ADDO

ALL DAY DEVOPS

NOVEMBER 6, 2019

Matteo Meucci

# OWASP Testing Guide

ADDO
ALL DAY DEVOPS

# OWASP Testing Guide

**Matteo Meucci - OWASP Testing Guide co-lead**


ADDO
ALL DAY DEVOPS

# AGENDA

**(1) OWASP Introduction**

1.1 Software Security Scenario

1.2 OWASP Guidelines and tools for Software Security

**(2) OWASP Testing Guide**

2.1 Project History

2.2 Inside the guide

2.3 How to use the guide

**(3) DevSecOps**

3.1 Is the report useful?

3.2 Security bugs integrated in lifecycle

3.3 Real scenario: manual with high level skills

**(4) Conclusions**

4.1 What do we take home

ADDO
ALL DAY DEVOPS

# Who am I?

Informatics Engineer (since 2001)

Research:

- OWASP contributor (since 2002)
- OWASP-Italy Chair (since 2005)
- OWASP Testing Guide Lead (since 2006)
- OWASP Sw Security 5D Framework Lead (since 2018)

Work: **https://www.mindedsecurity.com**

MINDED SECURITY
The Software Security Company

- 18+ years on Information Security focusing on Software Security
- CEO @ Minded Security – The Software Security Company (since 2007)

# (1) OWASP Introduction

## 1.1 Software Security scenario

ADDO
ALL DAY DEVOPS

# Software Security Scenario

User

How can I understand that I am using a secure software?

ADDO
ALL DAY DEVOPS

# Software Security Scenario

How can I understand that I am using a secure software?

Is it possible to develop Secure Software?

Dev

ADDO
ALL DAY DEVOPS

# Software Security Scenario

How can I understand that I am using a secure software?

Is it possible to develop Secure Software?

How can I measure the level of security of my software?

**Company**

# Insecure Software

So I understand that we are living in the era of insecure software: what is the best approach today?

# (1) Introduction to OWASP

# 1.2 OWASP Guidelines and tools

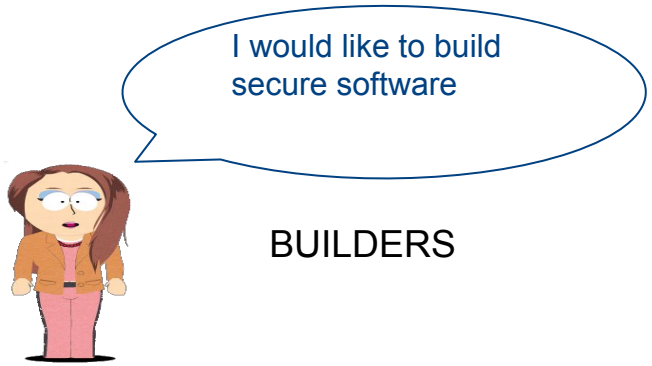
# Why OWASP?

Fight with the same weapons (knowledge)

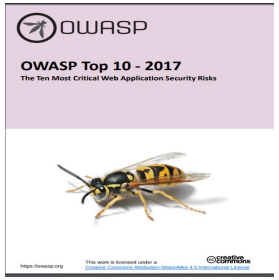
# OWASP has ~140 Projects

- PROTECT - These are tools and documents that can be used to guard against security-related design and implementation flaws.
- DETECT - These are tools and documents that can be used to find security-related design and implementation flaws.
- LIFE CYCLE - These are tools and documents that can be used to add security-related activities into the Software Development Life Cycle (SDLC).

# OWASP: The Open Web Application Security Project




BUILDERS


# OWASP: The Open Web Application Security Project

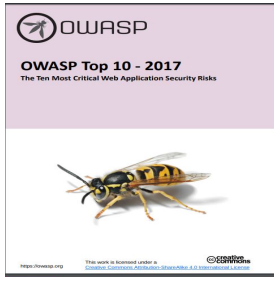
I would like to build secure software

BUILDERS

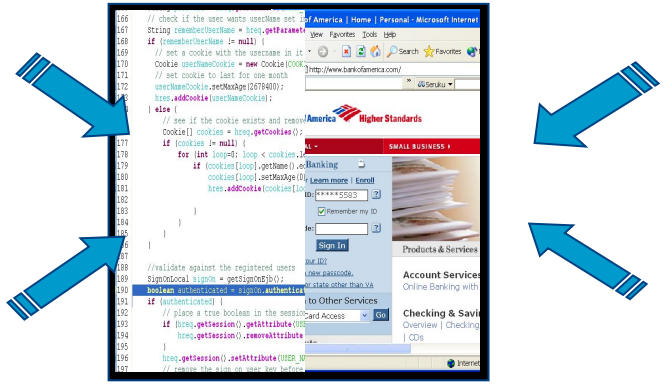
I would like to find all the security bugs in my software

BREAKERS

OWASP Pro Active CONTROLS





# OWASP: The Open Web Application Security Project

I would like to build secure software

BUILDERS

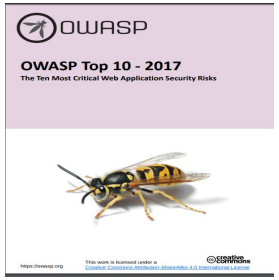
I would like to find all the security bugs in my software

BREAKERS

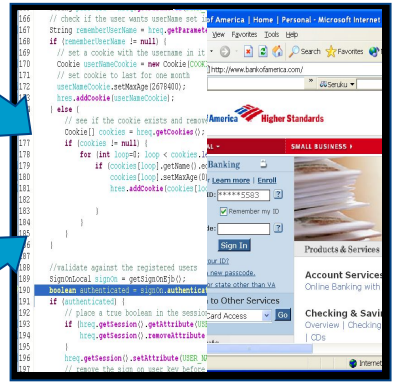
I would like to implement a Roadmap for Software Security

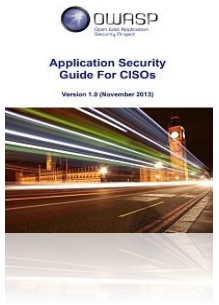
MANAGERS

OWASP Pro Active CONTROLS







# (2) OWASP Testing Guide

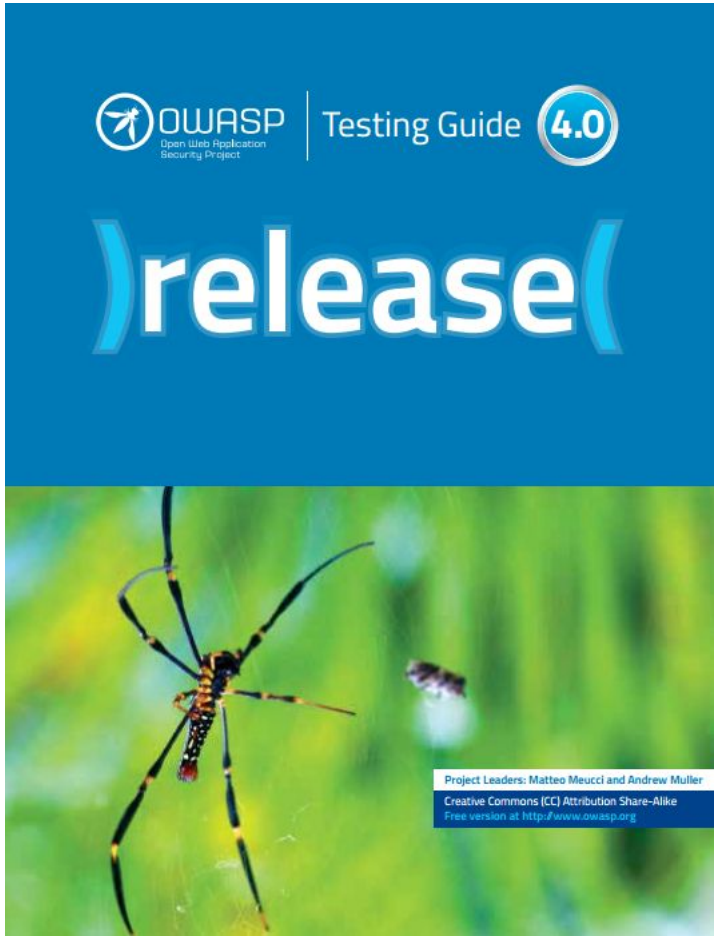
## 2.1 Project History

# The OWASP Testing Guide: Community driven for all the Enterprises

# OWASP Testing Guide

- Most comprehensive open source secure testing guide on the web
- Years of development effort
- Version 4.0 produced in 2014
- Hundreds of contributors
- Project Leader and Editor v4
  - Matteo Meucci, Andrew Muller

