VI Venice AppSeC Conference

In collaborazione con

Venezia, Università Ca’ Foscari
5 Ottobre 2018
Matteo Meucci

OWASP SwSec 5D Framework
1. Software Security Introduction
2. SDLC frameworks: how OWASP can help on software security
3. OWASP Software Security 5 Dimension Framework
4. Apply the models to a real case
5. Some wrong approaches
6. Conclusions
Who Am I?

Informatics Engineer (since 2001)

Research:
• OWASP contributor (since 2002)
• OWASP-Italy Chair (since 2005)
• OWASP Testing Guide Lead (since 2006)

Work:
• 17+ years on Information Security focusing on Software Security
• CEO @ Minded Security – The Software Security Company (since 2007)
What is the subject today

Secure Design

SDLC Costs to fix

Automation

Software Security Governance

DevOps

Tools

OWASP

SAST, DAST

Processes
1. Software Security Intro
Software Security: a brief history

From: Bill Gates
Sent: Tuesday, January 15, 2002 5:22 PM
To: to every full-time employee at Microsoft
Subject: Trustworthy computing

...new capabilities is the fact that it is designed from the ground up to deliver Trustworthy Computing.
Traditional SDLC (Software Development Life Cycle)

Traditional Software Development Life Cycle has the following phases:

- **Define**: definition of the software specifications and requirements;
- **Design**: identifies general requirements for design;
- **Develop**: software development phase;
- **Deploy**: software goes in production;
- **Maintain**: software change management.
Costs related to fixing a vulnerability

Sources: Official (ISC)2 Guide to CSSLP (2011)
A Software Development Life Cycle (SDLC) is a framework that defines the process used by organizations to build an application from its inception to its decommission. Over the years, multiple standard SDLC models have been proposed (Waterfall, Iterative, Agile, etc.) and used in various ways to fit individual circumstances.
2006: Microsoft Security Development Lifecycle

The Security Development Lifecycle (SDL) is a software development process that helps developers build more secure software and address security compliance requirements while reducing development cost.

Microsoft created the first SDLC: built internally for MS software. Released it for public in 2006.

<table>
<thead>
<tr>
<th>1. TRAINING</th>
<th>2. REQUIREMENTS</th>
<th>3. DESIGN</th>
<th>4. IMPLEMENTATION</th>
<th>5. VERIFICATION</th>
<th>6. RELEASE</th>
<th>7. RESPONSE</th>
</tr>
</thead>
</table>
SDL Microsoft is an example of a complete and efficient SDLC implementation that perfectly fit to Microsoft SDLC.

The Microsoft SDL Process Guidance illustrates the way Microsoft applies the SDL to its own technologies and software. Each organization being unique, it is important that you determine your own security requirements and which tools are appropriate for your organization.

![Microsoft products: Vulnerabilities reduction after SDL implementation](image)

Sources: Microsoft Security Blog and Microsoft TechNet Security Blog
2. How OWASP can help on software security
The Open Web Application Security Project (OWASP) is a 501c3 not-for-profit also registered in Europe as a worldwide charitable organization focused on improving the security of software.

Our mission is to make application security visible, so that people and organizations can make informed decisions about true application security risks.

Everyone is welcomed to participate in OWASP and all of our materials are available under free and open software licenses.
I would like to build secure software

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

I would like to implement a Roadmap for Software Security
The Software Assurance Maturity Model (SAMM) is an open framework to help organizations formulate and implement a strategy for software security that is tailored to the specific risks facing the organization. It represents the open standard worldwide to create an internal SDLC and a roadmap to improve it.
Case Studies: 2012-2015-2018

Source: Minded Security 2018
3. OWASP Software Security 5

Dimension Framework
Traditional SDLC is not enough

Traditional SDLC frameworks lack of:

– level of awareness
– security team
– security standards
– security testing tools

Minded Security has develop a new and more practical framework that focus on 5 dimensions to evaluate the maturity of a Software Security framework and now have:

