Bezpieczeństwo aplikacji webowych - standardy, przewodniki i narzędzia OWASP

Wojciech Dworakowski
OWASP Poland Chapter Leader
SecuRing
wojciech.dworakowski@owasp.org

OWASP
2012-04-19

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

Login

Wojciech Dworakowski

- Od 2003 - SecuRing – współwłaściciel
  - Zarządzanie zespołem testującym bezpieczeństwo aplikacji i systemów IT
- Od 2011 – OWASP Poland Chapter Leader

OWASP - wprowadzenie

Open Web Application Security Project

Misja: Poprawa stanu bezpieczeństwa aplikacji

> *„Make application security visible so that people and organizations can make informed decisions about true application security risk”*

- Projekty – dokumentacja, narzędzia
- Edukacja
- Współpraca (rządy, inne organizacje, twórcy standardów)

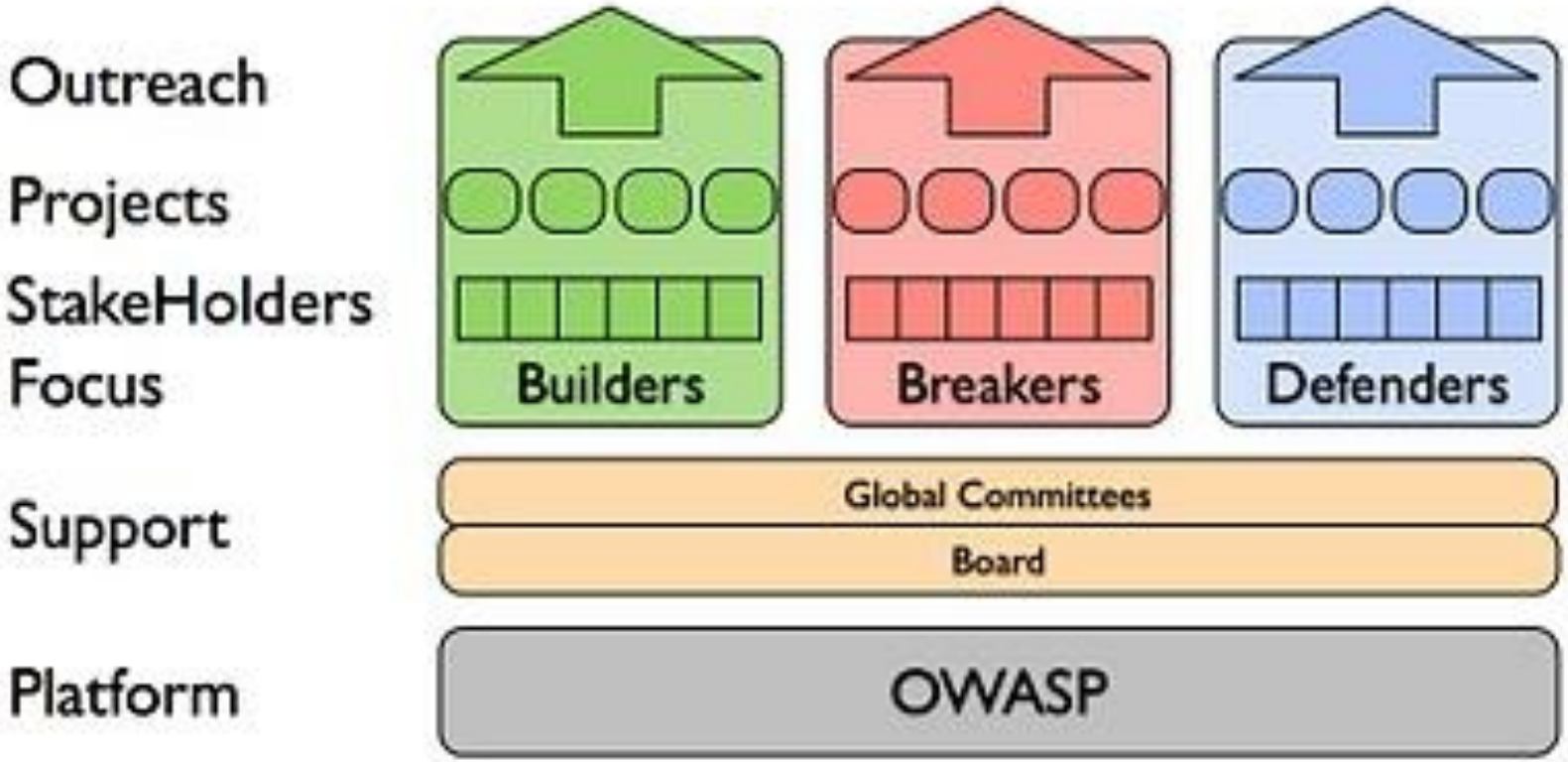
O = Open

- Organizacja non-profit
- Wszystkie materiały i narzędzia – darmowe
- Uczestnictwo w spotkaniach lokalnych – nieodpłatnie

10 lat OWASP

- Ludzie
  - 100+ Local Chapters
  - 1500+ OWASP Members
  - 20000+ uczestników spotkań
- Narzędzia i dokumentacja
  - 15 tys. pobrań / miesiąc
  - 30 tys. odwiedzających / mies
- Konferencje
  - Kilka konferencji w roku (7 w 2011)
  - Kilkuset uczestników na każdej z nich

