Curso: (62612) Diseño de aplicaciones seguras

Fernando Tricas García

Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
http://webdiis.unizar.es/~ftricas/
http://moodle.unizar.es/
ftricas@unizar.es

Tema XIV: En la web

Fernando Tricas García

Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
http://webdiis.unizar.es/~ftricas/
http://moodle.unizar.es/
ftricas@unizar.es

Ideas iniciales

- Aunque los programas hechos para la web pueden sufrir de todo lo que venimos hablando hasta ahora, hay algunos aspectos especialmente interesantes
- Recordar: que nosotros no seamos capaces, no quiere decir que nadie lo sea
- Veremos ejemplos de 'casos malos' para comprender mejor el problema

Ejemplo real–de–verdad

En un servidor web cualquiera ...

```
==> /var/log/apache/access_log <==
68.103.169.233 - - [04/Apr/2004:01:25:54 -0500] "SEARCH
/\x90\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02
\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02
\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02
\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02
\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02
\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x0
```

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Otro ejemplo

```
sshd[20178]: Illegal user test from 210.127.243.85
sshd[20558]: input_userauth_request: illegal user test
sshd[20558]: Failed password for illegal user test from 210.127.243.85 port 45024 ssh2
sshd[20558]: Received disconnect from 210.127.243.85: 11: Bye Bye
sshd[22571]: Illegal user guest from 210.127.243.85
sshd[2814]: input_userauth_request: illegal user guest
sshd[2814]: Failed password for illegal user guest from 210.127.243.85 port 45048 ssh2
sshd[2814]: Received disconnect from 210.127.243.85: 11: Bye Bye
sshd[23756]: Illegal user admin from 210.127.243.85
sshd[27715]: input_userauth_request: illegal user admin
sshd[27715]: Failed password for illegal user admin from 210.127.243.85 port 45084 ssh2
sshd[27715]: Received disconnect from 210.127.243.85: 11: Bye Bye
sshd[17588]: Illegal user admin from 210.127.243.85
sshd[30925]: input_userauth_request: illegal user admin
sshd[30925]: Failed password for illegal user admin from 210.127.243.85 port 45131 ssh2
```

Otro ejemplo

```
sshd[20178]: Illegal user test from 210.127.243.85
sshd[20558]: input_userauth_request: illegal user test
sshd[20558]: Failed password for illegal user test from 210.127.243.85 port 45024 ssh2
sshd[20558]: Received disconnect from 210.127.243.85: 11: Bye Bye
sshd[22571]: Illegal user guest from 210.127.243.85
sshd[2814]: input_userauth_request: illegal user guest
sshd[2814]: Failed password for illegal user guest from 210.127.243.85 port 45048 ssh2
sshd[2814]: Received disconnect from 210.127.243.85: 11: Bye Bye
sshd[23756]: Illegal user admin from 210.127.243.85
sshd[27715]: input_userauth_request: illegal user admin
sshd[27715]: Failed password for illegal user admin from 210.127.243.85 port 45084 ssh2
sshd[27715]: Received disconnect from 210.127.243.85: 11: Bye Bye
sshd[17588]: Illegal user admin from 210.127.243.85
sshd[30925]: input_userauth_request: illegal user admin
sshd[30925]: Failed password for illegal user admin from 210.127.243.85 port 45131 ssh2
```

```
/var/log/auth.log.2.gz:Nov 19 19:13:45 xxxx sshd[10087]: Failed passwor
for invalid user bob from xx.224.171.17 port 39751 ssh2

/var/log/auth.log.3.gz:Nov 13 10:35:50 xxxx sshd[2836]: Failed password
invalid user info from xx.251.11.128 port 45670 ssh2

/var/log/auth.log.3.gz:Nov 13 10:35:54 xxxx sshd[2838]: Failed password
invalid user support from xx.251.11.128 port 63806 ssh2

/var/log/auth.log.4.gz:Nov  6 18:32:01 xxxx sshd[16817]: Failed passwor
```

El protocolo HTTP

El navegador se conecta a una página web ...

```
GET / HTTP/1.0
Host: www.ejemplo.com
Accept: text/html, text/plain, image/*
Accept-Language: en
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)
```

Métodos: GET, POST, HEAD, ...

