



# OWASP Testing Guide

OWASP Training  
Paris – France  
26 Avril 2011

Sébastien Gioria (French Chapter Leader & OWASP Global  
Education Committee Member)

[sebastien.gioria@owasp.org](mailto:sebastien.gioria@owasp.org)

Copyright © 2009 - The OWASP Foundation  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License.

The OWASP Foundation  
<http://www.owasp.org>

---

# Agenda

- Historique
- Cible
- OWASP Testing Framework
- OWASP Testing Guide
- OWASP Risk Methodology Scoring

# Un peu d'histoire

- *Juillet 2004 => v1 :*
  - ▶ *OWASP Web Application Penetration Checklist*
  
- *Décembre 2006 => v2*
  - ▶ *OWASP Testing Guide v2.0*
  
- **Novembre 2008 => v3**
  - ▶ **OWASP Testing Guide v3.0**
  
- **2011 => v4**
  - ▶ **OWASP Testing Guide v4.0**



---

# Objectif du Guide v3

- Améliorer la v2 😊
- Créer un projet complet de test d'intrusions Web
- Devenir une référence pour le test des applications Web.
- Décrire la méthode de tests OWASP



# A qui s'adresse ce guide ?

- Développeurs :

- ⇒ Eviter les failles

- Equipes de Tests :

- => Améliorer les produits en ajoutant des tests sécurité

- Spécialistes de la sécurité / Auditeurs Sécurité :

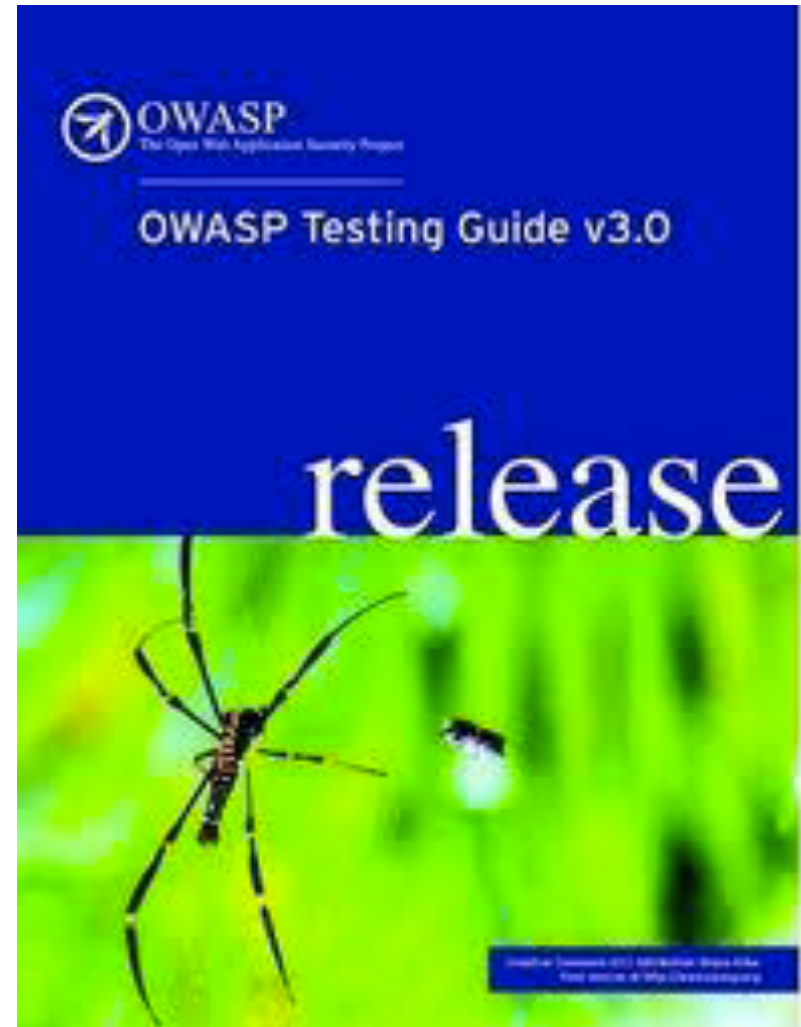
- ⇒ Vérifier que les produits/logiciels sont exempts de failles.

- ⇒ Disposer d'un référentiel commun et exhaustif



# Le contenu

1. Introduction
2. The OWASP Testing Framework
3. Web Application Penetration Testing
4. Writing Reports: value the real risk
5. Appendix A: Testing Tools
6. Appendix B: Suggested Reading
7. Appendix C: Fuzz Vectors
8. Appendix D: Encoded Injection



# Le contenu des tests

## ■ 66 tests répartis en 10 catégories :

- ▶ Découverte d'informations
- ▶ Gestion de la configuration
- ▶ Logique Métier
- ▶ Authentification
- ▶ Habilitations
- ▶ Gestion des sessions
- ▶ Validations des données
- ▶ Déni de service
- ▶ Web Services
- ▶ Ajax



