

TP3 de Cyber

Étape 2. Analyse des Vulnérabilités :

Pour cette étape, nous avons lancé un scan complet de la plateforme Juice Shop en utilisant l'outil ZAP OWASP.

Le scan a révélé plusieurs vulnérabilités OWASP que nous avons analysées et documentées ci-dessous.

Pour chacune d'elles, la catégorie OWASP se trouve en comprenant la vulnérabilité et en l'associant aux catégories données par le site de documentation.

Nous avons pu reprendre les descriptions et le niveau de gravité donnés par ZAP.

Enfin, la preuve d'exploitation se retrouve en trouvant des exemples vis-à-vis de cette vulnérabilité.

Content Security Policy (CSP) Header Not Set (2)

- Catégorie OWASP : A3 - Cross-Site Scripting (XSS)
- Description : L'absence de l'en-tête CSP permet l'exécution de scripts malveillants sur le site, augmentant ainsi le risque d'attaques XSS.
- Niveau de gravité : Moyen
- Preuves d'exploitation : Les attaquants peuvent injecter du code JavaScript malveillant dans les pages du site pour dérober des informations ou effectuer des actions non permises.

Mauvaise configuration inter-domaines (9)

- Catégorie OWASP : A4 - Insecure Direct Object References
- Description : Cette vulnérabilité peut permettre à un attaquant d'accéder à des ressources ou à des fonctionnalités sur le site auxquelles il n'est pas normalement censé avoir accès.
- Niveau de gravité : Moyen
- Preuves d'exploitation : Un attaquant pourrait exploiter cette faille pour accéder à des données sensibles ou effectuer des actions malveillantes sur le site.

Cross-Domain JavaScript Source File Inclusion (4)

- Catégorie OWASP : A8 - Insecure Deserialization
- Description : Cette vulnérabilité permet à un attaquant d'inclure des fichiers JavaScript provenant de domaines externes dans les pages du site.
- Niveau de gravité : Faible
- Preuves d'exploitation : Un attaquant pourrait inclure des scripts malveillants pour exécuter des attaques XSS ou autres.

Strict-Transport-Security Header Not Set (9)

- Catégorie OWASP : A5 - Security Misconfiguration
- Description : L'absence de l'en-tête Strict-Transport-Security peut rendre le site vulnérable aux attaques de type MITM (Man-In-The-Middle) et aux attaques de détournement de session.
- Niveau de gravité : Faible
- Preuves d'exploitation : Les attaquants pourraient exploiter cette vulnérabilité afin d'intercepter le trafic réseau et compromettre la sécurité.

Timestamp Disclosure - Unix (10)

- Catégorie OWASP : A6 - Sensitive Data Exposure
- Description : La divulgation de l'horodatage Unix permet à un attaquant de recueillir des informations sensibles sur le système ou sur les utilisateurs.
- Niveau de gravité : Faible
- Preuves d'exploitation : Les attaquants pourraient utiliser ces informations pour planifier des attaques ciblées ou effectuer des tentatives d'intrusion.

Information Disclosure - Suspicious Comments (2)

- Catégorie OWASP : A2 - Broken Authentication
- Description : La divulgation d'informations sensibles à travers des commentaires dans le code source peut faciliter les attaques de force brute ou d'ingénierie sociale.
- Niveau de gravité : Informatif
- Preuves d'exploitation : Les commentaires peuvent révéler des détails sur la structure du site ou sur les technologies utilisées, ce qui peut aider les attaquants.

Modern Web Application (2)

- Catégorie OWASP : A10 - Insufficient Logging & Monitoring
- Description : L'absence de techniques de logging et de monitoring adéquates peut rendre le site vulnérable aux attaques et aux incidents de sécurité non détectés.
- Niveau de gravité : Informatif
- Preuves d'exploitation : Sans des mécanismes de logging et de monitoring efficaces, les attaques pourraient passer inaperçues et causer des dommages considérables.

Re-examine Cache-control Directives (3)










- Catégorie OWASP : A7 - Cross-Site Request Forgery (CSRF)
- Description : Les directives de cache incorrectes peuvent exposer le site à des attaques de type CSRF en permettant la réutilisation de requêtes authentifiées.
- Niveau de gravité : Informatif

- Preuves d'exploitation : Les attaquants peuvent utiliser cette faille pour effectuer des actions malveillantes au nom des utilisateurs authentifiés.

