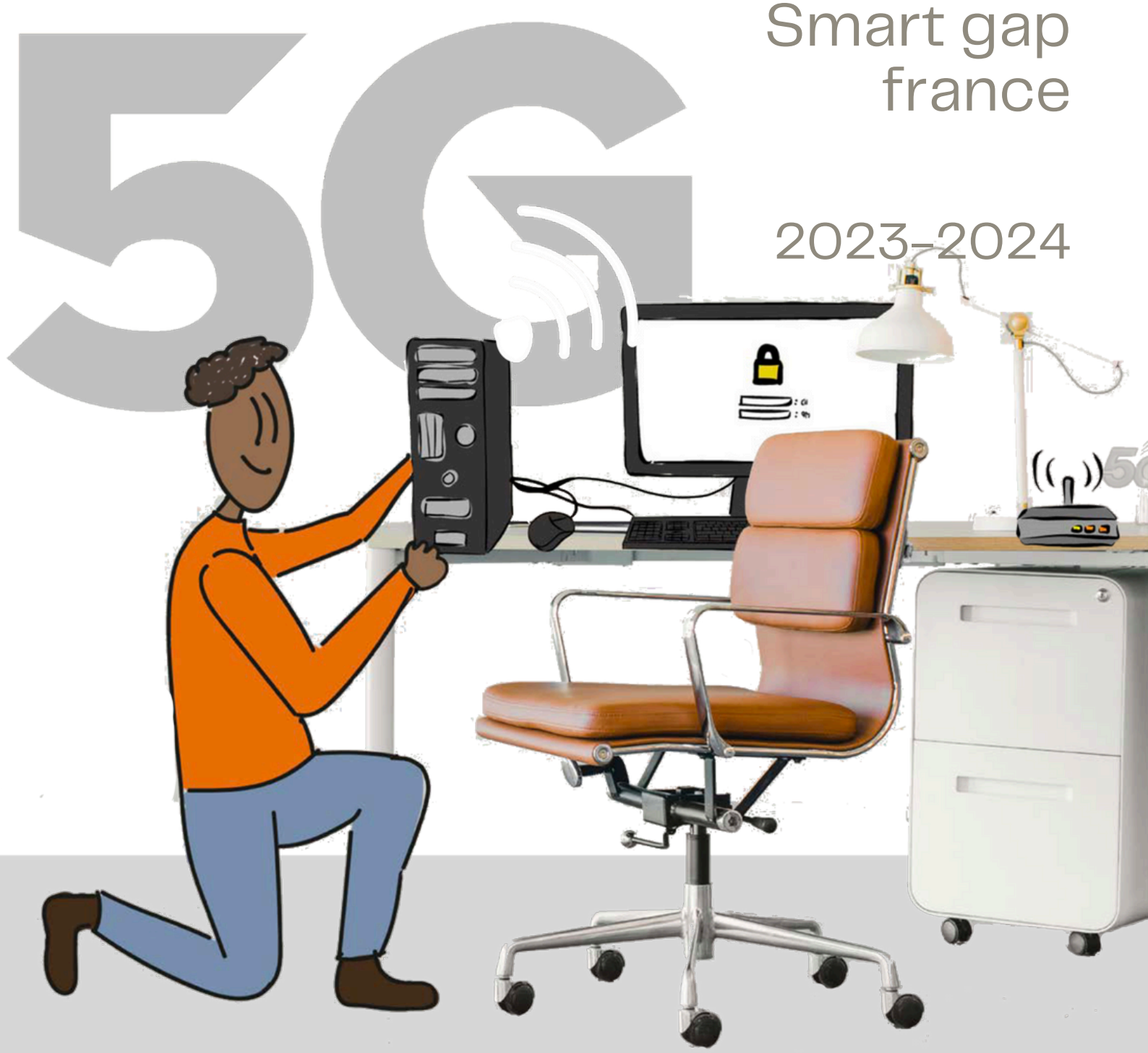
Smart Gap

Rapport de stage

Smart gap france

2023-2024



Gheorghe-Tony Lambert-Tathy

SUP'Chassagnes

Durablement vôtre !

Gheorghe -Tony
LAMBERT-TATHY
RAPPORT de STAGE
2023-2024

Sommaire

Remerciements

Ce rapport est le fruit des semaines de stage que j'ai eu le plaisir de passer au sein de la Smart gap france.

Avant tout développement sur cette expérience professionnelle, il me paraît opportun de commencer ce rapport de stage par
exprimer ma gratitude à Smart Gap pour m'avoir accueilli et offert cette opportunité d'apprentissage.

Merci à Cyril Trambouze, PDG de l'entreprise, pour la confiance accordée en me laissant gérer les missions de manière autonome.

Un grand merci également à Azzelarabe, mon maître de stage, pour son soutien et son aide précieuse tout au long de mon stage

Introduction

Je m'appelle G.Tony LAMBERT-TATHY, étudiant en première année de BTS Services Informatiques aux Organisations (option SISR) à Lyon. Passionné par l'informatique et autodidacte, j'ai obtenu un Bac Pro Système Numérique avec mention très bien. Mon expérience inclut des rôles chez Tibco Services et Palette & Co, où j'ai travaillé sur la programmation, la maintenance informatique, l'administration réseau et le support utilisateur. Mes emplois saisonniers m'ont également permis de développer un sens aigu du service client et de l'organisation.

J'ai choisi ce stage pour acquérir une expérience en cybersécurité, un domaine dans lequel je souhaite travailler à l'avenir.

Dans ce rapport, je vais détailler le déroulement de mon stage chez Smart Gap France.

Nous commencerons par une présentation de l'entreprise, son historique, son organigramme, et l'utilisation de l'informatique au sein de celle-ci.

Ensuite, nous aborderons le projet principal du stage, qui consistait à réaliser des test d'intrusion pour identifier et traiter les vulnérabilités des sites web de deux clients. Je décrirai les outils utilisés, les méthodologies appliquées, ainsi que les solutions mises en place pour surmonter les obstacles rencontrés.

Par la suite, je présenterai le déroulé du projet, en mettant en lumière les dimensions techniques et la planification du projet.

Enfin, je conclurai avec une évaluation de l'état du projet à la fin du stage, mon ressenti personnel, et une réflexion sur les leçons tirées et les compétences acquises.

Les annexes techniques viendront compléter ce rapport avec des documents et des informations supplémentaires nécessaires à la compréhension de mon travail.

Toutes informations juger comme étant sensibles par le tuteur seront censuré dans ce rapport

Présentation du contexte de mon stage

a) L'entreprise

Smart Gap France accompagne les créateurs d'entreprises dans la création de sites web et développe des stratégies de communication efficaces. Fondée en 2018, l'entreprise compte 5 employés.

L'organigramme de l'entreprise est simple :

Cyril Trambouze est le directeur général,

Azzelarabe ainsi que ses 3 autres de ses collègues sont développeur,

et moi(G.Tony Lambert-Tathy), en tant que stagiaire.

L'informatique est l'un des domaines d'activités de l'entreprise plus précisément la création de sites web l'autre domaine de l'entreprise est le développement des stratégies de communication efficaces .

Smart Gap utilise diverses infrastructures et systèmes pour gérer ses projets et assurer la sécurité des sites web de ses clients nous allons découvrir quelques outils utiliser par smartgap dans ce rapport.

Logo de smartgap france

b) Le projet

Smart Gap a besoin de réaliser des test d'intrusion pour identifier les vulnérabilités des sites web qu'elle conçoit pour ses clients.

Durant mon stage, j'ai travaillé sur deux projets pour des clients qui resterons anonymes pour des raisons de sécurité et de confidentialités, nous allons donc nommés ses clients Ecole.ma(car le clients était une ecole situer au maroc) et Info.fr(car le client était une entreprise d'informatique).

Mon travail consistait à identifier les vulnérabilités, évaluer les risques selon la norme ISO 27005, rédiger des rapports, établir des matrices de risques, proposer et tester des solutions de sécurité.

Aucun outil spécifique ne m'a été imposé pour ce projet, ce qui m'a permis de choisir les outils les plus appropriés pour répondre aux besoins des audits de sécurité.

J'ai utilisé Kali Linux, NordVPN, Zap, Burp Suite, WPScan, Nmap, Wappalyzer et Fuff.

Pour plus d'infos sur ses outils veuillez vous référé à l'annexe nommée "outils"

c) Le theme du stage

Pendant ce stage, j'ai réalisé des testes d'intrusion afin d'identifier les différentes vulnérabilités des applications web des clients et proposer des solutions à mettre en place. Après la mise en place des solutions proposées, j'ai refait des tests pour vérifier leur efficacité.

a) Dimension technique

Techniquement, j'ai réalisé des tests d'intrusion pour identifier les vulnérabilités des sites web.

J'ai évalué ces vulnérabilités selon la norme ISO 27005 et rédigé des rapports détaillant les informations sensibles du site trouvées, les vulnérabilités, les tests réalisés, et les solutions potentielles.

J'ai également établi des matrices de risques et mis en place des plans d'action.

Durant la réalisation des tests d'intrusion, j'ai rencontré des difficultés avec les systèmes de défense des sites web qui bloquaient mon adresse IP après l'envoi de plusieurs requêtes malgré l'utilisation d'un VPN classique.

J'ai résolu ce problème en utilisant Onion Over VPN de NordVPN, ce qui m'a permis de contourner les blocages.

b) Conduite du projet et planification

Le projet s'est déroulé de manière structurée et organisée, suivant une méthodologie précise pour atteindre les objectifs définis. Au début de mon stage, la première tâche consistait à cadrer mon intervention en collaboration avec mon tuteur, en établissant les objectifs spécifiques des tests d'intrusion à réaliser. Cette étape a permis de clarifier les attentes et de définir un plan cohérent.

Ensuite, j'ai préparé mon environnement de travail en installant et configurant les outils nécessaires pour mener à bien les tests d'intrusion. Cette phase a impliqué l'installation de Kali Linux, la configuration de NordVPN, ainsi que l'intégration de divers outils de test tels que Zap, Burp Suite, WPScan, Nmap, Wappalyzer, et Fuff. Une préparation minutieuse était essentielle pour assurer l'efficacité et la précision des tests.

Les semaines suivantes ont été consacrées à la recherche d'informations sur les cibles des test d'intrusion, Ecole.ma et Info.fr. J'ai collecté des données techniques détaillées sur les applications web concernées pour mieux comprendre leur architecture et identifier les potentielles vulnérabilités.

Une fois les informations collectées, j'ai réalisé les tests d'intrusion, en appliquant des techniques et des outils spécifiques pour identifier les failles de sécurité. Durant cette phase, j'ai rencontré des difficultés, notamment les systèmes de défense des sites web qui bloquaient mon adresse IP malgré l'utilisation d'un VPN classique. Pour contourner ces blocages, j'ai opté pour l'utilisation d'Onion Over VPN, une fonctionnalité de NordVPN offrant une couche de sécurité supplémentaire en routeant le trafic via le réseau Onion (Tor).