# Le framework de test OWASP



Before testing



Definition and Design



Development



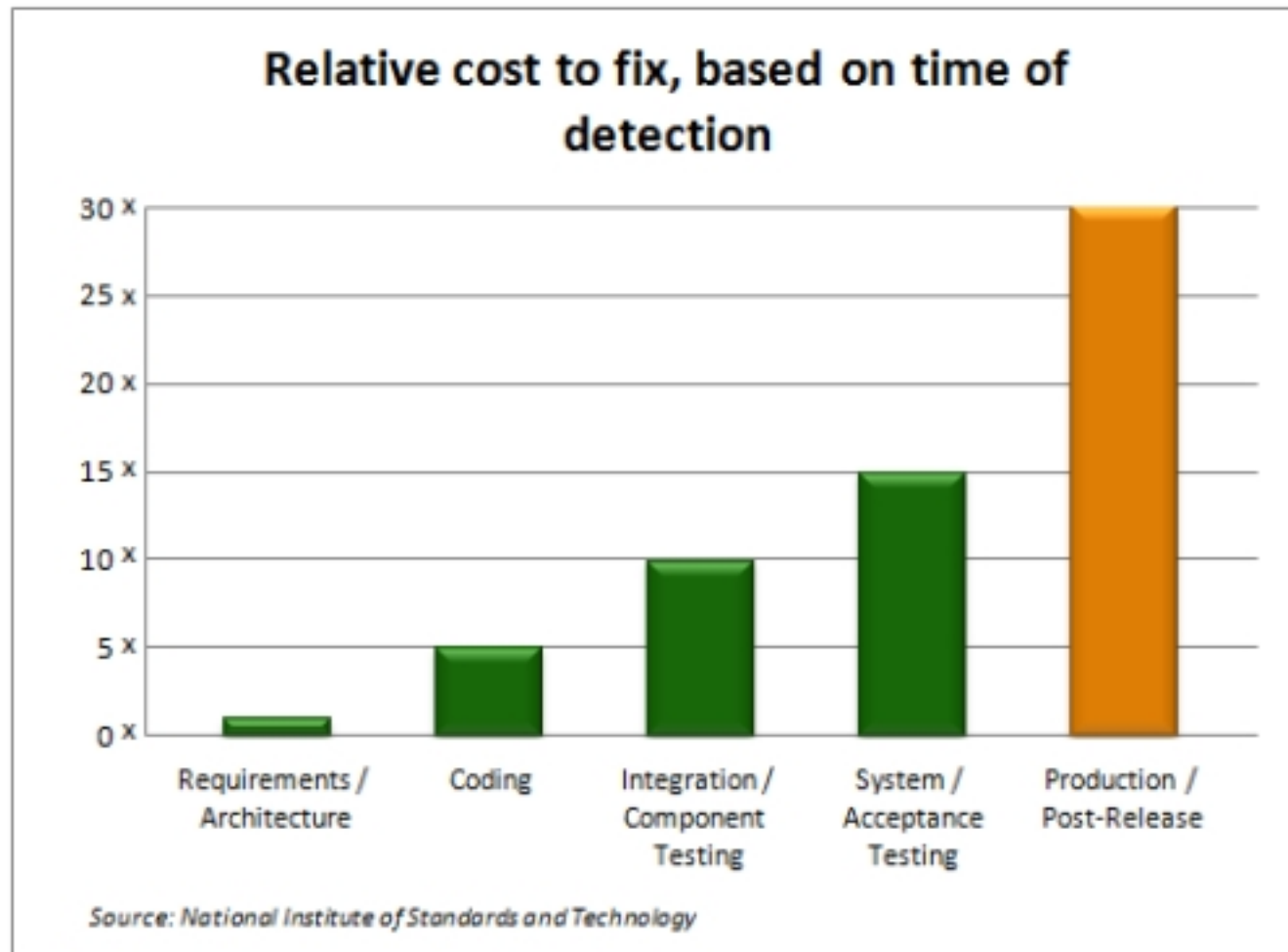
Deployment



Maintenance



# Le cout est important



# Inspections manuels et revues

## ■ Avantages

- ▶ Flexible
- ▶ Approprié a toute situation
- ▶ Encourage le travail en groupe
- ▶ Très tot dans le cycle
- ▶ Pas nécessaire de connaître les technologies

## ■ Inconvénients

- ▶ Peut etre trop long
- ▶ La documentation n'est pas toujours disponible
- ▶ Temps purement humain

# Modélisation des attaques

## ■ Avantages

- ▶ Vu d'un attaquant
- ▶ Flexible
- ▶ Tot dans le cycle

## ■ Inconvénients

- ▶ Technique assez récente
- ▶ Un bon modèle ne veut pas dire un bon logiciel



# Revue de code source

## ■ Avantages

- ▶ Rapide
- ▶ Précis
- ▶ Complet et efficient

## ■ Inconvénients

- ▶ Nécessite de très bon développeurs connaissant la sécurité
- ▶ Il est possible de louper des failles du aux librairies
- ▶ Difficile de détecter lors des executions
- ▶ Le code déployé peut etre différent du code analuyser



# Tests d'intrusions

## ■ Avantages

- ▶ Rapide (et parfois peu chers)
- ▶ Connaissance plus faible que lors de la revue de code
- ▶ Test du code exposé

## ■ Inconvénients

- ▶ Trop tard dans le cycle
- ▶ Visibilité uniquement de l'impact frontal



---

**Before Testing**  
*People can only do the right thing, if they know what the right thing is.*

Policy Review

Standards Review



# Definition & Design

Requirement Review

Design & Architecture Review

Create & Review UML Review

Create & Review Threat Model



---

# Development

Code Review

Code Walkthroughs

Unit & System Tests





# Deploiement

Acceptance Tests

Units & Systems Tests

Config & Management Review

Penetration Testing



# Maintenance

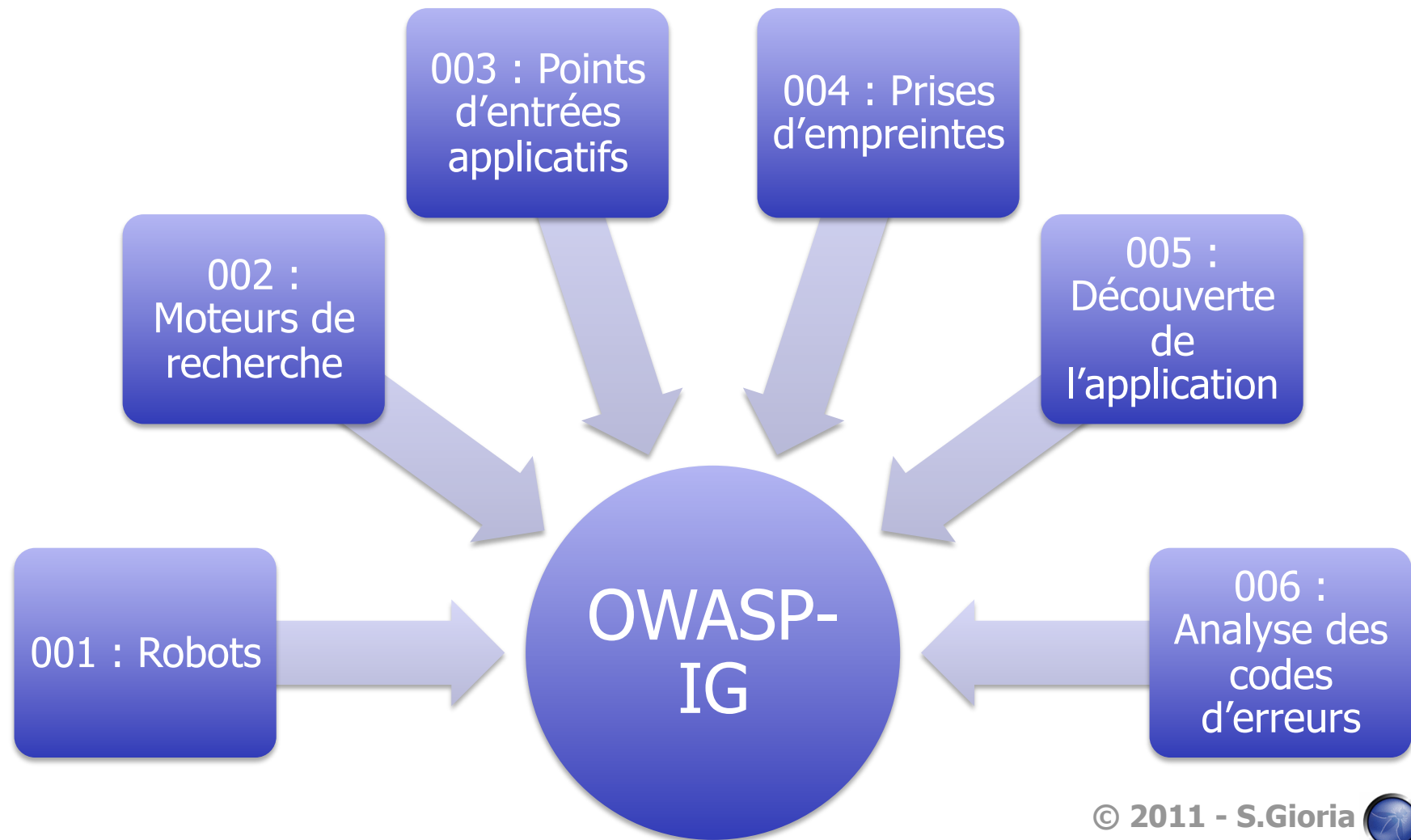
Change Verification

Health Checks

Operationnal Management  
reviews

Change Verification

# Découvertes d'informations



#### 4.8.5.3 SQL SERVER TESTING

##### BRIEF SUMMARY

In this section some [SQL Injection](#) techniques that utilize specific features of Microsoft SQL Server will be discussed.

