
GRUPO DE SEGURIDAD INFORMÁTICA

# Fundamentos de la Seguridad Informática

## Seguridad en aplicaciones: Gestión de sesiones Web

# Introducción

- Las aplicaciones “finas” de clientes guardan de forma innata datos locales
- Por diseño, el protocolo HTTP no tiene estado, y no mantiene nativamente una conexión entre el navegador cliente y el software del servidor
- Aplicaciones web no triviales pueden requerir mantener un estado a lo largo de una sesión

# Solución: sesiones Web!

- ¿Cómo se implementan? Algo compartido entre servidor Web y cliente
  - Cookies (RFC 6265: HTTP State Management Mechanism)
  - URL Rewriting

# Problemas

- Variables de sesión expuestas
- Falsificación
- Secuestro de sesiones
- Validación
- Best practices

# Variables de sesión expuestas

- Algunos frameworks usan áreas compartidas en el disco del servidor web para guardar datos de sesión
- Por ejemplo, PHP usa el /tmp en Unix y c:\windows\temp en Windows por defecto
- Pueden comprometer a la aplicación si el servidor web se comparte o es comprometido

# Usando tokens en acceso a páginas

- Algunos tokens (nonces) pueden usarse en conjunción con tokens específicos de sesión para proveer medidas de la autenticidad en pedidos de clientes
- El cliente del otro lado de la sesión es el mismo que pidió la página en la misma sesión
- Pueden guardarse en cookies o en las propias query strings, y deben ser completamente al azar

# Usando tokens en acceso a páginas (2)

- Crear un mapeo entre el cliente y el token en el lado del servidor
- Esta técnica debería incrementar la dificultad de usar fuerza bruta con los tokens de autenticación de la sesión

# Algoritmos criptográficos débiles en sesiones

- Si se generan tokens predecibles, un atacante no necesita capturar las variables de sesión de los usuarios remotos
- Puede simplemente adivinar algunos identificadores de sesión
- Los tokens de sesión deben ser únicos, no predecibles, y resistentes a ingeniería reversa

# Regeneración de tokens de la sesión

- Para reducir el riesgo de secuestro de sesiones y ataques por fuerza bruta, el servidor HTTP puede sin esfuerzo expirar y regenerar los tokens
- Esto decrece la ventana de oportunidad para un ataque por replay o fuerza bruta

# Falsificando la sesión

- Muchas aplicaciones web toman medidas en contra de intentos de adivinar claves
- Es menos común para las aplicaciones web detectar muchos intentos de continuar sesiones basadas en adivinar identificadores de sesión
- Los servidores de aplicación rara vez hacen logs o auditan dichos intentos

# Captura del token de sesión

- Si puede capturarse un token en tránsito, es posible hacer secuestro de dicha sesión
- Es más probable que suceda si sólo usamos HTTP
- Nunca debe transmitirse en texto claro
- Usar otro token para áreas que requieran mayor nivel de seguridad

# “Secuestro” de la sesión

- Si el identificador se transmite via un parámetro en la URL en vez de una cookie, los GETs pueden guardarse en la historia, el cache, etc.
- Usando pedidos mediante POST puede ayudar a aliviar este problema
- Hay propuestas de usar la IP dentro de la asociación a las sesiones - ¿Qué pasa con NAT/Proxies?

# Tokens de sesión al salir

- Se debe invalidar el token y borrar el identificador de sesión al salir el usuario
- En general, las cookies se asocian a la vida de la ventana del browser
- En una máquina compartida, el identificador de sesión puede quedar en el navegador y ser utilizado por el próximo usuario

# Ataques de validación de la sesión

- Como cualquier dato, la variable de sesión debe ser validada para asegurar que está de la forma correcta
- Que no contiene caracteres no esperados, y es válida en la tabla de sesiones
- Por ejemplo
  - usar bytes nulos para truncar objetos de sesión y debido a errores de codificación se comparaba el largo del string más corto

# Best practices

- Usar un manejador de sesión robusto y bien conocido dentro de algún framework de aplicación
- Siempre usar las versiones más actualizadas porque pueden contener bugs
- Si se está en duda, no deberían tomarse riesgos y guardar la información sensible del estado en sesiones en el servidor

# Best practices (2)

- Siempre, por motivos de seguridad, guardar del lado del servidor
- Los datos para tomar decisiones no críticas, como por ejemplo el lenguaje o el tema pueden guardarse en datos en el cliente
- Los campos ocultos nunca deberían usarse para guardar información de estado sensible

# Bibliografía y material de referencia


- D. Gollman, *Computer Security*, Wiley, 2006
- OWASP, Open Web Application Security Project, http://www.owasp.org
- G. Ollmann, "Web Based Session Management: Best Practices in Managing HTTP Based Client Sessions", http://www.technicalinfo.net/papers/WebBasedSessionManagement.html