Parallèlement aux tests, j'ai commencé à rédiger un rapport , documentant les informations sensibles trouvées sur les sites, les vulnérabilités identifiées, les tests réalisés et les réponses obtenues. Le rapport comprenait également une description détaillée des vulnérabilités et des solutions potentielles pour y remédier. Une matrice de risques a été établie pour évaluer et prioriser les vulnérabilités en fonction de leur gravité et de leur impact potentiel.

La dernière étape de la conduite du projet a consisté à mettre en place un plan d'action pour corriger les vulnérabilités identifiées. Ce plan incluait des recommandations spécifiques pour renforcer la sécurité des applications web, et j'ai effectué des tests supplémentaires après la mise en œuvre des solutions proposées pour vérifier leur efficacité.

La gestion du temps a été cruciale pour assurer le bon déroulement du projet. Le cadrage et la préparation de l'environnement de travail ont été réalisés lors de la première et de la deuxième semaine. Les recherches d'informations et les tests d'intrusion ont été menés de la deuxième à la quatrième semaine. Enfin, la cinquième semaine a été dédiée à la rédaction du rapport d'audit, à la réalisation de la matrice de risques et à la mise en place du plan d'action.
Cette planification rigoureuse a permis de progresser efficacement et de répondre aux exigences du projet dans les délais impartis.

c) Formations ou compléments de formation nécessaires

Pour répondre aux exigences du projet, j'ai suivi plusieurs formations :

- Réaliser un test d’intrusion – Open Classroom
- Comment auditer un site WordPress avec WPScan ? – IT-Connect
- Matrices de risques – Blog gestion de projet

Conclusion

À la fin du stage, le projet était terminé à 90%, avec les vulnérabilités à haut risque écartées, mais il restait à traiter les vulnérabilités à risque moyen.

Ce stage m'a permis de découvrir le métier de pentester et m'a confirmé mon intérêt pour la cybersécurité.

J'ai pu constater mes points forts et les domaines à améliorer.

Pour une première expérience, je suis satisfait de mes performances, bien que je doive encore progresser sur l'utilisation des outils et la rédaction de rapports.

Annexes tecniques

Les annexes sont des rapport contenant des information technique, c'est aussi les rapport que j'envoyait chaque semaine a mon tuteur.

Toutes informations juger comme étant sensibles par le tuteur seront censuré dans ce rapport. veuillez vous rendre à la dernière page du rapport pour avoir la liste de tous les élément qui ont été censuré.

Rapport de la semaine de stage chez smartgap 06/05-11/05

1. Présentation des entreprises clientes (par le tuteur)

ecole.ma: est une international business school qui délivre des formations diplômantes pour managers et dirigeants. Nos formations sont disponibles en ligne, en présentiel et en blended learning.

Les differents site de l'entreprise sont: CASABLANCA (MAROC), PARIS(FRANCE), BRAZZAVILLE(CONGO).

info.fr: est une entreprise qui accompagne les créateurs d'entreprises dans leur processus de création de site web. ils aident également les entreprises à mettre en place et développer une stratégie de communication efficace.

L'entreprise est situé: à Paris (france) .

Apres mettre rendue sur le site je remarque que le site est indisponible si ce n'est pas l'œuvre du propriétaire du site c'est que le site a subi une attaque DDoS (Distributed Denial of Service, déni de service distribué)

donc mon tuteur m'as dis que la suite mission se porteras sur **ecole.ma**

1. **Définition du périmètre du test**

ecole.ma:

- **Pour définir l'approche à tenir :**
  - **Que fait l'application fonctionnellement ?**

l'application est un site vitrine qui a pour objectif de présenter les different produit (formation dans ce cas) que l'entreprise propose**.**

voici l'arborescences du site:

- **Accueil**
- **Nos Formations**
  a. Bachelor
  b. Master
  c. DBA
    a. Présentation
    b. Objectifs
    c. Public Cible
    d. Programme
    e. Modalités
- **Admission**
  a. Procédure d'admission
  b. Documents requis
  c. Dates importantes
- **Actualités**
  a. À Propos
  b. Mission
  c. Équipe
  d. Partenaires
- **Contact**

En quoi est-elle importante pour l'entreprise, quels sont les principaux risques identifiés par le client sur cette application ?(vue avec le tuteur)

Elle est importante pour l'entreprise car c'est la première impression que le client à de l'entreprise. la plus part de leurs clients sont rencontrer sur internet et sur les réseaux sociaux donc le premier contact et la première impression se fait via leurs sites .

Pour quelles raisons des tests d'intrusion sont-ils souhaités ?(vue avec le tuteur)

Pour des test régulier.

Quelle est l'approche souhaitée pour les tests d'intrusion (boîte noire, grise ou blanche) ? (vue avec le tuteur)

Pour les test d'intrusion l'approche souhaitée est:
la boite noire: peu de connaissance de l'application.

Pour baliser les modalités techniques :

Quelle est son URL ou son IP ?

pour des raisons de securiter et de confidentialité ses informations ne seront pas communiquer dans ce rapport.

Est-ce qu'on regarde le serveur dans sa globalité ou juste l'application ?

Notamment, est-ce que les services autres que web sont dans le périmètre ?(vue avec le tuteur)

Non nous avons le droit de realiser des test que sur l'url ecole.ma
cette indication etait très importante le site avait d'autres site comme ecole.fr, ecole.cg, ecole.com

Est-ce que les tests seront effectués en environnement de production ou en recette ?(vue avec le tuteur)

les tests seront effectués en environnement de production.

Rapport de la semaine de stage chez smartgap 13/05-17/05

Toutes informations juger comme étant sensibles par le tuteur seront censuré dans ce rapport. veuillez vous rendre à la dernière page du rapport pour avoir la liste de tous les élément qui ont été censuré.

1er étape : se renseigne sur l'organisation et collecter les infos

Analyser tous les réseaux sociaux de l'entreprise, établir une arborescence presque complet du site:

- Accueil
- Nos Formations

a. Bachelor
b. Master
c. DBA

  - Présentation
  - Objectifs
  - Public Cible
  - Programme
  - Modalités

- Admission

a. Procédure d'admission
b. Documents requis
c. Dates importantes

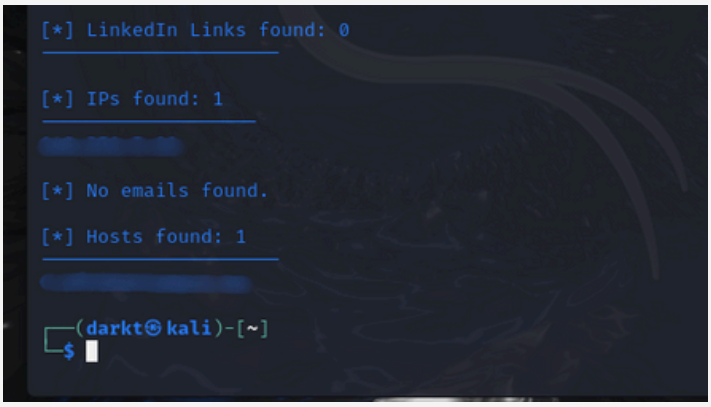
- Actualités

a. À Propos
b. Mission
c. Équipe
d. Partenaires

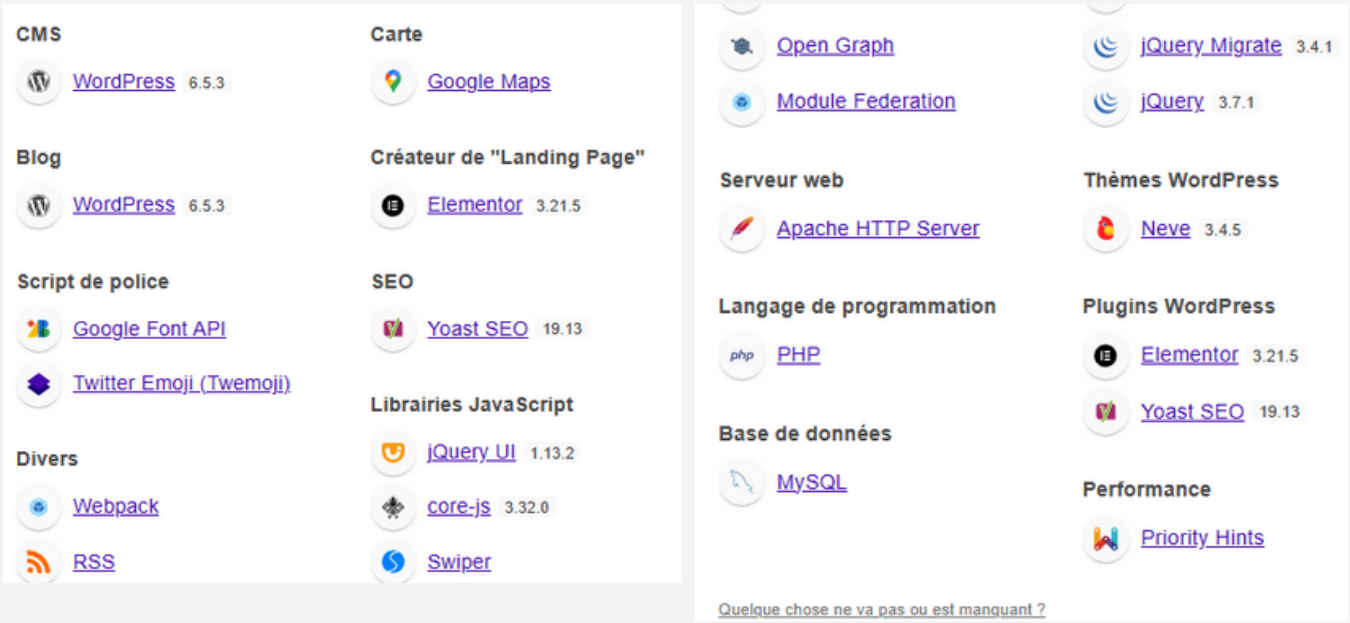
- Contact

Ensuite grâce à l'outil the Harvester j'ai pu avoir quellques données en plus qui n'etait pas fournis par le client. Comme l'adresse ip du site et un hostname liés au site.

resultats du The harvester:



Grace à l'outil wapalyzer j'ai pu savoir qu'elles langages de programmations, quel cms, quelles plugins ainsi que leurs versions ont été utiliser pour la création et l'administration:



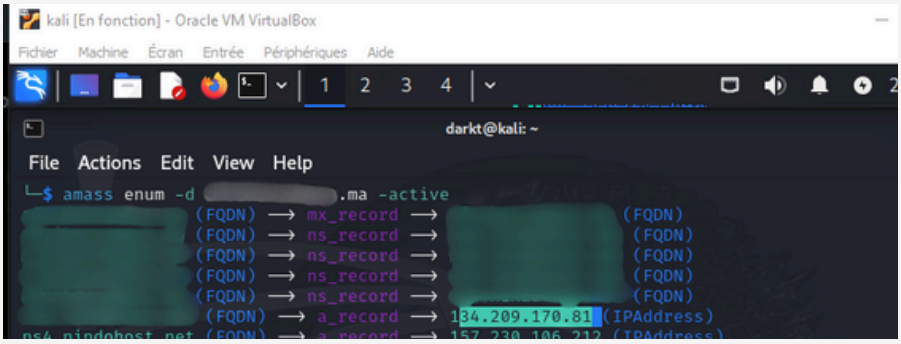
en parallèle de ça j'ai lancé d'autre test comme:

comme faire de l'écoute active grâce a l'outils enum sur kali linux.

