RÉPUBLIQUE DU BÉNIN
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ D'ABOMEY-CALAVI

INSTITUT DE FORMATION ET DE RECHERCHE EN INFORMATIQUE

BP 526 Cotonou Tel : +229 21 14 19 88
https://www.ifri-uac.bj Courriel : contact@ifri.uac.bj

MÉMOIRE

pour l'obtention du

Diplôme de Licence en Informatique

Option : Sécurité Informatique

Présenté par :
Ulrich Girès ASSOGBA

Test d'intrusion web : plateforme de demande en ligne des allocations d'études universitaires

Sous la supervision :
Ing. Victor **OYETOLA**
Chef Service de la Promotion des Technologies de l'Information et de la Communication à l'Université d'Abomey-Calavi

Membres du jury :

SAHO Nelson	Docteur	Ministère des Affaires Etrangères	Président
GOUDALO Jospy K.	Ingénieur	CAGD	Examinateur
OYETOLA Victor	Ingénieur	IFRI-UAC	Rapporteur

Année Académique : **2023 - 2024**

Sommaire

Dédicace ii

Remerciements iii

Résumé iv

Abstract v

Liste des figures vi

Liste des tableaux vii

Liste des acronymes viii

Glossaire 1

Introduction 2

1 Revue de littérature 4

2 Détection des vulnérabilités sur la plateforme. 12

3 Résultats et recommandations 24

Conclusion 29

Bibliographie 30

Webographie 31

Table des matières 32

Dédicace

Je dédie ce mémoire à ma famille, qui a toujours été un pilier dans mon parcours académique et personnel. À mes enseignants et encadreurs, dont les conseils et les savoirs partagés ont été des guides précieux.

Remerciements

Nous exprimons notre profonde gratitude à l'Éternel, Dieu Créateur, pour son amour infini, pour l'inspiration et la protection dont Il nous comble chaque jour. À travers ce mémoire, nous tenons à remercier :

- Tout d'abord le corps professoral et administratif de l'Institut de Formation et de Recherche en Informatique de l'Université d'Abomey-Calavi (IFRI-UAC), le Professeur Eugène C. EZIN pour la qualité de leurs enseignements et l'effort pour assurer aux étudiants une formation qualitative.

- Ensuite, mes sincères remerciements à mon maître de mémoire l'Ingénieur Victor OYETOLA, pour m'avoir accordé son temps et pour m'avoir suivi durant la réalisation de ce travail.

Et enfin mes mots de remerciements vont à l'endroit de toute personne ayant contribuer d'une quelconque manière à la réalisation de ce travail.

Résumé

La plateforme de demande en ligne des allocations universitaires est une application web accessible via un navigateur. Elle offre une interface numérique permettant aux bénéficiaires de bourses et d'aides universitaires de soumettre leurs demandes et d'en suivre l'évolution entièrement en ligne. La sécurisation de cette plateforme constitue un enjeu majeur, face à la multiplication des attaques informatiques visant à compromettre sa sécurité, ce qui suscite une inquiétude croissante. Aujourd'hui, la protection des applications web est devenue une priorité stratégique, aussi essentielle que leur performance ou l'expérience utilisateur. Les utilisateurs accordent d'ailleurs une importance croissante à la confidentialité et à la protection de leurs données.Afin de garantir une sécurité optimale, il est indispensable de recourir à des ressources comme celles proposées par l'OWASP, ainsi qu'à des outils d'analyse de vulnérabilités tels que OWASP ZAP, Nikto et SQLMap. Ce travail a pour objectif de mener un test d'intrusion approfondi sur la plateforme de demande en ligne des allocations universitaires, afin d'identifier ses vulnérabilités, d'évaluer leur impact potentiel et de formuler des recommandations concrètes pour renforcer sa protection contre les menaces.

Mots clés : Pentest, Vulnérabilités, OWASP, Cybersécurité.

Abstract

The online university grant application platform is a web application accessible via a browser. It provides a digital interface that allows scholarship and financial aid recipients to submit their applications and track their progress entirely online. Securing this platform is a major challenge, given the increasing number of cyberattacks aimed at compromising its security, which raises growing concerns. Today, web application security has become a strategic priority, as essential as performance or user experience. Moreover, users are increasingly concerned about the confidentiality and protection of their data. To ensure optimal security, it is essential to rely on resources such as those provided by OWASP, as well as vulnerability analysis tools like OWASP ZAP, Nikto, and SQLMap. The objective of this work is to conduct an in-depth penetration test on the online university grant application platform to identify its vulnerabilities, assess their potential impact, and provide concrete recommendations to strengthen its protection against threats.

Key words: Pentest, Vulnerabilities, OWASP, Cybersecurity.

Liste des figures

2.1 Page d'accueil du site web de la DBAU . . . 13
2.2 Page pour une nouvelle demande . . . 14
2.3 Page pour suivre les demandes . . . 14
2.4 Lancement de nmap . . . 15
2.5 Lancement de whatweb . . . 15
2.6 Lancement de gobuster . . . 17
2.7 Résultats avec gobuster . . . 17
2.8 Lancement de TheHarvester . . . 18
2.9 Resultat de TheHarvester . . . 19
2.10 Lancement de nikto . . . 19
2.11 Scan de vulnérabilités web présentes sur le serveur . . . 20
2.12 Scan de vulnérabilités web présentes sur le serveur . . . 20
2.13 Test d'intrusion avec OWASP ZAP . . . 21
2.14 Alerte 1(**priorité moyenne**) . . . 21
2.15 Alerte 2 (**priorité faible**) . . . 21
2.16 Alerte 3 (**pour information**) . . . 22

3.1 Sécurisation du CSP . . . 25
3.2 Sécurisation des coockies . . . 27
3.3 Sécurisation des informations du serveur . . . 28

Liste des tableaux

1.1 Tableau de comparaison des types d'intrusions. 6

2.1 Les ports ouverts 15
2.2 Vulnérabilités détectées et outils utilisés. 23

3.1 Vulnérabilités détectées et outils utilisés 24

Liste des acronymes

API :

Application Programming Interface *:*

CSP :

Content Security Policy *:*

CSRF :

Cross-Site Request Forgery *:*

DNS :

Domain Name System *:*

FTP :

File Transfer Protocol *:*

HSTP :

HTTP Strict Transport Security *:*

HTTP :

Hypertext Transfer Protocol *:*

HTTPS :

Hypertext Transfer Protocol Secure *:*

IP :

Internet Protocol *:*

Nginx :

Serveur web open-source et reverse proxy *:*

Nmap :

Network Mapper *:*

OWASP :

Open Web Application Security Project *:*

PHP :

Hypertext Preprocessor *:* Vuln[Pleaseinsert\PrerenderUnicode{Ãľ}intopreamble]rabilit[Pleaseinsert\Prere

SQL :

Structured Query Language *:* Pentest (Test d[Pleaseinsert\PrerenderUnicode{âĂŹ}intopreamble]intrusion)

SSH :

Secure Shell *:* Audit de s[Pleaseinsert\PrerenderUnicode{Ãľ}intopreamble]curit[Pleaseinsert\PrerenderUn

SSL :

Secure Sockets Layer *:* Plateforme web

TLS :

Transport Layer Security *:* Normes de s[Pleaseinsert\PrerenderUnicode{Ãľ}intopreamble]curit[Pleaseinsert

XSS :

Cross-Site Scripting *:*

Glossaire

Audit de sécurité : Processus visant à évaluer la sécurité d'un système d'information en identifiant les vulnérabilités, en évaluant les risques associés et en proposant des mesures correctives pour renforcer la protection des données et des infrastructures [16]. ix

Normes de sécurité : Ensembles de directives et de bonnes pratiques établies pour assurer la protection des systèmes d'information, des réseaux et des données contre les menaces potentielles. Ces normes, telles que l'ISO 27001, fournissent un cadre pour la gestion de la sécurité de l'information. ix

Pentest (Test d'intrusion) : Simulation contrôlée d'une attaque sur un système informatique ou une application, réalisée pour identifier et exploiter les vulnérabilités, afin d'évaluer la robustesse des mesures de sécurité en place [14]. ix

Plateforme web : Application accessible via un navigateur internet, conçue pour fournir des services en ligne aux utilisateurs, tels que des systèmes de gestion, des sites de commerce électronique ou des réseaux sociaux. ix

Vulnérabilité : Faiblesse ou faille dans un système informatique, une application ou un réseau, pouvant être exploitée par des attaquants pour compromettre la confidentialité, l'intégrité ou la disponibilité des informations [2]. ix

Introduction générale

Avec la digitalisation croissante des services, les applications web jouent un rôle central dans la simplification des processus et l'amélioration de l'accessibilité. Cependant, cette dépendance accrue au numérique s'accompagne de risques majeurs en matière de sécurité, notamment face à des cybermenaces de plus en plus sophistiquées. Pour garantir la protection des données et la continuité des services, il est essentiel de mener des tests de sécurité rigoureuses.

1. Contexte et justification

Les applications web occupent une place primordiale dans notre vie quotidienne en raison de leur accessibilité universelle via un simple navigateur. Cette facilité d'accès, que ce soit au travail, à domicile, représente un atout majeur. Toutefois, cette accessibilité constante augmente leur vulnérabilité aux cyberattaques. Ces dernières, menées par des acteurs malveillants ou des groupes organisés, cherchent souvent à exploiter des failles pour voler, altérer ou supprimer des informations sensibles. Les répercussions peuvent être considérables, allant de la perte de confiance des utilisateurs à de graves atteintes à la réputation des entités concernées. Dans ce contexte, la plateforme de demande en ligne des allocations des études universitaires, qui permet de soumettre des requêtes pour des allocations universitaires, représente une avancée significative dans la dématérialisation des services publics. Cependant, cette numérisation expose également la plateforme à des risques accrus en matière de sécurité, pouvant affecter la confidentialité, l'intégrité et la disponibilité des informations.Il devient donc crucial de réaliser un test de sécurité afin d'identifier les vulnérabilités potentielles et de mettre en œuvre des actions correctives appropriées. L'objectif de ce test est de protéger la plateforme contre les menaces informatiques en renforçant la sécurité des données. Il contribuera également à maintenir la confiance des utilisateurs tout en assurant la pérennité et la fiabilité des services fournis par cette plateforme.

2. Problématique

La plateforme de demande en ligne des allocations d'études universitaires, utilisée par les étudiants des universités publiques du Bénin (Université d'Abomey-Calavi, Université de Parakou, etc.), joue un rôle crucial en facilitant leurs démarches administratives pour l'obtention d'allocations. Afin d'assurer une expérience fluide et sécurisée à l'ensemble des utilisateurs, il est impératif de garantir la sécurité de cette plateforme. La protection des données personnelles des étudiants, comprenant des informations sensibles, constitue une priorité absolue. Le respect des normes de sécurité est donc essentiel pour préserver la confidentialité, l'intégrité, et la disponibilité des données. Dans cette optique, la réalisation d'un test de sécurité s'impose comme une étape stratégique incontournable. Ce mémoire vise à effectuer un test d'intrusion approfondi de la plateforme, permettant d'identifier les vulnérabilités existantes et de fournir les bases nécessaires à sa sécurisation future.

