Utiliser SonarQube pour la Sécurité

Application Security Forum West Switzerland

6 Novembre 2014

Yverdon les bains

Sébastien Gioria
Sebastien.Gioria@owasp.org
Chapter Leader & Evangelist OWASP France

OWASP
The Open Web Application Security Project

OWASP
The Open Web Application Security Project

http://www.google.fr/#q=sebastien gioria

- Innovation and Technology @Advens && Application Security Expert
- OWASP France Leader & Founder & Evangelist,
- OWASP ISO Project & OWASP SonarQube Project Leader
- Application Security group leader for the CLUSIF
- Proud father of youngs kids trying to hack my digital life.

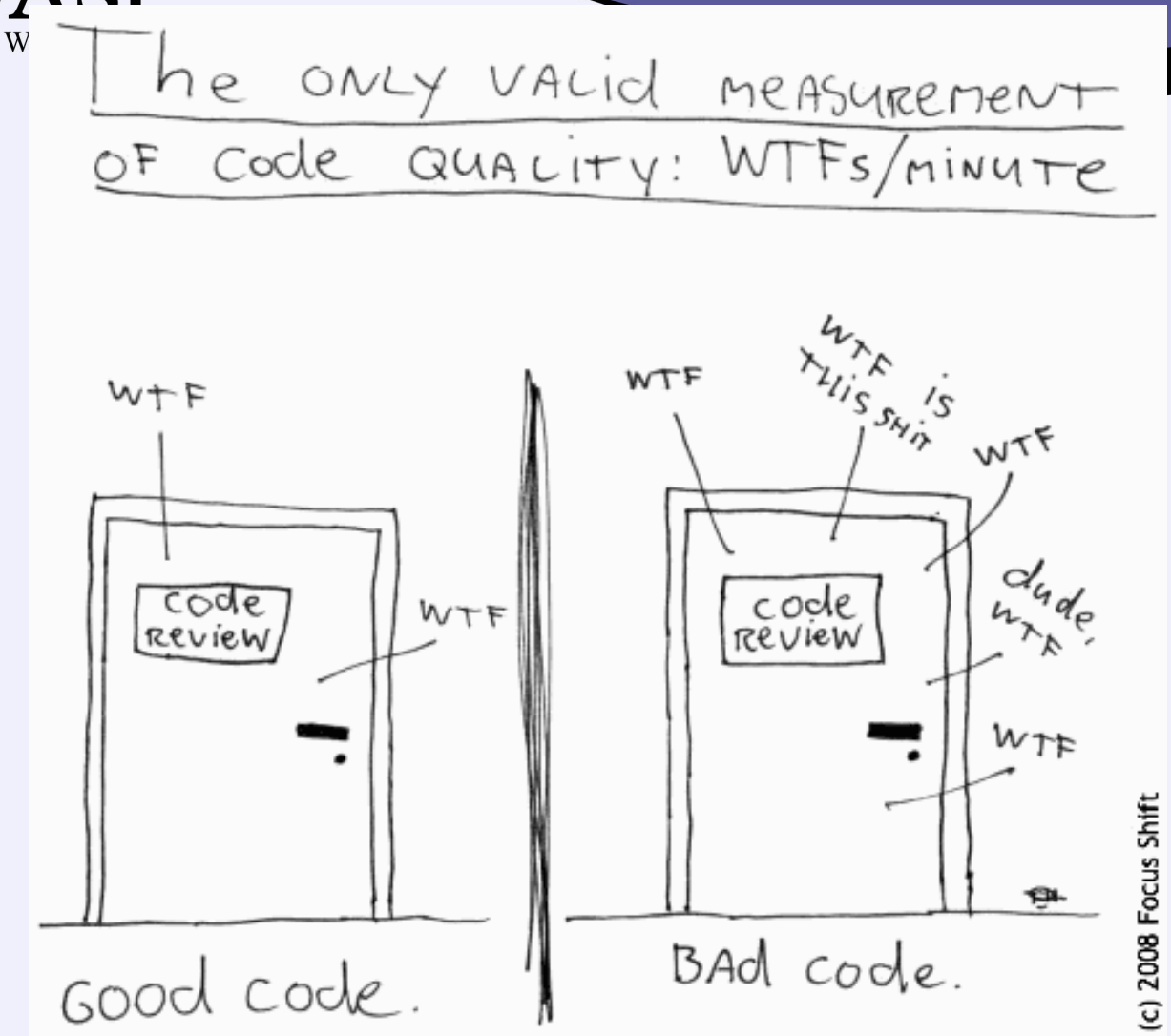
Twitter :@SPoint/@OWASP_France

Agenda

OWASP
The Open Web Application Security Project

- L’analyse de code source
- Qualité/Sécurité
- SonarQube
- Le projet OWASP SonarQube

