OWASP v3: nuovi projects and new goals

(**Matteo Meucci – CISA, CISSP – OWASP- Italy**)

INDEX PRESENTATION :

1. Presentazione di OWASP e del capitolo italiano
2. OWASP Projects and vision
3. OWASP Testing Guide v3

ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

Who am I?

- Research
  - –OWASP-Italy Chair
  - –OWASP Testing Guide Lead

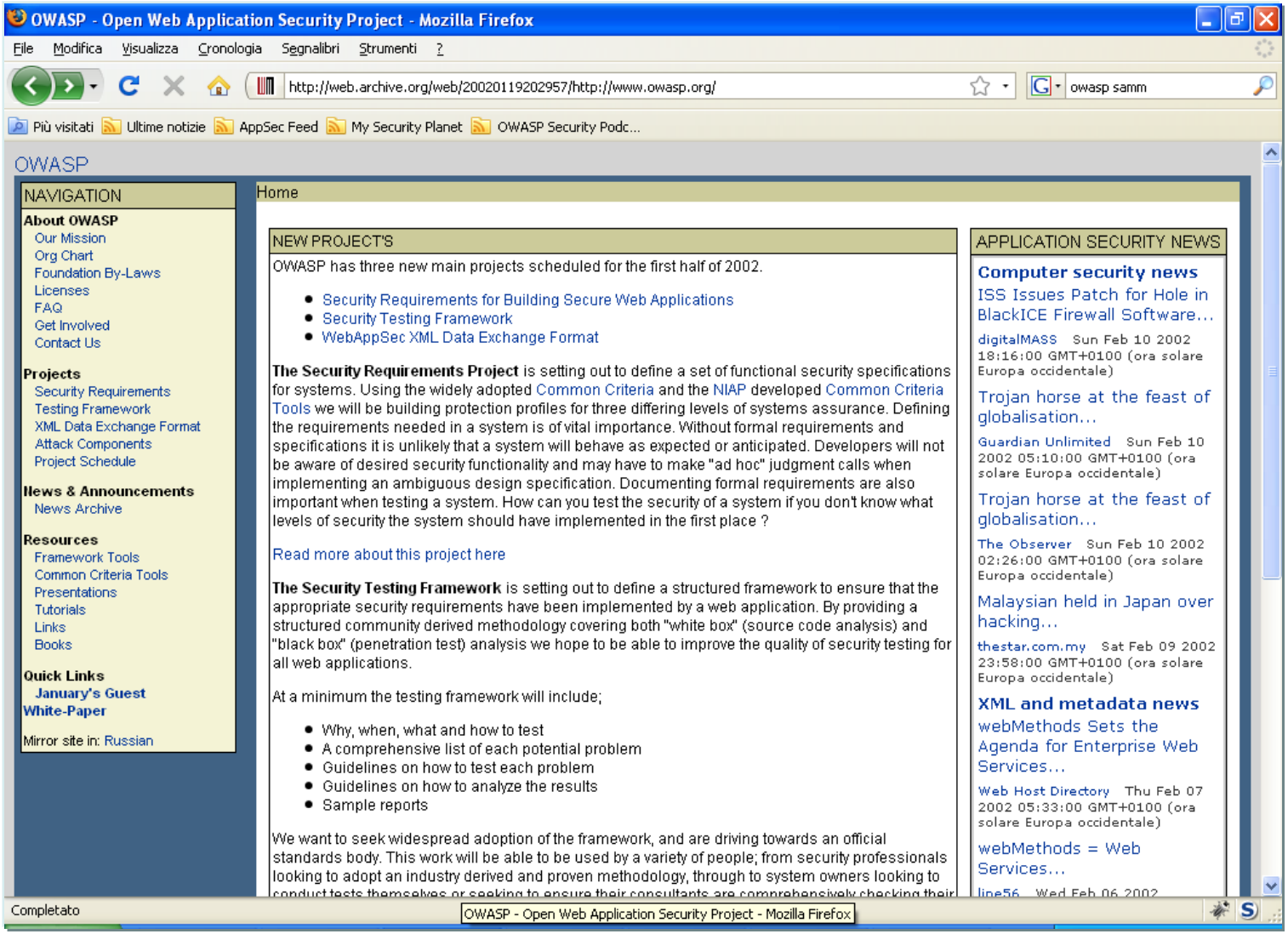
- Work
  - – CEO @ Minded Security

    Application Security Consulting
  - – 8+ years on Information Security

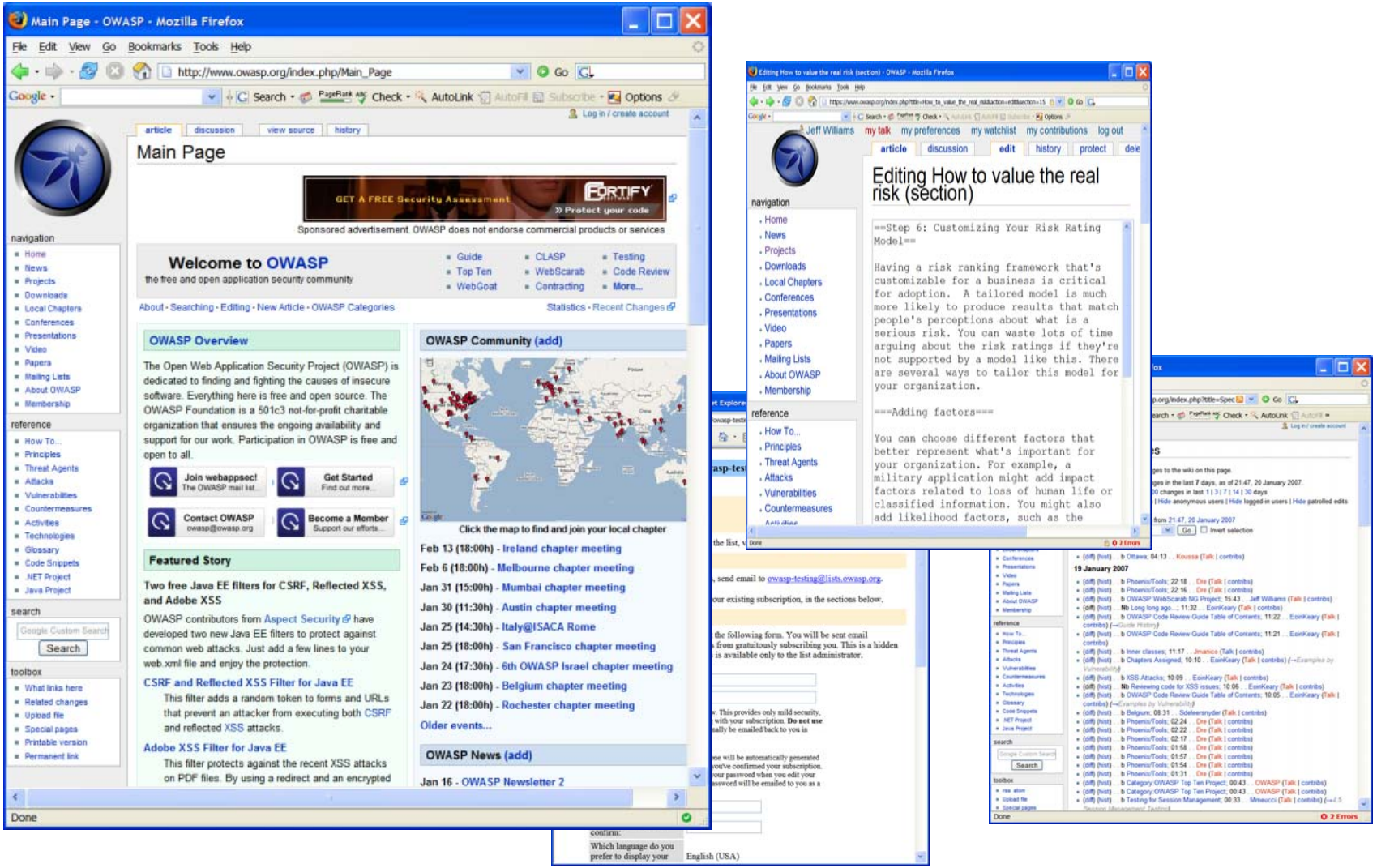
    focusing on Application Security
  - – www.mindedsecurity.com

- The Open Web Application Security Project (OWASP) is dedicated to finding and fighting the causes of insecure software. The OWASP Foundation is a 501c3 not-for-profit charitable organization that ensures the ongoing availability and support for our work.
- Participation in OWASP is free and open to all.
- Everything here is free and open source.
- Main objectives: producing tools, standards and documentations related to Web Application Security.
- Thousands active members, 100+ local chapters in the world
- Millions of hits on www.owasp.org

OWASP v1: start the community



OWASP v2: wiki and sharing



ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

OWASP v3: Improve Quality and Support

- Define Criteria for Quality Levels
  - Alpha, Beta, Release
- Encourage Increased Quality
  - Through Season of Code Funding and Support
  - Produce Professional OWASP books
- Provide Support
  - Full time executive director (Kate Hartmann)
  - Full time project manager (Paulo Coimbra)
  - Half time technical editor (Kirsten Sitnick)
  - Half time financial support (Alison Shrader)
  - Looking to add programmers (Interns and professionals)

OWASP v3

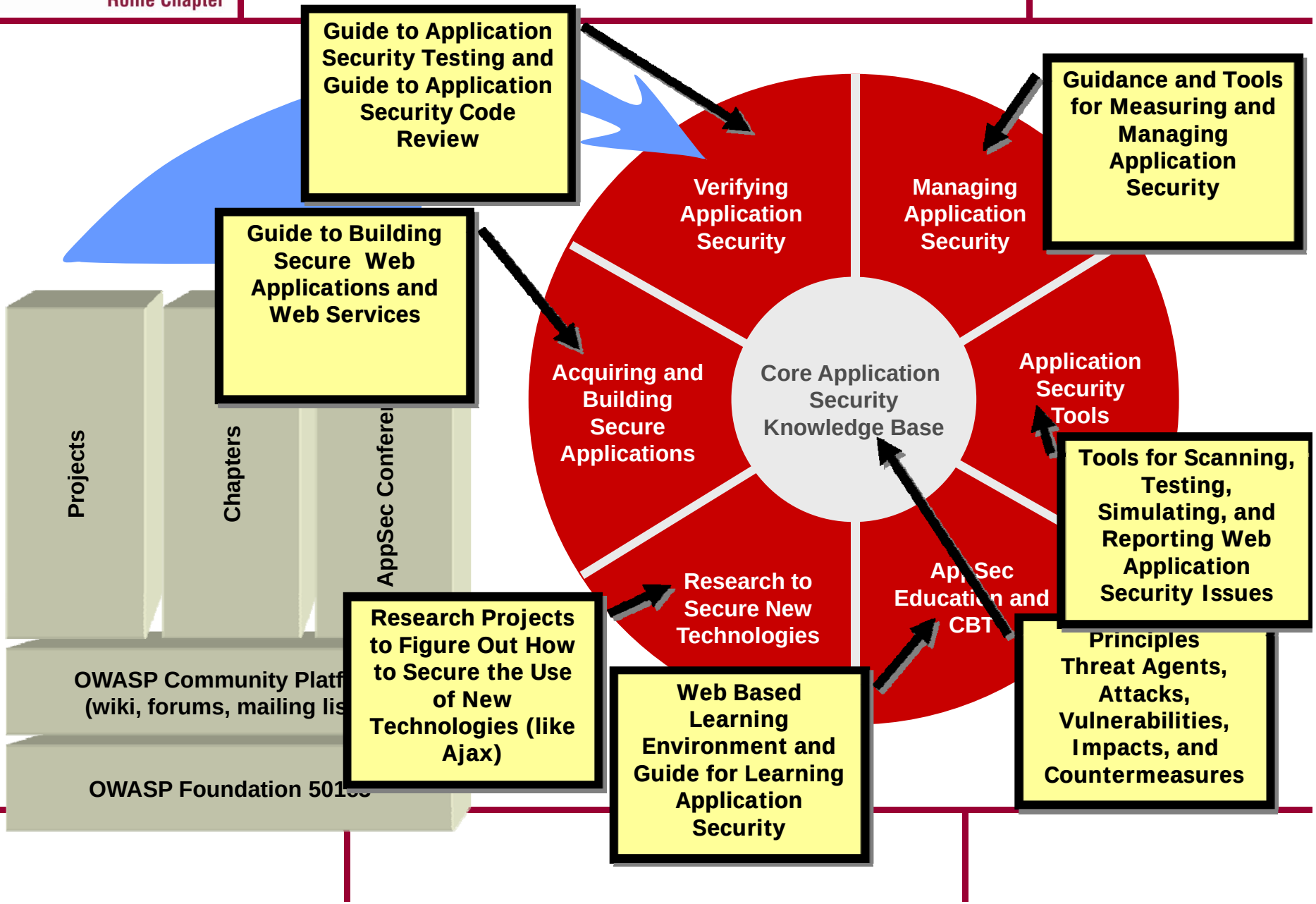
- > 30 project leaders
- OWASP GLOBAL COMMITTEES
  - Projects
  - Membership
  - Education
  - Conferences
  - Industry
  - Chapters

ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

La comunità OWASP

ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

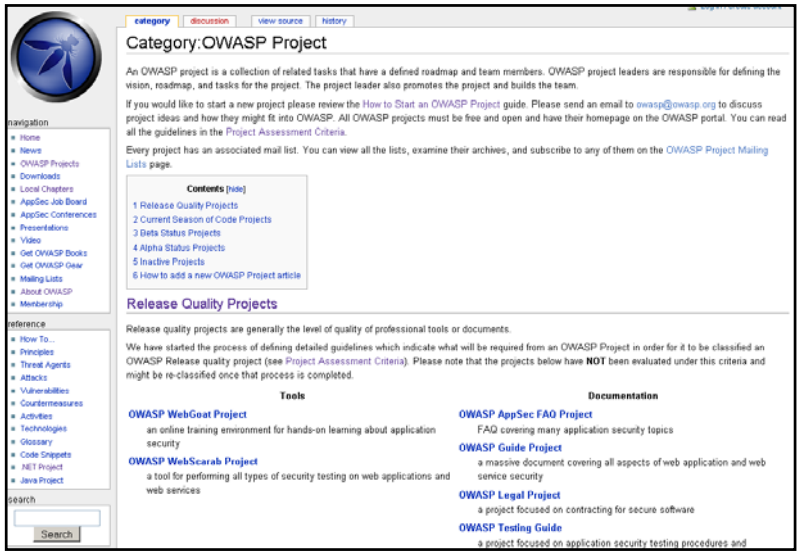
OWASP Body of Knowledge



ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

There are a lot of OWASP projects

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



Category:OWASP Project

An OWASP project is a collection of related tasks that have a defined roadmap and team members. OWASP project leaders are responsible for defining the vision, roadmap, and tasks for the project. The project leader also promotes the project and builds the team.

