

Exploitation d'une plateforme d'apprentissage des vulnérabilités des applications web

Propriétés	Description
Intitulé long	Exploitation d'une plateforme d'apprentissage des vulnérabilités des applications web
Intitulé court	Sécurisation des applications web
Formation concernée	BTS Services Informatiques aux Organisations
Matière	SLAM4 : Réalisation et maintenance de composants logiciels.
Présentation	<p>Ce Côté labo a pour objectif d'exploiter la plateforme d'apprentissage Mutillidae (OWASP) afin de se familiariser avec les principales vulnérabilités des applications web.</p> <p>Chaque activité couvre une problématique spécifique (SQLi, XSS, CSRF...) en référence au top 10 des vulnérabilités décrites par l'OWASP.</p> <p>Dans un premier temps, l'étudiant doit réaliser les attaques associées à chaque vulnérabilité.</p> <p>Dans un deuxième temps, l'objectif est d'analyser et de comprendre les codes sources des scripts présentés dans leur forme non sécurisée puis sécurisée en tant que contre-mesure.</p> <p>Cette première livraison comporte :</p> <ul style="list-style-type: none"> • un document de présentation, • un document permettant de mettre en place l'environnement de test, • une première activité sur les injections, SQL notamment, et sa correction.
Notions	<p>Activités supports de l'acquisition des compétences</p> <p>D4.1 – Maintenance d'une solution applicative</p> <ul style="list-style-type: none"> • A4.2.1 Analyse et correction d'un dysfonctionnement, d'un problème de qualité de service ou de sécurité. <p>Savoir-faire</p> <ul style="list-style-type: none"> • Programmer un composant logiciel. • Adapter un composant logiciel. • Valider et documenter un composant logiciel. <p>Savoirs associés</p> <ul style="list-style-type: none"> • Techniques de sécurisation.
Prérequis	Commandes de base d'administration d'un système Linux, langages PHP et JavaScript.
Outils	<p>Deux machines éventuellement virtualisées sont nécessaires avec Linux comme système d'exploitation.</p> <p>Site officiel : https://www.owasp.org</p>
Mots-clés	OWASP, Mutillidae, BurpSuite, vulnérabilités, SQLi, XSS, IDOR.

Table des matières

I Introduction.....	2
1 La sécurité des applications web.....	2
1.1 Les applications web sont partout.....	2
1.2 La sécurisation des applications web est indispensable.....	2
2 Les motifs des attaques.....	3
II Présentation d'OWASP.....	3
1 La communauté OWASP.....	3
2 Le top 10 d'OWASP.....	4
III Démarche et organisation du côté labo.....	5
IV Mise en garde juridique.....	6

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 cybermenaces¹.

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

¹ <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

OWASP fournit une liste des risques de sécurité des applications web les plus courants. En 2017, OWASP a mis à jour son classement afin de sensibiliser les développeurs web aux risques encourus. Les 10 risques classés par ordre de dangerosité 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 Démarche et organisation du côté labo

⇒ Pré-requis

Concernant le point de départ de ce côté labo, deux scénarios sont possibles :

- **L'installation des outils est réalisée par le professeur**

Le **professeur** doit donc disposer de pré-requis concernant les commandes de base de l'administration d'un système Linux.

L'étudiant doit disposer de connaissances de bases sur les langages PHP et Javascript et sur l'architecture associée au développement d'une application Web en général (notion de serveur web, http, https...).

- **L'installation des outils est réalisée par les étudiants**

Cette seconde option est plus longue et demande à des étudiants SLAM de connaître les bases de l'administration d'un système LINUX en plus des fondamentaux liés à la programmation des applications web (PHP, Javascript...).

Cette première production présente la plateforme de test des vulnérabilités puis étudiée dans l'activité 1 la plus courante de toutes selon OWASP : l'injection SQL.