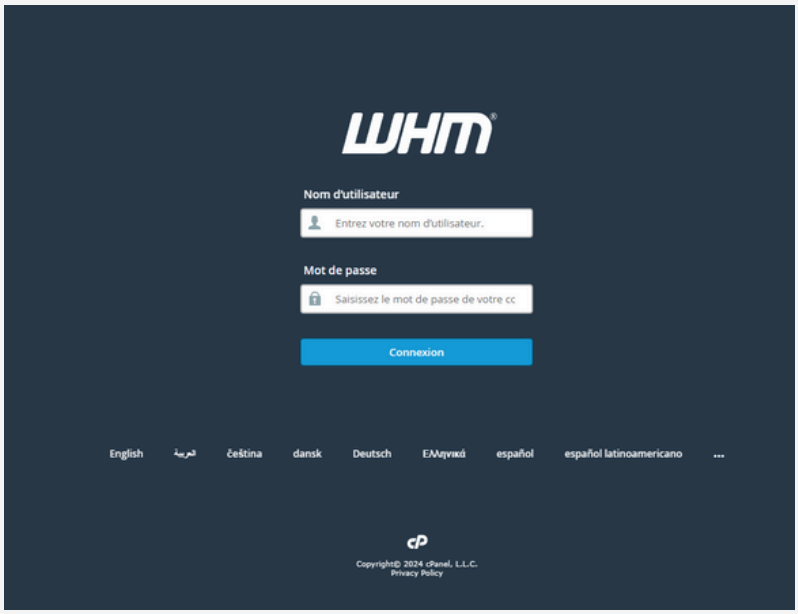
l'écoute active m'as conduit a un portail de connexion mais j'avais pas plus d'info sur ce que c'était après recherche et confirmation de mon tuteur je sais que c'est la admin du whm une plateforme qui permet de creer des cpanel aux client afin qu'il puisse accéder aux sites.

ensuite a l'aide de l'outil ffuf et un fichier texte contenant des mots clef j'ai essayer de trouver des pages cacher du site mais cela n'a rien donner

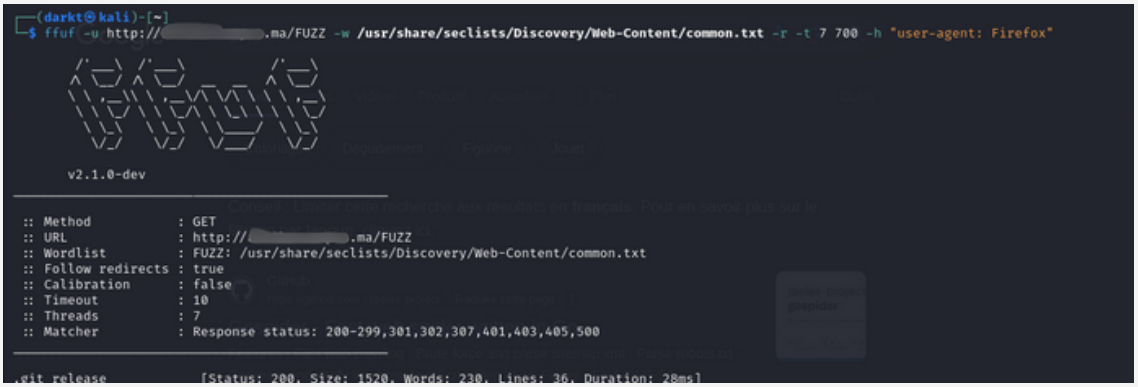
Screen de l’écoute active



page de conexion



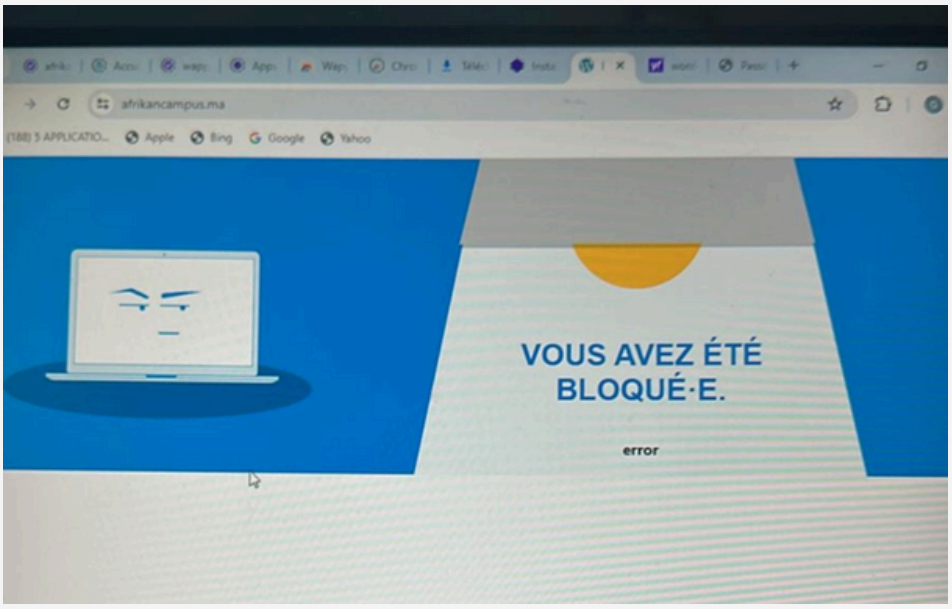
screen ffuf



Conclusion : cette semaine m’as permis d’avoir encore plus d’information

Rapport de la semaine de stage chez smartgap 20/05-24/05

Tout d'abord, cette semaine, en voulant accéder au site. J'ai eu que des pages blanches, car les mesures de sécurité du site avaient filtré mon adresse IP vu le nombre d'attaque que j'ai essayé de faire, notamment l'attaque par brut de force Mais cette mesure de sécurité a été facile à contourner. Juste en s'équipant d'un vpn.



Ensuite j'ai fait une récolte de données importantes en mode agressive En utilisant l'outil WPscan J'ai pu avoir lees vulnérabilités qui sont directement reliés à la version des plugins version du WordPress utilisée, mais avant même de m'attaquer à Ses vulnérabilité.

J'ai décidé de voir quels sont les utilisateurs admin du site Grace à l'outil WPscan et la commande

wpscan --url www.ecole.ma --enumerate u

j'ai pu avoir deux utilisateurs admin du site.
L'utilisateur admin et en plus l'utilisateur Meriem.

```
[i] User(s) Identified:

[+] admin
 | Found By: Wp Json Api (Aggressive Detection)
 |  - https://afrikancampus.ma/wp-json/wp/v2/users/?per_page=100&page=1
 | Confirmed By: Yoast Seo Author Sitemap (Aggressive Detection)
 |  - https://afrikancampus.ma/author-sitemap.xml

[+] miriem
 | Found By: Wp Json Api (Aggressive Detection)
 |  - https://afrikancampus.ma/wp-json/wp/v2/users/?per_page=100&page=1
```

J'ai donc décidé de faire des recherches sur la fameuse Meriem grâce à l'outil DirBuster j'ai pu retrouver un fichier caché du site dans lequel Il y avait écrit Miriam Hadoum

Donc j'avais non seulement son prénom, mais aussi son nom assez pour faire des recherches sur elle je me suis donc rendu sur Google, et j'ai tout simplement tapé Miriam hadoum Je suis tombé sur un profil LinkedIn.

J'ai donc pu identifier la fameuse Meriem comme la Web Designer du site Car celle-ci avait dans ses expériences professionnelles Smartgap En plus de ça, j'avais accès à son Facebook, ce qui m'a permis d'avoir sa date de naissance et ses centres d'intérêt avec toutes ces informations, j'ai pu établir une Word liste et donc procédé à une attaque par brut de force avec l'outils wpscan mais ça n'a rien donner….

J'ai encore été boquer j'ai donc dû trouver une solution pour me permette de contourner le filtrage d'adresse ip de Wordpress ce qui m'as pris beaucoup de temps mais j'ai reussi a en trouver une qui est : l'onion over vpn. Pour Plus de détails sur la solution Onion Over VPN : Mesures de Sécurité Ultimes | NordVPN
Et je vais donc pouvoir continuer tous les tests sans risque d'être bloqué.

Rapport de la semaine de stage chez smartgap 27/05-31/05

cet semaine j'ai commencer a envoyé les premières version du rapport, des matrices, et des plans d'action

G.Tony LAMBERT-TATHY Fait le : 24/05/2024

Rapport de vulnérabilité

final

Ce document a pour complément une matrice de risque dans laquelle Il y a les vulnérabilitées classées de la plus importante à la moins importante, mais aussi le niveau de risque ainsi que le nombres total de vulnérabilités trouvées.

Url du site

Adresse Ip du site :

Cms utiliser : Worpress

Version de cms : WordPress version 6.2 (Non sécurisée, sortie le 29-03-2023)

Theme wordpress : Neve

ID utilisateur du site : admin, meriem

Espace d'authentification : .ma/wp-login/

Server / version : imunify360-webshield/1.21

Plugins/version :Elementor/3.13.2 ; Yoast SEO/19.13 ; iThemes Security

Vulnérabilité liée à la version du cms

Outils utilisées : Wpscan, wappalyzer sur kali, ainsi que les différentes recherches faite manuellement.

```
Requête envoyées :  wpscan --url https://  .ma/ --enumerate vp,vt
```

SSRF aveugle non authentifié via DNS Rebinding

Description de la vulnérabilité :

Un SSRF (Server-Side Request Forgery) aveugle est un type d'attaque où un attaquant manipule un serveur pour qu'il envoie des requêtes HTTP à des ressources internes ou externes, mais sans voir directement les réponses à ces requêtes. L'attaquant ne reçoit pas directement le contenu de la réponse, d'où le terme "aveugle". Cependant, il peut souvent inférer des informations basées sur le comportement du serveur ou les temps de réponse.

Impact potentiel : Exécution de code arbitraire, compromission complète du serveur.