A Vision for OWASP



Aplikacje webowe - Pięta achillesowa współczesnych systemów IT

Większość aplikacji webowych posiada istotne podatności (o znacznym wpływie na ryzyko)

Threat rank	N of Vulns	N of Sites	N of Sites	% Sites
Urgent	8918	2287	**9.14%**	18.77%
Critical	44669	5511	**45.79%**	45.22%
High	35375	8807	**36.26%**	72.27%
Medium	4908	4455	**5.03%**	36.56%
Low	3663	3618	**3.75%**	29.69%

24678 aplikacji przebadanych metodami testu penetracyjnego black-box, white-box oraz skanerami automatycznymi w roku 2008 (8 różnych firm)

Źródło: WASC Web Application Secuirty Statistics
http://projects.webappsec.org/Web-Application-Security-Statistics

Z własnych doświadczeń

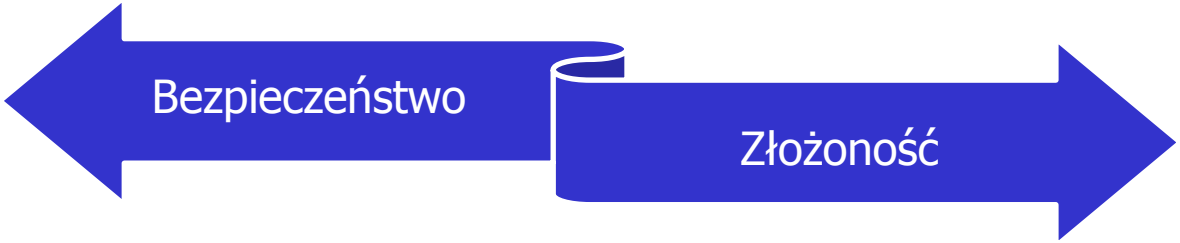
- Testy najczęściej są zamawiane tuż przed wdrożeniem produkcyjnym
  - W większości przypadków tylko testy penetracyjne
  - Często jest to jedyna forma weryfikacji bezpieczeństwa
- Kluczowe podatności znajdujemy w ok. 70% badanych aplikacji

Źródła problemów

- Stosunkowo młoda dziedzina oprogramowania bazująca na „starych” technologiach
  - HTTP/HTML nie były projektowane jako technologie do obsługi aplikacji. Nie tworzono ich z myślą o bezpieczeństwie
- Tradycyjne środki zabezpieczające (firewall, IDS) nie sprawdzają się

Źródła problemów

- Aplikacje tworzy się co raz łatwiej
  - Maleje „time to market”
  - Rośnie funkcjonalność (złożoność) ➔ Maleje bezpieczeństwo



„Wishful thinking”

Zamawiający

- Wykonawcą jest doświadczona firma, z pewnością wiedzą co robią
- Ich oprogramowania używają duże firmy – oni nie pozwoliliby sobie na niską jakość
- Testy bezpieczeństwa zaplanujemy N dni wstecz od wdrożenia produkcyjnego / pilotażowego
- Raczej nie będzie żadnych opóźnień

Wykonawca

- Zatrudniamy doświadczonych programistów, z pewnością wiedzą co robią
- Nasze nowoczesne narzędzia (famework, biblioteki) nie pozwolą na wykorzystanie ewentualnych niedoskonałości
- Nie otrzymaliśmy żadnych szczegółowych wytycznych – Pewnie ryzyko będzie ograniczone innymi metodami

Jak radzić sobie z tymi problemami?

