



Chapter Meeting OWASP France

Paris, 12 Mars 2013



OWASP

The Open Web Application Security Project

solucom 
management & IT consulting

Agenda

Ludovic Petit

Chapter Leader OWASP France

Introduction & Agenda



9:45

Jim Manico

VP of Security Architecture for
WhiteHat Security



Secure Coding:

- Authentication Best Practices for Developers
- Access Control Design Best Practices

10:05

Gérôme Billois

Solucom

Sécurité Applicative:

l'organisation, clé de la réussite



10:45

Sébastien Gioria

Chapter Leader OWASP France

OWASP News & Update



12:00

Ely de Travieso

OWASP France

Adhésions & Partenariats



Jim Manico
VP of Security Architecture
for WhiteHat Security



Secure Coding:

- Authentication Best Practices for Developers
- Access Control Design Best Practices



Web Application Access Control Design



Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document
under the terms of the OWASP License.

What is Access Control / Authorization?

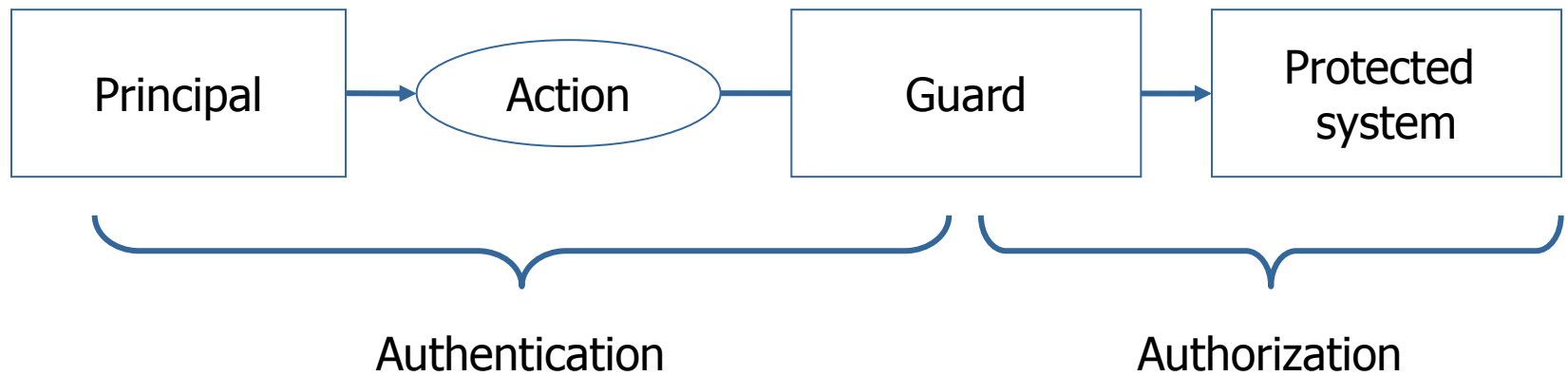
Authorization is the process where a system determines if a specific user has access to a particular resource

The intent of authorization is to ensure that a user only accesses system functionality to which he is entitled

Role based access control (RBAC) is commonly used to manage permissions within an application

RBAC has significant limits and does not address horizontal access control issues

General Access Control Model



Attacks on Access Control

Vertical Access Control Attacks

A standard user accessing administration functionality

Horizontal Access Control attacks

Same role, but accessing another user's private data

Business Logic Access Control Attacks

Abuse of workflow

Access Controls Impact

Loss of accountability

- Attackers maliciously execute actions as other users
- Attackers maliciously execute higher level actions

Disclosure of confidential data

- Compromising admin-level accounts often results in access to user's confidential data

Data tampering

- Privilege levels do not distinguish users who can only view data and users permitted to modify data

Access Control Anti-Patterns

Hard-coded role checks in application code

Lack of centralized access control logic

Untrusted data driving access control decisions

Access control that is “open by default”

Lack of addressing horizontal access control in a standardized way (if at all)

Access control logic that needs to be manually added to every endpoint in code

Access Control that is “sticky” per session

Access Control that requires per-user policy

Hard Coded Roles

```
if ((user.isManager() ||
     user.isAdministrator() ||
     user.isEditor()) &&
     user.id() != 1132))
{
    //execute action
}
```

How do you change the policy of this code?

Hard Coded Roles

Makes “proving” the policy of an application difficult for audit or Q/A purposes

Any time access control policy needs to change, new code need to be pushed

Fragile, easy to make mistakes

Is not “automatic” and needs to be “hand-coded” within each application feature

Order Specific Operations

Imagine the following parameters

- `http://example.com/buy?action=chooseDataPackage`
- `http://example.com/buy?action=customizePackage`
- `http://example.com/buy?action=makePayment`
- `http://example.com/buy?action=downloadData`

Can an attacker control the sequence?

What step would a "threat agent" like to skip?

Can an attacker abuse this with concurrency?