- **Impact : Élevé** car l'accès non autorisé à des systèmes internes peut entraîner des fuites de données sensibles et des compromissions importantes.

- **Probabilité : Moyen** car cette attaque nécessite un certain degré de préparation et d'accès initial.
- **Note de la vulnérabilité :** 8/10.

Solutions proposées :

1. **Validation stricte des entrées** : Valider et assainir les URL fournies par les utilisateurs pour s'assurer qu'elles ne pointent pas vers des adresses internes ou non autorisées.
2. **Segmentation du réseau** : Utiliser une segmentation du réseau pour limiter l'accès des serveurs frontaux aux services internes critiques.
3. **Contrôle des accès** : Implémenter des politiques de contrôle des accès pour restreindre quelles requêtes les serveurs peuvent effectuer, en particulier vers des adresses internes.
4. **Journaux et alertes** : Surveiller les journaux et mettre en place des alertes pour détecter des activités suspectes, comme des requêtes sortantes inhabituelles.

Mise à jour potentiel qui permet d'éviter ce genre d'attaque :

Corrigé dans : 6.2.1

Référence :

- https://wpscan.com/vulnerability/c8814e6e-78b3-4f63-a1d3-6906a84c1f11
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-3590
- https://blog.sonarsource.com/wordpress-core-unauthenticated-blind-ssrf

Déni de service via empoisonnement du cache

Description de la vulnérabilité :

Le déni de service (DoS) via empoisonnement du cache est une attaque où un attaquant manipule les données stockées dans le cache d'un serveur pour provoquer une interruption ou une dégradation des services. Cette attaque peut rendre le site web inaccessible ou ralentir considérablement ses performances.

Conséquences :

- **Interruption du service** : Les utilisateurs peuvent recevoir des pages web incorrectes ou ne pas pouvoir accéder aux ressources nécessaires, entraînant une interruption du service.
- **Dégradation des performances** : La réponse incorrecte du serveur peut entraîner un comportement inattendu, surchargeant le système et ralentissant ses performances.
- **Impact sur l'expérience utilisateur** : Les utilisateurs peuvent perdre confiance dans le site s'ils rencontrent des erreurs fréquentes ou des performances médiocres.

Impact potentiel : Indisponibilité du service.

- **Impact : Élevé** car l'indisponibilité du service peut gravement perturber les opérations et affecter les utilisateurs.
- **Probabilité : Haut** car cette attaque peut être facilement menée si le système de cache n'est pas correctement sécurisé.
- **Note de la vulnérabilité :** 8/10.

Solutions proposées :

1. **Validation des entrées** : S'assurer que toutes les entrées utilisateur sont soigneusement validées et nettoyées avant d'être mises en cache.
2. **Configuration du cache** : Configurer les systèmes de cache pour qu'ils ne stockent pas de contenu basé sur des entrées non fiables ou des requêtes non sécurisées.
3. **Utilisation de jetons CSRF** : Implémenter des jetons CSRF (Cross-Site Request Forgery) pour valider les requêtes et éviter les manipulations par des tiers non autorisés.

Mise à jour potentiel qui permet d'éviter ce genre d'attaque :

Corrigé dans : 6.2.3

Référence :

https://wpscan.com/vulnerability/6d80e09d-34d5-4fda-81cb-e703d0e56e4f

https://wordpress.org/news/2023/10/wordpress-6-3-2-maintenance-and-security-release/

Traversée de répertoires via fichiers de traduction

Description de la vulnérabilité :

La traversée de répertoires via les fichiers de traduction, aussi connue sous le terme anglais "Directory Traversal via Translation Files", est une vulnérabilité de sécurité où un attaquant peut exploiter des fichiers de traduction pour accéder à des répertoires et fichiers sensibles sur le serveur web. Cette attaque permet souvent à l'attaquant de lire, modifier ou exécuter des fichiers en dehors du répertoire prévu par l'application.

Conséquences :

- **Vol de données sensibles** : L'attaquant peut lire des fichiers contenant des informations sensibles, comme des configurations de base de données, des mots de passe, ou d'autres informations critiques.
- **Compromission du serveur** : Accéder à des fichiers sensibles peut permettre à l'attaquant de compromettre complètement le serveur, d'injecter du code malveillant, ou de prendre le contrôle de l'application.

Impact potentiel : Vol de données sensibles, compromission du système.

- **Impact : Élevé** car la traversée de répertoires peut permettre l'accès à des fichiers sensibles et critiques.
- **Probabilité : Moyen** car l'exploitation requiert souvent des conditions spécifiques et une connaissance préalable du système cible.
- **Note de la vulnérabilité :** 7/10.

Solutions proposées :

1. **Validation des entrées** : Toujours valider et assainir les chemins de fichiers fournis par l'utilisateur pour éviter les caractères et séquences spéciaux qui peuvent conduire à une traversée de répertoires (.., /, etc.).
2. **Utilisation de bibliothèques sécurisées** : Utiliser des fonctions de manipulation de fichiers qui empêchent automatiquement les traversées de répertoires.

3. **Restrictions de permissions** : Limiter les permissions des fichiers et répertoires accessibles par l'application web pour minimiser les dommages en cas de compromission.
4. **Mises à jour régulières** : Maintenir à jour l'application et ses composants pour bénéficier des correctifs de sécurité.

Mise à jour potentiel qui permet d'éviter ce genre d'attaque :

Corrigé dans : 6.2.1

Référence :

- https://wpscan.com/vulnerability/2999613a-b8c8-4ec0-9164-5dfe63adf6e6
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-2745
- https://wordpress.org/news/2023/05/wordpress-6-2-1-maintenance-security-release/

Exécution de shortcodes dans les données générées par les utilisateurs

Description de la vulnérabilité :

L'exécution de shortcodes dans les données générées par les utilisateurs est une vulnérabilité où un attaquant peut insérer des shortcodes malveillants dans du contenu généré par les utilisateurs, tels que des commentaires, des publications ou des profils. Les shortcodes sont des balises spécifiques à WordPress qui permettent d'ajouter des fonctionnalités dynamiques (comme des galeries, des vidéos, des formulaires, etc.) sans avoir à écrire de code HTML ou PHP complexe.

Conséquences :

- **Injection de contenu malveillant** : Un attaquant peut utiliser des shortcodes pour insérer du contenu malveillant, comme des scripts JavaScript, des iframes, ou des redirections vers des sites malveillants.
- **Prise de contrôle du site** : Dans certains cas, l'exécution de shortcodes malveillants peut permettre à un attaquant de prendre le contrôle du site, d'accéder à des données sensibles, ou de modifier le comportement du site.
- **Dégradation de l'expérience utilisateur** : Les utilisateurs peuvent être exposés à des contenus inappropriés ou nuisibles, ce qui peut affecter la crédibilité et la réputation du site.

Impact potentiel : Exécution de code malveillant, vol de données.

- **Impact : Élevé** car les shortcodes malveillants peuvent exécuter du code indésirable.
- **Probabilité : Moyen** car bien que cette attaque soit possible, elle nécessite des permissions spécifiques pour insérer des shortcodes.
- **Note de la vulnérabilité :** 7/10.

Solutions proposées :

- **Validation et filtrage des entrées** : Assurer que les contenus soumis par les utilisateurs sont soigneusement validés et filtrés pour empêcher l'inclusion de shortcodes non autorisés.
- **Limiter les permissions** : Restreindre l'utilisation de shortcodes aux rôles utilisateurs de confiance (comme les administrateurs et les éditeurs) et interdire leur utilisation par les contributeurs et les abonnés.
- **Utiliser des plugins de sécurité** : Installer et configurer des plugins de sécurité qui peuvent détecter et bloquer les shortcodes malveillants dans les contenus générés par les utilisateurs.

Mise à jour potentiel qui permet d'éviter ce genre d'attaque :

Corrigé dans : 6.2.2

Référence :

- https://wpscan.com/vulnerability/ef289d46-ea83-4fa5-b003-0352c690fd89
- https://wordpress.org/news/2023/05/wordpress-6-2-1-maintenance-security-release/
- https://wordpress.org/news/2023/05/wordpress-6-2-2-security-release/

XSS réfléchi via demandes de mot de passe d'application

Description de la vulnérabilité :

Le XSS réfléchi (ou Reflected Cross-Site Scripting) via les demandes de mot de passe d'application est une vulnérabilité de sécurité où un attaquant peut injecter du code malveillant dans les paramètres d'une requête web, qui est ensuite reflétée dans la réponse du serveur et exécutée par le navigateur de la victime. Cette attaque spécifique se produit dans le contexte des demandes de mot de passe d'application, qui sont des fonctionnalités de WordPress permettant aux utilisateurs de générer des mots de passe pour des applications tierces accédant à leur compte.

Conséquences :

- **Exécution de scripts malveillants** : Le navigateur de la victime exécute le code injecté, ce qui peut entraîner des actions non désirées comme le vol de cookies de session, l'enregistrement de frappes de touches ou la redirection vers des sites malveillants.
- **Accès non autorisé** : L'attaquant peut obtenir des informations sensibles ou même compromettre le compte de la victime.
- **Manipulation du contenu** : Le site peut afficher du contenu falsifié ou nuisible, portant atteinte à la crédibilité du site et à la confiance des utilisateurs.

Impact potentiel : Vol de session, compromission des comptes.

- **Impact : Élevé** car le XSS réfléchi peut compromettre les sessions et les comptes des utilisateurs.
- **Probabilité : Moyen** car l'exploitation de XSS réfléchi est souvent dépendante des pratiques de validation des entrées
- **Note de la vulnérabilité :** 7/10.

Solutions proposées :

1. **Validation et assainissement** : Toujours valider et assainir les entrées des utilisateurs. Utiliser des fonctions sécurisées pour échapper les données dynamiques avant de les inclure dans la sortie HTML.
2. **Utilisation de Content Security Policy (CSP)** : Implémenter une politique de sécurité du contenu qui restreint les sources de scripts pouvant être exécutés dans le navigateur.
3. **Audit et test de sécurité** : Effectuer régulièrement des audits de sécurité et des tests de pénétration pour identifier et corriger les vulnérabilités XSS.

Mise à jour potentiel qui permet d'éviter ce genre d'attaque :

Corrigé dans : 6.2.3

Référence :

- https://wpscan.com/vulnerability/da1419cc-d821-42d6-b648-bdb3c70d91f2
- https://wordpress.org/news/2023/10/wordpress-6-3-2-maintenance-and-security-release/
-

Injection de contenu Contributor+

Description de la vulnérabilité :

L'injection de contenu Contributor+ (Contributor+ Content Injection) est une vulnérabilité de sécurité où un utilisateur ayant un rôle de contributeur ou supérieur sur un site WordPress peut injecter du contenu malveillant dans des articles ou des pages. Cette vulnérabilité permet à un attaquant ayant un accès limité de contourner certaines restrictions et d'insérer du code ou des données non autorisés dans le contenu du site.

