GRUPO DE SEGURIDAD INFORMÁTICA

Fundamentos de la Seguridad Informática

Seguridad en aplicaciones Web: Autenticación

Introducción

- Necesaria para identificar los potenciales usuarios de las aplicaciones
- Varias técnicas, generalmente conocidas
- La mayoría de los entornos de desarrollo contemplan alguna de estas formas
- Posibles problemas

Plan

- Técnicas comunes de autenticación
- Passwords, cuentas y nombres de usuarios
- Auto registración
- CAPTCHAs
- Mejores prácticas

Autenticación

- Juntar una identidad del sistema a un usuario individual mediante el uso de una credencial
- Proveer controles de autenticación razonables dados los riesgos en la aplicación
- Negar el acceso a los atacantes que usan varios métodos para atacar el sistema de autenticación

Técnicas comunes de autenticación web

- HTTP Basic y Digest (DAA)
- Basada en formularios
- Basada en certificados
- “Fuerte”
- Federada (+SSO)
- FIDO

Autenticación Basic y Digest

- Basic
  - envía las credenciales en texto claro, codificadas en Base64. No debería usarse a no ser combinada con SSL
- Digest HTTP
  - usa un mecanismo de challenge-response, que es razonablemente seguro para aplicaciones de bajo costo. Está atado a MD5....

Problemas de Basic y Digest

- Las razones en contra del uso de estos tipos de autenticación son:
  - Transmisión insegura de las credenciales
  - Ambas formas de autenticación son pasibles de ataques de replay y man-in-the middle
  - Ambas requieren TLS para proveer algún tipo de confidencialidad e integridad
  - Interfaz de usuario primitiva

Usando Formularios

- Provee al diseñador de la aplicación mayor control sobre la interfaz hacia el usuario y los datos, por ello se utiliza ampliamente
- Disponible y bien soportada por múltiples plataformas

Problemas de los formularios

- Ataques de replay
- Man in the middle
- Credenciales en texto claro si no se usa sobre HTTPS
- Ataques de engaños (phishing)
- Débil control de passwords

Certificados

- Está ampliamente implementada en varios servidores de aplicación y web
- El sitio web genera certificados (o intenta confiar en certificados generados externamente)
- Los certificados públicos se cargan el la base de autenticación del servidor web
- Se comparan con los ofrecidos en conexiones recibidas de navegadores

Problemas con los certificados

- Muchos usuarios comparten sus PC's y necesitan llevar sus certificados con ellos
- El manejo de certificados en un navegador no es trivial en muchos casos
- Certificados de revocación con certificados emitidos por nosotros mismos es casi imposible en redes externas
- La confianza en servidores con certificados “privados” requiere que el usuario final tome decisiones de

Autenticación “fuerte”

- Puede proporcionar un nivel más alto de seguridad que usuario y password solamente
- Ejemplos de ésto serían:
  - Biometrics
  - One-Time password
  - Challenge-Response
  - SMS Challenge-Response
  - Transaction Signing

Problemas de estos tipos de autenticación

- La mayoría de los frameworks de aplicación son difíciles de integrar con los mecanismos de autenticación fuertes, con la posible excepción de certificados, presente en J2EE y .NET
- Debemos integrar el código con un servidor de autenticación, e implícitamente confiar en los resultados
- La mayoría de las organizaciones ven estos tipos de métodos como "costosos", para los riesgos que podrían enfrentar

Autenticación federada

- Permite hacer outsourcing de la base de usuarios a un tercero, para tener varios sitios con single-signon
- Ventajas:
  - Reduce la cantidad total de credenciales que los usuarios deben recordar
  - Puede ser apropiada con el/los sitio/s son parte de una alianza de ventas o comercio grande
  - Permite proveer servicios personalizados a usuarios que de otra forma serían anónimos

Autenticación federada (2)

- No debería usarse, a no ser que:
  - Se confíe en el proveedor de autenticación
  - Los requerimientos de privacidad sean alcanzados por el proveedor de autenticación
- Ejemplos
  - SAML, de Liberty Alliance
  - Microsoft Passport/Live
  - OpenID, OAUTH

El navegador y "Remember password"

- Los navegadores modernos ofrecen a los usuario el manejo de múltiples credenciales guardándolos de forma insegura
- Este es un riesgo particularmente grave para aplicaciones que contienen información sensible o financiera
- ¿Cómo protegerse?
- Se puede enviar en el HTTP.

Reset del password automático

- Son comunes en donde las organizaciones creen que necesitan evitar costos de mesas de ayuda para la autenticación
- Desde la perspectiva del manejo de riesgo, la funcionalidad parece aceptable en varias circunstancias
- Igualmente, dicha funcionalidad equivale a un mecanismo secundario, pero mucho más debil, de passwords

Logout

- Todas las aplicaciones deberían tener un método de salir de la aplicación
- Es particularmente vital para aplicaciones que contienen datos privados o pueden ser usadas para robo de identidad
- Se debe asegurar además que el logout borre todas la cookies y variables asociadas a la sesión
- También se puede incluir un texto que alerte al usuario

CAPTCHA

- "Completely automated public Turing test to tell computers and humans apart"
- Permiten a los diseñadores web bloquear a usuarios no humanos de registrarse con el sitio
- La razón tradicional para implementar un CAPTCHA es de prevenir a los spammers de registrarse y polucionar la aplicación con spam

Mejores prácticas

- La autenticación será tan fuerte como el proceso de gestión de usuarios
- Usar la forma más apropiada de autenticación de acorde a la clasificación de activos:
  - Por ejemplo, usuario y password sirven para sistemas de bajo valor, como blogs o foros, un SMS challenge-response puede servir para sistemas de e-commerce de bajo perfil (in 2005), mientras que transacciones firmadas sirven para sistemas de alto valor, bancos on-line, etc.

Mejores prácticas (2)

- Reautenticar a los usuarios para transacciones de alto valor y acceso a áreas protegidas
- Autenticar la transacción, no el usuario
  - Los *phishers* se basan en pobres esquemas de autenticación. El valor real de la seguridad en este caso sirve para identificar transacciones fraudulentas, no tanto por los usuarios (dado que se pueden perder/robar los usuarios y passwords)
- Los passwords únicamente no sirven para sistemas de gran valor!

Bibliografía y material de referencia

- D. Gollman, *Computer Security*, Wiley, 2006
- W. Stallings, *Cryptography and Network Security,* Prentice Hall, 2006.
- R. Anderson, *Security Engineering – A Guide to Building Dependable Distributed Systems,* Wiley, 2001
- OWASP, Open Web Application Security Project, http://www.owasp.org