



# Sécurité Applicative

M1 WEB - OWASP Top 10  
Me. 26 Juin 2019 - PHELIZOT Yvan

# Top 10 OWASP

# Top 10 OWASP

- OWASP : Open Web Application Security Project
- Top 10 : les 10 risques les plus courantes
  - Top 10 WEB, Top 10 Mobile
- Autre classement : SANS 25
- Existe depuis 2004, v5
- Propose de nombreux guides à destination des managers, développeurs, testeurs, ...

# OWASP Top 10 de 2007 à nos jours

2007	2010	2013	2017
A1 - XSS	A1-Injection	A1-Injection	A1-Injection
A2 - Injection flaws	A2-Cross Site Scripting (XSS)	A2-Broken Authentication and Session Management	A2-Broken Authentication
A3 - Malicious File Execution	A3-Broken Authentication and Session Management	A3-Cross-Site Scripting (XSS)	A3-Sensitive Data Exposure
A4 - Insecure Direct Object Reference	A4-Insecure Direct Object References	A4-Insecure Direct Object References	A4-XML External Entities
A5 - CSRF	A5-Cross Site Request Forgery (CSRF)	A5-Security Misconfiguration	A5-Broken Access Control

# OWASP Top 10 de 2007 à nos jours

2007	2010	2013	2017
A6 - Information Leakage and Improper Error Handling	A6-Security Misconfiguration	A6-Sensitive Data Exposure	A6-Security Misconfiguration
A7 - Broken Authentication and Session Management	A7-Insecure Cryptographic Storage	A7-Missing Function Level Access Control	A7-XSS
A8 - Insecure Cryptographic Storage	A8-Failure to Restrict URL Access	A8-Cross-Site Request Forgery (CSRF)	A8-Insecure Deserialization
A9 - Insecure Communications	A9-Insufficient Transport Layer Protection	A9-Using Components with Known Vulnerabilities	A9-Using Components with Known Vulnerabilities
A10 - Failure to Restrict URL Access	A10-Unvalidated Redirects and Forwards	A10-Unvalidated Redirects and Forwards	A10-Insufficient Logging & Monitoring

# Format

Threat Agents / Attack Vectors		Security Weakness		Impacts	
App Specific	Exploitability: 3	Prevalence: 2	Detectability: 2	Technical: 3	Business ?

A1 - Injection

# 1 - Injection

Definition : send untrusted data to an interpreter to change the course of normal execution

Three elements

- Untrusted source of data
- An interpreter
- Something which calls the interpreter and send data

Risks

- Modifying behavior of system



# 1 - Example : SQL Injection in Java

```
ResultSet res = st.executeQuery('SELECT * FROM  
User where userId=' + user + ' and password='  
+ password+'');
```

What happens if password == 'XXX' or '1'='1'?

Result :

```
SELECT * FROM User where userId='admin' and  
password='XXX' or '1'='1'
```

# 1 - Example : SQL Injection in Java

HQL : Take care!

```
session.createQuery("from User where userId=' ' +  
user + ' ' and password=' ' + password+' ' );
```

# 1 - Injection

- How to fix it?
  - Validate input
  - Escape input
  - Sanitize input
  - Limit rights (only allow a user to read certain tables)
  - Limit resources
  - SQL : Prepared Statements

Web Goat

```
var b64img = window.location.hash.substr(1);
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
  if (this.readyState == 4 && this.status == 200) {
    var reader = new FileReader();
    reader.onloadend = function() {
      document.write(`
<a href="${b64img}" alt="${atob(b64img)}">
  
</a>`);
    }
    reader.readAsDataURL(this.response);
  }
};
xhttp.responseType = 'blob';
xhttp.open("GET", b64img, true);
xhttp.send();
```

**Où est la faille?**  
**Comment l'exploiter?**  
**Comment la détecter?**  
**Comment la corriger?**

A2 - Broken authentication

# A2 - Broken Authentication

Définition : le processus d'authentification ne joue pas son rôle

- Authentification affaiblie
- Pas de MFA
- Mauvaise gestion des sessions

Risques

- Access to account

## A2 - Broken Authentication

- How to fix it?
  - 2-FA, M-FA
  - Format du mot de passe
  - Interdire les mots de passe trop simples
  - Interdire les mots de passe trop courants
  - Limiter le nombre de tentatives, en fonction de l'IP, de l'heure (soir) ou de la date (week-end), ...
  - Respecter les guidelines pour les sessions