3. Objectifs

Le but principal de ce projet est d'évaluer la sécurité de la plateforme en ligne de demande des d'études universitaires à travers un test d'intrusion, afin d'identifier les vulnérabilités critiques, d'évaluer leur impact et de proposer des mesures concrètes pour améliorer la protection des données sensibles et assurer la résilience de la plateforme face aux menaces. Plus spécifiquement il s'agira :

- d'identifier les vulnérabilités critiques de la plateforme à l'aide d'outils de test de sécurité spécifiques ;
- d'évaluer l'impact des vulnérabilités retenues ;
- de suggérer des recommandations pour corriger les vulnérabilités retenues ;

4. Organisation du document

Notre travail se structure en trois chapitres. Le premier chapitre expose les concepts fondamentaux nécessaires à la réalisation d'une mission de notre test d'intrusion. Le second chapitre est consacré à l'identification des vulnérabilités présentes sur la plateforme de demande en ligne des allocations d'études universitaires. Enfin, le troisième chapitre présente le rapport du test, comprenant la liste des vulnérabilités détectées, leur impact ainsi que les recommandations pour y remédier.

Chapitre 1

Revue de littérature

Introduction

La sécurité est un élément clé de la qualité d'une application web. Effectuer un test d'intrusion constitue une étape importante pour garantir sa protection. Pour mener à bien un test de sécurité d'une plateforme web, il est indispensable de maîtriser les connaissances fondamentales dans ce domaine. Ce chapitre présente les concepts indispensables à connaître avant d'entreprendre toute test de sécurité d'une plateforme.

1.1 Fondamentaux de la sécurité Informatique

La sécurité informatique a pour but de protéger les systèmes d'information contre toute forme de menace. Pour ce faire, il existe plusieurs mesures préventives et techniques. L'un des aspects fondamentaux de la sécurité informatique est la gestion des vulnérabilités, c'est-à-dire des faiblesses qui peuvent être exploitées par des attaquants pour compromettre la plateforme. La sécurité informatique repose sur trois grands principes, souvent appelés la triade CID :

- **Confidentialité** : Assurer que seules les personnes autorisées aient accès aux données sensibles.
- **Intégrité** : Garantir que les données ne soient pas modifiées de manière non autorisée.
- **Disponibilité** : Assurer que les services sont disponibles et accessibles aux utilisateurs légitimes.

1.2 Les différents types de hackers

Un hacker est un expert en informatique doté de compétences avancées en sécurité des systèmes et des réseaux. Son expertise lui permet d'identifier et de contourner des failles de sécurité. Le pentester, quant à lui, a la possibilité de détecter un maximum de vulnérabilités afin d'améliorer la protection des systèmes. Certains hackers agissent pour des raisons éthiques et militantes, comme les hacktivistes, qui défendent la liberté d'expression et l'accès à l'information tout en respectant une charte

éthique. Cependant, si certains hackers mettent leurs compétences au service de la légalité, d'autres les utilisent à des fins illicites [3].

1.2.1 Les black hats

Ce sont des hackers qui violent la sécurité informatique, souvent sans raison valable, au-delà de leur intention malveillante ou de leur désir de gain personnel. Un hacker de type "black hat" agit avec de mauvaises intentions. Il exploite une faille dans le système de sécurité d'un site, par exemple, pour pénétrer dans sa base de données, récupérer toutes les informations exploitables et les utiliser à des fins personnelles ou les revendre sur le marché noir(le dark web) [7].

1.2.2 Les white hats

Les "white hats" sont des hackers éthiques qui testent la sécurité des systèmes d'information en adoptant l'attitude d'un hacker malveillant. Tout comme les "black hats", ils cherchent à exploiter les vulnérabilités, mais la différence majeure réside dans leur intention. Un white hat n'utilise pas les données sensibles à des fins nuisibles. Au contraire, il avertit les propriétaires du site de la brèche afin qu'ils puissent corriger le problème. Leur objectif est de mener des actions bénéfiques. On peut les considérer comme des experts en cybersécurité ou des cyber-enquêteurs, toujours à la recherche de failles. De plus en plus d'entreprises font appel à leurs services pour prévenir les attaques malveillantes et renforcer leur sécurité.

1.2.3 Les grey hats

Les hackers dits "grey hats" se situent entre les "white hats" et les "black hats". Lorsqu'un hacker se revendique de cette catégorie, c'est pour marquer une distinction par rapport à ces deux groupes principaux, l'un étant bienveillant et l'autre malveillant. Par exemple, un hacker "grey hat" pourrait s'introduire illégalement dans un site en exploitant une vulnérabilité de son système de sécurité, puis fournir au propriétaire un correctif pour lui permettre de sécuriser la faille par laquelle il est entré, sans compromettre l'intégrité du site [16].

1.3 Les différents types de tests d'intrusion

Le test d'intrusion est une approche technique visant à évaluer la sécurité d'un système ou d'un réseau informatique. Cette méthode repose sur la simulation d'attaques réalisées par un acteur malveillant ou un logiciel nuisible. Les professionnels spécialisés dans ce domaine, appelés pentesters, utilisent à la fois des outils automatisés disponibles sur le marché et des scripts ou programmes développés par leurs soins. L'objectif principal d'un test d'intrusion est d'identifier et d'exploiter les vulnérabilités pour simuler une attaque. Dans le cadre de ce mémoire, l'analyse se concentre sur la plateforme de demande en ligne des allocations universitaires. Les tests d'intrusion se divisent en trois grandes catégories : le test boîte blanche, le test boîte noire et le test boîte grise.

1.3.1 Les tests d'intrusion boîte blanche

Un test d'intrusion en boîte blanche est une simulation d'attaque où le pentester dispose d'un accès total au système cible (codes sources, architecture, identifiants). Réalisé avec les équipes techniques,

il vise à détecter un maximum de vulnérabilités en utilisant des outils et stratégies avancés[6].

1.3.2 Les tests d'intrusion boîte noire

Un test d'intrusion en boîte noire simule une attaque externe sans aucune connaissance préalable du système cible. Le pentester doit collecter des informations et exploiter des vulnérabilités, imitant des cyberattaques réelles dans un cadre défini pour éviter tout dommage.

1.3.3 Les tests d'intrusion boîte grise

Un test d'intrusion en boîte grise est une simulation d'attaque où le pentester dispose d'un accès partiel à des informations (documents d'architecture, inventaires réseau). Il équilibre réalisme et efficacité, ciblant des zones spécifiques tout en restant moins exhaustif qu'un test en boîte blanche.

1.4 Comparaison des tests d'intrusion et outils utilisés

1.4.1 Avantages et inconvénients des tests d'intrusion

TABLE 1.1 : Tableau de comparaison des types d'intrusions.

Type d'intrusion	Avantages	Inconvenient
Les tests d'intrusion boîte blanche	Le pentester a la possibilité de détecter un maximum de failles de sécurité.	Les seules limites ici sont les compétences du pentester choisi et la perpétuelle chasse aux nouvelles techniques de hacking qui évoluent chaque jour.
Les tests d'intrusion boîte noire	Cette méthode permet donc de montrer au client quels types d'informations le hacker serait capable d'obtenir et ainsi de mettre en avant les risques encourus dans le cas d'une attaque.	Ils sont plus longs. Le pentester peut ne pas arriver à s'introduire dans le système, ce qui ne prouve pas qu'un hacker n'y arriverait pas.
Les tests d'intrusion boîte grise	Le pentester peut facilement simuler des attaques et aller au-delà de ce qu'il aurait pu faire en mode boîte noire.	Le pentester ne peut pas garantir que le hacker n'aura pas trouver une nouvelle technique de piratage et ainsi trouver une nouvelle faille exploitable.

Le tableau ci dessus offre une vision claire des avantages et inconvénients de chaque type de test d'intrusion. Il met en évidence que le choix de la méthode dépend des objectifs de l'évaluation et des contraintes du client. Par exemple, la boîte blanche est idéale pour une analyse approfondie, tandis que la boîte noire est plus adaptée pour simuler des attaques réalistes. La boîte grise, quant à elle, offre un compromis intéressant entre les deux. Ce tableau est donc un outil précieux pour aider les professionnels à choisir la méthode de test la plus adaptée à leurs besoins.

1.4.2 Les outils de pentest web

Autrefois, le piratage informatique était complexe et nécessitait des compétences manuelles poussées. Aujourd'hui, les attaquants ont accès à des ensembles d'outils automatisés et à des capacités de traitement avancées, leur permettant de mener des attaques bien plus élaborées. De manière similaire, les pentesters disposent désormais d'outils plus performants, rapides et efficaces pour réaliser leurs tests de pénétration.

1. **Kali Linux** est une distribution GNU/Linux dédiée à la sécurité informatique, incluant de nombreux outils de hacking. Bien qu'elle puisse être utilisée pour des activités malveillantes telles que le vol, la destruction ou la corruption de documents électroniques sur un ordinateur cible, elle peut aussi être employée de manière éthique, dans le cadre du piratage éthique. Ce dernier consiste à tester et identifier les vulnérabilités d'un système afin de les corriger et renforcer sa sécurité. En tant que système de test par excellence, Kali est livré avec une large gamme d'outils, ce qui en fait le choix par défaut pour de nombreux tests de pénétration.

2. **Nmap**, abréviation de Network Mapper, est un outil gratuit et open source utilisé pour l'analyse des vulnérabilités et la découverte de réseaux. Les administrateurs réseau s'en servent pour identifier les appareils connectés à leurs systèmes, découvrir les hôtes disponibles, connaître les services offerts, repérer les ports ouverts et évaluer les risques de sécurité.

3. **Metasploit** est un projet axé sur la sécurité des systèmes informatiques, visant à fournir des informations sur les vulnérabilités des systèmes, à faciliter les tests de pénétration et à développer des signatures pour les systèmes de détection d'intrusion [5].

4. **Gobuster** est un outil de reconnaissance et d'énumération de répertoires et de fichiers sur un serveur web. Il est souvent utilisé dans le cadre de tests d'intrusion et de sécurité pour identifier les ressources accessibles sur un site web ou un serveur. Gobuster fonctionne en effectuant des requêtes HTTP (ou HTTPS) et en analysant les réponses pour déterminer si un répertoire ou un fichier existe.