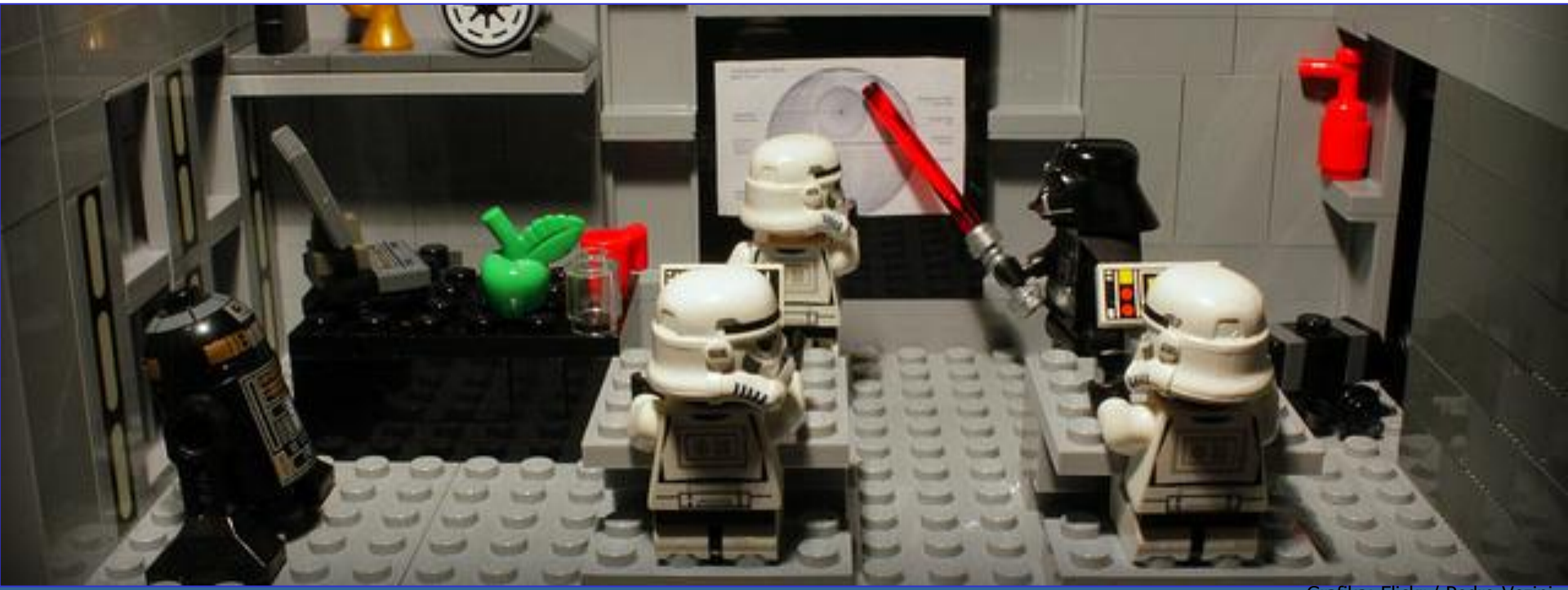
- Programiści
- Administratorzy
- Testujący bezpieczeństwo
- Szefowie projektów / zespołów

Grafika: Flickr / lucidtech

Wszyscy → Edukacja

Bez podstawowej znajomości istoty problemów nie da się

- Zaprojektować bezpiecznej aplikacji
- Napisać bezpiecznego kodu
- Dodać zabezpieczenia na poziomie sieci / serwerów



Grafika: Flickr / Pedro Vezini

OWASP Appsec Tutorial Video

http://www.youtube.com/user/AppsecTutorialSeries

- Episode 1: Appsec Basics
- Episode 2: SQL Injection
- Episode 3: Cross Site Scripting (XSS)

OWASP Top 10

https://www.owasp.org/index.php/Top_10_2010

A1 - Injection Flaws

A2 - Cross Site Scripting (XSS)

A3 - Broken Authentication and Session Management

A4 - Insecure Direct Object Reference

A5 - Cross Site Request Forgery (CSRF)

A6 – Security Misconfiguration

A7 - Failure to Restrict URL Access

A8 – Unvalidated Redirects and Forwards

A9 - Insecure Cryptographic Storage

A9 – Insufficient Transport Layer Protection

OWASP WebGoat

https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project

- Poligon doświadczalny
- Platforma treningowa w postaci aplikacji J2EE
- Zawiera przekrój błędnie zaimplementowanych:
  - popularnych fragmentów aplikacji webowych
  - mechanizmów bezpieczeństwa
- Każda podatność jest szczegółowo opisana
  - Uczenie + praktyka

OWASP podcast

https://www.owasp.org/index.php/OWASP_Podcast

Programiści

- Skąd brać rzetelne informacje o sposobach bezpiecznego programowania?
- Implementacja zabezpieczeń zabiera czas
- Czy sam będę w stanie poprawnie zaprogramować funkcje bezpieczeństwa?

Grafika: Flickr / georgivar

OWASP Secure Coding Practices - Quick Reference Guide

- Zestaw ogólnych, niezależnych od stosowanej technologii zasad dobrej praktyki
  - W formie listy kontrolnej
  - Wymagania a nie podatności i metody ataku
- Wdrożenie tych praktyk zapobiegnie większości powszechnie występujących podatności
- Tylko 17 stron

https://www.owasp.org/index.php/OWASP_Secure_Coding_Practices_-_Quick_Reference_Guide

