Rapport de Pentesting

Audit de sécurité – Plateforme Calimeg
Date : 07-08-2025

Résumé Exécutif | Méthodologie | Découvertes | Conclusion

Résumé Exécutif

Dans ce rapport, nous présentons les résultats de l'évaluation de sécurité automatisée réalisée. L'objectif était d'identifier les vulnérabilités critiques, moyennes et faibles.

Méthodologie et Outils

Outils utilisés :

- Analyse des headers et du contenu (code source)
- Nmap (scan de ports réseau)
- Sqlmap (injection SQL automatisée)
- PwnXSS (tests de cross-site scripting)
- Wordpress Scanner (détection des vulnérabilités)
- Agent IA Meg (orchestration et corrélation des données)

Approche :

1. Reconnaissance réseau
2. Enumeration des services
3. Tests d'injection SQL
4. Tests XSS et CSRF
5. Rapport et recommandations

Découvertes

1. Headers et contenu

- **Headers de sécurité manquants :** Content-Security-Policy, Strict-Transport-Security, X-Content-Type-Options, X-Frame-Options, Referrer-Policy, Permissions-Policy, X-XSS-Protection, Cross-Origin-Embedder-Policy, Cross-Origin-Opener-Policy, X-Permitted-Cross-Domain-Policies
- **Vulnérabilité XSS potentielle :** Les entrées utilisateur ne sont pas correctement échappées ou assainies
- **Note de sécurité critique :** Site noté F - Risques de sécurité élevés détectés

2. Scan Réseau (nmap)

Port	Service	Etat	Version
22	ssh	open	OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
80	http	open	Apache httpd 2.4.38 ((Debian))

3. Injection SQL (sqlmap)

URL	Paramètre	Payload	Impact
http://www.webexample.com	cat	cat=1 UNION ALL SELECT CONCAT(0x716b787871,0x424f727a6347646b485874656b596f665a4b7a6e75434e55625a5263687159624c59746450486466,0x717a7a6271),NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL-- -	Un attaquant peut exécuter des requêtes arbitraires sur la base de données, exfiltrer des données sensibles ou compromettre l'intégrité de la base.
http://www.webexample.com	artist	artist=-4473 UNION ALL SELECT CONCAT(0x7170767071,0x6651735657554f6f6979496470654e62634b6978766f6d7541435263646b764b536e52426f6c424e,0x716b626271),NULL,NULL-- -	Un attaquant peut exécuter des requêtes arbitraires sur la base de données, exfiltrer des données sensibles ou compromettre l'intégrité de la base.

4. Cross-Site Scripting (XSS)

Page	Paramètre	Payload	Impact
/search	test	`query`	Possibilité d'exécution de scripts malveillants dans le contexte du navigateur de la victime, compromettant la confidentialité des données et le contrôle du compte utilisateur.
/listproducts	cat	`<script>alert(document.cookie)</script>`	Exécution de scripts arbitraires pouvant voler les cookies de session et compromettre la session utilisateur.
/product	pic	`<script>alert(document.cookie)</script>`	Exécution de scripts arbitraires pouvant voler les cookies de session et compromettre la session utilisateur.
/showimage	file	`<script>alert(document.cookie)</script>`	Exécution de scripts arbitraires pouvant voler les cookies de session et compromettre la session utilisateur.

5. WordPress vulnérabiltés

Page	Faille	Description	CVE
http://www.webexample.com/blog	Vulnérabilité d'upload de fichier arbitraire non authentifié dans le plugin wp-file-manager (<= 6.8), pouvant conduire à une exécution de code à distance.	Compromission complète du site et du serveur, prise de contrôle totale possible.	http://www.cve.org/CVERecord?id=CVE-2020-25213
http://www.webexample.com/blog	Vulnérabilité d'exposition de données sensibles sans authentification dans le plugin wp-file-manager (<= 7.2.1).	Vol ou fuite d'informations sensibles pouvant mener à des attaques plus sophistiquées.	http://www.cve.org/CVERecord?id=CVE-2024-0761

Page	Faille	Description	CVE
http://www.webexample.com/blog	Vulnérabilité de traversée de répertoire authentifiée dans le plugin File Manager and File Manager Pro (plusieurs versions).	Accès non autorisé à des fichiers critiques pouvant compromettre la sécurité du site.	http://www.cve.org/CVERecord?id=CVE-2023-6825
http://www.webexample.com/blog	Vulnérabilité CSRF permettant l'inclusion locale de fichiers JS dans le plugin File Manager (<= 7.2.4).	Possibilité d'exécuter du code malveillant via inclusion locale de script JavaScript.	http://www.cve.org/CVERecord?id=CVE-2024-1538

