



La quête du code source maintenable, fiable et sécurisé

Par Sébastien Gioria et Freddy Mallet



OWASP

The Open Web Application Security Project

Attribution - Pas d'Utilisation
Commerciale - Partage dans
les Mêmes Conditions 3.0
France





OWASP

The Open Web Application Security Project

- [http://www.google.fr/#q=sebastien gioria](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.

@Spoint et @OWASP_France

Sebastien.gioria@owasp.org



Freddy Mallet



OWASP

The Open Web Application Security Project

- Créateur de la plateforme SonarQube
- Co-fondateur de la société SonarSource
- @FreddyMallet
- freddy.mallet@sonarsource.com





OWASP

The Open Web Application Security Project

- Enjeux autour de l'analyse du code source
- Tout ce que le code source peut dire
- Mise en œuvre méthodologique
- SonarQube / projet OWASP / Demo



OWASP

The Open Web Application Security Project

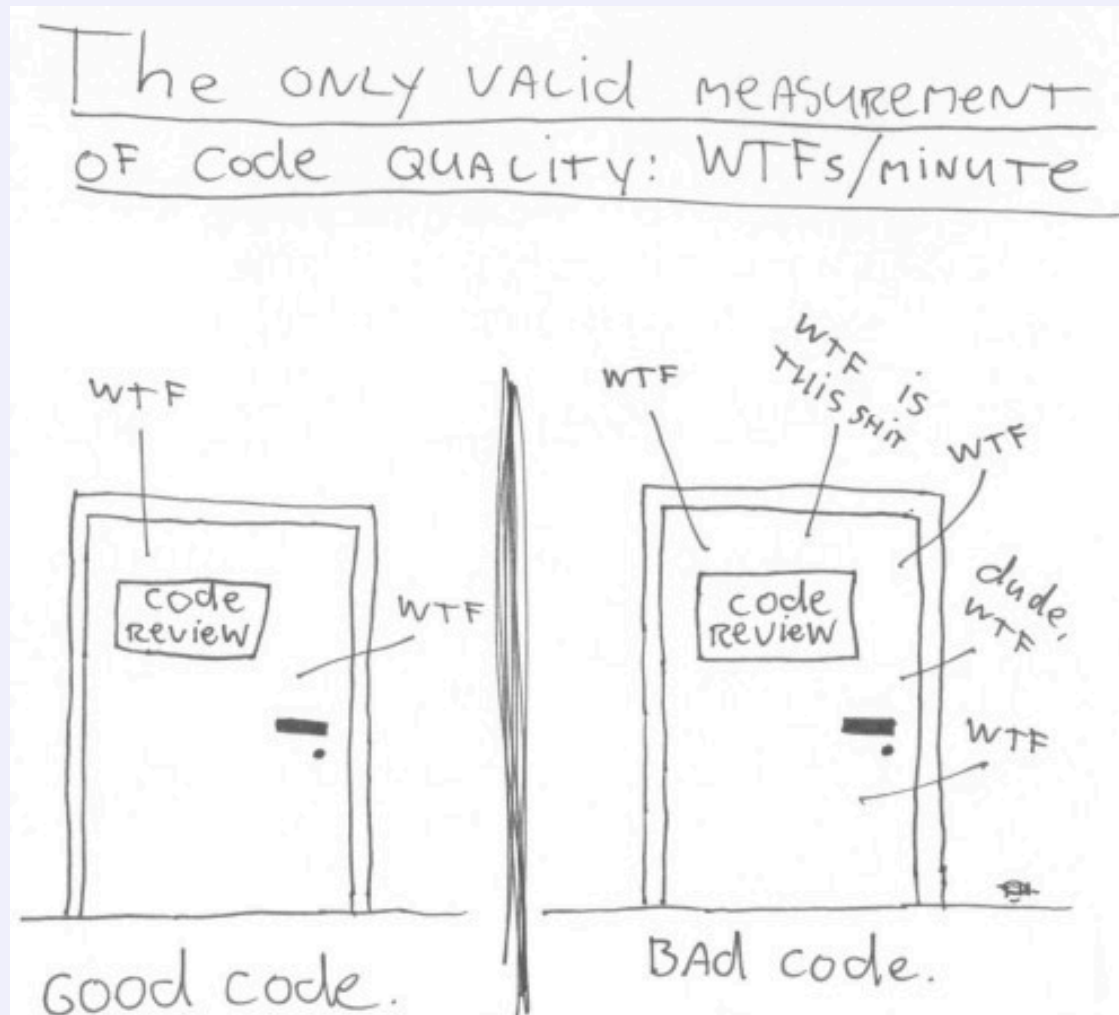
Enjeux autour de l'analyse du code source

Enjeux autour de l'analyse du code source



OWASP

The Open Web Application Security Project



(c) 2008 Focus Shift/OSNews/Thom Holwerda - <http://www.osnews.com/comics>





OWASP

The Open Web Application Security Project

Beaucoup de défauts pouvant conduire à des problèmes de maintenabilité, de stabilité et de sécurité peuvent être détectés automatiquement.

