OWASP Slovenija – predstavitev - ISACA

Stanka Šalamun
Vodja odseka OWASP Slovenija
ACROS d.o.o.

owasp-slovenia-owner@lists.owasp.org

OWASP
Ljubljana, 25.2.2010

The OWASP Foundation
http://www.owasp.org

OWASP Slovenija se zahvaljuje

podjetju HERMES SoftLab,

www.hermes-softlab.com

ki gosti današnje srečanje.

Open Web Application Security Project (OWASP)

OWASP: “The free and open application security community”

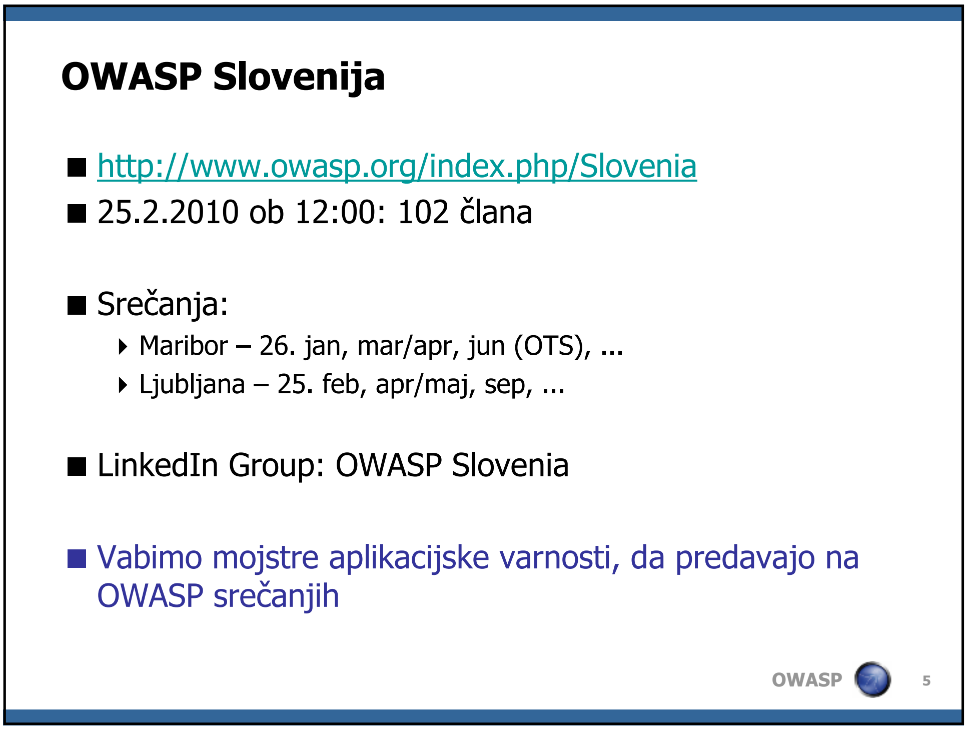
- Obstaja od leta 2000
- Neprofitno, prostovoljno, nekomercialno
- Aktivnosti: srečanja, izobraževalni projekti, konference, knjige, promocija znanja o aplikacijski varnosti
- Licence: Creative Commons 3.0, BSD

Članstvo

- Člani podporniki: okrog 10.000
- Člani z glasovalno pravico (ki plačajo članarino): okrog 750
- Število odsekov: okrog 150
- Komercialni podporniki: 28