Never Depend on Untrusted Data

Never trust request data for access control decisions

Never make access control decisions in JavaScript

Never make authorization decisions based solely on

- Hidden fields
- Cookie values
- Form parameters
- URL parameters
- Anything else from the request

Never depend on the order of values sent from the client

Access Control Best Practices, I

Implement role based access control to assign permissions to application users for vertical access control requirements

Implement data-contextual access control to assign permissions to application users in the context of specific data items for horizontal access control requirements

Perform consistent authorization checking routines on all application pages

Where applicable, apply DENY privileges last, issue ALLOW privileges on a case-by-case basis

Access Control Best Practices, II

Build a centralized access control mechanism

Code to the activity/permission, not the role

Design access control as a filter

Deny by default, fail securely

Access Control Best Practices, III

Apply same core logic to presentation and server-side access control decisions

Server-side trusted data should drive access control

Be able to change a users role in real time

Build grouping capability for users and permissions

Avoid assigning permissions on a per-user basis

Best Practice: Code to the Activity

```
if (AC.hasAccess("article:edit:12"))  
{  
    //execute activity  
}
```

Code it once, never needs to change again

Implies policy is centralized in some way

Implies policy is persisted in some way

Requires more design/work up front to get right

Best Practice: Centralized ACL Controller

Define a centralized access controller

- `ACLService.isAuthorized(ACTION_CONSTANT)`
- `ACLService.assertAuthorized(ACTION_CONSTANT)` throws `AccessControlException()`

Access control decisions go through these simple API's

Centralized logic to drive policy behavior and persistence

May contain data-driven access control policy information

Using a Centralized Access Controller

In Presentation Layer

```
if (isAuthorized(VIEW_LOG_PANEL))  
{  
    <h2>Here are the logs</h2>  
    <%=getLogs();%/>  
}
```

In Controller

```
try (assertAuthorized(DELETE_USER))  
{  
    deleteUser();  
}
```

Best Practice: Verifying policy server-side

Keep user identity verification in session

Load entitlements server side from trusted sources

Force authorization checks on ALL requests

- JS file, image, AJAX and FLASH requests as well!
- Force this check using a filter if possible

SQL Integrated Access Control

Example Feature

- <http://mail.example.com/viewMessage?msgid=2356342>

This SQL would be vulnerable to tampering

- `Select * from messages where messageid = 2356342`

Ensure the owner is referenced in the query!

- `Select * from messages where messageid = 2356342 AND messages.message_owner = <userid_from_session>`

Authorization Models

.NET (enable in web.config)

- File authorization (active when use Windows authentication)
- URL authorization (maps users and roles to pieces of URL namespace)

J2EE

- Declarative (defined in deployment descriptors of container components)
- Programmatic (extends declarative)
- Custom-coded (not recommended!)

Declarative .NET Authorization

**Enforce
permissions-
based access
to pages**

- Web.config: Web Container authorization-constraint example

- /admin/ is limited to "Admin" users

```
<location path = "/admin/">  
  <system.web>  
    <authorization>  
      <allow roles = "Admin" />  
      <deny users = "*" />  
    </authorization>  
  </system.web>  
</location>
```

Declarative J2EE Authorization

Enforce permissions-based access to servlets and EJB methods

- Web.xml: Web Container authorization-constraint example
- The getBalance transaction is limited to Authorized users

```
<security-constraint>
  <web-resource-collection>
    <url-pattern>/action/getBalance*</url-pattern>
    <http-method>POST</http-method>
  </web-resource-collection>
  <auth-constraint>
    <role-name>AuthorizedUser</role-name>
  </auth-constraint>
</security-constraint>
```


J2EE Programmatic Authorization

Extend declarative security using J2EE programmatic security for each web and EJB container

Use java.security API methods available to the HttpServletRequest object (getRemoteUser(), isUserInRole(), etc)

```
Java.security.Principal principal =  
    request.getUserPrincipal();  
String remoteUser = principal.getName();
```

Note: J2EE provides same security model for EJBs as for web container. Declarative security is defined in bean's deployment descriptor

Data Contextual Access Control

Data Contextual / Horizontal Access Control API examples:

- `ACLService.isAuthorized("car:view:321")`
- `ACLService.assertAuthorized("car:edit:321")`

Long form:

- `is Authorized(user, Perm.EDIT_CAR, Car.class, 14)`

Check if the user has the right role in the context of a specific object

Protecting data at the lowest level!

Data Contextual Access Control

User	
User ID	User Name

Permission	
Permission ID	Permission Name

Data Type	
Data ID	Data Name

Role	
Role ID	Role Name

Entitlement / Privilege				
User ID	Permission ID	Role ID	Data Type ID	Data Instance Id

Gérôme Billois
Solucom



Sécurité Applicative:
l'**organisation**, clé de la réussite



Sécurité applicative

Quelle organisation pour garantir son succès ?



OWASP

The Open Web Application Security Project

Gérôme BILLOIS – Solucom
gerome.billois@solucom.fr @gbillois

Qui sommes-nous ?