OWASP Development Guide

https://www.owasp.org/index.php/OWASP_Guide_Project

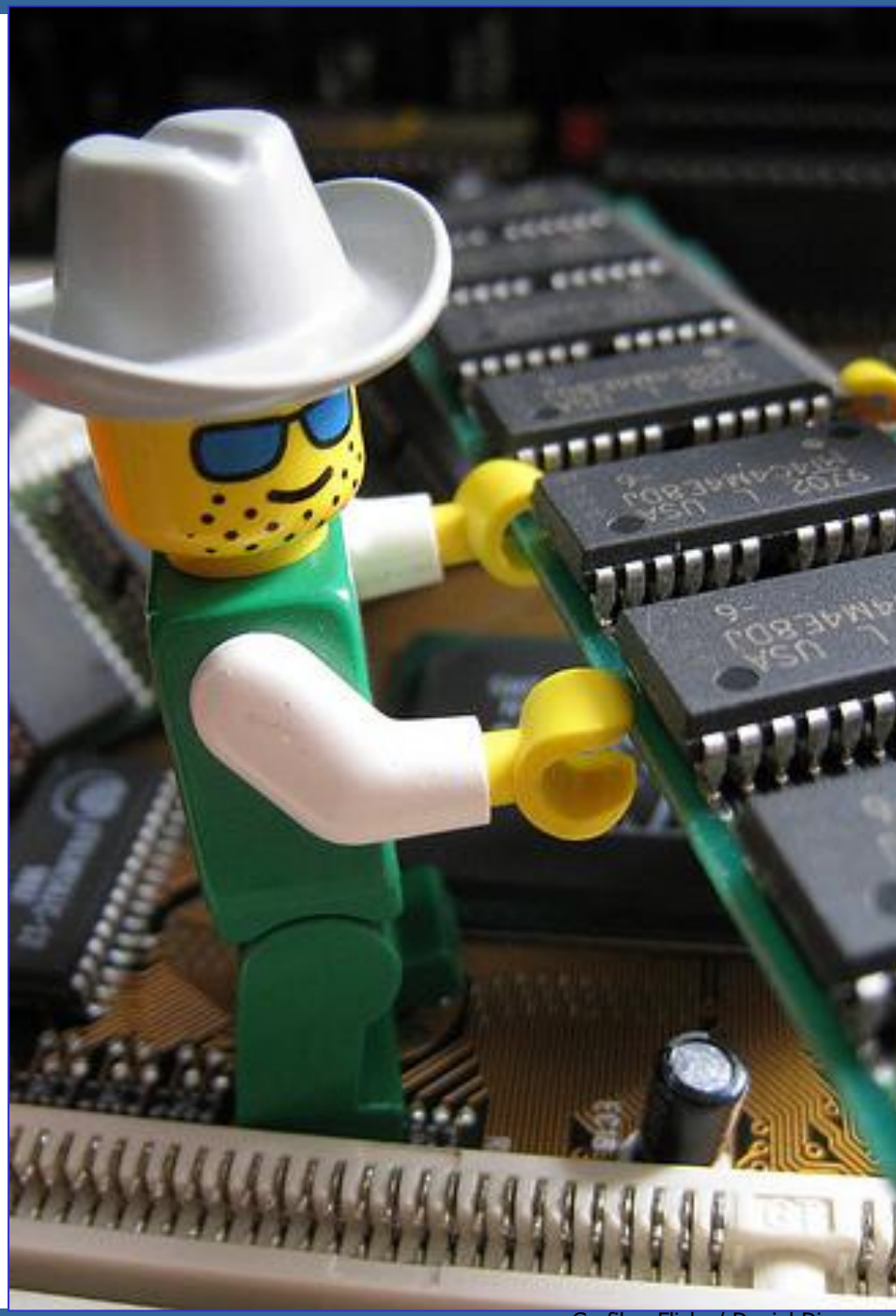
- Wyczerpująca instrukcja do projektowania, wytwarzania i wdrażania bezpiecznych aplikacji web
  - ~300 stron
- Dla architektów, developerów, konsultantów, audytorów
- Obecnie w trakcie aktualizacji

OWASP ESAPI
Enterprise Security API

- Problemy:
  - Brak czasu na bezpieczeństwo aplikacji
  - Brak wiedzy wśród programistów
  - Brak spójnej implementacji zabezpieczeń
  - Rozwijanie wielu aplikacji w wielu językach
- ESAPI to API dla kilku języków (J2EE, .NET, ASP, PHP, Python, …), które rozwiązuje te problemy i jest stworzone przez ekspertów
- https://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API

Administratorzy

- Tradycyjne firewalle i IPS-y są na tyle dobre na ile dobre są ich sygnatury
- Nie mam czasu na pisanie i aktualizowanie sygnatur
- Nawet najlepszy IPS wychwyci tylko bardzo ograniczony zbiór typów ataków



Grafika: Flickr / Daniel Dionne

ModSecurity Core Rule Set Project

- ModSecurity - web application firewall engine
- Potrzebne dobre reguły filtrowania

https://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Project

OWASP AppSensor

https://www.owasp.org/index.php/OWASP_AppSensor_Project

- Application Based Intrusion Detection
- Wykrywa i odpowiada na ataki
- WAF ale „wszyty” w kod aplikacji
- Ponad 50 „detection points” / 15 sposobów reakcji
- Wymaga „wszycia” przez programistów