L'analyse de code source résumée



L'analyse de code source

OWASP
The Open Web Application Security Project

- Identifier toutes les occurrences d'une faille
- Évaluer des facteurs contribuant à la sécurité
- Étudier l'application dans le détail
- Détecter les erreurs d'implémentation sournoises

```
1   static final String AB = "0123456789ABCDEFGHIJKLMNOPQRSTUWXYZabcdefghijklmnopqrstuvwxyz";
2   static Random rnd = new Random();
3
4   String randomString( int len )
5   {
6      StringBuilder sb = new StringBuilder( len );
7      for( int i = 0; i < len; i++ )
8         sb.append( AB.charAt( rnd.nextInt(AB.length()) ) );
9      return sb.toString();
10     }
11
12
```

Analyse du code vs Test d'intrusion applicatif (pour un CISO)

OWASP
The Open Web Application Security Project

Top10 Web	Tests d'intrusion	Analyse du code
A1 - Injection	++	+++
A2 – Violation de Session / Authentification	++	+
A3 – Cross Site Scripting	+++	+++
A4 – Référence Directes	+	+++
A5 – Mauvaise configuration	+	++
A6 – Exposition de données	++	+
A7 – Probleme d'habilitation fonctionnelle	+	+
A8 - CSRF	++	+
A9 – Utilisation de Composants vulnérables		+++
A10 – Redirection et transferts	+	+

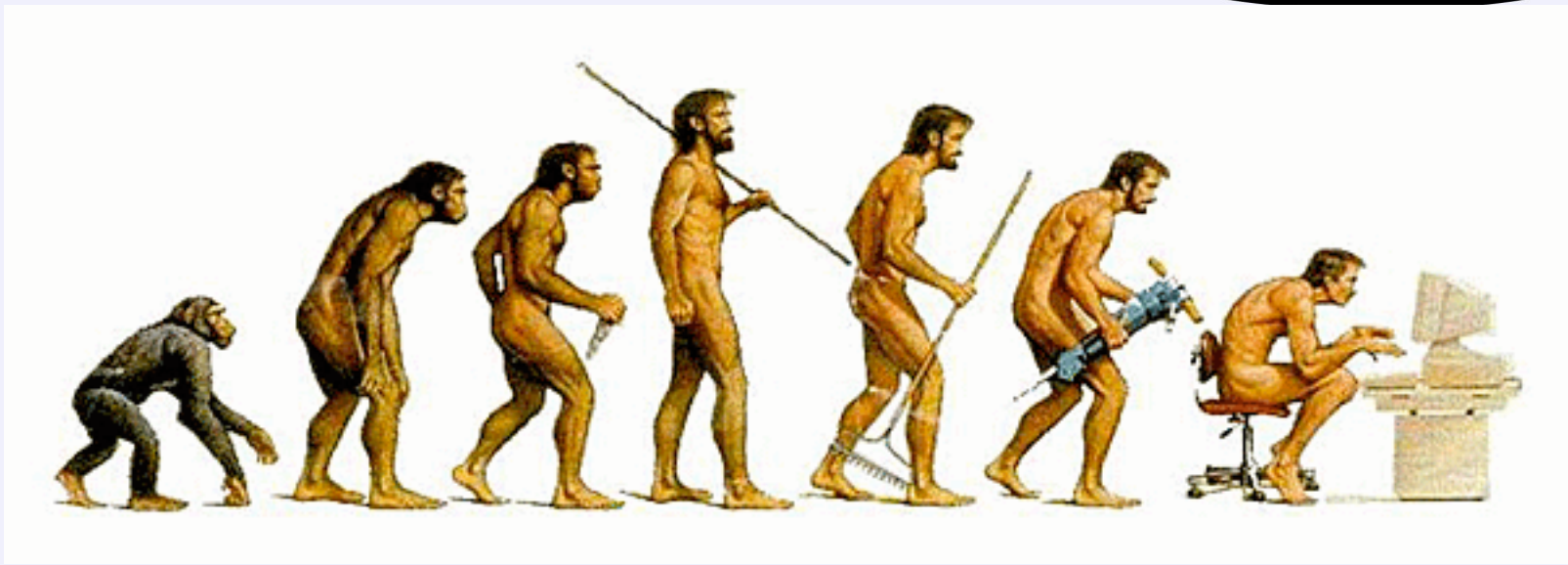
L’ analyse de code ou le test d’intrusion pour un développeur

OWASP

APPLICATION SECURITY FORUM WESTERN SWITZERLAND

L'évolution du développement logiciel

OWASP
The Open Web Application Security Project



Makefile

Gestionnaire de code source

Intégration continue

Tests unitaires

Inspection continue

OWASP
The Open Web Application Security Project

Les 7 péchés capitaux du développeur

Duplication de code....

