Secure Coding
di OWASP

Zaki Akhmad

OWASP Indonesia

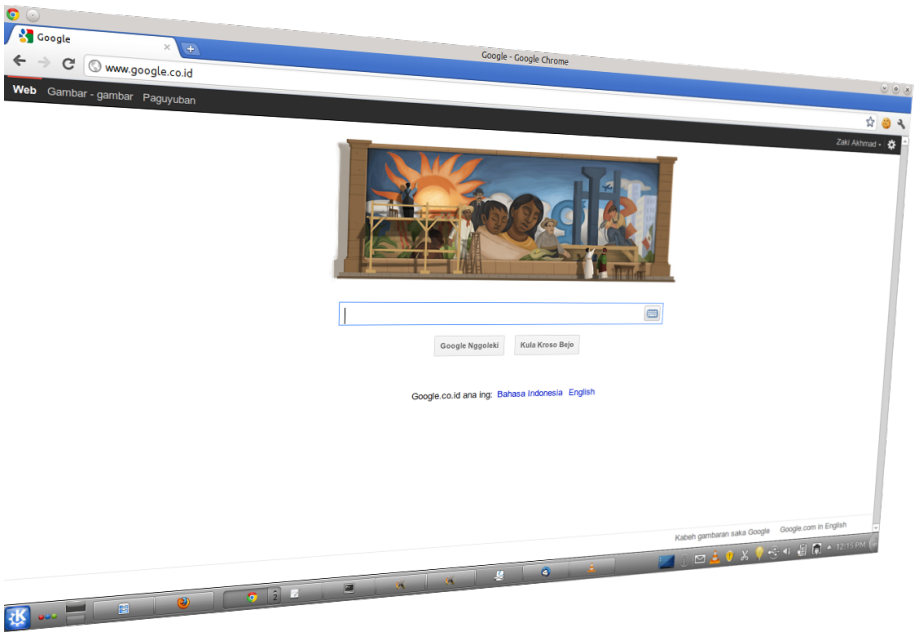
30 Mei 2012

Daftar Isi

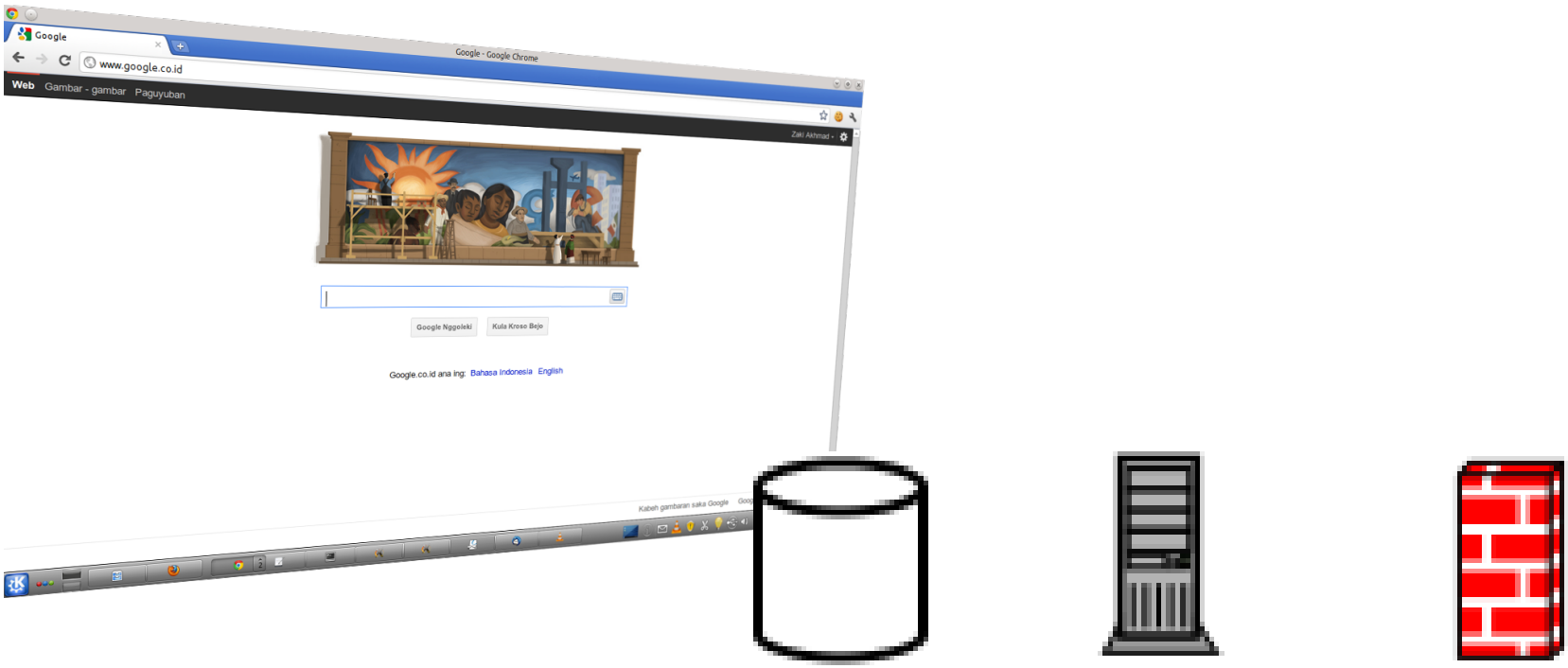
Tentang Zaki Akhmad

Surel	za@owasp.org
Twitter	@zakiakhmad
Pendidikan	
S2	Elektro ITB, 2007-2009
S1	Elektro ITB, 2001-2006
Pekerjaan	
indocisc	Analis Keamanan, 2007 - sekarang
PAU-ME	Peneliti, 2007-2009