If you would like to start a new project please review the How to Start an OWASP Project guide. Please send an email to owasp@owasp.org to discuss project ideas and how they might fit into OWASP. All OWASP projects must be free and open and have their homepage on the OWASP portal. You can read all the guidelines in the Project Assessment Criteria.

Every project has an associated mail list. You can view all the lists, examine their archives, and subscribe to any of them on the OWASP Project Mailing Lists page.

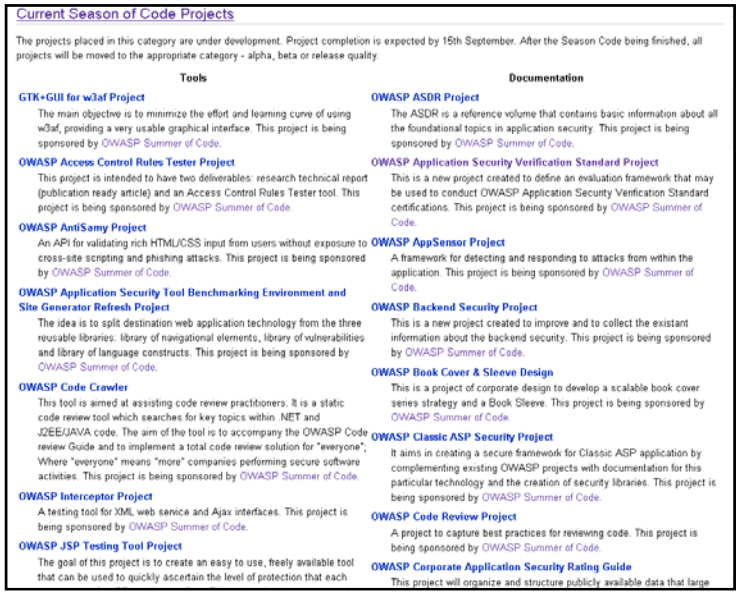
Contents [hide]
1 Release Quality Projects
2 Current Season of Code Projects
3 Beta Status Projects
4 Alpha Status Projects
5 Inactive Projects
6 How to add a new OWASP Project article

Release Quality Projects

Release quality projects are generally the level of quality of professional tools or documents.

We have started the process of defining detailed guidelines which indicate what will be required from an OWASP Project in order for it to be classified an OWASP Release quality project (see Project Assessment Criteria). Please note that the projects below have **NOT** been evaluated under this criteria and might be re-classified once that process is completed.

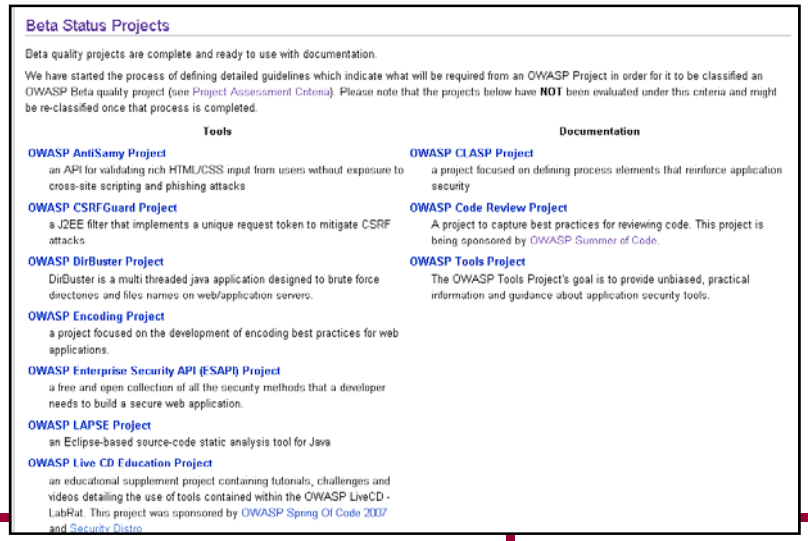
Tools	Documentation
OWASP WebGoat Project an online training environment for hands-on learning about application security	**OWASP AppSec FAQ Project** FAQ covering many application security topics
OWASP WebScarab Project a tool for performing all types of security testing on web applications and web services	**OWASP Guide Project** a massive document covering all aspects of web application and web service security
	OWASP Legal Project a project focused on contracting for secure software
	OWASP Testing Guide a project focused on application security testing procedures and



Current Season of Code Projects

The projects placed in this category are under development. Project completion is expected by 15th September. After the Season Code being finished, all projects will be moved to the appropriate category - alpha, beta or release quality.

Tools	Documentation
GTK+GUI for w3af Project The main objective is to minimize the effort and learning curve of using w3af, providing a very usable graphical interface. This project is being sponsored by OWASP Summer of Code.	**OWASP ASDR Project** The ASDR is a reference volume that contains basic information about all the foundational topics in application security. This project is being sponsored by OWASP Summer of Code.
OWASP Access Control Rules Tester Project This project is intended to have two deliverables: research technical report (publication ready article) and an Access Control Rules Tester tool. This project is being sponsored by OWASP Summer of Code.	**OWASP Application Security Verification Standard Project** This is a new project created to define an evaluation framework that may be used to conduct OWASP Application Security Verification Standard certifications. This project is being sponsored by OWASP Summer of Code.
OWASP AntiSamy Project An API for validating rich HTML/CSS input from users without exposure to cross-site scripting and phishing attacks. This project is being sponsored by OWASP Summer of Code.	**OWASP AppSensor Project** A framework for detecting and responding to attacks from within the application. This project is being sponsored by OWASP Summer of Code.
OWASP Application Security Tool Benchmarking Environment and Site Generator Refresh Project The idea is to split destination web application technology from the three reusable libraries: library of navigational elements, library of vulnerabilities and library of language constructs. This project is being sponsored by OWASP Summer of Code.	**OWASP Backend Security Project** This is a new project created to improve and to collect the existant information about the backend security. This project is being sponsored by OWASP Summer of Code.
OWASP Code Crawler This tool is aimed at assisting code review practitioners. It is a static code review tool which searches for key topics within .NET and J2EE/JAVA code. The aim of the tool is to accompany the OWASP Code review Guide and to implement a total code review solution for "everyone". Where "everyone" means "more" companies performing secure software activities. This project is being sponsored by OWASP Summer of Code.	**OWASP Book Cover & Sleeve Design** This is a project of corporate design to develop a scalable book cover series strategy and a Book Sleeve. This project is being sponsored by OWASP Summer of Code.
OWASP Interceptor Project A testing tool for XML web service and Ajax interfaces. This project is being sponsored by OWASP Summer of Code.	**OWASP Classic ASP Security Project** It aims in creating a secure framework for Classic ASP application by complementing existing OWASP projects with documentation for this particular technology and the creation of security libraries. This project is being sponsored by OWASP Summer of Code.
OWASP JSP Testing Tool Project The goal of this project is to create an easy to use, freely available tool that can be used to quickly ascertain the level of protection that each	**OWASP Code Review Project** A project to capture best practices for reviewing code. This project is being sponsored by OWASP Summer of Code.
	OWASP Corporate Application Security Rating Guide This project will organize and structure publicly available data that large

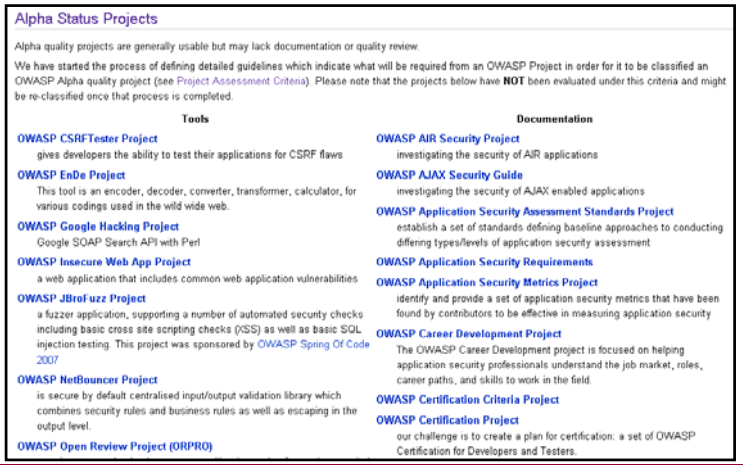


Beta Status Projects

Beta quality projects are complete and ready to use with documentation.

We have started the process of defining detailed guidelines which indicate what will be required from an OWASP Project in order for it to be classified an OWASP Beta quality project (see Project Assessment Criteria). Please note that the projects below have **NOT** been evaluated under this criteria and might be re-classified once that process is completed.

Tools	Documentation
OWASP AntiSamy Project an API for validating rich HTML/CSS input from users without exposure to cross-site scripting and phishing attacks	**OWASP CLASP Project** a project focused on defining process elements that reinforce application security
OWASP CSRFGuard Project a J2EE filter that implements a unique request token to mitigate CSRF attacks	**OWASP Code Review Project** A project to capture best practices for reviewing code. This project is being sponsored by OWASP Summer of Code.
OWASP DirBuster Project DirBuster is a multi threaded java application designed to brute force directories and files names on web/application servers.	**OWASP Tools Project** The OWASP Tools Project's goal is to provide unbiased, practical information and guidance about application security tools.
OWASP Encoding Project a project focused on the development of encoding best practices for web applications.	
OWASP Enterprise Security API (ESAPI) Project a free and open collection of all the security methods that a developer needs to build a secure web application.	
OWASP LAPSE Project an Eclipse-based source-code static analysis tool for Java	
OWASP Live CD Education Project an educational supplement project containing tutorials, challenges and videos detailing the use of tools contained within the OWASP LiveCD - LabRat. This project was sponsored by OWASP Spring Of Code 2007 and Security Distro	



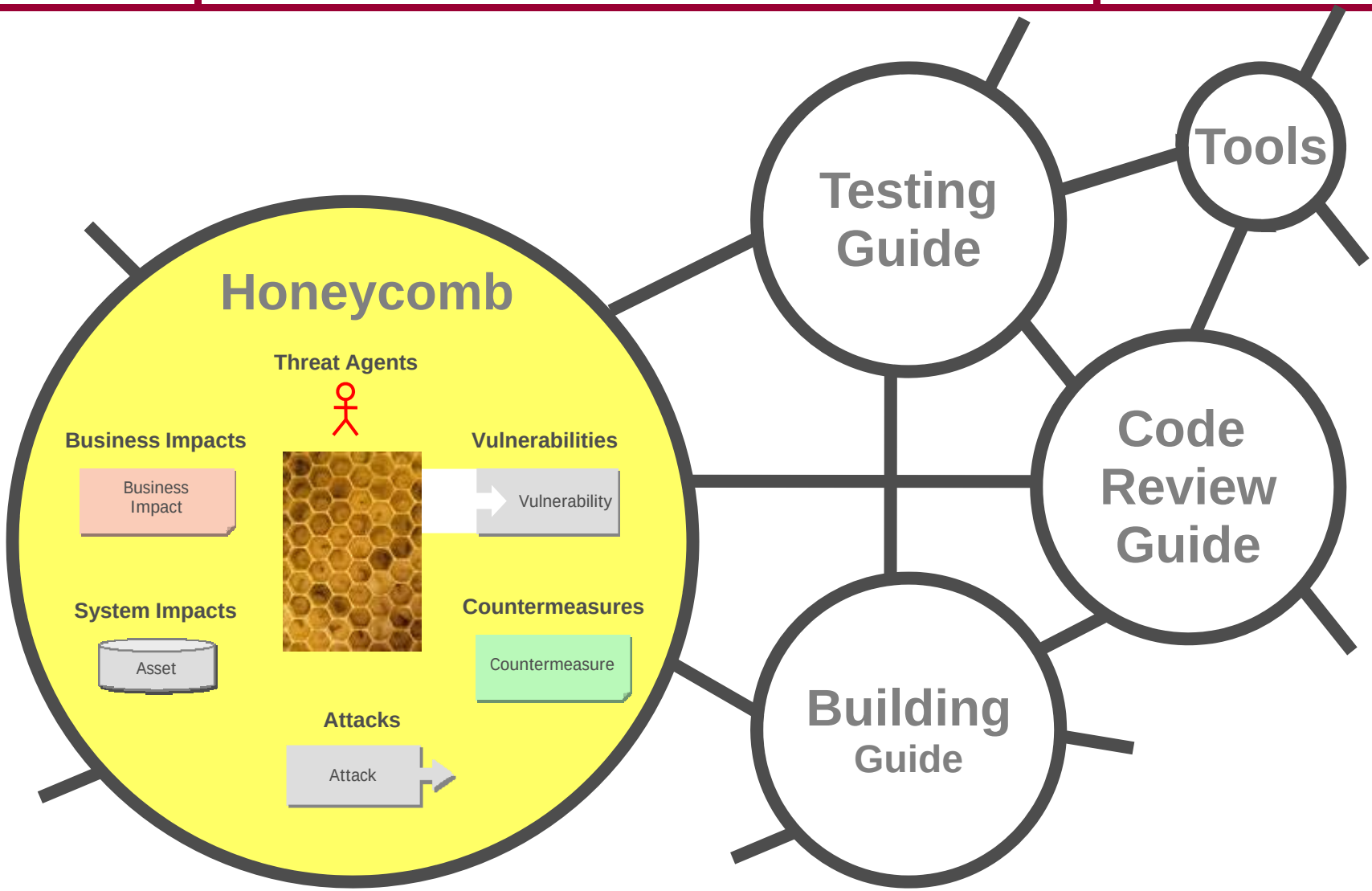
Alpha Status Projects

Alpha quality projects are generally usable but may lack documentation or quality review.

We have started the process of defining detailed guidelines which indicate what will be required from an OWASP Project in order for it to be classified an OWASP Alpha quality project (see Project Assessment Criteria). Please note that the projects below have **NOT** been evaluated under this criteria and might be re-classified once that process is completed.