OWASP

The Open Web Application Security Project

- Solucom, un cabinet **indépendant** de conseil en management et système d'information
- Une practice Sécurité & Risk Management dont la mission est d'accompagner nos clients dans la **maîtrise des risques** et la **conduite des projets** au bénéfice des métiers
- Nos convictions
 - Prioriser les risques en fonction des enjeux des métiers*
 - Faciliter l'évolution des usages en centrant la sécurité sur l'information*
 - Allier protection, détection et réaction face aux nouvelles menaces*
- Nos savoir-faire

Stratégie, gouvernance et pilotage des risques

Continuité

Infrastructures

Sécurité applicative

Gestion des identités

Gestion opérationnelle

Audits et tests d'intrusions

Pilotage et réalisation des projets / Conduite du changement

solucom
management & IT consulting



- ✓ 20 ans d'existence
- ✓ Près de 1 000 collaborateurs, dont 175 en sécurité et gestion de risques
- ✓ Implication forte dans les organismes professionnels (AFNOR, Club 27001, CLUSIF, Forum des Compétences...)



www.solucominsight.fr
Sécurité & Risk Mgt.

Sécurité applicative : Quelle organisation pour garantir son succès ?



OWASP

The Open Web Application Security Project



Quelle situation aujourd'hui ?



Construire une cellule SecApp



Et pour finir...

Retour sur des cas concrets...



OWASP

The Open Web Application Security Project

citibank

360000 données de clients volées
2.7 M€ de pertes

**Modification
de l'URL**

SONY

100 millions de données joueurs
dérobées

Injection SQL

Linked in

6 millions de mots de passe volés

Injection SQL

De multiples attaques applicatives potentielles



OWASP

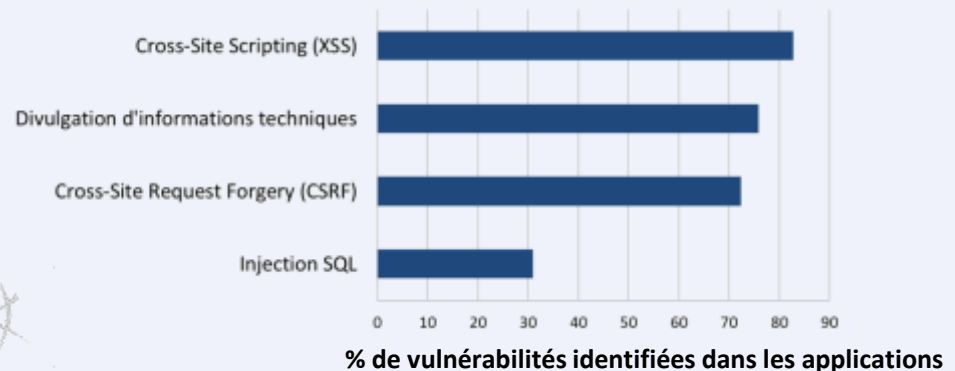
The Open Web Application Security Project

- Les applications sont de plus en plus **nombreuses**
- L'ouverture du SI s'intensifie : les applications sont de plus en plus **exposées**
- Les nouvelles **menaces** sont applicatives

**La sécurité applicative est indispensable...
mais manifestement, ça ne fonctionne pas !**

2011-2012 :
100% des applications auditées
par Solucom présentent au
moins une **faille de sécurité**

solucom
management & IT consulting



Des initiatives sécurité applicative qui échouent au quotidien



OWASP

The Open Web Application Security Project

RSSI

Intégration de la Sécurité dans les Projets (ISP)

Baucoup d'interlocuteurs

Sécurité réinventée à chaque projet

Coût non prédictible



Sécurité dans les développements

Manque de compétences et de priorisation sécurité

Études / MOA

Sensibilisation des développeurs

Standards techniques sécurité applicative

Peu d'expertise applicative

Revue de code

Sécurité applicative dans les contrats

Sécurité opérationnelle

Tests de vulnérabilités applicatifs

Trop tard

Configuration outillage sécurité applicative

Peu de connaissance des applications

Production

La sécurité de l'information dans les grandes organisations



OWASP

The Open Web Application Security Project

RSSI



La sécurité de l'information : une situation **déséquilibrée**



Études / MOA



Production

Que peut-on apprendre du passé ?



