



**OWASP**

The Open Web Application Security Project

# LOS 7 PECADOS DEL DESARROLLO WEB & TENTACIÓN DEL USUARIO EN APLICACIONES MOVILES

[www.owasp.org](http://www.owasp.org)



# OWASP

The Open Web Application Security Project

## ¿Quien soy?

### Ing. Elvin Vidal Mollinedo Mencia

“Profesional de seguridad

+ 9 años de experiencia en desarrollo seguro e infraestructura de telecomunicaciones, manejando de normativas CISA(COBIT), SOX(**Ley Sarbanes Oxley**), pentesting web, pentesting WI FI y electronica (hardware libre)”

Lider Capitulo Owasp Bolivia

Miembro activo de la comunidad OWASP

[Elvin.mollinedo@owasp.org](mailto:Elvin.mollinedo@owasp.org)

**it.rational**  
security systems  
[www.it-rational.com](http://www.it-rational.com)



# OWASP

The Open Web Application Security Project

## Pecado I: Inyección

Una inyección ocurre cuando los datos no validados (ni confiables) son enviados a un interprete como parte de un comando o consulta.

Los datos hostiles del atacante pueden engañar al interprete, ejecutar comandos, o acceder datos no autorizados

El atacante puede:

- Descargar BD completas
- Añadir o eliminar información a la BD
- Ejecutar comandos de SO
- Comprometer la BD o el servidor





# OWASP

The Open Web Application Security Project

## Solicitud numérica (cliente)

`http://www.foo.bar/ver_productos.XXX?id= 1 or 1=1`

## Aplicación (servidor)

```
select * from productos where prod_id = X
```

## Solicitud alfanumérica (cliente)

`http://www.foo.bar/ver_productos.XXX?nombre= nada' or 'a'='a`

## Aplicación (servidor)

```
select * from productos where prod_nombre = ' X'
```



# OWASP

The Open Web Application Security Project

## Inyecciones

### INICIO DE SESION

Usuario:

Password:

### Código que genera el Query

```
sql = "SELECT * FROM usuarios  
WHERE usuario = " + usuario +  
" and clave = " + clave + "
```

### Así se interpreta el query por el motor de BD

```
SELECT * FROM usuarios WHERE  
usuario = 'admin' --' and clave  
= 'no importa'
```

Si alguien se pregunta si esto tan grave puede ser real pues...

Este es un ejemplo de la vida real de un sitio que recibía pagos por tarjeta de crédito.



# OWASP

The Open Web Application Security Project

## Inyección

### INICIO DE SESION

Usuario:

Password:

```
'UNION/**/SELECT/**/  
CAST(usuario/**/as/**/int),  
Version(),1,1,1/**/  
FROM/**/usuarios/**/  
WHERE/**/usuario/**/ > /**/ 'a  
'--
```

Este es un ejemplo de la vida real de un sitio que recibía pagos por tarjeta de crédito.

### Código que genera el Query

```
sql = "SELECT * FROM usuarios  
WHERE usuario = " + usuario +  
" and clave = " + clave + "
```

### Así se interpreta el query por el motor de BD

```
SELECT * FROM usuarios WHERE  
usuario = ' UNION SELECT  
CAST(usuario as int),version(),1,1,1  
FROM usuarios  
WHERE usuario > 'a' --' and  
clave = 'no importa'
```



# OWASP

The Open Web Application Security Project

## OS Command Injection (OSi)

**Ejecución de comandos con información brindada por el “usuario”**

```
https://misitioinseguro.com/generarpdf.php?nombre=miinfo;  
cat /etc/passwd > /var/www/passwd
```

**Así se interpreta el comando al ejecutarse**

```
generarpdf reporte.html miinfo;  
cat /etc/passwd > /var/www/passwd.pdf
```

El comando generarpdf genera un archivo PDF a partir de una platilla html

Adicionalmente con el comando cat genera el archivo passwd.pdf en la raíz del sitio a partir del archivo passwd (Contraseñas)

**Por fortuna para el atacante el archivo passwd.pdf puede ser descargado desde:**

```
https://misitioinseguro.com/passwd.pdf
```



# OWASP

The Open Web Application Security Project

# Prevención y remediación

## 1. SQLi

- Consultas parametrizadas
- Procedimientos almacenados
- Validación de entradas `"* ' , ; : & % " -( ) / < > + \"`

## 2. OS Command Injection

- Listas blancas
- Validación de Entradas

## 3. Consejos comunes para hardening

- No utilizar el SA en su aplicación
- Cuenta de sistema operativo con permisos limitados