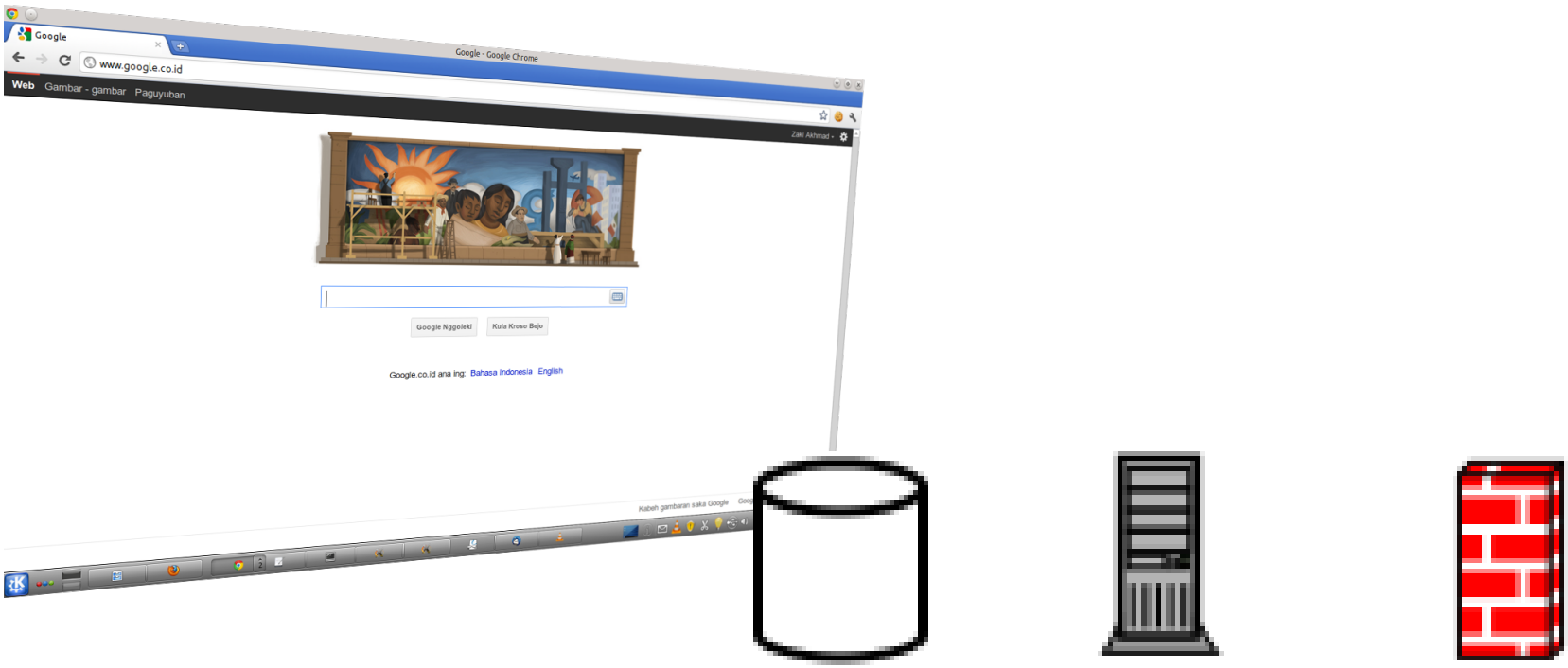
Mengapa Perlu *Secure Coding?*



Mengapa Perlu *Secure Coding?*



Mengapa Perlu *Secure Coding?*



.. . pada praktiknya firewall, IDS/IPS tidak mampu mencegah serangan *SQL injection*. Aplikasi sendiri harus aman.

OWASP

- Projects
  - Tools
  - Documentations
- Conferences
- Chapters

OWASP

- OWASP Tools
  - ZAP Proxy `http://goo.gl/Y6oWy`
  - WebGoat `http://goo.gl/9RN63`
  - GoatDroid `http://goo.gl/k7Rt4`