OWASP

The Open Web Application Security Project

Depuis 10 ans



→ Expertise / Compréhension

→ Proximité / Confiance

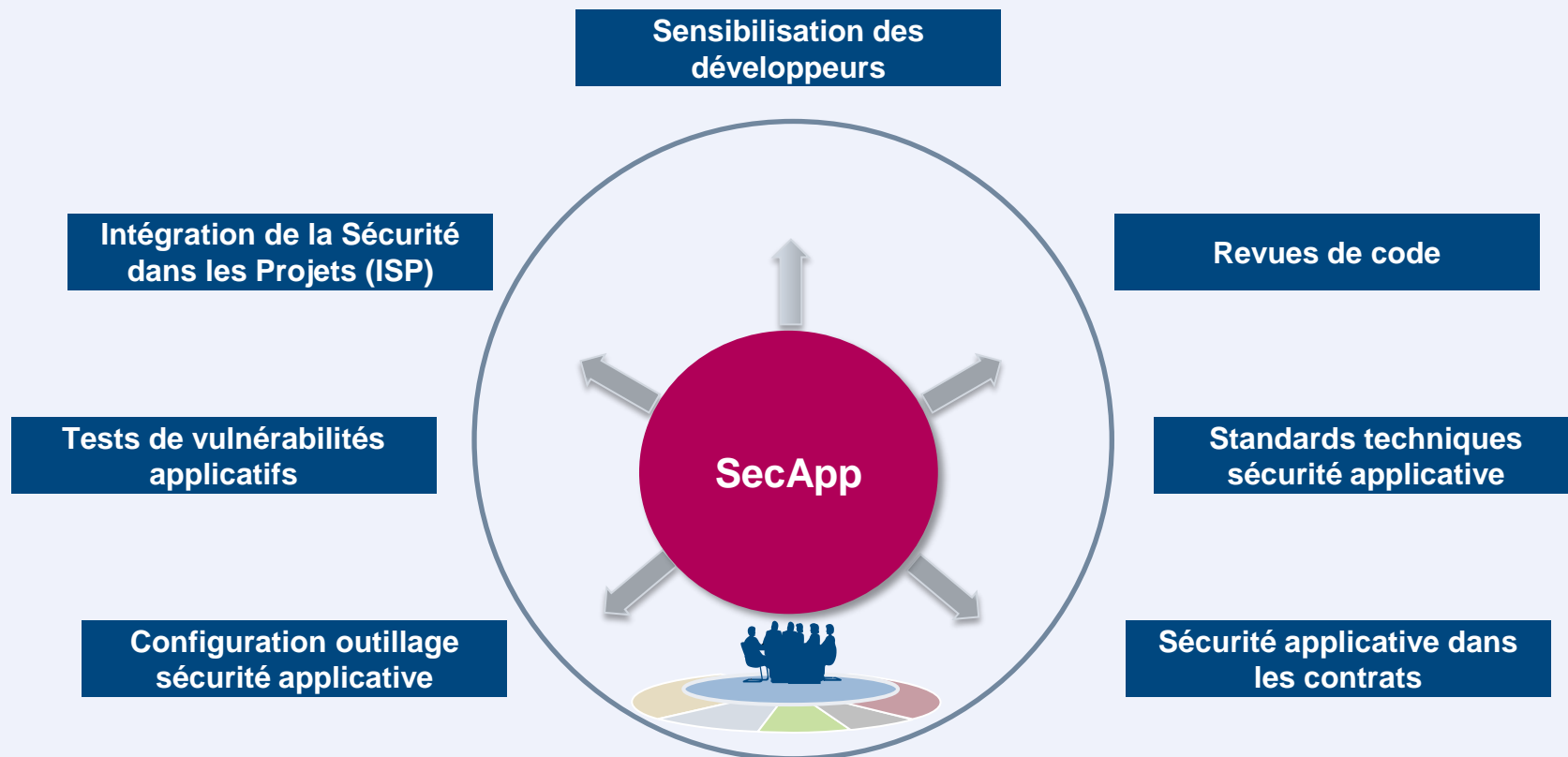
→ Réactivité / Anticipation

Aujourd'hui



Une nouvelle relation à construire !

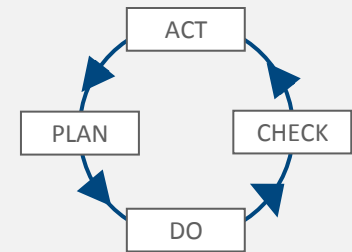
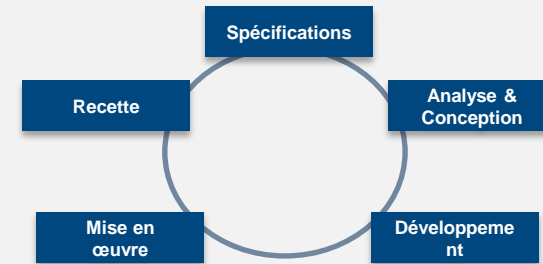
Centraliser l'ensemble des actions autour d'une équipe de spécialistes : la **Cellule de Sécurité Applicative**



Les objectifs de la SecApp



- **Sécuriser les applications** sur toutes les étapes du projet
- **Maintenir et améliorer** la sécurité dans le temps
- A terme, **simplifier la sécurité applicative** et **rendre autonome** les acteurs



Sécurité applicative : Quelle organisation pour garantir son succès ?



OWASP

The Open Web Application Security Project



Quelle situation aujourd'hui ?



Construire une cellule SecApp



Et pour finir...

Quels profils ? Quel rattachement ?



