Incorporando seguridad a las componentes de Interfaz de Usuario del Framework JSF (JAVA Server Faces) con soporte para clientes heterogéneos.

Pablo José Iuliano

Facultad de Informática, UNLP, La Plata,
Provincia de Buenos Aires, Argentina.

Directores: Francisco Javier Díaz
Claudia Queiruga

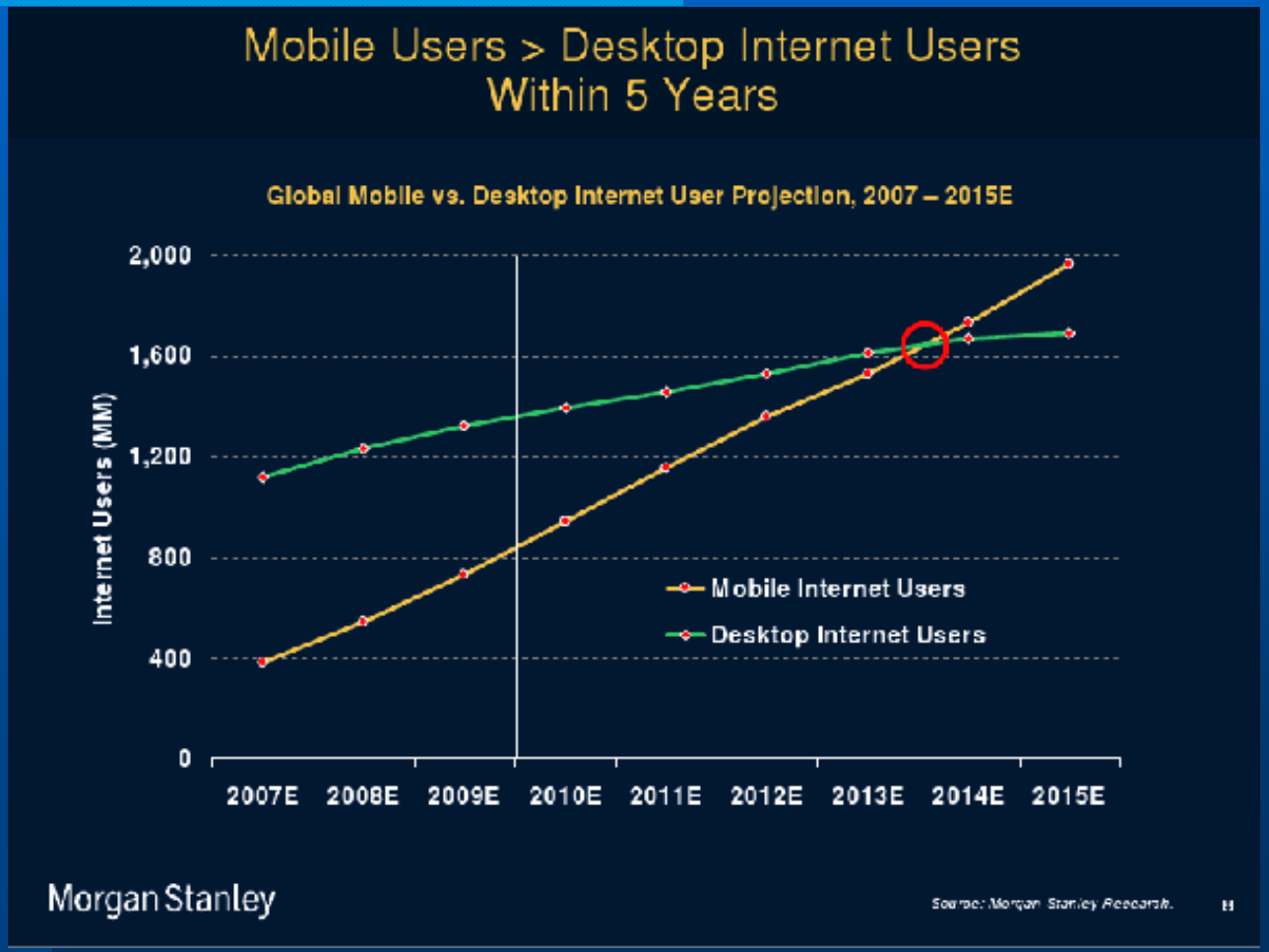
Eras Tecnológicas

- Era de los *mainframes*
  - Años 50s y 60s
- Era de las mini-computadoras:
  - Años 70s
- Era de las PCs:
  - Años 80s y 90s.