User Agent Fuzzer (12)

- Catégorie OWASP : A1 - Injection
- Description : L'utilisation d'un outil de fuzzing du User-Agent peut être utilisée pour identifier des vulnérabilités d'injection, telles que des injections SQL ou des injections de code.
- Niveau de gravité : Informatif
- Preuves d'exploitation : Un attaquant pourrait utiliser cet outil pour tester la résilience du site aux injections et compromettre sa sécurité.

Voici une capture d'écran de ce qui a été détecté :

- >  Content Security Policy (CSP) Header Not Set (2)
- >  Mauvaise configuration inter-domaines (9)
- >  Cross-Domain JavaScript Source File Inclusion (4)
- >  Strict-Transport-Security Header Not Set (9)
- >  Timestamp Disclosure - Unix (10)
- >  Information Disclosure - Suspicious Comments (2)
- >  Modern Web Application (2)
- >  Re-examine Cache-control Directives (3)
- >  User Agent Fuzzer (12)

Étape 3. Exploitation d'une vulnérabilité :

Pour la vulnérabilité "User Agent Fuzzer" identifiée dans Juice Shop, nous avons développé deux méthodes d'exploitation dans le but de se connecter au compte administrateur du site sans nécessiter de mot de passe.

Méthode : Utilisation de l'Injection SQL pour Contourner l'Authentification

Cette méthode exploite une vulnérabilité d'injection SQL pour contourner l'authentification et se connecter directement au compte administrateur.

Étapes de l'Exploitation :

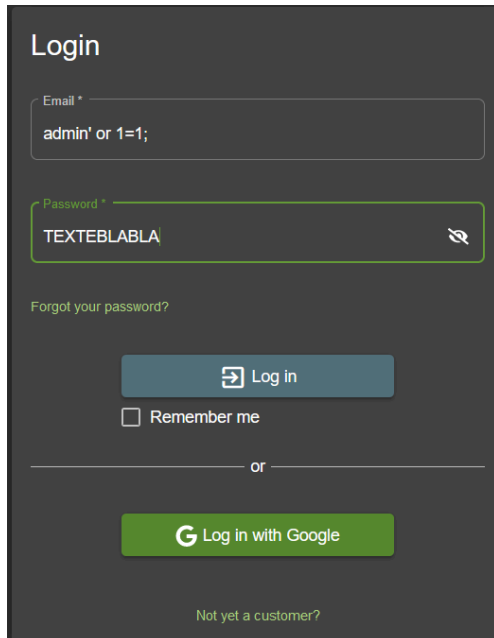
- Dans le champ d'identifiant, nous saisissons l'identifiant suivant : admin' OR 1=1;
- Dans le champ mot de passe, nous pouvons mettre ce que nous voulons, ce champ n'a pas d'importance ici.

Explication :

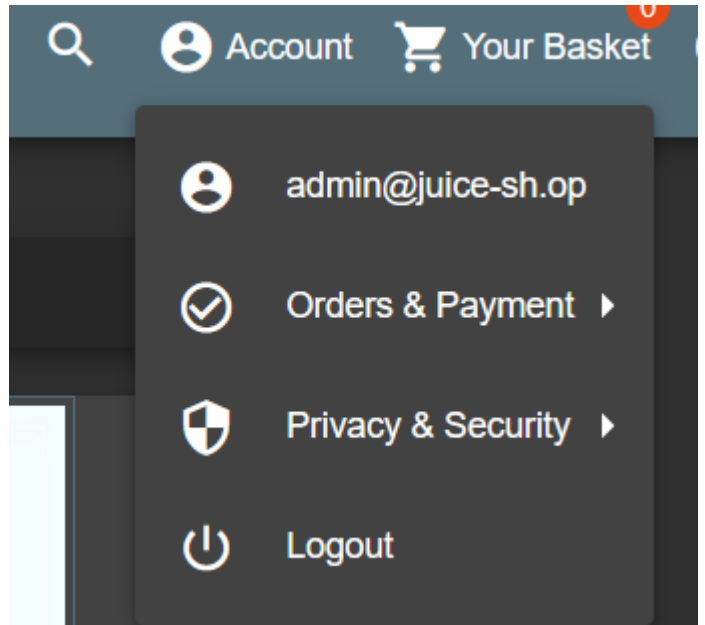
- L'apostrophe ' après admin est utilisée pour fermer la chaîne de caractères de l'identifiant.

- OR 1=1 est une condition SQL qui est toujours vraie, contournant ainsi la vérification du mot de passe.

Grâce à cette injection nous avons accès au compte admin ce qui est très dangereux car nous pouvons faire ce que nous voulons.