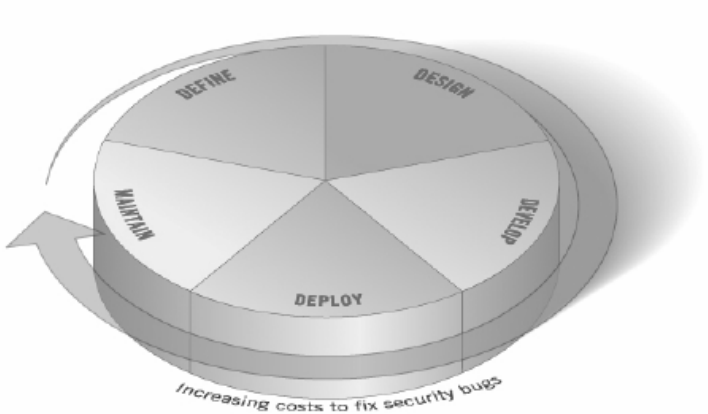
Tools	Documentation
OWASP CSRFTester Project gives developers the ability to test their applications for CSRF flaws	**OWASP AIR Security Project** investigating the security of AIR applications
OWASP EnDe Project This tool is an encoder, decoder, converter, transformer, calculator, for various codings used in the wild wide web.	**OWASP AJAX Security Guide** investigating the security of AJAX enabled applications
OWASP Google Hacking Project Google SOAP Search API with Perl	**OWASP Application Security Assessment Standards Project** establish a set of standards defining baseline approaches to conducting differing types/levels of application security assessment
OWASP Insecure Web App Project a web application that includes common web application vulnerabilities	**OWASP Application Security Requirements**
OWASP JBroFuzz Project a fuzzer application, supporting a number of automated security checks including basic cross site scripting checks (XSS) as well as basic SQL injection testing. This project was sponsored by OWASP Spring Of Code 2007	**OWASP Application Security Metrics Project** identify and provide a set of application security metrics that have been found by contributors to be effective in measuring application security
OWASP NetBouncer Project is secure by default centralised input/output validation library which combines security rules and business rules as well as escaping in the output level.	**OWASP Career Development Project** The OWASP Career Development project is focused on helping application security professionals understand the job market, roles, career paths, and skills to work in the field.
OWASP Open Review Project (ORPRO)	**OWASP Certification Criteria Project**
	OWASP Certification Project our challenge is to create a plan for certification: a set of OWASP Certification for Developers and Testers.

OWASP Guidelines: the big picture



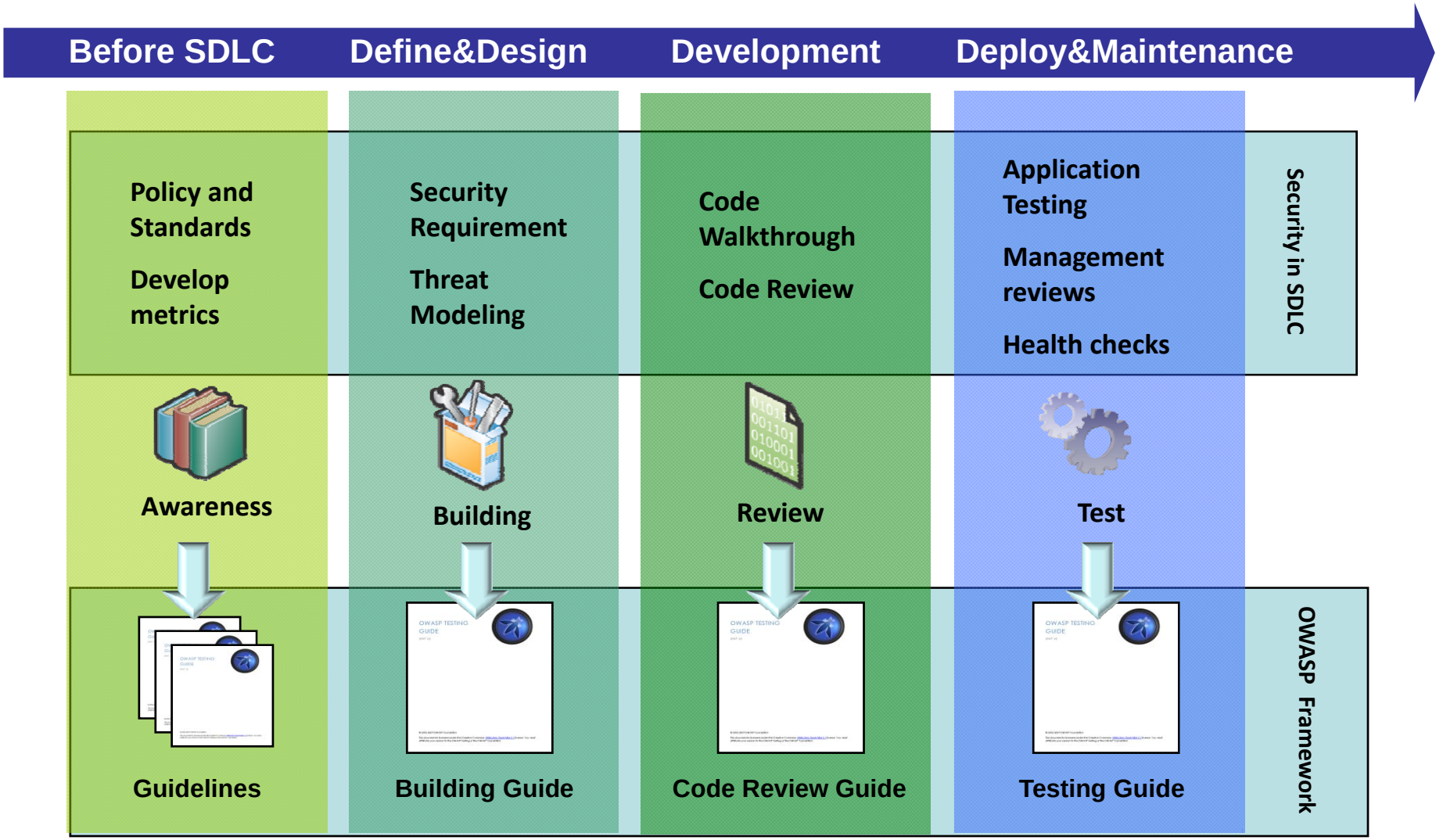
Software development life cycle

- Sofware Development Life Cycle, SDLC:
  - Define
  - Design
  - Develop
  - Deploy
  - Maintain
- Which are the controls to implement?
  - Training
  - Policy Review
  - Guidelines
  - Code Review
  - Web Application Penetration Testing



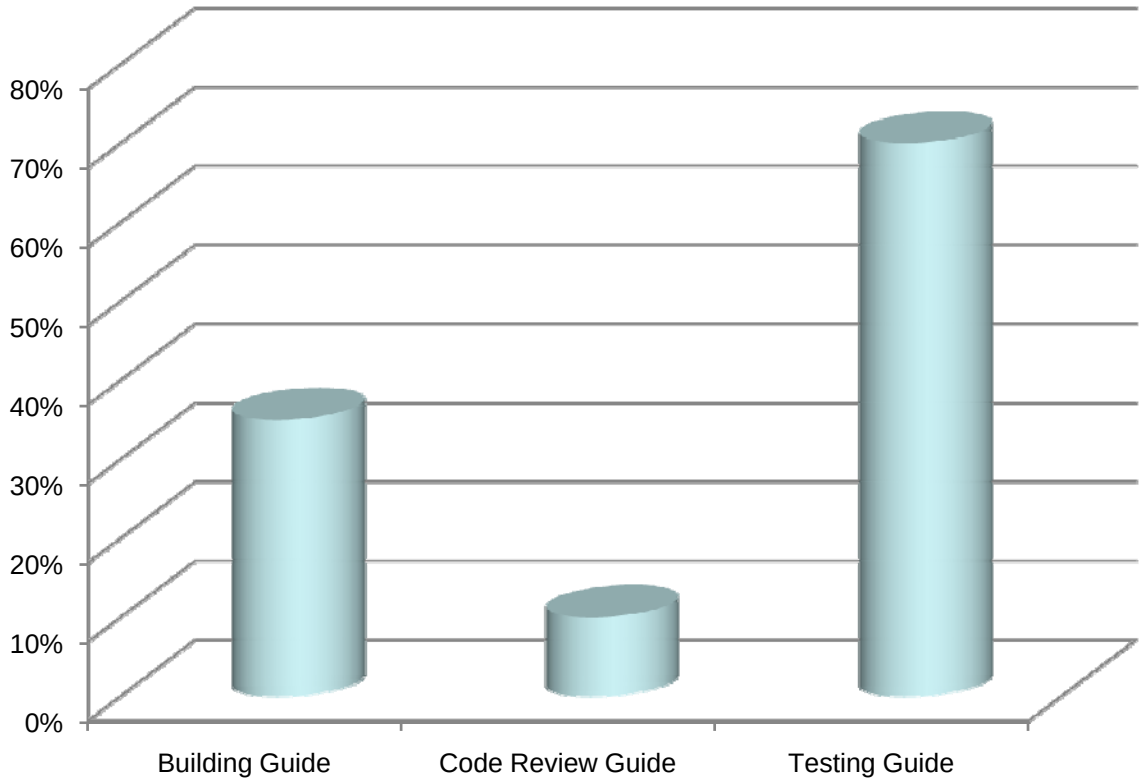
ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

SDLC & OWASP Guidelines in your organization



ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

OWASP guidelines in the italian companies



For a total of 15 Companies (Finance, Banking and Telco)

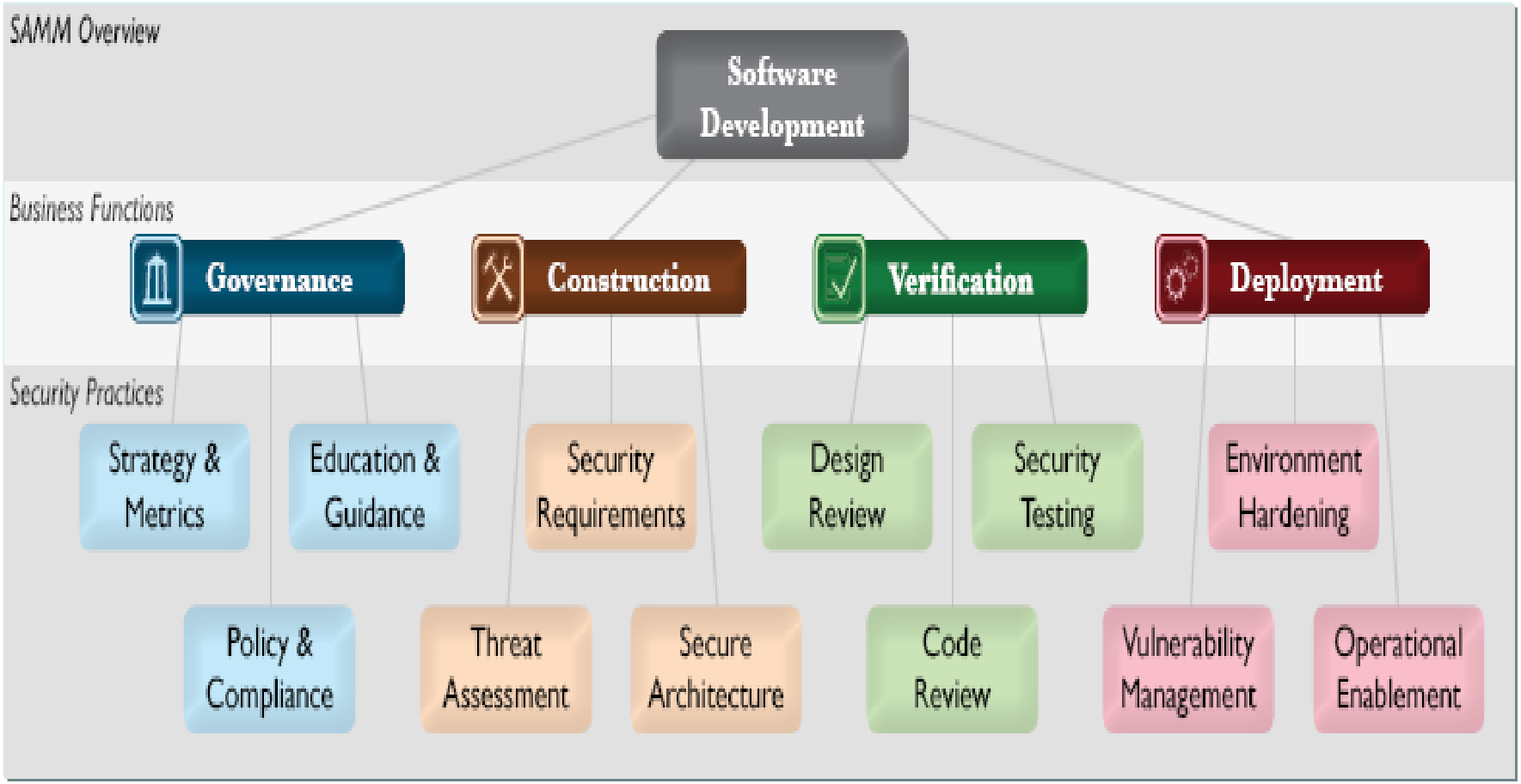
Source: Minded Security 2008

OWASP SAMM

- The Software Assurance Maturity Model (SAMM) project is committed to building a usable framework to help organizations formulate and implement a **strategy for application security** that's tailored to the specific business risks facing the organization.
- The goal is to create well-defined and measurable objectives that can be used by small, medium and large sized organizations in any line of business that involves software development.
- SAMM when:
  - Assess existing software assurance practice
  - Build a strategic roadmap for the organization
  - Implement or perform security activities

Project of Pravir Chandra
V1 released 25th March 09

OWASP SAMM: overview

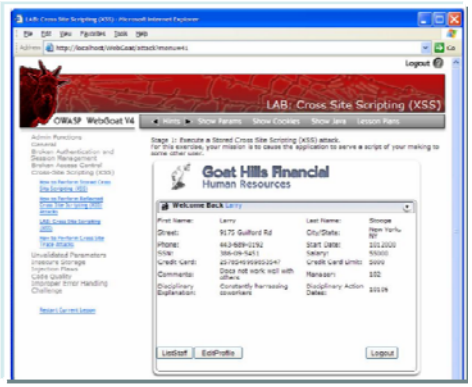


ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

Principali progetti OWASP

BOOKS

- Owasp top10
- Building guide
- Code review guide
- Testing guide
- Back end security



TOOLS

- WebGoat
- WebScarab
- SQLMap – SQL Ninja (indipendent projects)
- SWF Intruder
- Orizon
- Code Crawler

ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

OWASP-Italy e la ricerca

- OWASP Italy nasce nel Gennaio 2005
- Raccoglie centinaia di persone appassionate alla Web Application Security
- Obiettivi
  - Organizzazione conferenze
  - Scrittura articoli
  - Sviluppo tool
  - Sviluppo documentazione e linee guida
- La ricerca come base per l'industria
  - Mai come nell'application security si ha un'esigenza di ricerca per lo sviluppo di attività di innovazione

OWASP Italy Day

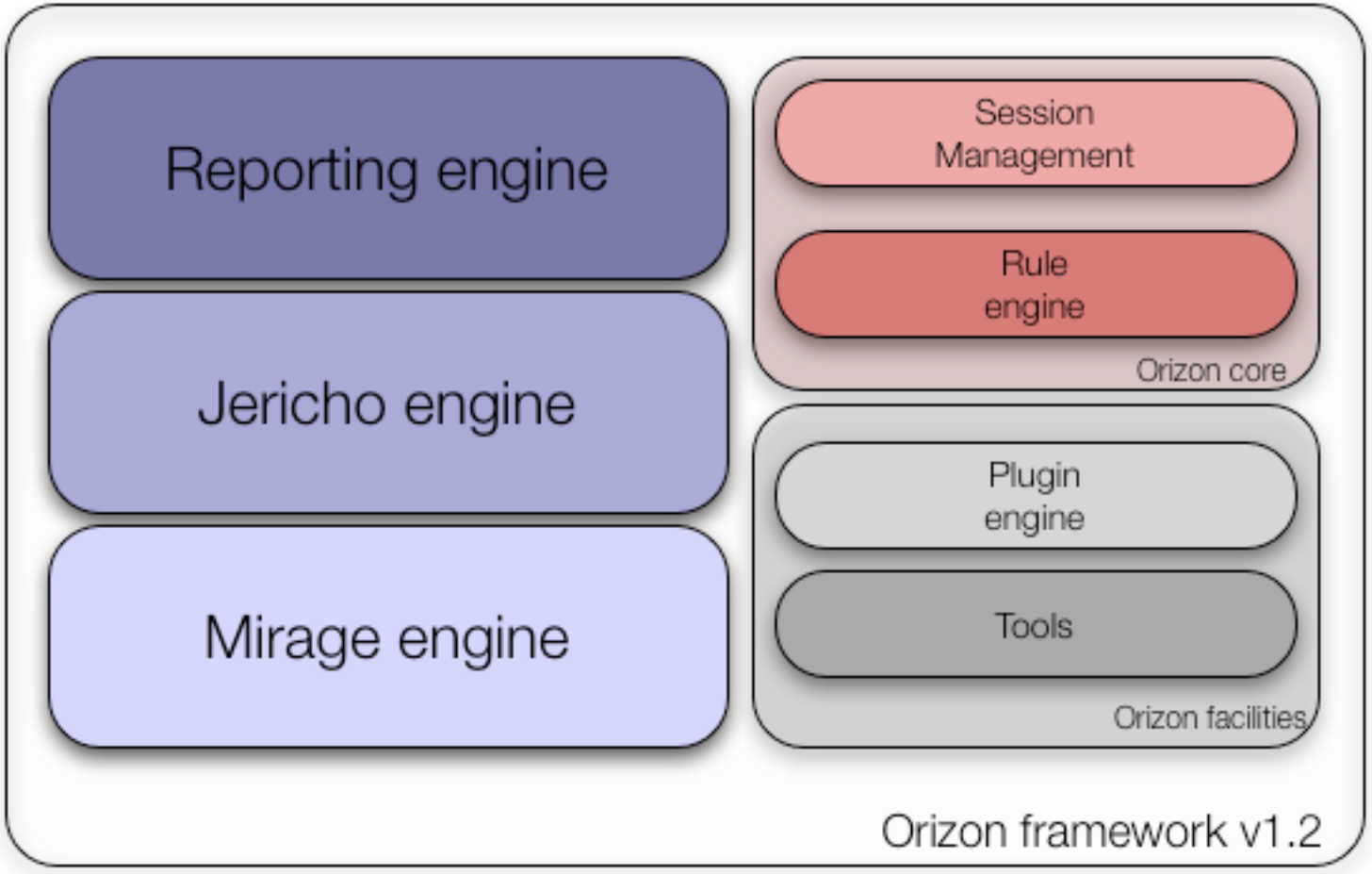
- Day I
  - Marzo 2008 – Università La Sapienza - Roma
- Day II
  - Settembre 2008 – Università La Sapienza - Roma
- Day III
  - Febbraio 2009 – Università di Bari
- Day IV
  - Ottobre/Novembre 2009 – Milano

http://www.owasp.org/Index.php/Italy

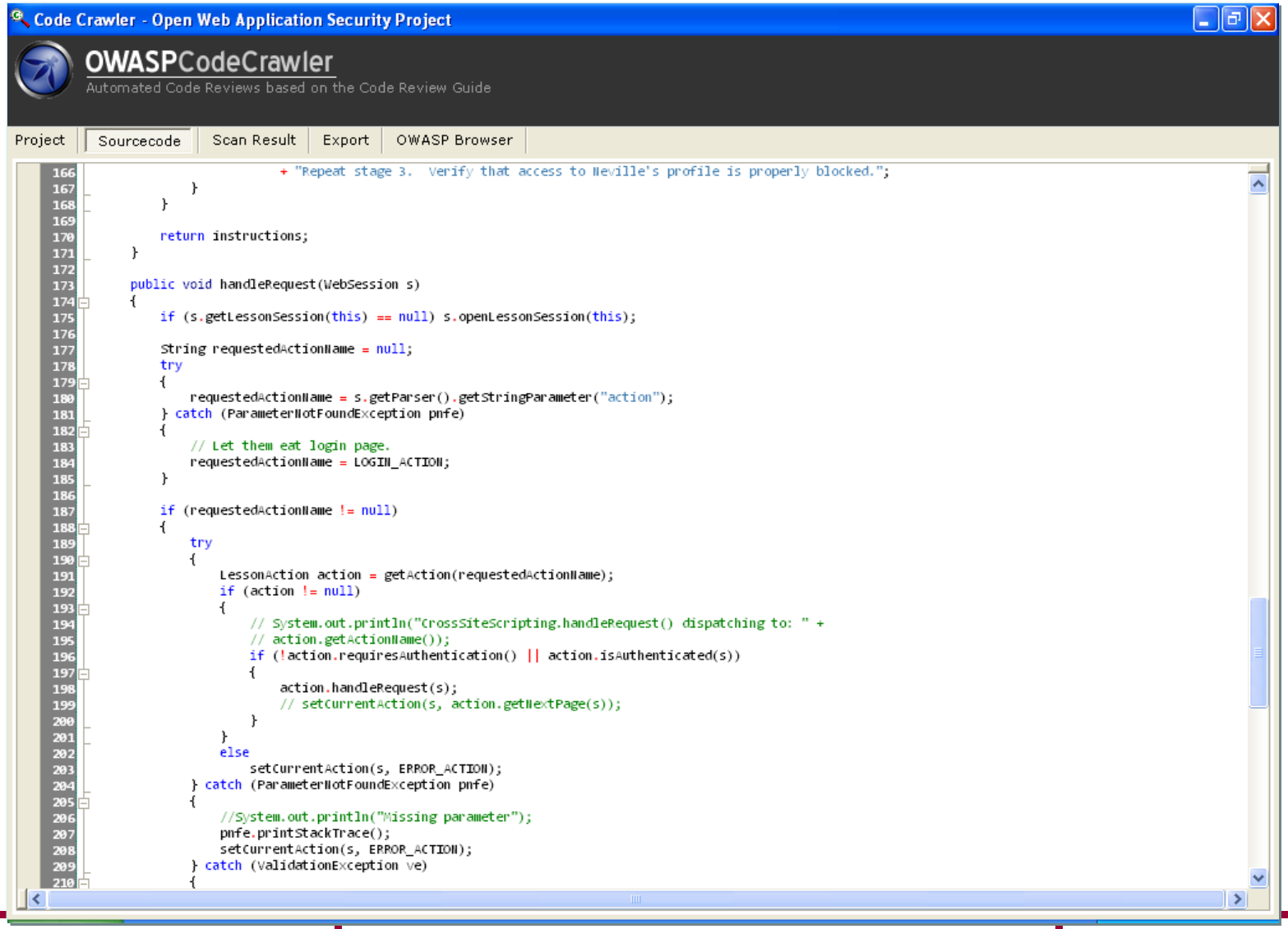
Mailing list!

OWASP-Italy tools: Orizon

by Paolo Perego

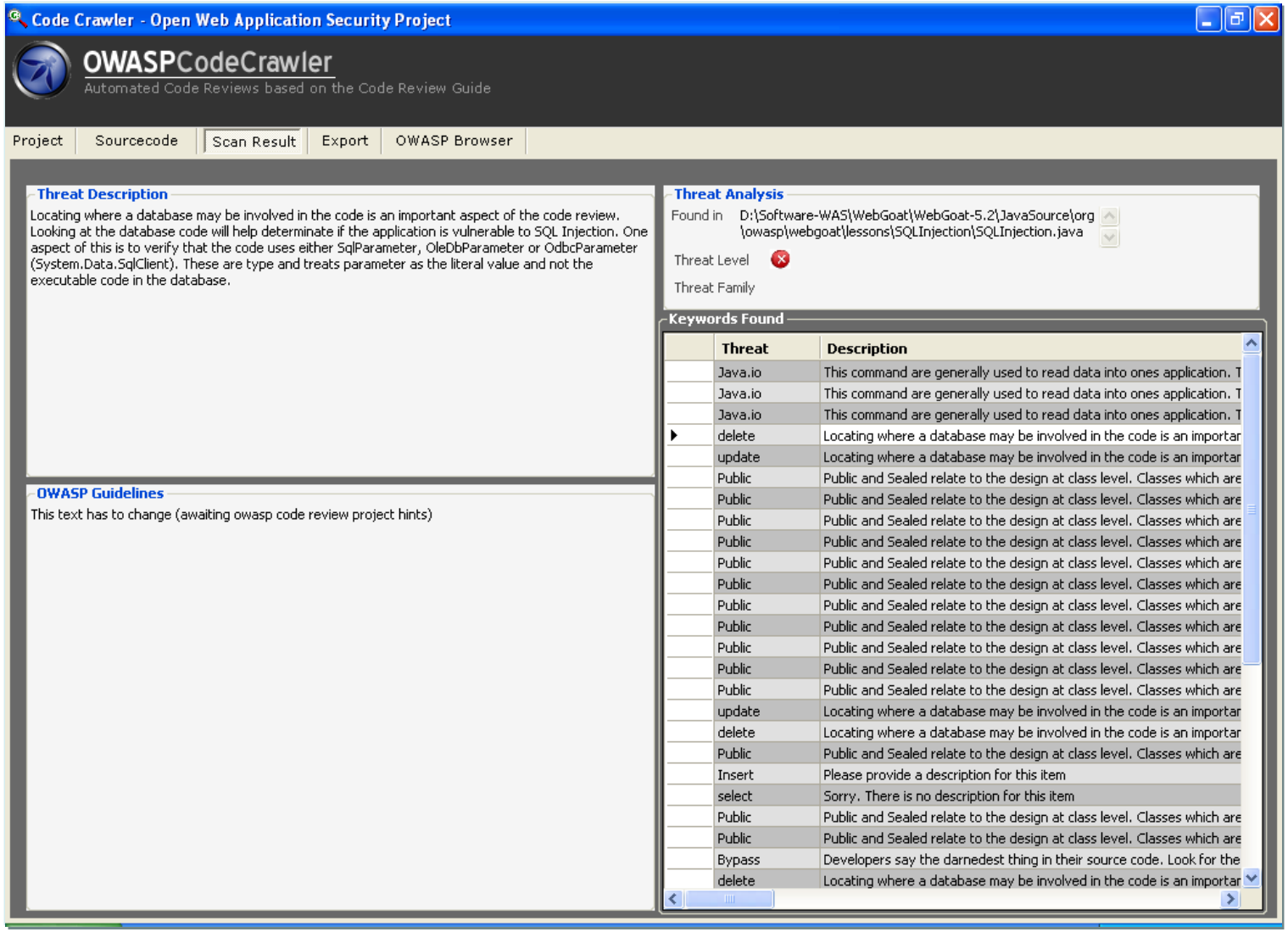


OWASP Code Crawler
by Alessio Marziali




```
                    + "Repeat stage 3.  Verify that access to Neville's profile is properly blocked.";
            }
        }

        return instructions;
    }

    public void handleRequest(WebSession s)
    {
        if (s.getLessonSession(this) == null) s.openLessonSession(this);

        String requestedActionName = null;
        try
        {
            requestedActionName = s.getParser().getStringParameter("action");
        } catch (ParameterNotFoundException pnfe)
        {
            // Let them eat login page.
            requestedActionName = LOGIN_ACTION;
        }

        if (requestedActionName != null)
        {
            try
            {
                LessonAction action = getAction(requestedActionName);
                if (action != null)
                {
                    // System.out.println("CrossSiteScripting.handleRequest() dispatching to: " +
                    // action.getActionName());
                    if (!action.requiresAuthentication() || action.isAuthenticated(s))
                    {
                        action.handleRequest(s);
                        // setCurrentAction(s, action.getNextPage(s));
                    }
                }
                else
                    setCurrentAction(s, ERROR_ACTION);
            } catch (ParameterNotFoundException pnfe)
            {
                //System.out.println("Missing parameter");
                pnfe.printStackTrace();
                setCurrentAction(s, ERROR_ACTION);
            } catch (ValidationException ve)
            {
```