OWASP

- OWASP Tools
  - ZAP Proxy
    - Web application proxy

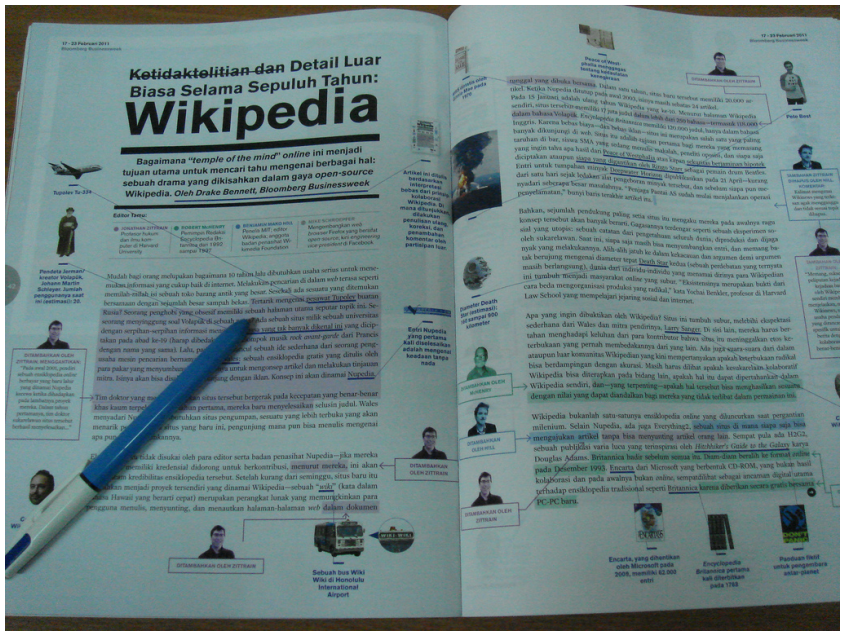
OWASP

- OWASP Tools
  - WebGoat
    - Deliberately insecure J2EE web application

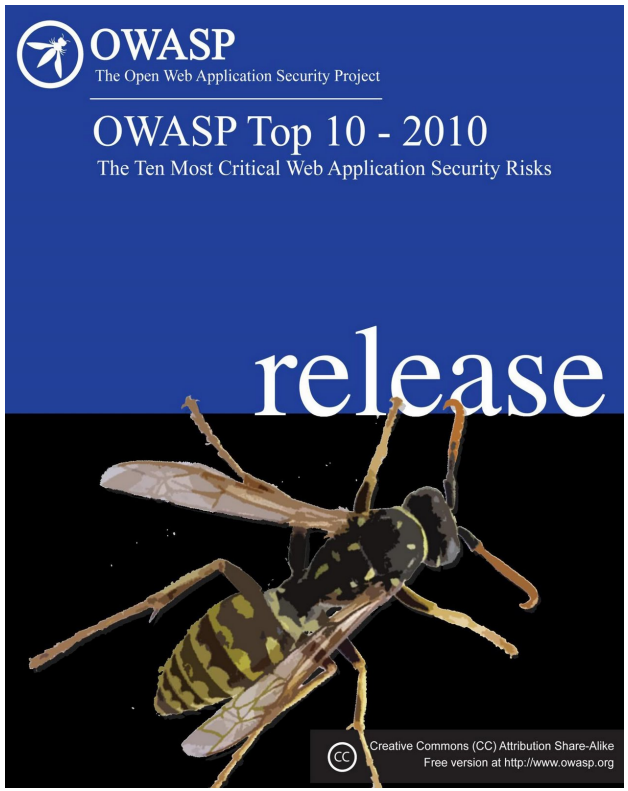
OWASP

- OWASP Tools
  - GoatDroid
    - A fully functional training environment for exploring Android mobile application security

- Dokumentasi OWASP
  - OWASP Top 10
  - OWASP Testing Guide
  - OWASP Development Guide
  - OWASP ASVS
  - ...