The screenshot shows a login form on a dark-themed website. The 'Email' field contains the text 'admin' or 1=1;'. The 'Password' field contains 'TEXTEBLABLA'. Below the fields are links for 'Forgot your password?' and a 'Log in' button. There is also a 'Remember me' checkbox and a 'Log in with Google' button. At the bottom, there is a link for 'Not yet a customer?'.



Preuve d'Exploitation :

Méthode : Utilisation du Brute Force avec Burp Suite

Cette méthode exploite une vulnérabilité de force brute pour craquer le mot de passe du compte administrateur à partir de l'adresse e-mail trouvée dans les reviews des articles sur le site.

Identification de l'Adresse E-mail de l'Admin :

- Identifiez l'adresse e-mail associée au compte administrateur. Pour cela nous avons juste trouver l'email dans les reviews des produits sur le site.
- Lancement de Burp Suite :
- Utilisez Burp Suite pour intercepter les requêtes HTTP.

Configuration de Burp pour le Brute Force :

- Dans Burp Suite, configurez l'outil "Intruder" pour le brute force.
- Utilisez l'adresse e-mail de l'admin comme identifiant.
- Utilisez une liste de mots de passe courants ou générés pour la tentative de connexion.

Lancement du Brute Force :

- Lancez le brute force à l'aide de Burp Suite.

Analyse des Réponses :

- Burp Suite tentera automatiquement différentes combinaisons de mots de passe.

- Lorsque le bon mot de passe est trouvé, Burp affiche une réponse réussie.

Accès au Compte Administrateur :

- Notez le mot de passe trouvé par Burp Suite.
- Connectez-vous manuellement en utilisant l'adresse e-mail de l'admin et le mot de passe trouvé.

Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type:

```
1 POST /rest/user/login HTTP/1.1
2 Host: localhost:3000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/json
8 Content-Length: 47
9 Origin: http://localhost:3000/
10 Connection: close
11 Referer: http://localhost:3000/
12 Cookie: io=aQc3SIbNDsPPmJ5oAAAA; language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss; continueCode=
myvd8DUZHrTyiyhKzfYMuYjcKDIeKUaNH3jCJQjWcnWTZxFLPtPvUDxdxg
13
14 {"email":"admin@juice-sh.op","password":"wasd"}
```

Payload Options [Simple list]

This payload type lets you configure a simple list of strings


Paste	aaa
Load ...	abc123
Remove	acc
Clear	access
	adfxcc
	adm
	admin
	admin123

[Pro version only]

Request ^	Payload	Status	Error	Timeout	Length
0		401	<input type="checkbox"/>	<input type="checkbox"/>	362
1	aaa	401	<input type="checkbox"/>	<input type="checkbox"/>	362
2	abc123	401	<input type="checkbox"/>	<input type="checkbox"/>	362
3	acc	401	<input type="checkbox"/>	<input type="checkbox"/>	362
4	access	401	<input type="checkbox"/>	<input type="checkbox"/>	362
5	adfxcc	401	<input type="checkbox"/>	<input type="checkbox"/>	362
6	adm	401	<input type="checkbox"/>	<input type="checkbox"/>	362
7	admin	401	<input type="checkbox"/>	<input type="checkbox"/>	362
8	admin123	200	<input type="checkbox"/>	<input type="checkbox"/>	1166
9	admin2	401	<input type="checkbox"/>	<input type="checkbox"/>	362
10	admin_1	401	<input type="checkbox"/>	<input type="checkbox"/>	362
11	administrator	401	<input type="checkbox"/>	<input type="checkbox"/>	362
12	adminstat	401	<input type="checkbox"/>	<input type="checkbox"/>	362

Login

Email *

Password * 

[Forgot your password?](#)