Testujący bezpieczeństwo

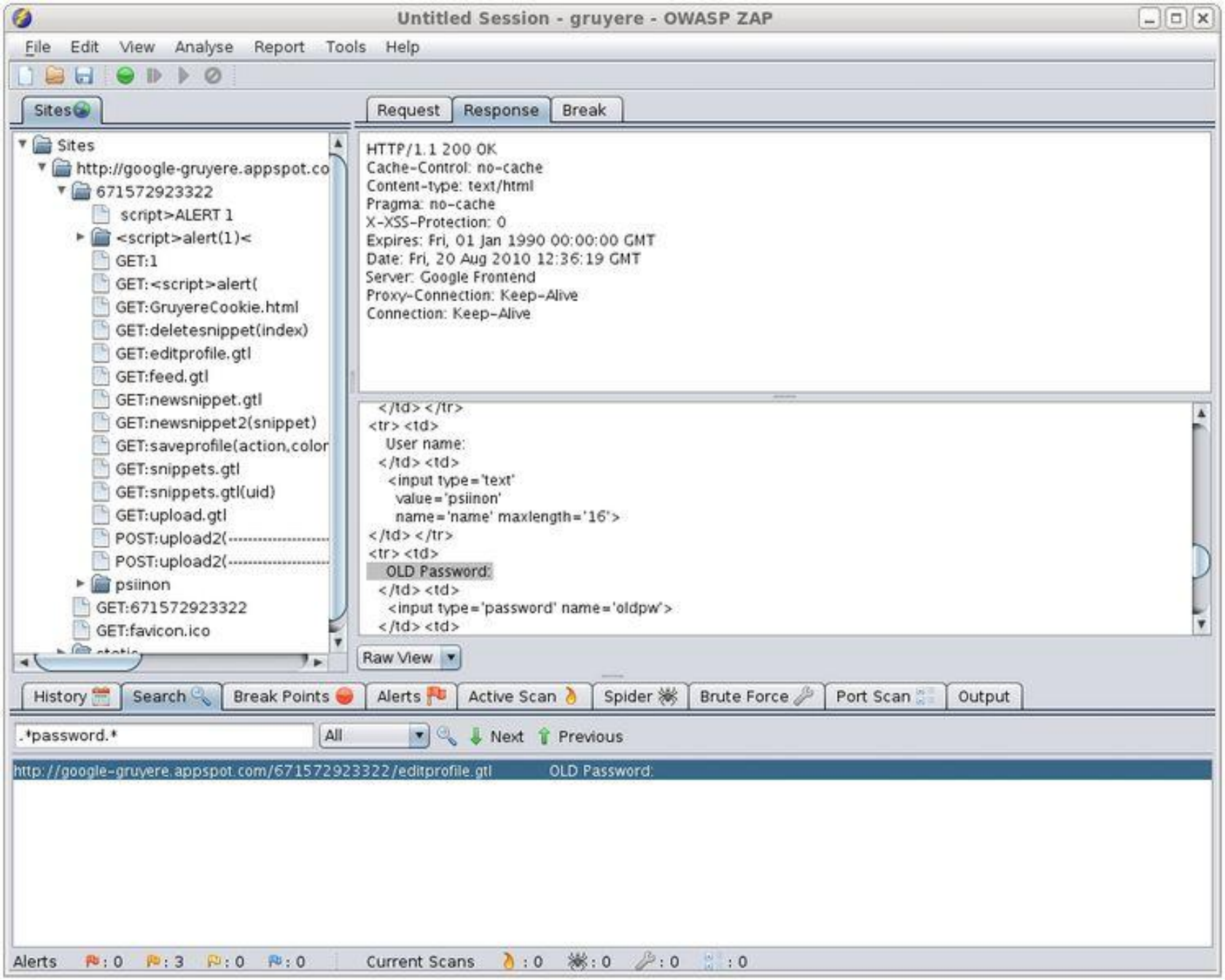
- Jakie metody testu/ataku zastosować w konkretnym przypadku?
- Czy testy które stosuje są wyczerpujące?
- Narzędzia