matteo.meucci@owasp.org,
andrew.muller@owasp.org




**www.owasp.org/index.php/Testing_Guide**

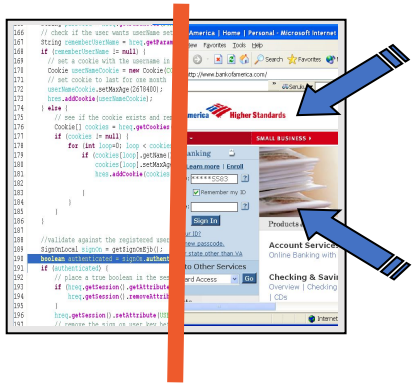
# OWASP Testing Guide History

- July 14, 2004
  - "OWASP Web Application Penetration Checklist", V1.0
- December 25, 2006
  - "OWASP Testing Guide", V 2.0
- December 16, 2008
  - "OWASP Testing Guide", V 3.0
- September 17, 2014
  - "OWASP Testing Guide", V 4.0
- 2019 - V 5.0 draft
  - GitHub migration, New OWASP TG co-leaders: Eliee Sad, Rick Mitchell
  - Migration from OWASP Wiki to Github (100% done) https://github.com/OWASP/wstg

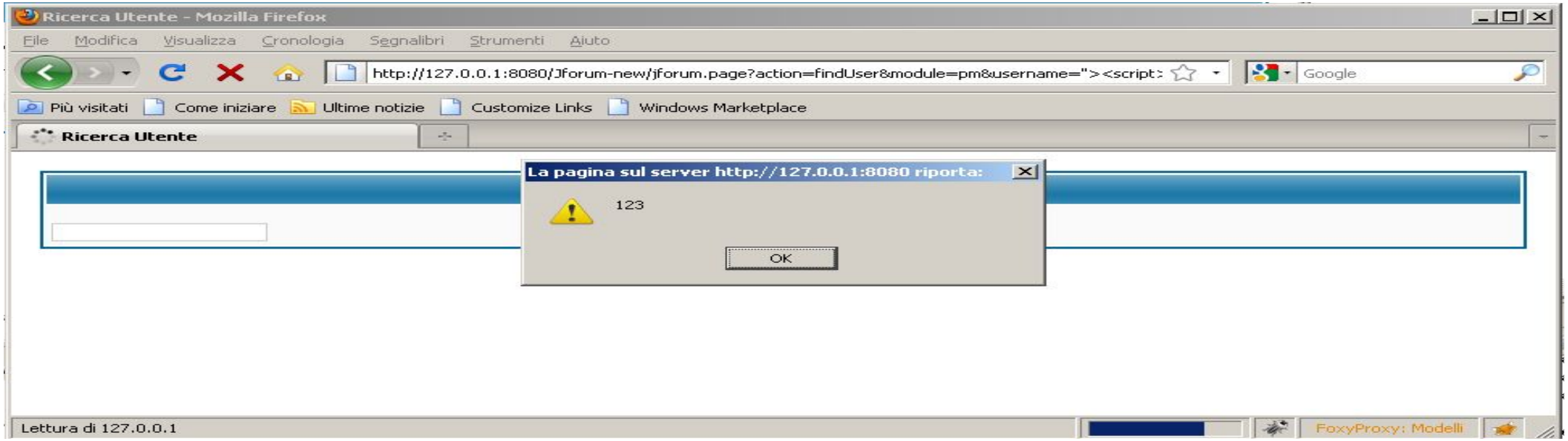
**Citations:**


- NIST SP800-115 "Technical Guide to Information Security Testing and Assessment"
- Gary McGraw (former CTO Cigital) says: "In my opinion it is the strongest piece of Intellectual Property in the OWASP portfolio" – OWASP Podcast by Jim Manico
- NSA's "Guidelines for Implementation of REST"
- Official (ISC)2 Guide to the CSSLP - Page: 70, 365
- Many books, blogs and websites

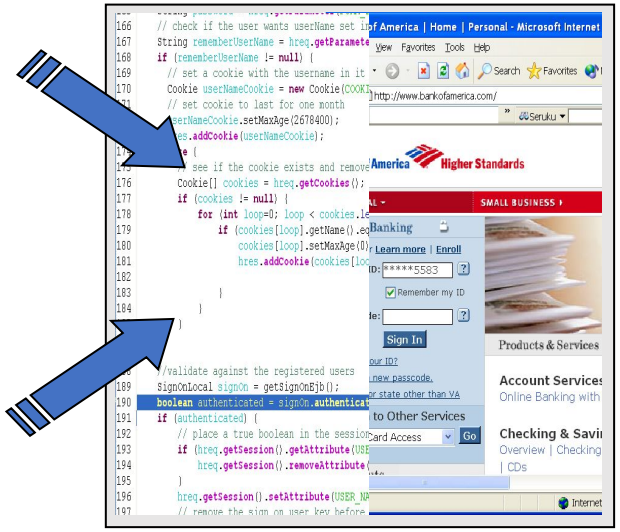
# TESTING GUIDE



http://127.0.0.1:8080/Jforum-new/jforum.page?action=findUser&module=pm&**usernam**
**e=%22%3E%3Cscript%3Ealert%28123%29%3C/script%3E%3C%22**



# OWASP CODE REVIEW GUIDE



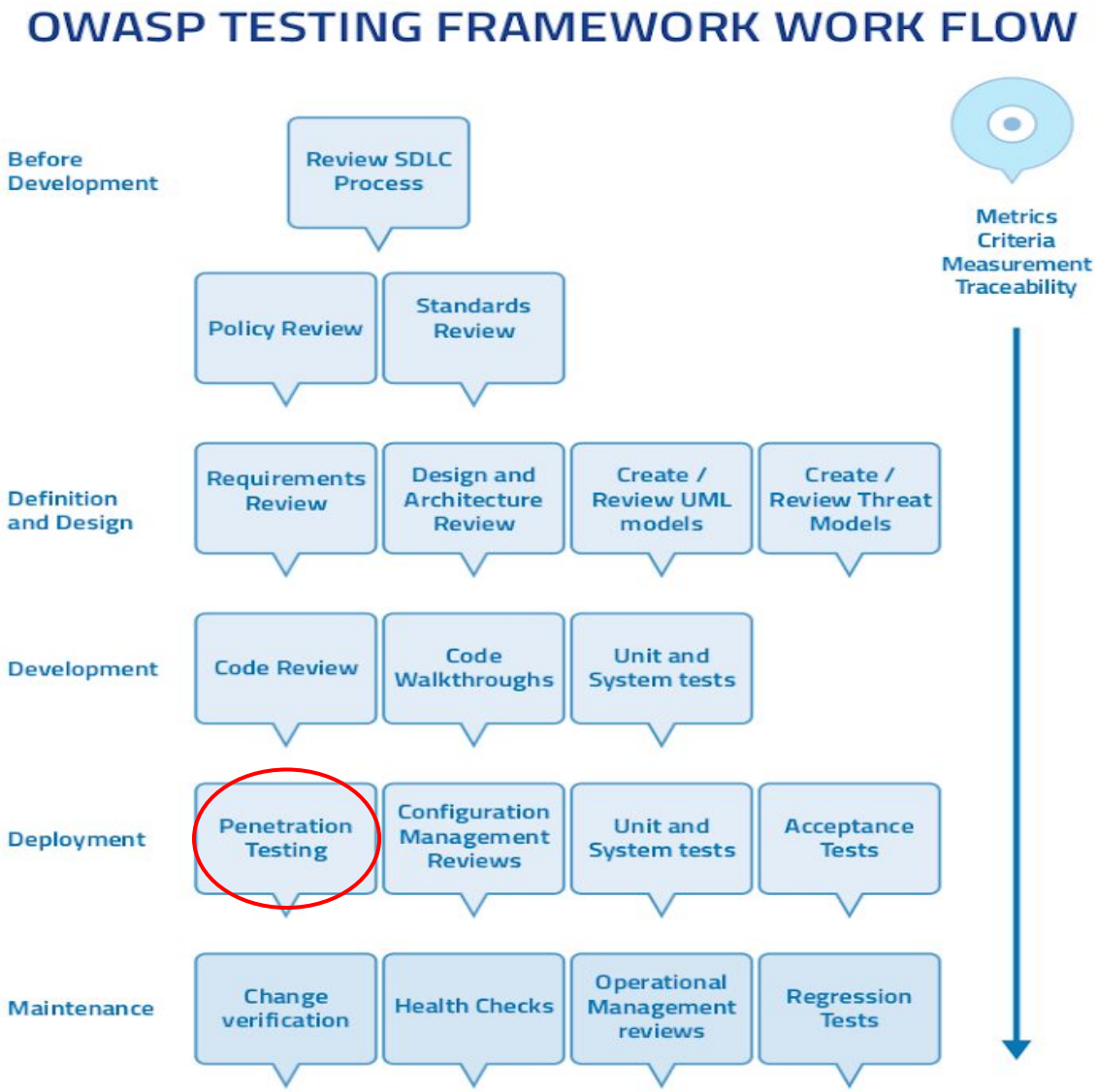
```
1    public void findUser()
2    {
3        boolean showResult = false;
4        String username =  this.request.getParameter("username");
5    ...
6        this.context.put("username", username);
7        this.context.put("showResult", showResult);
8    }
```