☐ Remember me

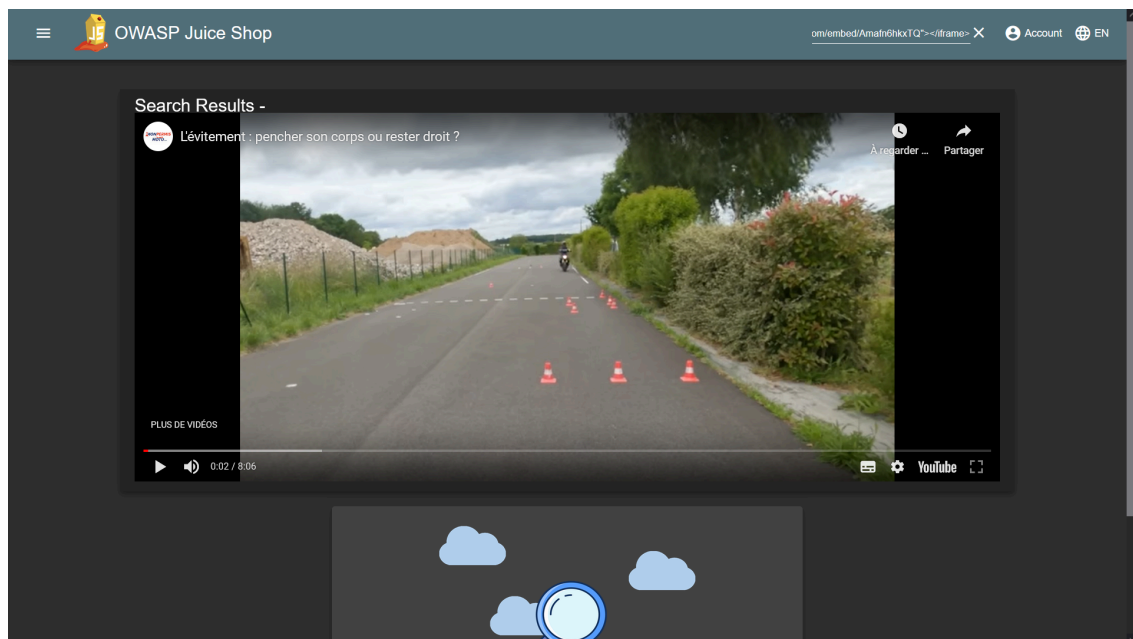
[BONUS] Injection XSS

En continuant à exploiter les failles nous en avons trouvé une simple.

En effet, en exploitant la première (CSP), nous pouvons injecter du code html dans la barre de recherche et ajouter par exemple une vidéo youtube au site qui n'était pas prévu à cet effet.

Commande utilisé :

```
<iframe width="100%" height="500" scrolling="yes" frameborder="no" allow="autoplay" src="https://www.youtube.com/embed/Amaf6hkxTQ"></iframe>
```



Étape 4 : Rapport d'Audit de Sécurité

1. Introduction

Cet audit de sécurité vise à évaluer la robustesse de la plateforme Juice Shop hébergée sur <https://juice-shop.herokuapp.com>, en utilisant l'outil ZAP OWASP. L'audit a été réalisé dans le cadre du cours "8SEC201 – Cybersécurité : Vulnérabilités et Incidents" dans l'ordre du travail pratique 3.

2. Objectifs de l'Audit

L'audit avait pour objectifs :

- Identifier, documenter et évaluer les vulnérabilités de sécurité présentes dans l'application Juice Shop.
- Déterminer le niveau de gravité de chaque vulnérabilité en fonction de sa faisabilité d'exploitation et de son impact potentiel sur la sécurité de l'application.
- Comprendre les risques associés à ces vulnérabilités pour la confidentialité, l'intégrité et la disponibilité des données.
- Formuler des recommandations spécifiques pour remédier à ces vulnérabilités et améliorer la sécurité globale de l'application.

3. Méthodologie

Pour réaliser cet audit de sécurité, les étapes suivantes ont été suivies :

3.1 Installation et Configuration de ZAP OWASP

Installation de ZAP OWASP:

- ZAP OWASP a été téléchargé depuis le site officiel (<https://www.zaproxy.org/download/>) et installé sur une machine locale.

Installation de Java Development Kit (JDK):

- Une JDK a été installée sur la machine pour permettre l'exécution de ZAP. La JDK correspondante a été téléchargée depuis le site officiel d'Oracle (<https://www.oracle.com/java/technologies/downloads/>).

Dans notre cas, celui-ci était déjà présent sur la machine locale.

Configuration de ZAP:

- ZAP a été configuré pour scanner la plateforme Juice Shop en spécifiant l'URL cible (<https://juice-shop.herokuapp.com>).

3.2 Analyse des Vulnérabilités

Scan Complet avec ZAP OWASP:

- Un scan complet de Juice Shop a été lancé avec ZAP OWASP pour détecter les vulnérabilités.

Identification des Vulnérabilités:

- Les vulnérabilités découvertes ont été identifiées à l'aide de ZAP OWASP.

3.3 Exploitation de Vulnérabilités

Méthode 1 : Injection SQL pour Contourner l'Authentification:

- Utilisation d'une injection SQL pour contourner l'authentification et accéder au compte administrateur.