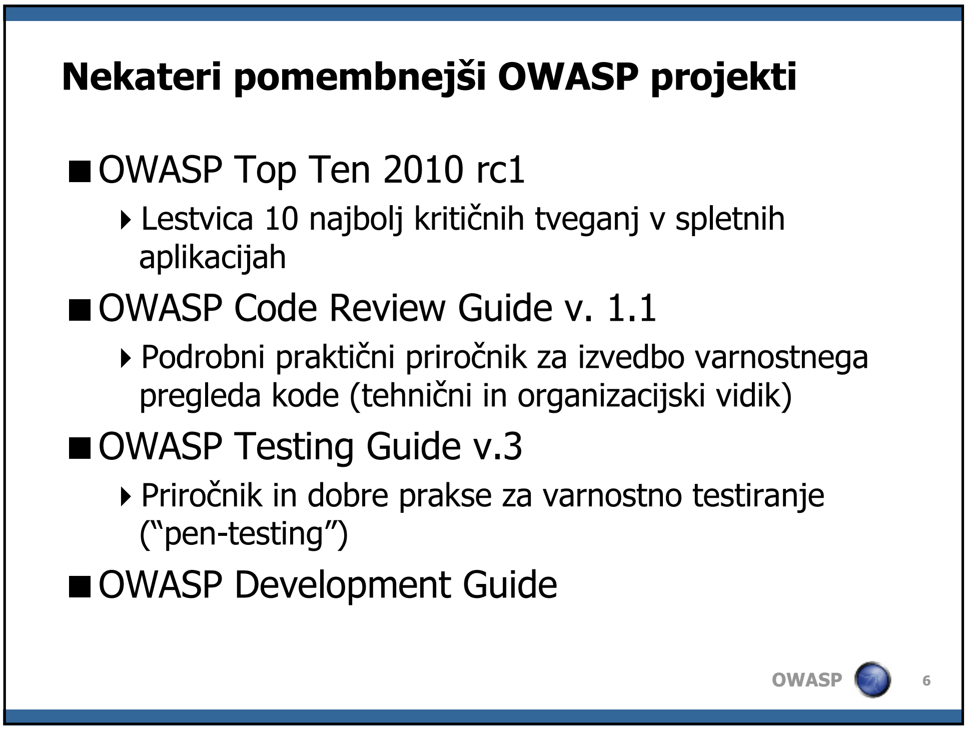
  Nokia, HP, Symantec, (ISC^2), Breach, Aspect Security, Cigital, Adobe...
- Akreditirani univerzitetne podpornice: 14

  UCLA, George Washington University, Universidad de Buenos Aires ...
- Projekt v teku: akreditirane vladne ustanove



OWASP Slovenija

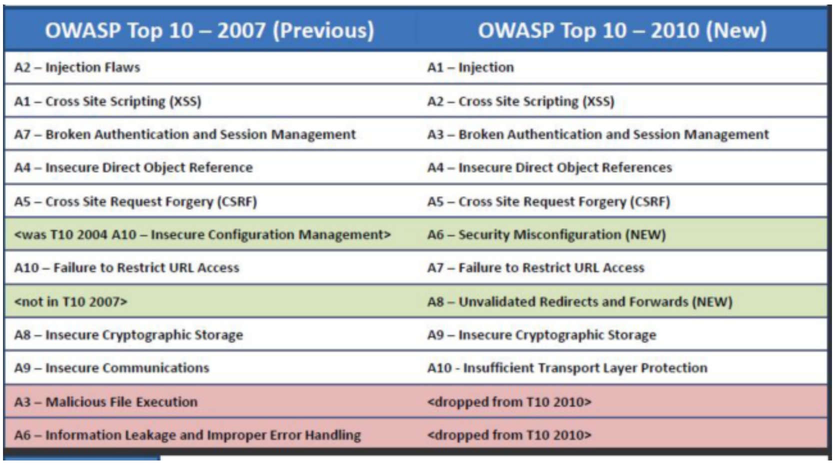
- http://www.owasp.org/index.php/Slovenia
- 25.2.2010 ob 12:00: 102 člana
- Srečanja:
  - Maribor – 26. jan, mar/apr, jun (OTS), ...
  - Ljubljana – 25. feb, apr/maj, sep, ...
- LinkedIn Group: OWASP Slovenia
- Vabimo mojstre aplikacijske varnosti, da predavajo na OWASP srečanjih



