



Aportación de la OWASP a la comunidad internacional.  
Seguridad en las relaciones de confianza.

**Vicente Aguilera Díaz**  
**OWASP Spain Chapter Leader**

CISA, CISSP, ITILF, CEH Instructor, OPSA, OPST  
vicente.aguilera@owasp.org

28 marzo 2008



**The OWASP Foundation**  
<http://www.owasp.org>

# ¿Quién soy?

Vicente Aguilera Díaz

- CISA, CISSP, ITILF, CEH Instructor, OPSA, OPST
- OWASP Spain chapter leader
- Socio co-fundador de Internet Security Auditors
- Miembro del consejo técnico asesor de RedSeguridad
- 6 años focalizado en seguridad en aplicaciones Web
- Colaborador de OWASP Testing Guide v2, WASC Threat Classification v2
- Artículos y conferencias sobre seguridad en aplicaciones
- Vulnerabilidades en Oracle, SquirrelMail, Hastymail, ISMail, GMail, ...

# Agenda

- Aportación de la OWASP a la comunidad internacional
- Capítulo español de la OWASP
- Seguridad en las relaciones de confianza
- ¿Preguntas?

## Aportación de la OWASP a la comunidad internacional

- El Open Web Application Seguridad Project (OWASP) está **dedicado a la búsqueda y la lucha contra las causas de software inseguro**. La OWASP Foundation es una organización sin ánimo de lucro que proporciona la infraestructura y apoya nuestro trabajo.
- La participación es gratuita y abierta para todos
- Aquí todo es gratuito y de código abierto
- Objetivos: crear herramientas, documentación y estándares relacionados con la seguridad en aplicaciones
- 7685 miembros y 114 capítulos locales en el mundo



- Todos los miembros son voluntarios
- Comunicación: MediaWiki ([www.owasp.org](http://www.owasp.org))
- Proporciona recursos gratuitos a la comunidad:  
publicaciones, artículos, estándares, aplicaciones de test  
y aprendizaje, capítulos locales, listas de correo y  
conferencias
- Modelo de licencia: Open Source y licencias comerciales  
para miembros

# OWASP



## El estilo OWASP: “open”

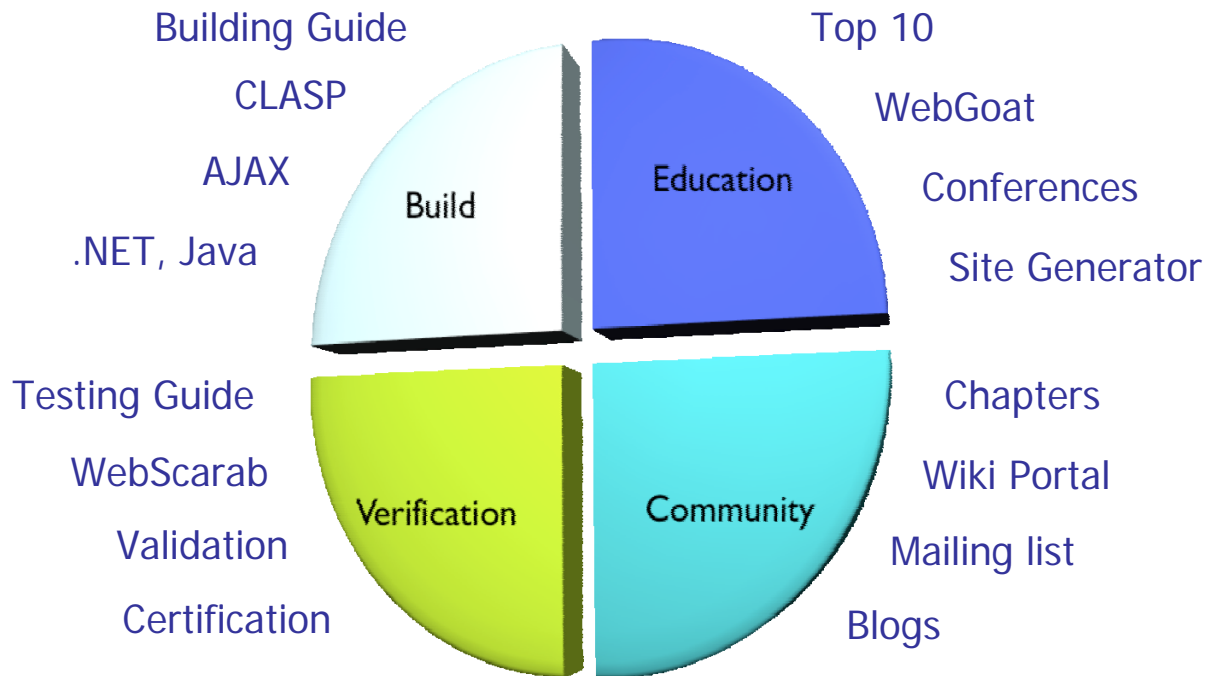
- Gratuito
- Consensuado
- Libre de utilizar y modificar
- Independiente
- Compartiendo conocimientos
- Ámbito público y participación

## Causas de software inseguro

- Vulnerabilidades
- Desarrolladores
- Estructura organizativa, procesos de desarrollo, tecnología
- Incremento de conectividad y complejidad
- Requerimientos legales



## Aportaciones



... y muchos otros proyectos!

[http://www.owasp.org/index.php/Category:OWASP\\_Project](http://www.owasp.org/index.php/Category:OWASP_Project)

## Aportaciones

### ■ Proyectos (**Herramientas**)

- ▶ OWASP WebGoat Project
- ▶ OWASP WebScarab Project
- ▶ OWASP AntiSamy Project
- ▶ OWASP CAL9000 Project
- ▶ OWASP CSRFGuard Project
- ▶ OWASP DirBuster Project
- ▶ OWASP Encoding Project
- ▶ OWASP LAPSE Project
- ▶ OWASP Live CD Education Project
- ▶ OWASP Live CD Project
- ▶ OWASP .NET Research

## Aportaciones

### ■ Proyectos (**Herramientas**)

- ▶ OWASP Pantera Web Assessment Studio Project
- ▶ OWASP Report Generator
- ▶ OWASP Site Generator
- ▶ OWASP SQLiX Project
- ▶ OWASP Tiger
- ▶ OWASP WeBekci Project
- ▶ OWASP WSFuzzer Project
- ▶ OWASP CSRFTester Project
- ▶ OWASP Insecure Web App Project
- ▶ OWASP Interceptor Project
- ▶ OWASP JBroFuzz Project

## Aportaciones

### ■ Proyectos (**Herramientas**)

- ▶ OWASP Sprajax Project
- ▶ OWASP Stinger Project
- ▶ OWASP Web 2.0 Project

25 PROYECTOS SOBRE HERRAMIENTAS

## Aportaciones

### ■ Proyectos (**Documentación**)

- ▶ OWASP AppSec FAQ Project
- ▶ OWASP Guide Project
- ▶ OWASP Legal Project
- ▶ OWASP Testing Guide
- ▶ OWASP Top Ten Project
- ▶ OWASP CLASP Project
- ▶ OWASP Code Review Project
- ▶ OWASP Tools Project
- ▶ OWASP AJAX Security Guide
- ▶ OWASP Application Security Assessment Standards Project
- ▶ OWASP Application Security Requirements

## Aportaciones

### ■ Proyectos (**Documentación**)

- ▶ OWASP Application Security Metrics Project
- ▶ OWASP Career Development Project
- ▶ OWASP Certification Criteria Project
- ▶ OWASP Certification Project
- ▶ OWASP Communications Project
- ▶ OWASP Honeycomb Project
- ▶ OWASP Java Project
- ▶ OWASP Logging Guide
- ▶ OWASP PHP Project
- ▶ OWASP Scholastic Application Security Assessment Project
- ▶ OWASP Validation Project

## Aportaciones

### ■ Proyectos (**Documentación**)

- ▶ OWASP WASS Guide
- ▶ OWASP Web Application Security Put Into Practice
- ▶ OWASP XML Security Gateway Evaluation Criteria
- ▶ OWASP Education Project
- ▶ OWASP on The Move Project
- ▶ OWASP Fuzzing Code Database

28 PROYECTOS DE DOCUMENTACIÓN

## Han adoptado el OWASP Top Ten...



**Defense Information Systems Agency**  
Department of Defense



- La Federal Trade Commission (EEUU) recomienda encarecidamente que todas las empresas usen el OWASP Top Ten y se aseguren de que sus partners hagan lo mismo.
- La Defense Information Systems Agency (EEUU) ha enumerado el OWASP Top Ten como las mejores prácticas a utilizar como parte del DOD Information Technology Security Certification and Accreditation (C&A) Process (DITSCAP).