Web Goat

# A3 - Sensitive Data Exposure

# A3 - Sensitive Data Exposure

Définition : des informations sensibles sont affichées

- Chiffrement incorrect

Risques

- Confidentialité

# A3 - Sensitive Data Exposure

- How to fix it?
  - Demander de l'aide à un expert
  - Utiliser des outils de vérification
  - Changer les secrets
  - Respecter les guidelines de sécurité

Web Goat

A4 - XXE

# A4 - XXE

Définition : exploitation des moteurs XML

- Exécution de XML non sûr

Risques

- Lecture de fichier
- DoS

# XXE example

```
<?xml version="1.0"?>
```

```
<!DOCTYPE lolz [
```

```
<!ENTITY lol "lol">
```

```
<!ELEMENT lolz (#PCDATA)>
```

```
<!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
```

```
<!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
```

```
<!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
```

```
<!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
```

```
<!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
```

```
<!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
```

```
<!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
```

```
<!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
```

```
<!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
```

```
]>
```

```
<lolz>&lol9;</lolz>
```



# A4 - XXE

- How to fix it?
  - Limiter les droits du moteur (processus indépendant, isolé, utilisateur avec des droits limités, quota linux, configuration...)
  - Echappement/Validation

Web Goat

# A5 - Broken Access Control

# A5 - Broken Access Control

Définition : Accéder à des informations sans avoir les autorisations

- Elevation of privilege
- CORS
- Insecure Direct Object References
- Missing Function Level Access Control

Risques

- Utiliser des fonctions non autorisées
- Manipuler à des données incorrectes

# A5 - Broken Access Control

- How to fix it?
  - Complete mediation : vérifier tous les droits
  - RBAC/ACL
  - Deny by default

# A5 - Broken Access Control

- Exemple avec Apache

Order Allow,Deny

Deny from 127.0.0.1

...

Web Goat

# A6 - Security Misconfiguration



# A6 - Security Misconfiguration

Définition : exploiter une faiblesse dans la configuration

- Pages d'installation laissées
- Droits insuffisants sur un dossier
- Stacktrace

Risques

- Configuration à la main de l'attaquant
- Accès à des informations du système

# Google Hacking Database

- Essayons de trouver la page d'installation Wordpress d'un site



Entrez ci-dessous les détails de connexion à votre base de données. Si vous ne les connaissez pas avec certitude, contactez votre hébergeur.

<b>Nom de la base de données</b>	<input type="text" value="wordpress"/>	Le nom de la base dans laquelle vous voulez installer WP.
<b>Identifiant</b>	<input type="text" value="username"/>	Votre identifiant MySQL.
<b>Mot de passe</b>	<input type="text" value="password"/>	...et votre mot de passe MySQL.
<b>Hôte de la base de données</b>	<input type="text" value="localhost"/>	Dans 99% des cas, vous n'aurez pas à modifier cette valeur.
<b>Préfixe de table</b>	<input type="text" value="wp_"/>	Si vous voulez installer plusieurs blogs WordPress dans une même base de données, modifiez ce champ.

Valider

# Stacktrace

```
org.h2.jdbc.JdbcSQLException: Column count does not match; SQL statement:  
INSERT INTO USERS(NAME, PASSWORD) values("", "") [21002-196]  
    at org.h2.message.DbException.getJdbcSQLException(DbException.java:345)  
    at org.h2.message.DbException.get(DbException.java:179)  
    at org.h2.message.DbException.get(DbException.java:155)  
    at org.h2.message.DbException.get(DbException.java:144)  
    at org.h2.command.dml.Insert.prepare(Insert.java:265)  
    at org.h2.command.Parser.prepareCommand(Parser.java:263)  
    at org.h2.engine.Session.prepareLocal(Session.java:578)  
    at org.h2.engine.Session.prepareCommand(Session.java:519)
```