Conséquences :

- **Exécution de code malveillant** : L'injection de scripts JavaScript peut permettre l'exécution de code malveillant dans le navigateur des utilisateurs visitant la page, pouvant conduire à des attaques de type Cross-Site Scripting (XSS).
- **Modification du contenu** : L'attaquant peut manipuler le contenu affiché sur le site, ce qui peut altérer la confiance des utilisateurs et endommager la crédibilité du site.
- **Escalade des privilèges** : Dans certains cas, l'injection de contenu peut être utilisée pour tenter d'obtenir des privilèges plus élevés ou un accès non autorisé à d'autres parties du site.

Impact potentiel : Exécution de code malveillant, défiguration de site.

- **Impact : Élevé** car les contributions non validées peuvent mener à des attaques réussies.
- **Probabilité : Moyen** car l'exploitation nécessite souvent des permissions de contributeur.
- **Note de la vulnérabilité :** 7/10.

Solutions proposées :

1. **Validation et assainissement des entrées** : Toujours valider et nettoyer les données soumises par les utilisateurs pour empêcher l'injection de code non autorisé. Utiliser des fonctions de validation fournies par WordPress ou des bibliothèques tierces fiables.
2. **Révision stricte du contenu** : S'assurer que le contenu soumis par les contributeurs est soigneusement revu par des éditeurs ou des administrateurs avant d'être publié.
3. **Utilisation de plugins de sécurité** : Installer des plugins de sécurité qui peuvent aider à détecter et prévenir les tentatives d'injection de contenu malveillant.

Mise à jour potentiel qui permet d'éviter ce genre d'attaque :

Corrigé dans : 6.2.1

Référence :

- https://wpscan.com/vulnerability/1527ebdb-18bc-4f9d-9c20-8d729a628670
- https://wordpress.org/news/2023/10/wordpress-6-3-2-maintenance-and-security-release/

La mise à jour de l'image miniature via CSRF

Description de la vulnérabilité :

La mise à jour de l'image miniature via CSRF (Cross-Site Request Forgery) est une vulnérabilité où un attaquant peut tromper un utilisateur authentifié pour qu'il effectue une action non désirée sur un site web auquel il est connecté, sans le consentement de cet utilisateur. Dans ce contexte, cela signifie que l'attaquant peut manipuler l'utilisateur pour mettre à jour ou changer l'image miniature (thumbnail) d'un contenu sur un site web.

Conséquences :

- **Manipulation de contenu** : L'attaquant peut remplacer les images miniatures par des images malveillantes ou inappropriées.
- **Dégradation de la confiance** : Cela peut entraîner une perte de confiance des utilisateurs envers le site web s'ils voient des images non appropriées ou malveillantes.
- **Exploitation ultérieure** : Les images miniatures modifiées pourraient contenir des liens ou des informations exploitables pour des attaques supplémentaires.

Impact potentiel : Changement non autorisé de contenu.

- **Impact : Moyen** car les changements non autorisés de contenu peuvent altérer l'apparence et la fonctionnalité.
- **Probabilité : Bas** car les attaques CSRF nécessitent des conditions spécifiques et peuvent être mitigées avec des jetons.
- **Note de la vulnérabilité :** 4/10.

Solutions proposées :

1. **Jetons CSRF (CSRF tokens)** : Utiliser des jetons uniques et imprévisibles pour chaque requête soumise par les utilisateurs authentifiés afin de vérifier la légitimité de la requête.
2. **Vérification des référents** : S'assurer que les requêtes proviennent du même domaine que celui de l'application pour empêcher les requêtes malveillantes provenant de sites externes.
3. **Limitation des méthodes HTTP** : Restreindre les actions sensibles aux méthodes HTTP sécurisées (comme POST) et éviter l'utilisation de GET pour les actions ayant des effets secondaires.

Remarques : grâce a cette vulnérabilité j' ai pu réaliser de nombreuse attaque sur le site car il y avait aucun contrôle sur les requêtés que j'envoyée vous aurais le temps de le constater dans la recherche de vulnérabilité en mode attaque

Mise à jour potentiel qui permet d'éviter ce genre d'attaque :

Corrigé dans : 6.2.1

Référence :

- https://wpscan.com/vulnerability/3b574451-2852-4789-bc19-d5cc39948db5
- https://wordpress.org/news/2023/05/wordpress-6-2-1-maintenance-security-release/

Vulnérabilité liée à la version des plugins

a) Yoast SEO

XSS Stocké Authentifié (Seo Manager+)

Description de la vulnérabilité :

XSS Stocké Authentifié (Seo Manager+) est une vulnérabilité de sécurité dans les systèmes web, en particulier dans le plugin Yoast SEO pour WordPress, qui permet à un utilisateur authentifié avec des privilèges spécifiques (comme un gestionnaire SEO) d'injecter du code malveillant dans le site.

Conséquences :

- **Injection de JavaScript** : L'attaquant peut injecter du JavaScript malveillant qui sera exécuté par le navigateur de chaque utilisateur accédant à la page vulnérable.
- **Exécution de commandes malveillantes** : Le code injecté peut voler des informations sensibles (comme des cookies de session), rediriger les utilisateurs vers des sites malveillants, ou effectuer des actions en leur nom.

Impact potentiel : Vol de session, défiguration de site, vol de données.

- **Impact : Élevé** car un XSS stocké peut permettre l'exécution de scripts malveillants persistants, compromettant les données et les sessions des utilisateurs.
- **Probabilité : Haut** car les entrées utilisateur sont souvent mal validées, rendant cette attaque relativement facile à exécuter.
- **Note de la vulnérabilité :** 8/10.

Solutions proposées :

1. **Sanitisation des entrées** : S'assurer que toutes les entrées utilisateur sont correctement filtrées et échappées avant d'être stockées ou affichées.
2. **Mises à jour régulières** : Garder les plugins et les systèmes à jour avec les dernières versions pour bénéficier des correctifs de sécurité.
3. **Contrôles d'accès** : Limiter les privilèges des utilisateurs pour minimiser les risques d'exploitation.

Mise à jour potentiel qui permet d'éviter ce genre d'attaque :

Corrigé dans : 21.1

Référence :

- WPScan

- CVE-2023-40680
- Wordfence

Vulnérabilité liées aux audits de pénétration réalisées

Outils utilisées : ZAP et Burpsuite sur kali, ainsi que les différentes recherches faite manuellement.

Injection de modèles côté serveur (aveugle)

Description de la vulnérabilité

Lorsque l'entrée utilisateur est insérée dans le modèle au lieu d'être utilisée comme argument dans le rendu, elle est évaluée par le moteur de modèle. Selon le moteur de templates, cela peut conduire à l'exécution de code à distance.

Impact potentiel : Exécution de code arbitraire, compromission complète du serveur.

- **Impact : Très Élevé parce que cette vulnérabilité peut entraîner une compromission complète du serveur, permettant aux attaquants d'exécuter des commandes arbitraires.**
- **Probabilité : Moyen car bien que les conséquences soient graves, l'exploitation nécessite souvent des conditions spécifiques.**
- **Note de risque** : 9/10

Requête envoyées

```
GET
http://[redacted].ma/z0f76a1d14fd21a8fb5fd0d03e0fdc3d3cedae52f?wsidch
k=%7B%7Brange.constructor%28%22return+eval%28%5C%22global.process.mainMo
dule.require%28%27child_process%27%29.execSync%28%27sleep+6%27%29.toStri
ng%28%29%5C%22%29%22%29%28%29%7D%7D HTTP/1.1
```

Explication de la requête

Le lien fourni contient un script conçu pour exploiter une vulnérabilité potentielle sur les serveurs web. Voici une explication de ce script :
{{range.constructor("return eval(\"global.process.mainModule.require('child_process').execSync('sleep 6').toString()\")")()}}

Détails :

- range.constructor(...): Utilise la propriété constructor pour créer une fonction à partir d'une chaîne de caractères.
- eval(...): Exécute cette chaîne comme du code.
- global.process.mainModule.require('child_process').execSync('sleep 6'): Exécute la commande sleep 6 sur le serveur, le mettant en pause pendant 6 secondes.

Cela peut être dangereux car cela permet l'exécution de commandes arbitraires sur le serveur.

Mesures recommandées : Valider et nettoyer les données des templates, utiliser des modèles sécurisés.

Ref : https://portswigger.net/blog/server-side-template-injection

Présence potentielle d'informations personnellement identifiable

Description de la vulnérabilité

La réponse contient des renseignements personnels identifiables, comme le numéro CC, le numéro SSN et des données sensibles semblables.

Impact potentiel : Vol d'identité, atteinte à la vie privée.

- **Impact : Élevé** car la divulgation de PII peut entraîner des violations graves de la vie privée et des vols d'identité.
- **Probabilité : Haut** car les données PII sont fréquemment mal protégées.
- **Note de risque** : 8/10

Requête envoyées

```
GET https://[redacted].ma/wp-content/plugins/elementor-
pro/assets/js/elements-handlers.min.js?ver=3.9.1 HTTP/1.1
host: [redacted].ma
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Referer: https://[redacted].ma/
Connection: keep-alive
Cookie:
wssplashuid=19ed3867aa0c81e1686f6186ab325cf48f458e97.1716556929.1
Sec-Fetch-Dest: script
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
```

Requête reçus

https://[redacted]ma/wp-content/plugins/elementor-pro/assets/js/elements-handlers.min.js?ver=3.9.1

Info perso potentiels : [redacted]

Solution : vérifiez la réponse pour la présence potentielle d'informations personnellement identifiables (PII), assurez-vous que rien de sensible n'est divulgué par l'application.

Tag de l'alerte

OWASP_2021_A04
OWASP_2017_A03

Absence de Jetons Anti-CSRF

Description de la vulnérabilité

Aucun jetons Anti-CSRF n'ont été trouvés dans un formulaire HTML.
La contrefaçon de requête intersites (Cross Site Request Forgery - CSRF) est une attaque qui consiste à forcer une victime à envoyer une requête HTTP vers une destination cible, sans qu'elle n'en aie ni connaissance ni intention, afin d'effectuer une action en se faisant passer pour la victime. La cause originelle est que les fonctionnalités de l'application sont appelées à l'aide d'URL ou d'actions de formulaires prévisibles et reproductibles. La nature de l'attaque est que le CSRF exploite la confiance qu'un site internet accorde à un utilisateur. En revanche, le cross-site scripting (XSS) exploite la confiance que l'utilisateur porte à un site internet. Comme XSS, les attaques CSRF ne sont pas nécessairement multi-sites, mais elles peuvent l'être. La contrefaçon de requête intersite est également connue sous les noms CSRF, XSRF, attaque en un clic (one-click attack), session riding, confused deputy et sea surf.

Les attaques CSRF sont efficaces dans de nombreuses situations, notamment:

- quand la victime a une session active sur le site cible.
- quand la victime est authentifiée via HTTP auth sur le site cible.
- quand la victime est sur le même réseau local que le site cible.