Actualidad Tecnológica

- La predilección de los usuarios está cambiando del *desktop* con acceso a la Web al Web *mobile.*
- Se predice que en los próximos 5 años habrá más usuarios conectados a Internet a través de dispositivos móviles que con PCs tradicionales.

Proyección de Usuarios de Internet *Mobile* vs. los *Desktop*



Aplicaciones Móviles

- **Disponibilidad a marzo del 2010:**
  - 150.000 aplicaciones móviles disponibles en *iTunes.*
  - *20.000 aplicaciones para Android.*
  - 5.000 para *Blackberry.*
- **Proyecciones de descargas:**
  - 6.6 millones de *downloads* en el 2010.
  - 16.2 millones de *downloads* para el 2013.

Preferencias de los Usuarios de Internet *Mobile*

- Los *smartphones* se están convertido en el nuevo estándar para conectarse a la Web.

- Porcentaje de subscriptos a servicios de Internet *mobile* a través de *smartphones* en los Estados Unidos*:*
  - En el 2008 era del 14 %.
  - En el 2009 escalo al 21 %.

Seguridad en las Aplicaciones Móviles

- Actualmente es un requerimiento crítico.
- Los riesgos de seguridad se incrementan debido a la gran disponibilidad de aplicaciones en los Marketplaces.
- Las aplicaciones manejan información personal del usuario (Por ejemplo: cuentas bancarias, datos de la tarjeta de crédito, itinerarios de vuelo, etc.).

Seguridad en JAVA

JAVA ha tenido presentes los problemas de seguridad y ha definido un modelo para controlar y limitar el acceso a los recursos desde las aplicaciones.

- Modelo de arenero (*Sandbox Model*).
- Arquitectura Criptográfica de JAVA (**JCA**).
- Extensión Criptográfica de JAVA (**JCE**).

Fundamentos de JAVA Server Faces

- **JAVA Server Faces** (JSF) comúnmente llamado **Faces**, es un framework JAVA estándar que facilita la construcción de interfaces de usuario server-side.
- Fue desarrollado por el Java Community Process, (JSR 252 JSF 1.2, JSR 314 JSF 2.0) y forma parte de la especificación de Java Enterprise Edition (JEE) versión 5.0.
- La idea básica de JSF es escribir aplicaciones Web al estilo Rapid Application Development (RAD), de la misma manera que se escriben aplicaciones de escritorio, tal como se haría con Microsoft Visual Basic, PowerBuilder o Borland Delphi.

Fundamentos de JAVA Server Faces (Cont.)

- **Faces** provee un conjunto de componentes de interfaz de usuario predefinidos, listas para usar.
- **Faces** es extensible, permite programar librerías de componentes de interfaz de usuario propias e incorporar extensiones a las librerías ya existentes, mediante los modelos UI (*User Interface*), *renderers*, validadores y conversores.
- **Faces**, **NO** provee mecanismos nativos para dotar de seguridad a las aplicaciones.

Identificación de Vulnerabilidades en Aplicaciones Web

- **OWASP** es una comunidad abierta dedicada a brindar asesoramiento de seguridad a organizaciones. (http://www.owasp.org).
- **OWASP Top Ten** consensúa cuáles son las fallas de seguridad más críticas en aplicaciones Web.
- Las vulnerabilidades que están dentro del ámbito de interés de esta tesis:

  A1 – Secuencia de Comandos en Sitios Cruzados (XSS).

  A2 – Fallas de Inyección.

  A5 – Vulnerabilidades de Falsificación de Petición en Sitios Cruzados (CSRF).