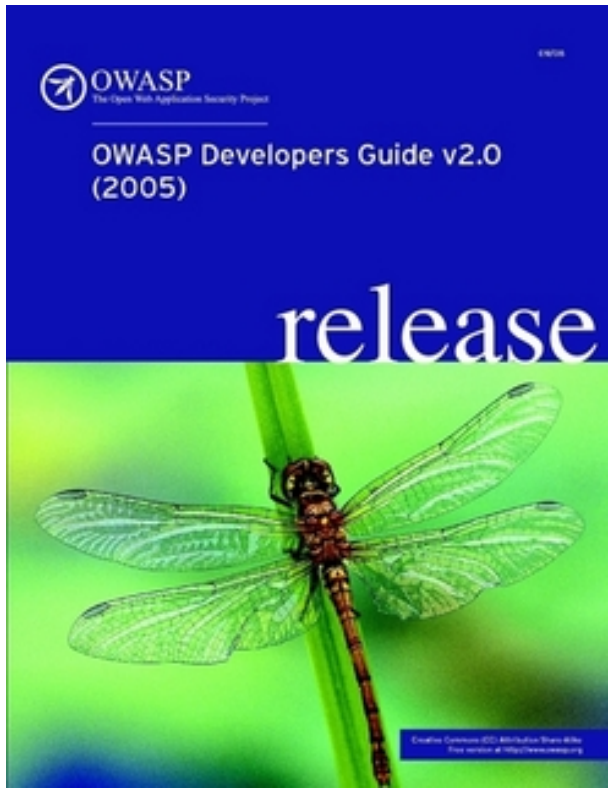


- Dokumentasi OWASP
  - OWASP Top 10



- Dokumentasi OWASP
  - OWASP Testing Guide

- Dokumentasi OWASP
  - OWASP Development Guide