Méthode 2 : Brute Force avec Burp Suite:

- Utilisation d'un brute force avec Burp Suite pour craquer le mot de passe du compte administrateur.

BONUS : Injection XSS

- Utilisation d'un code html dans la barre de recherche du site pour y intégrer des éléments.

4. Résultats de l'Analyse des Vulnérabilités

Les vulnérabilités suivantes ont été découvertes lors de l'audit :

4.1 Content Security Policy (CSP) Header Not Set

Catégorie OWASP: A6 - Sécurité de Contenu Non Sécurisé

Description: L'absence de l'en-tête CSP expose l'application aux risques de XSS.

Niveau de Gravité: Moyen

Analyse Détaillée:

- Le CSP est un mécanisme de sécurité qui permet de limiter les types de ressources chargées par une page web. Son absence expose l'application aux attaques XSS.
- Impact potentiel : Les attaquants pourraient injecter du code JavaScript malveillant dans les pages, compromettant ainsi la sécurité des utilisateurs.
- Recommandation : Mettre en place une politique de sécurité CSP stricte pour limiter les sources de scripts autorisées.

4.2 Mauvaise Configuration Inter-Domains

Catégorie OWASP: A10 - Mauvaise Gestion de la Sécurité

Description : Risque de CSRF et de manipulations de données entre domaines.

Niveau de Gravité : Moyen

Analyse Détaillée :

- Une mauvaise configuration inter-domaines peut permettre à un attaquant d'exécuter des actions indésirables au nom de l'utilisateur authentifié.
- Impact potentiel : Les attaquants pourraient mener des attaques CSRF, manipuler les données de l'utilisateur, ou voler des informations sensibles.
- Recommandation : Mettre en place des mécanismes de défense CSRF, comme les jetons anti-CSRF (CSRF tokens), et réviser la gestion des cookies et des sessions.

4.3 Cross-Domain JavaScript Source File Inclusion

Catégorie OWASP : A8 - Inclusion de Ressources Non Sécurisée

Description : Risque d'injection de code malveillant.

Niveau de Gravité : Faible

Analyse Détaillée :

- Cette vulnérabilité permet à un attaquant d'inclure des fichiers JavaScript malveillants depuis un domaine externe.
- Impact potentiel : Les attaquants pourraient exécuter du code JavaScript non autorisé sur les navigateurs des utilisateurs.
- Recommandation : Limiter les sources de fichiers JavaScript externes autorisées et mettre en place des contrôles de sécurité pour l'inclusion de ressources.

4.4 Strict-Transport-Security Header Not Set

Catégorie OWASP : A3 - Mauvaise Gestion de la Sécurité

Description : L'absence de l'en-tête Strict-Transport-Security expose l'application aux attaques de type Man-in-the-Middle.

Niveau de Gravité : Faible

Analyse Détaillée :

- L'en-tête Strict-Transport-Security permet de forcer l'utilisation de HTTPS, améliorant ainsi la sécurité des communications.
- Impact potentiel : Les attaquants pourraient intercepter les communications et voler des informations sensibles.
- Recommandation : Configurer l'en-tête Strict-Transport-Security avec une durée appropriée pour garantir l'utilisation de HTTPS.

4.5 Timestamp Disclosure - Unix

Catégorie OWASP : A2 - Vulnérabilités du Protocole

Description : Divulgateion d'informations sensibles à travers des timestamps Unix.

Niveau de Gravité : Faible

Analyse Détaillée :

- L'application divulgue des informations sensibles telles que des timestamps Unix qui pourraient être utilisés par un attaquant pour planifier des attaques.
- Impact potentiel : Les attaquants pourraient utiliser ces informations pour cibler des vulnérabilités spécifiques.
- Recommandation : Limiter les informations divulguées via les timestamps et mettre en place des contrôles d'accès stricts.

4.6 Information Disclosure - Suspicious Comments

Catégorie OWASP : A3 - Mauvaise Gestion de la Sécurité

Description : Divulgateion d'informations sensibles à travers des commentaires suspects.

Niveau de Gravité : Informatif

Analyse Détaillée :

- Des commentaires suspects dans le code peuvent révéler des informations sensibles sur l'architecture et les fonctionnalités de l'application.
- Impact potentiel : Les attaquants pourraient utiliser ces informations pour comprendre le fonctionnement interne de l'application.
- Recommandation : Supprimer ou masquer les commentaires contenant des informations sensibles.