Métricas de Seguridad Aplicadas los Frameworks Web JAVA y .NET

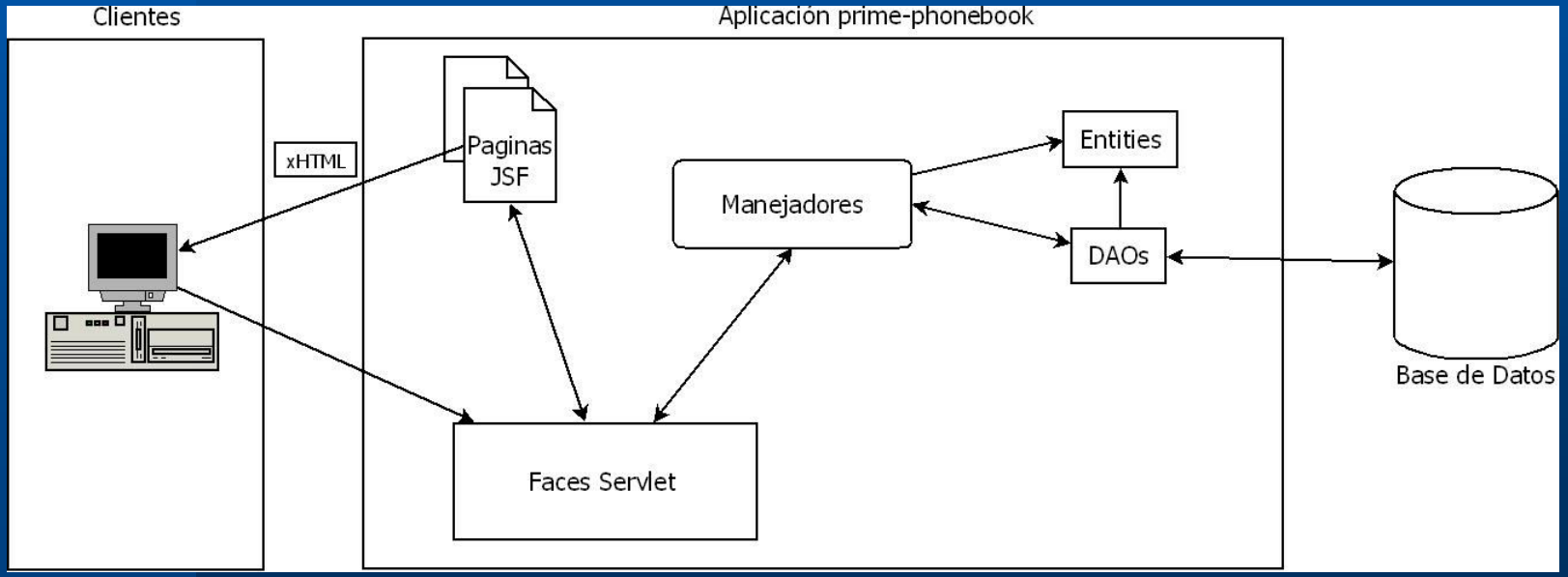
Web Framework	Binding	Integrity	Confidentiality	Generic Validations	Escape Characters	Random Tokens	Monitorization
Struts	✗	✗	✗	✗	✓	✗	✗
Struts²	✓	✗	✗	✗	✓	✗	✗
Spring	✓	✗	✗	✗	✓	✗	✗
webwork	✓	✗	✗	✗	✓	✗	✗
Stripes	✓	✗	✗	✗	✓	✗	✗
JavaServer™ Faces Sun	✓	✗ Partially Secure	✗	✗	✓	✗	✗
MyFaces	✓	✗ Partially Secure	✗	✗	✓	✗	✗
WICKET	✓	✗ Partially Secure	✗	✗	✓	✗	✗
Microsoft .net	✓	✗ Partially Secure	✗	✗	✓	✓	✗

Distribución PrimeFaces

- Librería *open source* para *JavaServer Faces* disponible en http://primefaces.googlecode.com/.
- Se compone de tres módulos:
  - El conjunto de componentes para la interfaz de usuario.
  - Módulo *Optimus*.
  - Módulo *FacesTrace*.
- **Mobile TouchFaces:** el módulo Web *Mobile* de *PrimeFaces*
  - Contiene un conjunto de componentes para crear aplicaciones que puedan visualizarse en clientes móviles.
  - *Suite* de componentes ligera.