# OWASP



Han adoptado el OWASP Top Ten...

- Un gran número de organizaciones internacionales



a little word for a big life



... y muchas otras en todo el mundo!

Han adoptado el OWASP Top Ten...

- El estándar **Payment Card Industry** (PCI) ha adoptado el OWASP Top Ten, y requiere (entre otros aspectos) que todos los comercios realicen una auditoría de código de las aplicaciones que desarrollan.
  - ▶ PCI Requirement 6.5 points towards the OWASP project as a source of secure coding guidance.
  - ▶ PA-DSS (Payment Application Data Security Standard)

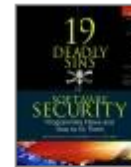
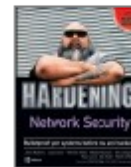
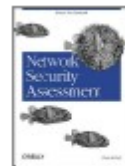
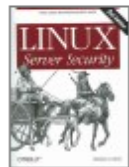
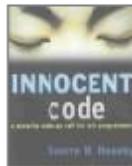


<http://www.pcisecuritystandards.org>

## Referencias a nuestros proyectos...

### ■ Gran número de libros:

[http://books.google.com/books?as\\_q=owasp](http://books.google.com/books?as_q=owasp)



## A nivel español

- Entidades financieras
- Universidades
- Empresas de desarrollo
- Empresas de seguridad

### ■ El portal EPOCA ja és accessible des d'Internet

#### COM HI PUC ACCEDIR?

Heu d'introduir al vostre navegador l'adreça <https://epoca.gencat.net>, i a continuació se us demanaran els vostres nom d'usuari i contrasenya d'EPOCA. També hi podeu accedir des de la pàgina web de la Funció Pública, que és a l'adreça: <http://www.gencat.net/governacio-ap/administracio/index.htm>.

#### ÉS SEGUR ACCEDIR A EPOCA DES D'INTERNET?

Per tal de garantir un nivell de seguretat adient, s'utilitza una connexió segura (SSL), que suposa, entre altres característiques, l'encriptació de totes les dades.

Així mateix, i per tal de garantir una valoració objectiva d'aquesta seguretat, s'ha encomanat al Centre de Telecomunicacions i Tecnologies de la Informació (CTITI) la realització d'una auditoria del portal per tal de dotar l'aplicació de la màxima seguretat. Aquesta auditoria l'ha realitzat una empresa homologada per a aquest tipus d'activitat, tot seguint les directrius OWASP (Open Web Application Security Project).

ity Project ). Aquestes directrius són consultables a l'adreça [www.owasp.org](http://www.owasp.org). Totes les recomanacions derivades de l'esmentada auditoria han estat implantades.



L'adjudicatari està obligat a guardar secret respecte les dades o informació prèvia que no essent públics o notoris estiguin relacionats amb l'objecte del contracte.

Qualsevol comunicat de premsa o inserció als mitjans de comunicació que el proveïdor realitzi referent al servei que presta a la Generalitat haurà de ser aprovat prèviament pel client.

#### Propietat intel·lectual

Tota la documentació que es generi al llarg del servei és propietat exclusiva de la Generalitat de Catalunya. El licitador no la podrà fer servir per altres finalitats sense el consentiment exprés del client.

#### Seguretat i protecció de dades

L'adjudicatari dels serveis es compromet a complir els requeriments de seguretat i continuïtat aplicables a l'objecte del contracte especificats a:

- La legislació vigent en general i, en particular, quan es tractin dades de caràcter personal, el Reglament de Seguretat del Reial Decret 994/1999 de la Llei Orgànica de Protecció de Dades de Caràcter Personal (LOPD).
- Les normes ISO/IEC/UNE 17799 de millors pràctiques de seguretat de la informació i UNE71502 de gestió de la seguretat de la informació, adaptades a l'estructura administrativa, personal i entorn tecnològic del client i aplicades de forma proporcional als riscos reals
- Els requeriments de seguretat de webs que publiqui l'IQUA (Agència de Qualitat d'Internet).

Adicionalment, l'adjudicatari es compromet a:

- complir amb les directives tecnològiques i de seguretat i qualitat que estableixi el client.
- implementar les mesures, processos, i requeriments que el client sol·liciti amb aquesta finalitat i li proposarà els que consideri necessaris per millorar les solucions.
- facilitar tota aquella informació que el client requereixi per tal que aquest pugui donar compliment a la legislació i normativa referida en aquest apartat.

A més, es valorarà el compliment d'altres estàndards de seguretat i continuïtat, com:

- millors pràctiques i indicadors de seguretat: COBIT (Control Objectives for Information and related Technology).
- pràctiques de programació de webs segures: OWASP (Open Web Application Security Project).

#### Compartició de recursos.

Per motius de garantir la seguretat, qualsevol compartició de recursos tècnics (infraestructura de maquinari, etc.) utilitzats en el marc de l'execució del contracte serà prèviament justificada al client amb un informe d'anàlisi de beneficis i riscos, que aquest haurà d'aprovar.



## Principales proyectos

### ■ Documentación:

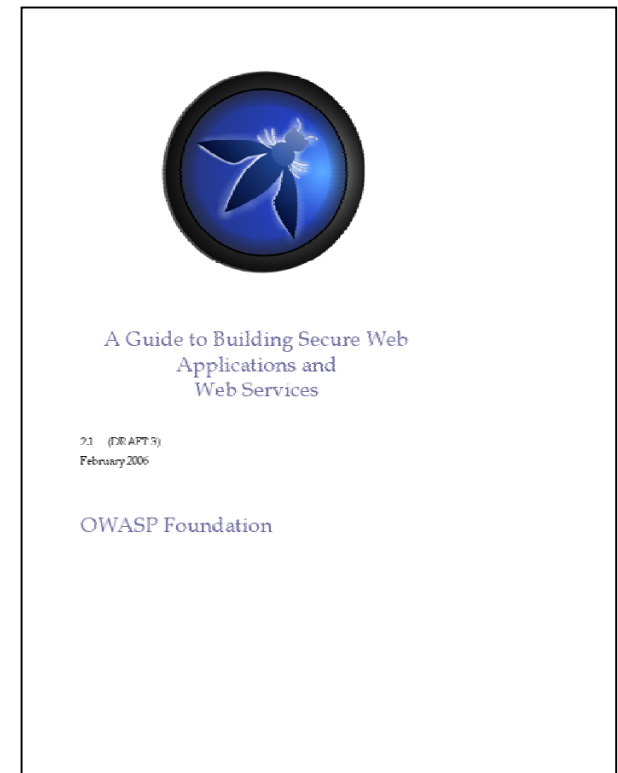
- ▶ A Guide to Building Secure Web Applications and Web Services
- ▶ Testing Guide
- ▶ Top Ten

### ■ Herramientas:

- ▶ WebScarab
- ▶ WebGoat

## A Guide to Building Secure Web Applications and Web Services

- Es un libro: 310 páginas
- Gratuito
- Muchos colaboradores
- Aplicaciones y Web Services
- Ejemplos en J2EE, PHP, ASP.NET
- Exhaustivo
- Evoluciona (1ª versión en 2002)



[http://www.owasp.org/index.php/Category:OWASP\\_Guide\\_Project](http://www.owasp.org/index.php/Category:OWASP_Guide_Project)

## A Guide to Building Secure Web Applications and Web Services

### ■ Orientado a:

- ▶ **Desarrolladores:** guía para implementar mecanismos de seguridad y evitar vulnerabilidades
- ▶ **Jefes de Proyecto:** identificar actividades a realizar (modelado de amenazas, revisión de código, pentest, etc.)
- ▶ **Equipos de Seguridad:** estructurar las pruebas, conocer mecanismos de seguridad y soluciones

## A Guide to Building Secure Web Applications and Web Services

### ■ 27 capítulos:

- 1. About the Open Web Application Security Project
- 2. Introduction
- 3. What are Web Applications?
- 4. Policy Frameworks
- 5. Secure coding principles
- 6. Threat risk modeling
- 7. Handling e-commerce payments
- 8. Phishing
- 9. Web Services
- 10. AJAX and other “rich” interface technologies
- 11. Authentication
- 12. Authorization
- 13. Session Management
- 14. Data Validation
- 15. Interpreter injection
- 16. Canoncalization, locale and unicode
- 17. Error handling, auditing and logging
- 18. File system
- 19. Distributed computing
- 20. Buffer overflows
- 21. Administrative interfaces
- 22. Cryptography
- 23. Configuration
- 24. Software quality assurance
- 25. Deployment
- 26. Maintenance
- 27. GNU free documentation license



## A Guide to Building Secure Web Applications and Web Services

### ■ Cubre las **áreas clave** de la seguridad en aplicaciones:

- ▶ Autenticación
- ▶ Autorización
- ▶ Gestión de sesiones
- ▶ Validación de datos
- ▶ Canonicalización
- ▶ Gestión de errores, log y auditoría
- ▶ Configuración

## A Guide to Building Secure Web Applications and Web Services

- Para cada aspecto tratado:
  - ▶ Objetivos
  - ▶ Teoría
  - ▶ Buenas prácticas
  - ▶ Cómo determinar si somos vulnerables
  - ▶ Cómo protegernos

## A Guide to Building Secure Web Applications and Web Services

### ■ Autenticación:

- ▶ Técnicas habituales de autenticación Web
- ▶ Fuerza bruta
- ▶ CAPTCHA
- ▶ Autenticación fuerte
- ▶ Autenticación positiva
- ▶ Selección de nombres de usuario
- ▶ Cambios de contraseña
- ▶ Logout
- ▶ ...

