

## **Bloc 3**

### **OWASP**

#### **Principales vulnérabilités des applications web**

Merci aux auteurs qui m'ont inspirée :  
Patrice DIGNAN, Pierre François ROMEUF et Yann BARROT.(Réseau certa)

# Table des matières

I Introduction.....	3
1 La sécurité des applications web.....	3
1.1 Les applications web sont partout.....	3
1.2 La sécurisation des applications web est indispensable.....	3
2 Les motifs des attaques.....	3
II Présentation d'OWASP.....	4
1 La communauté OWASP.....	4
2 Le top 10 d'OWASP 2017.....	5
III MUTILLIDAE et son installation.....	6
1 Installation de MUTILLIDAE.....	6
2 Installation de BurpSuite.....	8
IV Mise en garde juridique.....	9

# I Introduction

## 1 La sécurité des applications web

### 1.1 Les applications web sont partout

Aujourd'hui, les applications web sont partout. Elles sont utilisées quotidiennement dans nos activités personnelles ou professionnelles (réseaux sociaux, achats en lignes, démarches administratives...). Toute entreprise ou administration se doit d'avoir un site web. Ces applications facilitent les échanges et les transactions car elles sont accessibles de partout à l'aide d'un simple navigateur sur un smartphone ou un ordinateur de bureau.

Si au début des sites web, les aspects techniques et fonctionnels étaient suffisants, ce n'est plus du tout le cas aujourd'hui. L'actualité nous rappelle régulièrement que des entreprises voient leur site web attaqué. Les conséquences peuvent être lourdes (perte de données, baisse du chiffre d'affaire, effondrement de la réputation...). Avec comme enjeu, la survie de l'entreprise selon la gravité de l'attaque subie.

En outre, Le **règlement règlement général sur la protection des données (RGPD)**, mis en place au sein de l'Union Européenne, oblige les entreprises à assurer la sécurité des données personnelles qu'elles collectent.

### 1.2 La sécurisation des applications web est indispensable

La sécurité des applications web est donc devenue un enjeu stratégique. Lors de son édition 2016, la société EY (<http://www.ey.com/fr>) a montré qu'une majorité des entreprises mondiales n'a pas de stratégie en matière de lutte contre les cybermenaces<sup>1</sup>.

Au delà de l'aspect fonctionnel des outils de développement, il est indispensable pour tout développeur de savoir identifier les vulnérabilités potentielles et de prendre en compte les menaces en adaptant son développement à l'aide de bonnes pratiques. La phase de test ne doit pas se limiter au fonctionnement attendu du code mis en œuvre mais elle doit aussi anticiper les utilisations malveillantes comme les injections de code SQL dans les formulaires.

Afin de mettre en place une veille stratégique sur la sécurisation des applications web, le groupe OWASP a développé une base de données qui recense la liste des incidents de sécurité recensés sur les applications web. Cette base nommée WASC-WHID (Web application Security Consortium - Web Hacking Database Project) permet de disposer de statistiques sur les failles de sécurité relevées sur les applications web. Les incidents sont déclarés et enregistrés afin d'alimenter une base de connaissance.

Le lien permettant d'accéder aux outils WHID est le suivant :

[https://www.owasp.org/index.php/OWASP\\_WASC\\_Web\\_Hacking\\_Incidents\\_Database\\_Project](https://www.owasp.org/index.php/OWASP_WASC_Web_Hacking_Incidents_Database_Project)

---

<sup>1</sup> <http://www.lemondeinformatique.fr/actualites/lire-55-des-entreprises-mondiales-n-identifient-pas-les-vulnerabilites-67146.html>

## 2 Les motifs des attaques

Les sites web peuvent être attaqués pour plusieurs raisons :

MOTIFS	EXPLICATIONS
Propagande	Certaines personnes peuvent attaquer un site pour des raisons politiques ou pour défendre une cause particulière (hacktivistes).
Distraction	Des personnes peuvent voir dans l'attaque de sites web une distraction ou un défi à relever.
Vol de données	Le vol de données lucratif, pour effectuer du chantage ou pour copier le savoir d'un concurrent sur un marché.
Machine zombie	Le serveur cible est utilisé dans un réseau destiné à faire des attaques distribuées par déni de service (DDOS), du stockage de fichiers illégaux, de l'envoi de spams ...