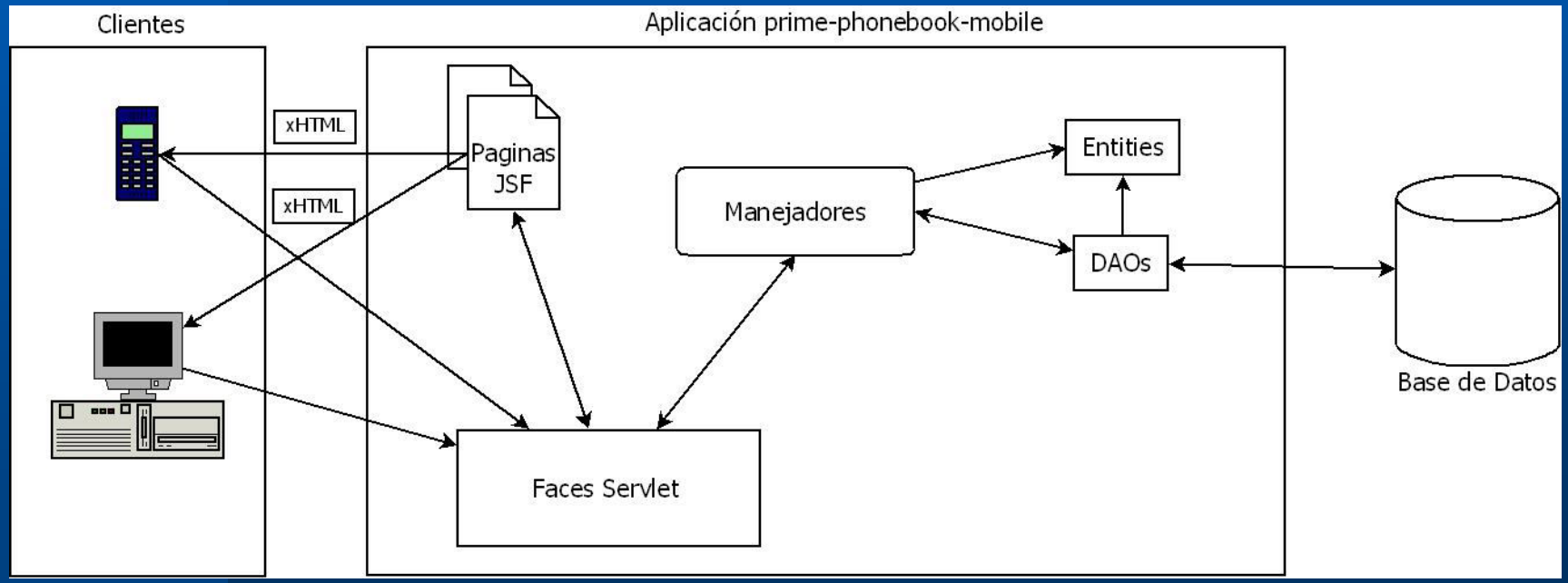
Pruebas de Vulnerabilidades

prime-phonebook: publicada en el sitio oficial de la distribución *PrimeFaces* de **JSF** (http://primefaces.googlecode.com/svn/examples/trunk/prime-phonebook/).



prime-phonebook-mobile: prime-phonebook rediseñada con Mobile TouchFaces

Modificación del código de **prime-phonebook** para que pudiera funcionar tanto en plataformas Web tradicionales como móviles.

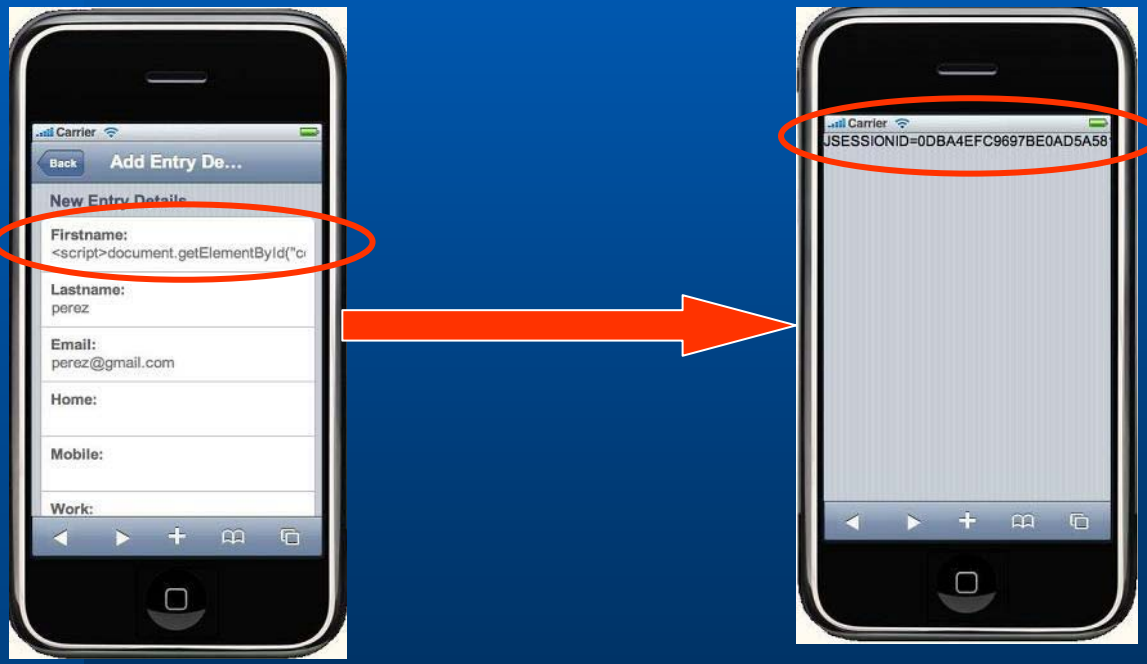


Secuencia de Comandos en Sitios Cruzados (XSS)