OWASP

The Open Web Application Security Project

En priorité des profils applicatifs formés à la sécurité



Architectes applicatif



Chefs de projet applicatif



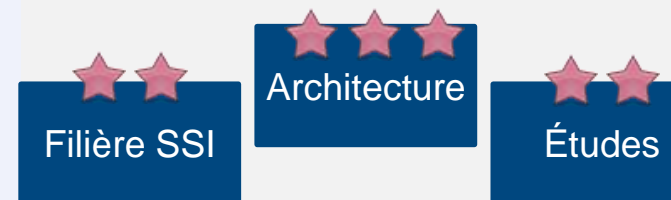
Développeurs

Tout en capitalisant sur les équipes existantes
en particulier les équipes ISP

En moyenne, sur des structures avancées :
1 membre SecApp pour 50 ETP développeurs

Quelques critères pour choisir un rattachement

- ☑ Capacité à capter les projets
- ☑ Légitimité
- ☑ Expertise applicative
- ☑ Connaissance infrastructure



La SecApp dans son environnement



OWASP

The Open Web Application Security Project

RSSI

Politique /
Réglementaire

Indicateurs / Reporting

Achats

Contrats

Sec
App

Captation projet

Développement
externalisé

Écosystème de
développement

Études / MOA

Écosystème
infrastructure

Mise en production

Sécurité opérationnelle

Incidents / Audits /
Crises

Production

3 niveaux de maturité pour la SecApp





1 Proximité terrain

2 Capitalisation et outillage

3 Approche services

Des référentiels de bonnes pratiques disponibles pour construire sa feuille de route



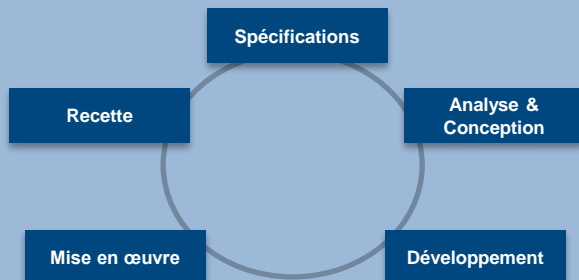
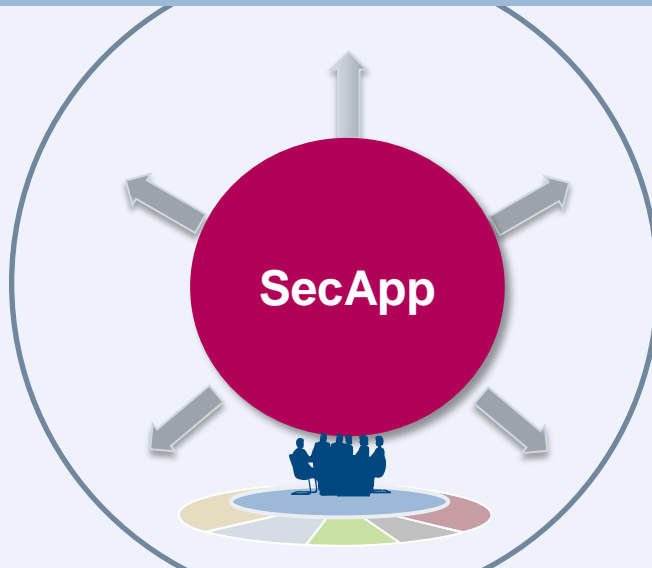
① La SecApp réalise



OWASP

The Open Web Application Security Project

Pilotage / Gouvernance
Reporting & Indicateurs



Support aux projets



- **Aller sur le terrain** pour constater et comprendre
- **Réaliser soi-même** les actions pour gagner en maturité
- **Cibler les 5 à 10 projets clés**
 - Développements externalisés
 - Progiciels
 - Legacy
 - Cloud...

② La SecApp capitalise



OWASP

The Open Web Application Security Project

Pilotage / Gouvernance

Reporting, Indicateurs, Communication, Sensibilisation

Outillage

Conception

- Architecture : modèles, pattern...
- Développement : guides / framework
- Achats: contrats types, env. de développement...

Contrôle

- Architecture : checklist conformité, analyse de risque
- Code : analyseur
- Recette : scanneur de vulnérabilités...