Los métodos

GET

Los datos van en la URL

POST

Los datos van incluidos en el cuerpo de la petición

HEAD

Igual que GET, pero el servidor envía la información sin 'body'

PUT

Lo que se envía debe almacenarse donde se indica.

DELETE

Borrar un recurso.

...

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

La respuesta

```
HTTP/1.1 200 OK
Date: Fri, 16 Apr 2004 15:41:32 GMT
Server: Apache/1.3.26 (Unix) Debian GNU/Linux PHP/4.1.2 DAV/1.0.3
Last-Modified: Wed, 20 Aug 2003 20:31:11 GMT
Content-Length: 84
Connection: close
Content-Type: text/html

<html>
<head><title>Test</title></head>
<body>
<p>Hello, world!</p>
</body>
</html>
```

Otros métodos

```
POST /path/script.cgi HTTP/1.0
From: frog@jmarshall.com
User-Agent: HTTPTool/1.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 32

home=Cosby&favorite+flavor=flies
```

Referers: Más información interesante

- Los navegadores envían habitualmente información acerca de la página en la que estábamos cuando pinchamos el enlace
  - Primer problema: proporciona información sobre nuestra navegación (algunos usuarios lo bloquean).
  - Segundo problema: lo genera el cliente (nunca usarlo como método de autentificación o autorización)

Cuidado con las caches

- Los documentos se almacenan temporalmente
  - El navegador (en el disco y en memoria)
  - Los 'intermediarios' (*proxies*)
    - Locales
    - Pero también lejanos

La idea es buena, pero poco conveniente para algunos tipos de aplicaciones

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Memorias intermedias

- HTTP especifica distintos mecanismos en las diferentes versiones
  - HTTP 0.9 cabecera: Expires
  - HTTP 1.0 Pragma: no-cache
  - HTTP 1.1 cabecera: Cache-Control (private, no-cache, no-store)
- Lo mejor es mandarlos todos!
- Solución para pobres (cuidado):

<meta http−equiv="Expires" content="Thu, 01 Dec 1994 12:00:00 GMT" />

Las galletitas (cookies)

- HTTP no tiene estado, no hay relación entre peticiones sucesivas de los clientes
- Las 'cookies' se introdujeron para proporcionar una forma de obtenerlo
- El cliente tiene que 'recordar' un poco de información

El servidor:

```
Set-Cookie: Customer="79"; Version="1"; Path="/"; Max-Age=1800
```

El cliente:

```
Cookie: $Version="1"; Customer="79"; $Path="/"
```

Sesiones

- Las cookies no solucionan completamente el problema:
  - Tamaño limitado
  - Manejadas por el cliente
- Los objetos de sesión son conjuntos de variables en el lado del servidor que mantienen información sobre el estado
- Ahora hace falta asociarlas con el usuario: el identificador de sesión (*session id*)

Robo de sesiones

- Si un usuario es capaz de conseguir el identificador de sesión de otro, tendremos problemas
- ¿Cómo?
  - Adivinarla, calcularla, fuerza bruta, prueba y error,
  - XSS
  - Referers
  - Husmeadores (*packet sniffing*)

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Medidas contra el robo de sesiones

- La seguridad reside en mantener el secreto
- Se pueden utilizar estrategias secundarias (pero sólo ayudan)
  - La IP
    - Puede haber varios clientes con la misma
    - Puede haber un cliente con varias (mejor la red)
  - Alguna cabecera (User-Agent, p.ej.)
  - Cambiar el identificador en cada petición
  - Combinaciones ...