- Víctima: usuarios finales de una aplicación.
- Consiste en la inserción de código malicioso (scripts) en páginas Web que serán vistas por otros usuarios.
- Lenguajes: HTML, JavaScript, VBScript, ActiveX, Shockwave, Flash, etc.
- Clasificación:
  - **Reflejado**: el código malicioso es ejecutado inmediatamente en la página de respuesta al usuario.
  - **Almacenado**: el código malicioso es almacenado en un servidor antes de ser enviado al usuario.

Ataque XSS contra prime-phonebook-mobile.

- Código JavaScript inyectado:
  <script>document.getElementById("content").innerHTML = document.cookie</script>



Inyección del Javascript malicioso

Ataque exitoso, resultando en la exposición del SessionId

Fallas de Inyección

- Víctima: aplicación Web.
- Se transmite código malicioso a través de una página Web hacia otros sistemas (DBMS, SO).
- **Inyección SQL**: se insertan sentencias SQL que son enviados al motor de base de datos para su ejecución.
- Clasificación:
  - In band: la información es retornada directamente por la aplicación Web atacada.
  - Out of band: la información es retornada por otro canal (Ej.: e-mail).

Ataques de Fallas de Inyección contra prime-phonebook-mobile.

Pantalla inicial de la aplicación

Inyección del SQL malicioso

Ejecución del ataque

Ataque exitoso, resultando todos los contactos eliminados

Falsificación de Petición en Sitios Cruzados (CSRF)

- Víctima: aplicación Web.
- Fuerza al navegador de una víctima a enviar una petición a una aplicación Web vulnerable, la cual realiza la acción elegida a través de la víctima.
- Cualquier aplicación que no dispone de pruebas de autorización para acciones vulnerables, procesará una acción si se pueden pasar credenciales predefinidas en la petición.
- Todos los entornos de aplicación Web son vulnerables a **CSRF**

Ataques de CSRF contra prime-phonebook-mobile.

Pantalla inicial de la aplicación

Aplicación maliciosa que efectuará el ataque

Ejecución del ataque

Ataque exitoso, resultando en la creación de un nuevo contacto no deseado

Falta de Confidencialidad de los Datos

- Víctima: aplicación Web.
- Con solo inspeccionar el HTML que retorno el servidor, se tendrá acceso a ***información sensible*** *o* ***privilegiada*** que no debería ser publica.
- Cualquier aplicación que no disponga de mecanismos de ocultamiento de información sensible, expondrá la misma en el cliente.
- Todos los entornos de aplicación Web son vulnerables la ***exposición de información sensible*** *o* ***privilegiada***.

Explotación de Falta de Confidencialidad de los Datos en prime-phonebook-mobile.

Página retornada por la aplicación

```
<input id="firstname" type="text" name="firstname" />
<input id="lastname" type="text" name="lastname" />
<input id="email" type="text" name="email" />
<input id="home" type="text" name="home" />
<input id="mobile" type="text" name="mobile" />
<input id="work" type="text" name="work" />
<input id="fax" type="text" name="fax" />
```

Inspección del código del la página retornada

Tabla PhonebookEntry

id: long
firstname: varchar
lastname: varchar
email: varchar
home: varchar
mobile: varchar
work: varchar
fax: varchar