## A Guide to Building Secure Web Applications and Web Services

### ■ Autorización:

- ▶ Principio de mínimo privilegio
- ▶ Rutinas centralizadas de autorización
- ▶ Matriz de autorización
- ▶ Control de acceso sobre recursos protegidos
- ▶ Protección del acceso a recursos estáticos
- ▶ Re-autorización en acciones de alto valor
- ▶ ...

## A Guide to Building Secure Web Applications and Web Services

### ■ Gestión de sesiones:

- ▶ Exposición de variables de sesión
- ▶ Tokens de sesión
- ▶ Detección de ataques de fuerza bruta
- ▶ Debilidad de los algoritmos criptográficos
- ▶ Regeneración de tokens de sesión
- ▶ Secuestro de sesión
- ▶ Ataques a la validación de sesiones
- ▶ ...

## A Guide to Building Secure Web Applications and Web Services

### ■ Validación de datos:

- ▶ Donde incluir validaciones
- ▶ Donde incluir verificaciones de integridad
- ▶ Donde incluir validaciones de las reglas de negocio
- ▶ Estrategias de validación de datos
- ▶ Prevención de la alteración de datos
- ▶ Codificación de URL y HTML
- ▶ ...

## A Guide to Building Secure Web Applications and Web Services

### ■ Canonicalización:

- ▶ Formatos de entrada
- ▶ UNICODE
- ▶ Doble (o N-) codificación
- ▶ HTTP Request Smuggling
- ▶ ...

## A Guide to Building Secure Web Applications and Web Services

### ■ Gestión de errores, log y auditoría:

- ▶ Mensajes de error detallados
- ▶ Logging
- ▶ Ruído
- ▶ Cubrir pistas
- ▶ Falsas alarmas
- ▶ Pistas de auditoría
- ▶ Gestión de errores
- ▶ ...



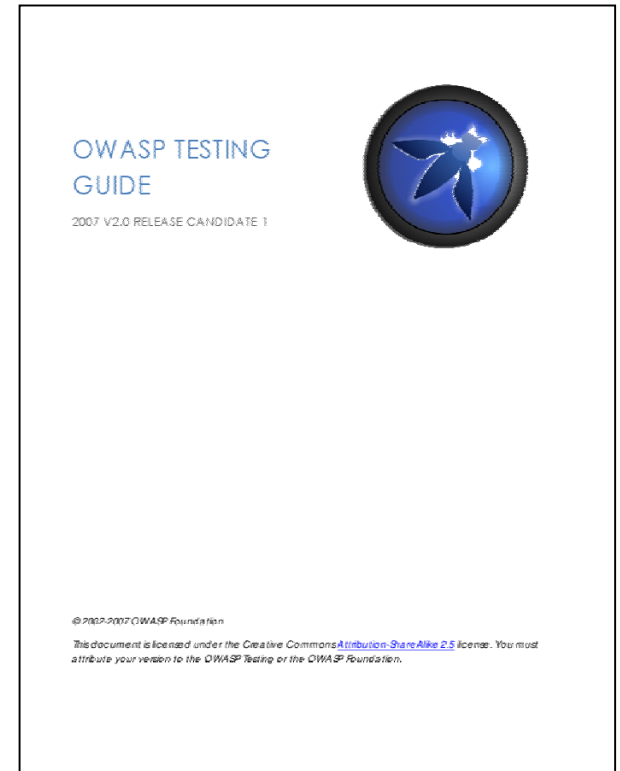
## A Guide to Building Secure Web Applications and Web Services

### ■ Configuración:

- ▶ Contraseñas por defecto
- ▶ Cadenas de conexión seguras
- ▶ Transmisión por red segura
- ▶ Cifrado de datos
- ▶ Seguridad en la base de datos
- ▶ ...

## Testing Guide

- Es un libro: 270 páginas
- Gratuito
- Muchos colaboradores
- Aborda todo el SDLC
- Exhaustivo
- Evoluciona (1ª versión en 2004)
- Traducido a español!



[http://www.owasp.org/index.php/Category:OWASP\\_Testing\\_Project](http://www.owasp.org/index.php/Category:OWASP_Testing_Project)

## Testing Guide

### ■ Pruebas de seguridad en aplicaciones

- ▶ Si los coches se construyeran como las aplicaciones...
  - Los tests de seguridad únicamente asumirían impactos frontales
  - No existirían tests para verificar la estabilidad en maniobras de emergencias
  - No existirían tests para verificar la efectividad de los frenos
  - No existirían tests para verificar la resistencia al robo
  - ...

Denis Verdon, Head of Information Security (Fidelity National Financial)  
OWASP AppSec 2004 Conference (New York)

## Testing Guide

### ■ Pruebas de seguridad en aplicaciones

***"Me alegro de que los desarrolladores de software no construyan coches"***

Eoin Keary, líder del proyecto OWASP Code Review

## Testing Guide

### ■ Autores

- Vicente Aguilera
- Mauro Bregolin
- Tom Brennan
- Gary Burns
- Luca Carettoni
- Dan Cornell
- Mark Curphey
- Daniel Cuthbert
- Sebastien Deleersnyder
- Stephen DeVries
- Stefano Di Paola
- David Endler
- Giorgio Fedon
- Javier Fernández-Sanguino
- Glyn Geoghegan
- Stan Guzik
- Madhura Halasgikar
- Eoin Keary
- David Litchfield
- Andrea Lombardini
- Ralph M. Los
- Claudio Merloni
- Matteo Meucci
- Marco Morana
- Laura Nunez
- Gunter Ollmann
- Antonio Parata
- Yiannis Pavlosoglou
- Carlo Pelliccioni
- Harinath Pudipeddi
- Alberto Revelli
- Mark Roxberry
- Tom Ryan
- Anush Shetty
- Larry Shields
- Dafydd Studdard
- Andrew van der Stock
- Ariel Weissbein
- Jeff Williams

Con colaboración de miembros del capítulo español!

## Testing Guide

### ■ Contenido

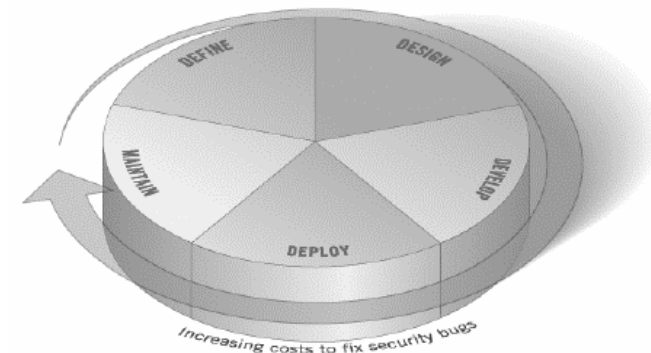
- ▶ 1. Portada
- ▶ 2. Introducción
- ▶ 3. El entorno de pruebas OWASP
- ▶ **4. Pruebas de intrusión en aplicaciones Web**
- ▶ 5. Redacción de informes: valorar el riesgo real
- ▶ Apéndice A: Herramientas de comprobación
- ▶ Apéndice B: Lectura recomendada
- ▶ Apéndice C: Vectores de fuzzing