Gestión de sesiones

- El identificador de sesión en la URL

```
http://www.example.com/news.asp?article=27781;sessionid=IE60012219
```

- Ventajas:
  - Puede usarse sin 'cookies'
  - Se puede compartir
  - Se puede almacenar en los favoritos
- Inconvenientes
  - La URL queda registrada (posiblemente en muchos sitios)
  - Es trivial de atacar

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Gestión de sesiones

- Utilización de campos ocultos en un formulario

```
<FORM METHOD=POST ACTION="/cgi-bin/news.pl">
<INPUT TYPE="hidden" NAME="sessionid" VALUE="IE60012219">
<INPUT TYPE="hidden" NAME="allowed" VALUE="true">
<INPUT TYPE="submit" NAME="Read␣News␣Article">
```

- Ventajas:
  - No es tan obvio
  - Permite compartir información, sin compartir la sesión
  - Se puede utilizar sin 'cookies'
- Inconvenientes
  - Es trivial de atacar (con herramientas ?)
  - Si no se tiene cuidado, puede terminar en la URL (GET)

Gestión de sesiones: cookies

- Ventajas
  - Mas control sobre la duración
  - Más raro que se almacene en el camino
  - En casi todos los navegadores
- Desventajas
  - Hay gente que las bloquea
  - Las persistentes se pueden copiar
    - Limitación de tamaño
    - En cada petición

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Identificador de sesión

¡Son identificadores! (en muchos casos)

Características deseables

- Aleatorio
- Impredecible (conocido uno, no se puede saber el siguiente)
- Irreproducible (si se usa dos veces el generador, con los mismos datos de entrada, produce identificadores distintos)
- No identificativo/descriptivo
  `PHPSESSID (PHP)`,
  `JSESSIONID (J2EE)`,
  `CFID & CFTOKEN (ColdFusion)`,
  `ASP.NET_SessionId (ASP .NET)`

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Identificador de sesión

Longitud:

- Suficientemente largo (para resistir los ataques de fuerza bruta)
  (por lo menos 128 bits (16 bytes) aleatorios, más, mejor)
  - Velocidad de la conexión
  - Complejidad (no es lo mismo 0-9 que 0-9a-zA-Z)

Tranporte

- HTTPS (SSL/TLS) para toda la sesión, no sólo para el proceso de autenticación
- Atributo `Secure` para las cookies
  Las cookies sólo viajan a través de canales cifrados.
- Atributo `HttpOnly`
  No se permite a los scripts (JavaScript, VBscript) acceder a las cookies.
- Atributos `Domain` y `Path`
  Para restringir a quién y a dónde se manda la información.

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Fallos frecuentes

Sesiones predecibles

- Asignación secuencial de identificadores
- Valores cortos
- Técnicas de hash usadas comunes y fáciles (se pueden construir diccionarios)
- Ofuscación de sesiones (utilizar datos del cliente y ofuscarlos)

Para leer:

'Session ID Brute Force Exploitation', David Endler. (2001)
`http://www.cgisecurity.com/lib/SessionIDs.pdf`

'Web Based Session Management Best practices in managing HTTP-based client sessions.' Gunter Ollmann. (Un poco viejo).
`http://www.technicalinfo.net/papers/WebBasedSessionManagement.html`

'Session Management Cheat Sheet'
`https://www.owasp.org/index.php/Session_Management_Cheat_Sheet`

Validez de la sesión

Debe haber límite en el tiempo

- Cancelable por el usuario
- Expiración basada en tiempo
  - Tiempo de inactividad
  - Algún valor absoluto
- Revocación en el servidor (cambio de identificador, detección de ataques, ...)

Hay que validar

- Longitud del identificador:
  - Coincide en longitud
  - Coincide en tipo
  - No contiene elementos 'desagradables'
- Fuente del identificador
  - Asegurarse de que está donde se supone que debe (GET vs POST, ...)

¡Son datos de entrada!

Más errores

Sesiones únicas

- Es frecuente asignar un identificador de sesión al empezar, incluso sin haberse autentificado
- El problema es que, a veces, se mantiene aún después de haberse identificado

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Más errores

Sesiones únicas

- Es frecuente asignar un identificador de sesión al empezar, incluso sin haberse autentificado
- El problema es que, a veces, se mantiene aún después de haberse identificado
- ¿Problemas?
  - El identificador viajó en texto claro
  - El atacante puede ahora comportarse como usuario identificado
  - Puede incluso enviarle el identificador de sesión antes de que se identifique

    `https://banco.ejemplo.com/login.php?PHPSESSID=123ABC`
- La solución: cambiar el identificador de sesión cuando se produzca la autentificación

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Además...