4.7 Modern Web Application

Catégorie OWASP : A9 - Utilisation de Composants Vulnérables

Description : Risque lié à l'utilisation de composants logiciels vulnérables.

Niveau de Gravité : Informatif

Analyse Détaillée :

- L'application utilise des composants logiciels qui pourraient être obsolètes ou comporter des vulnérabilités connues.
- Impact potentiel : Les attaquants pourraient exploiter ces vulnérabilités pour compromettre l'application.
- Recommandation : Mettre à jour les composants logiciels et surveiller régulièrement les vulnérabilités connues.

4.8 Re-examine Cache-control Directives

Catégorie OWASP : A5 - Fonctionnalités Malveillantes et Abusives

Description : Risque lié à la mise en cache de données sensibles.

Niveau de Gravité : Informatif

Analyse Détaillée :

- L'application peut être configurée pour mettre en cache des données sensibles sans les directives appropriées.
- Impact potentiel : Les données sensibles pourraient être accessibles à des utilisateurs non autorisés via le cache.
- Recommandation : Mettre en place des directives de cache appropriées pour éviter la mise en cache de données sensibles.

4.9 User Agent Fuzzer

Catégorie OWASP : A8 - Inclusion de Ressources Non Sécurisée

Description : Risque lié à l'inclusion de fichiers JavaScript externes.

Niveau de Gravité : Informatif

Analyse Détaillée :

- L'application inclut des fichiers JavaScript externes sans validation appropriée.
- Impact potentiel : Les attaquants pourraient exécuter du code JavaScript non autorisé sur les navigateurs des utilisateurs.
- Recommandation : Limiter les sources de fichiers JavaScript externes autorisées et mettre en place des contrôles de sécurité pour l'inclusion de ressources.

5. Exploitation de Vulnérabilités

5.1 Méthode 1 : Injection SQL pour Contourner l'Authentification

- Description :

Cette méthode utilise une injection SQL pour contourner l'authentification et accéder au compte administrateur.

- Étapes de l'exploitation :

1. Identifier le formulaire de connexion.
2. Saisir l'identifiant "admin' OR 1=1;"
3. Soumettre le formulaire de connexion.
4. Accéder au compte administrateur sans mot de passe.

- Preuve d'Exploitation :

Voir Étape 3.

5.2 Méthode 2 : Brute Force avec Burp Suite

- Description :

Cette méthode utilise un brute force avec Burp Suite pour craquer le mot de passe du compte administrateur.

- Étapes de l'Exploitation :

1. Identifier l'adresse e-mail de l'administrateur dans les reviews des articles.
2. Configurer Burp Suite pour le brute force.
3. Lancer le brute force avec une liste de mots de passe.
4. Trouver le mot de passe du compte administrateur.

- Preuve d'Exploitation :

Voir Étape 3.

6. Conclusions et Recommandations

6.1 Conclusions

Après l'analyse approfondie de Juice Shop, les conclusions suivantes ont été tirées :

- Juice Shop présente plusieurs vulnérabilités de sécurité, notamment des risques de XSS, CSRF, et d'interception de communications.
- L'absence de l'en-tête CSP, la mauvaise configuration inter-domaines, les inclusions de fichiers JavaScript et l'absence de l'en-tête Strict-Transport-Security sont des points critiques.

6.2 Recommandations

Les recommandations suivantes sont formulées pour renforcer la sécurité de Juice Shop :

- Mettre en œuvre une Politique de Sécurité du Contenu (CSP) pour limiter les risques de XSS.
- Réviser et corriger les configurations inter-domaines pour prévenir les attaques CSRF.
- Mettre en place l'en-tête Strict-Transport-Security pour sécuriser les communications.
- Limiter les sources de fichiers JavaScript externes autorisées et mettre en place des contrôles de sécurité pour l'inclusion de ressources.
- Mettre en place des mécanismes de défense CSRF, comme les jetons anti-CSRF (CSRF tokens).
- Réexaminer les directives de cache pour éviter la mise en cache de données sensibles.

7. Documents à l'Appui

Voir captures d'écran Étape 3.

8. Conclusion

Cet audit de sécurité a permis d'identifier et d'évaluer les vulnérabilités de sécurité de Juice Shop.

Les recommandations formulées visent à améliorer la sécurité globale de l'application et à réduire les risques pour les données des utilisateurs.

Il est essentiel de mettre en œuvre ces recommandations de manière proactive pour assurer un environnement d'application web plus sûr.