##### SHORT DESCRIPTION OF THE ISSUE

SQL injection vulnerabilities occur whenever input is used in the construction of an SQL query without being adequately constrained or sanitized. The use of dynamic SQL (the construction of SQL queries by concatenation of strings) opens the door to these vulnerabilities. SQL injection allows an attacker to access the SQL servers and execute SQL code under the privileges of the user used to connect to the database.

As explained in [SQL injection](#), a SQL-injection exploit requires two things: an entry point and an exploit to enter. Any user-controlled parameter that gets processed by the application might be hiding a vulnerability. This includes:

- Application parameters in query strings (e.g., GET requests)
- Application parameters included as part of the body of a POST request
- Browser-related information (e.g., user-agent, referrer)
- Host-related information (e.g., host name, IP)
- Session-related information (e.g., user ID, cookies)

Microsoft SQL server has a few unique characteristics, so that some exploits need to be specially customized for this application.

##### BLACK BOX TESTING AND EXAMPLE

### Example 3: Testing in a POST request

SQL Injection, HTTP POST Content: email=%27&whichSubmit=submit&submit.x=0&submit.y=0

A complete post example:

```
POST https://vulnerable.web.app/forgotpass.asp HTTP/1.1
Host: vulnerable.web.app
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.0.7) Gecko/20060909
Firefox/1.5.0.7 Paros/3.2.13
Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*
;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Proxy-Connection: keep-alive
Referer: http://vulnerable.web.app/forgotpass.asp
Content-Type: application/x-www-form-urlencoded
Content-Length: 50

email=%27&whichSubmit=submit&submit.x=0&submit.y=0
```

The error message obtained when a ' (single quote) character is entered at the email field is:

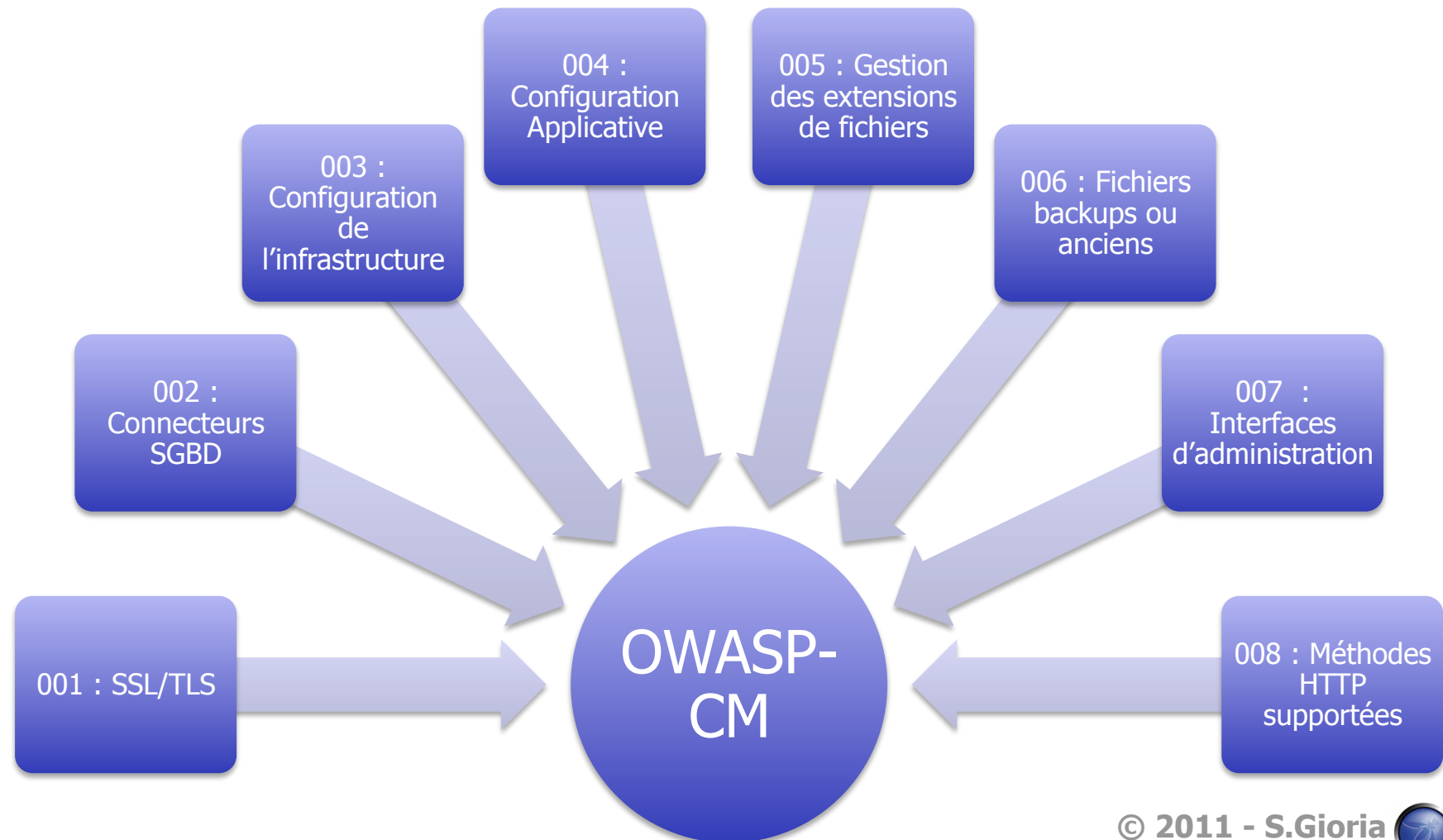
```
Microsoft OLE DB Provider for SQL Server error '80040e14'
```

Unclosed quotation mark before the character string '.