CSRF a d'abord été utilisée pour effectuer une action contre un site cible en utilisant les privilèges de la victime, mais des techniques récentes permettent d'avoir accès à des renseignements en accédant à la réponse. Le risque de divulgation d'informations est considérablement augmenté lorsque le site cible est vulnérable aux XSS, parce que XSS peut être utilisé comme une plateforme pour CSRF, permettant à l'attaque d'opérer dans les limites de la politique de même origine.

Impact potentiel : Prise de contrôle de session, actions non autorisées.

- **Impact : Élevé** car les attaques CSRF peuvent permettre aux attaquants d'effectuer des actions non autorisées au nom des utilisateurs.
- **Probabilité : Haut** car l'absence de jetons anti-CSRF est une vulnérabilité courante et facilement exploitable.
- **Note de risque** : 8/10

Solutions :

Phase: Architecture et Design
Utilisez une librairie ou un framework approuvé qui ne permet pas cette vulnérabilité, ou qui contient des fonctionnalités permettant d'éviter plus facilement cette vulnérabilité.
Utilisez par exemple des librairies anti-CSRF telles que CSRFGuard de l'OWASP.

Phase: Implémentation
Assurez-vous que votre application soit exempte de problèmes de cross-site scripting, parce que la plupart des défenses contre le CSRF peuvent être contournées en utilisant des scripts contrôlés par le pirate.

Phase: Architecture et Design
Générez une valeur à usage unique pour chaque formulaire, placez-la dans le formulaire et vérifiez-la à la réception du formulaire. Assurez-vous que cette valeur unique ne soit pas prévisible (CWE-330).
Notez que ceci peut aussi être contourné en utilisant XSS.

Identifiez les opérations particulièrement dangereuses. Quand l'utilisateur exécute une opération dangereuse, envoyez une requête de confirmation distincte pour vérifier que l'utilisateur veut effectivement effectuer cette opération.
Notez que ceci peut aussi être contourné en utilisant XSS.

Utilisez la librairie de gestion de session ESAPI.
Cette librairie comprend un composant pour le contrôle de CSRF.

N'utilisez pas la méthode GET pour les requêtes entraînant un changement d'état.

Phase: Implémentation

Vérifiez l'en-tête HTTP Referer pour voir si la requête provient d'une page attendue. Ceci pourrait toutefois restreindre la fonctionnalité de l'application, car les utilisateurs ou les serveurs proxy pourraient avoir désactivé le renvoi du HTTP Referer pour des raisons de confidentialité.

Requêtes envoyées

```
Get
http://[redacted].ma/z0f76a1d14fd21a8fb5fd0d03e0fdc3d3cedae52f?wsidchk=%7B%7Brange.c
onstructor%28%22return+eval%28%5C%22global.process.mainModule.require%28%27child_proces
s%27%29.execSync%28%27sleep+200%27%29.toString%28%29%5C%22%29%22%29%28%29%7D%7D
HTTP/1.1
Get
http://[redacted]ma/z0f76a1d14fd21a8fb5fd0d03e0fdc3d3cedae52f?wsidchk=%7B%7Brange.c
onstructor%28%22return+eval%28%5C%22global.process.mainModule.require%28%27child_proces
s%27%29.execSync%28%27sleep+6%27%29.toString%28%29%5C%22%29%22%29%28%29%7D%7D HTTP/1.1
GET
http://[redacted].ma/z0f76a1d14fd21a8fb5fd0d03e0fdc3d3cedae52f?wsidchk=%7B%7Brange.c
onstructor(%22return+eval(%5C%22global.process.mainModule.require('child_process').exec
Sync('ls').toString()%5C%22)%22)()%7D%7D HTTP/1.1
GET https://[redacted].ma/contact/
```

Ref :

http://projects.webappsec.org/Cross-Site-Request-Forgery

http://cwe.mitre.org/data/definitions/352.html

Tag de l'alerte

OWASP_2021_A01
WSTG-v42-SESS-05
OWASP_2017_A05

En-tête de politique de sécurité du contenu (CSP) non défini

Description de la vulnérabilité

La politique de sécurité du contenu (CSP) est une couche de sécurité supplémentaire qui permet de détecter et d'atténuer certains types d'attaques, notamment les attaques de type Cross Site Scripting (XSS) et par injection de données. Ces attaques sont utilisées pour tout, du vol de données à la dégradation de sites ou à la distribution de logiciels malveillants. CSP fournit un ensemble d'en-têtes HTTP standard qui permettent aux propriétaires de sites Web de déclarer les sources de contenu approuvées que les navigateurs devraient être autorisés à charger sur cette page. Les types couverts sont JavaScript, CSS, les cadres HTML, les polices, les images et les objets intégrables tels que les applets Java, ActiveX, fichiers audio et vidéo.

Impact potentiel : Vulnérabilité aux attaques XSS et injection de contenu.

- **Impact : Élevé** car l'absence de CSP rend les applications vulnérables aux attaques XSS et à l'injection de contenu.
- **Probabilité : Moyen** car la mise en œuvre de CSP est souvent négligée mais réalisable.
- **Note de risque** : 7/10

Solution : Assurez-vous que votre serveur Web, serveur d'applications, équilibreur de charge, etc. est configuré pour définir l'en-tête Content-Security-Policy.

Requêtes envoyées

```
GET https://[redacted].ma/
GET https://[redacted].ma/nos-formations/
```

Ref : https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy
https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
http://www.w3.org/TR/CSP/
http://w3c.github.io/webappsec/specs/content-security-policy/csp-specification.dev.html
http://www.html5rocks.com/en/tutorials/security/content-security-policy/
http://caniuse.com/#feat=contentsecuritypolicy
http://content-security-policy.com/

Tag de l'alerte

OWASP_2021_A05

Métadonnée du cloud potentiellement vulnérable

Description de la vulnérabilité

De telles erreurs pourraient être utilisées pour contourner les schémas de listes autorisées en introduisant des entrées dangereuses après leur vérification.

L'attaque de métadonnées cloud tente d'abuser d'un serveur NGINX mal configuré afin d'accéder aux métadonnées d'instance gérées par des fournisseurs de services cloud tels qu'AWS, GCP et Azure.

Tous ces fournisseurs fournissent des métadonnées via une adresse IP interne non routable '169.254.169.254' - cela peut être exposé par des serveurs NGINX mal configurés et accessible en utilisant cette adresse IP dans le champ d'en-tête Host.

Impact potentiel : Fuite d'informations sensibles, compromission du système.

- **Impact : Élevé** car les métadonnées du cloud peuvent contenir des informations critiques.
- **Probabilité : Bas** car l'exploitation nécessite souvent un accès initial au cloud et une connaissance spécifique des configurations.
- **Note de risque** : 6/10

Requête envoyées

```
GET http://[redacted].ma/latest/meta-data/ HTTP/1.1
host: 169.254.169.254
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36 OPR/102.0.0.0
pragma: no-cache
cache-control: no-cache
```

Réponse reçu s dans l'entête

```
HTTP/1.1 200 OK
Date: Fri, 24 May 2024 12:41:37 GMT
Content-Length: 1500
Connection: keep-alive
Cache-Control: no-cache, no-store, must-revalidate, max-age=0
Cache-Control: no-store, max-age=0
Server: imunify360-webshield/1.21
```

Ref : https://www.nginx.com/blog/trust-no-one-perils-of-trusting-user-input/

Tag de l'alerte

OWASP_2021_A05
OWASP_2017_A06

En-tête Anti-clickjacking manquante

Description de la vulnérabilité

La réponse n'inclut ni Content-Security-Policy avec la directive 'frame-ancestors' ni X-Frame-Options pour se protéger contre les attaques 'ClickJacking'.

Impact potentiel : Fuite d'informations sensibles, compromission du système.

- **Impact : Élevé** car les attaques de clickjacking peuvent permettre aux attaquants d'exécuter des actions non autorisées.
- **Probabilité : Bas** car la mise en œuvre des en-têtes anti-clickjacking est simple et souvent négligée.
- **Note de risque** : 6/10

Requête envoyée

GET
https://[redacted].ma/
https://[redacted].ma/nos-formations/

Mesures recommandées : Implémenter des en-têtes anti-clickjacking.

Ref : https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

Tag de l'alerte

OWASP_2021_A05
WSTG-v42-CLNT-09
OWASP_2017_A06

En-tête strict-transport-security non défini

Description de la vulnérabilité

HTTP Strict Transport Security (HSTS) est un mécanisme de politique de sécurité Web par lequel un serveur Web déclare que les agents utilisateurs conformes (tels qu'un navigateur Web) doivent interagir avec lui en utilisant uniquement des connexions HTTPS sécurisées (c'est-à-dire HTTP superposé à TLS/SSL). HSTS est un protocole de suivi des normes IETF et est spécifié dans la RFC 6797.

Impact potentiel : Vulnérabilité aux attaques de type "man-in-the-middle"

- **Impact : Moyen** car l'absence de HSTS augmente la probabilité d'attaques MITM.
- **Probabilité : Moyen** car l'implémentation de HSTS est courante mais peut être oubliée.
- **Note de risque** : 5/10

Mesures recommandées : Implémenter l'en-tête HSTS.
Ref :
https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html
https://owasp.org/www-community/Security_Headers
http://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security
Tag de l'alerte
OWASP_2021_A05
OWASP_2017_A06

Cookie sans indicateur sécurisé

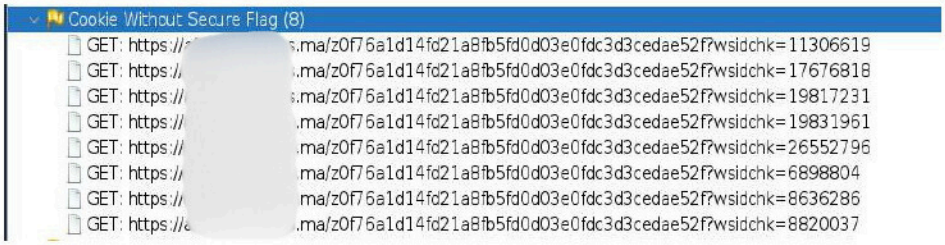
Description de la vulnérabilité

Un cookie a été défini sans l'indicateur sécurisé, ce qui signifie que le cookie est accessible via des connexions non cryptées.

Impact potentiel : Exposition des cookies aux attaques MITM.