Spécifications

Recette

Analyse &
Conception

Mise en œuvre

Développement

Support aux projets



- **Simplifier** et éviter de tout réinventer à chaque projet

- S'appuyer sur les **références internationales** pour gagner du temps



OWASP
The Open Web Application Security Project

Microsoft

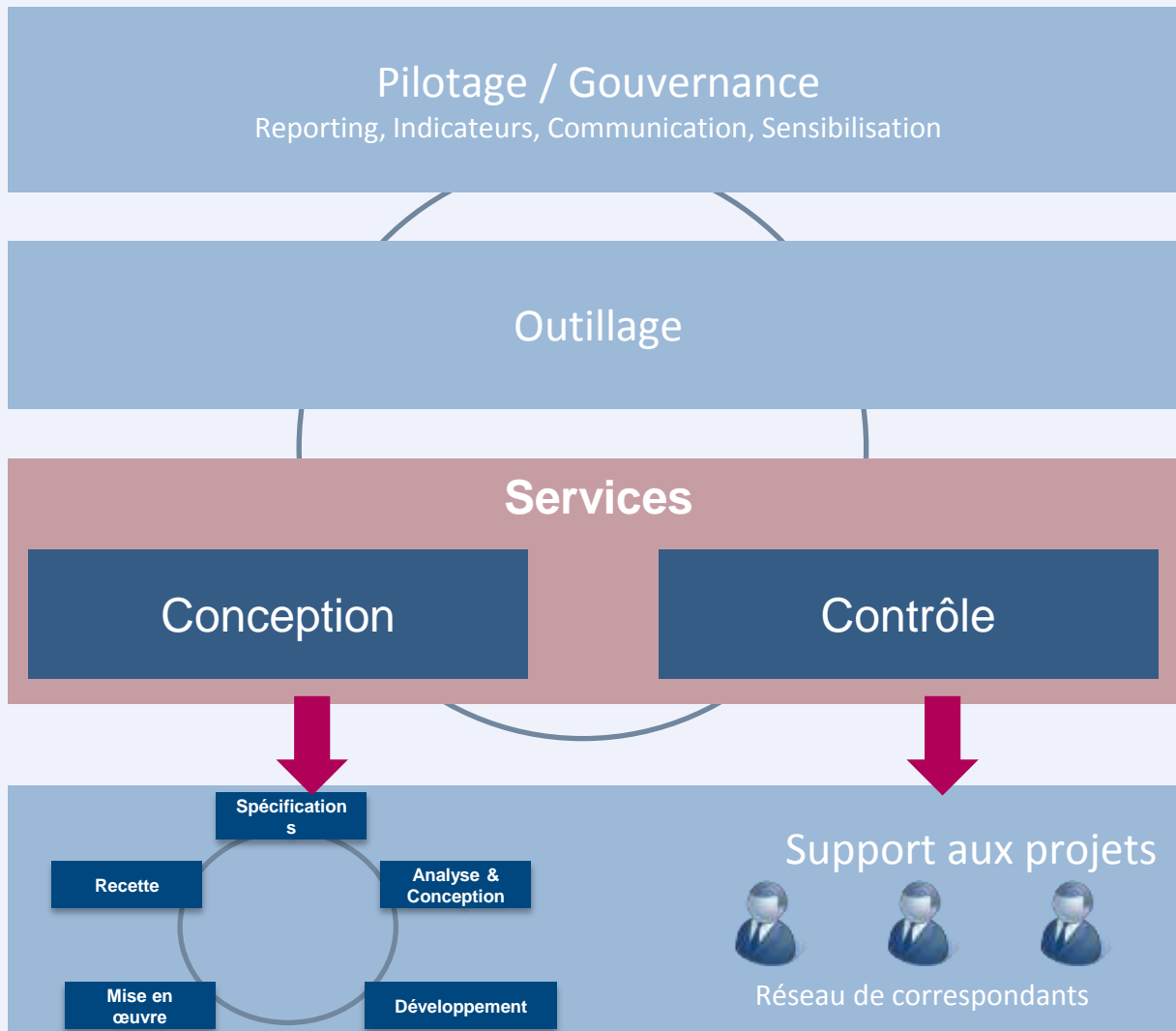
- **Former** les opérationnels (chef de projets, développeurs...)

③ La SecApp offre des services



OWASP

The Open Web Application Security Project

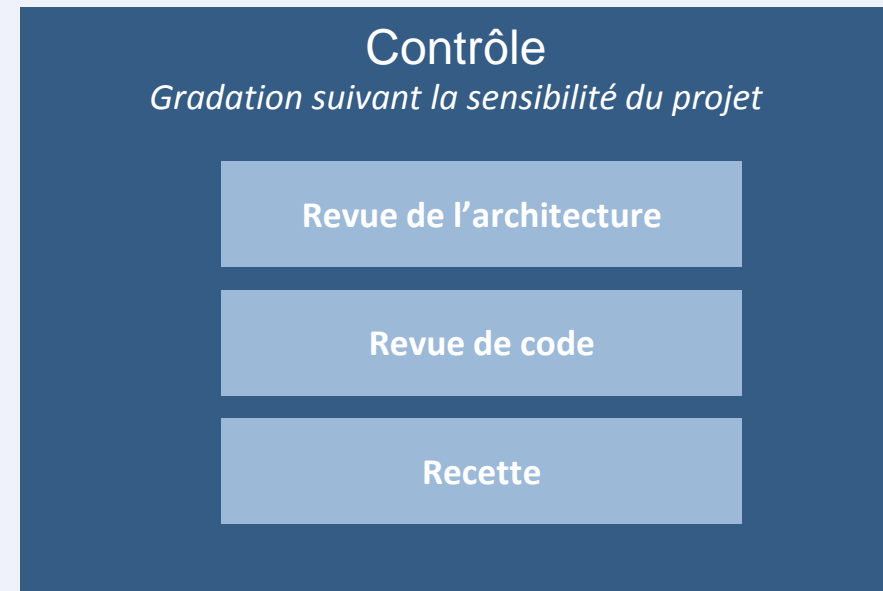
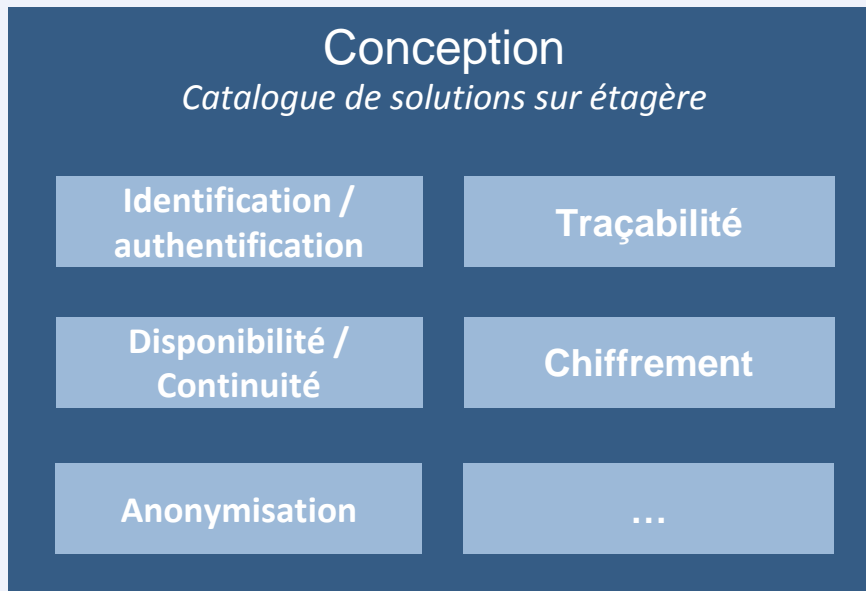