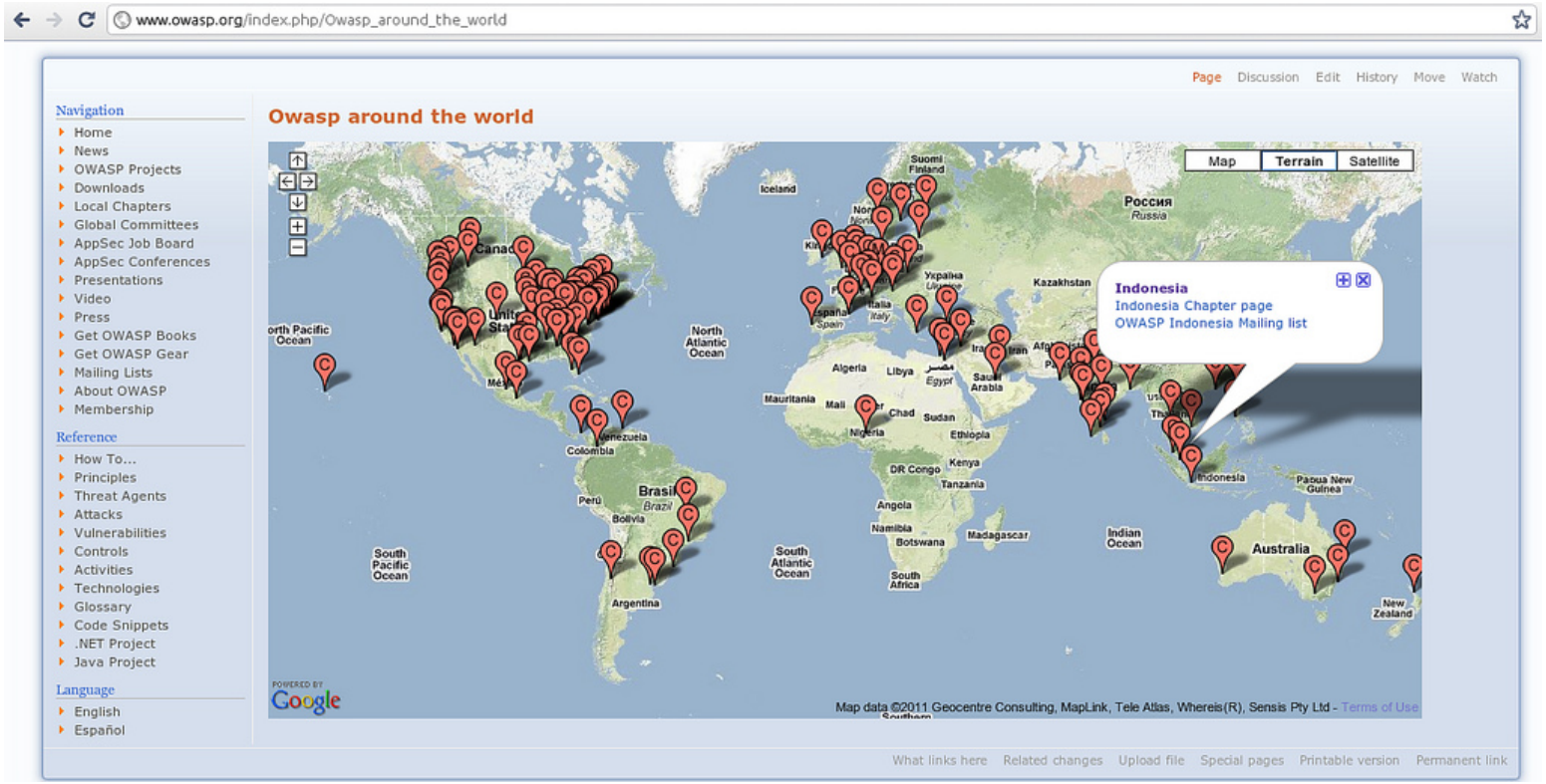


- Dokumentasi OWASP
  - OWASP ASVS

Konferensi

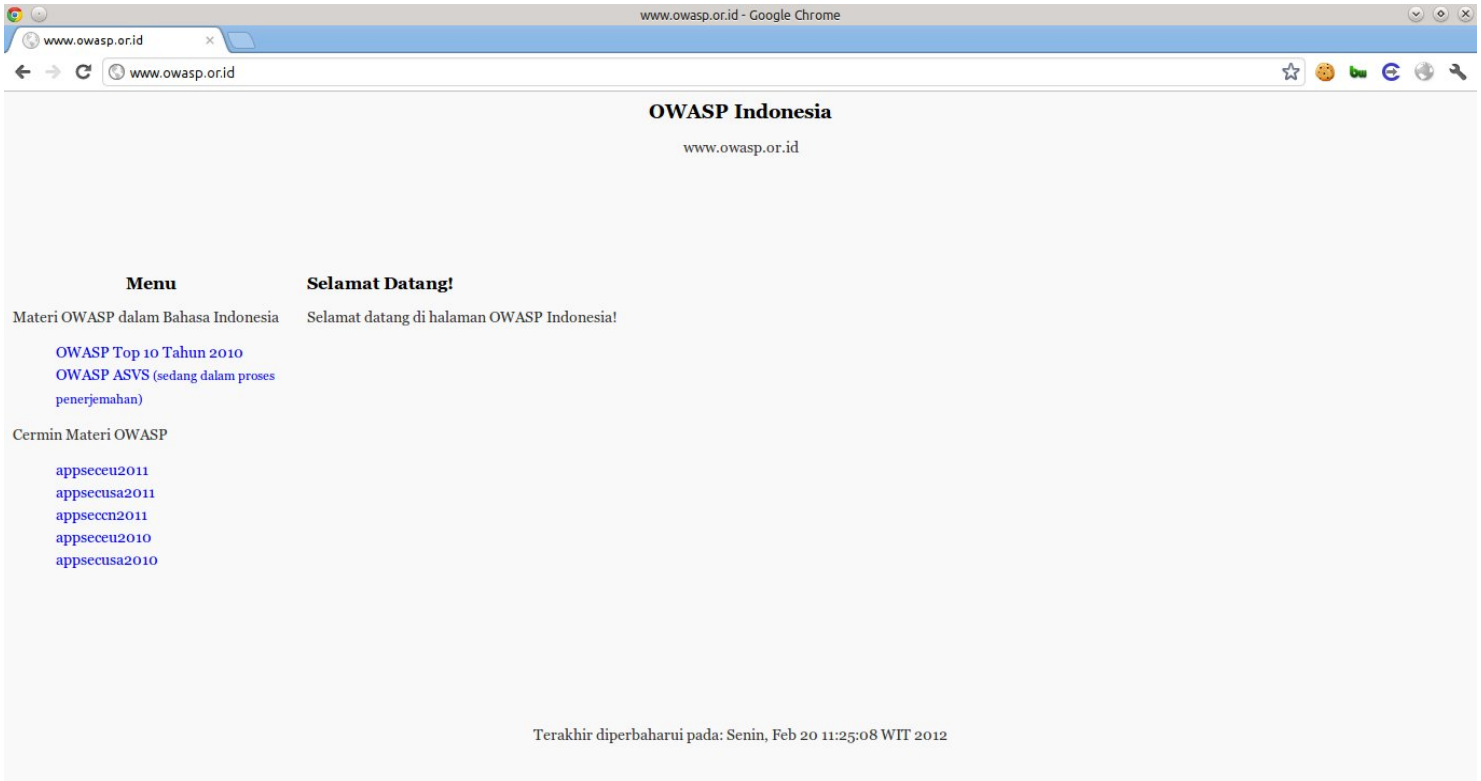
Summit, Konferensi (Asia-Pasifik, Eropa, Amerika Utara, Amerika Latin)