# (2) OWASP Testing Guide

## 2.2 The methodology

# Inside the Guide

- The OWASP Testing Framework
  (today OWASP SAMM, OWASP 5D)



# Inside the Guide

- 11 sub-categories:
  - **Information Gathering**
  - **Configuration and Deployment Management Testing**
  - **Identity Management Testing**
  - **Authentication Testing**
  - **Authorization Testing**
  - **Session Management Testing**
  - **Input Validation Testing**
  - **Error Handling**
  - **Cryptography**
  - **Business Logic Testing**
  - **Client Side Testing**
- A total of 87 controls

# Inside the Guide

- 11 sub-categories:
  - **Information Gathering**
  - **Configuration and Deployment Management Testing**
  - **Identity Management Testing**
  - **Authentication Testing**
  - **Authorization Testing**
  - **Session Management Testing**
  - **Input Validation Testing**
  - **Error Handling**
  - **Cryptography**
  - **Business Logic Testing**
  - **Client Side Testing**
- A total of 87 controls

**4.8 Input Validation Testing**

**4.8.1 Testing for Reflected Cross Site Scripting (OTG-INPVAL-001)**

**4.8.2 Testing for Stored Cross Site Scripting (OTG-INPVAL-002)**

**4.8.3 Testing for HTTP Verb Tampering (OTG-INPVAL-003)**

**4.8.4 Testing for HTTP Parameter pollution (OTG-INPVAL-004)**

**4.8.5 Testing for SQL Injection (OTG-INPVAL-005)**

**4.8.5.1 Oracle Testing**

**4.8.5.2 MySQL Testing**

**4.8.5.3 SQL Server Testing**

**4.8.5.4 Testing PostgreSQL (from OWASP BSP)**

**4.8.5.5 MS Access Testing**

**4.8.5.6 Testing for NoSQL injection**

**4.8.6 Testing for LDAP Injection (OTG-INPVAL-006)**

**4.8.7 Testing for ORM Injection (OTG-INPVAL-007)**

**4.8.8 Testing for XML Injection (OTG-INPVAL-008)**

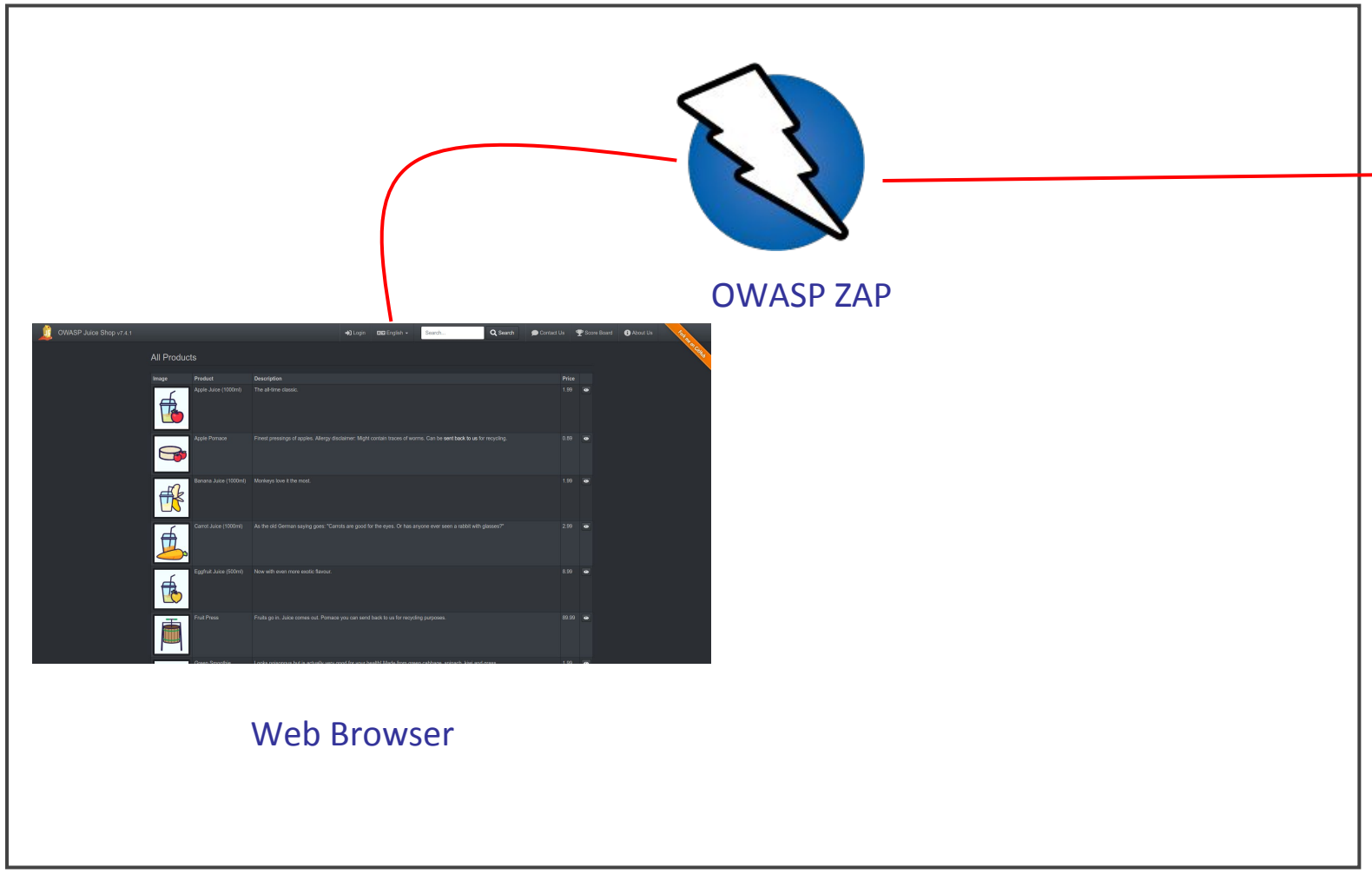
**4.8.9 Testing for SSI Injection (OTG-INPVAL-009)**

**4.8.10 Testing for XPath Injection (OTG-INPVAL-010)**

**4.8.11 IMAP/SMTP Injection (OTG-INPVAL-011)**

**…. and more….**

# Tools: OWASP ZAP and many more



# Testing for XSS: example

- Description
- How to test
- Examples
- Expected results
- Tool to use
- References

**How to Test**

**Black Box testing**

A black-box test will include at least three phases:

**[1]** Detect input vectors. For each web page, the tester must determine all the web application's user-defined variables and how to input them. This includes hidden or non-obvious inputs such as HTTP parameters, POST data, hidden form field values, and predefined radio or selection values. Typically, in-browser HTML editors or web proxies are used to view these hidden variables. See the example below.