## II Présentation d'OWASP

### 1 La communauté OWASP

OWASP (Open Web Application Security project) est une communauté travaillant sur la sécurité des applications web. Elle a pour but de publier des recommandations de sécurisation des sites web et propose des outils permettant de tester la sécurité des applications web.



Site officiel : <https://www.owasp.org/>

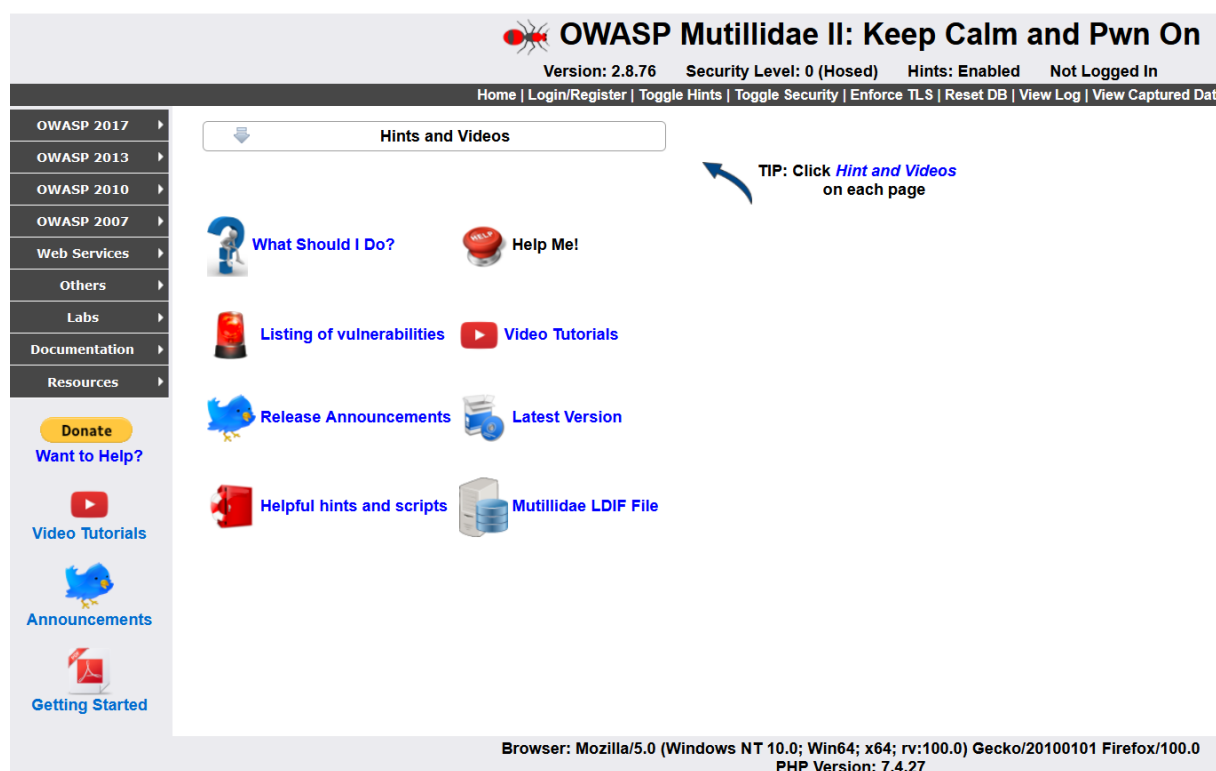
## 2 Le top 10 d'OWASP 2017

OWASP fournit une liste des risques de sécurité des applications web les plus courants. OWASP met à jour son classement afin de sensibiliser les développeurs web aux risques encourus. Les 10 risques classés par ordre de dangerosité en **2017**, sont les suivants :