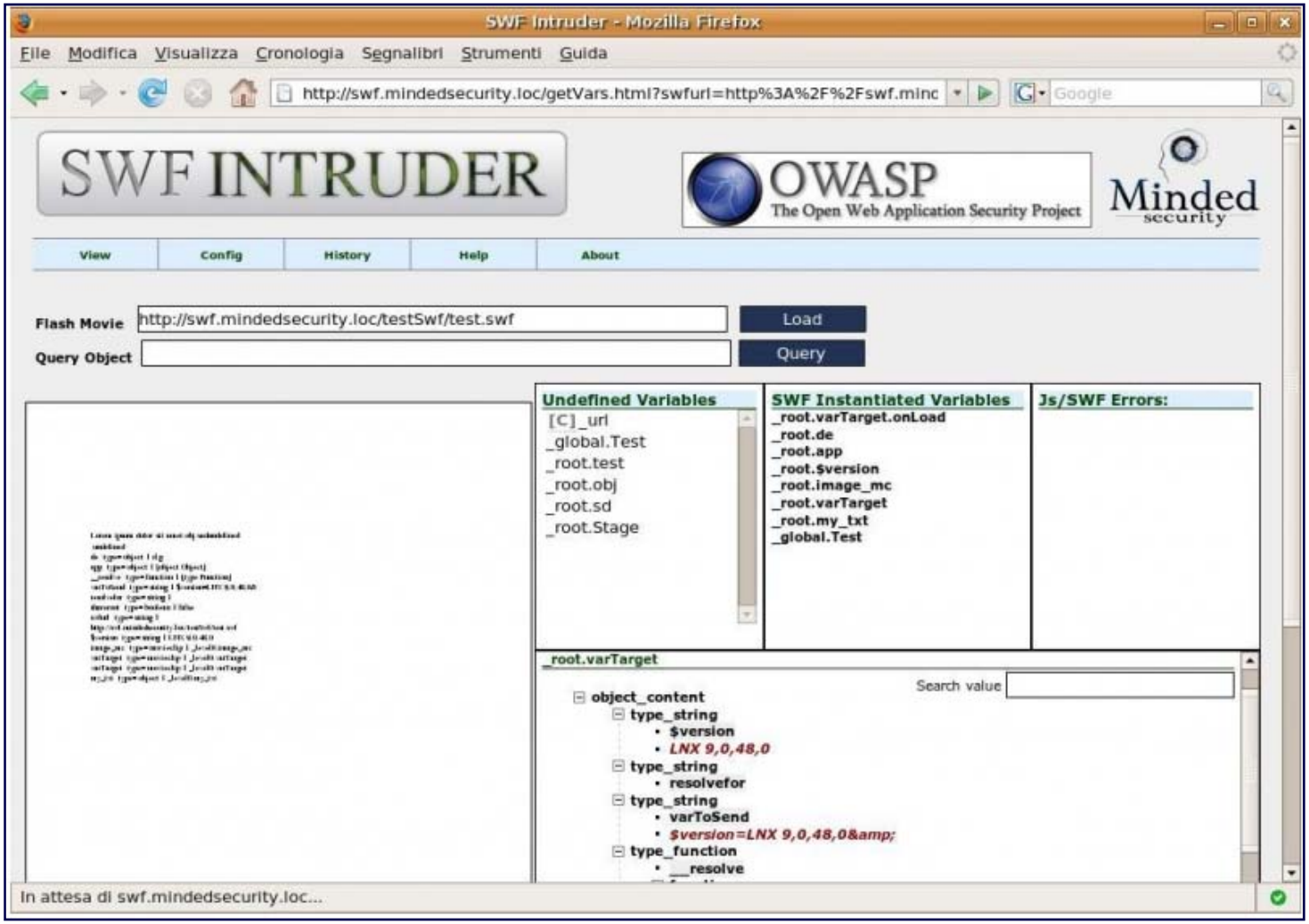
OWASP Code Crawler
by Alessio Marziali



ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

OWASP SWF Intruder

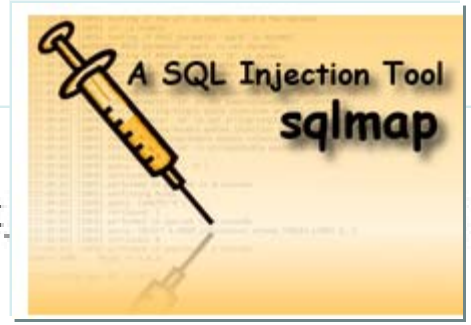
by Stefano Di Paola



ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

SQLMap

By Bernardo Damele e Daniele Bellucci




```
inquis@leboyer:~/sqlmap$ python sqlmap.py -u "http://192.168.1.47/sqlmap/mysql/get

    sqlmap/0.6-rc5 coded by inquis <bernardo.damele@gmail.com>
                    and belch <daniele.bellucci@gmail.com>

[*] starting at: 16:59:37

remote DBMS:    active fingerprint: MySQL >= 5.0.2 and < 5.1
                comment injection fingerprint: MySQL 5.0.45
                html error message fingerprint: MySQL

[*] shutting down at: 16:59:39
```

Screenshot: Database management system back-end extensive fingerprint

SQL Ninja

by Alberto Revelli

- Sqlninja è sviluppato in PERL da Alberto Revelli (aka Icesurfer). Tool che sfrutta SQL Injection per MS SQL Server.
- Non individua SQL Injection, ma si focalizza nel creare una shell interattiva sul DB remoto e sfruttare questa per avere una “base” nella rete target.
  - Fingerprint del SQL Server
  - Bruteforce della password dell’utente 'sa'
  - Privilege escalation to 'sa'
  - Creazione di custom xp_cmdshell
  - Upload di file eseguibili
  - DNS tunneled pseudoshell, when no ports are available for a bindshell
  - E molto altro…

OWASP Back End Security

by Carlo Pelliccioni

- Objectives:
  - Guide that could allow developers, administrators and testers to comprehend any parts of the security process about back-end components that directly communicate with the web applications as well as databases, ldaps, payment gateway...
- Composed of three sections:
  - security development
  - security hardening
  - security testing

OWASP Anti-Malware

by Giorgio Fedon

- La diffusione di Malware risulta in continuo aumento. Nel solo anno 2008 su Internet si sono contati circa 15 milioni di malware.
- Banking Malware: sempre più sofisticati. Si aggiornano in base al paese e alle configurazioni del server su cui si installano.
- Obiettivi:
  - Descrivere i comuni problemi di sicurezza nel design per la protezione di siti di banking
  - Fornire best-practice che dovrebbero essere considerate per realizzare soluzioni antimalware

The OWASP Testing Guide v3

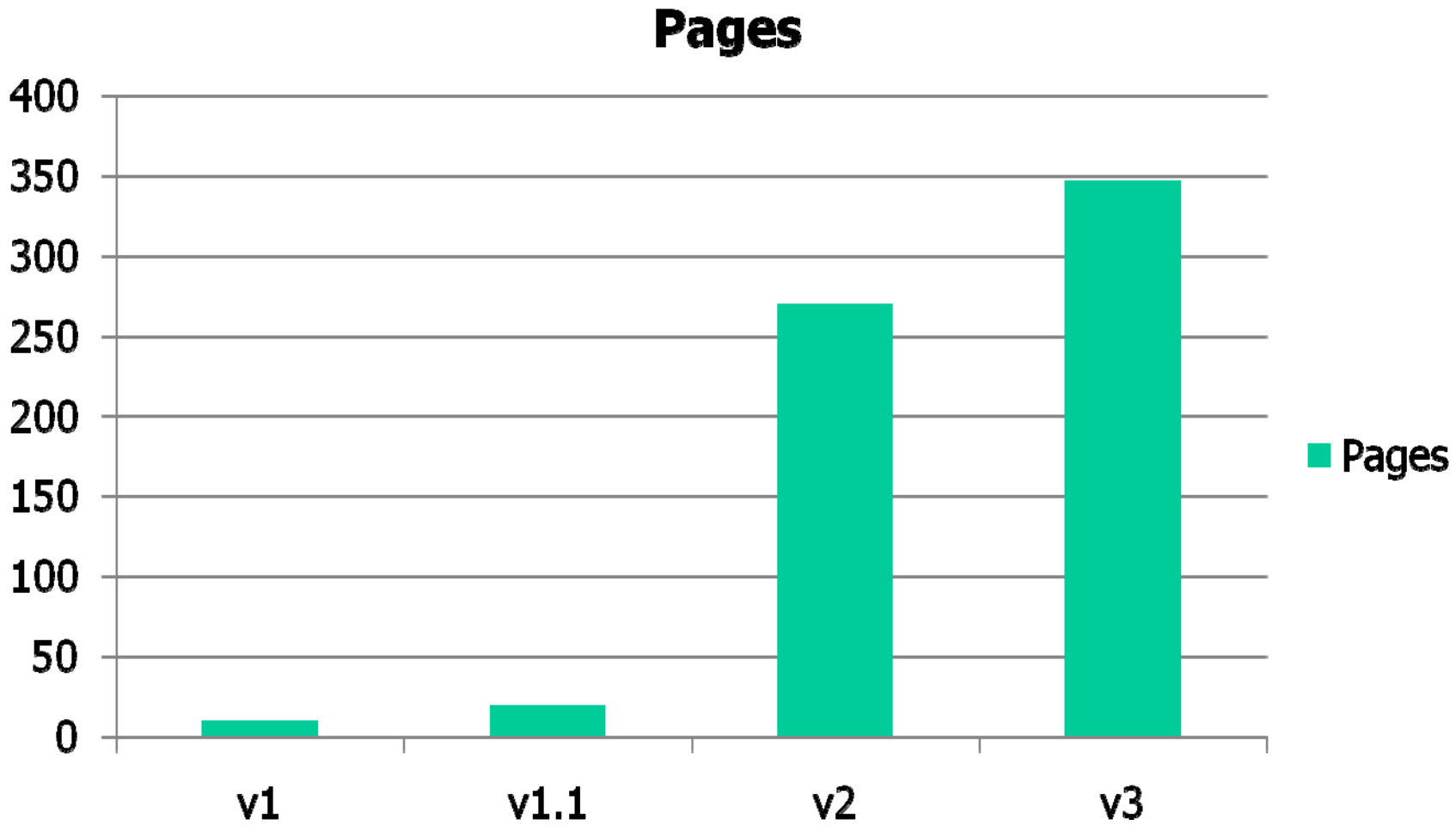
ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

Testing Guide history

- January 2004
  - "The OWASP Testing Guide", Version 1.0
- July 14, 2004
  - "OWASP Web Application Penetration Checklist", Version 1.1
- December 25, 2006
  - "OWASP Testing Guide", Version 2.0
- December 16, 2008
  - "OWASP Testing Guide", Version 3.0 – Released at the OWASP Summit 08

ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

Project Complexity



OWASP Testing Guide v2: Goals

- Review all the documentation on testing:
  - July 14, 2004
    - "OWASP Web Application Penetration Checklist", Version 1.1
- Create a complete new project focused on Web Application Penetration Testing
- Create a reference for application testing
- Describe the OWASP methodology

OWASP Testing Guide v2: Action Plan (2)

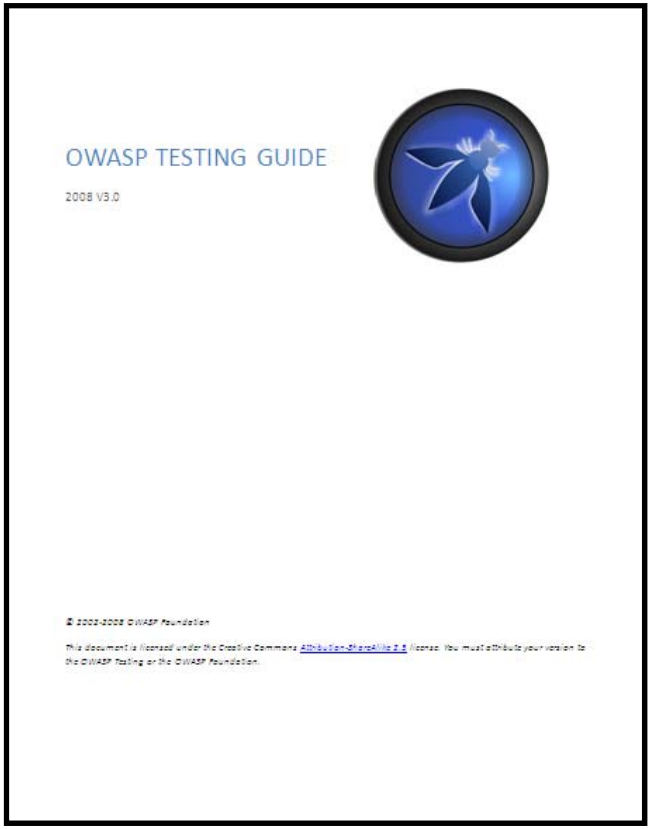
- Nov 2006:
  - Brainstorming for index and templates
  - Write articles using our Wiki model
  - Review articles
- Dec 2006:
  - Review all the Guide
  - Write the Guide in doc format
- Jan 2007:
  - OWASP Testing Guide Release Candidate 1: 272 pages, 46 tests
  - Feedback and review
- Feb 2007:
  - OWASP Testing Guide v2 officially released

ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

OWASP Testing Guide v3: roadmap

- 26th April 2008: start the new project
- OWASP Leaders brainstorming
- Call for participation: 21 authors (-18!)
- Index brainstorming
- Discuss the article content
- 20th May 2008: New draft Index
- 1st June 2008: Let's start writing!
- 27th August 2008: started the reviewing phase: 4 Reviewers (-16!)
- October 2008: Review all the Guide
- December 2008: published the new version of the OWASP Testing Guide: http://www.owasp.org/index.php/OWASP_Testing_Project (347pages +80!)

Testing Guide v3: Index



ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

What's new in v3?

- V2→ 8 sub-categories (for a total amount of 48 controls)
- V3 →10 sub-categories (for a total amount of 66 controls)
- 36 new articles!

- Information Gathering
- Business Logic Testing
- Authentication Testing
- Session Management Testing
- Data Validation Testing
- Denial of Service Testing
- Web Services Testing
- Ajax Testing

- Information Gathering
- Config. Management Testing
- Business Logic Testing
- Authentication Testing
- Authorization Testing
- Session Management Testing
- Data Validation Testing
- Denial of Service Testing
- Web Services Testing
- Ajax Testing
- Encoded Appendix

Web Application Penetration Testing

- What is a Web Application Penetration Testing?
  - The process involves an active analysis of the application for any weaknesses, technical flaws or vulnerabilities
  - It's a Black Box process (we don't know the source code of the application)
  - Methodology + tools (OWASP WebScarab)
- Our approach in writing this guide
  - Open
  - Collaborative
- Defined testing methodology
  - Consistent
  - Repeatable
  - Under quality

Testing paragraph template

- **Brief Summary**
  Describe in "natural language" what we want to test. The target of this section is non-technical people (e.g.: client executive)

- **Description of the Issue**
  Short Description of the Issue: Topic and Explanation

- **Black Box testing and example**
  - **How to test for vulnerabilities:**
  - **Result Expected:**
  - ...
- **Gray Box testing and example**
  - **How to test for vulnerabilities:**
  - **Result Expected:**
  - ...
- **References**
  - **Whitepapers**
  - **Tools**

Example

Black Box vs. Gray Box

Black Box

- The penetration tester does not have any information about the structure of the application, its components and internals

Gray Box

- The penetration tester has partial information about the application internals. E.g.: platform vendor, sessionID generation algorithm

White box testing, defined as complete knowledge of the application internals, is beyond the scope of the Testing Guide and is covered by the OWASP Code Review Project

Introduction to the methodology

In the next slides we will look at a few examples of tests/attacks and at some real-world cases ....