Chapter



Singapura, Malaysia, Korea Selatan, **Indonesia**, Jepang

OWASP Indonesia

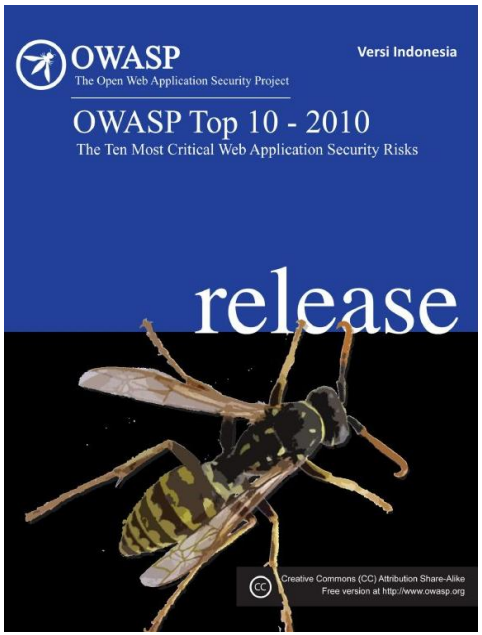


Situs www.owasp.or.id

Twitter @owaspid

Milis owasp-indonesia@lists.owasp.org

OWASP Indonesia



Top 10 - 2010

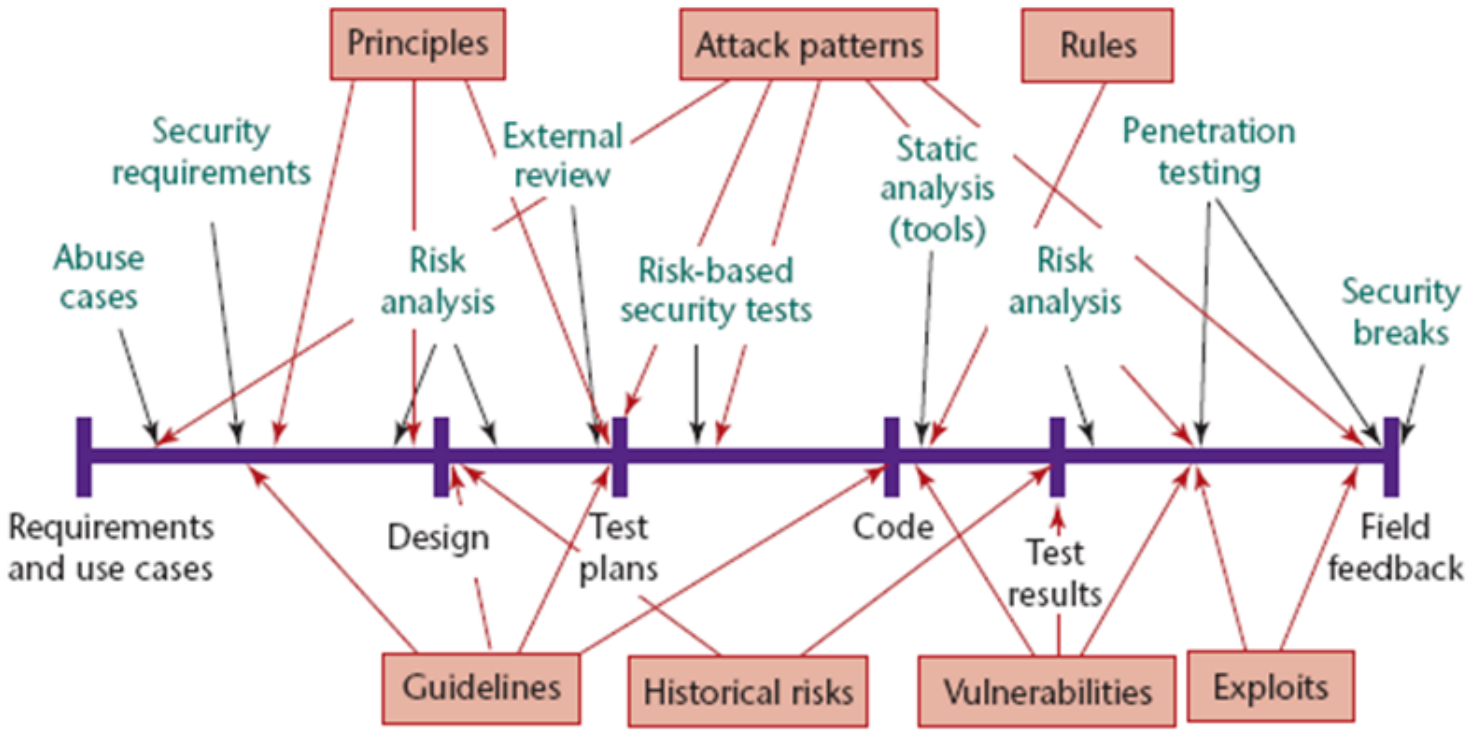
ASVS

Proyek Penerjemahan

Secure Coding Projects di OWASP

- Secure Coding Principles
- Quick Reference Guide
- ESAPI

Where is Secure Coding



Software Security, Gary McGraw

Code Snippets

```
1  <?php
2
3  if(isset($_GET['Submit'])){
4          // Retrieve data
5          $id = $_GET['id'];
6
7          $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
8          $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>' );
9
10         $num = mysql_numrows($result);
11
12     $i = 0;
13
14         while ($i < $num) {
15
16                 $first = mysql_result($result,$i,"first_name");
17                 $last = mysql_result($result,$i,"last_name");
18
19                 $html .= '<pre>';
20                 $html .= 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last
21                 $html .= '</pre>';
22
23                 $i++;
24         }
25 }
26 ?>
```