**[2]** Analyze each input vector to detect potential vulnerabilities. To detect an XSS vulnerability, the tester will typically use specially crafted input data with each input vector. Such input data is typically harmless, but trigger responses from the web browser that manifests the vulnerability. Testing data can be generated by using a web application fuzzer, an automated predefined list of known attack strings, or manually.

Some example of such input data are the following:

```
<script>alert(123)</script>
```

```
"><script>alert(document.cookie)</script>
```

For a comprehensive list of potential test strings, see the XSS Filter Evasion Cheat Sheet.

**[3]** For each test input attempted in the previous phase, the tester will analyze the result and determine if it represents a vulnerability that has a realistic impact on the web application's security. This requires examining the resulting web page HTML and searching for the test input. Once found, the tester identifies any special characters that were not properly encoded, replaced, or filtered

```
\" (double quote)
\\ (backslash)
\uXXXX (unicode values)
```

For a more complete reference, see the Mozilla JavaScript guide. [2]

**Example 1**

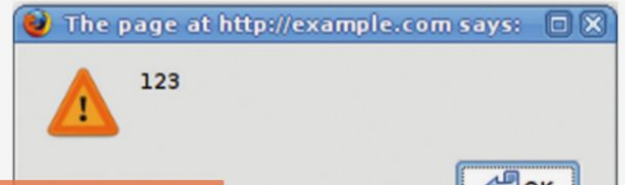
For example, consider a site that has a welcome notice " Welcome %username% " and a download link.

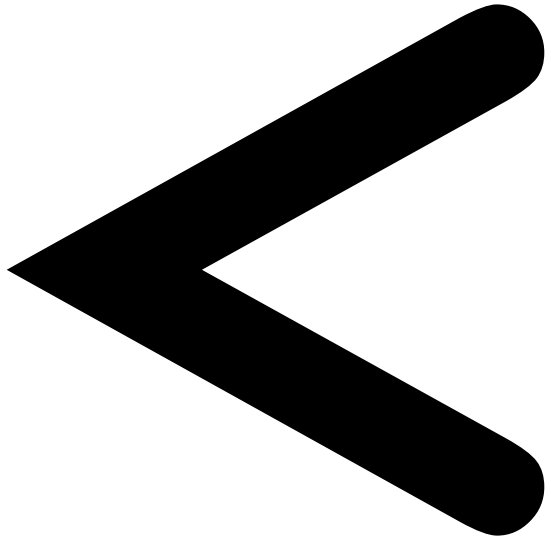

The tester must suspect that every data entry point can result in an XSS attack. To analyze it, the tester will play with the user variable and try to trigger the vulnerability.

Let's try to click on the following link and see what happens:

```
http://example.com/index.php?user=<script>alert(123)</
script>
```

If no sanitization is applied this will result in the following popup:





ADDO
ALL DAY DEVOPS

# Cross Site Scripting (XSS): description

- Cross Site Scripting attacks aim to the execution of code inside the interpreters present in the browser. Common interpreters are: Javascript, Java Applet, Flash

- Browsers interpret different subset of metacharacters within the page. In particular the characters "<" and ">" form elements called tags.

  **<script>alert(document.cookie> </script>** → Executed in the JS Engine

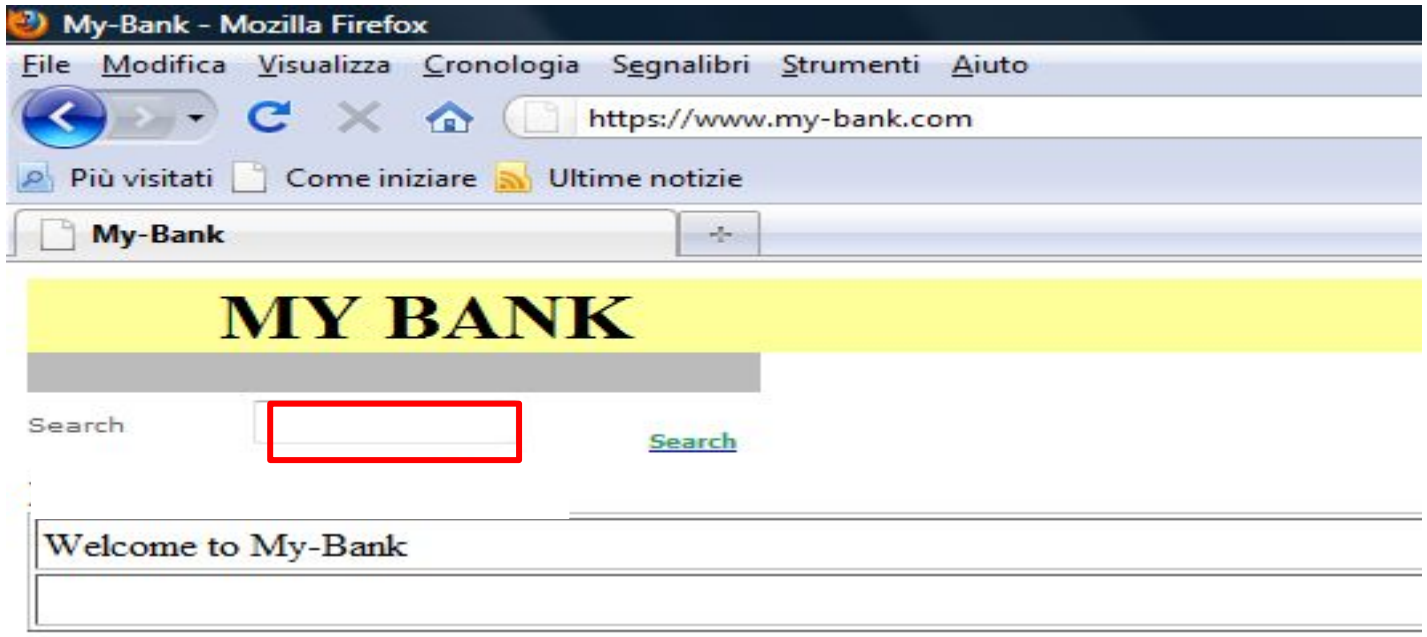
  **&ltscript&gtalert(document.cookie)&lt/script&gt** → Not executed

# Cross Site Scripting: how to test

Mario Rossi

**-- Testing fox XSS in “search”field --**

My-Bank



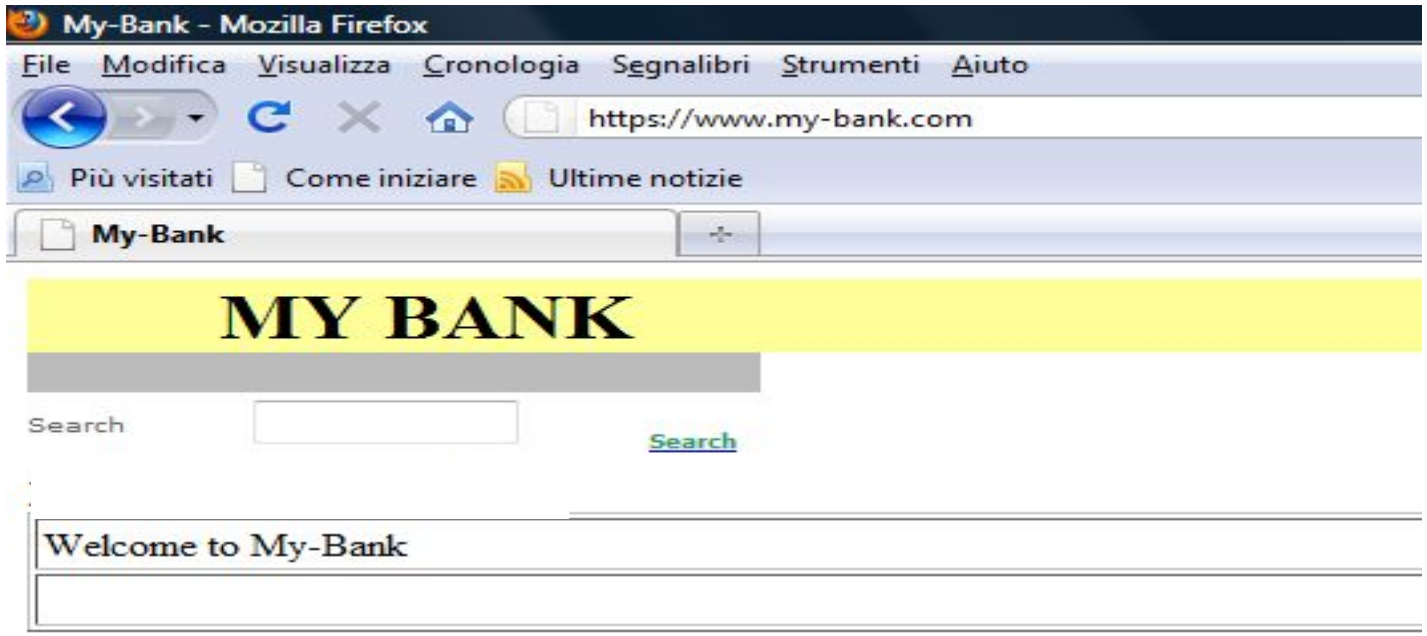
# Cross Site Scripting: how to test

Mario Rossi

**-- Testing fox XSS in "search"field --**

My-Bank

[1] search "XSSTest"