Grafika: Flickr / quimby

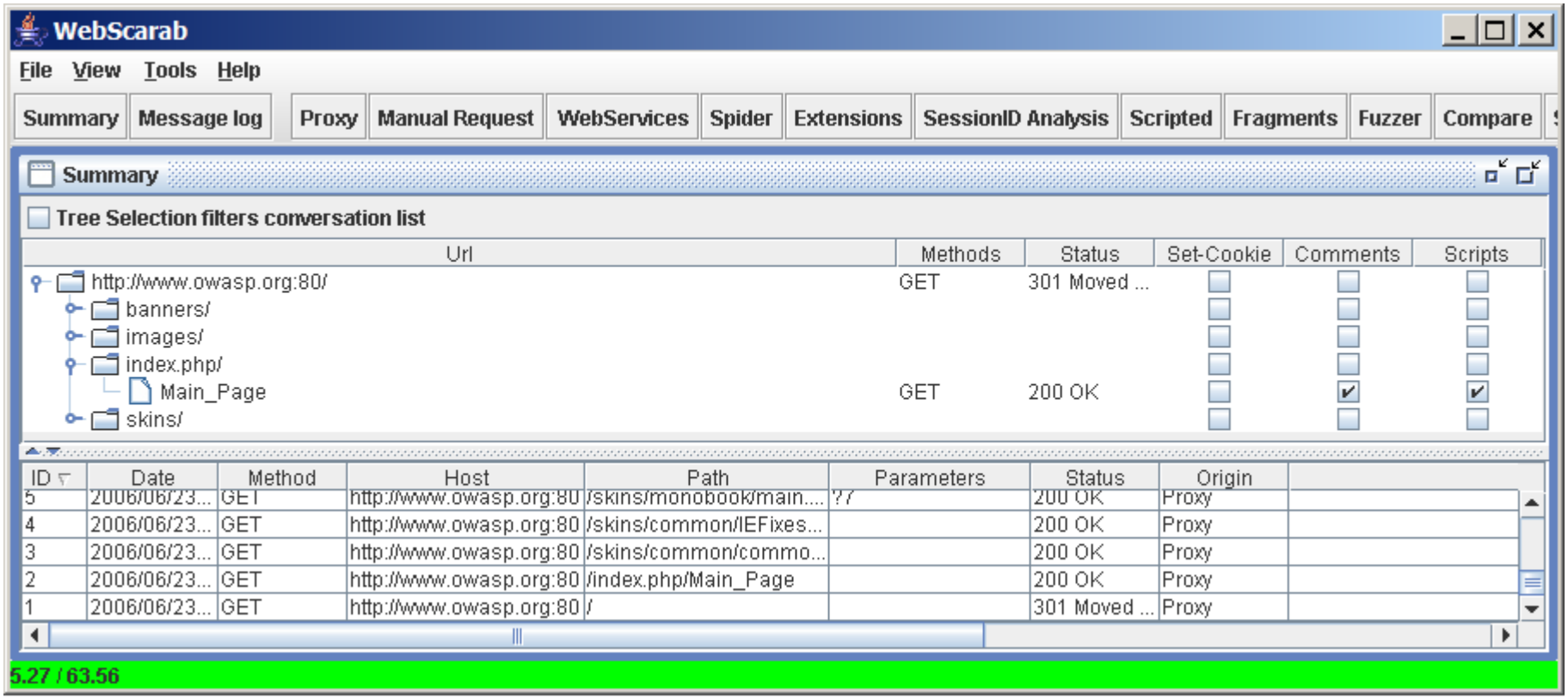
OWASP Testing Guide

https://www.owasp.org/index.php/OWASP_Testing_Project

- Praktyczny opis technik wykorzystywanych przy testach
- 349 stron, 9 kategorii, 66 podatności
- W praktyce:
  - „Encyklopedia” technik testowania
  - Pokazuje „jak”, a nie tylko „co”


Untitled Session - gruyere - OWASP ZAP
File Edit View Analyse Report Tools Help
Sites
Request Response Break
Sites
http://google-gruyere.appspot.co
671572923322
script>ALERT 1
<script>alert(1)<
GET:1
GET:<script>alert(
GET:GruyereCookie.html
GET:deletesnippet(index)
GET:editprofile.gtl
GET:feed.gtl
GET:newsnippet.gtl
GET:newsnippet2(snippet)
GET:saveprofile(action,color
GET:snippets.gtl
GET:snippets.gtl(uid)
GET:upload.gtl
POST:upload2(
POST:upload2(
psiinon
GET:671572923322
GET:favicon.ico
HTTP/1.1 200 OK
Cache-Control: no-cache
Content-type: text/html
Pragma: no-cache
X-XSS-Protection: 0
Expires: Fri, 01 Jan 1990 00:00:00 GMT
Date: Fri, 20 Aug 2010 12:36:19 GMT
Server: Google Frontend
Proxy-Connection: Keep-Alive
Connection: Keep-Alive
</td></tr>
<tr><td>
User name:
</td><td>
<input type='text'
value='psiinon'
name='name' maxlength='16'>
</td></tr>
<tr><td>
OLD Password:
</td><td>
<input type='password' name='oldpw'>
</td><td>
Raw View
History Search Break Points Alerts Active Scan Spider Brute Force Port Scan Output
.*password.*
All
Next
Previous
http://google-gruyere.appspot.com/671572923322/editprofile.gtl OLD Password:
Alerts : 0 : 3 : 0 : 0
Current Scans : 0 : 0 : 0 : 0