- **Industrialisation** : création de services
- Définition de **catalogues de services** en fonction de la **sensibilité** du projet
- Identification des **coûts et des délais** associés à chaque niveau de service

Zoom sur les deux catalogues de services



- Des services que la cellule SecApp doit construire en interne



Service Contrôle / Recette :
Tests d'intrusion



L'enjeu : simplifier la sécurité applicative pour tous



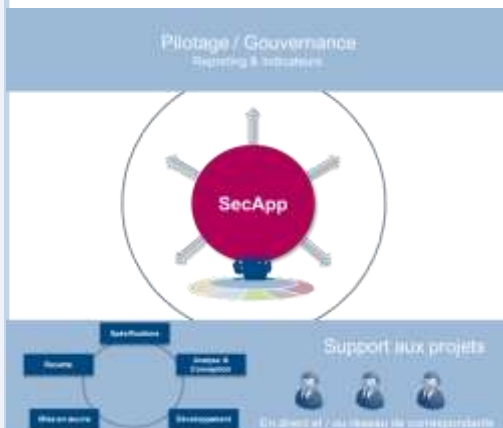
OWASP

The Open Web Application Security Project

3 à 5 ans

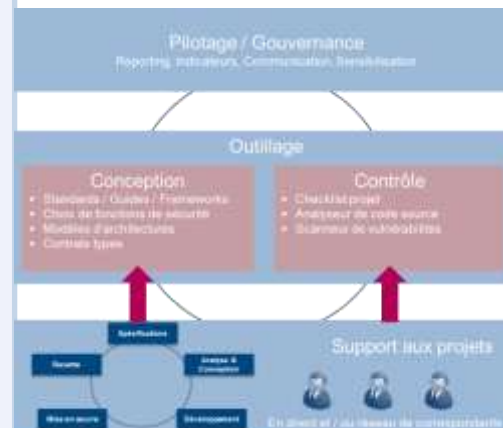
Faire

La SecApp réalise



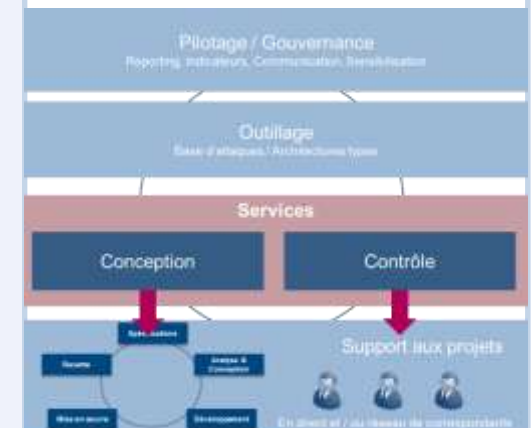
Savoir-Faire

La SecApp capitalise



Savoir-Faire Faire

La SecApp s'industrialise



A la cible, autonomie et simplification pour une prise en compte de la sécurité applicative la plus transparente possible

Sécurité applicative : Quelle organisation pour garantir son succès ?



OWASP

The Open Web Application Security Project



Quelle situation aujourd'hui ?



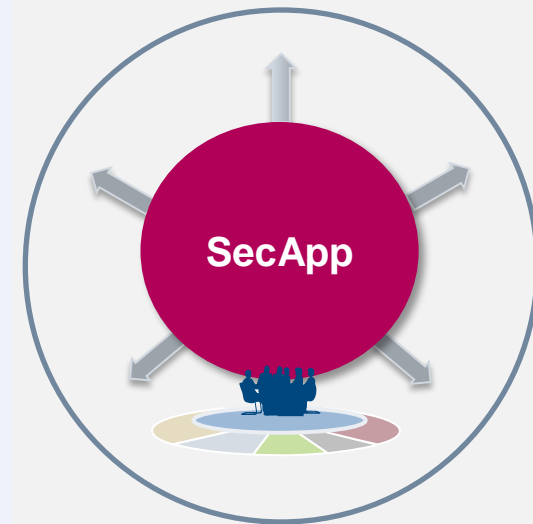
Construire une cellule SecApp



Et pour finir...