## Testing Guide

### El entorno de pruebas OWASP

#### ■ Fase 1: Antes de comenzar el desarrollo

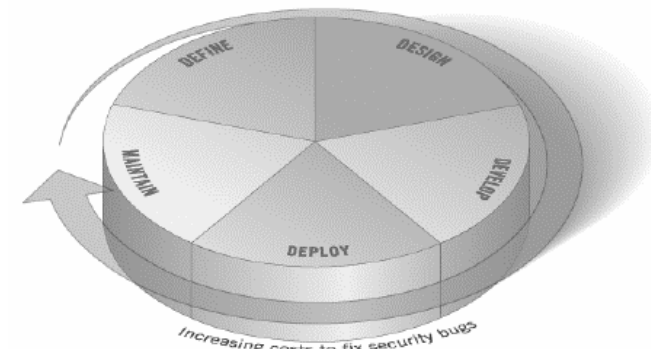
- ▶ Verificar que el SDLC propuesto es adecuado e incorpora la capa de seguridad
- ▶ Verificar que la política y estándares son conocidos por los miembros del equipo
- ▶ Desarrollar métricas



## Testing Guide

### El entorno de pruebas OWASP

- Fase 2: Durante definición y diseño
  - ▶ Revisar requerimientos de seguridad
  - ▶ Revisión del diseño y arquitectura
  - ▶ Crear y revisar modelos de amenazas



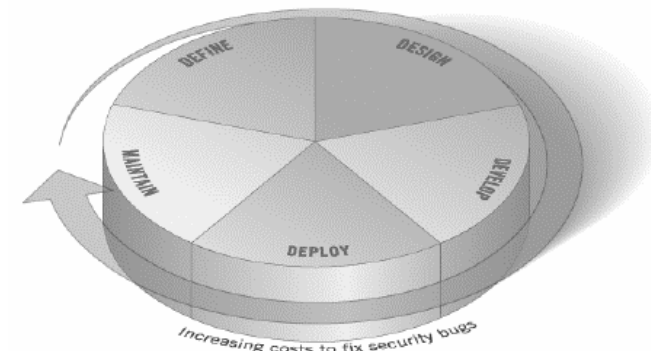


## Testing Guide

### El entorno de pruebas OWASP

#### ■ Fase 3: Durante el desarrollo

- ▶ Inspección de código
- ▶ Revisión de código



## Testing Guide

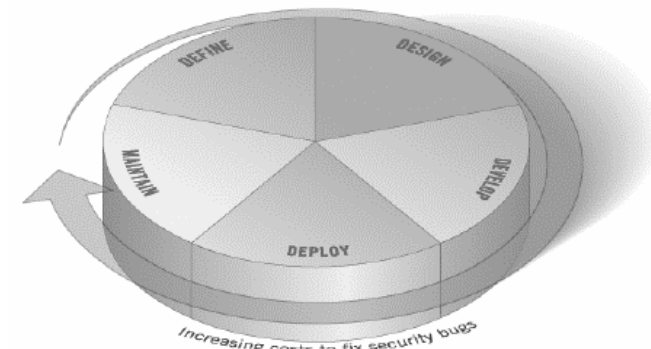
### El entorno de pruebas OWASP

#### ■ Fase 4: Durante el despliegue

- ▶ Pentest de aplicaciones
- ▶ Verificar la gestión de configuraciones

#### ■ Fase 5: Mantenimiento y Operación

- ▶ Revisiones de la gestión operativa
- ▶ Verificaciones periódicas
- ▶ Asegurar la verificación de cambios



## Testing Guide

### Pruebas de intrusión en aplicaciones Web

- ¿Qué es una prueba de intrusión en aplicaciones Web?
  - ▶ Método de evaluación de la seguridad mediante la simulación de un ataque
- ¿Qué es una vulnerabilidad?
  - ▶ Debilidad en un activo que hace posible una amenaza
- Nuestra aproximación a la hora de escribir esta guía:
  - ▶ Abierta
  - ▶ Colaborativa

## Testing Guide

### Pruebas de intrusión en aplicaciones Web

■ Hemos dividido el conjunto de pruebas en 8 subcategorías (para un total de 48 controles):

- ▶ Recopilación de información
- ▶ Comprobación de la lógica de negocio
- ▶ Pruebas de autenticación
- ▶ Pruebas de gestión de sesiones
- ▶ Pruebas de validación de datos
- ▶ Pruebas de denegación de servicio
- ▶ Pruebas de servicios Web
- ▶ Pruebas de AJAX

## Testing Guide

### Pruebas de intrusión en aplicaciones Web

#### ■ Por ejemplo...

Pruebas de validación de datos	OWASP-DV-001	Cross site scripting
	OWASP-DV-002	Métodos HTTP y XST
	OWASP-DV-003	Inyección SQL
	OWASP-DV-004	Inyección de procedimientos almacenados
	OWASP-DV-005	Inyección ORM
	OWASP-DV-006	Inyección LDAP
	OWASP-DV-007	Inyección XML
	OWASP-DV-008	Inyección SSI
	OWASP-DV-009	Inyección XPath
	OWASP-DV-010	Inyección IMAP/SMTP
	OWASP-DV-011	Inyección de código
	OWASP-DV-012	Inyección de comandos de sistema
	OWASP-DV-013	Desbordamientos de búfer
	OWASP-DV-014	Vulnerabilidades incubadas
	OWASP-DS-001	Bloqueo de cuentas de usuario

## Testing Guide

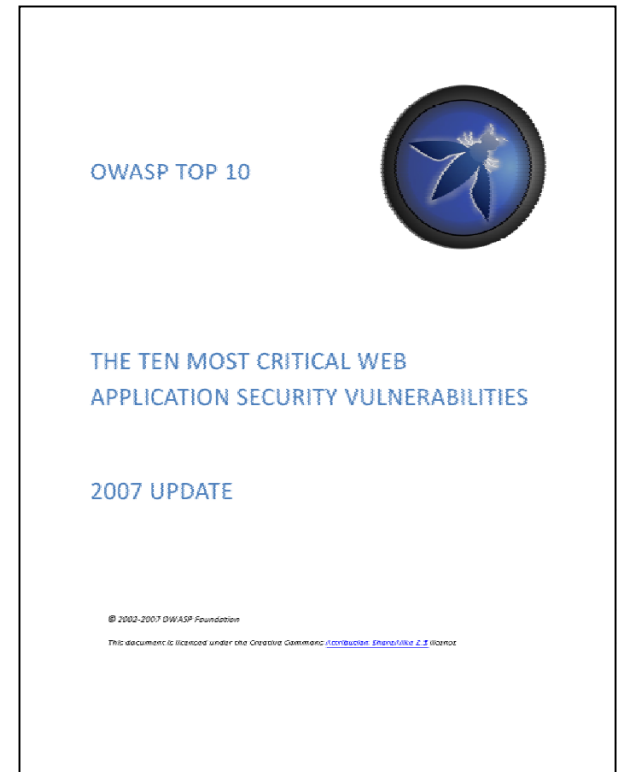
### Pruebas de intrusión en aplicaciones Web

#### ■ Plantilla para las pruebas

- ▶ Breve resumen
  - Descripción en “lenguaje natural” qué se espera probar
- ▶ Descripción
  - Breve descripción del problema
- ▶ Pruebas de caja negra y ejemplo
- ▶ Pruebas de caja gris y ejemplo
- ▶ Referencias
  - Whitepapers
  - Herramientas

## Top Ten

- Es un documento: 35 páginas
- Gratuito
- Muchos colaboradores
- Las 10 vulnerabilidades más críticas
- Evoluciona y se adapta



[http://www.owasp.org/index.php/Category:OWASP\\_Top\\_Ten\\_Project](http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project)

## Top Ten

- Enumera y describe las 10 vulnerabilidades **más críticas**
- Aporta recomendaciones
- Crecimiento en su aceptación
  - Federal Trade Commission (US Gov)
  - US Defense Information Systems Agency
  - VISA (Cardholder Information Security Program)
- Fuerte empuje para ser considerado un estándar



## Top Ten

### ■ Top Ten actual (2007)

- ▶ A1. Cross Site Scripting (XSS)
- ▶ A2. Injection Flaws
- ▶ A3. Malicious File Execution
- ▶ A4. Insecure Direct Object Reference
- ▶ A5. Cross Site Request Forgery (CSRF)
- ▶ A6. Information Leakage and Improper Error Handling
- ▶ A7. Broken Authentication & Session Management
- ▶ A8. Insecure Cryptographic Storage
- ▶ A9. Insecure Communications
- ▶ A10. Failure to Restrict URL Access

## Top Ten

### ■ Plantilla para cada vulnerabilidad

- ▶ Entornos afectados
- ▶ Vulnerabilidad
- ▶ Verificación de seguridad
- ▶ Protección
- ▶ Ejemplos
- ▶ Referencias

#### A4 – INSECURE DIRECT OBJECT REFERENCE

A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, database record, or key, as a URL or form parameter. An attacker can manipulate direct object references to access other objects without authorization, unless an access control check is in place.