- No mezclar contenidos cifrados y sin cifrar desde el mismo dominio.
  - www.example.com // secure.example.com
  - static.example.com // www.example.com

Yahoo! $\rightarrow$ yimg.com

Amazon $\rightarrow$ images-amazon.com

Estático // Dinámico (también prestaciones)

```
GET /so/js/master.js?v=4143 HTTP/1.1
Host: sstatic.net
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.2)
            Gecko/20090729 Firefox/3.5.2 (.NET CLR 3.5.30729)
Accept: */*
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://stackoverflow.com/questions/1252349
Pragma: no-cache
Cache-Control: no-cache
```

```
HTTP/1.1 200 OK
Cache-Control: max-age=604800
Content-Type: application/x-javascript
Content-Encoding: gzip
Last-Modified: Sun, 09 Aug 2009 18:45:13 GMT
Accept-Ranges: bytes
ETag: "75e6f1872119ca1:0"
Vary: Accept-Encoding
Server: Microsoft-IIS/7.0
Date: Sun, 09 Aug 2009 23:40:45 GMT
Content-Length: 10417
(... gzipped data ...)
```

Jeff Atwood. 'A Few Speed Improvements'.

`http://blog.stackoverflow.com/2009/08/a-few-speed-improvements/`

En resumen ...

- La aplicación debe ser capaz de identificar el tipo de sesión en función del identificador. Si algo no va como debe, se manda al usuario a la página de comienzo, con nuevos identificadores.
- En la parte cifrada, el comportamiento con los identificadores igual que antes
- Un modo de salir de la aplicación y cancelar la sesión, y las etiquetas 'meta' deben tratar el almacenamiento temporal (cache) adecuadamente

Phishing

En un mensaje de correo


Normas de Seguridad (Aviso)

Estimado cliente,

Entramos en contacto con Ud.para informarle que en fecha 16/08/2006 nuestro equipo de revision de cuentas identifica cierta actividad inusual en su cuenta, que ha sido verificada por nosotros, hallando todas las operaciones aceptables. Hemos realizado un escueto informe sobre todos los movimientos habidos en su cuenta el mes pasado.

Compruebe, por favor, este informe pulsando en ac

http://web.lerelaisinternet.com/rosian/bog/sbi/santander.htm
Haga clic para seguir vínculo

https://gruposantander.es/bog/sbi

Servicio De Santander Central Hispano

Esta notificaciyn de Santander fue enviada a [redacted].com . Por favor no responda a este correo electrynico, esto es un correo automatizado solo para notificaciones.

© Santander Central Hispano, 2006. Todos los derechos reservados

Phishing

En un mensaje de correo

Phishing


http://oicajamadrid.net/
yuda al usu...
Ayuda con los plugins
CAJA MADRID
oficina internet
CAJA MADRID
> Demo
> Hágase cliente
Información de seguridad
Introduzca:
1. Su identificador (D.N.I., Pasaporte, Tarjeta Residencia), sin letras, en el campo D.N.I.
2. Su clave de acceso en el campo Clave.
D.N.I.
Clave
Firma
Ir a
Inicio
> Entrar
Servicio de atención al cliente: 902 2 4 6 8 10
El servicio está optimizado para Explorer 5.0 o superior y Netscape 6.0 o superior
CAJA MADRID
Caja de Ahorros y Monte de Piedad de Madrid. CAJA MADRID. C.I.F. G-28029007. Plaza de Celenque, 2. 28013 Madrid. Inscrita en el Rº Mercantil de Madrid al folio 20, tomo 3067 General, hoja 62464, y en el Rº Especial de Cajas de Ahorros con el número 99. Código B.E.: 2038. Código BIC: CAHMESMMXXX. Entidad de crédito sujeta a supervisión del Banco de España
© Caja Madrid, 2001 - 2004. España. Todos los derechos reservados.