5. **OWASP ZAP** est un outil gratuit développé par la communauté OWASP, spécialisé dans l'analyse de la sécurité des sites web. C'est l'outil de sécurité web le plus utilisé au monde. Basé sur Java, il propose une interface graphique intuitive et des fonctionnalités puissantes pour le fuzzing, le scripting, le spidering, le proxy et l'attaque des applications web. En tant qu'outil tout-en-un, OWASP ZAP permet d'automatiser l'analyse des vulnérabilités des applications web [4].

6. **Nikto** est un outil en ligne de commande, open source et écrit en Perl, conçu pour scanner la sécurité des serveurs web. Il analyse rapidement un serveur, un hôte virtuel ou un site web afin d'identifier des failles de sécurité potentielles et des erreurs de configuration connues. Grâce à sa capacité à effectuer un grand nombre de tests, il est incontournable pour les professionnels de la cybersécurité et les administrateurs système.

7. **Whatweb** est un outil open-source conçu pour identifier les technologies utilisées par un site web. Il reconnaît notamment les systèmes de gestion de contenu (CMS), les plateformes de blogs, les packages statistiques ou analytiques, les bibliothèques JavaScript, les serveurs web et les appareils intégrés. WhatWeb dispose de plus de 1 800 plugins, chacun étant conçu pour détecter une technologie spécifique.

8. **Theharvester** est un outil open-source de reconnaissance utilisé pour la collecte d'informations sur des entreprises, des domaines ou des adresses e-mail à partir de sources publiques (OSINT - Open Source Intelligence). Il est principalement utilisé en cybersécurité pour effectuer des tests de sécurité et de la reconnaissance en phase de test d'intrusion. Développé en Python, TheHarvester permet de récupérer des informations précieuses en interrogeant divers moteurs de recherche, bases de données publiques et services en ligne.

9. **SQLMap** est un outil open source spécialisé dans l'automatisation de la détection et de l'exploitation des failles d'injection SQL. Il simplifie l'analyse des bases de données pour identifier et exploiter les vulnérabilités, facilitant ainsi les tests de sécurité des systèmes utilisant des bases de données.

1.5 Les vulnérabilités web selon OWASP

L'Open Web Application Security Project (OWASP) est une organisation mondiale à but non lucratif dédiée à l'amélioration de la sécurité des applications web. Elle propose de nombreuses ressources gratuites pour aider les organisations à sécuriser leurs services web, notamment le célèbre projet OWASP Top 10, qui recense les dix principaux risques de sécurité affectant les applications web. L'objectif d'OWASP est de sensibiliser les individus et les entreprises aux menaces liées à la sécurité des systèmes d'information, en leur fournissant des outils et des guides pratiques [1].

Le Top 10 de l'OWASP classe les attaques selon leur niveau de gravité. Les vulnérabilités répertoriées dans ce classement se retrouvent dans les applications web, quels que soient les langages de programmation ou les frameworks utilisés. Aucun langage de programmation n'est totalement épargné par les failles de sécurité.

1.5.1 Violation du contrôle d'accès

La violation du contrôle d'accès peut être défini comme une mauvaise application des restrictions sur ce que les utilisateurs authentifiés sont autorisés à faire. Il permet aux attaquants d'obtenir un accès non autorisé aux fonctionnalités ou aux données d'une application web. Les attaquants peuvent accéder, modifier ou supprimer des données sensibles, effectuer des actions non autorisées ou élever leurs privilèges.

1.5.2 Les défaillances cryptographiques

Les défaillances cryptographiques se produisent lorsque les processus de cryptage et de décryptage ne sont pas mis en œuvre correctement. Des données sensibles peuvent ainsi être exposées ou manipulées par des pirates. Il existe plusieurs types de défaillances cryptographiques : algorithmes de chiffrement faibles,gestion non sécurisée des clés,mauvaise génération de nombres aléatoires [9].

1.5.3 Injection

Les vulnérabilités par injection font référence à un type de vulnérabilité où un attaquant peut injecter un code malveillant dans les champs d'entrée ou les paramètres d'une application web. Cette

vulnérabilité est due à une mauvaise validation des entrées dans le code de l'application. Les attaquants peuvent exploiter ces vulnérabilités en injectant des commandes SQL, des commandes shell ou d'autres types de codes malveillants dans les champs de saisie [13].

1.5.4 Conception non sécurisée

La conception non sécurisée désigne la pratique consistant à créer des logiciels ou des systèmes sans prendre en compte les implications en matière de sécurité dès le départ. Cela peut conduire à des vulnérabilités qui peuvent être exploitées par des attaquants, leur permettant d'obtenir un accès non autorisé ou de causer des dommages [10].

1.5.5 Mauvaise configuration de sécurité

Ce risque survient lorsqu'une application ou un système n'est pas configuré correctement, ce qui le rend vulnérable aux attaques. Cela peut se produire pour diverses raisons, comme l'utilisation de configurations par défaut sans modification ou l'utilisation de versions de logiciels obsolètes. Les paramètres de sécurité désignent les options de configuration disponibles dans une application ou un système qui affectent son niveau de sécurité global. Ces paramètres comprennent les contrôles d'accès, les protocoles de cryptage, les politiques de mot de passe, les configurations de réseau, etc.

1.5.6 Composants vulnérables et obsolètes

Les composants vulnérables et obsolètes sont des bibliothèques ou des cadres tiers utilisés dans les applications web qui présentent des vulnérabilités connues ou qui ne sont plus pris en charge par leurs développeurs. Ces composants peuvent être exploités par des attaquants pour obtenir un accès non autorisé à des données sensibles ou prendre le contrôle du système [11].

1.5.7 Défauts d'identification et d'authentification

Vulnérabilités liées aux processus d'identification et d'authentification des utilisateurs. Ces vulnérabilités permettent aux attaquants de contourner les mécanismes d'authentification ou de voler les informations d'identification des utilisateurs. Parmi les exemples d'échecs d'identification et d'authentification, on peut citer les mots de passe faibles ou les politiques de mots de passe, l'absence d'authentification multifactorielle, les attaques par détournement de session, les attaques par force brute sur les pages de connexion ou les API qui n'ont pas mis en place de contrôles de limitation de débit.

1.5.8 Défauts d'intégrité des logiciels et des données

: Les défauts d'intégrité des logiciels et des données se produisent lorsqu'un pirate peut modifier ou supprimer des données de manière non autorisée. Cela peut être dû à des vulnérabilités dans le logiciel ou à de mauvaises pratiques de codage. Les attaquants peuvent exploiter ces vulnérabilités pour accéder à des informations sensibles ou endommager le système [12].

1.5.9 Défaillances de la journalisation et de la surveillance de la sécurité

La journalisation de sécurité consiste à enregistrer les événements qui se produisent dans une application ou un système. Il s'agit de l'activité de l'utilisateur, des événements du système, des erreurs

et d'autres informations pertinentes. La surveillance consiste à analyser ces journaux afin de détecter les menaces ou les problèmes de sécurité potentiels [8].

1.5.10 Falsification des requêtes côté serveur

Une faille SSRF se produit lorsqu'une application web récupère une ressource distante sans valider l'URL fournie par l'utilisateur. Elle permet à un attaquant de contraindre l'application à envoyer une requête élaborée à une destination inattendue, même si elle est protégée par un pare-feu, un VPN ou un autre type de liste de contrôle d'accès au réseau (ACL).

1.6 Référentiel

Afin de garantir une évaluation rigoureuse et conforme aux standards en vigueur, plusieurs référentiels de sécurité ont été utilisés dans le cadre de cette démarche. Parmi ceux-ci :

- **OWASP Testing Guide** : Bien que ce ne soit pas une norme ISO, le guide de test de l'OWASP (Open Web Application Security Project) est une référence mondiale pour les tests de sécurité des applications web. Il propose des méthodologies détaillées pour identifier et exploiter les vulnérabilités courantes, telles que les injections SQL, les failles XSS et les problèmes de gestion des sessions. Ce guide a été utilisé pour structurer les tests d'intrusion et détecter les failles spécifiques aux applications web.
- **Politique de Sécurité des Systèmes d'Information de l'État (PSSIE)** : La PSSIE est une politique qui établit les règles et bonnes pratiques pour la sécurité des systèmes d'information au sein des administrations publiques. Elle vise à garantir un niveau de sécurité élevé et homogène pour les systèmes d'information de l'État. La conformité à la PSSIE a permis d'aligner les pratiques de sécurité de la plateforme sur les exigences nationales, en mettant en œuvre des mesures de sécurité adaptées et en réalisant des tests réguliers.

1.6.1 Conformité aux référentiels

Pour respecter les exigences de la PSSIE, et des bonnes pratiques de l'OWASP, les étapes suivantes ont été suivies :

- **Évaluation des risques** : Identification des menaces et vulnérabilités potentielles.
- **Mise en œuvre des contrôles** : Application de mesures de sécurité techniques.
- **Amélioration** : Réalisation des tests réguliers pour vérifier la conformité et identifier les axes d'amélioration.
- **Documentation** : Maintien d'une documentation complète sur les mesures de sécurité.

1.7 Formulation du problème théorique

Pour réaliser le test d'intrusion de la plateforme web, nous avons opté pour la méthode technique du test d'intrusion en boîte noire. Dans ce cadre, la seule information mise à disposition est l'URL

d'accès de la plateforme de demande en lignes des allocations d'études universitaires.

1.8 Présentation des hypothèses explicatives

Le choix du test d'intrusion en boîte noire permet de simuler le comportement d'un attaquant externe n'ayant aucune connaissance préalable sur la plateforme de demande en ligne des allocations universitaires. Cette approche vise à identifier les vulnérabilités exploitables en se basant uniquement sur les informations accessibles publiquement.

Conclusion

Ce chapitre a permis de poser les bases théoriques nécessaires pour comprendre les concepts liés au test d'intrusion d'une plateforme web.Il prépare le terrain pour l'analyse pratique des vulnérabilités présentée dans le chapitre suivant.

Chapitre 2

Détection des vulnérabilités sur la plateforme.

Introduction

Le test d'intrusion d'une plateforme web repose sur l'identification des failles qui pourraient être exploitées par des attaquants. Le test d'intrusion répertorie les points forts, et surtout les points faibles (vulnérabilités) de tout ou partie du système. Dans ce chapitre, nous mettons en œuvre une méthodologie rigoureuse pour analyser la sécurité de la plateforme de demande en lignes des allocations d'études universitaires.

2.1 Le processus de réalisation des tests d'intrusions web selon OWASP

Le pentest suit généralement ces étapes :

- **Explorer** : Le testeur tente de se renseigner sur le système testé. Ceci comprend essayer de déterminer quel logiciel est utilisé. Il comprend également la recherche sur le site de contenu caché,connu vulnérabilités et autres signes de faiblesse ;
- **Attaque** : Le testeur tente d'exploiter les vulnérabilités connues ou suspectées pour prouver qu'ils existent ;
- **Rapport** : Le testeur rend compte des résultats de ses tests, présente les vulnérabilités trouvées,et propose des améliorations.