OWASP Software Security 5D framework
**SwSec PROCESSES**
- Risk Assessment - Security Requirements
- Threat Modeling - Security Design
- SCR, WAPT
- Software Acceptance - Security bug Fixing

**Sw Sec TESTING**
- SAST, DAST, IAST, RASP
- External manual SCR, WAPT

**Sw Sec TEAM**
- AppSec manager/CISO, Sec Champions,
  AppSec Specialists, Satellite Architects, Sat
  Developers, Sat Testers

**Sw Sec AWARENESS**
- Management, Application Owners, Analysts,
  Auditors, Architects, Developers, Engineers

**Sw Sec STANDARDS**
- Sw Security Roadmap (SAMM) - Risk analysis -
  Threat Modeling - Sec Architecture - Sec Coding -
  Software Assurance
(1) SwSec 5D - Process dimension

- **Risk Assessment**
  - Strumento decisionale per le attività da svolgere

- **Secure Requirements**
  - Linee guida di sviluppo sicuro per ogni ambiente di sviluppo

- **Threat Modeling**
  - Delinea le caratteristiche di sicurezza del software

- **Secure Design**
  - Documento di design rivisto dal punto di vista della sicurezza

---

**DEFINE**

- **Secure Code Review**
  - Review del codice

- **Web App Penetration Testing**
  - Review dell’applicazione running

---

**DEVELOP**

- **Fixing**

---

**DEPLOYMENT**

- **Software Assurance/Retest**
  - Verifica che tutte le practice siano state implementate

- **Secure Monitoring**
  - Verifica del change

---

Source: Minded Security
(2) SwSec5D - Testing dimension

- **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**
A fast fixing process is the key to have a mature SwSec Program:

- Satellite architects: should fix flaws asap
- Satellite developers: should fix bugs asap
- Satellite tester: should test if the remediations are strong enough asap.

A strong satellite is the key of a mature software security initiative.
(4) SwSec 5D - Awareness dimension

Source: Official (ISC)2 Guide to CSSLP (2012)
(5) SwSec - Standards dimension

Sw Security Roadmap (SAMM)

Risk analysis

Secure Software Requirements

Threat modeling use cases

Secure Architecture

Secure Coding Guidelines

Software Assurance
4. Apply the models to a real case
SDLC phase 1

The first implementations of a Secure SDLC started in 2007. The fist results appears as the following:

1) basic training
2) generic secure coding guidelines
3) buy a tool
4) test at the end

Basic trainings: 8hs webinar on xss, sqli, major attacks
what about building specific secure software?
what about management? analysts, auditors? Architects?

Basic guidelines:
what about how to implement
secure authentication in J2EE?
SDLC phase 1: MS SLD

- Core Security Training
  - Establish Security Requirements
  - Create Quality Gates / Bug Bars
  - Security & Privacy Risk Assessment
- Design
  - Establish Design Requirements
  - Analyze Attack Surface
  - Threat Modeling
- Implementation
  - Use Approved Tools
  - Deprecate Unsafe Functions
  - Static Analysis
- Verification
  - Dynamic Analysis
  - Fuzz Testing
  - Attack Surface Review
- Release
  - Incident Response Plan
  - Final Security Review
  - Release Archive
- Response
  - Execute Incident Response Plan
## SDLC phase 1: OWASP SAMM

<table>
<thead>
<tr>
<th>Business Functions</th>
<th>Security Practices</th>
<th>Maturity Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Governance</td>
<td>Strategy &amp; Metrics</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Policy &amp; Compliance</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Education &amp; Guidance</td>
<td>0,3</td>
</tr>
<tr>
<td>Construction</td>
<td>Threat Assessment</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Security Requirements</td>
<td>0,3</td>
</tr>
<tr>
<td></td>
<td>Secure Architecture</td>
<td>0</td>
</tr>
<tr>
<td>Verification</td>
<td>Design Review</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Code Review</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Security Testing</td>
<td>0,6</td>
</tr>
<tr>
<td>Deployment</td>
<td>Vulnerability Management</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Environment Hardening</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Operational Enablement</td>
<td>0</td>
</tr>
</tbody>
</table>
SDLC phase 1: OWASP SwSec 5D

ISACA - Software Security 5D - Assessment Report - 03-10-2018

Software Security 5D Maturity Model
5. Some wrong approaches
1) buy a tool and your software will be secure!