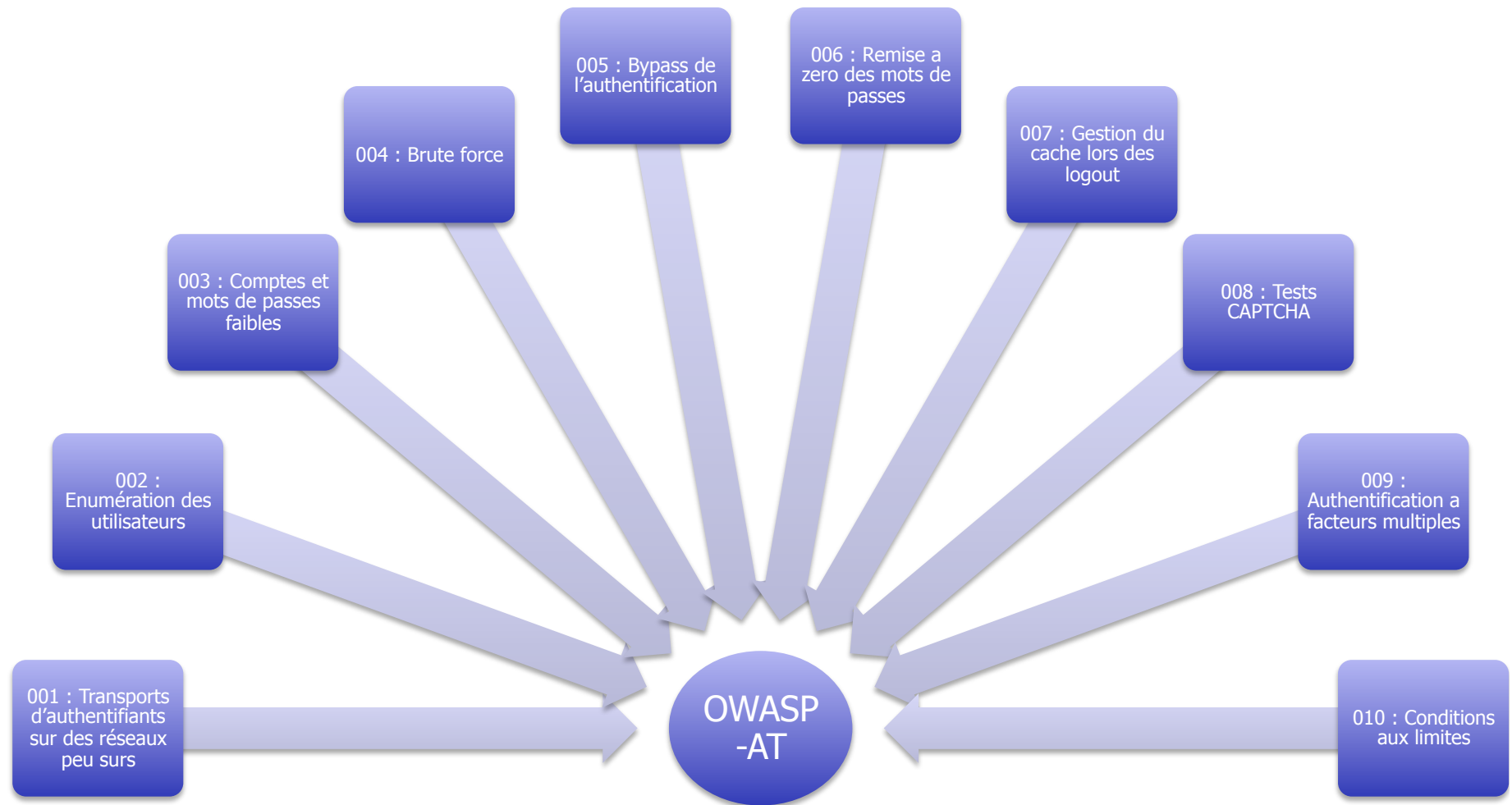
```
/forgotpass.asp, line 15
```

**Example 4: Yet another (useful) GET example**

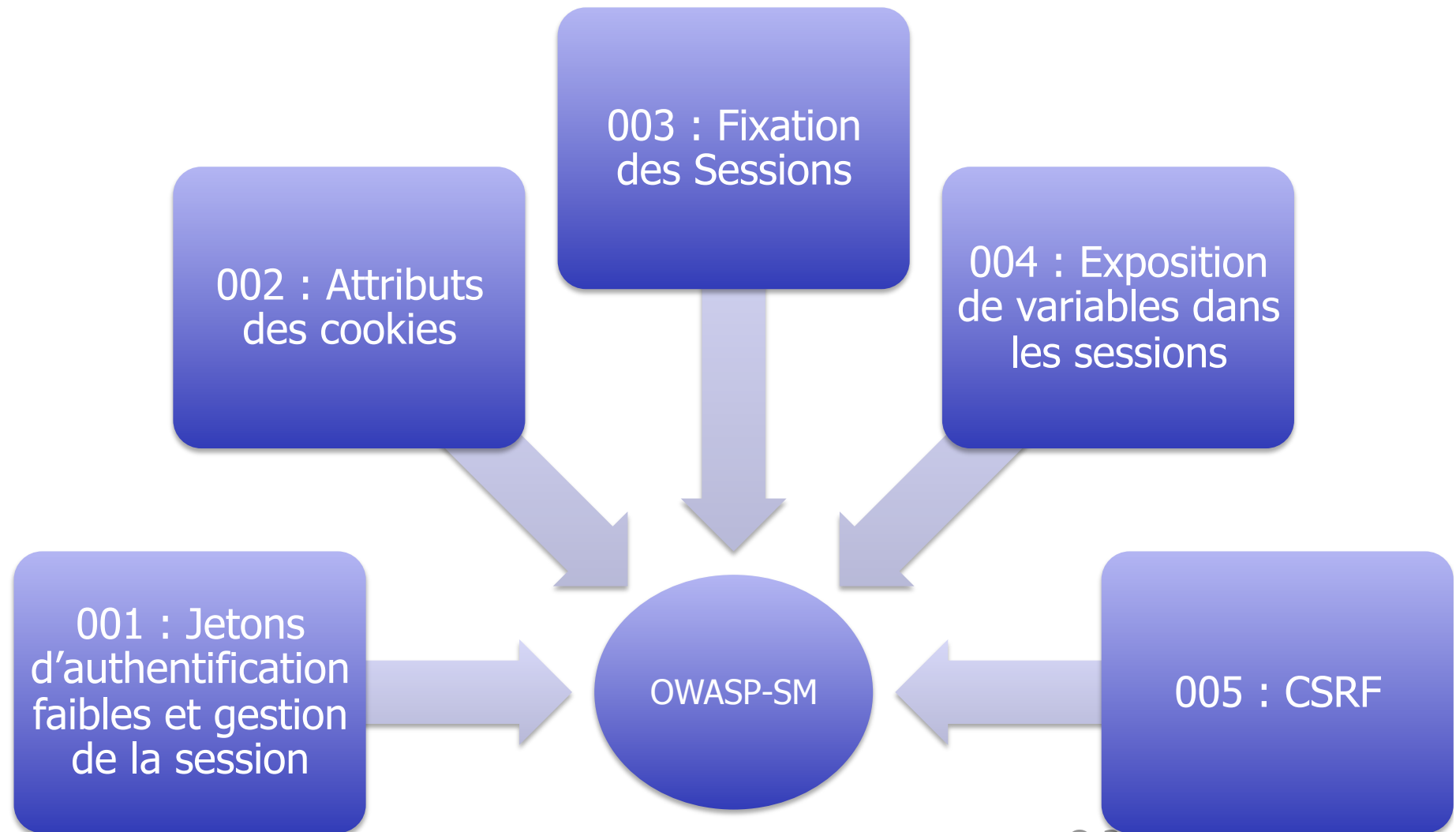
# Gestion de la configuration



# Authentification

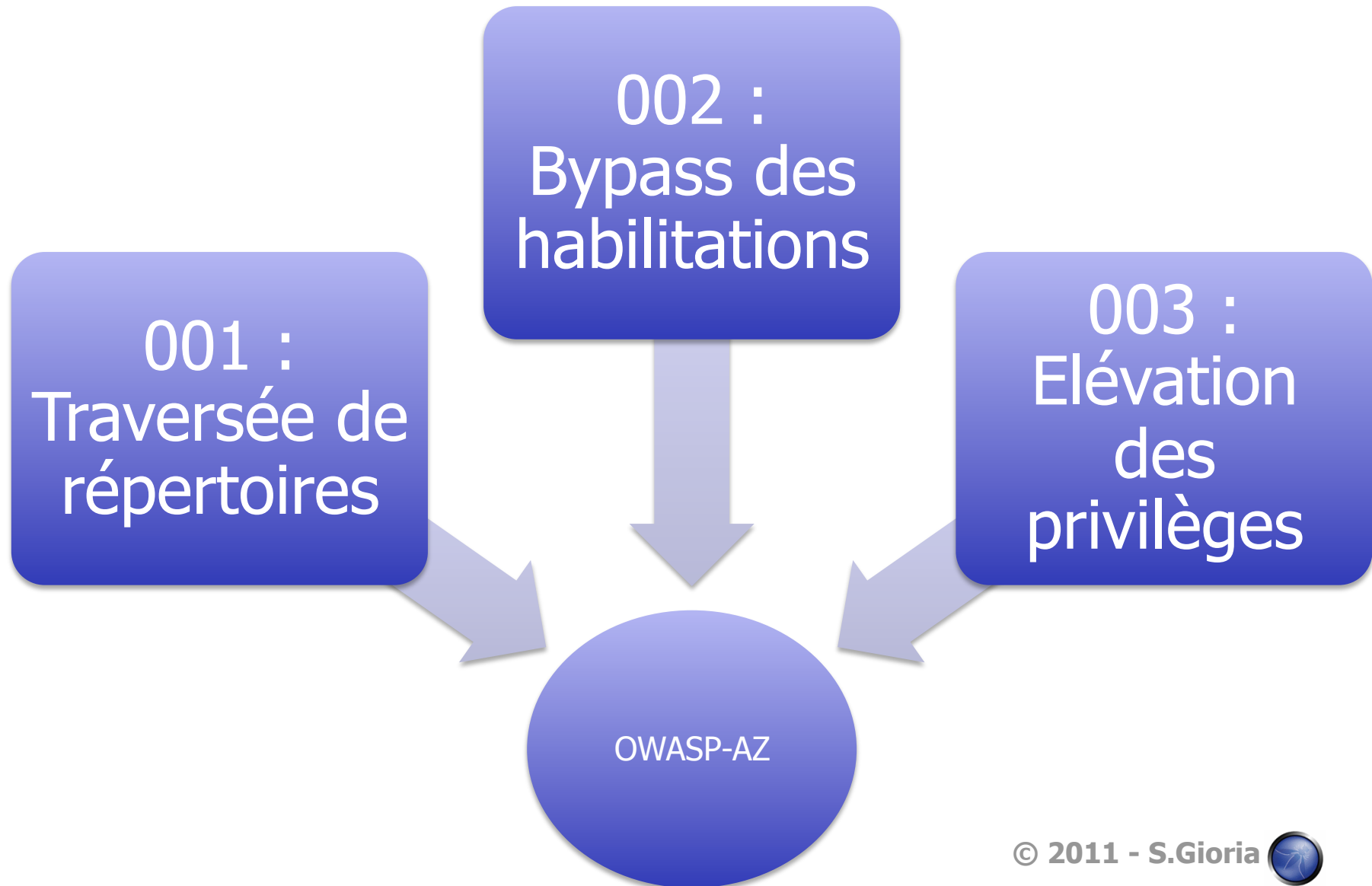


# Gestion des Sessions



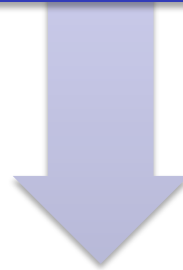


# Habilitations



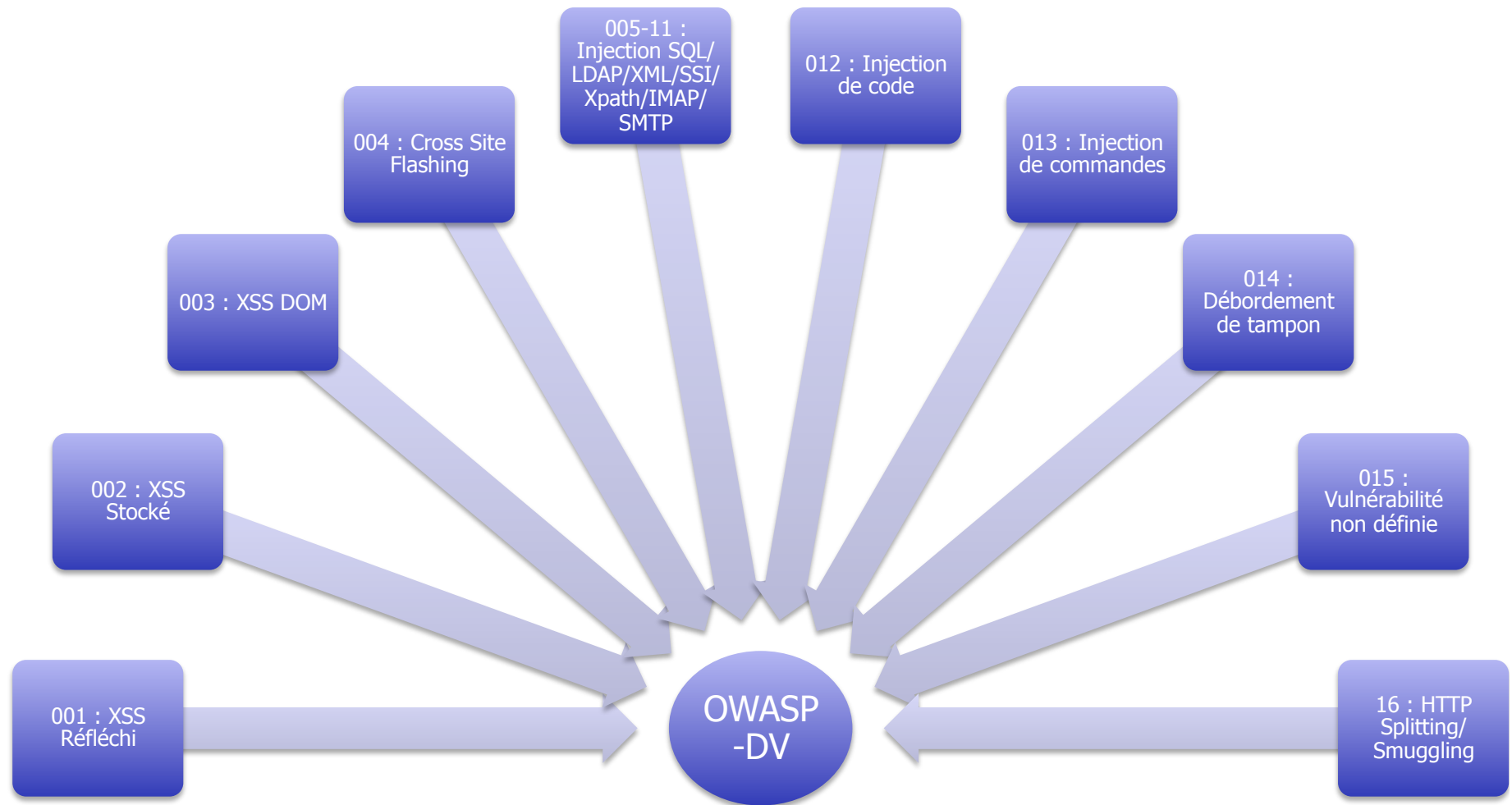
# Logique Métier

001 :Tests  
de la logique  
métier

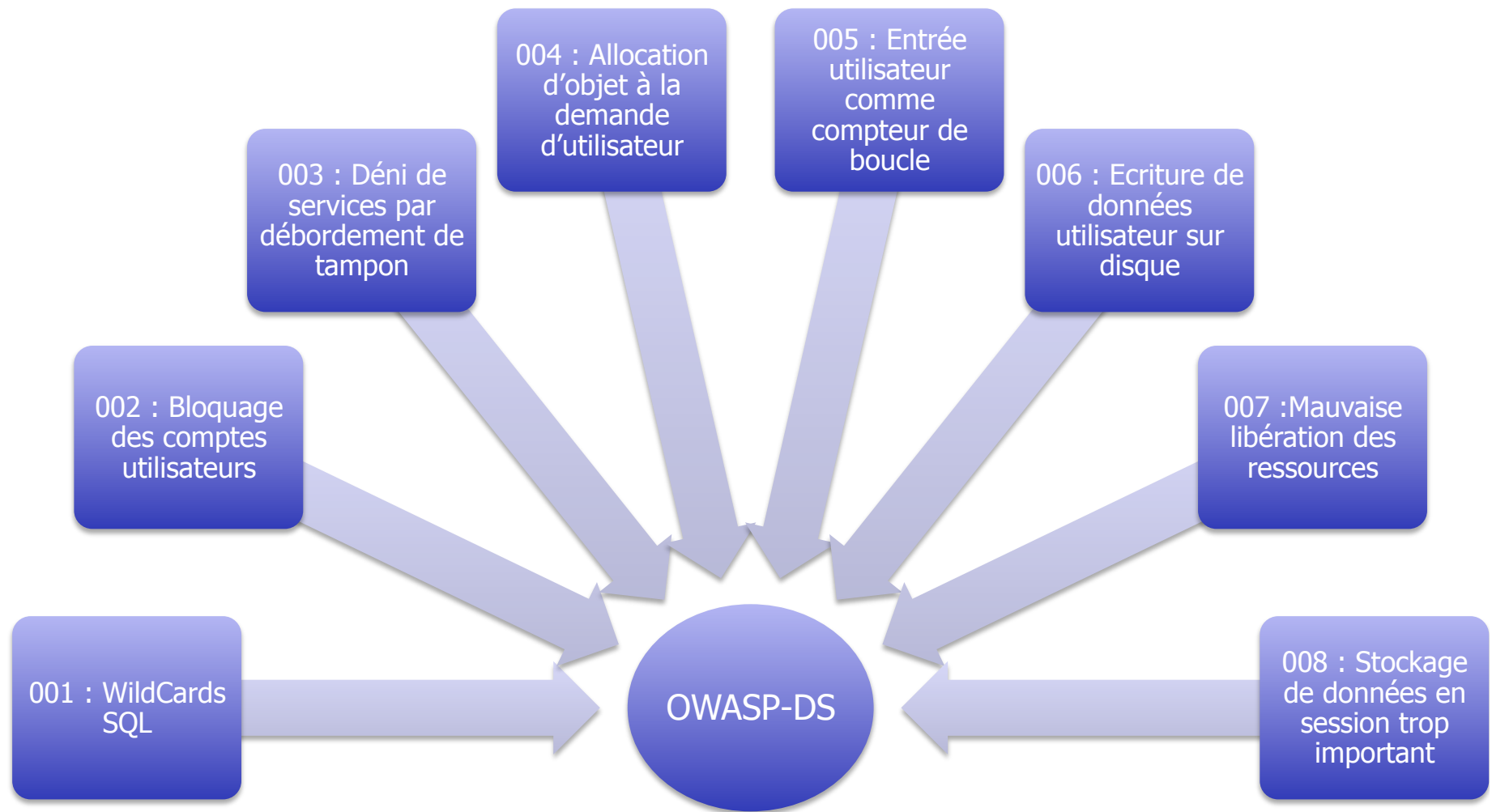


OWASP-BL

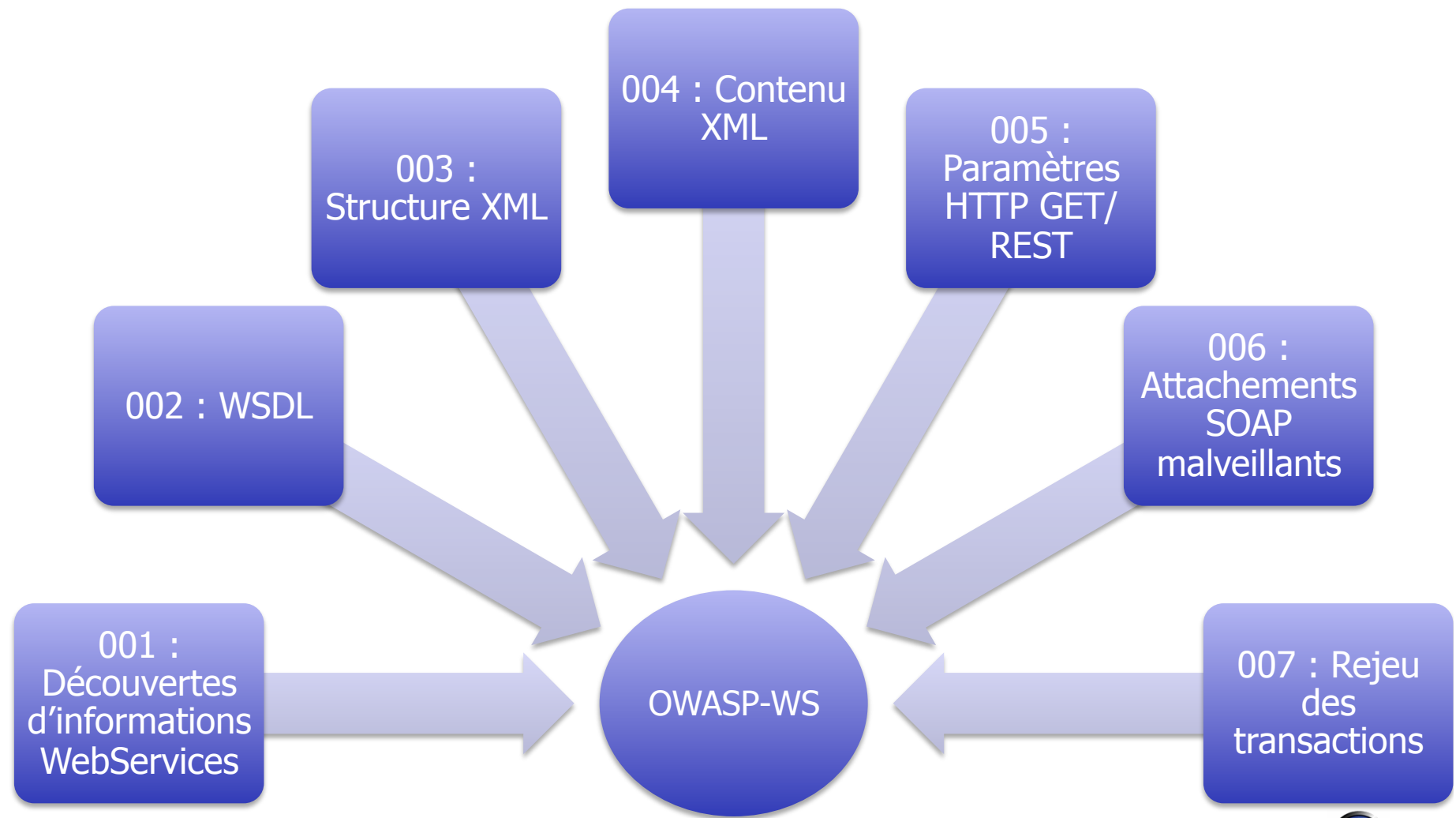
# Validation des données



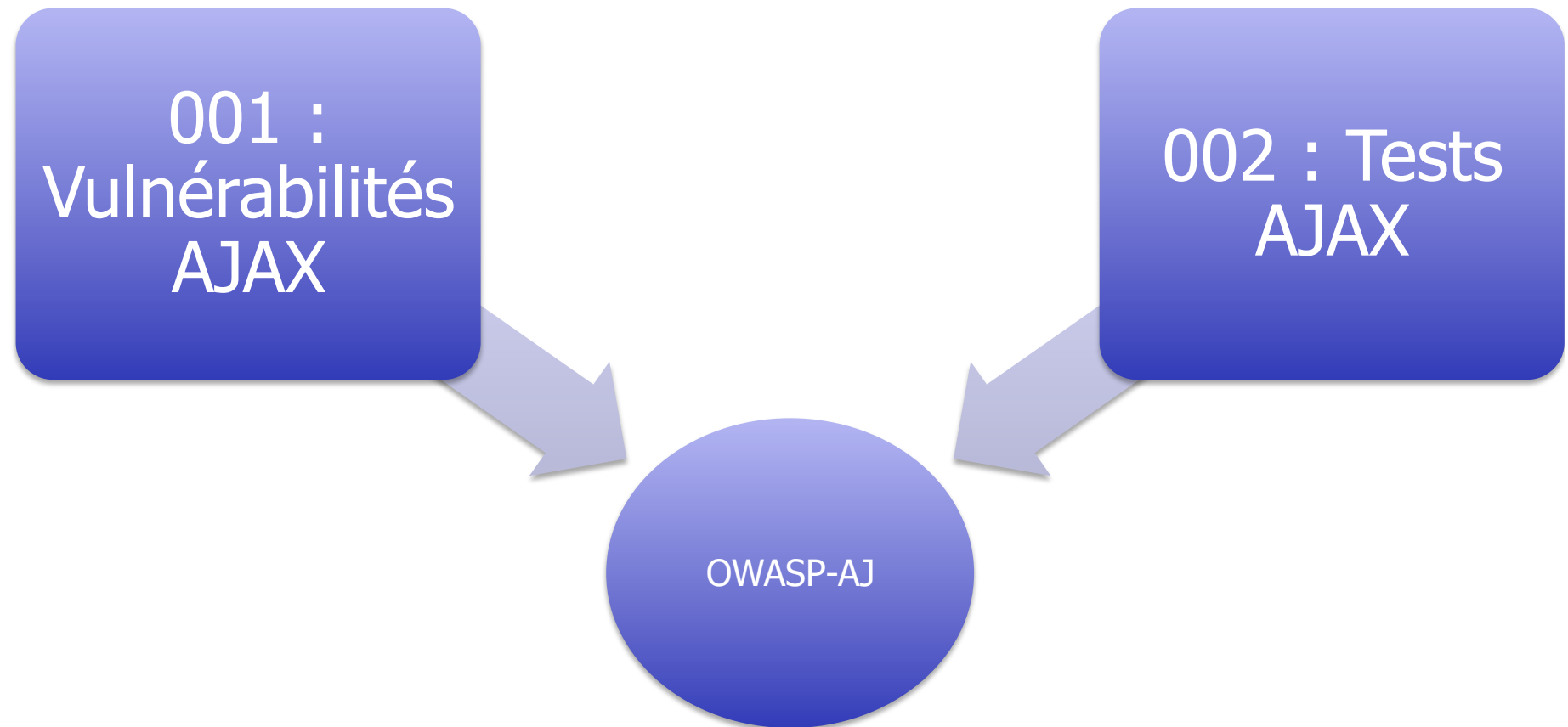