For example, in Internet Banking applications, it is common to use the account number as the primary key. Therefore, it is tempting to use the account number directly in the web interface. Even if the developers have used parameterized SQL queries to prevent SQL injection, if there is no extra check that the user is the account holder and authorized to see the account, an attacker tampering with the account number parameter can see or change all accounts.

This type of attack occurred to the Australian Taxation Office's *GST Start Up Assistance* site in 2000, where a legitimate but hostile user simply changed the ABRN (a company tax id) present in the URL. The user formed around 17,000 company details from the system, and then e-mailed each of the 17,000 companies with details of his attack. This type of vulnerability is very common, but is largely untested in many applications.

#### ENVIRONMENTS AFFECTED

All web application frameworks are vulnerable to attacks on insecure direct object references.

#### VULNERABILITY

Many applications expose their internal object references to users. Attackers use parameter tampering to change references and violate the intended but unenforced access control policy. Frequently, these references point to file systems and databases, but any exposed application construct could be vulnerable.

For example, if code allows user input to specify filenames or paths, it may allow attackers to jump out of the application's directory, and access other resources.

```
<select name="language"><option value="fr">Français</option></select>
...
require_once ($_REQUEST['language']."lang.php");
```

Such code can be attacked using a string like `../../../../etc/passwd%00` using [null byte injection](#) (see the [OWASP Guide](#) for more information) to access any file on the web server's file system.

Similarly, references to database keys are frequently exposed. An attacker can attack these parameters simply by guessing or searching for another valid key. Often, these are sequential in nature. In the example below, even if an application does not present any links to unauthorized carts, and no SQL injection is possible, an attacker can still change the cartID parameter to whatever cart they want.

```
inc cartID = Integer.parseInt( request.getParameter( "cartID" ) );
Building query = "SELECT * FROM table WHERE cartID=" + cartID;
```

#### VERIFYING SECURITY

The goal is to verify that the application does not allow direct object references to be manipulated by an attacker.

## WebScarab

- Es una herramienta desarrollada en Java
- Permite realizar pruebas de seguridad en aplicaciones y servicios Web
- Analiza tráfico HTTP y HTTPs
- Múltiples usos
  - ▶ Desarrollador: tareas de debug y testing
  - ▶ Auditor de seguridad: identificación de vulnerabilidades

[http://www.owasp.org/index.php/Category:OWASP\\_WebScarab\\_Project](http://www.owasp.org/index.php/Category:OWASP_WebScarab_Project)

## WebScarab

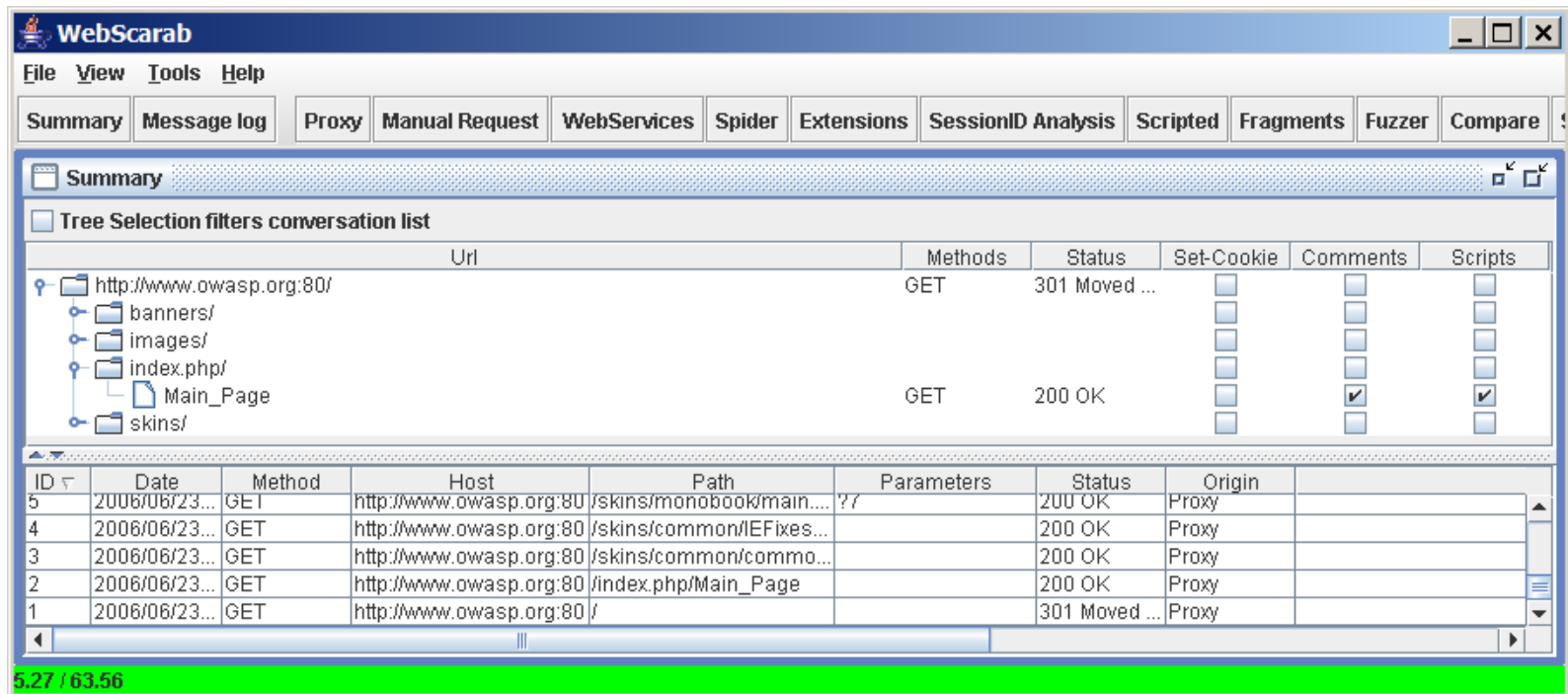
### ■ ¿Qué puede hacer?

- ▶ Proxy – observa tráfico entre el navegador y el servidor, incluyendo la capacidad de modificar la información transmitida
- ▶ Spider – identifica nuevas URLs en cada recurso visitado
- ▶ SessionID Analysis – genera y analiza cookies para determinar como de predecibles resultan los tokens de sesión
- ▶ Fuzzer – realiza sustitución automatizada de valores en los parámetros con el objetivo de detectar validaciones deficientes
- ▶ Fragment Analysis – extrae scripts y comentarios del código HTML

... y mucho más!

## WebScarab

### ■ Aspecto de la herramienta



## WebGoat

- Es una aplicación web, intencionadamente **insegura**
- Desarrollada en Java, basada en Tomcat y JDK 1.5
- Diseñada para aprender lecciones de seguridad en aplicaciones web
- En cada lección, los usuarios deben demostrar su conocimiento sobre un problema de seguridad, explotando una vulnerabilidad real en la aplicación WebGoat

[http://www.owasp.org/index.php/Category:OWASP\\_WebGoat\\_Project](http://www.owasp.org/index.php/Category:OWASP_WebGoat_Project)

## WebGoat

- Aplicación de aprendizaje
- Herramienta **educativa** para aprender sobre seguridad en aplicaciones web
- Entorno para analizar herramientas de seguridad