# Cross Site Scripting: how to test

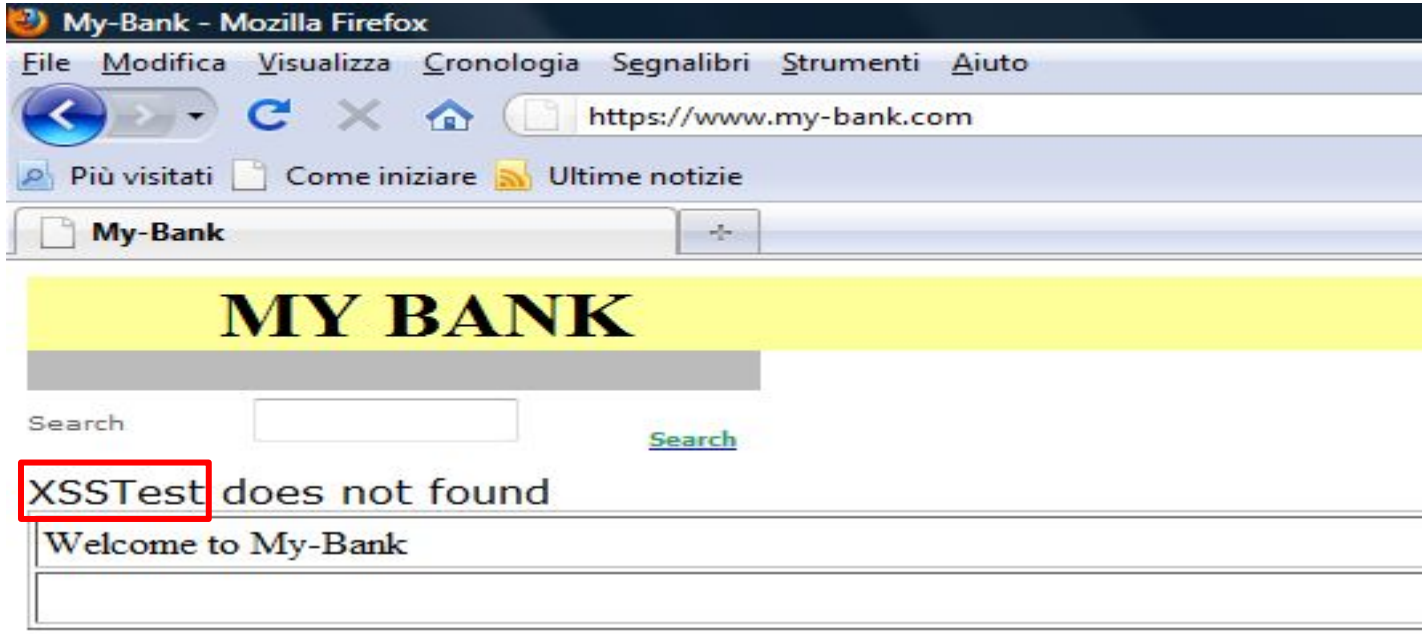
Mario Rossi

**-- Testing fox XSS in "search"field --**

My-Bank

[1] search "XSSTest"

[2] Answer: "XSSTest does not found"



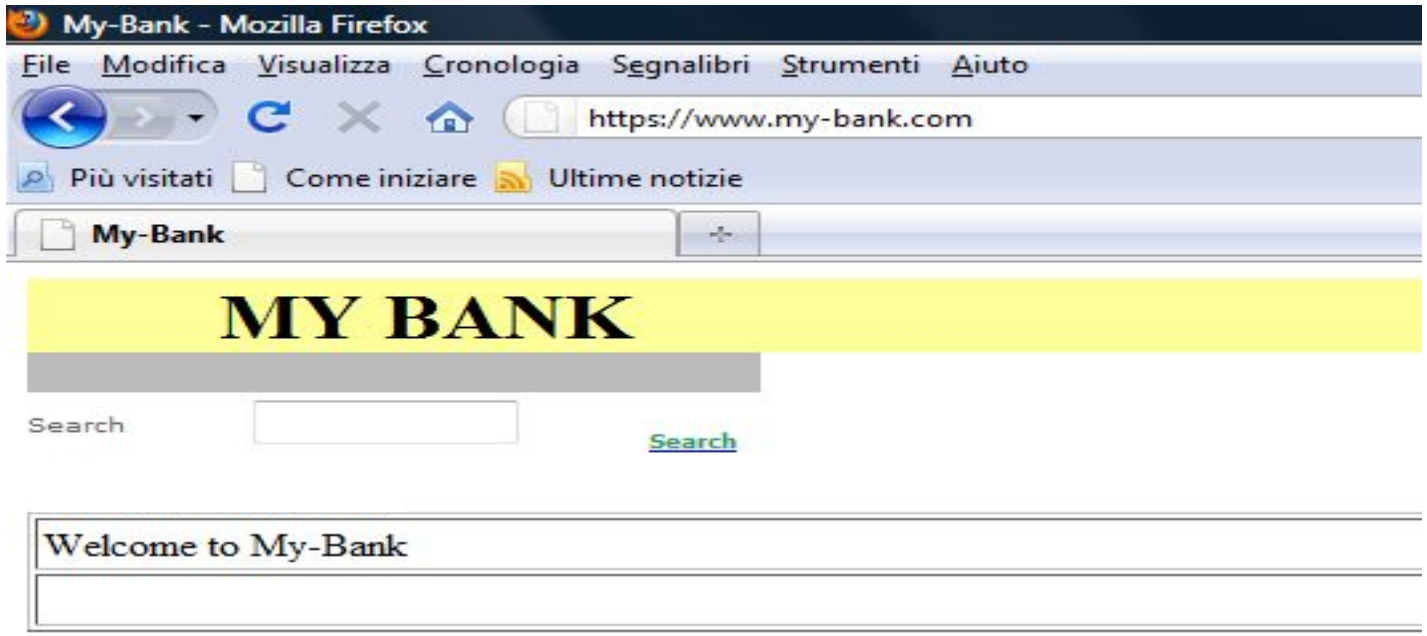
# Cross Site Scripting: how to test

Mario Rossi

**-- Testing fox XSS in "search"field --**

My-Bank

[1] search: "<script>alert(document.cookie)</script>"



# Cross Site Scripting: how to test

Mario Rossi

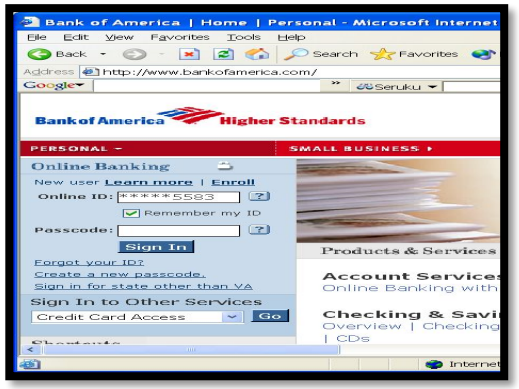
**-- Testing fox XSS in "search"field --**