D'autres activités suivront selon le plan annoncé dans le tableau ci-dessus. Elles feront l'objet d'autres côtés labo.

Dans cette série de productions, nous utiliserons Mutillidae, plateforme de test des vulnérabilités web développée par Owasp.

OWASP Mutillidae II: Keep Calm and Pwn On

Version: 2.6.62 Security Level: 0 (Hosed) Hints: Enabled (1 - Try easier) Not Logged In

Home Login/Register Toggle Hints Show Popup Hints Toggle Security Enforce SSL Reset DB View Log View Captured Data

OWASP 2017
OWASP 2013
OWASP 2010
OWASP 2007
Web Services
HTML 5
Others
Documentation
Resources

PayPal - The safer, easier way to pay online!
[Want to Help?](#)

Hints and Videos

TIP: Click *Hint and Videos* on each page

[What Should I Do?](#)

[Video Tutorials](#)

[Help Me!](#)

[Listing of vulnerabilities](#)

[Bug Tracker](#)

[Bug Report Email Address](#)

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 :

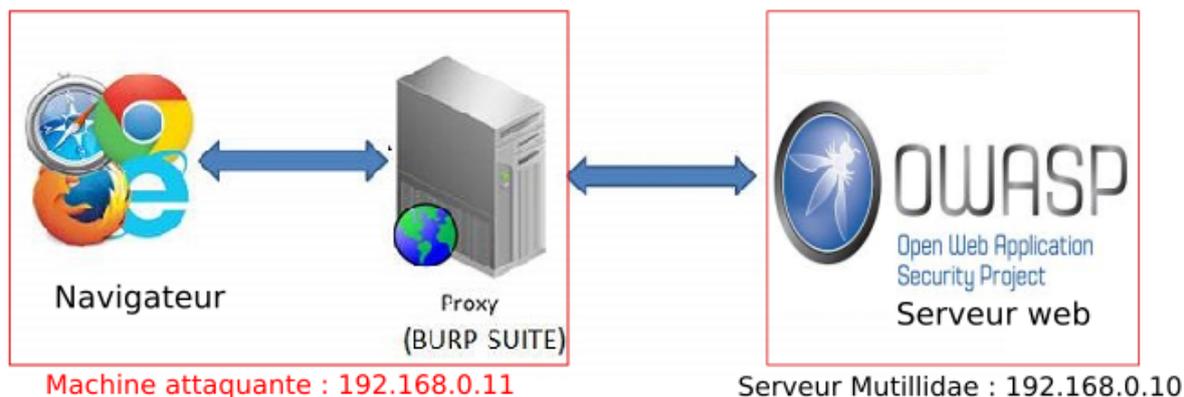
- à partir de la version non sécurisée de la page concernée et à mettre en évidence la faille de sécurité. Il s'agira de réaliser une injection SQL par exemple, dans l'activité 1.
- nous constaterons ensuite que dans la version sécurisée de cette page fournie par Mutillidae, l'attaque n'est plus possible.
- l'étude des mécanismes de sécurisation utilisés, donc du code de la page associée, permettra de dégager des bonnes pratiques de programmation.

Concrètement, Mutillidae sera accessible depuis une machine virtuelle particulière. Son installation est décrite dans le document `owasp-mise_en_place.odt`. Le professeur pourra fournir la machine virtuelle à ses étudiants ou choisir de leur faire installer.

Certaines activités auront aussi recours à Burpsuite.

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 l'activité 2, 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 une deuxième machine virtuelle qui pourra aussi accueillir le navigateur web.

Cet outil ne sera pas utilisé dans l'activité 1 mais son installation est quand même décrite dans le document `owasp-mise_en_place.odt`.

Piste d'exploitation pédagogique : cette production peut aussi être exploitée dans le cadre de PPE3-4 en SLAM. Il serait possible de partir d'une application faiblement sécurisée et de demander aux étudiants de renforcer sa sécurité en indiquant des objectifs précis.

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.