OWASP
The Open Web Application Security Project

```
1   public SQLInjectionIdAndPass(String userid, String password )
2       String query = "SELECT * FROM users WHERE userid ='"+ userid + "'" + " AND password='" + password + "'";
3       Statement stmt = connection.createStatement();
4       ResultSet rs = stmt.executeQuery(query);
5       ......
6
7   }
8
9   public SQLInjectionPassAndId(String userid, String password )
10      String query = "SELECT * FROM users WHERE  password='" + password + "' AND userid ='"+ userid + "'";
11      Statement stmt = connection.createStatement();
12      ResultSet rs = stmt.executeQuery(query);
13      ......
14
15  }
16
```

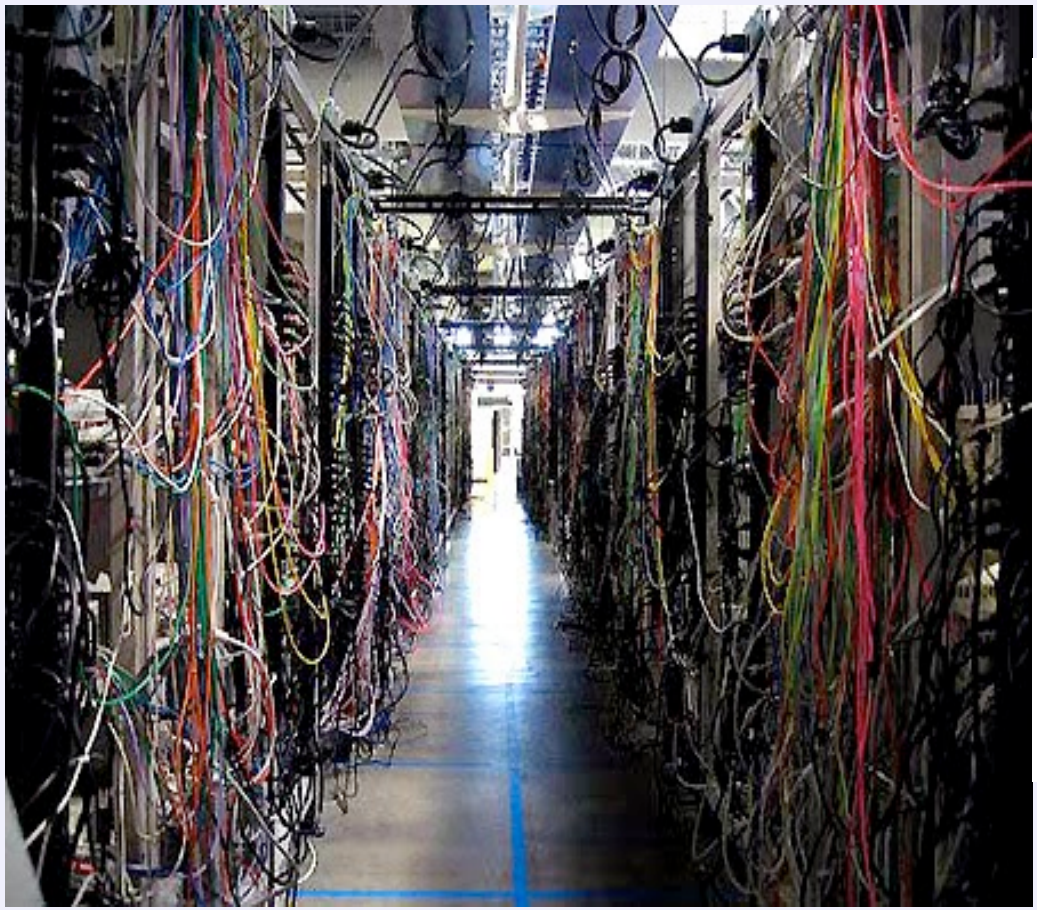

1x30 ou 10x3 ?