RISQUES		DÉFINITIONS	ACTIVITÉS
A1	INJECTION	Correspond au risque d'injection SQL (SQLi).	1
A2	BROKEN AUTHENTICATION AND SESSION MANAGEMENT	Correspond au risque de casser la gestion de l'authentification et de la session. Comprend notamment le vol de session (session hijacking) ou la récupération de mots de passe.	2
A3	CROSS SITE SCRIPTING (XSS)	Correspond au XSS soit l'injection de contenu dans une page, ce qui provoque des actions non désirées sur une page Web. Les failles XSS sont particulièrement répandues parmi les failles de sécurités Web.	3
A4	INSECURE DATA OBJECT REFERENCE (IDOR)	Correspond aux failles de sécurité des identifiants (ID) de données visualisées. Nécessite de mettre en place un contrôle d'accès aux données.	4
A5	SECURITY MISCONFIGURATION	Correspond aux failles de configuration liés aux serveurs Web, applications, base de données ou framework.	5
A6	SENSITIVE DATA EXPOSURE	Correspond aux failles de sécurité liées aux données sensibles comme les mots de passe, les numéros de carte de crédit ou encore les données personnelles et la nécessité de crypter ces données.	6
A7	MISSING FUNCTION LEVEL ACCESS CONTROL	Correspond aux failles de sécurité liées aux accès non souhaitables à une fonctionnalité.	7
A8	CROSS SITE REQUEST FORGERY	Correspond aux failles liées à l'exécution de requêtes à l'insu de l'utilisateur.	8
A9	USING COMPONENT WITH KNOWN VULNERABILITIES	Correspond aux failles liées à l'utilisation de composants tiers.	9
A10	UNVALIDATED REDIRECTS AND FORWARDS	Correspond aux failles liées aux <i>redirect</i> et <i>forward</i> générique des applications.	10

### III MUTILLIDAE et son installation

Dans quelques missions, nous utiliserons Mutillidae, une plateforme de test des vulnérabilités web développée par OWASP.



Mutillidae est un site web conçu pour identifier et tester les failles de sécurité. Il est possible pour chacune d'entre elles, de définir le niveau de sécurité appliqué.

Ainsi, typiquement, notre démarche consistera, pour une faille de sécurité particulière :

- de mettre en évidence la faille à partir de la version non sécurisée de la page concernée
- de constater ensuite que dans la version sécurisée de cette page fournie par Mutillidae, l'attaque n'est plus possible.
- d'explorer des mécanismes de sécurisation utilisés (code de la page associée), pour dégager des bonnes pratiques de programmation.

#### 1 Installation de MUTILLIDAE

Concrètement, Mutillidae sera accessible depuis votre espace alwaysdata.

On commencera par se positionner dans le bon dossier dédié aux applications web :  
`/home/user/www/`

Ensuite, on récupère à partir de Github le code de l'application :  
`$ git clone https://github.com/webpwnized/mutillidae.git`

Un nouveau dossier sera créé et il sera accessible via l'URL suivante :  
`https://user.alwaysdata.net/mutillidae`

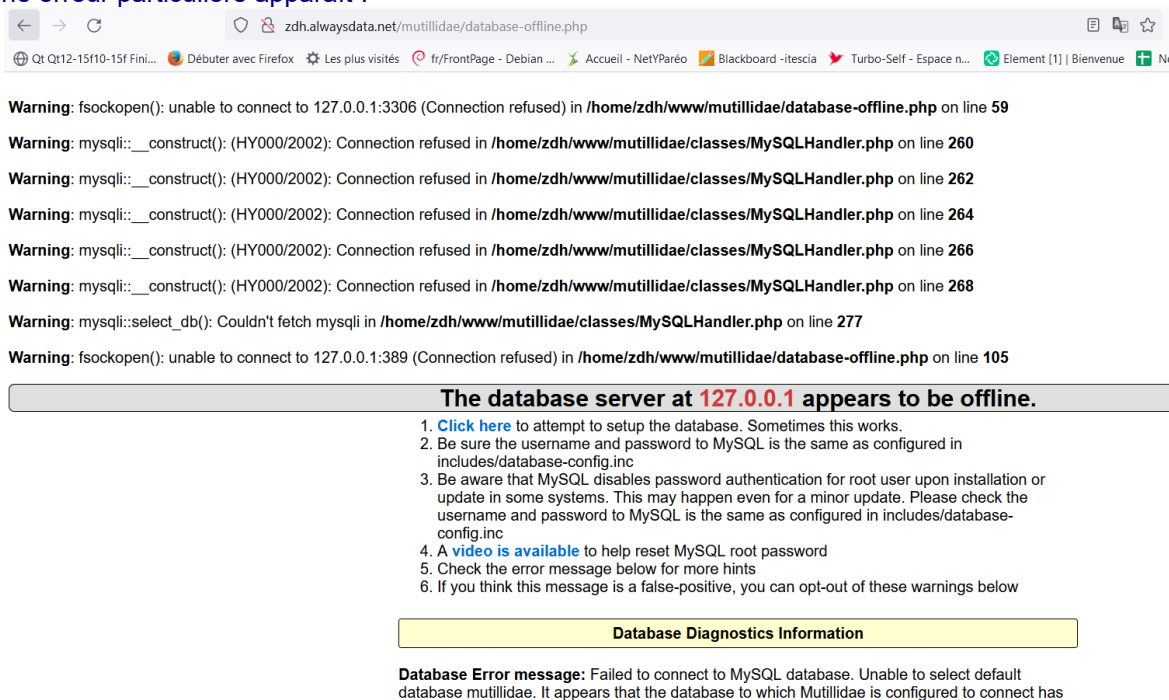
Une premier message bloquant est déclenchée :