My-Bank

[1] search: "<script>alert(document.cookie)</script>"

[2] Answer: "<script> does not found"

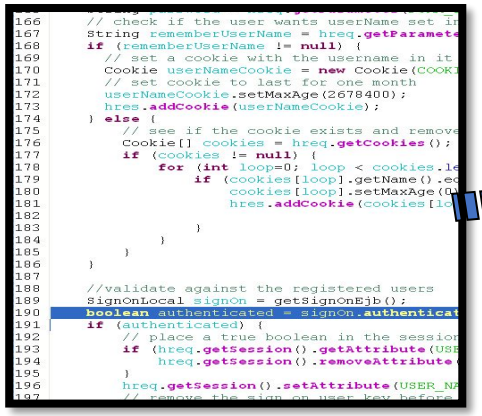
# How to use the methodology




Web Application


Methodology


Report



Source Code

```
public void findUser()
{ boolean showResult = false;
String username =
this.request.getParameter("username");
...
this.context.put("username",
ESAPI.encoder().encodeForHTMLAttribute
(username));

this.context.put("showResult",
showResult);

        }
```

Fixing


Methodology

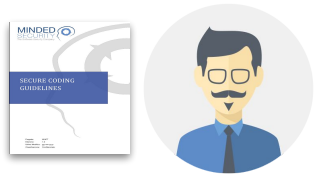

Retest Report

# (3) DevSecOps

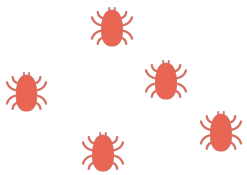
## 3.1 Is the report useful today?


12:34

# Yesterday: too much time to remediate

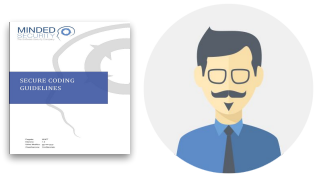



Manager



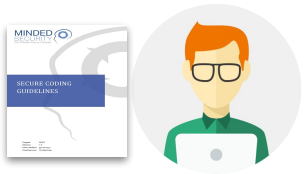
Time

# Yesterday: too much time to remediate

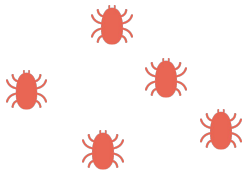


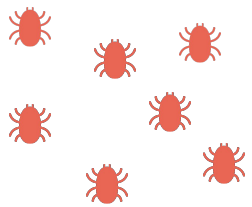

Manager

1w to 1 month



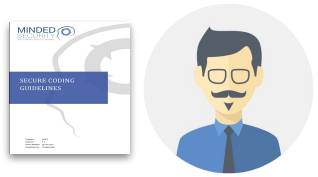

Dev team





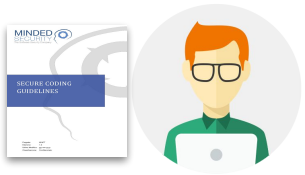
Time

# Yesterday: too much time to remediate



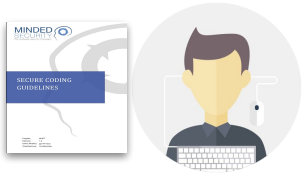

Manager

1w to 1 month

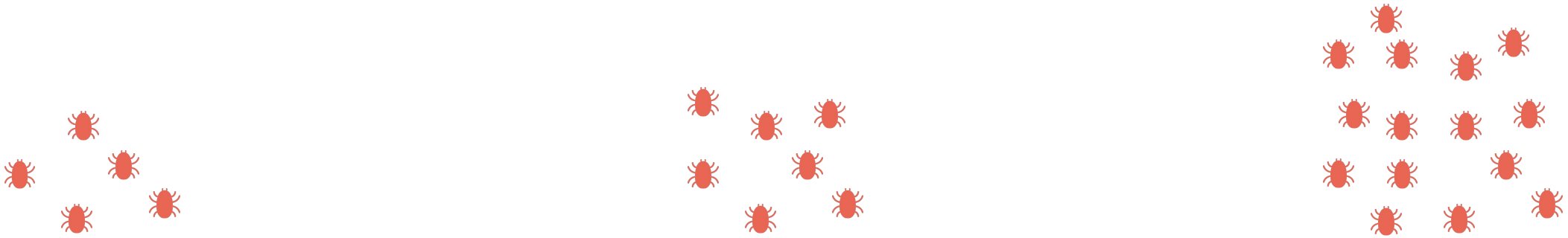



Dev team

2ws to 6 months




Fixing



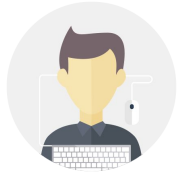
Time

# What we need today?

1. We need instant security feedback
2. Security must be shared

Manager

Dev team



Fixing

AppSec
specialists

ADDO
ALL DAY DEVOPS

# Some real data...

| COMPANY | DEPLOY FREQUENCY | DEPLOY LEAD TIME | RELIABILITY | CUSTOMER RESPONSIVENESS |
|---|---|---|---|---|
| Amazon | 23,000/day | minutes | high | high |
| Google | 5,500/day | minutes | high | high |
| Netflix | 500/day | minutes | high | high |
| Facebook | 1/day | minutes | high | high |
| Twitter | 3/week | minutes | high | high |
| Typical enterprise | once every 9 months | months or quarters | low/medium | low/medium |

*Source: XebiaLab: ITRev_DevOps_Guide_5_2015*

# Testing activity: automation or manual

```
public void findUser()
{

boolean showResult = false;
String username =
this.request.getParameter("username
");

this.context.put("username",
username);
this.context.put("showResult",
showResult);

}
```

**Software**

**AppSec Specialist in action or tool**

**There is a Vulnerability!**

**Pervasive security**

ADDO
ALL DAY DEVOPS

# MANUAL VERIFICATION STILL MATTER

Automate everything!

Security Automation!

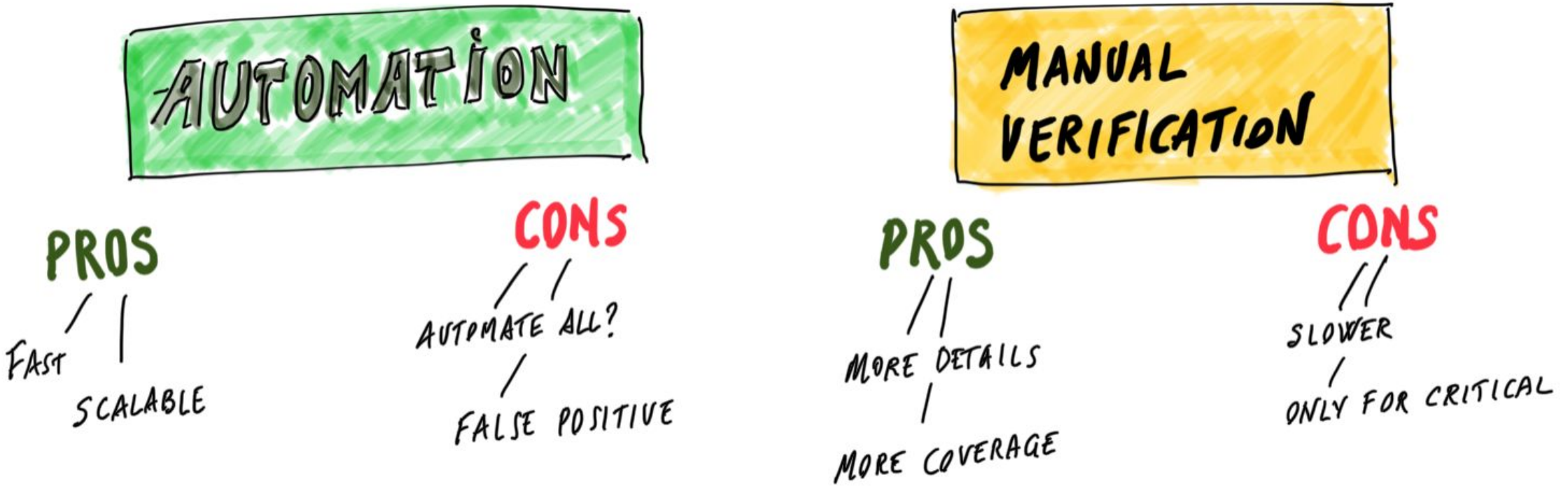
Does it really work?