OWASP WebScarab



OWASP Hacking-Lab

https://www.hacking-lab.com
/Remote_Sec_Lab/free-owasp-top10-lab.html

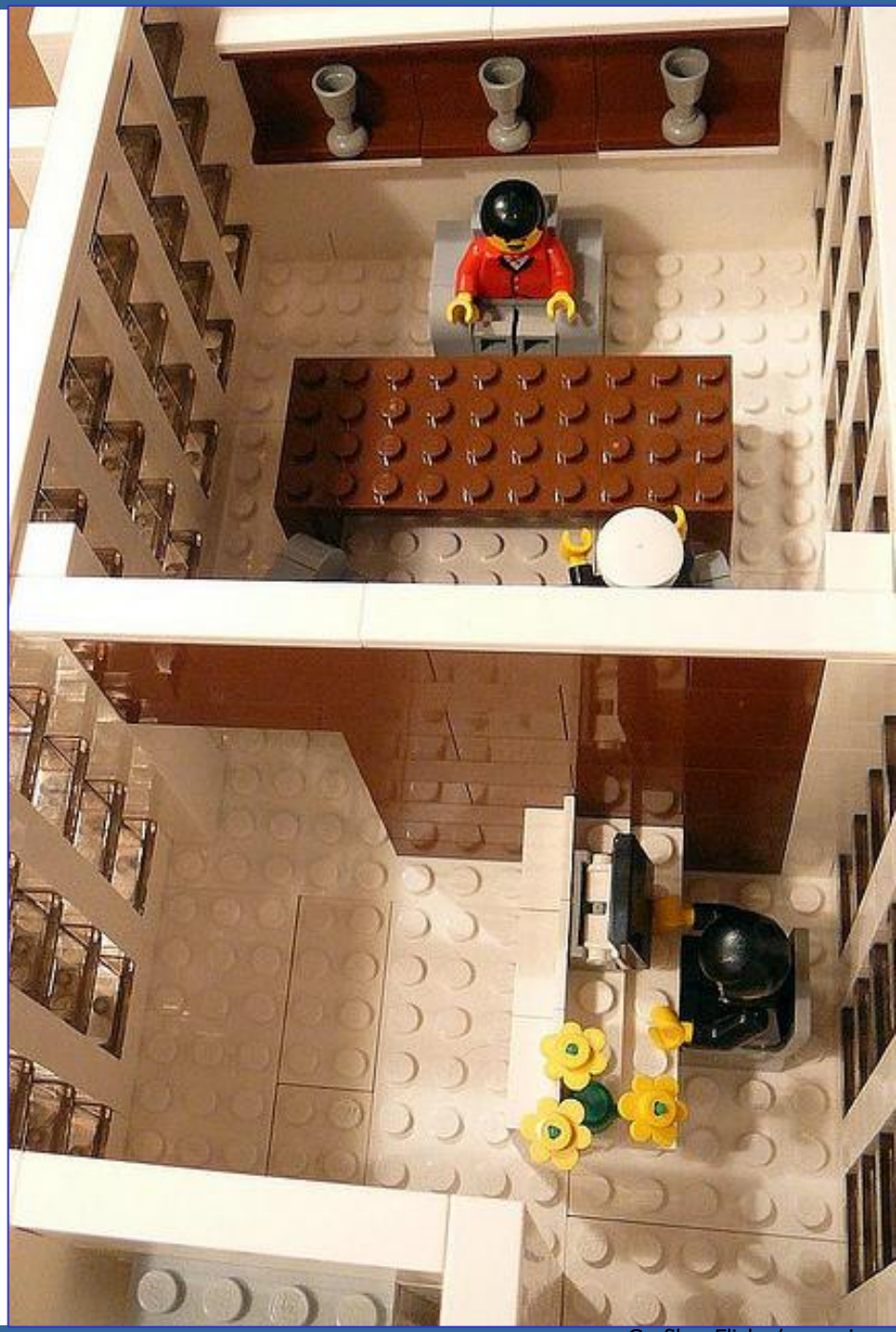
ASVS

Application Security
Verification Standard

Szefowie projektów / zespołów

- Jak zaprojektować bezpieczną aplikację?
- Jak wpisać bezpieczeństwo w cały cykl życia aplikacji?

- Jak sformułować wymagania bezpieczeństwa?
- Jak zamówić bezpieczną aplikację?



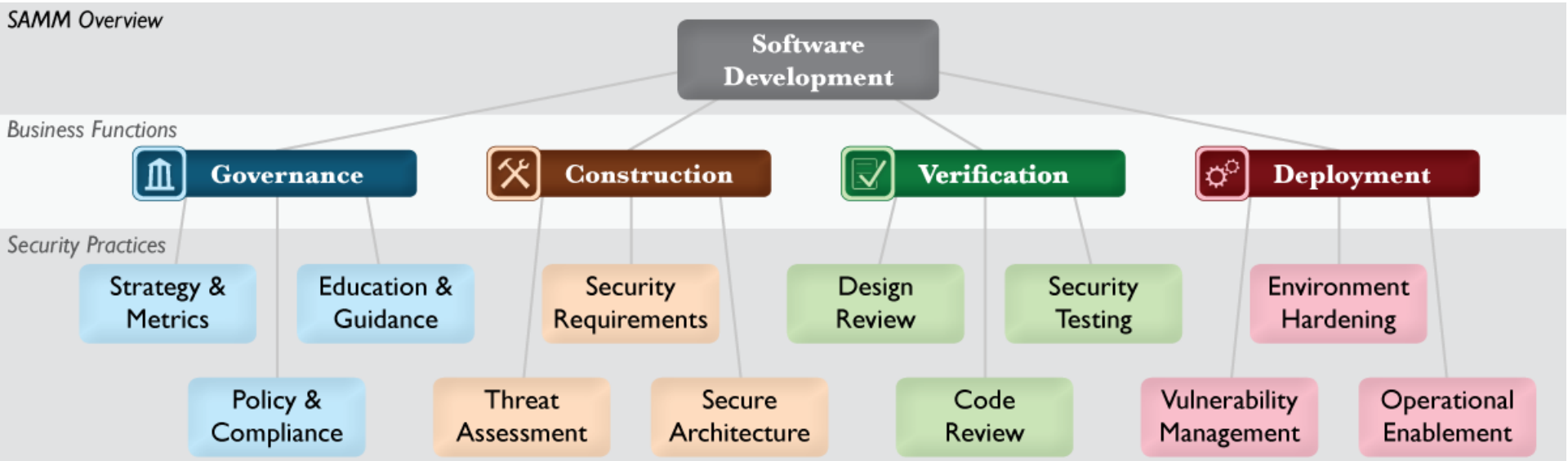
Grafika: Flickr / georgivar

OpenSAMM

http://www.opensamm.org

Software Assurance Maturity Model

- Model dojrzałości dotyczący bezpieczeństwa w procesie wytwarzania oprogramowania
- 4 Business Functions x 3 Security Practices
- Każda z 12 „security practices” ma zdefiniowane 3 poziomy dojrzałości + poziom 0 jako punkt wyjściowy



Źródło: www.owasp.org

OpenSAMM

http://www.opensamm.org

- Dla każdej praktyki / poziomu dojrzałości (4x3x3) opisane są:
  - Cel
  - Czynności
  - Pytania do audytu
  - Rezultat wdrożenia
  - Miara sukcesu
  - Wpływ na koszty, niezbędny personel

OWASP Application Security Verification Standard (ASVS)

- Testowanie zabezpieczeń, które chronią przed typowymi zagrożeniami
- Celem oceny wg ASVS nie jest poszukiwanie podatności
- ale sprawdzenie czy istnieją odpowiednie zabezpieczenia – zasady dobrej praktyki

OWASP ASVS – c.d.

