frames.html#h-16.5*.