## WebGoat

### ■ ¿Qué puedes aprender?

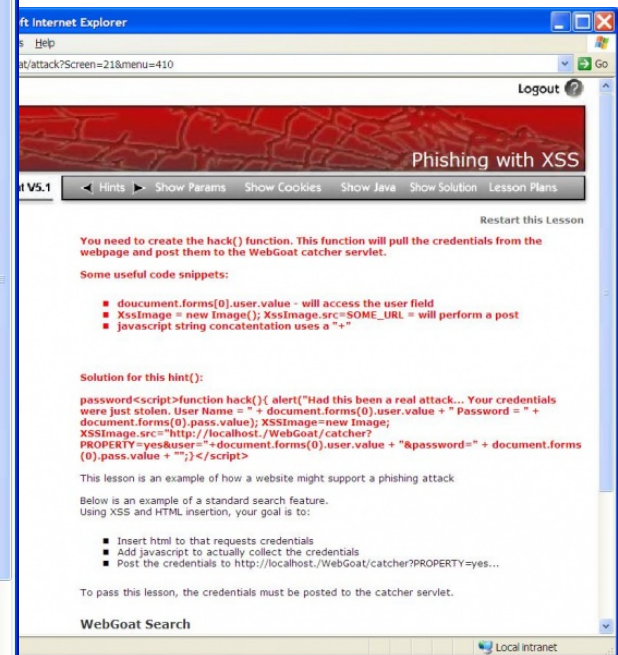
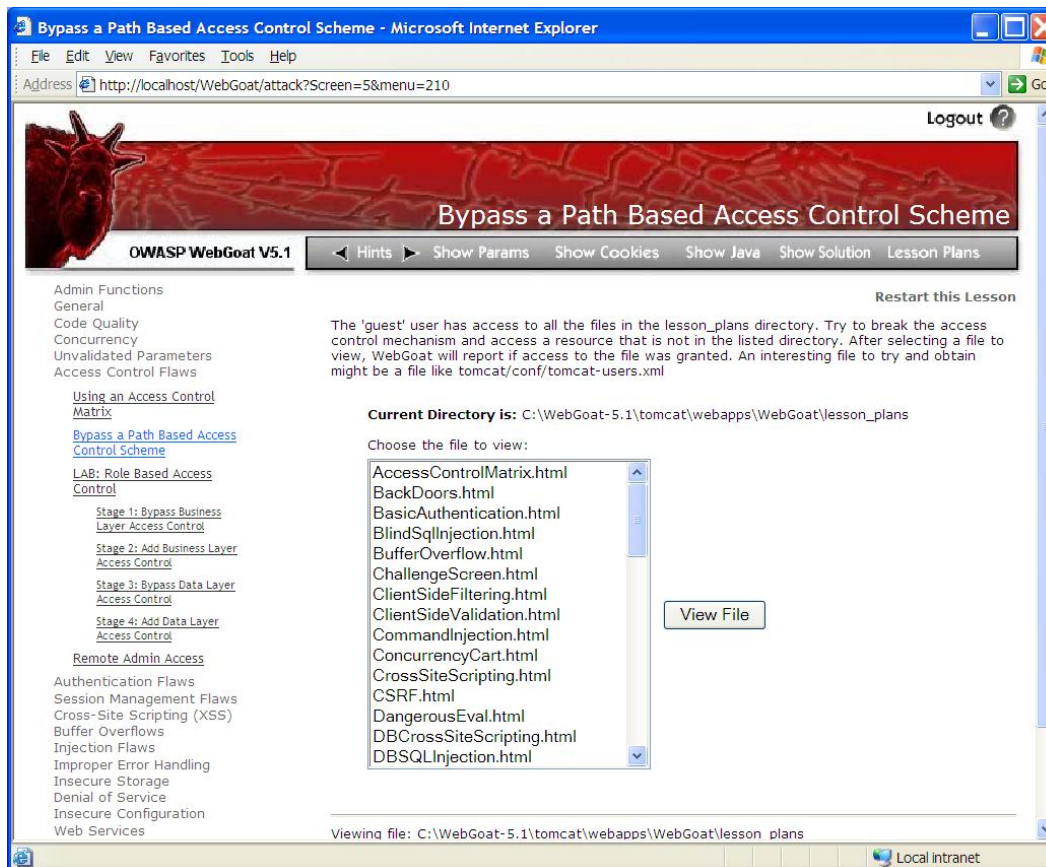
- ▶ Cross Site Scripting
- ▶ SQL Injection (Blind, numeric, etc.)
- ▶ Manipulación de parámetros y campos de formulario
- ▶ Cookies de sesión
- ▶ Control de acceso
- ▶ Web Services
- ▶ Fugas de información

... y mucho más a lo largo de 30 lecciones!



## WebGoat

### ■ Aspecto de la aplicación



## Capítulo español de la OWASP

# OWASP-Spain



Marzo / Abril 2006

- <http://www.owasp.org/index.php/Spain>
- Fundado en diciembre de 2005 por Vicente Aguilera
- Ubicado en Barcelona
- 176 miembros (cuarto capítulo europeo en número de miembros)
- Uno de los 50 recursos más visitados de la OWASP

Patrocinador del capítulo



## En marcha el Capítulo de España de la Fundación OWASP

Ya ha comenzado la actividad en el Capítulo de España de la OWASP (*Open Web Application Security Project*), en favor de la seguridad de las aplicaciones web. En este sentido, se tiene previsto, próximamente, llevar a cabo congresos de seguridad, abiertos y gratuitos, en los que participarán expertos de prestigio internacional.

Dirigido por Vicente Aguilera, colaborador de proyectos open source de organismos como la WASC (*Web Application Security Consortium*) y la OISSG (*Open Information Systems Security Group*), el capítulo español fue fundado en diciembre de 2005 y su sede se encuentra en Barcelona.

## Reglas

- Gratuito y abierto para todos
- Idioma: preferiblemente en español
- No a las presentaciones comerciales
- Respeto sobre las distintas opiniones
- 1 CPE por cada hora de asistencia a nuestros OWASP Spain chapter meetings

## ¿Qué ofrecemos?

- Eventos semestrales
- Lista de correo local:  
<https://lists.owasp.org/mailman/listinfo/owasp-spain>
- Presentaciones
- Foro abierto de discusión
- Punto de encuentro de profesionales de la seguridad
- Difusión del conocimiento en España sobre seguridad en aplicaciones Web
- ¿Proyectos locales? ¡Necesitamos iniciativas!

En un futuro próximo...

- Colaboraciones con otras asociaciones/entidades
- Desarrollo de proyectos con iniciativas locales
- Eventos en distintas ciudades españolas
- Eventos con mayores actividades
- Nuevos contenidos en la Web del capítulo: nuevas secciones, herramientas, videos, etc.

## OWASP Spain chapter meeting

- Nuevo evento: octubre 2008
- Ubicación: por decidir... (¿alguna sugerencia?)
- Programa habitual: breve introducción a la OWASP, panel con cuatro ponentes sobre distintas temáticas... y coffee-break! ;-)
- Ponentes y temáticas: aceptamos propuestas!
- **140 asistentes** en nuestro último evento (marzo 2008)



## Sugerencias...

- Mantente actualizado: suscríbete a nuestra lista de correo
- Envía tus cuestiones sobre aplicaciones Web
- Participa en las discusiones
- Colabora en proyectos existentes
- Ten iniciativa: propuesta de nuevos proyectos
- Conoce a profesionales de la seguridad y amplía tus conocimientos: ¡asiste a nuestros meetings!



# ¿Dudas?



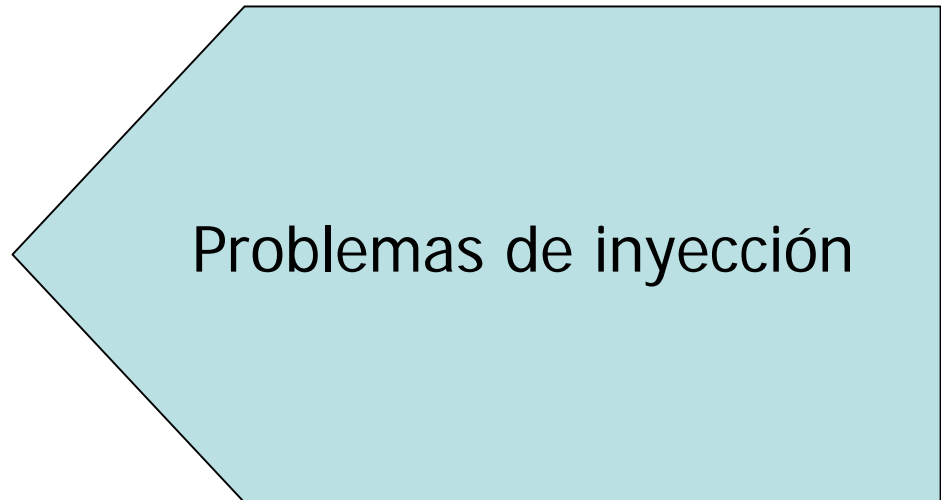
## Seguridad en las relaciones de confianza

# Relaciones de confianza

La aplicación Web no está aislada:

## ■ Relaciones con:

- ▶ Sistema operativo
- ▶ Servidores
  - Base de datos
  - Autenticación
  - Aplicación
  - FTP
  - Correo
  - ...
- ▶ Usuarios!



