

**OWASP Guide 3.0**

**Building Secure Applications**

**3.0 Draft 1**

© 2007 OWASP Foundation

This document is licensed under the Creative Commons [Attribution-ShareAlike 2.5](http://creativecommons.org/licenses/by-sa/2.5/) license

# Table of Contents

Table of Contents 2

Introduction 6

About the Open Web Application Security Project 6

Security Architecture 7

Fundamentals 7

Principles 7

Defense in depth 7

Keep it simple 7

Inside, outside, partner – the weakest link will be attacked 7

Keys to the kingdom – minimize shared mechanisms 7

Complete Mediation 7

Deny all by default 7

Least Privilege 7

Separation of privilege 7

Fail safe 7

Security through obscurity never works … for long 7

Trust but verify 7

Privacy 7

Policy 7

Risk management 7

SDLC 7

Coding Standards 7

Threat modeling 7

Peer Review 7

Security Awareness 7

Security Assurance 7

Internal 7

External 7

Time between reviews 7

White box 7

Grey box 7

Black box 7

Fraud prevention 7

Initiator approver 7

Initiator approver receiver 7

Detection controls 7

Limits 7

Separation of privilege 7

User interfaces 8

Administrative interfaces 8

Help desk interfaces 8

Identity Management 8

Principles 8

Evidence of identity 8

Positive Authentication 8

Complete mediation 8

Deny all by default 8

Principle of least privilege 8

Separation of Privilege 8

Positive Access Control 8

Best Practices 8

When to use 2FA 8

SMS Authentication 8

Transaction signing 8

Certificates and PKI 8

Federated Identity 8

Trust based identity 8

Authentication 8

2FA Login Flowchart 8

Certificate Login Flowchart 8

Form based login flowchart 8

SSO Systems 8

Web Services 8

Password controls 8

Worst Practices 8

Session Management 8

Use the session manager 8

Absolute timeouts 8

Idle timeouts 8

Access control 8

Principles 8

Mandatory access control 8

Role based access control 8

Ajax and client-side controls 9

Data protection 9

Asset classification 9

Protect value transactions 9

Confidentiality 9

State management 9

Data in transit 9

Data at rest 9

Privacy 9

Sensitive information 9

Integrity 9

Minimize data retrieved from client 9

Input validation 9

Input Fields 9

Business rule validation 10

Output encoding 10

Transaction Signing 10

Accountability 10

Error handling 10

Logging 10

Auditing 10

Services 10

Files 10

Filenames from user 10

Uploads 10

Downloads 10

Reporting 10

Batch and scripts (2nd order injections) 10

ftp feeds, uploads and downloads 10

Databases 10

Prepared statements 10

Stored procedures 10

Email Best Practices 10

Displaying e-mail addresses 10

Contact us form 10

Sending large quantities of e-mail 10

Remoting 10

.NET Remoting 10

Spring Remoting 10

JNDI, JMS and others 10

XMLRPC and others 10

Web Services 10

SOAP and WS-Security\* 10

REST 10

Feeds 10

Credit cards 10

PCI Guidelines 10

Working with a cc gateway 10

Payments 10

Paypal 10

Google Adsense 10

Tracking visitors 11

WebTrends Live 11

Cross-domain XHR 11

Availability 11

Event management considerations 11

Clustering vs scale out 11

Denial of Service 11

Injections 11

CSS 11

CSRF 11

Request / Response Splitting 11

SQL Injection 11

MDB Injection 11

LDAP Injection 11

XPath / XQuery 11

XSLT… 11

Buffer Overflows 11

Race Conditions 11

Quality 11

SQA 11

Unit tests 11

Web tests 11

Integration tests 11

Fuzz tests 11

Configuration 11

Deployment 11

Security Response 11

… 11

# Introduction

Welcome to the totally re-written OWASP Guide 3.0! The OWASP Guide has been re-written to be:

* Shorter
* More applicable
* More functional

The previous Guide contained information on how to review, attack and protect code. That is no longer necessary now that the Code Review Guide and Penetration Testing Guides have been completed. This version of the Guide concentrates upon writing solid, safe and secure code. By reducing the length of the Guide to no more than 150 pages, we hope that more architects, designers, business analysts, software engineers and developers will be able to digest the new version, thus creating safe and more secure applications. There will never be a one page version of this book.

Architects and designers should digest the first section and use the remaining sections like an encyclopedia or dictionary – looking up controls as necessary. Software engineers should read the entire Guide. The reasoning behind choosing certain controls is in the first section, and the controls themselves in the remaining sections.

Those who set policy are recommended to read as much as they can – only by knowledge of what can go wrong can organizations set policy to prevent the acquisition or development of insecure software.

It is far harder to write solid code than to destroy it. Necessarily, this book contains a great deal of information. Not every application will require every control, and thus it is necessary to

# About the Open Web Application Security Project

# Security Architecture

Security architecture is not separate to the normal architecture of an application – it is an innate aspect of the architecture of even the simplest systems. Security architecture should be stable for at least two-three years in the average application.