En effet, on ne doit pas laisser sur le web accès à une application web aussi défaillante. OWASP a donc bloqué les accès des clients extérieurs (accès via internet). Ceci est possible grâce au fichier : `/home/user/www/mutillidae/.htaccess` qui permet de filtrer les adresses IP publiques.

Pour ouvrir les accès temporairement, et nous permettre d'installer l'application, nous allons renommer ce fichier : `/home/user/www/mutillidae/no.htaccess`

Une erreur particulière apparaît :



The screenshot shows a web browser window with the address bar displaying `zdh.alwaysdata.net/mutillidae/database-offline.php`. The page content displays several warning messages from PHP and MySQL, all indicating connection failures to a MySQL database at `127.0.0.1:3306`. A prominent yellow box with a red border contains the message: "The database server at 127.0.0.1 appears to be offline." Below this, a list of six troubleshooting steps is provided, including checking the database configuration, ensuring correct credentials, and verifying the MySQL service status. At the bottom, a yellow box titled "Database Diagnostics Information" contains the "Database Error message": "Failed to connect to MySQL database. Unable to select default database mutillidae. It appears that the database to which Mutillidae is configured to connect has".

La base de données est cherchée sur la même machine où est installé le serveur web Apache. Su Alwaysdata, les différentes applications serveurs ne sont pas nécessairement sur les mêmes machines.

On doit donc préparer une Base de données MySQL dédiée avec un utilisateur dédié ayant TOUS les droits et ainsi compléter correctement le fichier de configuration suivant : `/home/user/www/mutillidae/includes/database-config.inc`

Version Originale :

```
zdh@ssh1:~/www/mutillidae/includes$ cat database-config.inc
<?php
define('DB_HOST', '127.0.0.1');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', 'mutillidae');
define('DB_NAME', 'mutillidae');
define('DB_PORT', 3306);
?>
zdh@ssh1:~/www/mutillidae/includes$
```

On créera une nouvelle Base de données via l'interface d'Alwaysdata (menu MySQL) : **user\_mutillidae**

On ajoutera un nouvel utilisateur dédié à cette base de données (menu MySQL, Onglet utilisateurs) et ayant tous les droits : **user\_mutilliuser** et avec le mot de passe **mutillipwd**.

L'URL du serveur MySQL est visible sur cette même interface.

On rechargeant l'application mutillidae, une autre erreur peut apparaître :

Warning: fsockopen(): unable to connect to 127.0.0.1:389 (Connection refused) in /home/zdh/www/mutillidae/database-offline.php on line 105

**The database server at [mysql-zdh.alwaysdata.net](https://mysql-zdh.alwaysdata.net) appears to be offline.**

1. [Click here](#) to attempt to setup the database. Sometimes this works.
2. Be sure the username and password to MySQL is the same as configured in includes/database-config.inc
3. Be aware that MySQL disables password authentication for root user upon installation or update in some systems. This may happen even for a minor update. Please check the username and password to MySQL is the same as configured in includes/database-config.inc
4. A [video is available](#) to help reset MySQL root password
5. Check the error message below for more hints
6. If you think this message is a false-positive, you can opt-out of these warnings below

**Database Diagnostics Information**

**Database Error message:** Failed to connect to MySQL database. Failed to execute test query on blogs\_table in the MySQL database but we appear to be connected Table 'zdh\_mutillidae.blogs\_table' doesn't exist

First, try to reset the database (ResetDB button on menu)

The blogs table should exist in the zdh\_mutillidae database if the database configuration is correct. If the system made it this far, the username and password are probably correct. Perhaps the database name is wrong.

6 pistes sont énumérée, mais souvent la première suffit (en cliquant sur le lien proposée qui permet de reconstruire la BDD).

Désormais votre application est opérationnelle.