- **Impact : Moyen** car l'absence de HSTS augmente la probabilité d'attaques MITM.
- **Probabilité : Moyen** car l'implémentation de HSTS est courante mais peut être oubliée.
- **Note de risque** : 4/10

Requêtes envoyées



Requête reçus

```
HTTP/1.1 302 Moved Temporarily
Date: Fri, 24 May 2024 12:22:09 GMT
Content-Length: 0
Connection: keep-alive
```

```
X-Forwarded-For: 185.220.100.244
X-Real-IP: 185.220.100.244
X-Remote-IP: 185.220.100.244
Location: https://[redacted].ma
Set-Cookie:
wssplashuid=19ed3867aa0c81e1686f6186ab325cf48f458e97.1716556929.1;
Path=/; Domain=afrikancampus.ma; Max-Age=2592000; HttpOnly; SameSite=Lax
```

Solution : Chaque fois qu'un cookie contient des informations sensibles ou est un jeton de session, il doit toujours être transmis via un canal crypté. Assurez-vous que l'indicateur sécurisé est défini pour les cookies contenant de telles informations sensibles.

Ref : https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes.html

Tag de l'alerte

OWASP_2021_A05
WSTG-v42-SESS-02
OWASP_2017_A06

En-tête X-Content-Type-Options manquant

Description de la vulnérabilité

L'en-tête Anti-MIME-Sniffing X-Content-Type-Options n'était pas défini sur 'nosniff'. Cela permet aux versions antérieures d'Internet Explorer et de Chrome d'effectuer un reniflage MIME sur le corps de la réponse, ce qui peut entraîner l'interprétation et l'affichage du corps de la réponse comme un type de contenu autre que le type de contenu déclaré. Les versions actuelles (début 2014) et héritées de Firefox utiliseront le type de contenu déclaré (s'il y en a un), plutôt que d'effectuer un reniflage MIME.

Impact potentiel : Exécution de contenu non prévu.

- **Impact : Moyen** car les attaques MIME sniffing peuvent permettre l'exécution de contenu malveillant.
- **Probabilité : Moyen** car l'en-tête est simple à implémenter mais souvent négligé. Note de **risque** : 5/10

Requêtes envoyées

```
GET http://[redacted].ma HTTP/1.1
```

Requête reçus

```
HTTP/1.1 200 OK
Date: Fri, 24 May 2024 12:21:42 GMT
Content-Length: 1446
Connection: keep-alive
```

```
Cache-Control: no-cache, no-store, must-revalidate, max-age=0
Cache-Control: no-store, max-age=0
Server: imunify360-webshield/1.21
```

Solution : Assurez-vous que l'application/le serveur Web définit l'en-tête Content-Type de manière appropriée et qu'il définit l'en-tête X-Content-Type-Options sur 'nosniff' pour toutes les pages Web. Si possible, assurez-vous que l'utilisateur final utilise un navigateur Web moderne et conforme aux normes qui n'effectue pas du tout de reniflage MIME, ou qui peut être dirigé par l'application Web/le serveur Web pour ne pas effectuer de reniflage MIME.

Ref :
http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx
https://owasp.org/www-community/Security_Headers

Tag de l'alerte
OWASP_2021_A05
OWASP_2017_A06

Fuite d'information .htaccess

Description de la vulnérabilité

Les fichiers .htaccess peuvent être utilisés pour altérer la configuration du serveur web Apache afin d'activer/désactiver des fonctionnalités et caractéristiques que le serveur web Apache peut offrir.

Impact potentiel : Divulgation de configurations de sécurité

- **Impact : Moyen** car la divulgation de configurations peut aider les attaquants.
- **Probabilité : Bas** car l'accès aux fichiers .htaccess est généralement restreint.
- **Note de risque** : 4/10

Solution : S'assurer que le fichier .htaccess n'est pas accessible.

Requêtes envoyées

```
GET http://[redacted].ma/.htaccess HTTP/1.1
```

Requêtes reçus

```
HTTP/1.1 200 OK
```

Ref :
http://www.htaccess-guide.com/

Tag de l'alerte

OWASP_2021_A05
WSTG-v42-CONF-05
OWASP_2017_A06

Fichier caché trouvé

Description de la vulnérabilité

Un dossier de nature délicate a été identifié comme étant accessible ou disponible. Cela peut divulguer des informations administratives, de configuration ou d'identification qui peuvent être exploitées par une personne malveillante pour attaquer davantage le système ou mener des efforts d'ingénierie sociale.

Impact potentiel : Exposition de données sensibles.

- **Impact : Moyen** car les fichiers cachés peuvent contenir des informations importantes.
- **Probabilité : Bas** car les fichiers cachés ne sont pas toujours accessibles ou exploitables.
- **Note de risque** : 4/10

Solution : Déterminez si le composant est réellement requis ou non en production, si ce n'est pas le cas, désactivez-le. Si c'est le cas, assurez-vous que l'accès à celui-ci nécessite une authentification et une autorisation appropriées, ou limitez l'exposition aux systèmes internes ou à des adresses IP sources spécifiques, etc.

Requêtes envoyées

```
GET http://[redacted].ma/._darcs HTTP/1.1
GET http://[redacted].ma/.bzr HTTP/1.1
GET http://[redacted].ma/.hg HTTP/1.1
GET http://[redacted].ma/BitKeeper HTTP/1.1
```

Requêtes reçus

```
HTTP/1.1 200 OK
```

Ref :
https://blog.hboeck.de/archives/892-Introducing-Snallygaster-a-Tool-to-Scan-for-Secrets-on-Web-Servers.html

Tag de l'alerte
OWASP_2021_A05
WSTG-v42-CONF-05
OWASP_2017_A06

Informations sur la version du serveur divulguées via le champ d'en-tête de réponse HTTP "Serveur"

Description de la vulnérabilité

Le serveur Web/d'applications divulgue des informations de version via l'en-tête de réponse HTTP « Serveur ». L'accès à ces informations peut permettre aux attaquants d'identifier d'autres vulnérabilités auxquelles votre serveur Web/d'applications est soumis.

Impact potentiel : Aide les attaquants à identifier et exploiter des vulnérabilités spécifiques.

- **Impact : Faible car bien que cela puisse aider les attaquants, ce n'est pas une vulnérabilité directe.**
- **Probabilité : Moyen car les serveurs exposent souvent cette information par défaut.**
- **Note de risque** : 3/10

Requêtes envoyées

```
GET http://[redacted].ma HTTP/1.1
```

Requête reçus

```
HTTP/1.1 200 OK
Date: Fri, 24 May 2024 12:21:42 GMT
Content-Length: 1446
Connection: keep-alive
Cache-Control: no-cache, no-store, must-revalidate, max-age=0
Cache-Control: no-store, max-age=0
Server: imunify360-webshield/1.21
```

Solution : Assurez-vous que votre serveur Web, serveur d'applications, équilibreur de charge, etc. est configuré pour supprimer l'en-tête « Serveur » ou fournir des détails génériques.

Ref : http://httpd.apache.org/docs/current/mod/core.html#servertokens
http://msdn.microsoft.com/en-us/library/ff648552.aspx#ht_urlscan_007
http://blogs.msdn.com/b/varunm/archive/2013/04/23/remove-unwanted-http-response-headers.aspx
http://www.troyhunt.com/2012/02/shhh-dont-let-your-response-headers.html

Tag de l'alerte

OWASP_2021_A05
OWASP_2017_A06
WSTG-v42-INFO-02

Matrice des risques

Ce document est le complément d'un rapport de vulnérabilités où il y a plus d'informations (les requêtes envoyées, les outils utilisées, etc).

Les vulnérabilités ont été classées par ordre allant des vulnérabilités le plus importants aux vulnérabilités dites acceptable.

Vulnérabilité	Description	Impact potentiel	Niveau de risque	Mesures recommandées	Probabilité	Note de la vulnérabilité
Injection de modèles côté serveur (aveugle)	Permet l'exécution de code malveillant via des templates vulnérables	Exécution de code arbitraire, compromission complète du serveur	Très Élevé	Valider et nettoyer les données des templates, utiliser des modèles sécurisés	Moyen	9/10
SSRF aveugle non authentifié via DNS Rebinding	Permet à un attaquant d'envoyer des requêtes vers des serveurs internes en utilisant le DNS rebinding	Accès non autorisé à des systèmes internes, vol de données sensibles	Élevé	Restreindre les requêtes sortantes, utiliser des listes blanches pour les domaines	Moyen	8/10
Déni de service via empoisonnement du cache	Utilisation malveillante du cache pour provoquer un déni de service	Indisponibilité du service	Élevé	Implémenter des mécanismes de validation pour le cache, surveiller les accès	Haut	8/10
XSS Stocké Authentifié (Seo Manager+)	Exécution de scripts malveillants via des entrées stockées	Vol de session, défiguration de site, vol de données	Élevé	Valider et encoder les entrées utilisateur, utiliser des en-têtes de sécurité	Haut	8/10
Présence potentielle d'informations personnellement identifiable	Divulgation de données PII non sécurisées	Vol d'identité, atteinte à la vie privée	Élevé	Anonymiser les données, sécuriser les bases de données	Haut	8/10
Absence de Jetons Anti-CSRF	Absence de protection contre les attaques CSRF	Prise de contrôle de session, actions non autorisées	Élevé	Implémenter des jetons anti-CSRF	Haut	8/10
Traversée de répertoires via fichiers de traduction	Permet à un attaquant d'accéder à des fichiers en dehors du	Vol de données sensibles, compromission du système	Élevé	Valider et nettoyer les entrées utilisateur, restreindre	Moyen	7/10

	répertoire prévu			l'accès aux répertoires critiques		
En-tête de politique de sécurité du contenu (CSP) non défini	Absence de politiques de sécurité pour le contenu chargé par le navigateur	Vulnérabilité aux attaques XSS et injection de contenu	Élevé	Définir une politique CSP appropriée	Moyen	7/10
Exécution de shortcodes dans les données générées par les utilisateurs	Permet l'injection de contenu malveillant via des shortcodes	Exécution de code malveillant, vol de données	Élevé	Valider et restreindre l'utilisation de shortcodes	Moyen	7/10
XSS réfléchi via demandes de mot de passe d'application	Permet l'exécution de scripts malveillants via des entrées utilisateur non validées	Vol de session, compromission des comptes	Élevé	Valider et encoder les entrées utilisateur	Moyen	7/10
Injection de contenu Contributor+	Permet à des contributeurs d'injecter du contenu malveillant	Exécution de code malveillant, défiguration de site	Élevé	Restreindre les permissions, valider les contributions	Moyen	7/10
Métadonnée du cloud potentiellement vulnérable	Divulgation d'informations sensibles via les métadonnées du cloud	Fuite d'informations sensibles, compromission du système	Élevé	Restreindre l'accès aux métadonnées, sécuriser les configurations cloud	Bas	6/10
En-tête Anti-clickjacking manquante	Absence de protection contre les attaques de clickjacking	Redirection malveillante, vol de clics	Moyen	Ajouter l'en-tête X-Frame-Options	Moyen	6/10
En-tête strict-transport-security non défini	Absence de l'en-tête HSTS pour forcer les connexions HTTPS	Exposition aux attaques de type MITM	Moyen	Ajouter l'en-tête Strict-Transport-Security	Moyen	6/10
La mise à jour de l'image miniature via CSRF	Permet à un attaquant de modifier des images via une attaque CSRF	Compromission de l'intégrité des images, défiguration	Moyen	Implémenter des jetons anti-CSRF	Moyen	6/10
Cookie sans indicateur sécurisé	Transmission de cookies sensibles sans l'indicateur sécurisé	Exposition aux attaques MITM	Moyen	Ajouter l'indicateur Secure aux cookies	Moyen	6/10
En-tête X-Content-Type-Options	Absence de protection contre les	Vulnérabilité à l'exécution de scripts	Moyen	Ajouter l'en-tête X-Content-Type-Options	Moyen	5/10