Conclusion & Recommandations

Injections SQL:

- [haute] Mettre en place une validation et une sanitation strictes des entrées utilisateurs pour tous les paramètres GET, en particulier 'cat' et 'artist'.
- [haute] Utiliser des requêtes paramétrées (prepared statements) pour toutes les interactions avec la base de données afin d'éviter les injections SQL.
- [moyenne] Mettre à jour les versions du serveur web, du serveur PHP et du SGBD pour bénéficier des derniers patchs de sécurité.
- [moyenne] Effectuer un audit de sécurité complet et régulier de l'application web pour détecter et corriger d'autres vulnérabilités potentielles, y compris les risques XSS.

Failles XSS:

Le scan a détecté un ensemble important de vulnérabilités critiques de type XSS sur plusieurs points d'entrée du site http://www.webexample.com. Ces vulnérabilités exposent le site au vol de cookies, à l'exécution de scripts malveillants et à une compromission possible des sessions utilisateur. Une intervention immédiate est recommandée pour corriger ces failles par une gestion stricte des entrées utilisateur et une politique de sécurité renforcée.
RAPPORT DE SÉCURITÉ - ====================================== Cible: http://www.webexample.com Date: 2025-08-07T06:09:18.957Z Statut: HTTP 200 Niveau d'alerte: HIGH RÉSUMÉ: 65 vulnérabilité(s) XSS détectée(s) sur http://www.webexample.com VULNÉRABILITÉS DÉTECTÉES: 65 1. CRITICAL - XSS sur http://www.webexample.com/search.php?test=query (Méthode: POST) 2. CRITICAL - XSS sur http://www.webexample.com/search.php?test=query (Méthode: POST) 3. CRITICAL - XSS sur http://www.webexample.com/search.php?test=query (Méthode: POST) 4. CRITICAL - XSS sur http://www.webexample.com/listproducts.php?cat=%3Cscript%3Ealert(document.cookie)%3C/script%3E (Méthode: GET) 5. CRITICAL - XSS sur http://www.webexample.com/listproducts.php?cat=%3Cscript%3Ealert(document.cookie)%3C/script%3E (Méthode: GET) 6. CRITICAL - XSS sur http://www.webexample.com/listproducts.php?cat=%3Cscript%3Ealert(document.cookie)%3C/script%3E (Méthode: GET) 7. CRITICAL - XSS sur http://www.webexample.com/listproducts.php?cat=%3Cscript%3Ealert(document.cookie)%3C/script%3E (Méthode: GET) 8. CRITICAL - XSS sur http://www.webexample.com/search.php?test=query (Méthode: POST) 9. CRITICAL - XSS sur http://www.webexample.com/product.php?pic=%3Cscript%3Ealert(document.cookie)%3C/script%3E (Méthode: GET) 10. CRITICAL - XSS sur http://www.webexample.com/showimage.php?file=%3Cscript%3Ealert(document.cookie)%3C/script%3E (Méthode: GET) RECOMMANDATION: ACTION IMMÉDIATE REQUISE

- **Recommandation 1:** Filtrer et échapper correctement toutes les entrées utilisateur avant affichage pour prévenir les attaques XSS - *Priorité Haute* - Action: Implémenter une validation stricte côté serveur et côté client, utiliser des fonctions d'échappement sécurisées (ex: htmlentities en PHP).
- **Recommandation 2:** Revoir la politique de gestion des entrées des paramètres GET et POST pour inclure des whitelistings - *Priorité Haute* - Action: Mettre en place des filtres lors de la réception des données entrantes, rejeter toute donnée suspecte ou non conforme.
- **Recommandation 3:** Mettre en place un Content Security Policy (CSP) efficace pour limiter l'exécution de scripts non autorisés - *Priorité Moyenne* - Action: Configurer les en-têtes HTTP CSP avec directives restrictives (ex: script-src 'self').

Vulnérabiltés WordPress:

- [Haute] Mettre à jour immédiatement le plugin wp-file-manager vers la dernière version disponible corrigée qui traite toutes les vulnérabilités listées.
- [Haute] Restreindre les accès au plugin via des politiques d'authentification fortes et limiter l'exposition publique tant que la mise à jour n'est pas appliquée.
- [Moyenne] Effectuer un audit complet des logs et configurations pour détecter toute compromission éventuelle liée à ces vulnérabilités.
- [Moyenne] Mettre en place des mécanismes de détection d'intrusion pour surveiller toute activité anormale liée au plugin vulnérable.

Utilisateurs Wordpress identifiés:

- WP-Admin