Information Gathering

- The first phase in security assessment is of course focused on collecting all the information about a target application.
- Using public tools it is possible to force the application to leak information by sending messages that reveal the versions and technologies used by the application
- Available techniques include:
  - Testing: Spiders, robots, and Crawlers (OWASP-IG-001)
  - Search engine discovery/Reconnaissance (OWASP-IG-002)
  - Identify application entry points (OWASP-IG-003)
  - Web Application Fingerprint (OWASP-IG-004)
  - Application Discovery (OWASP-IG-005)
  - Analysis of Error Codes (OWASP-IG-006)

Information Gathering (cont.)

Example

ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

Configuration Management Testing

- SSL/TLS Testing (OWASP-CM-001)
- DB Listener Testing (OWASP-CM-002)
- Infrastructure Configuration Management Testing (OWASP-CM-003)
- Application Configuration Management Testing (OWASP-CM-004)
- Testing for File Extensions Handling (OWASP-CM-005)
- Old, Backup and Unreferenced Files (OWASP-CM-006)
- Infrastructure and Application Admin Interfaces (OWASP-CM-007)
- Testing for HTTP Methods and XST (OWASP-CM-008)

SSL Testing

Example

Session Management Testing

Session management is a critical part of a security test, as every application has to deal with the fact that HTTP is by its nature a stateless protocol. Session Management broadly covers all controls on a user from authentication to leaving the application

Tests include the following areas:

- Testing for session management scheme
- Testing for cookie attributes
- Session Fixation
- Exposed session variables
- Cross Site Request Forgery

ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

Session Management Testing

Mario Rossi

--Authentication process-

Web Application

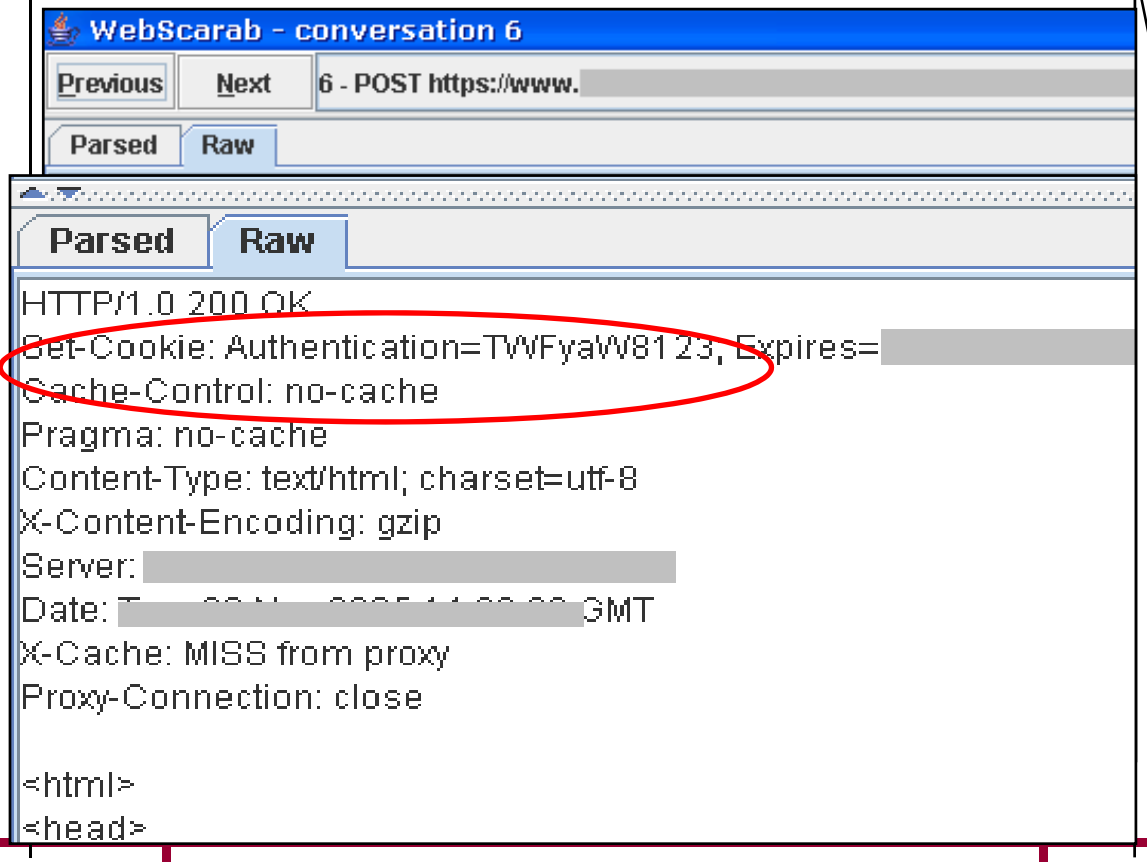
[3] Insert username/password via HTTPS

Credential verify: if ok → client authenticated

→ Cookie generation

Cookie=TWFyaW8123

Token di autenticazione



Session Management Testing

1st Authentication:

User = Mario Rossi; password=12aB45cD:

Cookie=TWFyaW8123

2nd Authentication :

User = Mario Rossi; password=12aB45cD:

Cookie=TWFyaW8125

3rd Authentication :

User = Mario Rossi; password=12aB45cD:

Cookie=TWFyaW8127

Cookie Guessable: Cookie=TWFyaW8129

ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

Session Management Testing

Mario Rossi

--Following request--

Web Application

Cookie=TWFyaW8179

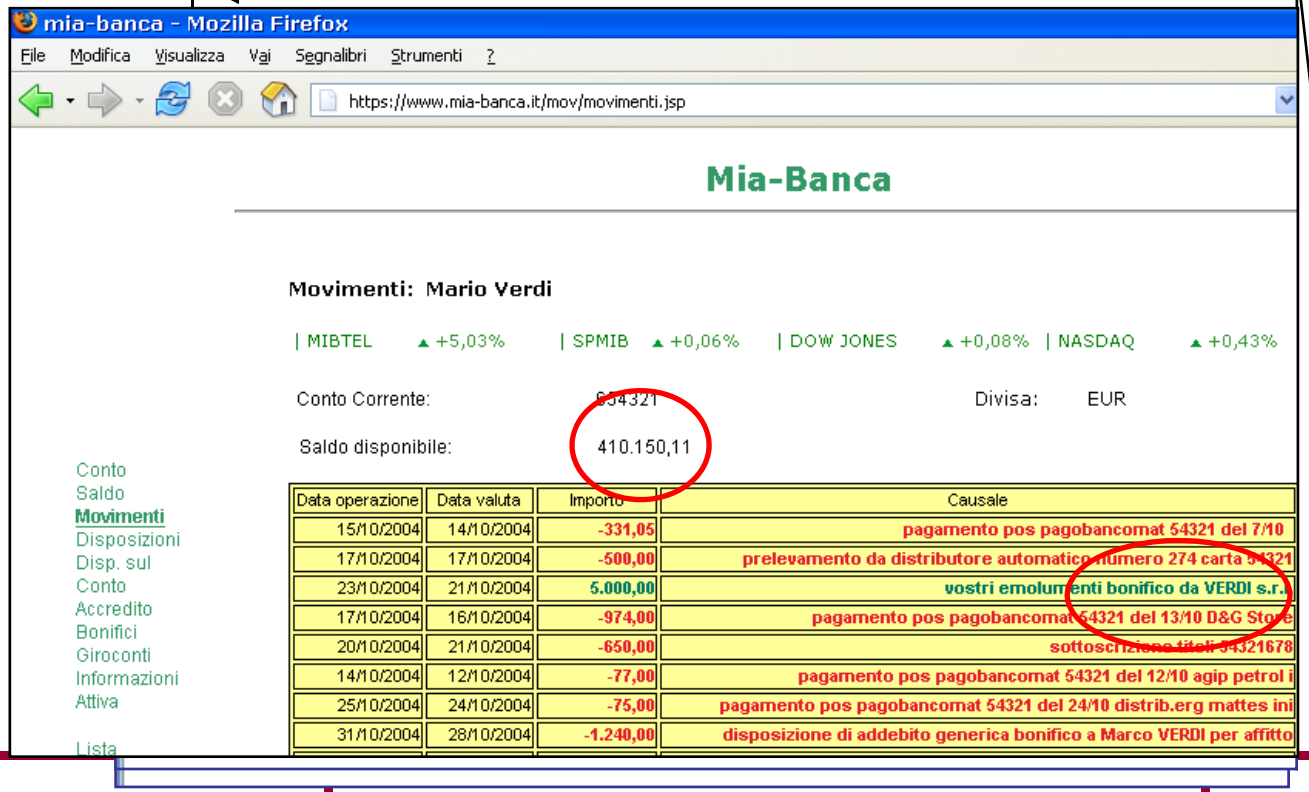
Authentication Token

[5] Request "movimenti"

Cookie=TWFyaW8179

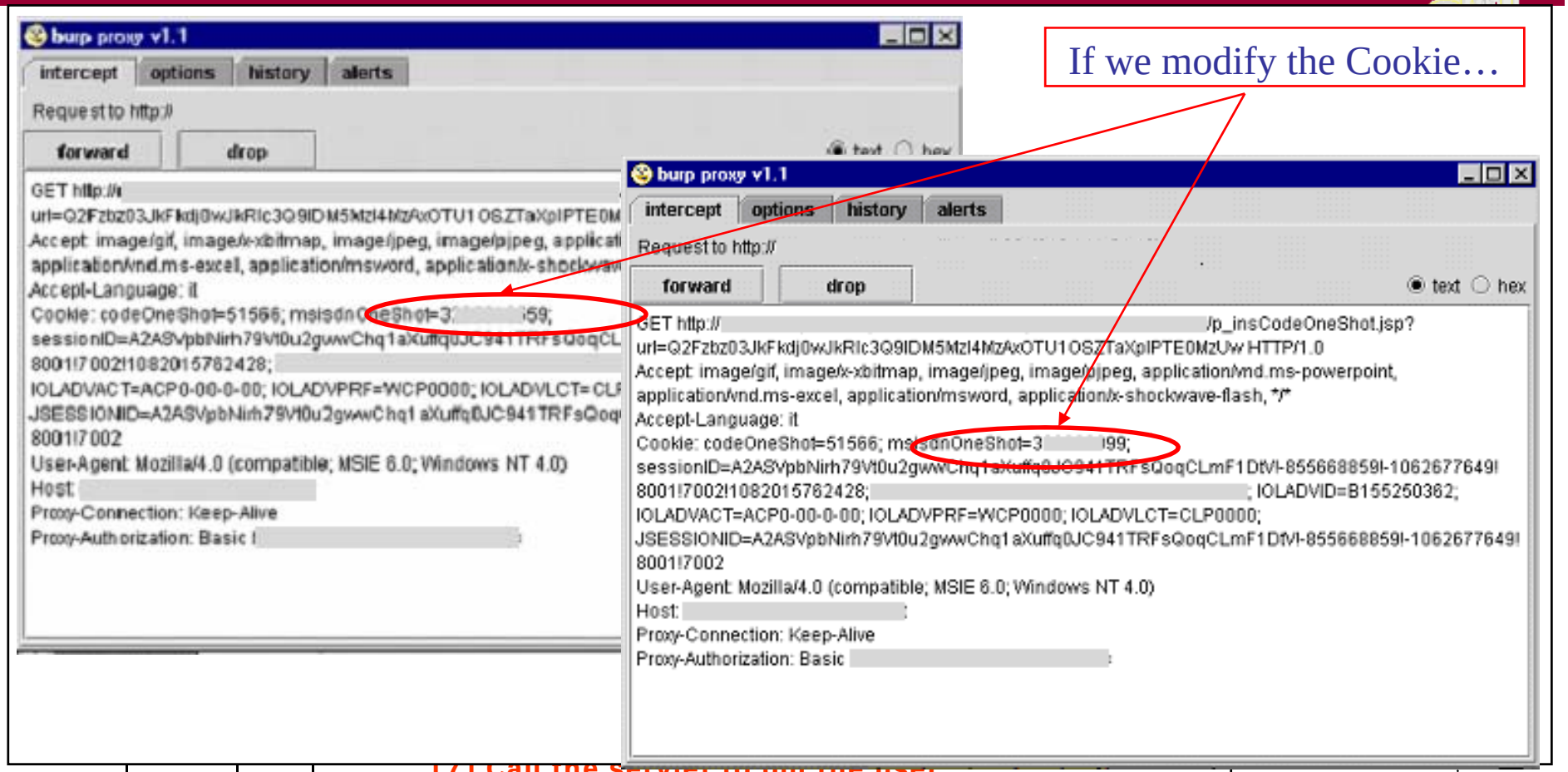
[6] Send Mario Verdi data

Cookie verify:
TWFyaW8177
Identify user Mario Verdi
Send Mario Verdi data



ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

Cookie Manipulation



Cross Site Request Forgery (CSRF)

- CSRF: When
  - The application permits to send requests in a not authorized manner or send duplicate requests
  - The application uses implicit authentication (session cookie)

- CSRF: How

  The <IMG> attack:
  - A Web site contains an <IMG> TAG inside the HTML code that runs an action on the target site (<IMG> TAG has no restriction to origin level)

ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

Testing for CSRF

- 1st Step: find the vulnerable function
  - Create a new user (admin)
  - Fund transfer (users)
- 2nd Step: Force the user to perform that action
  - Malicious Email
  - Malicious site
- 3rd Step: the user will authenticate on the application
  - Browser will be forced to execute an HTTP request
- Result
  - The authorized action will be executed

Testing for Cross Site Request Forgery

First Step

- Online banking. We analyze the transfer fund mechanism
- We notice that after inserting the receiver coordinate and the money amount we will generate the following HTTP GET
- https://exampleBank.com/transfer?eu=1000&to=1234

Second Step

- User must be authenticated on the application and forced to go to a malicious site or read an email
- IMG tag will execute the request without user interaction

Testing for Cross Site Request Forgery

Third Step

```
<html>
<title> This is a new interesting site..visiting me </title>
<body>
..
<img src="https://exampleBank.com/transfer?eu=10000&to=evilcount"
width="0" height="0">
...
</body>
</html>
```

Forth Step

- User browser will execute the action
- There is no way for the application to understand that the action is not forced → log file
- It works!

ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

Authentication testing

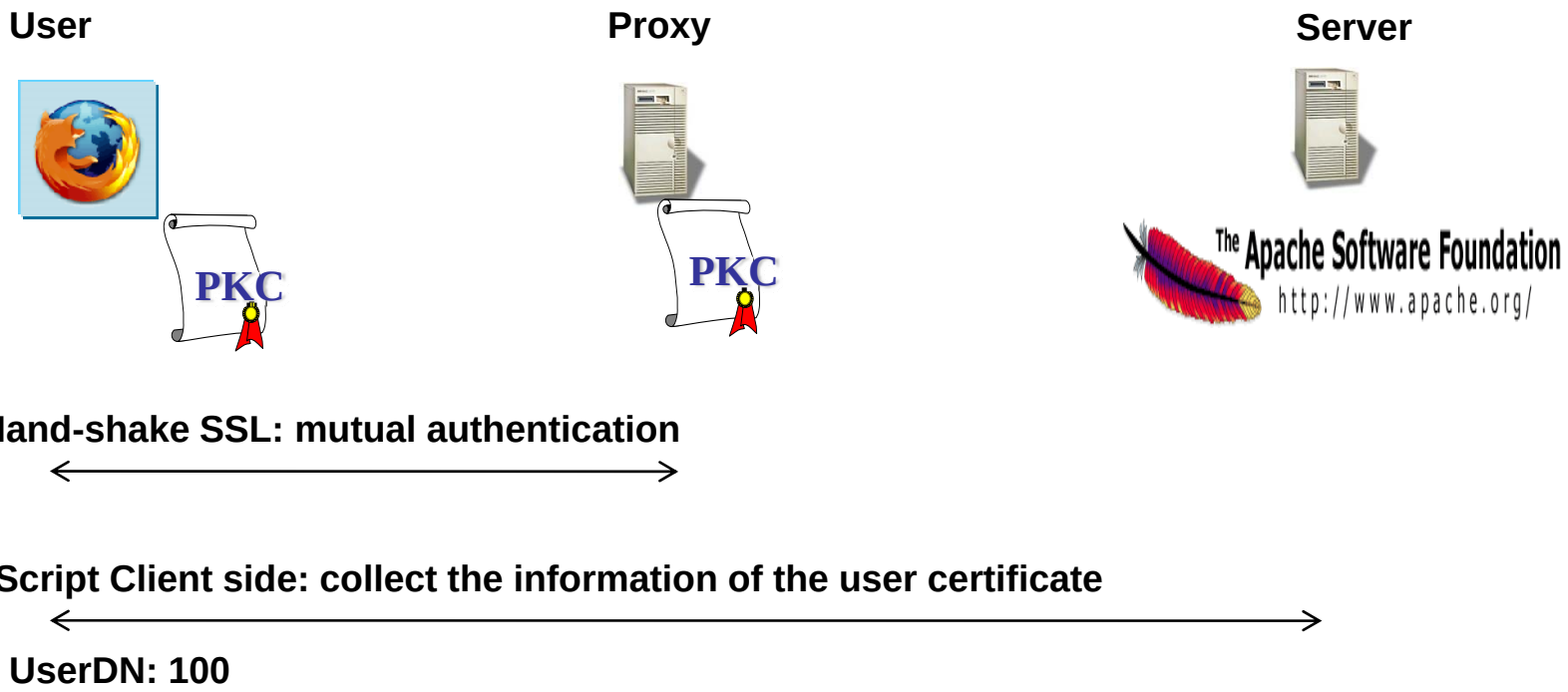
Testing the authentication scheme means understanding how the application checks for users' identity and using that information to circumvent that mechanism and access the application without having the proper credentials

Tests include the following areas:

- Credentials transport over an encrypted channel (OWASP-AT-001)
- Testing for user enumeration (OWASP-AT-002)
- Default or guessable (dictionary) user account (OWASP-AT-003)
- Testing For Brute Force (OWASP-AT-004)
- Testing for Bypassing authentication schema (OWASP-AT-005)
- Testing for Vulnerable remember password and pwd reset (OWASP-AT-006)
- Testing for Logout and Browser Cache Management (OWASP-AT-007)
- Testing for Captcha (OWASP-AT-008)
- Testing for Multiple factors Authentication (OWASP-AT-009)
- Testing for Race Conditions (OWASP-AT-010)

Testing for Bypassing Authentication Schema

- Application with mutual digital certificate authentication:



Testing for Broken Authentication (2)

Here is the Authentication POST:

```
POST https://192.168.1.1:1443/AuthenticationServlet HTTP/1.1
Host: 192.168.1.1:1443
…
Referer: https://192.168.1.1:1443/logonDN.jsp
Cookie:IV_JCT=AncDfj8439Fdfjci454;
Content-Type: application/x-www-form-urlencoded
Content-length: 44

login=true&action=MenuCommand&userDN=100
```

userDN is a value contained in the Digital Certificate . Why the Application does not take this information from the digital certificate received (once verified the CA signature and the certificate integrity)?

Authorization Testing

Authorization is the concept of allowing access to resources only to those permitted to use them. Testing for Authorization means understanding how the authorization process works, and using that information to circumvent the authorization mechanism.

Tests include the following areas:

- Testing for path traversal (OWASP-AZ-001)
- Testing for bypassing authorization schema (OWASP-AZ-002)
- Testing for Privilege Escalation (OWASP-AZ-003)

Testing for Path Traversal

Example

Testing for privilege escalation

Server Response after the user authentication

```
HTTP/1.1 200 OK
Server: Netscape-Enterprise/6.0
Date: Wed, 1 Apr 2006 13:51:20 GMT
Set-Cookie: USER=aW78ryrGuTWs4MnOd32Fs51yDqp; path=/; domain=.dom.it
Set-Cookie: SESSION=k+KmKpHXTgDi1J5fT7Zz; path=/; domain=.dom.it
Cache-Control: no-cache
Pragma: No-cache
Content-length: 247
Content-Type: text/html
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Connection: close

<form  name="autoriz" method="POST" action = "visual.jsp">
<input type="hidden" name="profile" value="SistInf1">
    <body onload="document.forms.autoriz.submit()">
 </td>
... .
```

Authorization parameter

- What if the user modifies the value SistInf1 to SistInf3?

Business logic testing

Business logic may include:

- Business rules that express business policy (such as channels, location, logistics, prices, and products); and
- Workflows based on the ordered tasks of passing documents or data from one participant (a person or a software system) to another.

This step is the most difficult to perform with automated tools, as it requires the penetration tester to perfectly understand the business logic that is (or should be) implemented by the application

Data validation testing

In this phase we test that all input is properly sanitized before being processed by the application, in order to avoid several classes of attacks

- **Cross site scripting (Reflected, Stored, DOM, Flashing)**

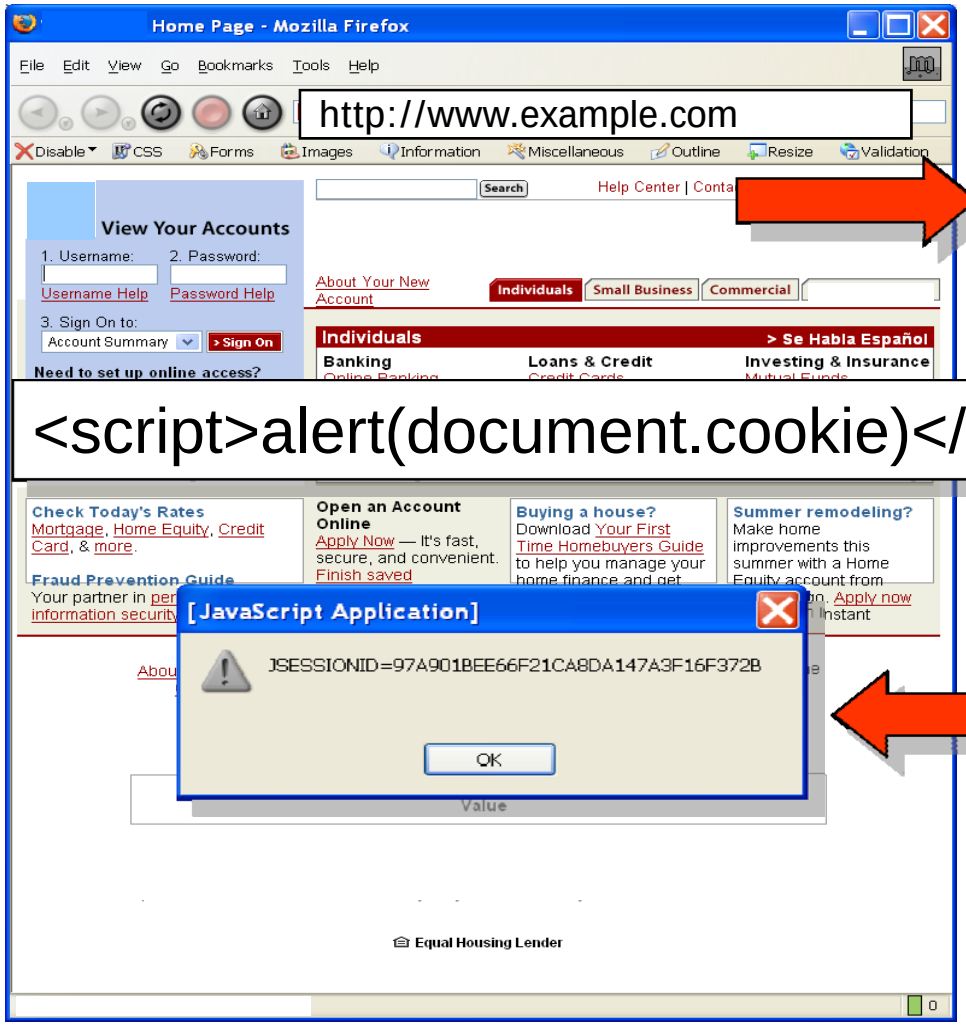
  Test that the application filters JavaScript code that might be executed by the victim in order to steal his/her cookie

- **SQL Injection**

  Test that the application properly filters SQL code embedded in the user input

- **Other attacks based of faulty input validation...**
  - LDAP/XML/SMTP/Command injection
  - Buffer overflows

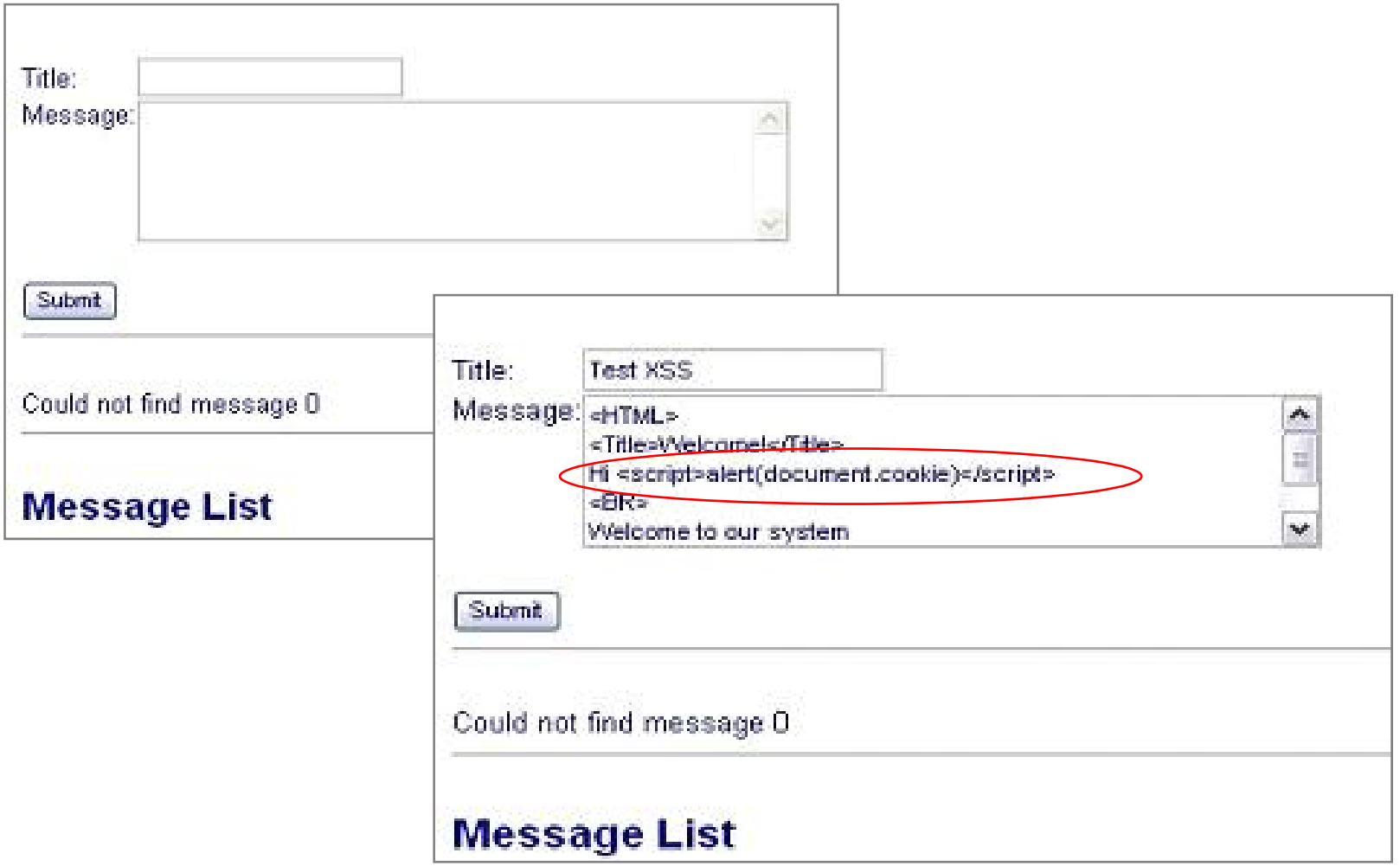
Reflected Cross Site Scripting



Search field print in output the word searched.