Et tout particulièrement les plus sournois

```
1 static final String AB = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
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
```

Approche boîte noire VS boîte blanche



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 – Références 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 du point de vue du développeur ?



OWASP

The Open Web Application Security Project

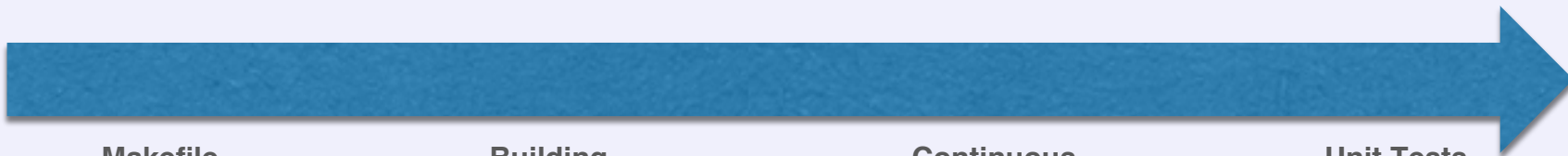
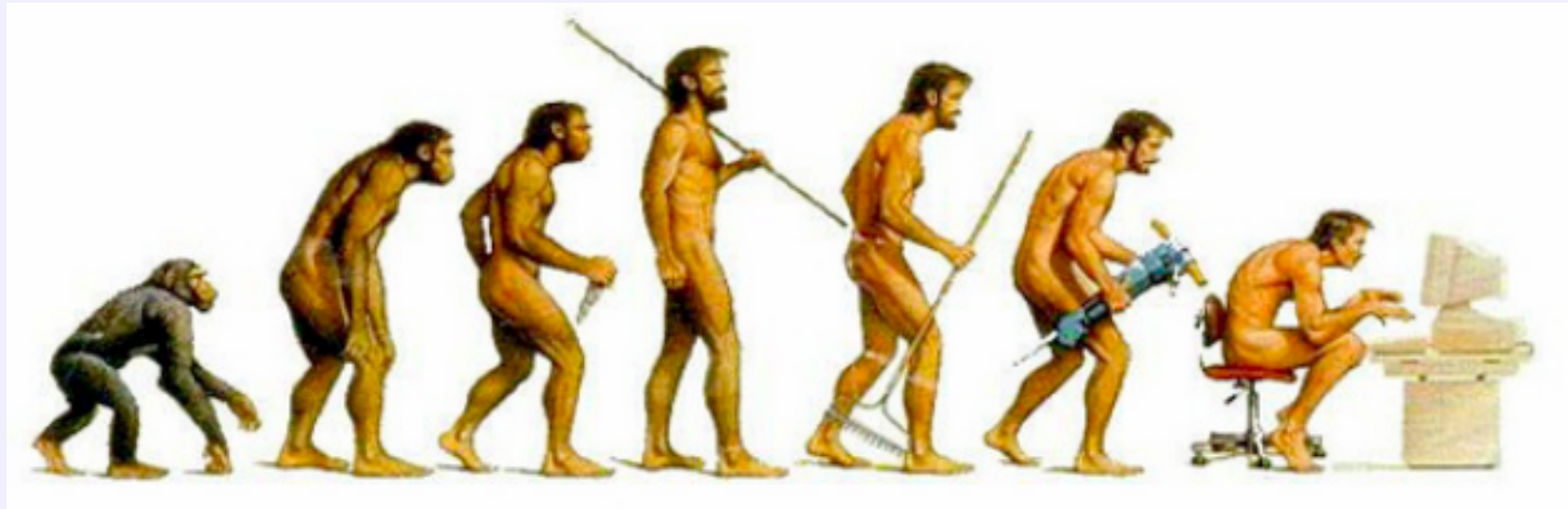


L'évolution du développement logiciel



OWASP

The Open Web Application Security Project



Vi / Emacs	Makefile	Version Control	Building Tools	Ticket Tracking	Continuous Integration	Refactoring From the IDE	Unit Tests	Continuous Inspection
------------	----------	-----------------	----------------	-----------------	------------------------	--------------------------	------------	-----------------------



OWASP

The Open Web Application Security Project

Tout ce que le code source
peut dire

Les principaux types de défauts



OWASP

The Open Web Application Security Project



Sécurité

OWASP Top 10, SANS TOP 25, ...