# Déni de service



# Web Services



# Ajax



# L' évaluation du risque

- Nous utilisons la méthodologie « *OWASP Risk Rating* » qui permet de s'adapter à tout environnement.

$$\textbf{Risque} = \textbf{Probabilité} * \textbf{Impact}$$

- Cette méthodologie se base sur 3 éléments principaux composés par différents métriques, calculés par la méthode de la moyenne:
- La probabilité de l'attaque qui se base sur :
  - ▶ Les menaces pesant sur le système: compétences de l'attaquant, motivations, ressources nécessaires, type d'attaquant
  - ▶ La description de la vulnérabilité : facilité d'exploitation et de découverte, connaissance de la vulnérabilité, capacité de détection et de mitigation.
- L'impact Technique : pertes en confidentialité, disponibilité, intégrité et tracabilité
- L'impact Business : pertes financières, d'image, non conformité, et violation de la vie privée.

Niveau de probabilité et d'impact	
0 à <3	Faible
3 à <6	Moyen
6 à 9	Elevé

## Evaluation du risque

- Le calcul suivant permet ensuite d'évaluer le risque :

Risque				
Impact	Elevé	Moyen	Elevé	Critique
	Moyen	Faible	Moyen	Elevé
	Faible	Nul	Faible	Moyen
		Faible	Moyen	Elevé
	Probabilité			



<b><u>Menace</u></b>		<b><u>Vulnérabilité</u></b>	
<b><u>Expertise</u></b>	<b><u>Moyens</u></b>	<b><u>Découverte</u></b>	<b><u>Connaissance</u></b>
None	1 Complexe	0 Impossible	1 Inconnu
Faible	3 Moyen	4 Difficile	3 Caché
Moyenne	4 Faible	7 simple	6 Evident
Avancée	6 Aucun	9 Très simple	9 Public
Expert	9		

<b><u>Motivation</u></b>	<b><u>Typologie</u></b>	<b><u>Exploitation</u></b>	<b><u>Contre Mesure</u></b>
Faible	1 Développeur	2 Théorique	1 Applicative
Moyenne	4 Administrateur Système	3 Difficile	3 Loggé et revu
Elevée	9 Utilisateur interne	5 simple	8 Loggé et non revu
	Partenaire	9 Très simple	9 Non loggé
	Utilisateur authentifié		
	Utilisateur anonyme		