**Todas las validaciones se deben realizar  
en ambos lados: cliente y servidor**



# OWASP

The Open Web Application Security Project

## Pecado II: Fallo de autenticación

Ocurre cuando usuarios anónimos intentan acceder a cuentas de usuarios válidos del sistema, cuando usuarios validos intentan obtener mayores privilegios de los reales o intentan “disfrazar” o eliminar acciones dentro del sistema



# OWASP

The Open Web Application Security Project

## Sesiones

1. Para mantener el estado de la sesión se provee un identificador (Session\_ID o Token) que es compartido entre el usuario y la aplicación web

2. Los atacantes pueden ejecutar dos tipos de Session Hijacking (secuestro de sesión):

**Orientados:** el atacante impersonaliza a un usuario específico y privilegiado

**Genérico:** impersonaliza a un usuario genérico del cual desconoce sus accesos

### Nombre de la Sesión de las aplicaciones

PHP --> PHPSESSID

J2EE --> JSESSIONID

ColdFusion --> CFID y CFTOKEN

ASP --> ASPSESSIONID

ASP.NET --> ASP.NET\_SessionId



# OWASP

The Open Web Application Security Project

## Sesiones

La sesión no debe tener información sensible (o debe estar cifrada) y debe ser almacenada en el servidor. Podría (no deseable) contener dirección IP, User-Agent, e-mail, nombre de usuario, rol, privilegio, preferencias del usuario, último acceso, timeouts, etc.

### Credenciales y Sesiones

No enviar información sensible en el URL

`http://example.com/accion;`

`jsessionId=2P0OC2JDPXM0OQSNDLPSKHCJUN2JV?parametro=valor`

No almacenar información sensible en Cookies

Cookie: User=Juanito

Cookie: Password=123



# OWASP

The Open Web Application Security Project

## Sesiones



2

Ingeniería Social  
[http://seguro.com/login.xxx?  
SessionID=1234](http://seguro.com/login.xxx?SessionID=1234)



**GET cuenta.XXX?SessionID=1234**  
**USUARIO Y CLAVE**

1

**LOGIN** 3  
**SessionID=1234**  
**GET cuenta.XXX?SessionID=1234**



SERVIDOR



# OWASP

The Open Web Application Security Project

## Como evitar la Perdida de Autenticación y Gestión de Sesiones

- **Verificar la arquitectura**

Autenticación debería ser simple, centralizada y estandarizada

Utilizar el gestor de sesiones estándar provisto por el servidor de aplicaciones – no inventar uno propio!

Estar seguro que SSL protege tanto las credenciales como las sesiones de usuario todo el tiempo

- **Verificar la implementación**

No utilizar solamente análisis automático

Verificar el certificado SSL

Examinar todas las funciones relacionadas a autenticación

Verificar que “cierre de sesión” efectivamente destruya la sesión

Utilizar OWASP's WebScarab para testear la implementación



# OWASP

The Open Web Application Security Project

## Pecado III

# XSS (Cross Site Scripting)

- Cross Site Scripting (XSS): ejecución de scripts y comandos no deseados a través de aplicaciones web, explotando la confianza del usuario
- Se originan por la validación incorrecta de variables que permiten ejecutar scripts en campos de entrada
- El atacante inyecta código malicioso (HTML y scripts) que son ejecutados en el entorno del navegador del cliente afectado



# OWASP

The Open Web Application Security Project

## XSS (Cross Site Scripting)

- Mediante el control del navegador del usuario es posible realizar ataques de:
  - Robo de sesión e identidad, mediante la
  - manipulación de las cookies
  - Phishing, mediante la modificación de la interface normal del sitio
  - Redirección a sitios dañinos
- Existen tres tipo de XSS:
  - Reflejados/No almacenados
  - Almacenados
  - DOM-XSS



# OWASP

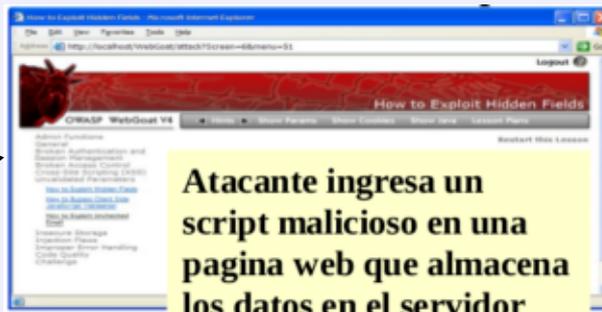
The Open Web Application Security Project