OWASP
The Open Web Application Security Project

```
if (size > 0) {
    Object otherValue = null;
    switch (size) {  // drop through
        case 3:
            if (other.containsKey(key3) == false) {
                return false;
            }
            otherValue = other.get(key3);
            if (value3 == null ? otherValue != null : !value3.equals(otherValue)) {
                return false;
            }
        case 2:
            if (other.containsKey(key2) == false) {
                return false;
            }
            otherValue = other.get(key2);
            if (value2 == null ? otherValue != null : !value2.equals(otherValue)) {
                return false;
            }
```

Mauvais Design

OWASP
The Open Web Application Security Project



```
10 Input A
15 Input B
20 B = A + 10
30 IF B > 12 GOTO 60
40 C = B / 3
50 IF C < 24 GOTO 10
60 Write C
70 IF Write Failed GOTO 15
80 Input D
```


APPLICATION SECURITY FORUM WESTERN SWITZERLAND

Super l'objet...

OWASP
The Open Web Application Security Project

```
public class QqChoseImportant {
    // .......
  public static void maSuperMethode(String[] args) {
    String original = "insecure";
    original.replace( 'i', '9' );
    // ........
  }
}
```


Non Respect des standards

OWASP
The Open Web Application Security Project

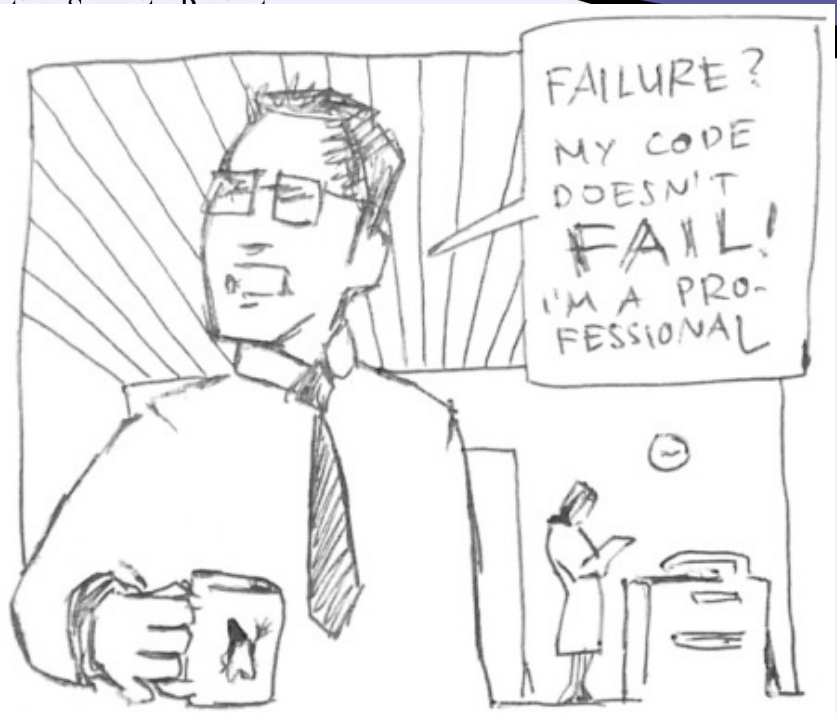
```
try {
  // ...
} catch (SecurityException se) {
  System.err.println(se);
}
```