Le but ultime du pentest est de rechercher des vulnérabilités afin d'évaluer le niveau de sécurité de la cible. Nous allons nous inspirer de ce processus pour effectuer le test d'intrusion de la plateforme de demande en ligne des allocations d'études universitaires.

2.2 Détection de vulnérabilités

Dans cette section, après avoir collecté des informations sur la plateforme de demande en ligne des allocations d'études universitaires, nous détecterons les vulnérabilités qui y sont présentes. La collecte d'information se fera avec Nmap et TheHarvester. La détection de vulnérabilités se fera avec Nikto et OWASP ZAP.

2.2.1 Environnement de travail

La plateforme de demande en ligne des allocations d'études universitaires, est accessible par le lien suivant : https://bourses.enseignementsuperieur.bj. Sa page d'accueil se présente comme suit :

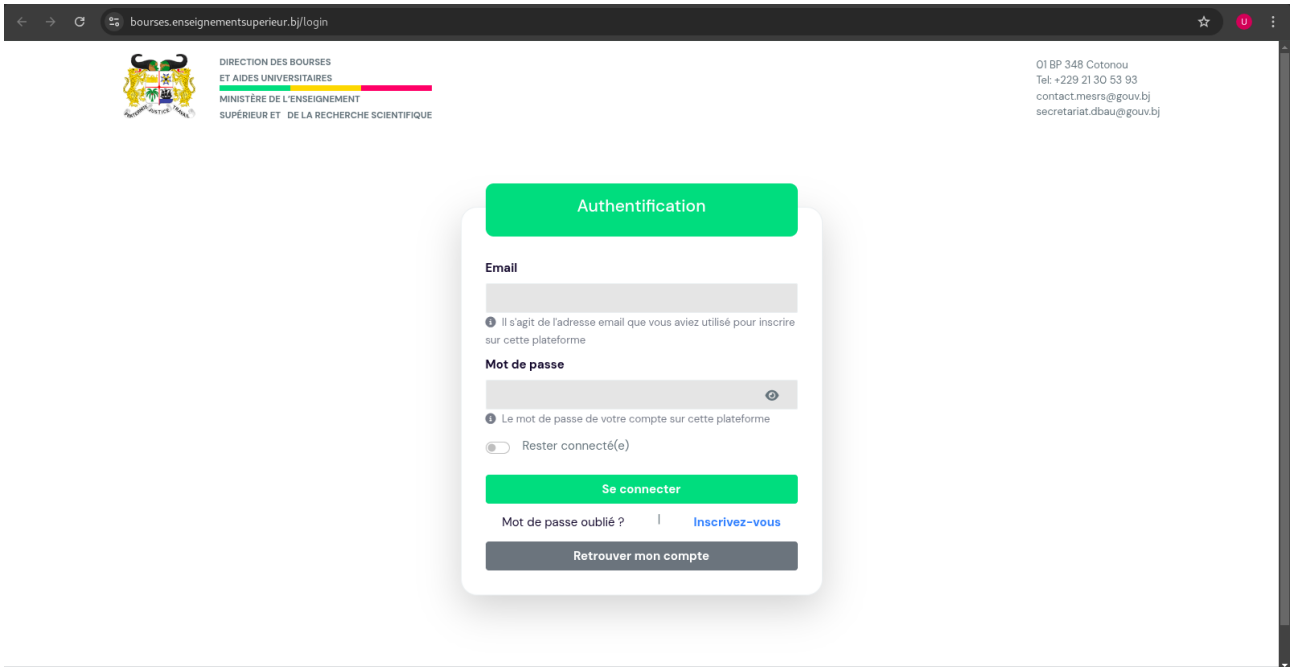


FIGURE 2.1 : Page d'accueil du site web de la DBAU

Cette page est l'interface de connexion à la plateforme de gestion des bourses universitaires du Bénin. Elle permet aux étudiants, de s'authentifier avec leur adresse e-mail et leur mot de passe pour accéder à leurs informations. Elle propose également des options pour rester connecté, récupérer un mot de passe oublié ou créer un compte pour les nouveaux utilisateurs.

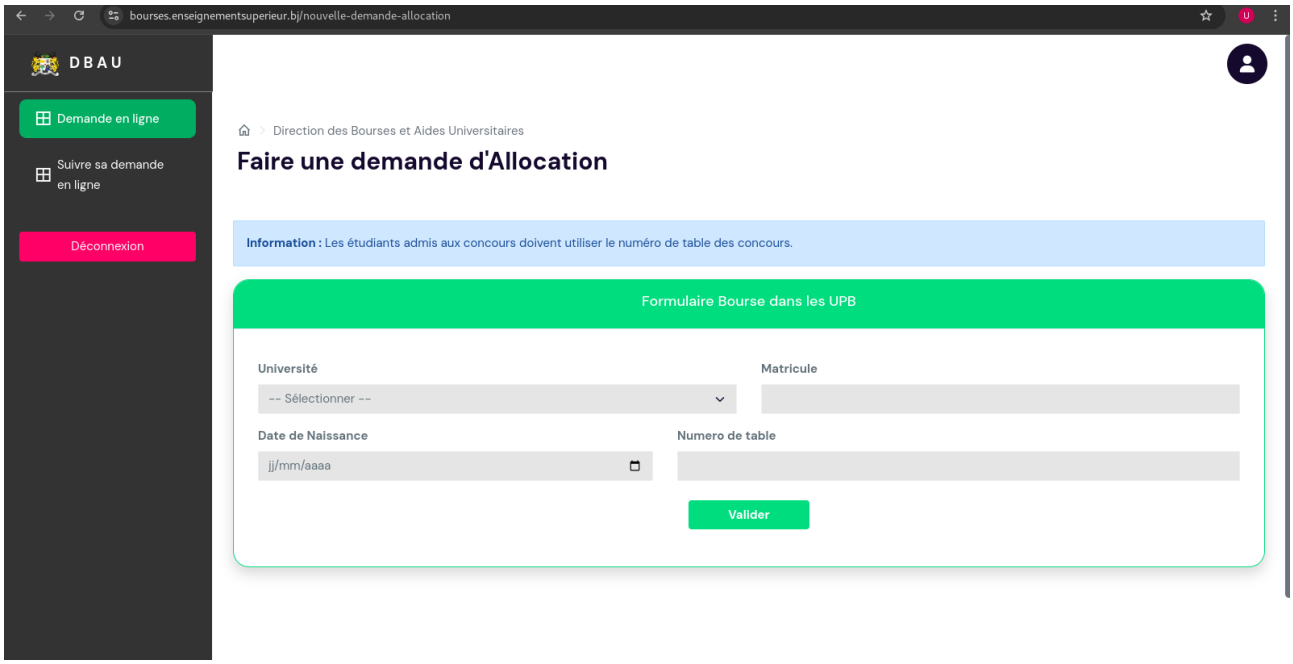


FIGURE 2.2 : Page pour une nouvelle demande

Cette page permet aux étudiants de faire une demande d'allocation de bourse en ligne. Elle contient un formulaire où l'utilisateur doit renseigner son université, son matricule, sa date de naissance et son numéro de table pour valider sa demande. Une section d'information précise que les étudiants admis aux concours doivent utiliser leur numéro de table. Le menu latéral propose des options pour effectuer une demande, suivre son état et se déconnecter.

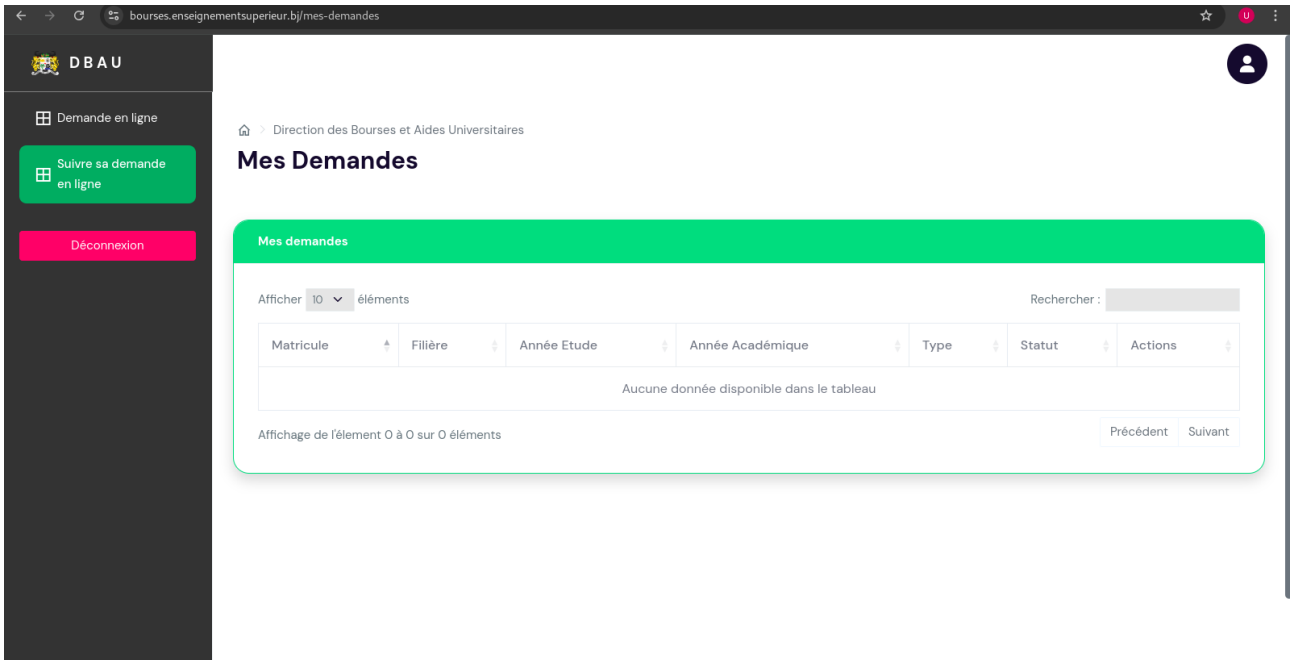


FIGURE 2.3 : Page pour suivre les demandes

Cette page permet aux utilisateurs de suivre l'état de leurs demandes de bourse. Elle affiche un tableau listant les informations clés telles que le matricule, la filière, l'année d'étude, l'année académique, le type de demande, son statut et les actions disponibles. Actuellement, aucune demande n'est enregistrée, ce qui est indiqué par le message "Aucune donnée disponible dans le tableau".

1. **Recherche d'infomation avec Nmap**

   Nous avons lancé un scan de ports pour obtenir des informations sur la plateforme.