# XSS

1

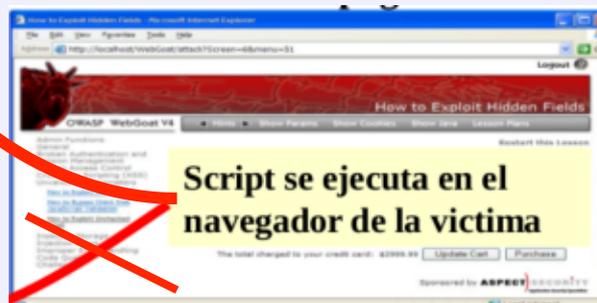
Atacante establece una trampa - actualiza perfil

Aplicación con vulnerabilidad



2

Victima visualiza la página - accede al perfil



3

Script silenciosamente envia la sesion de la victima al atacante



# OWASP

The Open Web Application Security Project

# Como evitar Fallas de XSS

No incluir entradas suministradas por el usuario en la página de salida

Recomendación Principal: Codificar todos los datos de entrada en la página de salida (Utilizar OWASP' s ESAPI para dicha tarea):

**<http://www.owasp.org/index.php/ESAPI>**

Siempre efectuar una validación 'positiva' de todas las entradas realizadas por el usuario

Definir políticas de Content Security Policy (W3C) (HTML5)



# OWASP

The Open Web Application Security Project

## Pecado IV

# Referencia directa a objetos

Una referencia directa a objetos ocurre cuando un desarrollador expone una referencia a un objeto de implementación interno, tal como un fichero, directorio, o base de datos. Sin un chequeo de control de acceso u otra protección, los atacantes pueden manipular estas referencias para acceder datos no autorizados.



# OWASP

The Open Web Application Security Project

## Referencia directa a objetos

Income and Expenses from Sep 26, 2004 to Jan 16, 2005

Date	Description	Category	Amount
Nov 22, 2004	Interest Payment	Interest	-\$25
Nov 22, 2004	ATM Withdrawal, myBank, San Rafael, CA	Cash	\$100.00
Nov 19, 2004	ATM Withdrawal, myBank, San Francisco, CA	Cash	\$100.00
Nov 16, 2004	SBC Phone Bill Payment	Phone	\$94.23
Nov 16, 2004	myBank Credit Card Bill Payment	Credit Card	\$2,853.57
Nov 15, 2004	ATM Withdrawal, myBank, San Rafael, CA	Cash	\$100.00
Nov 15, 2004	myBank Payroll	Payroll	\$4,373.79
Nov 10, 2004	ATM Withdrawal, myBank, San Francisco, CA	Cash	\$100.00
Nov 4, 2004	ATM Withdrawal, myBank, San Francisco, CA	Cash	\$100.00
Nov 3, 2004	myBank Credit Card Bill Payment	Credit Card	\$10.00
Nov 1, 2004	Working Assets Bill Payment	Phone	\$13.57
Nov 1, 2004	Prudential Insurance Bill Payment	Insurance	\$435.00
Nov 1, 2004	Chase Manhattan Mortgage Corp Bill Payment	Mortgage	\$2,184.42
Oct 29, 2004	ATM Withdrawal, myBank, San Francisco, CA	Cash	\$100.00
Nov 16, 2004	myBank Payroll	Payroll	\$4,338.96

Net Cash Flow: \$435.29

Atacante identifica su número de cuenta 6065  
?acct=6065

Lo modifica a un número parecido  
?acct=6066

Atacante visualiza los datos de la cuenta de la víctima



# OWASP

The Open Web Application Security Project

## Como evitar Referencias Directas Inseguras a Objetos

Eliminar la referencia directa a objetos

Reemplazarla con un valor temporal de mapeo (ej. 1, 2, 3)

ESAPI proporciona soporte para mapeos numéricos y aleatorios

<http://app?file=Report123.xls>  
<http://app?file=1>

<http://app?id=9182374>  
<http://aoo?id=7d3J93>

Access  
Reference  
Map

Report123.xls

Acct:9182374

Validar la referencia directa al objeto

Verificar que el valor del parámetro se encuentra adecuadamente formateado

Verificar que el usuario se encuentra autorizado a acceder el objeto determinado

Restricciones en los parámetros funcionan muy bien!



# OWASP

The Open Web Application Security Project

## Pecado V

# Configuración defectuosa

Una buena seguridad requiere tener definida e implementada una configuración segura para la aplicación, marcos de trabajo, servidor de aplicación, servidor web, base de datos, y plataforma. Todas estas configuraciones deben ser definidas, implementadas, y mantenidas ya que por lo general no son seguras por defecto. Esto incluye mantener todo el software actualizado, incluidas las librerías de código utilizadas por la aplicación.