Nekateri pomembnejši OWASP projekti

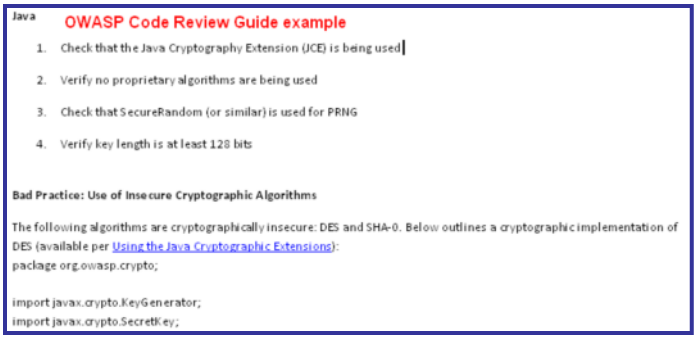
- OWASP Top Ten 2010 rc1
  - Lestvica 10 najbolj kritičnih tveganj v spletnih aplikacijah
- OWASP Code Review Guide v. 1.1
  - Podrobni praktični priročnik za izvedbo varnostnega pregleda kode (tehnični in organizacijski vidik)
- OWASP Testing Guide v.3
  - Priročnik in dobre prakse za varnostno testiranje (“pen-testing”)
- OWASP Development Guide

Nekateri pomembnejši OWASP projekti – OWASP TOP 10



OWASP Top 10 – 2007 (Previous)	OWASP Top 10 – 2010 (New)
A2 – Injection Flaws	A1 – Injection
A1 – Cross Site Scripting (XSS)	A2 – Cross Site Scripting (XSS)
A7 – Broken Authentication and Session Management	A3 – Broken Authentication and Session Management
A4 – Insecure Direct Object Reference	A4 – Insecure Direct Object References
A5 – Cross Site Request Forgery (CSRF)	A5 – Cross Site Request Forgery (CSRF)
<was T10 2004 A10 – Insecure Configuration Management>	A6 – Security Misconfiguration (NEW)
A10 – Failure to Restrict URL Access	A7 – Failure to Restrict URL Access
<not in T10 2007>	A8 – Unvalidated Redirects and Forwards (NEW)
A8 – Insecure Cryptographic Storage	A9 – Insecure Cryptographic Storage
A9 – Insecure Communications	A10 - Insufficient Transport Layer Protection
A3 – Malicious File Execution	<dropped from T10 2010>
A6 – Information Leakage and Improper Error Handling	<dropped from T10 2010>

Nekateri pomembnejši OWASP projekti – OWASP Code Review Guide



Java

OWASP Code Review Guide example

1. Check that the Java Cryptography Extension (JCE) is being used
2. Verify no proprietary algorithms are being used
3. Check that SecureRandom (or similar) is used for PRNG
4. Verify key length is at least 128 bits

Bad Practice: Use of Insecure Cryptographic Algorithms

The following algorithms are cryptographically insecure: DES and SHA-0. Below outlines a cryptographic implementation of DES (available per Using the Java Cryptographic Extensions):

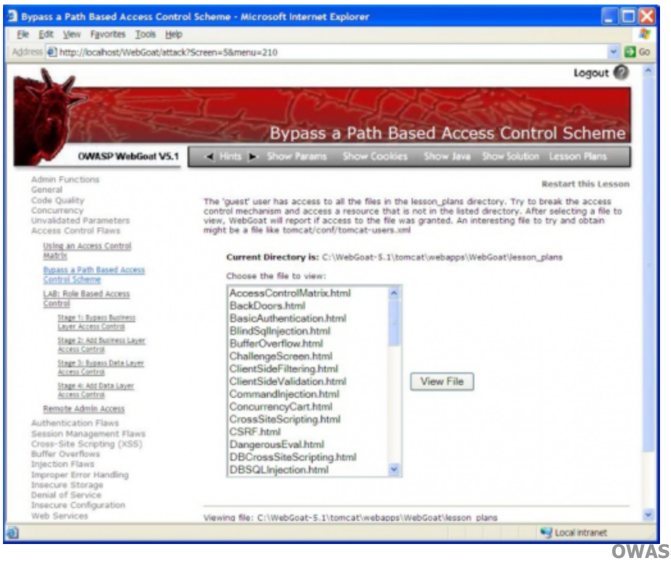
```
package org.owasp.crypto;

import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
```

Nekateri pomembnejši OWASP projekti (2)

- WebGoat 5.2– namerno “nevarna” spletna aplikacija za učenje po lekcijah
- WebScarab – orodje za preizkušanje spletnih aplikacij in spletnih storitev
- ESAPI – Enterprise Security API – nabor programskih knjižnic z varnostnimi funkcijami