Phishing


http://cicajamadrid.net/
yuda al usu...
Ayuda con los plugins
CAJA MADRID
Introduzca:
1. Su identificador (D.N.I., Pasaporte, Tarjeta Residencia), sin letras, en el campo D.N.I.
2. Su clave de acceso en el campo Clave.
¿ http://cicajamadrid.net/ ?
> Demo
> Hágase cliente
Información de seguridad
Ir a
Inicio
> Entrar
Servicio de atención al cliente: 902 2 4 6 8 10
El servicio está optimizado para Explorer 5.0 o superior y Netscape 6.0 o superior
CAJA MADRID
Caja de Ahorros y Monte de Piedad de Madrid. CAJA MADRID. C.I.F. G-28029007. Plaza de Celenque, 2. 28013 Madrid.
Inscrita en el Rº Mercantil de Madrid al folio 20, tomo 3067 General, hoja 52454, y en el Rº Especial de Cajas de Ahorros con el número 99. Código B.E.: 2038. Código BIC: CAHMESMMXXX. Entidad de crédito sujeta a supervisión del Banco de España
© Caja Madrid, 2001 - 2004. España. Todos los derechos reservados.

Nadie está libre

Nadie está libre

Phishing

- Es un problema social (ingeniería social)
- Hay que educar a los usuarios (y no confiar mucho en eso)
  - Que haya una política (qué se hace, y qué no se hace)
- Que sea fácil comunicar problemas (`abusos@tudominio.es`)

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Phishing

- Educación
  - Hay que copiar la url, no pinchar en ella
  - No enviamos enlaces para pinchar
  - Nunca pedimos la clave ni datos secretos
  - Si reciben un mensaje 'raro', contactar con nosotros
- Consistencia (marca y más)
  - Cuidado con el dominio (siempre el mismo, siempre igual)
  - No enviar correo.
  - O enviarlo en formato de texto plano
  - No usar redirecciones para abreviar URLs (`http://redir.ejemplo.es/Xlji`)
  - Firmar digitalmente los mensajes
  - No enviar mensajes si se bloquean las cuentas o hay problemas. Mejor proporcionar una dirección de contacto, o contactar directamente.

Phishing

- No preguntar secretos. Nunca.
- No usar pop-ups
- No usar 'frames' o 'iframes'
  <**A HREF**="http://www.ejemplo.es/" TARGET="_top">
  (abre una nueva página en la misma ventana)
  ¿Mirar DOM?

Phishing

- Separar la aplicación de la página frontal
  - Autentificación en una página separada
  - Comprobar el referrer.
  - Que los usuarios tecleen
- Comprobar los 'referrers' para las imágenes y otros recursos (¿Ponerles 'marcas de agua'? ¿Comprobar descargas de imágenes?)
- No esconder la barra de direcciones, usar SSL, no usar IPs
- No mostrar datos personales

Phishing

- Desactivar cuentas no usadas
- Consistencia de los datos
- Límites diarios
- Operaciones retrasadas (para poder repudiarlas)
- Entregar bienes a direcciones verificadas y registradas

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Phishing

- Si se permite actualizar datos, notificar al viejo y al nuevo
- No enviar claves. Enviar verificadores de un sólo uso y válidos por tiempo limitado.
- Enviar avisos de la actividad
- Limitar la actividad en periodos de tiempo (ataques automáticos)
- Autentificación de dos factores

Phishing

- Controlar actividades poco habituales
  - Borrar cuentas (o vaciarlas)
  - Muchas transacciones pequeñas
  - Envíos de varias cuentas a la misma dirección
  - Transacción repetida desde la misma IP
- Actuar contra los 'malos' con rapidez: policía, reguladores, ISPs, ...

Phishing

- Tratar de hacerse con los dominios fraudulentos
  `http://www.ejjemplo.es/`
- Colaborar con la ley
- Y cuando pase . . .
  - Tratar bien a los usuarios (son víctimas)
  - Tener una política de actuación

`http://www.antiphishing.org/`

Servicios web

- Los mensajes SOAP deberían enviarse de forma confidencial y sin modificaciones
- El servidor debería conocer con quién habla, y qué pueden hacer los clientes
- Los clientes tiene que estar seguros de que hablan con el servidor correcto
- Registro, auditoría, trazabilidad, ...

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Servicios web.