# OWASP

The Open Web Application Security Project

## Configuraciones inseguras

Se requiere un proceso concertado, repetible y replicable, para desarrollar y mantener una correcta configuración de seguridad de la aplicación.

Ejemplos:

- Plataformas vulnerables sin los últimos parches de seguridad
- Configuraciones iniciales o funciones innecesarias activadas
- Usuarios de sistema operativo o bases de datos con permisos excesivos

LINUX & MAC OS

Falla del bash  
Kernel

MICROSOFT

Configuración por defecto  
Administrator  
sa



# OWASP

The Open Web Application Security Project

## Prevenir configuraciones inseguras

Los programas de auditoría de seguridad son por lo general buenos para abarcar grandes cantidades de pruebas para detectar la falta de parches de seguridad así como la existencia de configuraciones por defecto, la verificación manual posteriormente será una responsabilidad muy importante por parte del profesional de seguridad.



# OWASP

The Open Web Application Security Project

## Pecado VI

# Protección Insuficiente en la capa de Transporte

Las aplicaciones frecuentemente fallan al autenticar, cifrar, y proteger la confidencialidad e integridad de tráfico de red sensible. Cuando esto ocurre, es debido a la utilización de algoritmos débiles, certificados expirados, inválidos, o sencillamente no utilizados correctamente.



# OWASP

The Open Web Application Security Project

## Protección Insuficiente en la capa de Transporte (TLS)

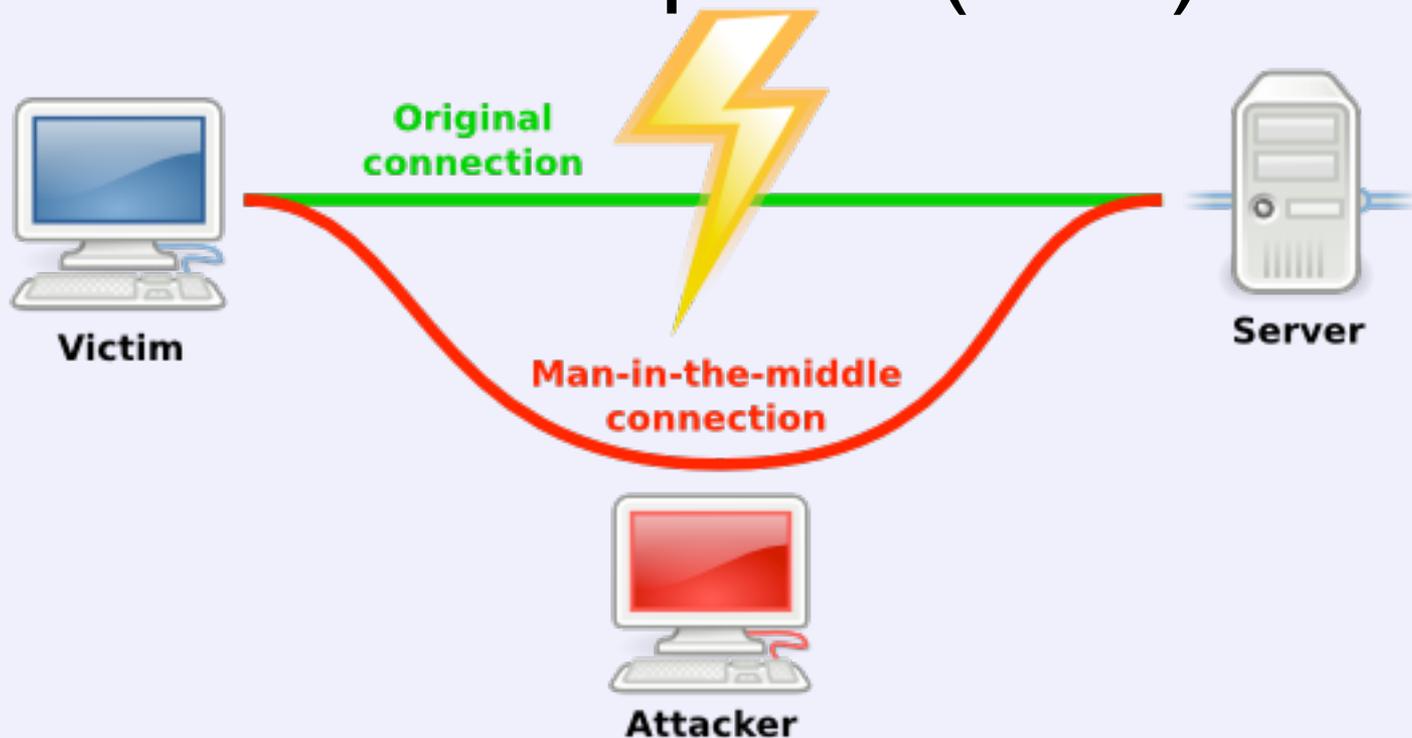
- Se utiliza SSL para proteger todo el tráfico relacionado con la autenticación? (Contra ataques MiTM)
- Se utiliza SSL para todos los recursos de páginas y servicios privados?
- Se debe evitar el acceso SSL únicamente a determinados recursos de una página ya que esto provoca advertencias en el navegador y puede exponer el identificador de sesión de los usuarios.
- Sólo se soportan algoritmos considerados fuertes.
- Todas las cookies de sesión tienen el atributo “secure” activado.
- El certificado debe ser legítimo y estar configurado correctamente para este servidor. Un usuario acostumbrado a lidiar con mensajes de error en su sitio perderá la desconfianza a un sitio falso con las mismas características