Fiabilité

Multi-threadings, null pointers, buffer overflows, unclosed resources, ...

Maintenabilité

???

Exemples liés à la sécurité



OWASP

The Open Web Application Security Project

- Injection SQL, LDAP, ...
- Login/mot de passe en dur
- Utilisation d'algorithmes de hashage trop faibles: MD5, SHA1, ...
- Injection de code
- Redirection web vers un site inconnu
- ...

Exemples liés à la fiabilité



OWASP

The Open Web Application Security Project

- Déréférencement de pointeurs null
- Débordement d'entier
- Conditions invalides
- Assignation d'une variable à elle même
- Ressources non libérées
- Assignation d'une valeur jamais utilisée
- ...

Quid de la maintenabilité ?



OWASP

The Open Web Application Security Project

- ?
- ?
- ?
- ?
- ?
- ?



Que choisir entre la peste et le choléra ?

```
public Set<AsmResource> getResourceBlock(AsmResource fromResource) {
```



```
public Set<AsmResource> getResourceBlockNew(AsmResource fromResource) {
```


Mauvaise distribution de la complexité



OWASP

The Open Web Application Security Project

Vaut-il mieux une méthode d'une complexité de 30 ou 10 méthodes d'une complexité de 3 ?

```
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;
            }
    }
}
```

Pas ou peu de tests unitaires



OWASP

The Open Web Application Security Project





Quel classe/package
est responsable de quoi ?





Comme par exemple la stratégie de gestion des exceptions

```
try {  
    computeOrder(order);  
} catch (RuntimeException e) {  
    System.out.println("The order can't be computed!");  
}
```



Cette expression régulière “match” quoi
par exemple ?

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



OWASP

The Open Web Application Security Project

- Duplication de code
- Mauvaise distribution de la complexité
- Peu ou pas de tests unitaires
- Mauvais design
- Non respect des standards
- Pas ou peu de commentaires



OWASP

The Open Web Application Security Project

Mise en oeuvre méthodologique

Approche traditionnelle



OWASP

The Open Web Application Security Project

- Retour trop tardif
- **Manque d'implication des développeurs**
- Pushback de ces derniers
- Pas de réelle douane applicative
- **Outils, processus et personnes différentes pour chasser les différents types de défauts**



Comment rembourser la dette ?



OWASP

The Open Web Application Security Project

- Le montant total peut être déprimant
- Faut-il demander un budget dédié ?
- **Le risque d'injection d'une régression fonctionnelle existe**
- Ce n'est pas très "fun" !



Ce qu'il faut changer



OWASP

The Open Web Application Security Project

- La boucle de rétroaction doit être beaucoup plus rapide
- Une douane applicative non négociable doit exister
- **Les développeurs doivent être au coeur du processus**
- Le coût doit être non significatif
- L'approche doit être unifiée

Se focaliser sur la fuite



OWASP

The Open Web Application Security Project



Debit





OWASP

The Open Web Application Security Project

SonarQube OWASP Demo

SonarQube



OWASP

The Open Web Application Security Project

- Intégrable dans la chaîne de build
- Support de nombreux langages: C/C++, Java, PHP, JavaScript, COBOL, C#, PL/SQL, ...
- Support du concept de douane applicative
- Extensible: nombreux plugins
- Gestion temporelle des défauts
- Open-Source

SonarQube pour la sécurité applicative



OWASP

The Open Web Application Security Project

- S'intègre dans le SDLC
 - liens possible avec Jenkins/Hudson/Bamboo
 - Reporting sur les défauts
 - Possibilité d'ajouter des règles (en XPath)
- Dispose de règles permettant de couvrir
 - non respect des règles 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 démarrez 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 objectifs prévus
 - Lier les règles au référentiel MITRE CWE
 - Tagguers les règles suivant les catégories OWASP Top 10 2013
 - Développement de nouvelles règles “sécurité”
 - Nouveau widget pour offrir une perspective “sécurité” sur le code source
 - Etendre ces objectifs à tous les plugins langage



OWASP

The Open Web Application Security Project





OWASP

The Open Web Application Security Project

@Spoint

@FreddyMallet

@OWASPSonarQube

@OWASP_France



Attribution - Pas d'Utilisation
Commerciale - Partage dans
les Mêmes Conditions 3.0
France