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

**Software**

**Security tool in action**

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

**Secure Software**
public void findUser()
{
    boolean showResult = false;
    String username = this.request.getParameter("username");
    this.context.put("username", username);
    this.context.put("showResult", showResult);
}

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

The model lacks of strategy, skill sets to configure the tools, skill
sets to manage the results
2) SECURE SDLC phase 0: Test at the end

A gate to control *at the end* of the development?
2) SECURE SDLC phase 0: Test at the end

A gate to control at the end of the development?

Software is ready and it is coming to the gate...
3) Use the right tools for DevOps

(1) In DevOps only SAST but used by a developer on a set of applications.

(2) In DevOps SAST, DAST, IAST, RASP, manual SCR and WAPT are adopted but more than 1y to fix
SwSec 5D: When it fails

SwSec PROCESSES
- Risk Assessment - Security Requirements
- Threat Modeling - Security Design
- SCR, WAPT
- Software Acceptance - Security bug Fixing

SwSec TESTING
- SAST, DAST, IAST, RASP
- External manual SCR, WAPT

SwSec TEAM
- AppSec manager/CISO, Sec Champions, AppSec Specialists, Satellite Architects, Sat Developers, Sat Testers

SwSec AWARENESS
- Management, Application Owners, Analysts, Auditors, Architects, Developers, Engineers

SwSec STANDARDS
- Sw Security Roadmap (SAMM) - Risk analysis - Threat Modeling - Sec Architecture - Sec Coding - Software Assurance
6. Conclusions
Conclusions

- Building a Secure Software Roadmap for SDL is a journey
- The journey is different for each company.
- SDLC frameworks like MS SDL helps to see what MS is doing
- OWASP SAMM helps to manage all the aspects of the path of the roadmap
- OWASP SwSec 5D helps to understand that not only processes and tools are the focus of Companies roadmap.

Companies need to see the maturity growing in 5 Dimensions
**WELCOME**

**Introduction**

Welcome to the OWASP Day 2018, Cagliari Edition Conference. Following on from the great successes of last OWASP Days, the new conference will take place next **19th October 2018 at the University of Cagliari**. Address: Auditorium of the Faculty of Engineering and Architecture, Piazza d’Armi, Cagliari.

---

<table>
<thead>
<tr>
<th>Time</th>
<th>Session Title</th>
<th>Speaker/Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:00h</td>
<td>&quot;Welcome and opening of the works&quot;</td>
<td>Prof. Giorgio Giacinto, Ing. Davide Ariu - Università di Cagliari, Matteo Meucci OWASP Italy</td>
</tr>
<tr>
<td>10:45h</td>
<td>&quot;API Security (or insecurity)&quot;</td>
<td>Marco Pacchiardo</td>
</tr>
<tr>
<td>11:15h</td>
<td>&quot;Let me introduce you the Owasp Mobile App Security Testing: How to test your mobile applications against security vulnerabilities.&quot;</td>
<td>Giuseppe Porcu, Minded Security</td>
</tr>
<tr>
<td>11:45h</td>
<td>Coffee Break</td>
<td></td>
</tr>
<tr>
<td>12:00h</td>
<td>&quot;Are you focusing on the root causes? A Unified Framework for Web Security&quot;</td>
<td>Dr. Igino Corona, Computer Security Researcher, Co-Founder &amp; Security CTO at Pluribus One</td>
</tr>
<tr>
<td>12:45h</td>
<td>&quot;How we turned spaghetti (code) into mHackeroni&quot;</td>
<td>Marco Festa, POLIMI</td>
</tr>
<tr>
<td>13:10h</td>
<td>Light lunch</td>
<td></td>
</tr>
</tbody>
</table>
Thanks!

MATTEO.MEUCCI@owasp.org

https://twitter.com/matteo_meucci

www.OWASP.org