manquant	interprétations de types MIME non sûres	malveillants				
Fuite d'information .htaccess	Divulgation de configurations sensibles via le fichier .htaccess	Fuite d'informations sensibles, compromission de la sécurité	Moyen	Restreindre l'accès au fichier .htaccess, examiner les configurations	Bas	5/10
En-tête Content-Type manquant ou vide	Absence d'indication sur le type de contenu retourné par le serveur	Risque d'interprétation incorrecte du contenu par le navigateur	Bas	Définir l'en-tête Content-Type de manière appropriée	Bas	4/10
Fichier caché trouvé	Existence de fichiers cachés pouvant contenir des informations sensibles	Divulgation d'informations sensibles	Bas	Scanner et supprimer les fichiers inutiles, restreindre les permissions	Moyen	4/10
Informations sur la version du serveur divulguées via le champ d'en-tête de réponse HTTP 'Serveur'	Divulgation de la version du serveur pouvant faciliter les attaques	Identification de vulnérabilités spécifiques	Bas	Masquer ou généraliser l'en-tête 'Serveur'	Bas	4/10

Vulnérabilités total trouvées : 21

Fait par G.Tony LAMBERT-TATHY

Plan d'Action pour Traiter les Vulnérabilités du rapport.

Ce plan d'action est accompagné par un rapport de vulnérabilité et une matrice.

Voici un plan d'action détaillé pour traiter les vulnérabilités identifiées, classées par ordre de priorité en fonction de leur niveau de risque :

Priorité 1 : Vulnérabilités à Risque Très Élevé et Élevé

1. **Injection de modèles côté serveur (aveugle)**
    - **Action :** Mettre à jour les bibliothèques de templates pour utiliser des versions sécurisées.
    - **Action :** Valider et assainir toutes les données d'entrée avant de les intégrer dans les templates.
    - **Délai :** Immédiatement (1-2 semaines).
2. **SSRF aveugle non authentifié via DNS Rebinding**
    - **Action :** Restreindre les requêtes sortantes aux domaines connus et approuvés.
    - **Action :** Mettre en place des mécanismes de validation et de filtrage pour les requêtes DNS.
    - **Délai :** Immédiatement (1-2 semaines).
3. **Déni de service via empoisonnement du cache**
    - **Action :** Mettre en œuvre des mécanismes de validation et d'authentification pour les accès au cache.
    - **Action :** Surveiller les accès au cache pour détecter les comportements anormaux.
    - **Délai :** Immédiatement (1-2 semaines).
4. **XSS Stocké Authentifié (Seo Manager+)**
    - **Action :** Valider et encoder toutes les entrées utilisateur pour éviter l'injection de scripts.
    - **Action :** Utiliser des en-têtes de sécurité appropriés comme Content Security Policy (CSP).
    - **Délai :** Immédiatement (1-2 semaines).
5. **Présence potentielle d'informations personnellement identifiable (PII)**
    - **Action :** Auditer et anonymiser les données PII présentes dans les systèmes.
    - **Action :** Mettre en œuvre des mesures de chiffrement pour les données sensibles.
    - **Délai :** Immédiatement (1-2 semaines).
6. **Absence de Jetons Anti-CSRF**
    - **Action :** Implémenter des jetons anti-CSRF pour toutes les actions sensibles.
    - **Action :** Mettre à jour les applications pour vérifier les jetons anti-CSRF.
    - **Délai :** Immédiatement (1-2 semaines).

Priorité 2 : Vulnérabilités à Risque Moyen

7. **Traversée de répertoires via fichiers de traduction**

- **Action :** Valider et assainir les chemins de fichiers pour éviter la traversée de répertoires.
- **Action :** Restreindre l'accès aux répertoires critiques.
- **Délai :** Court terme (2-4 semaines).

8. **En-tête de politique de sécurité du contenu (CSP) non défini**
   - **Action :** Définir et implémenter une politique CSP adaptée.
   - **Action :** Tester et ajuster la politique CSP pour s'assurer de sa compatibilité.
   - **Délai :** Court terme (2-4 semaines).
9. **Exécution de shortcodes dans les données générées par les utilisateurs**
   - **Action :** Restreindre l'utilisation des shortcodes aux utilisateurs de confiance.
   - **Action :** Valider et assainir les données des shortcodes avant exécution.
   - **Délai :** Court terme (2-4 semaines).
10. **XSS réfléchi via demandes de mot de passe d'application**
    - **Action :** Valider et encoder toutes les entrées utilisateur.
    - **Action :** Utiliser des en-têtes de sécurité appropriés.
    - **Délai :** Court terme (2-4 semaines).
11. **Injection de contenu Contributor+**
    - **Action :** Restreindre les permissions de contenu pour les contributeurs.
    - **Action :** Valider et assainir toutes les contributions avant publication.
    - **Délai :** Court terme (2-4 semaines).
12. **Métadonnée du cloud potentiellement vulnérable**
    - **Action :** Restreindre l'accès aux métadonnées du cloud.
    - **Action :** Mettre en œuvre des politiques de sécurité strictes pour les configurations cloud.
    - **Délai :** Moyen terme (1-2 mois).

Priorité 3 : Vulnérabilités à Risque Faible

13. **En-tête Anti-clickjacking manquante**
    - **Action :** Implémenter les en-têtes anti-clickjacking (X-Frame-Options ou Content Security Policy).
    - **Délai :** Moyen terme (1-2 mois).
14. **En-tête strict-transport-security non défini**
    - **Action :** Définir et implémenter l'en-tête HSTS pour forcer l'utilisation de HTTPS.
    - **Délai :** Moyen terme (1-2 mois).
15. **En-tête X-Content-Type-Options manquant**

- **Action :** Ajouter l'en-tête X-Content-Type-Options: nosniff pour empêcher le MIME sniffing.
- **Délai :** Moyen terme (1-2 mois).

16. **L'en-tête Content-Type manquant ou vide**
    - **Action :** S'assurer que tous les contenus ont un en-tête Content-Type approprié.
    - **Délai :** Moyen terme (1-2 mois).
17. **La mise à jour de l'image miniature via CSRF**
    - **Action :** Implémenter des jetons anti-CSRF pour les actions de mise à jour.
    - **Délai :** Moyen terme (1-2 mois).
18. **Cookie sans indicateur sécurisé**
    - **Action :** Utiliser l'indicateur "Secure" pour tous les cookies sensibles.
    - **Délai :** Moyen terme (1-2 mois).
19. **Fuite d'information .htaccess**
    - **Action :** Restreindre l'accès aux fichiers .htaccess.
    - **Action :** Vérifier et corriger les permissions de fichiers.
    - **Délai :** Moyen terme (1-2 mois).
20. **Fichier caché trouvé**
    - **Action :** Supprimer ou sécuriser les fichiers cachés contenant des informations sensibles.
    - **Délai :** Moyen terme (1-2 mois).
21. **Informations sur la version du serveur divulguées via le champ d'en-tête de réponse HTTP "Serveur"**
    - **Action :** Masquer ou modifier les en-têtes de version du serveur pour ne pas divulguer d'informations sensibles.
    - **Délai :** Moyen terme (1-2 mois).

Suivi et Évaluation

- **Audits réguliers :** Planifier des audits de sécurité réguliers pour vérifier la correction des vulnérabilités.
- **Tests de pénétration :** Effectuer des tests de pénétration après la mise en œuvre des mesures pour s'assurer de leur efficacité.
- **Mises à jour et formations :** Mettre à jour régulièrement les logiciels et former le personnel aux meilleures pratiques de sécurité.

Ce plan d'action vise à prioriser les vulnérabilités les plus critiques tout en assurant une couverture complète de toutes les vulnérabilités identifiées.

outils utilisés

- **Kali Linux**

Kali Linux est une distribution Linux spécialisée dans les tests d'intrusion et l'audit de sécurité. Elle est préchargée avec de nombreux outils de piratage éthique et de cybersécurité.[1]

- **NordVPN**

NordVPN est un service VPN populaire qui permet de chiffrer le trafic internet et de masquer l'adresse IP pour plus de confidentialité et de sécurité en ligne.

- **OWASP ZAP**

ZAP (Zed Attack Proxy) est un outil d'analyse de sécurité web gratuit et open source développé par OWASP. Il permet de scanner les applications web à la recherche de failles de sécurité.[1]

- **Burp Suite**

Burp Suite est un ensemble d'outils graphiques pour tester la sécurité des applications web. Il comprend un proxy intercepteur, un scanner de vulnérabilités, un outil d'attaque par fuzzing, etc.[1]

- **WPScan**

WPScan est un outil de reconnaissance de failles de sécurité spécifique aux sites WordPress. Il peut détecter les plugins, thèmes et versions installés pour identifier les vulnérabilités.[3]

- **Nmap**

Nmap (Network Mapper) est un outil de scanning de réseaux et d'audit de sécurité. Il permet de détecter les ports ouverts, les services en cours d'exécution et d'obtenir des informations sur les systèmes cibles.[1]

- **Wappalyzer**

Wappalyzer est une extension de navigateur qui détecte les technologies utilisées par un site web, comme les CMS, frameworks, bibliothèques, etc.[1][4]

- **FFuF**

FFuF (Fuzz Faster U Fool) est un outil de fuzzing rapide pour trouver des répertoires, fichiers et sous-domaines potentiellement intéressants sur un site web.[4]

- **The Harvester**

The Harvester est un outil de reconnaissance qui collecte des informations publiques comme les adresses e-mail, sous-domaines, hôtes virtuels, etc. à partir de différentes sources en ligne.[4]

- **Amass**

Amass est un outil de cartographie de surface d'attaque qui utilise des techniques avancées pour collecter des informations sur les infrastructures d'une cible à partir de sources publiques.[1][4]

ref:
[1] https://www.billdietrich.me/PenetrationTestingAndBugBountyHuntingTools.html
[2] https://kr-labs.com.ua/blog/testuvannya-na-pronyknennya-pentest-vid-a-do-ya/
[3] https://kalitut.com/penetration-testing-resources/
[4] https://github.com/allwinnoah/CyberSecurity-Tools
[5] https://github.com/zeev-mindali/cyber-security-tools