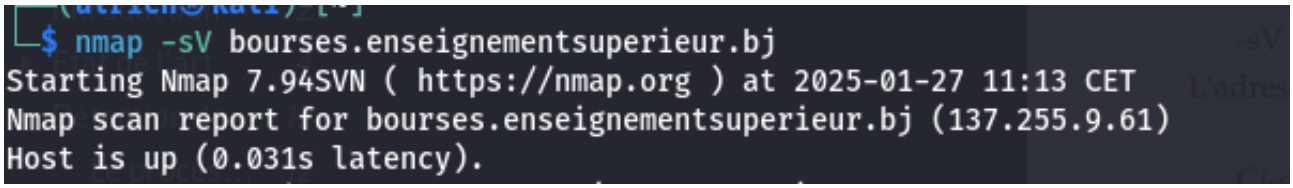


FIGURE 2.4 : Lancement de nmap

-sV permet de sonder les ports ouverts pour déterminer les informations de service/version. L'adresse IP du serveur qui héberge la plateforme de la DBAU est : 137.255.9.61

Ci-dessous, sont présentés les ports et les services qui y sont associés.

TABLE 2.1 : Les ports ouverts .

Port	Service
53	DNS(Domain Name system
80	HTTP(Hyper Text Transfer Protocol)
443	HTTPS(Hyper Text Transfer Protocol Secure)

2. **Recherche d'information avec whatweb**

   Grâce à l'ingénierie sociale, nous avons appris auprès d'anciens stagiaires que le site web aurait été conçu avec le framework Laravel. Pour vérifier cette information, nous avons décidé de lancer un scan avec whatweb afin de confirmer l'utilisation de cette technologie.

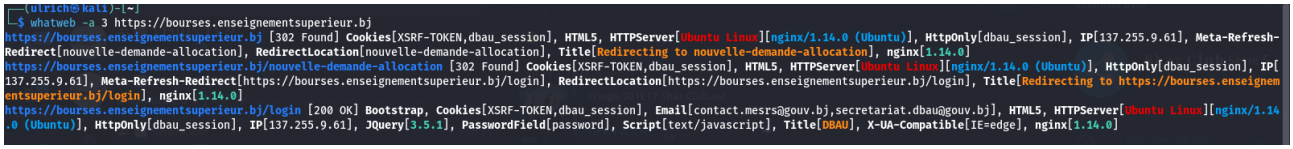


FIGURE 2.5 : Lancement de whatweb

L'analyse du résultat de la commande whatweb sur le site https://bourses.enseignementsuperieur.bj révèle plusieurs informations importantes sur la structure, les technologies utilisées et les redirections du site.

-a : L'option -a permet de spécifier le niveau d'agressivité du scan.

-3 : C'est la valeur passée à l'option -a, définissant le niveau d'agressivité avec le niveau -3

Voici une décomposition :

1. Page principale (https://bourses.enseignementsuperieur.bj)

Statut HTTP : 302 Found (Redirection).

Cookies :XSRF-TOKEN : Probablement utilisé pour la protection contre les attaques CSRF.

dbau_session : Un cookie de session pour maintenir l'état utilisateur.

Technologies détectées

- **Serveur HTTP** : Nginx version 1.14.0 sur Ubuntu Linux.
- **HTML5** : Le site utilise HTML5.

Redirection : La page redirige vers /nouvelle-demande-allocation via un Meta-Refresh et une entête HTTP RedirectLocation.

Adresse IP : Le site est hébergé à l'adresse IP 137.255.9.61.

2. Redirection intermédiaire
(https://bourses.enseignementsuperieur.bj/nouvelle-demande-allocation)

Statut HTTP : 302 Found (Redirection).

Cookies : Les mêmes cookies XSRF-TOKEN et dbau_session sont présents.

Technologies détectées

- **Serveur HTTP** : Nginx version 1.14.0 sur Ubuntu Linux.
- **HTML5** : Le site utilise HTML5.

Redirection : Cette page redirige à nouveau, cette fois vers https://bourses.enseignementsuperieur.bj/login (page de connexion).

Adresse IP : Toujours 137.255.

3. Page de connexion (https://bourses.enseignementsuperieur.bj/login)

Statut HTTP : 200 OK (page atteinte avec succès).

Cookies : les mêmes cookies XSRF-TOKEN et dbau_session.

Technologies détectées

- **Framework CSS** : Utilisation de Bootstrap pour la mise en page.
- **JQuery** : Version 3.5.1 utilisée pour les fonctionnalités dynamiques.
- **HTML5**
- **Serveur HTTP** : Toujours nginx version 1.14.0 sur Ubuntu Linux.

Adresses e-mail trouvées :

- contact.mesrs@gouv.bj.

- secretariat.dbau@gouv.bj.

Après avoir analysé ces informations, nous pouvons conclure que le site a été développé avec Laravel. Laravel utilise fréquemment des en-têtes spécifiques qui peuvent trahir son utilisation.

- Laravel génère un cookie nommé XSRF-TOKEN, utilisé pour la protection CSRF.
- Un autre cookie, généralement appelé **laravel_session**, sert à gérer les sessions utilisateur

Dans notre analyse, la présence des cookies XSRF-TOKEN et dbau_session peut confirmer l'utilisation de Laravel, ce dernier ressemblant à un cookie de session Laravel personnalisé.

3. **Recherche d'information sur les dossiers et les fichiers avec gobuster**

   Voici une analyse de gobuster qui permet de confirmer que le site est développée avec le framework Laravel

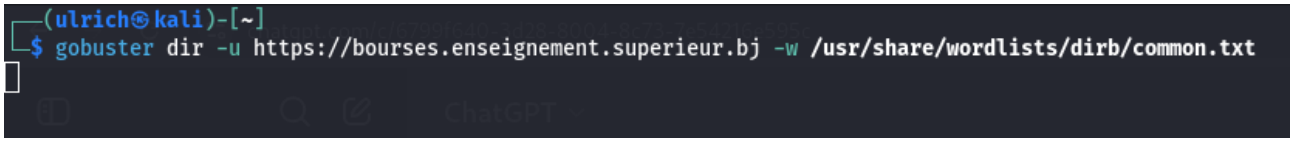


FIGURE 2.6 : Lancement de gobuster

Explication des options :

gobuster dir → Mode de scan basé sur l'énumération de répertoires et fichiers.

-u https ://bourses.enseignement.superieur.bj → URL cible.

-w /usr/share/wordlists/dirb/common.txt → Liste de mots utilisée pour deviner les répertoires et fichiers cachés.

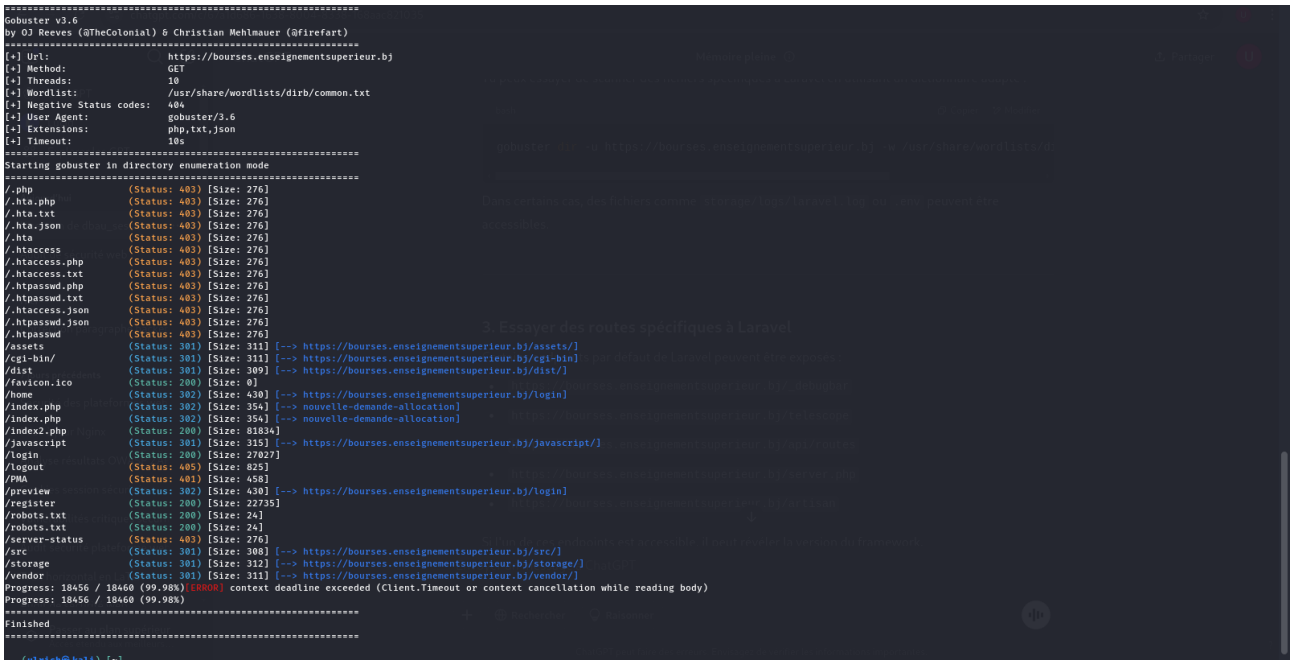


FIGURE 2.7 : Résultats avec gobuster

Après analyse, nous avons identifié plusieurs éléments :

- Le fichier **index.php**, ce qui prouve que la plateforme utilise PHP.
- Le dossier **vendor**, qui est typiquement présent dans les projets utilisant Composer, notamment le framework Laravel.

Sur la base de cette analyse approfondie, nous pouvons conclure que la plateforme est développée avec Laravel. La présence d'éléments caractéristiques comme le fichier **index.php**, le dossier **vendor** et les en-têtes HTTP conforte cette hypothèse.

4. **Recherche des sous domaines de la plateforme avec TheHarvester**

Avant de procéder à une analyse approfondie, nous avons utilisé **theHarvester**, un outil puissant permettant d'extraire des données publiques liées à un domaine spécifique.



FIGURE 2.8 : Lancement de TheHarvester

Explication des options :

- **-d bourses.enseignementsuperieur.bj** : Spécifie le domaine cible (bourses.enseignementsuperieur.bj).
- **-b all** : Demande à TheHarvester d'utiliser toutes les sources disponibles pour la recherche (Google, Bing, LinkedIn, etc.).