  (Java, .NET, ASP, PHP, ColdFusion/CFML, Python, JS)
- ASVS - Application Security Verification Standard
- OpenSAMM – model za aplikacijsko zrelost v organizacijah

Nekateri pomembnejši OWASP projekti (2) - WebGoat


Bypass a Path Based Access Control Scheme - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost/WebGoat/attack?Screen=5&menu=210
Go
Logout
Bypass a Path Based Access Control Scheme
OWASP WebGoat V5.1
Hints
Show Params
Show Cookies
Show Java
Show Solution
Lesson Plans
Admin Functions
General
Code Quality
Concurrency
Unvalidated Parameters
Access Control Flaws
Using an Access Control Matrix
Bypass a Path Based Access Control Scheme
LAB: Role Based Access Control
Stage 1: Bypass Business Layer Access Control
Stage 2: Add Business Layer Access Control
Stage 3: Bypass Data Layer Access Control
Stage 4: Add Data Layer Access Control
Remote Admin Access
Authentication Flaws
Session Management Flaws
Cross-Site Scripting (XSS)
Buffer Overflows
Injection Flaws
Improper Error Handling
Insecure Storage
Denial of Service
Insecure Configuration
Web Services
Restart this Lesson
View File
Local intranet

The 'guest' user has access to all the files in the lesson_plans directory. Try to break the access control mechanism and access a resource that is not in the listed directory. After selecting a file to view, WebGoat will report if access to the file was granted. An interesting file to try and obtain might be a file like tomcat/conf/tomcat-users.xml

Current Directory is: C:\WebGoat-5.1\tomcat\webapps\WebGoat\lesson_plans

Choose the file to view:

AccessControlMatrix.html
BackDoors.html
BasicAuthentication.html
BlindSqlInjection.html
BufferOverflow.html
ChallengeScreen.html
ClientSideFiltering.html
ClientSideValidation.html
CommandInjection.html
ConcurrencyCart.html
CrossSiteScripting.html
CSRF.html
DangerousEval.html
DBCrossSiteScripting.html
DBSQLInjection.html

Viewing file: C:\WebGoat-5.1\tomcat\webapps\WebGoat\lesson_plans

Nekateri pomembnejši OWASP projekti (2) - ASVS

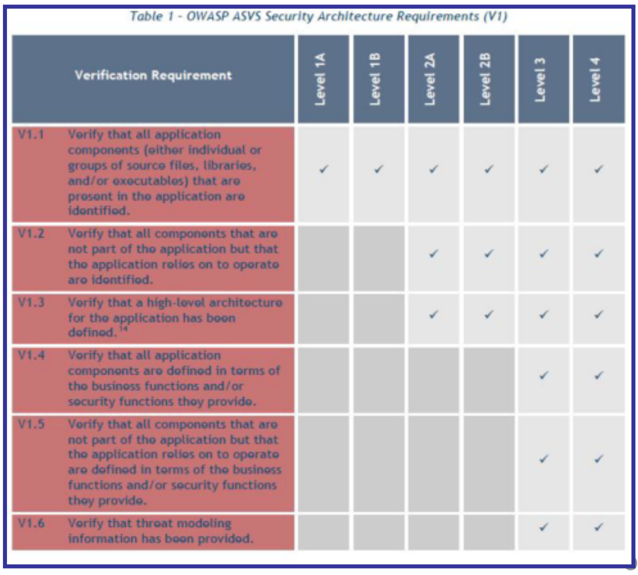
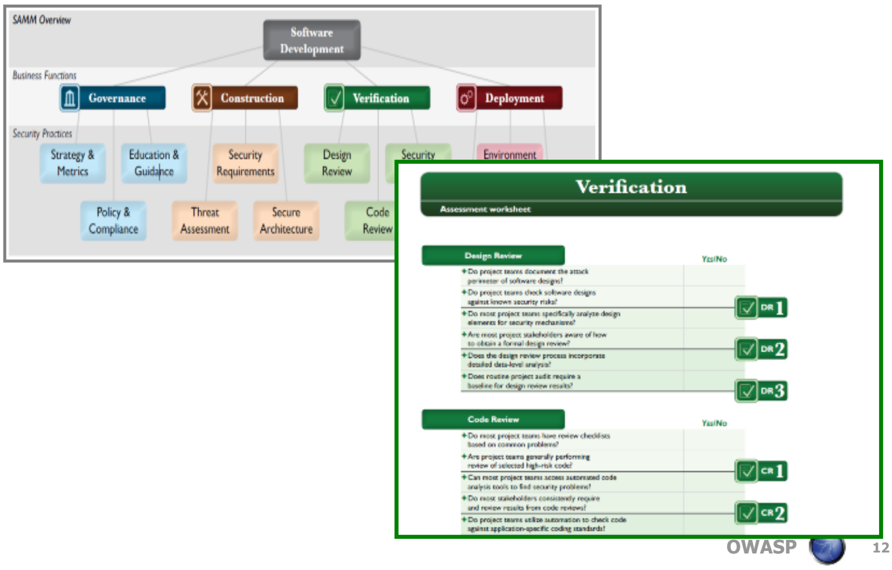