- Seguridad de las comunicaciones
  - Sólo proporciona seguridad punto a punto (Comunicación con varios saltos)
  - Almacenamiento
  - Falta de interoperabilidad
- Transmisión de credenciales
  - XML, traducción a texto
  - Más puntos de divulgación

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Servicios web.

Hay que tener en cuenta ..

- Actualidad de los mensajes ('replay')
- Integridad de los mensajes
- Confidencialidad de los mensajes
- Control de acceso (identificación, autentificación, autorización)
- Auditoría

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Servicios web. WS-Security Standard

Incluye:

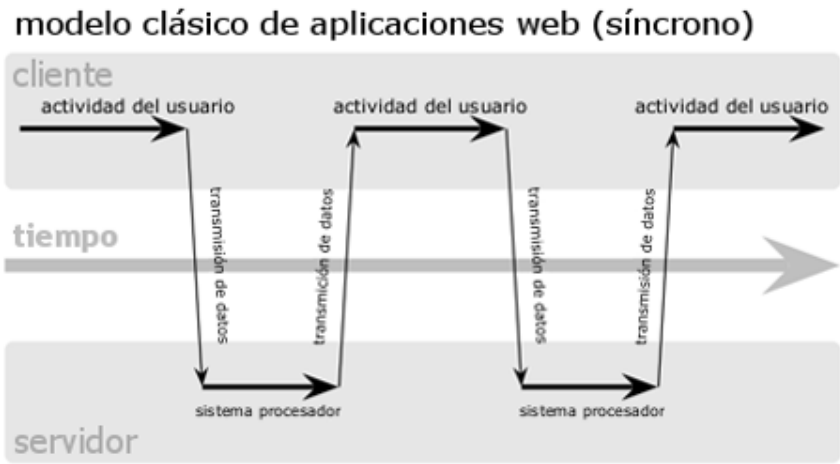
- Formas de añadir cabeceras de seguridad a 'Envelopes' SOAP
- Adjuntar objetos de seguridad y credenciales al mensaje
- Añadir un 'timestamp'
- Firmar el mensaje
- Cifrar el mensaje
- Extensibilidad

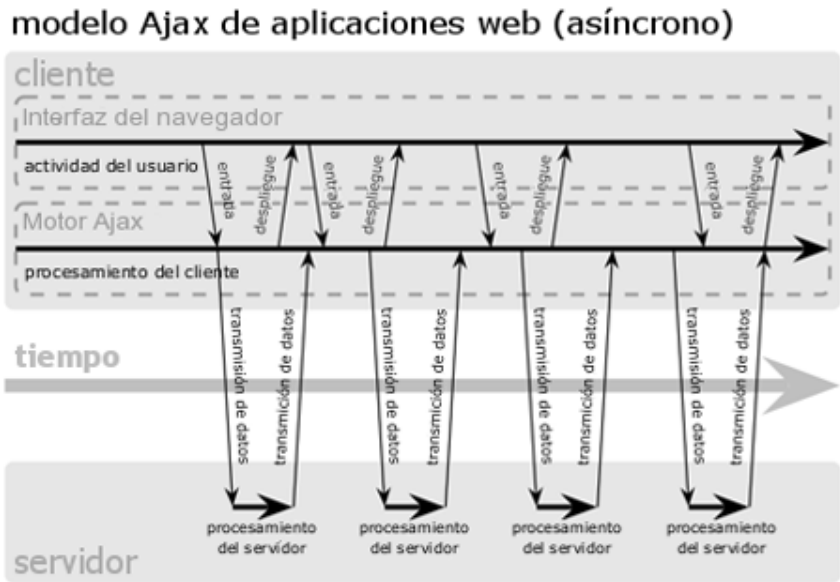
Servicios web. WS-Security Standard

Problemas:

- Falta de madurez
- Prestaciones
- Complejidad e interoperabilidad
- Gestión de claves