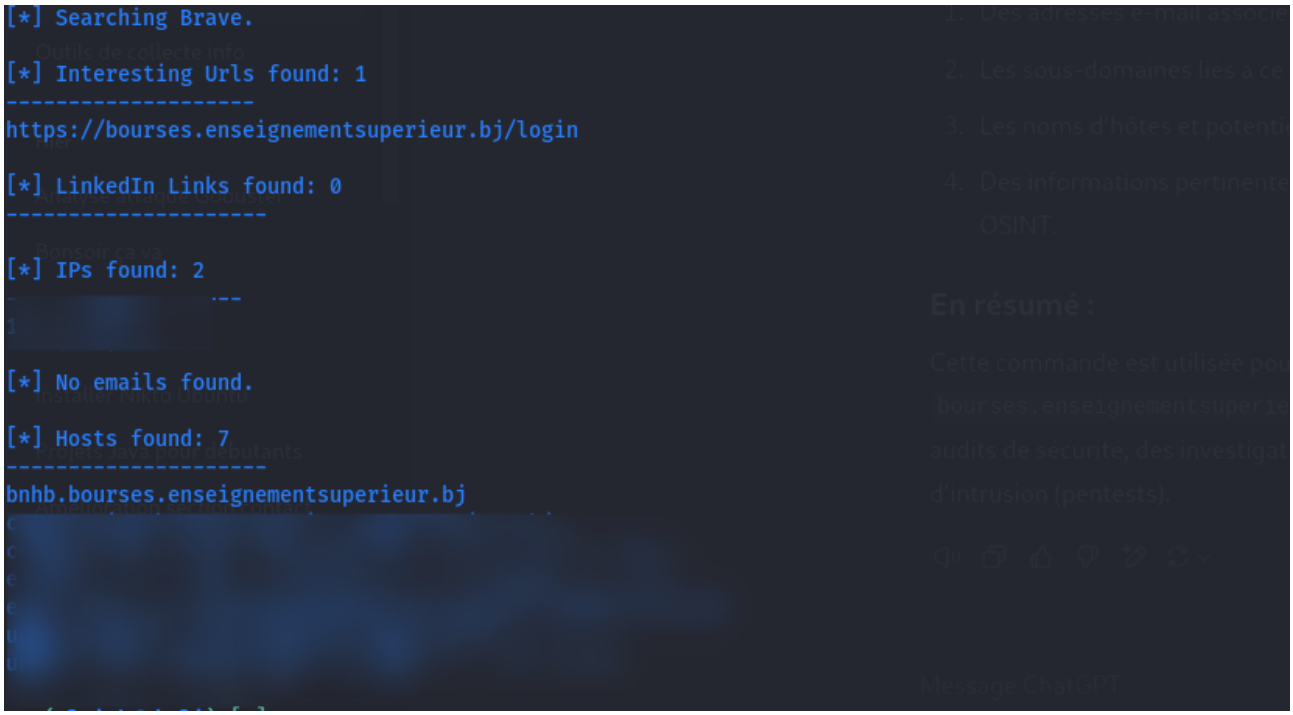


FIGURE 2.9 : Resultat de TheHarvester

Dans ce cas, l'analyse a révélé plusieurs sous-domaines (cooperation., excellence., etc.), une adresse **(137.255.9.61).**Ces résultats mettent en évidence des informations clés sur l'infrastructure de la plateforme. Bien que cette collecte soit basée sur des sources publiques, elle peut servir de point de départ pour des analyses plus approfondies.

2.2.2 Détection de vulnérabilités

Pour la détection des vulnérabilités, nous allons utiliser trois programmes à savoir **OWASP ZAP** et **Nikto** . Cela nous permettra d'avoir une certitude vis-à-vis des résultats.

1. **Scan de vulnérabilités avec Nikto**

   Nous allons lancer un scan pour détecter les vulnérabilités présentes sur le serveur web sur lequel est héberger la plateforme. Grâce à **Nmap**, nous avions déjà pu avoir certaines informations sur le serveur. Son IP est : 137.255.9.61



FIGURE 2.10 : Lancement de nikto

Explication des options :

-h permet de préciser la cible. Elle peut être précisée sous forme d'IP ou d'URL.

-p permet de préciser le port à utiliser. Par défaut, c'est le port 80 (http).



FIGURE 2.11 : Scan de vulnérabilités web présentes sur le serveur

Maintenant nous allons lancer un scan de vulnérabilités de la plateforme de demande en ligne des allocations d'études universitaires. Nous allons utiliser l'URL.

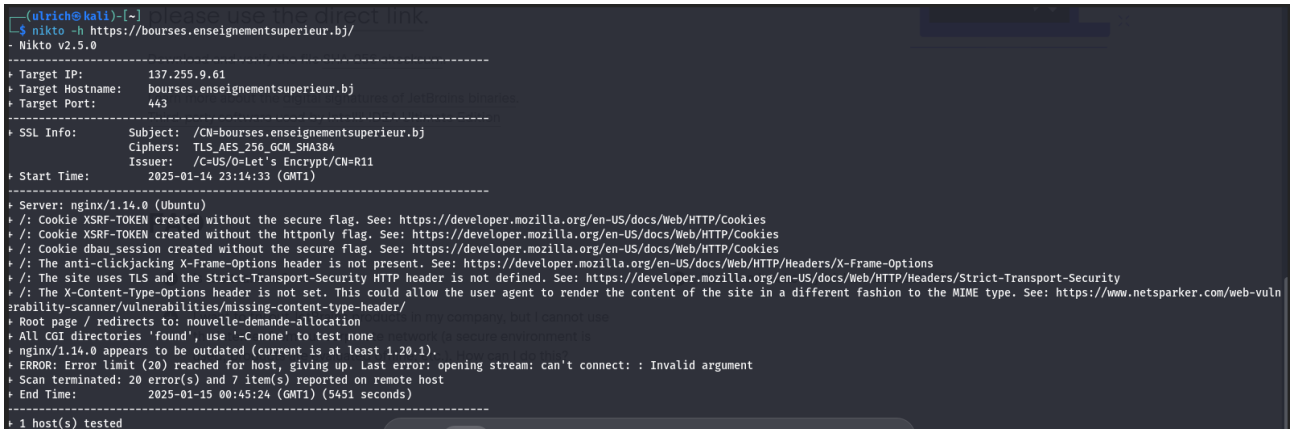


FIGURE 2.12 : Scan de vulnérabilités web présentes sur le serveur

Les scans Nikto, illustrés dans les deux figures, ont identifié des problèmes de sécurité critiques. Pour le premier scan, l'absence des en-têtes X-Frame-Options et X-Content-Type-Options expose le site à des attaques de type clickjacking et MIME sniffing. Pour le deuxième scan, des cookies sans les flags Secure et HttpOnly ont été détectés, ce qui les rend vulnérables à des attaques de vol de session.

Ce scan nous a permis de retenir les vulnérabilités suivantes :

- *clikjaking*
- *Cross site scripting (XSS)*

2. **Test d'intrusion avec OWASP ZAP**

Le test d'intrusion se lancera en sélectionnant **le mode attaque** de OWASP ZAP et en renseignant l'URL

La figure **2.13** montre l'interface d'OWASP ZAP lorsqu'on lance notre attaque sur la plateforme. Dans l'onglet **Alertes**, nous avons les alertes suivantes :

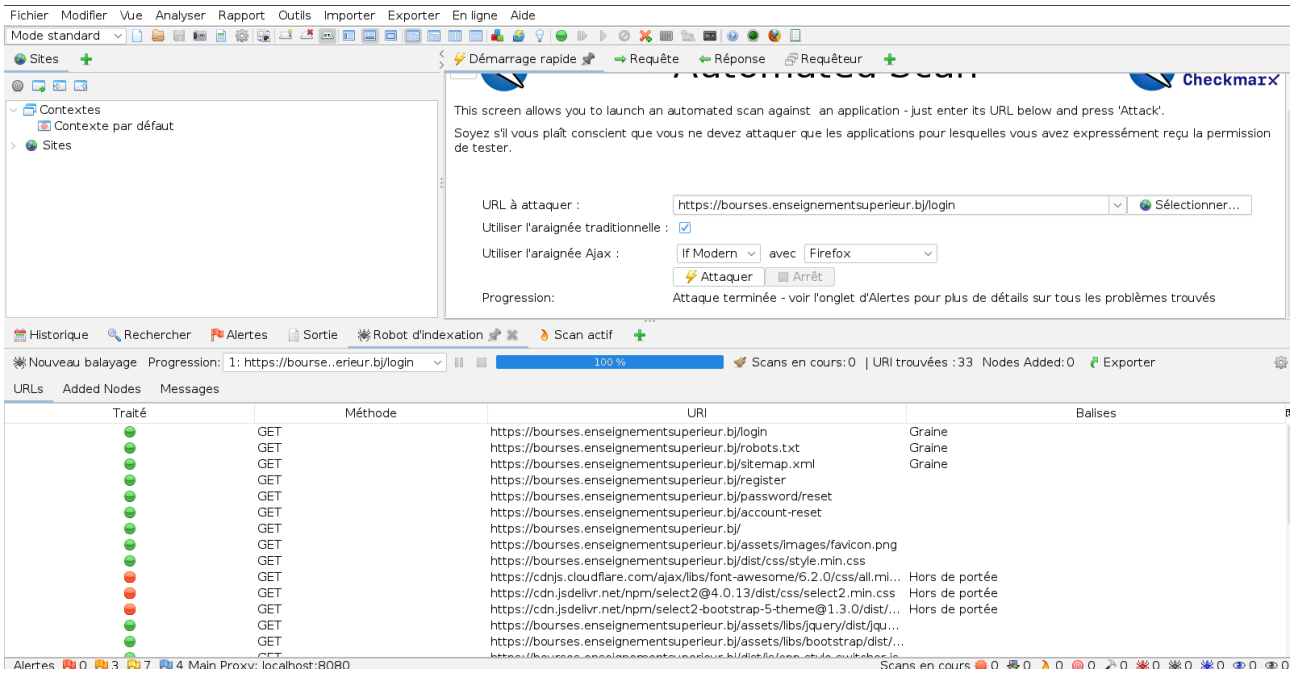


FIGURE 2.13 : Test d'intrusion avec OWASP ZAP

Les figures **2.14**, **2.15**, **2.16** révèlent plusieurs vulnérabilités de sécurité, notamment des en-têtes HTTP manquants ou mal configurés, des risques de fuite d'informations sensibles et des problèmes de gestion des sessions.