Perhatikan baris ke-5 s/d 8

Code Snippets

```
1  <?php
2
3  define( 'DVWA_WEB_PAGE_TO_ROOT', '' );
4  require_once DVWA_WEB_PAGE_TO_ROOT.'dvwa/includes/dvwaPage.inc.php';
5  dvwaPageStartup( array( 'phpids' ) );
6  dvwaDatabaseConnect();
7
8  if( isset( $_POST[ 'Login' ] ) ) {
9
10     $user = $_POST[ 'username' ];
11     $user = stripslashes( $user );
12     $user = mysql_real_escape_string( $user );
13
14     $pass = $_POST[ 'password' ];
15     $pass = stripslashes( $pass );
16     $pass = mysql_real_escape_string( $pass );
17     $pass = md5( $pass );
18
19     $qry = "SELECT * FROM `users` WHERE user='$user' AND password='$pass';";
20     $result = @mysql_query($qry) or die('<pre>' . mysql_error() . '</pre>' );
21     ...
22     <!-- <img src=\"".DVWA_WEB_PAGE_TO_ROOT."dvwa/images/RandomStorm.png\" /> -->
23
24     <p>Damn Vulnerable Web Application (DVWA) is a RandomStorm OpenSource project</p>
25     </div> <!-- end align div -->
26     </body>
27 </html>
28 ";
29 ?>
```

Perhatikan baris ke-17

OWASP Secure Coding - Quick Reference Guide

Overview

- Technology agnostic coding practices

OWASP Secure Coding - Quick Reference Guide

Overview

- Technology agnostic coding practices
- What to do, **not** how to do it

OWASP Secure Coding - Quick Reference Guide

Overview

- Technology agnostic coding practices
- What to do, **not** how to do it
- Compact, but comprehensive checklist format

OWASP Secure Coding - Quick Reference Guide

Overview

- Technology agnostic coding practices
- What to do, **not** how to do it
- Compact, but comprehensive checklist format
- Focuses on secure coding requirements, rather than on vulnerabilities and exploits

OWASP Secure Coding - Quick Reference Guide

Overview

- Technology agnostic coding practices
- What to do, **not** how to do it
- Compact, but comprehensive checklist format
- Focuses on secure coding requirements, rather than on vulnerabilities and exploits
- Includes a cross-referenced glossary to get **developers and security folks** talking the same language

OWASP Secure Coding - Quick Reference Guide

Daftar Isi

OWASP Secure Coding- Quick Reference Guide

Penggunaan

1. Sebagai dokumen panduan dalam pengembangan
2. Sebagai dokumen pendukung SDLC
3. Sebagai dokumen *requirement* dalam *outsource*
    1. Identifikasi *security requirement* dalam proyek pengembangan
    2. Masukkan ke dalam RFP dan Kontrak

OWASP Secure Coding - Quick Reference Guide

No	Secure Coding Practices Checklist
1	Input Validation
2	Output Encoding
3	Authentication and Password Management
4	Session Management
5	Access Control
6	Cryptographic Practices
7	Error Handling and Logging
8	Data Protection
9	Communication Security
10	System Configuration
11	Database Security
12	File Management
13	Memory Management
14	General Coding Practices

OWASP Secure Coding - Quick Reference Guide

Input Validation

- If any potentially hazardous characters must be allowed as input, be sure that you implement additional controls like output encoding, secure task specific APIs and accounting for the utilization of that data throughout the application. Examples of common hazardous characters include:

  `< > " ’ ( ) & + \ \’ \"u`

OWASP Secure Coding - Quick Reference Guide

Authentication and Password Management

- If your application manages a credential store, it should ensure that only crytograpically strong one-way salted hashes of passwords are stored and that the table/file that stores the password and keys is write-able only by the application. (Do not use the MD5 algorithm if it can be avoided)
- Use only HTTP POST request to transmit authentication credentials

OWASP Secure Coding - Quick Reference Guide

Error Handling and Logging

- Do not disclose sensitive information in error responses, including system details, session identifiers or account information
- Use error handlers that do not display debugging or stack trace information
- Implement generic error messages and use custom error pages

Referensi/Bacaan Lanjut

- Gary McGraw, Software Security
- Mozilla Secure Coding Guidelines
- OWASP Secure Coding Practices, Quick Reference Guide
- Damn Vulnerable Web Application

Terima Kasih

hatur nuhun, matur suwun,
thank you, arigatou, danke, merci beaucoup

foto-foto flickr.com/zakiakhmad