# Relaciones de confianza

## Ataques de inyección

- SQL Injection
- LDAP Injection
- SSI Injection
- IMAP/SMTP Injection
- XML Injection
- XPath Injection
- OS Command Injection
- User Agent Injection
- Code Injection
- ...

# Relaciones de confianza

## Ataques de inyección

### ■ ¿Cuándo puede ocurrir?

- ▶ Cuando datos facilitados por el usuario son enviados a un intérprete (o invocan otros procesos) como parte de un comando o consulta

### ■ Resultado

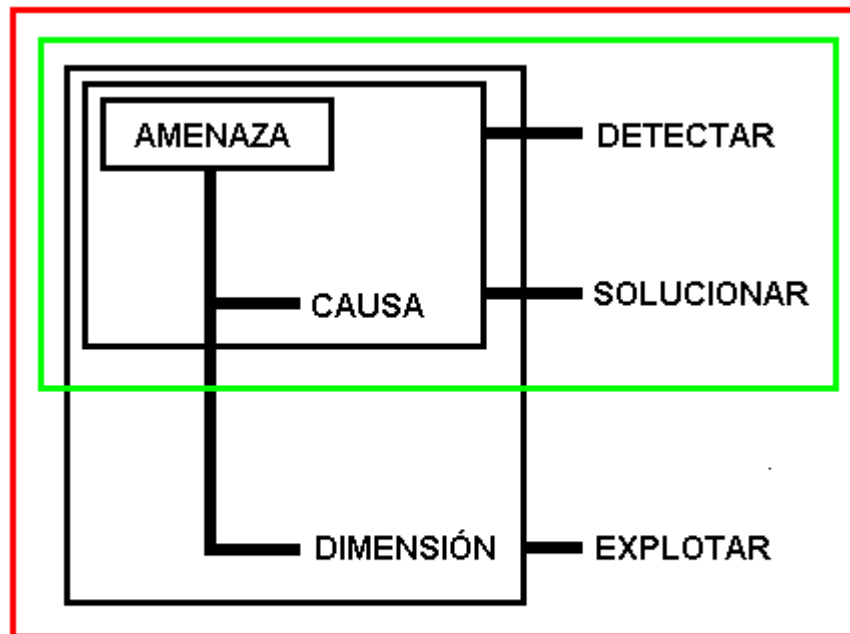
- ▶ El intérprete/proceso ejecuta comandos no deseados, controlados por el atacante

### ■ Causa del problema



- ▶ Validación deficiente de los datos de entrada/salida

# Relaciones de confianza

## Comprensión global de las amenazas



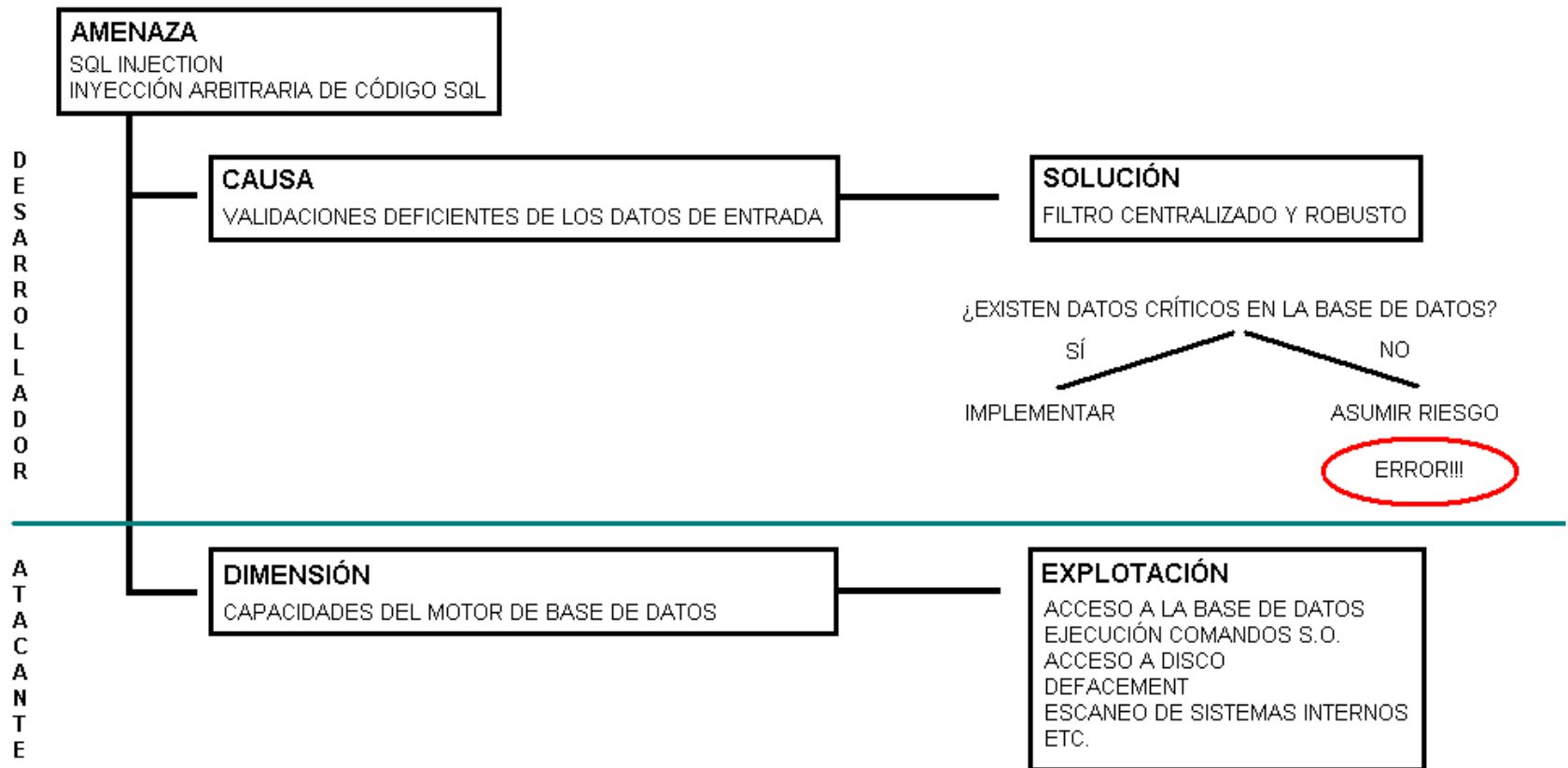
- Consciencia
- Entendimiento
- Solución conociendo la dimensión

 Visión típica del desarrollador  
 Visión típica del atacante

**Si se desconoce la dimensión  
es posible que no se implemente la solución**

# Relaciones de confianza

## Comprensión global de las amenazas



# Relaciones de confianza

## Relación con BBDD: SQL Injection

### ■ SQL Injection

- ▶ Alteración de la sentencia SQL que se ejecuta contra la base de datos mediante la manipulación de parámetros de entrada a la aplicación

### ■ Ejemplo típico

```
SQLQuery = "SELECT Username FROM Users WHERE Username = '"  
            & strUsername & "' AND Password = '" & strPassword & "'"  
strAuthCheck = GetQueryResult(SQLQuery)
```





# Relaciones de confianza

## Relación con BBDD: SQL Injection

### ■ Distintos tipos de inyección

- ▶ SQL Injection “normal”
- ▶ SQL Injection “a ciegas”
- ▶ SQL Injection “numérico”
- ▶ PL/SQL Injection

### ■ Clasificación

- ▶ Inband
- ▶ Out-of-band
- ▶ Inferential

¡Hay que entender la problemática de todos ellos!