Deducción de parte de la estructura de la base de datos

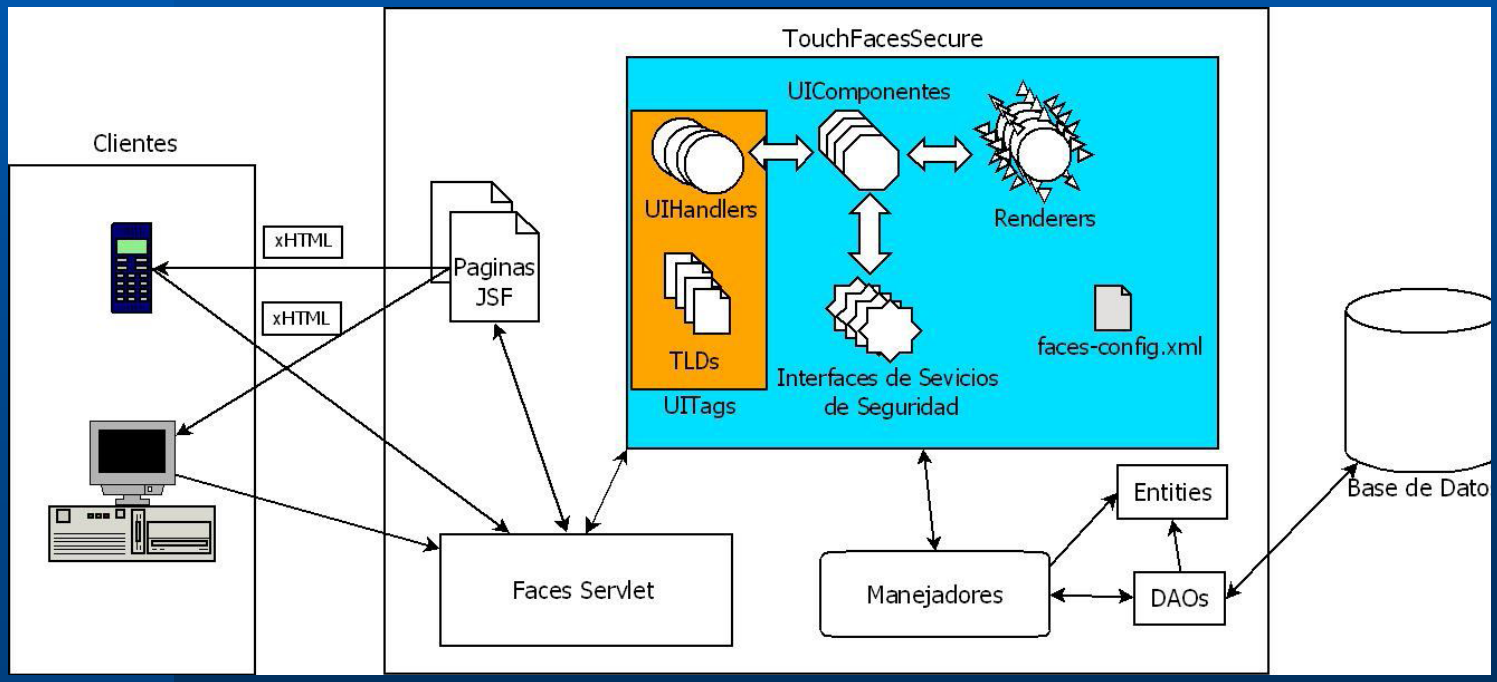
OWASP y JAVA Server Faces

Consideraciones de **OWASP** sobre Faces:

- Fortalezas:
  - Validadores.
- Debilidades:
  - Almacenando estado en el lado del cliente.
  - Convertidores.
  - Acceso estático al **FacesContext.**

TouchFacesSecure: una extensión de TouchFaces con soporte para seguridad

Arquitectura de **TouchFacesSecure**: aplicación *Faces* construida a partir de **TouchFacesSecure**



TouchFacesSecure: Librerías de Soporte

- **Bouncy Castle** (http://www.bouncycastle.org/)
- **ValidatingHttpRequest de OWASP** (http://www.owasp.org/index.php/How_to_add_validation_logic_to_HttpServletRequest)

TouchFacesSecure: Componentes Seguros

- **Application Seguro**
  - Incorpora un mecanismo que previene que *scripts* en el cliente accedan a la *cookie*.
  - Utiliza el *flag HttpOnly* en *header* en el *header* de la respuesta **HTTP**.

TouchFacesSecure: Componentes Seguros

- **InputText Seguro**
  - La información crítica que especificó el desarrollador se cifra.
    - El cifrado de la información se resuelva con clases que implementan **tesis.tool.seguridad.encryptionToolkit.Encryptor**.
  - La segunda medida de seguridad implementada busca evitar la concreción de ataques **XSS** y de inyección de **SQL**.

TouchFacesSecure: Componentes Seguros

- **Label Seguro**
  - Usar una clase tipo **Encryptor** para encriptar la información crítica.

TouchFacesSecure: Componentes Seguros

- **CommandLink y CommandButton Seguros**
  - Se genera un *token* aleatorio el cual es almacenado en el cliente para su posterior verificación.
    - La generación del *token* aleatorio al objeto del componente es realizada por una instancia que implementa la interfase **tesis.tool.seguridad.CSRFToolkit.CSRFTokenManageable.**

TouchFacesSecure: Componentes Auxiliares

Desempeñan funciones auxiliares para que los componentes seguros puedan funcionar correctamente, pero en sí no contribuyen con ningún mecanismo de seguridad extra.