# MANUAL VERIFICATION STILL MATTER

Automate everything!

Security Automation!

Does it really work?



# Tools Vs Manual Testing

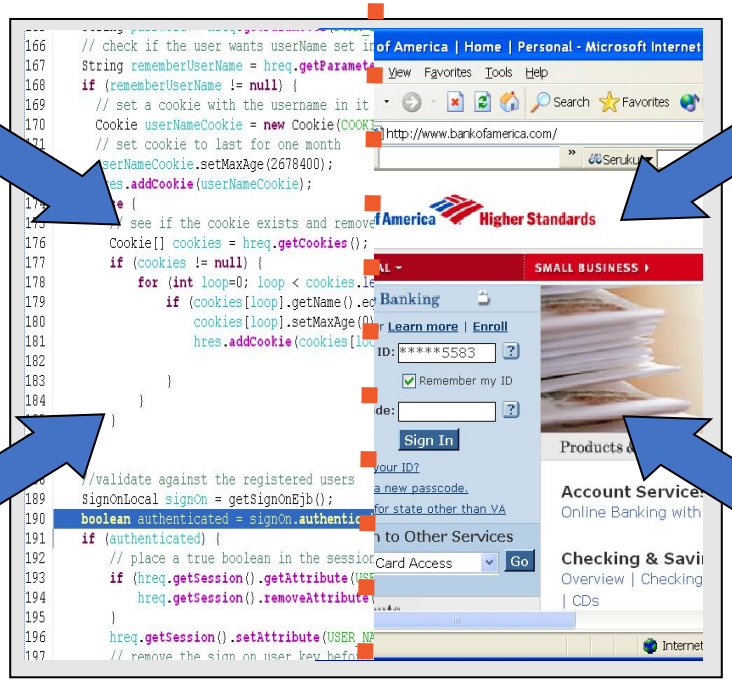
**Source Code (White Box Testing)**

**Running application (Black Box Testing)**

**The adoption of all the 4 techniques produces better results**

**Manual Code Review**

**Manual Penetration Testing**




**Automated Static Code Analysis**

**Automated Vulnerability Scanning**

# Tools Vs Manual Testing

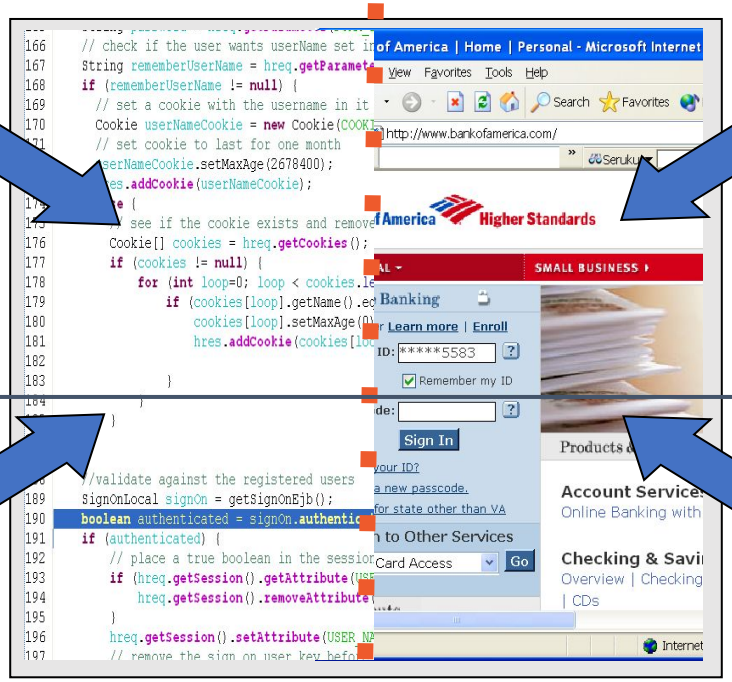
**Source Code (White Box Testing)**

**Running application (Black Box Testing)**

**The adoption of all the 4 techniques produces better results**

**Manual Code Review**

**Manual Penetration Testing**




**Automated Static Code Analysis**

**Automated Vulnerability Scanning**

**Baseline**

# SwSec - Testing dimension (OWASP SwSec 5D Framework)

**DevSecOps**

Code → Build → Test → Stage → Deploy → Monitor

**SAST - Static Application Security Testing**

**DAST - Dynamic Application Security Testing**

**IAST - Interactive Application Security Testing**

**RASP - Runtime Application Self Protection**

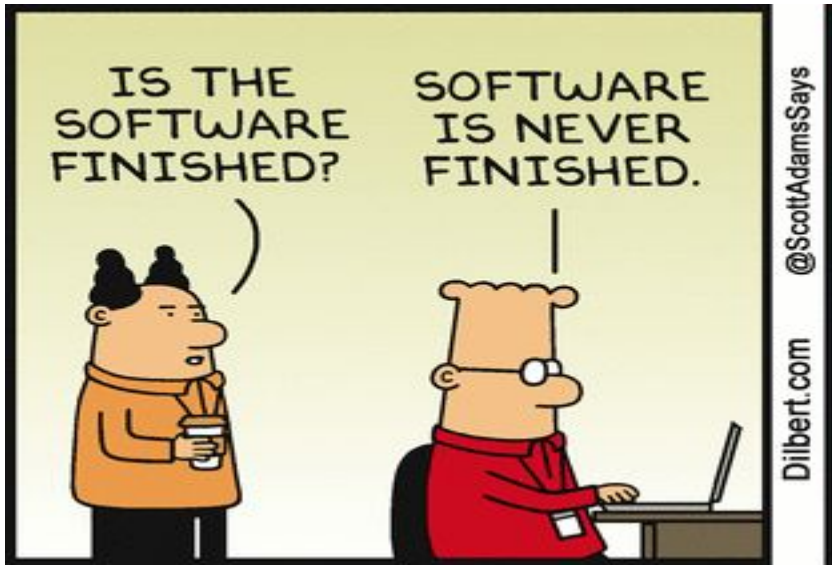
**Manual Secure Code Review (SCR)**


**Manual Penetration Testing**

# (4) Conclusions

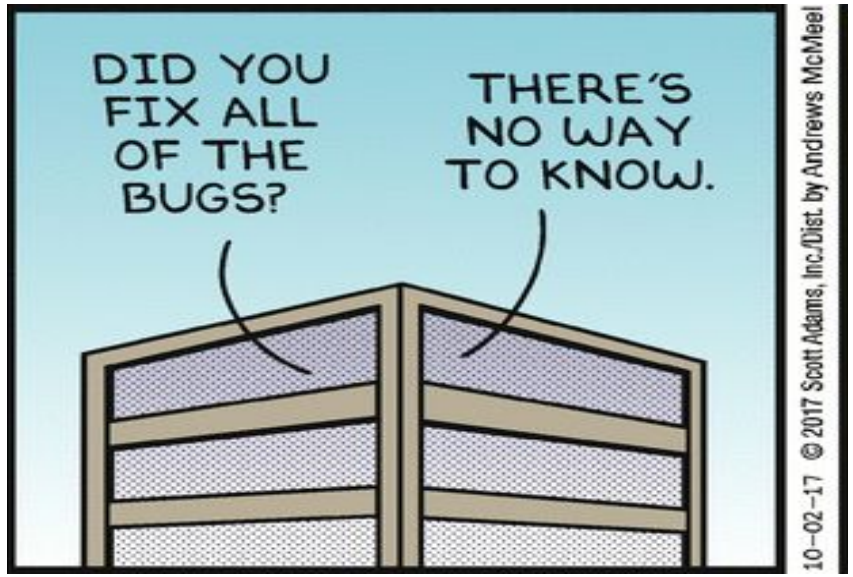
## 4.1 How to use the OWASP Testing Guide and OWASP Resources