# A6 - Security Misconfiguration

- How to fix it?
  - Utiliser un scanner de vulnérabilités
  - Suivre les guidelines
  - Filtrer les informations (détection du mot de passe dans un message de log ⇒ suppression)

Web Goat

A7 - XSS

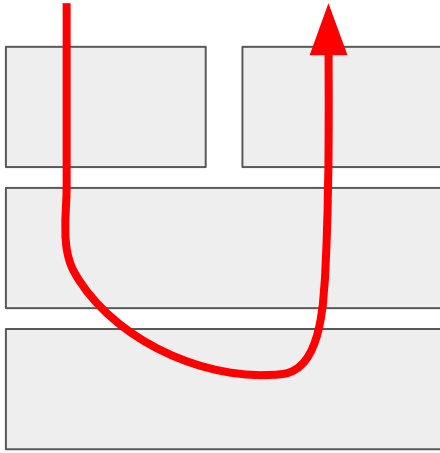
# A7 - Cross-Site Scripting (XSS)

- A type of injection
- Front page interprets unsafe data

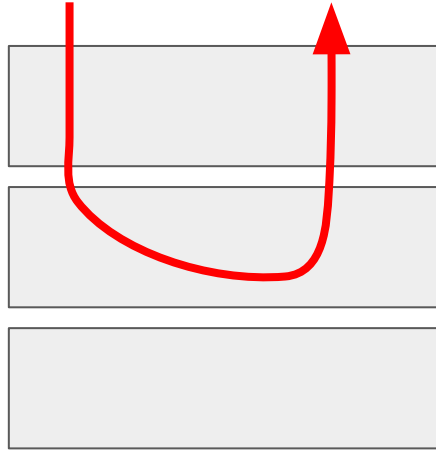
## Risks:

- Site defacing
- Cookie theft
- Redirection
- Malicious operation on corrupted website

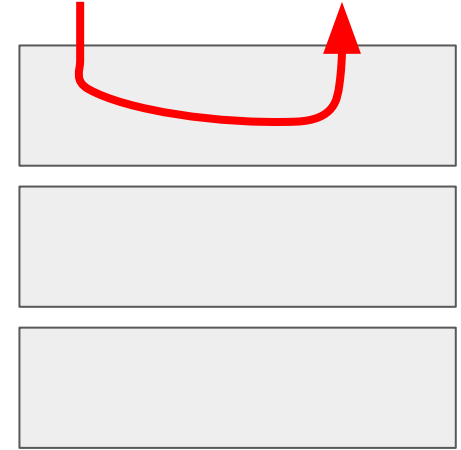
# Stored XSS / Reflected XSS / DOM-based XSS



Stored  
XSS



Reflected  
XSS



DOM-based  
XSS



# Don't Mix Code and Data (*Writing Secure Code*)

- Violation d'un principe de sécurité
- HTML DOM & Code Javascript
- Rend la page dynamique

# Cross-Site Scripting (XSS)

- How to fix XSS?
  - Like others injections (sanitize, escape, ...)
  - Use anti-xss frameworks (eg. Angular {{ }})
  - Use HTTP headers
    - X-XSS-Protection
    - CSP

Web Goat

# A9 - Using Components with Known Vulnerabilities

# A9 - Using Components with Known Vulnerabilities

- A type of injection
- Front page interprets unsafe data

## Risks:

- Site defacing
- Cookie theft
- Redirection
- Malicious operation on corrupted website

# A9 - Using Components with Known Vulnerabilities

- How to fix XSS?
  - Utiliser des libraires connues et reconnues
  - Utiliser un scanner de paquets vulnérables (OWASP Dependency Check, npm audit, SafeNuGet, ...)

# Demo NPM Audit

# Attention!

- Il existe beaucoup d'autres attaques/failles
  - Time attack
  - Race condition
  - Injection de fautes
  - Insecure Redirect
  - CRLF/Template/HTML Injection/SSRF
  - Integer Overflow
  - ...
- De nouvelles attaques (ré-)apparaissent régulièrement



# New solutions, new problems

- HTTP/2
- Worker
- WebSocket

# New protections

- CSP
- SRI (Sub Resource Integrity)

# Browser protection

- Sandboxing
- dépendance à l'OS : GDI Windows

CVE