Table 1 - OWASP ASVS Security Architecture Requirements (V1)

Verification Requirement		Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V1.1	Verify that all application components (either individual or groups of source files, libraries, and/or executables) that are present in the application are identified.	✓	✓	✓	✓	✓	✓
V1.2	Verify that all components that are not part of the application but that the application relies on to operate are identified.			✓	✓	✓	✓
V1.3	Verify that a high-level architecture for the application has been defined. [14]			✓	✓	✓	✓
V1.4	Verify that all application components are defined in terms of the business functions and/or security functions they provide.					✓	✓
V1.5	Verify that all components that are not part of the application but that the application relies on to operate are defined in terms of the business functions and/or security functions they provide.					✓	✓
V1.6	Verify that threat modeling information has been provided.					✓	✓

Nekateri pomembnejši OWASP projekti (2) - OpenSAMM



Verification

Assessment worksheet

Design Review — Yes/No

- Do project teams document the attack perimeter of software designs?
- Do project teams check software designs against known security risks?

DR 1

- Do most project teams specifically analyze design elements for security mechanisms?
- Are most project stakeholders aware of how to obtain a formal design review?

DR 2

- Does the design review process incorporate detailed data-level analysis?
- Does routine project audit require a baseline for design review results?

DR 3

Code Review — Yes/No

- Do most project teams have review checklists based on common problems?
- Are project teams generally performing review of selected high-risk code?

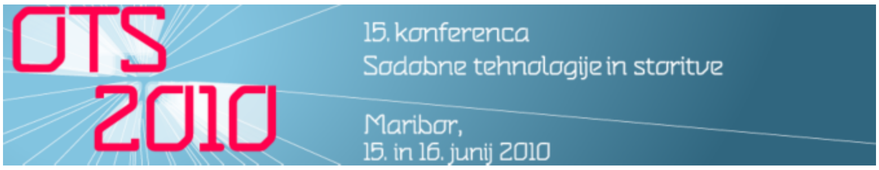
CR 1

- Can most project teams access automated code analysis tools to find security problems?
- Do most stakeholders consistently require and review results from code reviews?

CR 2

- Do project teams utilize automation to check code against application-specific coding standards?

Sodelovanje z OTS 2010



- 16.6.2010 – OWASP sekcija na konferenci OTS v Mariboru (http://cot.uni-mb.si/ots2010/)
- Vabilo k sodelovanju s prispevkom
- 31. 3. 2010: Rok za prijavo prispevka – oddajo razširjenega povzetka
- 4-8 strani članka, predstavitev na konferenci
- Tema: aplikacijska varnost

Nazaj na današnji vozni red ...

- Kratka predstavitev OWASP

- Tadej Vodopivec, HERMES SoftLab: **Kdo pije, kdo plača... za varnost spletnih aplikacij?**

- Luka Treiber, ACROS: **Izpoved "white hat" hekerja: Kako sem dobil tisto dragoceno datoteko z vašega računalnika** (demonstracija, ponovitev)