Commentaire

OWASP
The Open Web Application Security Project

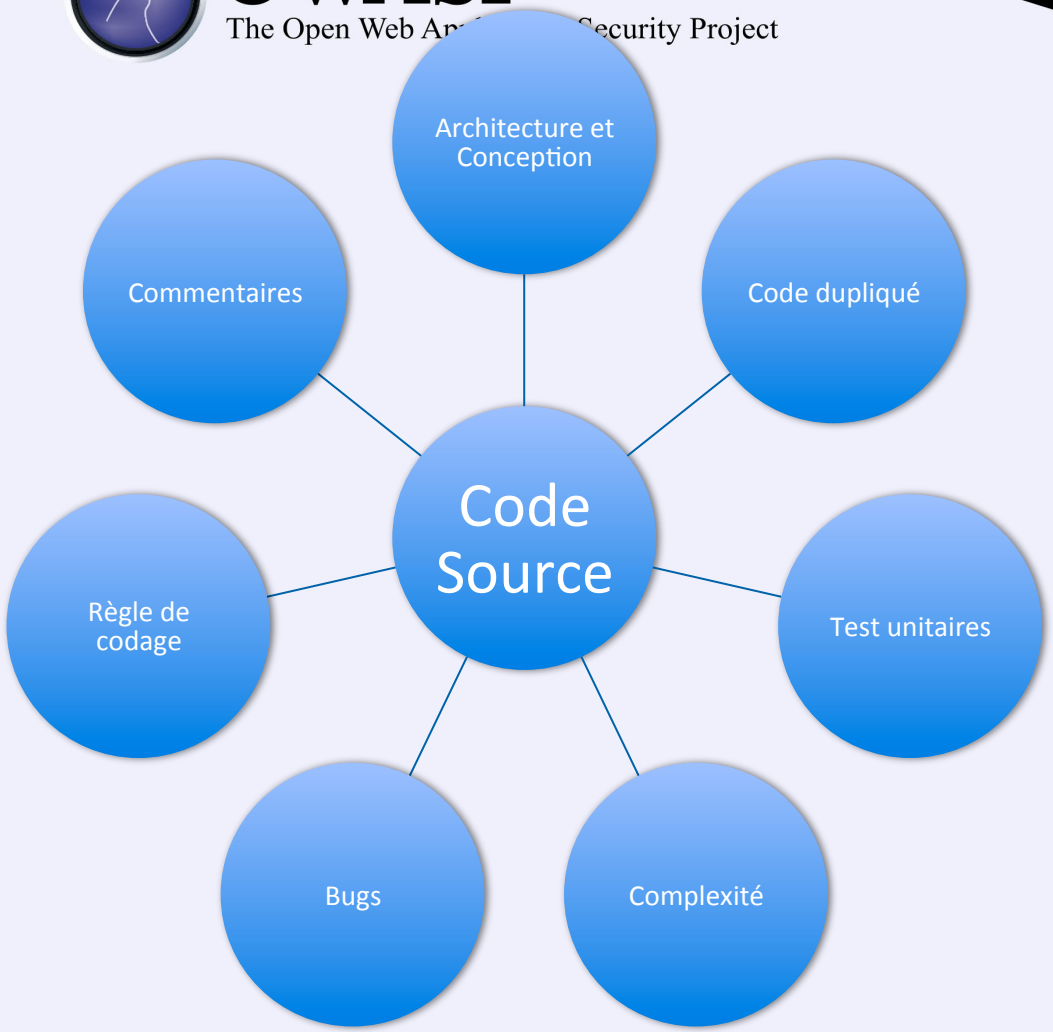
```
// Pattern matching du probleme

pattern = "\bword1\W+(?:\w+\W+){1,6}?word2\b";
String updated = EXAMPLE_TEST.replaceAll(pattern, "$2");
```


Les tests unitaires ?



- En programmation informatique, le **test unitaire** (ou "T.U.") est une procédure permettant de vérifier le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme (appelée « unité » ou « module »). (c) Wikipedia

7 axes pour couvrir la qualité d'un code



- Bugs
- Non respect des standards de codage
- Duplication de code
- Manque de tests unitaires
- Code trop complexe
- Conception spaghetti (mauvais design)
- Trop ou pas assez de code commenté.

SonarQube

OWASP
The Open Web Application Security Project

- Plateforme centralisé de gestion de la qualité :
  - Profils de qualité
  - Intégrable dans la chaine de build
  - Support de nombreux languages (C/C++, java, php, javascript, ...)
  - Plugins/extensions disponibles
  - Gestion de rapports et visualisation de l'évolution
  - Existe en version Open-Source

SonarQube pour la sécurité applicative

OWASP
The Open Web Application Security Project

- S'intègre dans le SDLC
  - liens possible avec Jenkins/Hudson
  - Reporting sur les violations
  - Possibilité d'ajouter des règles
- Dispose de règles permettant de couvrir
  - non respect des regles de codage
  - découverte de bugs sécurité(XSS, SQl-Injection)

SonarQube pour la sécurité applicative

OWASP
The Open Web Application Security Project

- Ce n'est pas un outil de revue de code !
  - Il fonctionne sur la violation de règles; détection de patterns uniquement
- Il tire toute sa puissance
  - si vous disposez d'une politique de Secure Coding
  - si vous démarrer un nouveau projet
- Il n'est pas "tres" orienté sécurité actuellement
  - peu de plugins de sécurité
  - pas de profils type pour les violations de secure coding.

Le projet OWASP SonarQube

OWASP
The Open Web Application Security Project

- Collaboration OWASP / SonarSource
  - Mettre a disposition de la communauté un ensemble de règles, profils, et plugins pour analyser la sécurité avec SonarQube.
- Plusieurs buts prévus
  - Livraison d'un profil OWASP Top10 supporté et maintenu par le projet début Octobre 2014 vis a vis du langage Java.
  - Livraison d'autres profils (probablement en 2015):
    - ASVS
    - ISO 27034-5
    - CERT Secure Coding
  - Développement de plugins spécifiques OWASP
    - pour les autres langages

OWASP
The Open Web Application Security Project

Démo

License

OWASP
The Open Web Application Security Project

@SPoint

sebastien.gioria@owasp.org