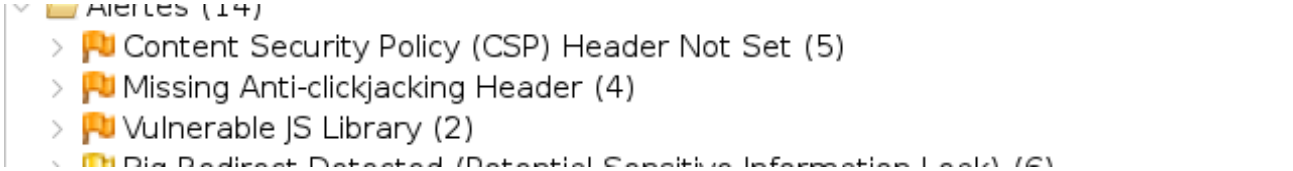


FIGURE 2.14 : Alerte 1(**priorité moyenne**)

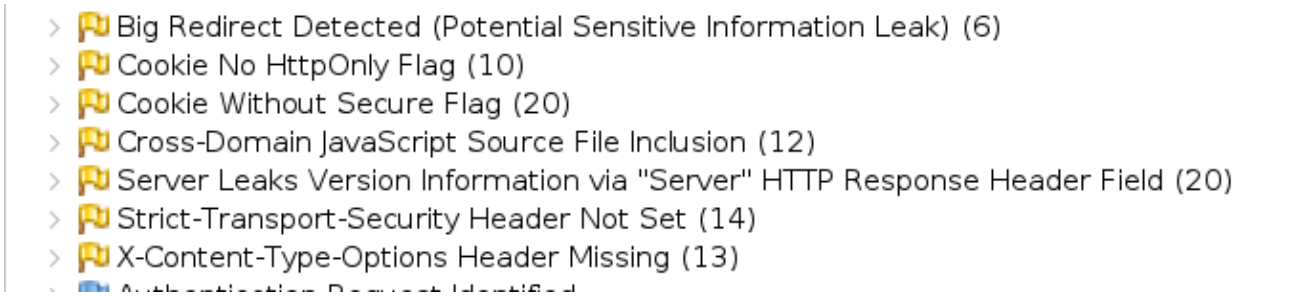


FIGURE 2.15 : Alerte 2 (**priorité faible**)

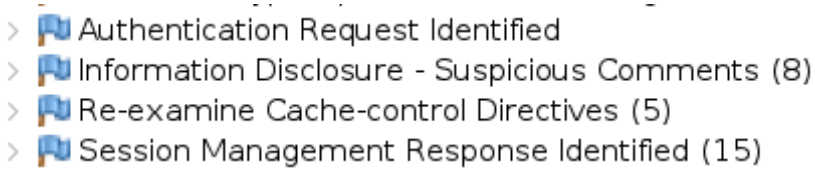


FIGURE 2.16 : Alerte 3 (**pour information**)

Description des alertes générées

- **Content Security AlertsPolicy (CSP) Header Not Set** : la politique de sécurité du contenu (CSP) n'est pas définie. Cela signifie que le site pourrait être vulnérable aux attaques de type Cross-Site Scripting (XSS) et à l'injection de contenu malveillant.Ces attaques sont utilisées pour tout, du vol de données à la dégradation du site ou à la distribution de logiciels malveillants.
- **Missing Anti-clickjacking Header** : l'en-tête X-Frame-Options est absent. Cela signifie que le site peut être intégré dans un iframe sur un site malveillant, ce qui facilite les attaques de type clickjacking.
- **Vulnerable JS Library** : une bibliothèque JavaScript utilisée présente une vulnérabilité connue.
- **Big Redirect Detected (Potential Sensitive Information Leak)** : des redirections importantes ont été détectées, ce qui peut exposer des informations sensibles si mal configurées.
- **Cookie No HttpOnly Flag** : les cookies ne sont pas protégés avec l'attribut HttpOnly, ce qui signifie qu'ils peuvent être accessibles via JavaScript et potentiellement volés via une attaque XSS.S'il s'agit d'un cookie de session alors le détournement de session pourra être possible.
- **Cookie Without Secure Flag** : les cookies ne sont pas marqués comme Secure, ce qui signifie qu'ils peuvent être envoyés en clair sur une connexion HTTP non sécurisée.
- **Cross-Domain JavaScript Source File Inclusion** : Des fichiers JavaScript sont inclus depuis un domaine externe, ce qui peut être une faille de sécurité si ces fichiers sont compromis.
- **Server Leaks Version Information via "Server" HTTP Response Header Field** : le serveur expose sa version dans l'en-tête Server, ce qui peut aider les attaquants à identifier plus facilement d'éventuelles vulnérabilités liées au serveur d'applications Web.
- **Strict-Transport-Security Header Not Set** : l'en-tête HTTP Strict-Transport-Security (HSTS) n'est pas défini.La sécurité de transport HTTP stricte est un mécanisme de politique de sécurité Web par lequel un serveur Web déclare que les agents utilisateurs conformes doivent interagir avec lui en utilisant uniquement des connexions https sécurisées, ce qui signifie que les utilisateurs pourraient être exposés à des attaques de type downgrade (passage en HTTP non sécurisé).
- **X-Content-Type-Options Header Missing** : l'en-tête **X-Content-Type-Options** est absent, permettant à un attaquant d'exécuter des fichiers sous un autre type MIME.
- **Authentication Request Identified** : une requête d'authentification a été détectée. Si mal sécurisée, cela peut exposer des identifiants.

- **Information Disclosure - Suspicious Comments** : des commentaires suspects (potentiellement contenant des informations sensibles) ont été trouvés dans le code source.
- **Re-examine Cache-control Directives** : les directives Cache-Control ne sont peut-être pas bien configurées, ce qui peut entraîner une mise en cache de données sensibles côté client.
- **Session Management Response Identified** : une réponse liée à la gestion de session a été détectée, ce qui peut signifier un risque si les sessions ne sont pas correctement protégées.

Dans ce tableau nous résumons le travail fait dans cette section.

TABLE 2.2 : Vulnérabilités détectées et outils utilisés.

Outils	Utilisation	Vulnérabilités Déctectés
Nikto	Scan de vulnérabilités	Clickjaking ; Cross site scripting(XSS).
Owasp zap	Test d'intrusion	Entête X-Frame-Options non renseigné ; Content Security Policy Header Not Set ; Cookie sans HttpOnly Flag) ; Missing Anti-clickjacking Header ;Cross-Domain JavaScript Source File Inclusion.

Conclusion

De tout ce qui précède, le test d'intrusion de la plateforme de demande en ligne des allocations d'études universitaires permet de mettre en évidence les vulnérabilités et les points faibles du système, tout en soulignant ses aspects solides. Cette démarche rigoureuse permet non seulement d'identifier les failles susceptibles d'être exploitées par des attaquants, mais aussi de proposer des mesures correctives pour renforcer la sécurité globale de la plateforme. Il est essentiel d'effectuer régulièrement de tels tests afin d'assurer la protection des données sensibles et la confidentialité des utilisateurs, tout en garantissant la fiabilité et l'intégrité du système face à d'éventuelles attaques.

Chapitre 3

Résultats et recommandations

Introduction

L'analyse des vulnérabilités réalisée sur la plateforme de demande en ligne des allocations d'études universitaires dans le chapitre précédent a permis d'obtenir des résultats significatifs. Dans ce chapitre, nous présentons une synthèse des résultats obtenus et proposons des recommandations précises pour corriger ces failles.

TABLE 3.1 : Vulnérabilités détectées et outils utilisés

Outils	Utilisation	Vulnérabilités Déctectés
Nikto	Scan de vulnérabilités	Clickjaking ; Cross site scripting(XSS).
Owasp zap	Test d'intrusion	Entête X-Frame-Options non renseigné ; Content Security Policy Header Not Set ; Cookie sans HttpOnly Flag) ; Missing Anti-clickjacking Header ;Cross-Domain JavaScript Source File Inclusion

Nous constatons que le test d'intrusion a fourni des résultats plus détaillés et approfondis que le scan de vulnérabilités. Les alertes générées par OWASP ZAP offrent davantage d'informations et de preuves sur les vulnérabilités de la plateforme web . Dans les sections suivantes, nous présenterons les alertes les plus critiques. Étant donné que la plateforme est développée avec Laravel, nos recommandations et implémentations proposées en tiennent compte.

3.1 Content Security Policy (CSP) Header Not Set

URL : https://bourses.enseignementsuperieur.bj/sitemap.xml

Risque : Moyen

Description : L'en-tête CSP (Content Security Policy) n'est pas défini, ce qui expose l'application à des attaques XSS (Cross-Site Scripting) et à l'injection de contenu malveillant.

Impact : Les attaquants peuvent injecter du JavaScript malveillant dans l'application, compromettant les sessions utilisateur et volant des informations sensibles.

Recommandation : Mettre en place une politique de sécurité du contenu via l'en-tête Content-Security-Policy.

Implémentation de la recommandation : Ajouter un en-tête HTTP comme :



FIGURE 3.1 : Sécurisation du CSP

L'implémentation de cette recommandation est détaillé aussi sur : https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP

3.2 Missing Anti-clickjacking Header

URL : https://bourses.enseignementsuperieur.bj/

Risque : Moyen

Description : La réponse ne protège pas contre les attaques de type « ClickJacking ». Elle devrait inclure soit Content-Security-Policy avec la directive X-Frame-Options.

Impact : Un attaquant peut charger votre site dans un iframe invisible et inciter un utilisateur à effectuer des actions non désirées.

Recommandation : Ajouter l'en-tête X-Frame-Options avec une valeur appropriée

Implémentation de la recommandation : L'implémentation de cette recommandation est détaillé aussi sur : https://developer.mozilla.org/en-US/docs/Web/Security/Attacks/Clickjacking

3.3 Vulnerable JS Library

URL : https://bourses.enseignementsuperieur.bj/assets/libs/jquery/dist/jquery.min.js/

Risque : Moyen

Description : L'application utilise une bibliothèque JavaScript avec des vulnérabilités connues. La bibliothèque identifiée jquery, version 3.3.1 est vulnérable.

Impact : Les attaquants peuvent exploiter ces failles pour compromettre l'application ou exécuter du code malveillant.

Recommandation : Mettre à jour les bibliothèques JavaScript concernées.Dans notre cas c'est la bibliothèque jquery

Implémentation de la recommandation : Mettre à jour les bibliothèques JavaScript concernées. Utiliser des outils comme Snyk ou Retire.js pour identifier les versions vulnérables.

3.4 Cross-Domain JavaScript Source File Inclusion

URL : https://bourses.enseignementsuperieur.bj

Risque : Moyen

Description : L'inclusion de fichiers JavaScript provenant de domaines externes peut introduire des vulnérabilités si ces ressources ne sont pas sécurisées

Impact : Un attaquant pourrait exploiter cette vulnérabilité pour exécuter du code JavaScript malveillant dans le contexte de votre site, entraînant des attaques de type Cross-Site Scripting (XSS) ou vol de données sensibles.

Recommandation : Éviter d'inclure des fichiers JavaScript provenant de domaines externes non fiables. Privilégier l'utilisation de ressources locales ou de CDNs (Content Delivery Networks) de confiance pour les fichiers JavaScript.

3.5 Cookie No HttpOnly Flag

URL : https://bourses.enseignementsuperieur.bj/

Risque : Moyen

Description : Certains cookies ne sont pas marqués avec l'attribut HttpOnly, ce qui permet à des scripts malveillants de les lire via JavaScript.

Impact : Si un script malveillant peut être exécuté sur cette page, le cookie sera accessible et pourra être transmis à un autre site. S'il s'agit d'un cookie de session, le détournement de session peut être possible.

Recommandation : Activer l'attribut HttpOnly sur tous les cookies sensibles.