- **NavBarControl**.
- **View**.

TouchFacesSecure: Deployment y Modo de utilización

- Para la construcción de **touchFacesSecure-1.0.0.jar** se utilizó *Apache Maven.*
  - **Dependencias**:
    - *Bouncy Castle*
    - *Space4j*.

- **Para el uso:**
  - **Enfoque 1: Fácil Migración**
    - Aplicación Resultante: **prime-phonebook-mobile-segura-facil-migracion**.
  - **Enfoque 2: Funcionalidades Personalizables Seguras**
    - Aplicación Resultante: **prime-phonebookmobile-segura-funcional-personalizable**.

Ataque XSS contra prime-phonebook-mobile-segura-facil-migracion

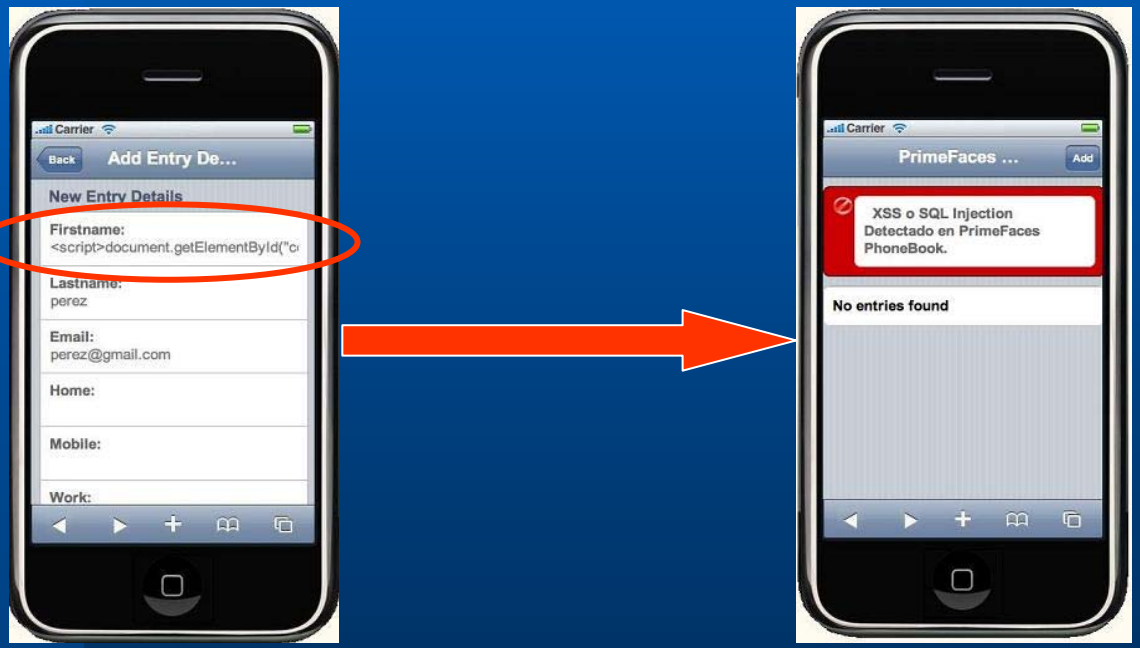
- Código JavaScript inyectado:
  <script>document.getElementById("content").innerHTML = document.cookie</script>

Inyección del Javascript malicioso

Ataque frustrado

Ataque XSS contra prime-phonebook-mobile-segura-funcional-personalizable

- Código JavaScript inyectado:
  <script>document.getElementById("content").innerHTML = document.cookie</script>



Inyección del Javascript malicioso

Ataque frustrado, notificando al usuario del intento de ataque

Ataque Fallas de Inyección contra prime-phonebook-mobile-segura-facil-migracion

Pantalla inicial de la Aplicación

Inyección del SQL malicioso

Ejecución del ataque

Ataque frustrado

Ataque Fallas de Inyección contra prime-phonebook-mobile-segura-funcional-personalizable

Pantalla inicial de la Aplicación

Inyección del SQL malicioso

Ejecución del ataque

Ataque frustrado, notificando al usuario del intento de ataque

Ataque CRSF contra prime-phonebook-mobile-segura-facil-migracion

Pantalla inicial de la Aplicación

Aplicación maliciosa que efectuará el ataque

Ejecución del ataque

Ataque frustrado

Ataque CSRF contra prime-phonebook-mobile-segura-funcional-personalizable

Pantalla inicial de la Aplicación

Aplicación maliciosa que efectuará el ataque

Ejecució n del ataque

Ataque frustrado, notificando al usuario del intento de ataque

Explotación de Falta de Confidencialidad de los Datos en ambas Aplicaciones

Página retornada por la aplicación

Inspección del código de la página retornada

Imposibilidad de deducir la de estructura de la BD.