# Testing is the solution to SwSec?



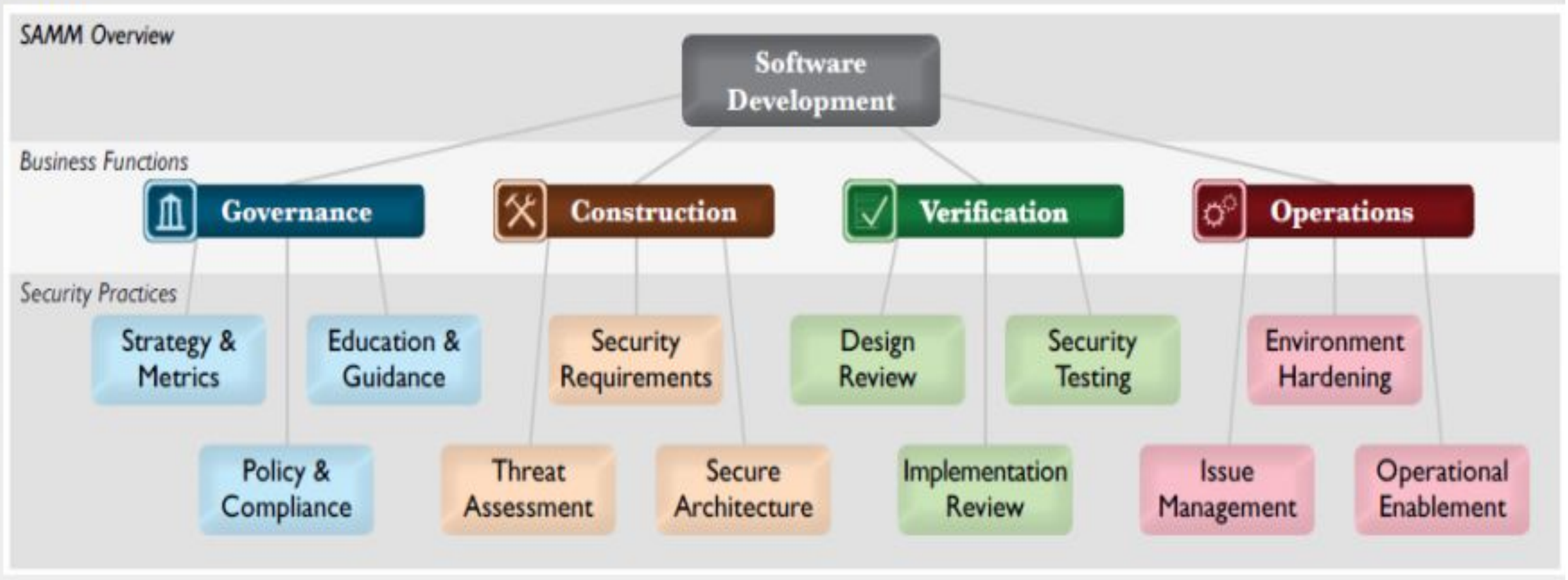

Testing is just one of AppSec dimensions

Fixing ASAP is the most important aspect of AppSec



ADDO
ALL DAY DEVOPS

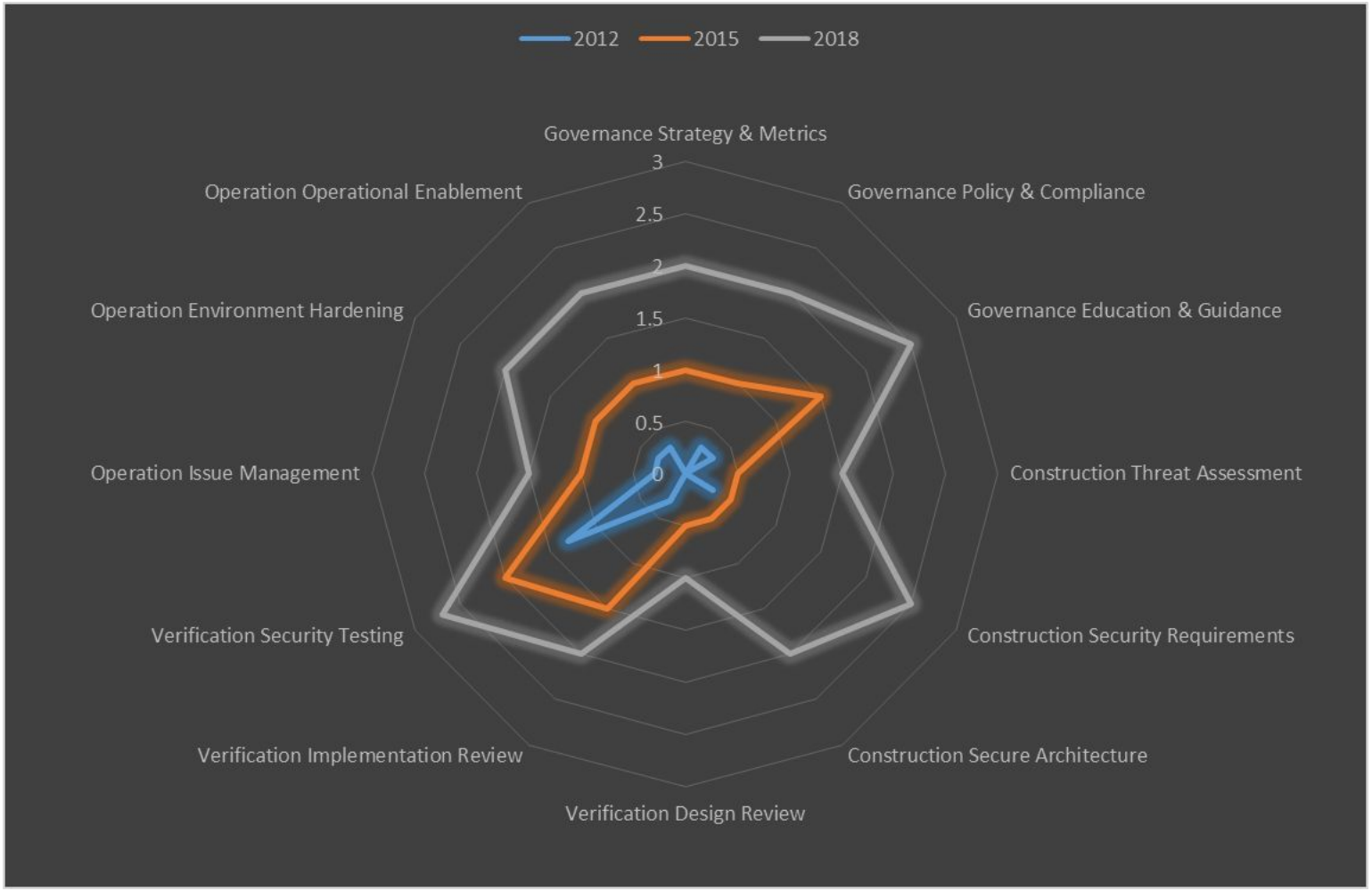
# OWASP SAMM: 12 Security Practices



# What is wrong with the Testing Guide?

Testing is NOT the solution of Software Security

Testing is just a part of your Software Security journey




Source: Minded Security - Assessment on 15 Companies

ADDO
ALL DAY DEVOPS

# Conclusions: what do we take home?

- Stop thinking swsec is only testing! Testing Guide is a piece of the puzzle
- Stop thinking swsec is a developers responsabilities

# Conclusions: what do we take home?

- Stop thinking swsec is only testing! Testing Guide is a piece of the puzzle
- Stop thinking swsec is a developers responsabilities

- Start sharing security bugs
- Fix flaws and bugs asap

# Conclusions: what do we take home?

- Stop thinking swsec is only testing! Testing Guide is a piece of the puzzle
- Stop thinking swsec is a developers responsabilities

- Start sharing security bugs
- Fix flaws and bugs asap

- OWASP Testing Guide supports company to identify vulnerabilities in your software.
- Adopt others OWASP Standard such as OWASP SAMM and 5D Framework that allows you to create a Software Security program and a 360 vision of the Software Security.

ADDO
ALL DAY DEVOPS

# Questions?

Go to our slack channel for questions!

**@Matteo_Meucci**

**matteo.meucci@owasp.org**

**https://www.owasp.org**

Thanks!

**References:**

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

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

https://www2.owasp.org/www-project-juice-shop/

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

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

https://owasp.org/index.php/OWASP_DevSlop_Project

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

https://www.owasp.org/index.php/OWASP_Software_Security_5D_Framework

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

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