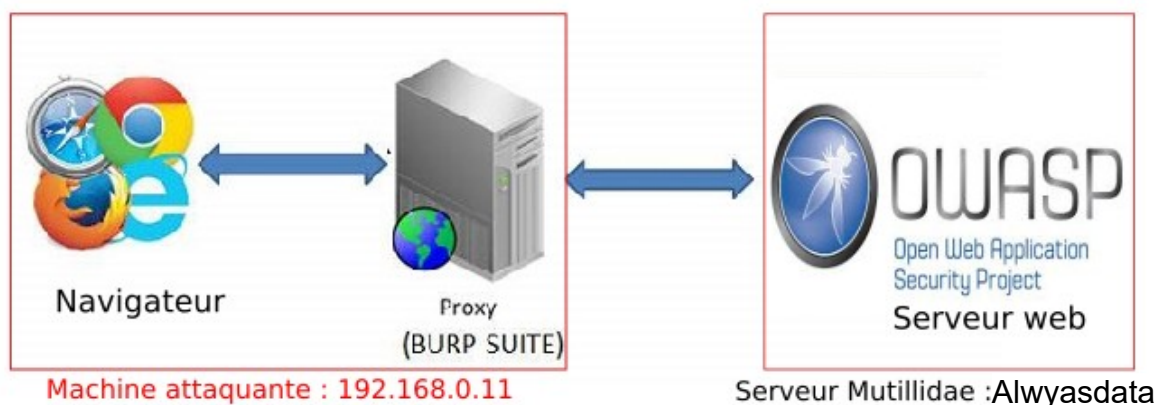
**N'oubliez pas de rebloquer votre application web en renommant le fichier no.htaccess en .htaccess.**

## 2 Installation de BurpSuite

Certaines activités auront aussi recours à Burpsuite (Community Edition).

C'est une plateforme qui permet d'effectuer des tests de sécurité sur les applications web. Elle joue le rôle d'un proxy qui se positionne entre le navigateur de l'attaquant et le serveur contenant l'application web à tester (Mutillidae pour nous). Elle capture les requêtes effectuées afin de pouvoir les analyser, les modifier et les rejouer en modifiant les paramètres.

Nous utiliserons Burpsuite pour obtenir des informations sur une page fournie par Mutillidae selon le schéma suivant :



Ainsi, dans une des activités, Burpsuite permet de filtrer les requêtes adressées à Mutillidae et de repérer le comportement spécifique de la page (non sécurisée) lorsque le login utilisé existe. Il est alors possible de tester une liste de login : Burpsuite est capable d'utiliser un fichier contenant ces login, de réaliser la même requête vers la page fournie par Mutillidae et de déterminer les login qui existent.

Dans notre cas, Burpsuite sera installée sur notre machine personnelle qui a déjà un navigateur web installé.



## IV Mise en garde juridique

Il convient de préciser que la loi interdit le fait d'accéder ou de se maintenir de manière frauduleuse dans un système de traitement automatisé de données (STAD) et que les organisations doivent garantir la confidentialité des données conservées.

Du point de vue de l'attaquant, on peut rappeler l'article de loi suivant :

*L'article 323-1 du code pénal, lequel dispose que « Le fait d'accéder ou de se maintenir, frauduleusement, dans tout ou partie d'un système de traitement automatisé de données est puni de deux ans d'emprisonnement et de 30000 euros d'amende. Lorsqu'il en est résulté soit la suppression ou la modification de données contenues dans le système, soit une altération du fonctionnement de ce système, la peine est de trois ans d'emprisonnement et de 45000 euros d'amende. »*

Du point de vue des organisations, un minimum de précautions est nécessaire. La CNIL peut sanctionner les entreprises trop laxistes en la matière.

On peut citer les articles de loi suivant :

Article 34 de la loi du 6 janvier 1978 : Le responsable du traitement est tenu de prendre toutes précautions utiles, au regard de la nature des données et des risques présentés par le traitement, pour préserver la sécurité des données et, notamment, empêcher qu'elles soient déformées, endommagées, ou que des tiers non autorisés y aient accès.

Article 226-17 : Le fait de procéder ou de faire procéder à un traitement de données à caractère personnel sans mettre en œuvre les mesures prescrites à l'article 34 de la loi n° 78-17 du 6 janvier 1978 précitée est puni de cinq ans d'emprisonnement et de 300 000 euros d'amende.



**Toutes les manipulations décrites sont réalisées uniquement sur la plateforme pédagogique présentée. Elles ne doivent en aucun cas être testées sur d'autres sites web.**