Tabla PhonebookEntry
????
????
????
????

```
<input
id="ZmWvXLA1fm+8zt3h9IGgSHnjFNasFIQCYxFlqTZaoTFEo3XptSxEdp0xlrgLrBVt
9BBbPJT2jAB/2lVOTn0N4HRW+BRdG3sTtBw8VLQlWdVtNV/Yx0Yd7yCT7Q3+Woq87hYC
XwAlbJZ98w4gY8GlvnxniZ0Gxhi09jnLhZIkJ+nEEgtH7rSBmu1HzZM0wflHOPDMBmae
Z5r1jZl1BTXkf3vOO/EM3HrHRZcZeTE2dADe25Inep6DqnATa5VYZB2XPrLwvtaR2UHx
Clq8lCP7xVM4qV8nMvt1P8IpU5rTCeHdafId/u5m9lLFkGQZ94jFY6Nb300LflnNnRd5
S9fs9A==" type="text"
name="Px9qbnvoY/IpkMIadiVPeRIxiNyiZMA93npNnMmf32sD32FxiqSZZhiGAlDfa1
ImyH24HO7sBWTAia19+1AUUsUTOlmEpQgAe5KXPMlc8w8s9h04cQOEChfu5oshaiR/PO
BC1Q7ata7MakqYkfTOGBe/GIaBX/I+0nDK+WZf5TPiiumBmWpk865ys0IMAkd9ng34Mu
2wsk+I091ju8DQqdhLerFYIdTjOm/ePLK0RyX0hs7Hf+CJQXyZFPLmtgZ6DoNdYp09Rg
s3UV3mMcXvsyuS61Gb3SWh117/dKbFOtKLmAOXxY3aV/apWeyQot7zYTANuJpm6yYsVM
cd1FQtLQ==" />
<input
id="p8XEixGk/3+AB9Kccn/VN4ha5+8qEaibt6MGuTZpwmY3LDpZY3R45mgPfG/yauqa
j78PsJB1F40UW/Q7kcLpU6W3sOB/OeVVRKGQorDE1+w/FCuu80JClVCLggE5uh0taoSQ
pRDt/w+sAaL31sBsg605AEtV48UPGpdnuAtYToddRFE7g3LkYPTzjb21PIQjXoJVSAGN
zCK0yx8x+fnVnT9LDsARDz2Kmws3hALZtrKE3Tg9cnk5M8NBm25jtBqIJfLqPpP44ds5
kPpx0Wbz5lpwFzMnSBG9TfKiQvEj7Ao965r0FDk3AV2ewwyuq9QT3uS7P1es9AqQVgm4
jWsTA==" />
type="text"
name="GHZuTiUnsN95pDix7vkKTgwWrJ6QRx0q4zuC2HrAFiRccu3aebBcNRp3iqTIFS
rzHrygsR55M75ywh6ZrRlsyxZ8IDJJYjENBTQQg3t0m/aoKYcyALVSDDxL2gZzd59kq2
40fP1V8uX+KDpzob0Hd+vuqFlz4HwCPMwa0K7CF4a0qFuh8X7YOkgrTMScKiw7PZCYjQ
/3XagUEDLGdwwTRAwlF6WIkNRSz7TmcLsPPVmHKGqwAKT8iU+86U+4vvHyM87dtfZHAc
/N73m6RF+VpJxN24hMbONXnAm6nIyqEpCAPs5YaJFOLPyKpU3Ox6kulUI9Xu2gX64v05
o5q81AQQ==" />
```

• • •

Conclusiones

TouchFacesSecure…

- Sanea o mitiga las vulnerabilidades Web.
- Fácilmente integrable, adaptable, distribuible, reutilizable, centralizada y extensible.
- Simple para usar.
- Logra migraciones con el mínimo costo.
- Flexible en la seleccion de los mecanismos de seguridad a utilizar.
- Simplifica y reduce las etapas de construcción y de *testing* de aplicaciones **JSF**.
- Contribuye con material educativo.

Trabajos Futuros

- Extender la librería para tratar nuevas vulnerabilidades.
- Implementar/Incorporar componentes adicionales a la libreria.
- Analizar la eficiencia del uso de la librería en tiempo de carga/ejecución.
- Contribuir con la comunidad de Software Libre de JSF.

Muchas Gracias!