## Fundamentals

The fundamentals of application security have not changed in over 30 years. However, there was an interregnum where these details were simply forgotten and not taught and certainly not implemented. Indeed, it is rare to find computer science and software engineering degrees with even a rudimentary security course. Thankfully, this is beginning to change. However, it will take a very long time

## Principles

### Defense in depth

### Keep it simple

### Inside, outside, partner – the weakest link will be attacked

### Keys to the kingdom – minimize shared mechanisms

### Complete Mediation

### Deny all by default

### Least Privilege

### Separation of privilege

### Fail safe

### Security through obscurity never works … for long

### Trust but verify

### Privacy

## Policy

## Risk management

Applications developed without a risk management framework are expensive and insecure. Risk management allows businesses

## SDLC

## Coding Standards

## Threat modeling

## Peer Review

## Security Awareness

## Security Assurance

### Internal

### External

### Time between reviews

### White box

### Grey box

### Black box

## Fraud prevention

### Initiator approver

### Initiator approver receiver

### Detection controls

### Limits

## Separation of privilege

### User interfaces

### Administrative interfaces

### Help desk interfaces

# Identity Management

## Principles

### Evidence of identity

### Positive Authentication

### Complete mediation

### Deny all by default

### Principle of least privilege

### Separation of Privilege

### Positive Access Control

## Best Practices

### When to use 2FA

### SMS Authentication

### Transaction signing

### Certificates and PKI

### Federated Identity

### Trust based identity

## Authentication

### 2FA Login Flowchart

### Certificate Login Flowchart

### Form based login flowchart

### SSO Systems

### Web Services

### Password controls

#### Complexity

#### Expiry

#### Soft lockout

#### Administrative lockout

#### Brute force controls

#### Change password flow chart

#### Reset password flow chart

#### Forgot username flow chart

#### Admin reset password flow chart

### Worst Practices

#### Questions and answers and other back doors

#### Send clear text passwords via e-mail

#### Keep clear text passwords

## Session Management

### Use the session manager

### Absolute timeouts

### Idle timeouts

## Access control

### Principles

### Mandatory access control

### Role based access control

#### Environment access control

#### Coarse Grained Authorization - Controller

#### Medium Grained Authorization – Business Logic

#### Fine Grained Authorization – Data Layer

#### Forced Browsing

#### Indirect reference maps

### Ajax and client-side controls

# Data protection

## Asset classification

## Protect value transactions

## Confidentiality

### State management

#### Application state

#### Access tokens

#### Hidden fields, cookies and headers

### Data in transit

#### SSL

#### XML Encryption

### Data at rest

#### Proper use of cryptography

#### Storage of keys

#### …

#### XML encryption

### Privacy

#### EU, AU, …

#### US – Safe harbor

#### Unregulated

### Sensitive information

#### Personally identifiable information (PII)

#### SSNs and other government IDs

#### Health records

#### Credit Cards

## Integrity

### Minimize data retrieved from client

### Input validation

#### All sources

#### Centralized routine

#### Enforced validation

#### Canocalization

#### Strong typing

#### Strings – syntax, length and purpose

#### Integers – Range

#### Floats

#### Booleans

### Input Fields

#### Check all fields exist

#### Text fields

#### Labels

#### Check boxes and radio buttons

#### Selects

#### Hidden fields, cookies and headers

### Business rule validation

### Output encoding

#### Setting an appropriate code page and locale

#### HTML

#### Attributes

#### URL

#### XML

#### JSON

#### LDAP

#### XPath / XQuery

#### JavaScript, VBScript et al

#### Base64

### Transaction Signing

# Accountability

## Error handling

## Logging

## Auditing

# Services

## Files

### Filenames from user

### Uploads

### Downloads

### Reporting

### Batch and scripts (2nd order injections)

### ftp feeds, uploads and downloads

## Databases

### Prepared statements

### Stored procedures

## Email Best Practices

### Displaying e-mail addresses

### Contact us form

### Sending large quantities of e-mail

## Remoting

### .NET Remoting

### Spring Remoting

### JNDI, JMS and others

### XMLRPC and others

## Web Services

### SOAP and WS-Security\*

### REST

### Feeds

## Credit cards

### PCI Guidelines

### Working with a cc gateway

## Payments

### Paypal

### Google Adsense

## Tracking visitors

### WebTrends Live

## Cross-domain XHR

# Availability

## Event management considerations

## Clustering vs scale out

## Denial of Service

## Injections

### CSS

### CSRF

### Request / Response Splitting

### SQL Injection

### MDB Injection

### LDAP Injection

### XPath / XQuery

### XSLT…

## Buffer Overflows

## Race Conditions

# Quality

## SQA

### Unit tests

### Web tests

### Integration tests

### Fuzz tests

## Configuration

## Deployment

# Security Response

## …