Zastosowanie:

- Jako wzorzec – przy weryfikacji
  (da się zastosować jako wzorzec audytowy)
- Jako wytyczne – dla developerów
- Jako specyfikacja – w kontraktach na wykonanie aplikacji

Jest dostępny **po polsku!**

Owasp.org -> ASVS -> Downloads -> ASVS in Polish

http://owasp-asvs.googlecode.com/files/asvs-webapp-release-2009-pl.pdf

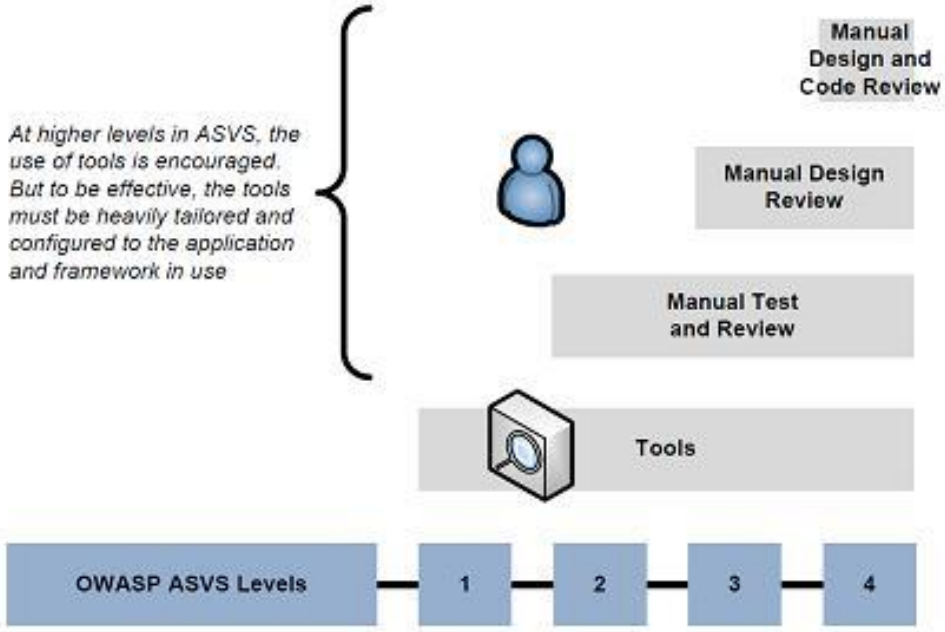
OWASP ASVS – Poziomy weryfikacji

Poziom 1 – Weryfikacja automatyczna

- 1A – Dynamic Scan
- 1B – Source Code Scan

Poziom 2 – Weryfikacja ręczna

- 2A – Penetration Test
- 2B – Code Review



Poziom 3 – Weryfikacja projektu

- L2 + wszystkie biblioteki i serwisy + modelowanie zagrożeń i weryfikacja projektu

Poziom 4 – Weryfikacja wewnętrzna

- L3 + framework, narzędzia, etc + weryfikacja możliwości wprowadzenia złośliwego kodu

Projekty OWASP

	DETECT	PROTECT	LIFE-CYCLE
Documentation	Top10 ASVS Testing Guide Code Review Guide	Development Guide Secure Coding Practices – Quick Reference	OpenSAMM
Tools	WebScarab Zed Attack Proxy JBroFuzz	ESAPI AppSensor ModSecurity Core Ruleset	WebGoat Education Project

~140+ Projektów

https://www.owasp.org/index.php/Category:OWASP_Project

OWASP Poland Local Chapter

- Od połowy 2007
- 24 spotkania
- Konferencja AppSec EU 2009
  - Ponad 200 uczestników z całego Świata
- Lista dyskusyjna owasp-poland
- Tłumaczenia dokumentów OWASP
- Spotkania w Krakowie i Warszawie

  http://www.owasp.org/index.php/Poland

Zapraszamy na spotkania OWASP Poland

- Wstępny termin: 2012-05-16 (środa), 18:00
- Krakowski Park Technologiczny
  Al. Jana Pawła II 41 L, III piętro
  - Ochrona aplikacji J2EE
  - Zabezpieczanie IIS (?)
- Wstęp wolny

OWASP – bieżące informacje

- https://www.owasp.org/index.php/Poland
- http://lists.owasp.org/mailman/listinfo/owasp-poland
- Blip: http://owasppoland.blip.pl/
- Twitter: @owasppoland
- Facebook: OWASP Poland Local Chapter

Pytania ?

OWASP