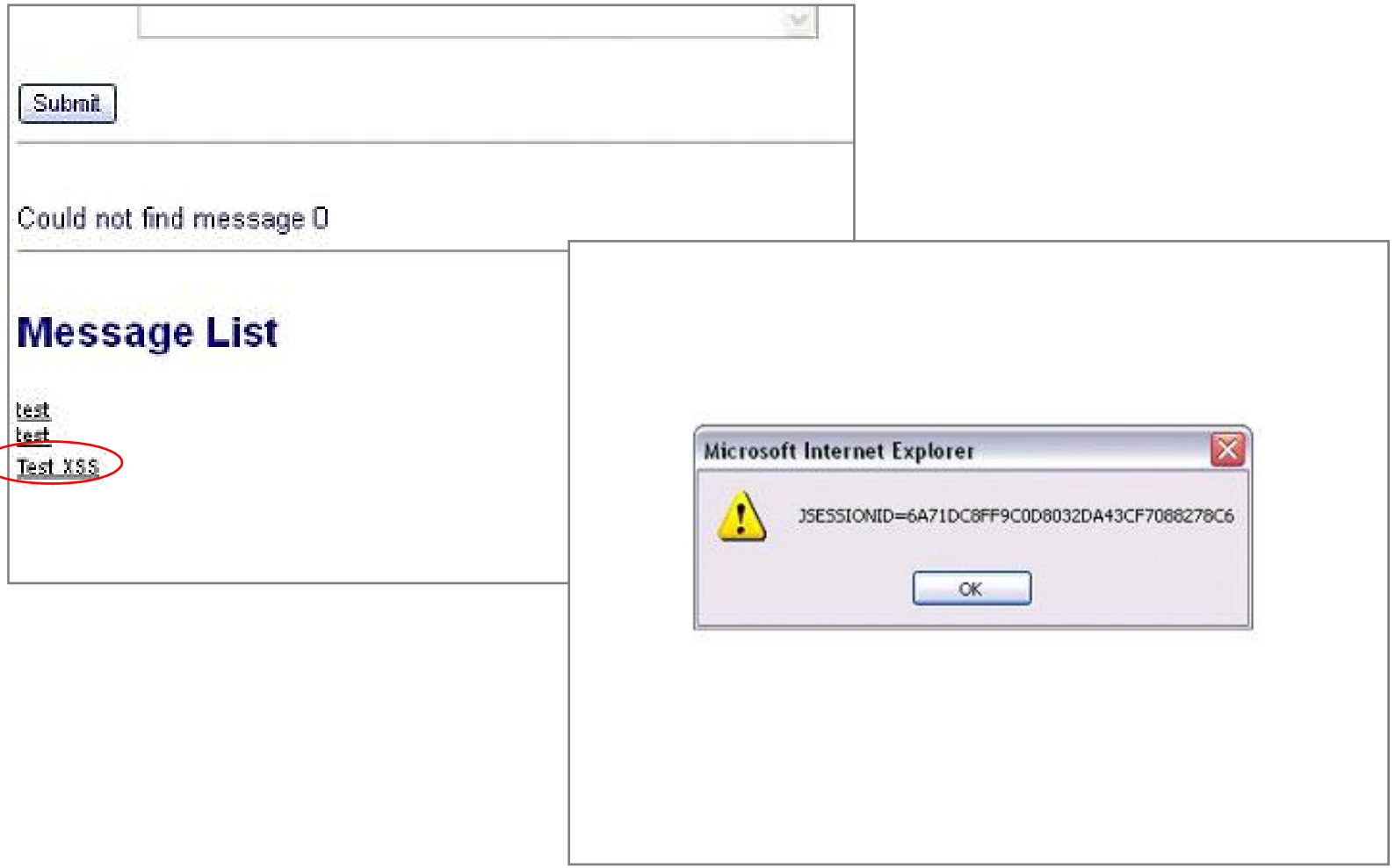
Site sends the script to the user that see his session cookie.

Stored XSS


Title:
Message:
Submit
Could not find message 0
Message List
Title:
Test XSS
Message:
<HTML>
<Title>Welcome!</Title>
Hi <script>alert(document.cookie)</script>

Welcome to our system
Submit
Could not find message 0
Message List

Stored XSS (2)


Submit
Could not find message 0
Message List
test
test
Test XSS
Microsoft Internet Explorer
JSESSIONID=6A71DC8FF9C0D8032DA43CF7088278C6
OK

ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

Testing for Command Injection

```
POST http://127.0.0.1:80/WebGoat/attack HTTP/1.1
Host: 127.0.0.1
…
HelpFile=BasicAuthentication.help | dir:
```

```
ExecResults for 'cmd.exe /c type
    "D:\Prog\WebGoat\tomcat\webapps\WebGoat\lesson_plans\"Basic
    Authentication.html | dir c:'
Output...
Il volume nell'unit? C ? WinXP
Numero di serie del volume: 1871-8F02
Directory di C:\
27/12/2007 03.51 0 AUTOEXEC.BAT
27/12/2007 03.51 0 CONFIG.SYS
18/06/2008 09.54 cygwin
18/06/2008 11.43 Dev-Cpp
…
```

Denial of Service Testing

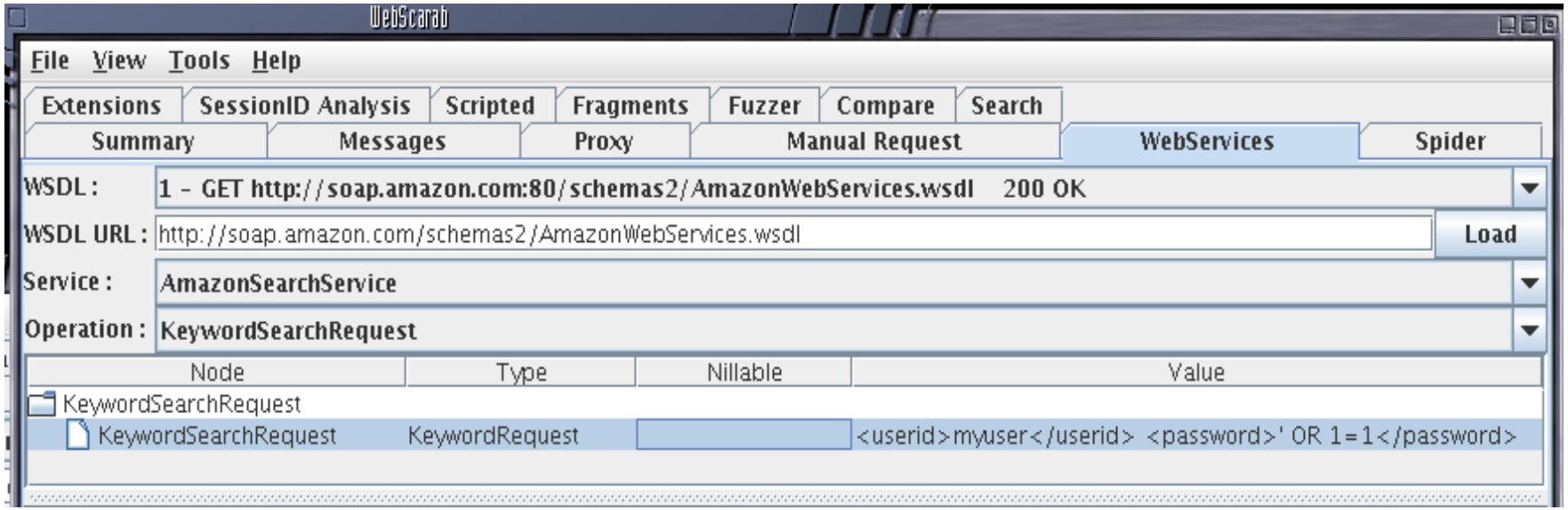
DoS are types of vulnerabilities within applications that can allow a malicious user to make certain functionality or sometimes the entire website unavailable. These problems are caused by bugs in the application, often resulting from malicious or unexpected user input

- Testing for SQL Wildcard Attacks (OWASP-DS-001)
- Locking Customer Accounts (OWASP-DS-002)
- Buffer Overflows (OWASP-DS-003)
- User Specified Object Allocation (OWASP-DS-004)
- User Input as a Loop Counter (OWASP-DS-005)
- Writing User Provided Data to Disk (OWASP-DS-006)
- Failure to Release Resources (OWASP-DS-007)
- Storing too Much Data in Session (OWASP-DS-008)

Usually not performed in performed on production environments

Web Services Testing

- The vulnerabilities are similar to other “classical” vulnerabilities such as SQL injection, information disclosure and leakage etc but web services also have unique XML/parser related vulnerabilities.
- WebScarab (available for free at www.owasp.org) provides a plug-in specifically targeted to Web Services. It can be used to craft SOAP messages that contains malicious elements in order to test how the remote system validates input



Web Services Testing

- **XML Structural Testing**

  In this example, we see a snippet of XML code that violates the hierarchical structure of this language. A Web Service must be able to handle this kind of exceptions in a secure way

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note id="666">
<to>OWASP
<from>EOIN</from>
<heading>I am Malformed </to>
</heading>
<body>Example of XML Structural Test</body>
</note>
```

Web Services Testing (cont.)

- **XML Large payload**

  Another possible attack consists in sending to a Web Service a very large payload in an XML message. Such a message might deplete the resource of a DOM parser

```
<Envelope>
<Header>
   <wsse:Security>
     <Hehehe>I am a Large String (1MB)</Hehehe>
     <Hehehe>I am a Large String (1MB)</Hehehe>
     <Hehehe>I am a Large String (1MB)</Hehehe>…
    <Signature>…</Signature>
   </wsse:Security>
 </Header>
 <Body>
   <BuyCopy><ISBN>0098666891726</ISBN></BuyCopy>
 </Body></Envelope>
```

Testing Report: model

- The OWASP Risk Rating Methodology
  - Estimate the severity of all of these risks to your business
  - This is not universal risk rating system: vulnerability that is critical to one organization may not be very important to another

- Simple approach to be tailored for every case
  - standard risk model: **Risk = Likelihood * Impact**

- Identifying a risk

  You'll need to gather information about:
  - the vulnerability involved
  - the threat agent involved
  - the attack they're using
  - the impact of a successful exploit on your business.

Testing Report: likelihood

- **Step 2: factors for estimating likelihood**

  Generally, identifying whether the likelihood is low, medium, or high is sufficient.

Threat Agent Factors:

- Skill level (0-9)
- Motive (0-9)
- Opportunity (0-9)
- Size (0-9)

Vulnerability Factors:

- Ease of discovery (0-9)
- Ease of exploit (0-9)
- Awareness (0-9)
- Intrusion detection (0-9)

Testing Report: impact

- Step 3: factors for estimating impact

Technical impact:

- Loss of confidentiality (0-9)
- Loss of integrity (0-9)
- Loss of availability (0-9)
- Loss of accountability (0-9)

Business impact:

- Financial damage (0-9)
- Reputation damage (0-9)
- Non-compliance (0-9)
- Privacy violation (0-9)

Testing Report: value the risk

- Step 4: determining the severity of the risk

Threat agent factors				Vulnerability factors			
Skill level	Motive	Opportunity	Size	Ease of discovery	Ease of exploit	Awareness	Intrusion detection
5	2	7	1	3	6	9	2
Overall likelihood=4.375 (MEDIUM)							

Technical Impact				Business Impact			
Loss of confidentiality	Loss of integrity	Loss of availability	Loss of accountability	Financial damage	Reputation damage	Non-compliance	Privacy violation
9	7	5	8	1	2	1	5
Overall technical impact=7.25 (HIGH)				Overall business impact=2.25 (LOW)			

- In the example above, the likelihood is MEDIUM, and the technical impact is HIGH, so from technical the overall severity is HIGH. **But business impact is actually LOW**, so the overall severity is best described as **LOW** as well.

Testing Report: decide what to fix

- Deciding What To Fix

  As a general rule, you should fix the most severe risks first.

  Some fix seems to be not justifiable based upon the cost of fixing the issue but may be reputation damage from the fraud that could cost the organization much more than implement a security control

- Customizing Your Risk Rating Model
  - Adding factors
  - Customizing options
  - Weighting factors

Writing Report

- **I. Executive Summary**
- **II. Technical Management Overview**
- **III Assessment Findings**
- **IV Toolbox**

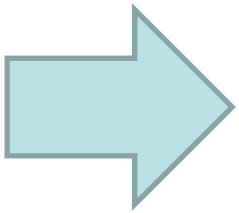
Category	Ref. Number	Name	Affected Item	Finding	Comment/Solution	Risk
Authentication Testing	OWASP-AT-003	Bypassing authentication schema				
	OWASP-AT-004	Directory traversal/file include				
	OWASP-AT-005	Vulnerable remember password and pwd reset				
	OWASP-AT-006	Logout and Browser Cache Management Testing				
Session Management	OWASP-SM-001	Session Management Schema				
	OWASP-	Session Token				

ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

Risk Rating Example

- SSL v2
- Error pages
- Access to information not authorized
- Reflected XSS
- Stored XSS
- Session Fixation
- Cross Site Request Forgery

R=(P,I)



ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

How the Guide will help the security industry

Pen-testers

- A structured approach to the testing activities
- A checklist to be followed
- A learning and training tool

Clients

- A tool to understand web vulnerabilities and their impact
- A way to check the quality of the penetration tests they buy

More in general, the Guide aims to provide a pen-testing standard that creates a 'common ground' between the pen-testing industry and its client.

This will raise the overall quality and understanding of this kind of activity and therefore the general level of security in our infrastructures

ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

Status and Future Steps

- Discuss how to integrate the Develop, Code Review, Testing and ASDR Guide
- Improve Client Side Security
- You should adopt this guide in your organization

ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

Thanks!

V3 Authors

- Anurag Agarwwal
- Daniele Bellucci
- Arian Coronel
- Stefano Di Paola
- Giorgio Fedon
- Alan Goodman
- Christian Heinrich
- Kevin Horvath
- Gianrico Ingrosso
- Roberto Suggi Liverani
- Alex Kuza
- Pavol Luptak
- Ferruh Mavituna
- Marco Mella
- Matteo Meucci
- Marco Morana
- Antonio Parata
- Cecil Su
- Harish Skanda Sureddy
- Mark Roxberry
- Andrew Van der Stock

V3 Reviewers

- Marco Cova
- Nam Nguyen
- Kevin Fuller
- Matteo Meucci

Thank you!

Matteo Meucci
matteo.meucci@owasp.org
matteo.meucci@mindedsecurity.com

ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

References

OWASP Italy:
www.owasp.org/index.php/Italy

OWASP Italy mailing list:
http://lists.owasp.org/mailman/listinfo/owasp-italy

OWASP Guidelines:
http://www.owasp.org/index.php/Category:OWASP_Guide_Project
http://www.owasp.org/index.php/Category:OWASP_Code_Review_Project
http://www.owasp.org/index.php/Category:OWASP_Testing_Project
http://www.opensamm.org