Implémentation de la recommandation : Les cookies sont utilisés sur toute l'application, il faut donc une configuration globale pour sécuriser tous les cookies. Dans laravel, vous devez modifier la configuration **config / session.php**, définir l'indicateur sécurisé sur "true" comme illustré ci-dessous :

```
'secure' ⇒ env( 'SESSION_SECURE_COOKIE', true ),
```

FIGURE 3.2 : Sécurisation des coockies

Lorsque cette option est définie sur "true", les cookies de session seront transmis au serveur uniquement si la connexion du navigateur est sécurisée via HTTPS [15]. Cela empêche l'envoi du cookie dans des conditions non sécurisées.

3.6 Server Leaks Version Information via "Server" HTTP Response Header Field

URL : https://bourses.enseignementsuperieur.bj/

Risque : Moyen

Description : L'en-tête Server divulgue des informations sur la version du serveur, facilitant l'exploitation des vulnérabilités connues.

Impact : Un attaquant peut cibler des vulnérabilités spécifiques en fonction de la version du serveur.

Recommandation : Supprimer ou masquer l'en-tête Server dans la configuration du serveur web.

Implémentation de la recommandation : Après avoir analysé les vulnérabilités, nous avons constaté que le site était hébergé sur un serveur Nginx. Afin de sécuriser les informations, nous procéderons à une mise à jour du fichier de configuration de Nginx.

```
nginx

# Nginx
server_tokens off;


<!— IIS web.config —>
<httpProtocol>
   <customHeaders>
       <remove name="Server" />
   </customHeaders>
</httpProtocol>
```

FIGURE 3.3 : Sécurisation des informations du serveur

L'implémentation de cette recommandation est détaillée aussi ici : https://www.iothreat.com/blog/server-leaks-version-information-via-server-http-response-header-field2

Conclusion

Le recours à des frameworks modernes comme laravel facilite la création de sites internet en optimisant le temps de développement et en simplifiant leur maintenance. Toutefois, l'aspect sécuritaire reste un enjeu majeur. Le choix d'un environnement robuste ne suffit pas à garantir une protection optimale, tout dépend de son exploitation. Il arrive fréquemment que certaines protections intégrées soient désactivées afin d'accélérer le développement, ce qui expose davantage les plateformes aux vulnérabilités. Par ailleurs, les solutions les plus répandues sont souvent la cible privilégiée des attaques, car une faille sur un système largement adopté peut impacter un grand nombre d'utilisateurs. Même sans manipulation de données confidentielles, assurer une protection efficace reste essentiel. De nombreux intrus prennent le contrôle de sites pour les utiliser à des fins malveillantes, comme relais d'attaques ou hébergement d'activités frauduleuses. Aucune infrastructure n'est totalement à l'abri, d'où la nécessité de contrôles réguliers. Ce suivi exige une vigilance constante, d'autant plus que le domaine du numérique est en constante évolution.

Conclusion générale et perspectives

La sécurité des applications web représente un enjeu majeur à ne pas négliger. Avec l'essor rapide des plateformes en ligne, il est essentiel de garantir leur protection, car les cybercriminels ne cessent de représenter une menace. Une plateforme en ligne est jugée fiable lorsqu'elle dispose d'un haut niveau de sécurité. L'objectif de la cybersécurité est d'assurer **l'intégrité** et la **disponibilité** des systèmes informatiques, de préserver la **confidentialité** des données en les rendant accessibles uniquement aux personnes autorisées, et de permettre la traçabilité des actions suspectes. Bien que cela puisse sembler complexe et coûteux, il s'agit d'une nécessité pour toute organisation soucieuse de protéger son infrastructure numérique. Il est donc indispensable d'intégrer les bonnes pratiques de sécurité dès la conception et tout au long du développement d'une application web.

Aucune solution ne garantit une protection absolue, mais il est possible de rendre l'accès aux attaquants suffisamment difficile pour les dissuader, sauf en cas d'intérêt stratégique. Les mesures de protection doivent être proportionnées à l'importance de la plateforme concernée, et la première étape vers une meilleure sécurité consiste à réaliser un test d'intrusion. Ce mémoire s'attache ainsi à mener une évaluation approfondie de la plateforme web de demande en ligne des allocations universitaires afin d'identifier ses vulnérabilités et de proposer des mesures correctives visant à renforcer sa protection. Les recommandations formulées s'appuient sur les résultats des analyses de scans de sécurité et des tests d'intrusion réalisés.

À l'avenir, une piste d'amélioration serait de suivre l'évolution du niveau de protection de la plateforme web de demande en ligne des allocations d'études universitaire grâce à travers :

- **la réalisation de tests d'intrusion réguliers** pour identifier et corriger les potentielles failles de sécurité
- **l'application continue des recommandations** issues des tests de sécurité pour renforcer la protection de la plateforme.
- **la mise en place d'une politique de sécurité spécifique** afin d'instaurer des règles et bonnes pratiques adaptées aux besoins de la plateforme.

Bibliographie

[1] F. OWASP. OWASP TOP 10. “OWASP”. In : (2007). Consulté le 08 janvier 2025.

[3] Patrick ENGEBRETSON. “Les bases du hacking”. In : (2013). Consulté le 05 janvier 2025.

[4] OWASP ZAP 2.8 Getting Started GUIDE. “O. Z. . G. S. Guide”. In : (2008). Consulté le 08 janvier 2025.

[5] D. KENNEDY. “Hacking, sécurité et tests d’intrusion avec Metasploit.” In : (2017). Consulté le 08 janvier 2025.

[6] A. LEGEEK. “Les différents types de hackers”. In : (2013). Consulté le 08 janvier 2025.

[7] Steven LEVY. “L’éthique des hackers”. In : (2013). Consulté le 08 janvier 2025.

[16] Raphael YENDE. “HAL. SUPPORT DE COURS DE L’AUDIT DES SYSTEMES D’INFORMATION (INFORMA- TIQUE)”. In : (2018). Consulté le 08 janvier 2025.

Webographie

[2] ALLISTIC. *Audit de vulnérabilité et test d'intrusion*. Consulté le 12 décembre 2024. URL : https://www.allistic.fr/pentest-test-dintrusion/.

[8] OWASP. *Défaillances de la journalisation et de la surveillance de la sécurité*. Consulté le 05 janvier 2025. URL : https://www.vumetric.com/fr/blogue/owasp-top-10-a09-security-logging-and-monitoring-failures/.

[9] OWASP. *OWASP Top 10 – A02 Défaillances cryptographiques*. Consulté le 14 décembre 2024. URL : https://www.vumetric.com/fr/blogue/owasp-top-10-a02-defaillances-cryptographique.

[10] OWASP. *OWASP Top 10 – A04 Conception non sécurisée, tout ce que vous devez savoir*. Consulté le 18 décembre 2024. URL : https://www.vumetric.com/fr/blogue/owasp-top-10-a04-insecure-design-tout-ce-quil-faut-savoir/.

[11] OWASP. *OWASP Top 10 – A06 Explication des composants vulnérables et obsolètes*. Consulté le 10 décembre 2024. URL : https://www.vumetric.com/fr/blogue/owasp-top-10-a06-vulnerable-and-outdated-components-explained//.

[12] OWASP. *OWASP Top 10 – A08 Les failles d'intégrité des logiciels et des données expliquées*. Consulté le 15 décembre 2024. URL : https://www.vumetric.com/fr/blogue/owasp-top-10-a08-software-and-data-integrity-failures-explique/.

[13] OWASP. *OWASP Top 10 – Analyse des vulnérabilités de l'injection A03*. Consulté le 13 décembre 2024. URL : https://www.vumetric.com/fr/blogue/owasp-top-10-a03-injection-vulnerabilites-analysees/.

[14] scope et types de tests d'intrusion. PENTEST : MÉTHODOLOGIE déroulement. *Audit de vulnérabilité et test d'intrusion*. Consulté le 18 décembre 2024. URL : https://www.vaadata.com/blog/fr/pentest-methodologie-deroulement-scope-et-types-de-tests-dintrusion/.

[15] STACKOVERFLOW. *how to set secure flag on cookies in laravel*. Consulté le 01 février 2025. URL : https://stackoverflow.com/questions/47299809/how-to-set-secure-flag-on-cookies-in-laravel.

Table des matières

Dédicace ii

Remerciements iii

Résumé iv

. iv

Abstract v

. v

Liste des figures vi

Liste des tableaux vii

Liste des acronymes viii

Glossaire 1

Introduction 2

1 Revue de littérature 4

Introduction 4

1.1 Fondamentaux de la sécurité Informatique 4

1.2 Les différents types de hackers 4

1.2.1 Les black hats 5

1.2.2 Les white hats 5

1.2.3 Les grey hats 5

1.3 Les différents types de tests d'intrusion 5

1.3.1 Les tests d'intrusion boîte blanche 5

1.3.2 Les tests d'intrusion boîte noire 6

1.3.3 Les tests d'intrusion boîte grise 6

1.4 Comparaison des tests d'intrusion et outils utilisés 6

1.4.1 Avantages et inconvénients des tests d'intrusion 6

1.4.2 Les outils de pentest web 7

1.5 Les vulnérabilités web selon OWASP 8

1.5.1 Violation du contrôle d'accès 8

1.5.2 Les défaillances cryptographiques 8

1.5.3 Injection 8

1.5.4 Conception non sécurisée 9

1.5.5 Mauvaise configuration de sécurité . . . 9
1.5.6 Composants vulnérables et obsolètes . . . 9
1.5.7 Défauts d'identification et d'authentification . . . 9
1.5.8 Défauts d'intégrité des logiciels et des données . . . 9
1.5.9 Défaillances de la journalisation et de la surveillance de la sécurité . . . 9
1.5.10 Falsification des requêtes côté serveur . . . 10
1.6 Référentiel . . . 10
1.6.1 Conformité aux référentiels . . . 10
1.7 Formulation du problème théorique . . . 10
1.8 Présentation des hypothèses explicatives . . . 11

2 Détection des vulnérabilités sur la plateforme. 12
Introduction . . . 12
2.1 Le processus de réalisation des tests d'intrusions web selon OWASP . . . 12
2.2 Détection de vulnérabilités . . . 13
2.2.1 Environnement de travail . . . 13
2.2.2 Détection de vulnérabilités . . . 19
Conclusion . . . 23

3 Résultats et recommandations 24
Introduction . . . 24
3.1 Content Security Policy (CSP) Header Not Set . . . 25
3.2 Missing Anti-clickjacking Header . . . 25
3.3 Vulnerable JS Library . . . 26
3.4 Cross-Domain JavaScript Source File Inclusion . . . 26
3.5 Cookie No HttpOnly Flag . . . 26
3.6 Server Leaks Version Information via "Server" HTTP Response Header Field . . . 27
Conclusion . . . 28

Conclusion 29

Bibliographie 30

Webographie 31

Table des matières 32