# OWASP

The Open Web Application Security Project

## Protección Insuficiente en la capa de Transporte (TLS)





# OWASP

The Open Web Application Security Project

## Pecado VII

### LFI y LFD

- Vulnerabilidades que se hacen presentes cuando se permite que el usuario incluya archivos en la aplicación
- Local File Inclusion (LFI): inclusión de archivos locales, donde se encuentre el sitio web vulnerable
- Local File Disclosure (LFD): permite la descarga de archivos en código fuente directo.



# OWASP

The Open Web Application Security Project

## Evitar LFD y LFI

- UrlScan por defecto bloquea: exe, bat, cmd, com, htw, ida, idq, htr, idc, printer, ini, pol, dat, etc.
  - Validar extenciones mediante un lista blanca
- En PHP se puede usar `filter_var` Security y/o `filter_allow_url_schemes`

**REGLA DE ORO**

Todas las validaciones se deben validar en ambos lados: cliente y servidor



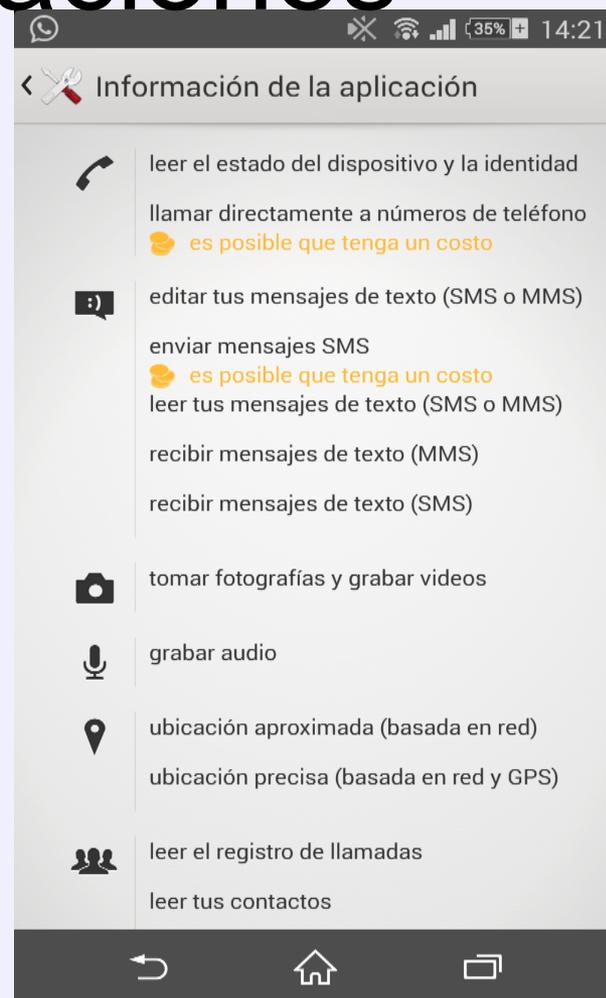
# OWASP

The Open Web Application Security Project

# Pecado con las aplicaciones moviles



elvin.mollinedo@owasp.org





**OWASP**

The Open Web Application Security Project

# GRACIAS

Ing. Elvin Vidal Mollinedo Mencia

[elvin.mollinedo@owasp.org](mailto:elvin.mollinedo@owasp.org)