# Relaciones de confianza

## Relación con BBDD: SQL Injection

### ■ Recomendaciones

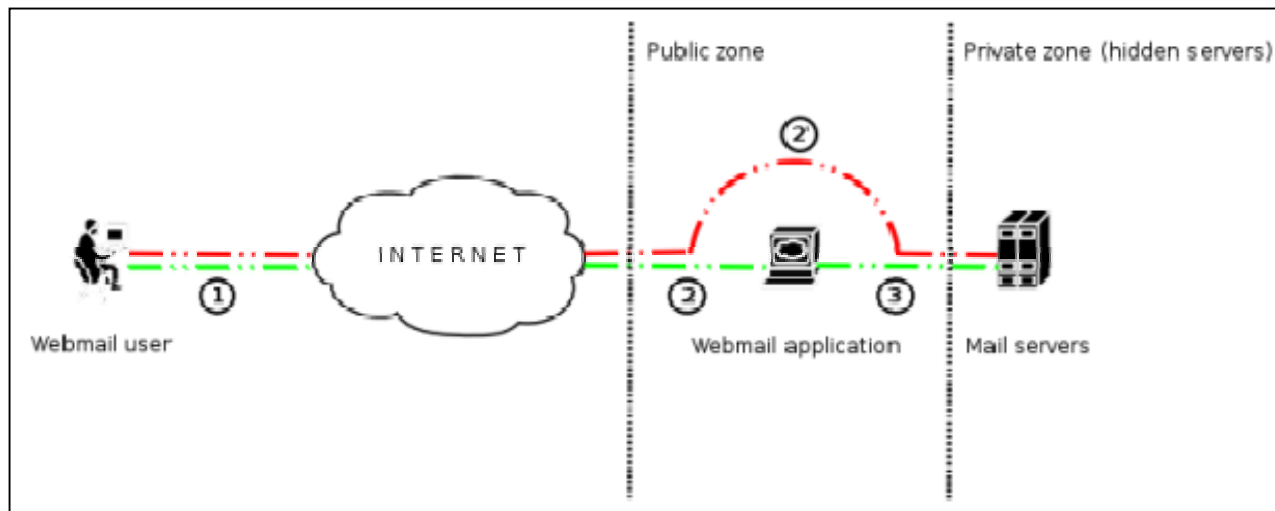
- ▶ Validación exhaustiva de todas las entradas
  - Filtro positivo (siempre que sea posible)
- ▶ Securización de la base de datos
- ▶ Uso de APIs parametrizadas
- ▶ No utilizar simples funciones de escape
- ▶ Mínimo privilegio
  - En las conexiones con la BBDD u otros componentes
- ▶ Limitar el tamaño de las entradas
  - No sólo en el cliente!
- ▶ Gestión robusta de errores
  - Evitar mensajes excesivamente detallados y fugas de información

# Relaciones de confianza

## Relación con MailServers: IMAP/SMTP Injection

### ■ Ejemplo 2: IMAP/SMTP Injection

- ▶ Alteración de comandos IMAP/SMTP que se ejecutan contra los servidores de correo mediante la manipulación de parámetros de entrada a la aplicación



# Relaciones de confianza

## Relación con MailServers: IMAP/SMTP Injection

- Se basa en la inyección de la secuencia CRLF
  - ▶ %0d%0a, \r\n
- Ataques posibles
  - ▶ Explotación de vulnerabilidades en el protocolo IMAP/SMTP
  - ▶ Evadir restricciones a nivel de aplicación
  - ▶ Provocar fugas de información
  - ▶ SPAM
  - ▶ ¿otros?
- Las posibilidades dependen del tipo y ámbito de la inyección y del servidor de correo analizado

# Relaciones de confianza

## Relación con MailServers: IMAP/SMTP Injection

### ■ Ejemplo

- ▶ [http://<webmail>/read\\_email.php?message\\_id=4791](http://<webmail>/read_email.php?message_id=4791)

```
FETCH 4791 BODY[HEADER]
```

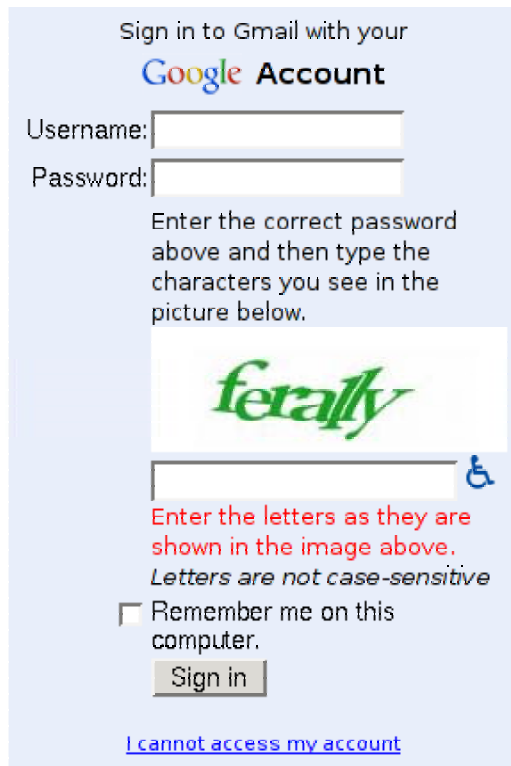
- ▶ `message_id=4791 BODY[HEADER]%0d%0aV100  
CAPABILITY%0d%0aV101 FETCH 4791`

```
???? FETCH 4791 BODY[HEADER]  
V100 CAPABILITY  
V101 FETCH 4791 BODY[HEADER]
```

# Relaciones de confianza

## Relación con MailServers: IMAP/SMTP Injection

### ■ Ejemplo práctico: evasión de CAPTCHA





Sign in to Gmail with your  
**Google Account**

Username:

Password:

Enter the correct password above and then type the characters you see in the picture below.





Enter the letters as they are shown in the image above.  
*Letters are not case-sensitive*

☐ Remember me on this computer.

[I cannot access my account](#)

El CAPTCHA evitaría ataques automáticos contra las contraseñas de los usuarios...

# Relaciones de confianza

## Relación con MailServers: IMAP/SMTP Injection

### ■ Ejemplo práctico: evasión de CAPTCHA

- ▶ Supongamos que un usuario utiliza las siguientes credenciales:
  - username = victima
  - Password = pwdok
- ▶ Supongamos que un atacante desea ejecutar un password cracking utilizando el siguiente diccionario de contraseñas:
  - pwderror1, pwderror2, pwdok, pwderror3
- ▶ Supongamos que el esquema de autenticación es vulnerable a IMAP Injection (en el campo "Password")



# Relaciones de confianza

## Relación con MailServers: IMAP/SMTP Injection

### ■ Ataque

- ▶ `http://<webmail>/src/login.jsp?login=victim&password=%0d%0aZ900 LOGIN victim pwderror1%0d%0aZ910 LOGIN victim pwderror2%0d%0aZ920 LOGIN victim pwdok%0d%0aZ930 LOGIN victim pwderror3`

```
C: Z900 LOGIN victim pwderror1
S: Z900 NO Login failed: authentication failure
C: Z910 LOGIN victim pwderror2
S: Z910 NO Login failed: authentication failure
C: Z920 LOGIN victim pwdok
S: Z920 OK User logged in
C: Z930 LOGIN victim pwderror3
S: Z930 BAD Already logged in
```

# Relaciones de confianza

## Relación con MailServers: IMAP/SMTP Injection

### ■ Recomendaciones

- ▶ Validación exhaustiva de todas las entradas
  - Filtro positivo (siempre que sea posible)
- ▶ No permitir la secuencia CRLF
- ▶ Utilizar librerías seguras
- ▶ Securitización de los servidores de correo
- ▶ Limitar el tamaño de las entradas
  - No sólo en el cliente!
- ▶ Gestión robusta de errores
  - Evitar mensajes excesivamente detallados y fugas de información

# REFERENCIAS

## ■ OWASP

- ▶ <http://www.owasp.org>

## ■ Capítulo español de la OWASP

- ▶ <http://www.owasp.org/index.php/Spain>

## ■ OWASP Guide

- ▶ [http://www.owasp.org/index.php/Category:OWASP\\_Guide\\_Project](http://www.owasp.org/index.php/Category:OWASP_Guide_Project)

## ■ OWASP Testing Guide

- ▶ [http://www.owasp.org/index.php/Category:OWASP\\_Testing\\_Project](http://www.owasp.org/index.php/Category:OWASP_Testing_Project)

## ■ OWASP Top Ten

- ▶ [http://www.owasp.org/index.php/Category:OWASP\\_Top\\_Ten\\_Project](http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project)

## ■ WebScarab

- ▶ [http://www.owasp.org/index.php/Category:OWASP\\_WebScarab\\_Project](http://www.owasp.org/index.php/Category:OWASP_WebScarab_Project)

# REFERENCIAS

## ■ WebGoat

- ▶ [http://www.owasp.org/index.php/Category:OWASP\\_WebGoat\\_Project](http://www.owasp.org/index.php/Category:OWASP_WebGoat_Project)

## ■ Advanced SQL Injection in Oracle databases

- ▶ <http://www.blackhat.com/presentations/bh-usa-05/bh-us-05-fayo.pdf>

## ■ Capturing and Exploiting Hidden Mail Servers

- ▶ <http://www.webappsec.org/projects/articles/121106.pdf>

# ¡Gracias!



<http://www.owasp.org/index.php/Spain>

[vicente.aguilera@owasp.org](mailto:vicente.aguilera@owasp.org)