- Une équipe de **spécialistes applicatifs**
- Au plus près des **interlocuteurs concernés**
- Qui **appuie, conseille**, et progressivement **industrialise** les services qu'elle rend
- Pour **faciliter** progressivement l'intégration de la sécurité et augmenter le **niveau d'autonomie** des équipes



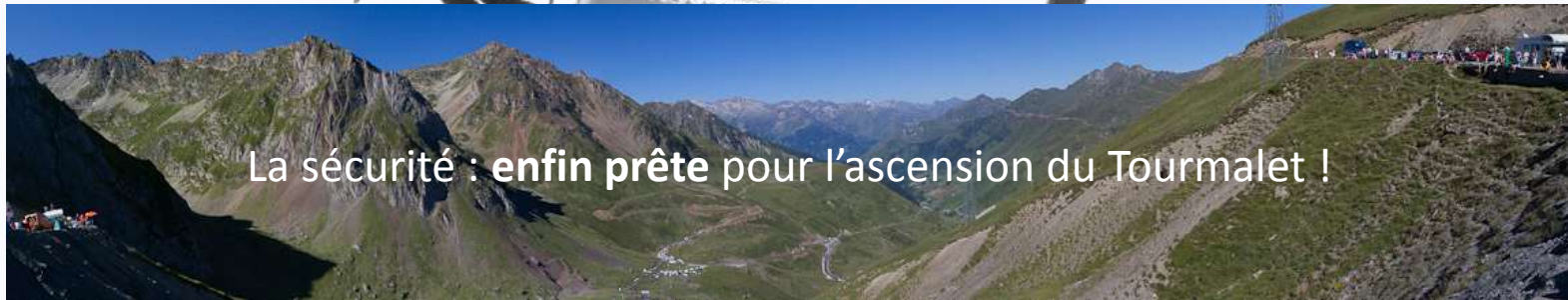
A la cible, la SecApp rééquilibre
la sécurité de l'information !



OWASP

The Open Web Application Security Project

RSSI



La sécurité : enfin prête pour l'ascension du Tourmalet !



Études / MOA



Production

Sécurité applicative : Quelle organisation pour garantir son succès ?



OWASP

The Open Web Application Security Project



Gérôme BILLOIS – Solucom
gerome.billois@solucom.fr @gbillois

Sébastien Gioria
Chapter Leader OWASP France

OWASP News & Update





- Nouveaux projets :
 - OWASP Scada Project
 - OWASP Periodic Table of Vulnerabilities Project
 - OWASP OpenStack Security Project

- Mises a jour :
 - OWASP Zap 2.0 !
 - OWASP Anti-Sammy 1.5
 - OWASP iGoat



- Publication de l'OWASP Top10 2013
 - Prévu courant Avril/Mai 2013
 - Fin de l'appel à commentaires 30/03/2013
 - => OWASP-TopTen@lists.owasp.org.
- Traduction en français prévue
 - Nombreux traducteurs; liste close maintenant.



OWASP

The Open Web Application Security Project

A1: Injection

**A2: Mauvaise
gestion des
sessions et de
l'authentification**

**A3: Cross Site
Scripting (XSS)**

**A4: Référence
directe non
sécurisée à un
objet**

**A5: Mauvaise
configuration
sécurité**

**A6 : Exposition de
données sensibles**

**A7 : Mauvais
contrôle d'accès**

**A8: Cross Site
Request Forgery
(CSRF)**

**A9: Utilisation de
composants non
sécurisés**

**A10: Mauvaise
gestion des
redirections et des
transferts**



- GSDays 2013 : <http://www.gsdays.fr>
 - 4 Avril 2013
 - OWASP France Partenaire (code de réduction sur demande) - HTML5 et la sécurité, un point d'étape
- OWASP EU Tour 2013 :
 - Planifié entre Avril et Juin 2013
- AppSec Research Europe 2013 : 20/23 Aout –
Hambourg – Allemagne
- OWASP Benelux : 28/29 Novembre 2013



- Différentes solutions :
 - Membre Individuel : 50 \$
 - Membre Entreprise : 5000 \$
 - Donation Libre



- Soutenir juste le chapitre France :
 - Single Meeting supporter
 - Nous offrir une salle de meeting !
 - Participer par un talk ou autre !
 - Donation simple
 - Local Chapter supporter :
 - 500 \$ à 2000 \$



OWASP

The Open Web Application Security Project

- Juin 2013
 - Salle : a définir
 - Speaker : a définir
- Septembre 2013
 - Salle : a définir
 - Speaker : a définir
- Novembre 2013
 - Salle : a définir
 - Speaker : a définir

Q&A



OWASP

The Open Web Application Security Project



Ludovic Petit

+33 (0) 6 11 72 61 64

Ludovic.Petit@owasp.org

Chapter Leader OWASP France
Global Connections Committee



Sébastien Gioria

+33 (0) 6 70 59 11 44

Sebastien.Gioria@owasp.org

Chapter Leader OWASP France
Global Education Committee



Ely de Travieso

+33 (0) 6 29 42 42 86

Ely.deTravieso@owasp.org

Relations Partenaires
OWASP France