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

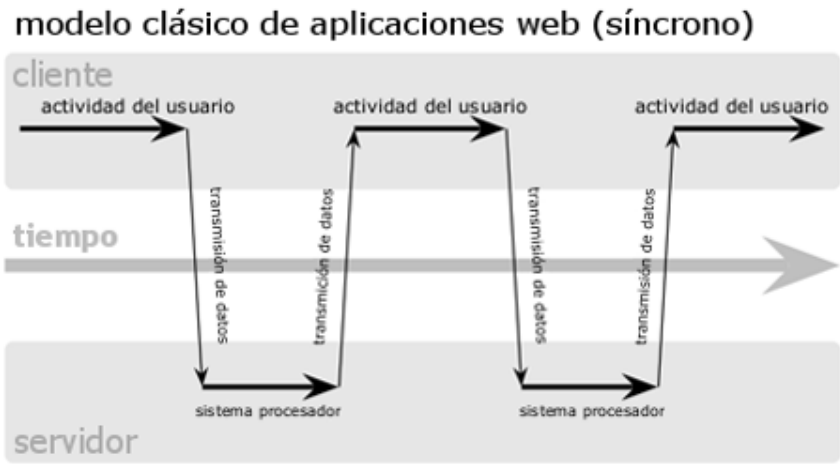


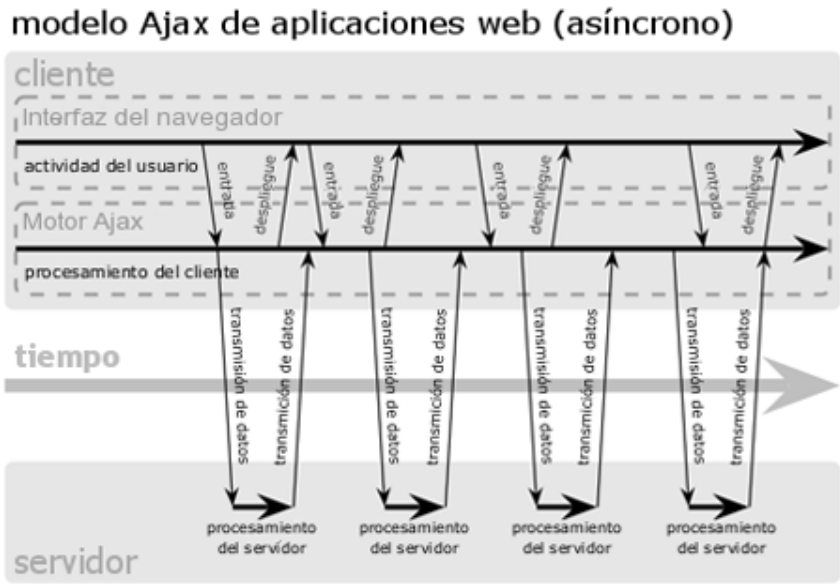


Ajax

Asynchronous JavaScript + XML

- Permite que el navegador haga consultas al servidor sin recargar la página.
- Puede haber interacción con el servidor sin que el usuario lo note.





Ajax

Asynchronous JavaScript + XML

- Permite que el navegador haga consultas al servidor sin recargar la página.
- Puede haber interacción con el servidor sin que el usuario lo note.

Tener en cuenta . . .

- Comunicaciones seguras
- Autentificación y manejo de sesiones
- Control de acceso
- Validación de entradas
- Gestión de errores y registro

Ajax

- Cada función que pueda ser llamada con Ajax debería verificar la sesión y la autorización

```
<?php
function calculate_tax($sales_amount)
{
return($sales_amount * 0.075);?
}
?>
```

versus

```
<?php

function calculate_tax($sales_amount)
{
        // check that the session is logged in ?
        assert_login();

        // check that the user has the USER role to prevent
        // guest and admin access
        assert_role('USER');

        // Validate data and business rules
        if ( is_numeric($sales_amount) && $sales_amount > 0 )
        {
                // Perform the calculation and return
                return($sales_amount * 0.075);?
        }
        // Data failed validation and business rules
        return -1;
}
?>
```

Ajax

¡Me suena!

- Autentificación
- Autorización y separación de usuarios
- Validación de datos
- Validación reglas de negocio

- Problemas de XMLHttpRequest
  - Peticiones y respuestas: HTML, XML, JSON (Javascript Object Notation)
  - En claro
  - Inyecciones variadas