### **Impact Technique**

<b><u>Confidentialité</u></b>	<b><u>Disponibilité</u></b>	<b><u>Financier</u></b>	<b><u>Non conformité</u></b>
Faible sur données non critique	2 Faible sur services secondaires	1 Mineur	2 Faible
Faible sur données critiques	3 Faible sur services primaires	3 Faible	5 Avérée
Fort sur données non critiques	5 Fort sur services secondaires	7 Signifiant	5 Complète
Fort sur données critiques	7 Fort sur services primaires	9 Stratégique	7 Stratégique
Toutes les données	9 Tous les services		











<b><u>Intégrité</u></b>	<b><u>Tracabilité</u></b>	<b><u>Réputation</u></b>	<b><u>Données personnelles</u></b>
Faible sur peu de données	1 Totalement tracable	1 Mineur	3 Une personne
Faible sur un fort nombre de données	3 En partie tracable	4 Majeur	5 Des centaines
Forte sur peu de données	5 Impossible a tracer	5 Fort	7 Des milliers
Fort sur un fort nombre de données	7	9 Stratégique	9 Des millions

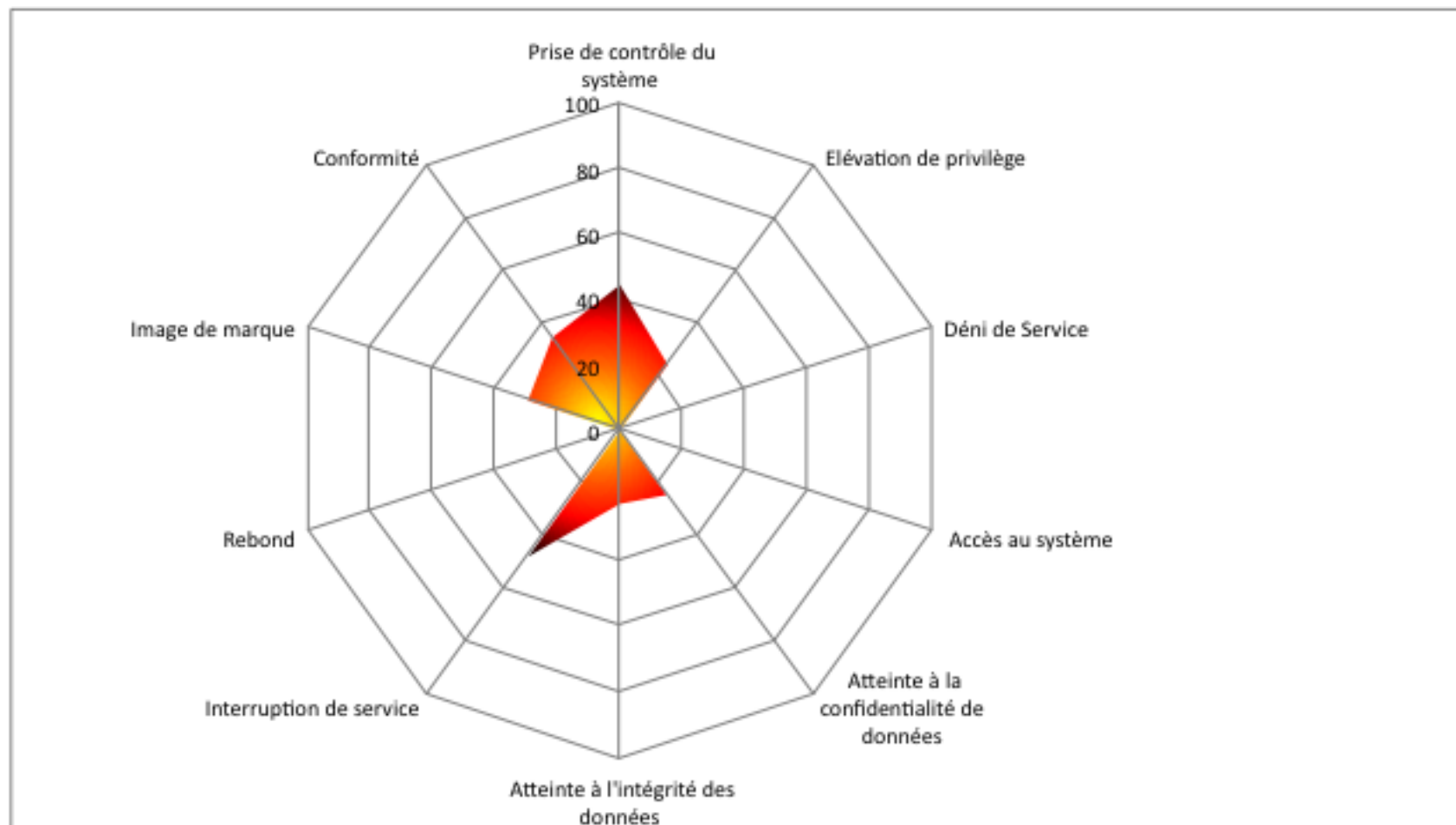


# Exemple

		Menace				Vulnérabilité				Impact technique				Impact Métier			
		Expertise	Motivation	Moyens	Typologie	Découverte	Exploitation	Connaissance	Contre-Mesure	Confidentialité	Intégrité	Disponibilité	Tracabilité	Financier	Réputation	Non-conformité	Données personnelles
AT-003	Valeur	3	4	8	9	9	9	5	1	9	3	0	1	5	9	7	9
	Métrique	6								3,25				7,5			
	Probabilité	Elevé															
	Impact									Moyen				Elevé			
	Impact Final									Elevé							
	Risque Final	Critique															

# Exemple

Référence	Description du test	Résultat	Risque	Solution	Référence
OWASP-AT-001	Transport d'éléments d'authentification sur des canaux non chiffrés		Critique	Simple	GFR_CM_001-1
OWASP-AT-002	Enumération des utilisateurs		Critique	Complexe	GFR-AT-002-1 GFR-AT-002-2
OWASP-AT-004	Attaque de l'authentification par force brute		Critique	Difficile	GFR-AT-002-1 GFR-AT-002-2
OWASP-CM-001	Tests SSL/TLS		Elevé	Moyenne	GFR-CM-001-1
OWASP-DV-001	Cross Site Scripting réfléchi		Elevé	Moyenne	GFR-DV-001_1 GFR_DV-001_2 GFR_DV-001_3 GFR-DV-002_1 GFR_DV-002_2 GFR_DV-002_3 GFR-DV-002_4 GFR_DV-002_5 GFR_DV-002_6
OWASP-DV-002	Cross Site Scripting stocké		Elevé	Moyenne	GFR_DV-002_6
OWASP-DV-015	Vulnérabilité non connue		Moyen	Difficile	GFR_DV-015_1
OWASP-IG-004	Prise d'empreintes du serveur Web.		Moyen	Moyenne	GFR_IG-004-1
OWASP-IG-005	Découverte de l'application		Moyen	Moyenne	GFR_IG-005-1
OWASP-WS-001	Découverte des points d'entrée WebServices		Moyen	Moyenne	GFR-WS-001-1



# Exemple

