Hackazon Website Pentest

Keissy Bod, Milan Pouteau

October 2024

SOMMAIRE

1. Préambule

1.1 Présentation des résultats

> FIXME: Brief introduction to the findings, summarizing the key takeaways.

1.2 Contexte

Dans le cadre de cette mission, il nous a été demandé de réaliser un test d'intrusion sur l'application web **Hackazon** accessible via l'URL https://hackazon.trackflaw.com/. Hackazon est une plateforme de test et d'évaluation de la sécurité, souvent utilisée pour simuler des scénarios d'attaques web afin d'améliorer les pratiques de sécurisation des applications.

Le test d'intrusion avait pour objectif d'identifier les vulnérabilités potentielles de l'application et de fournir des recommandations en matière de sécurité. Ce test s'inscrit dans une démarche d'amélioration continue de la sécurité de l'infrastructure et des applications exposées à des utilisateurs externes.

Objectifs principaux :

- Identifier et analyser les vulnérabilités présentes sur l'application web Hackazon.
- Évaluer la sécurité de l'infrastructure sous-jacente (serveurs, services réseau).
- Proposer des recommandations pour la remédiation des vulnérabilités détectées.

Portée du test :

Le test a principalement couvert deux aspects :

1. **L'infrastructure** : Évaluation de la configuration réseau, des services exposés, et des mécanismes de protection en place.
2. **L'application web** : Analyse des points d'entrée de l'application, de la gestion des identités, des sessions, et des mécanismes de validation des entrées utilisateurs.

Contraintes :

- Le temps alloué pour cette prestation était limité, ce qui a restreint l'analyse exhaustive de tous les points d'entrée possibles.
- Aucun accès aux codes sources de l'application ou aux serveurs hébergeant l'application n'a été fourni. Le test a été réalisé dans une approche « boîte noire », simulant l'attaque d'un utilisateur malveillant sans connaissances internes sur l'application.

1.3 Pilotage de la Prestation

Le pilotage de cette mission a suivi une approche structurée afin d'assurer une exécution fluide et alignée sur les attentes du client. Le test d'intrusion a été réalisé en plusieurs phases, chacune encadrée par des points de contact réguliers avec le client pour garantir la transparence et la bonne progression du projet.

Phases de la mission :

1. **Phase de préparation** :
    - Recueil des besoins du client et définition du périmètre du test.
    - Planification des outils et méthodes à utiliser pour le test d'intrusion.
    - Configuration d'un environnement sécurisé pour l'exécution des tests.
2. **Phase de tests** :
    - Réalisation des tests d'intrusion en suivant une approche **boîte noire**, simulant le comportement d'un attaquant sans accès aux informations internes.
    - Utilisation d'outils automatisés et manuels pour identifier les vulnérabilités potentielles, notamment :
        - Outils de scan de vulnérabilités (ex. **SQLMap**, **Nmap**).
        - Analyse manuelle des points d'entrée utilisateur et des services exposés.
3. **Phase d'analyse** :
    - Analyse approfondie des résultats obtenus durant les tests pour en extraire les vulnérabilités les plus critiques.
    - Classement des vulnérabilités selon leur impact, leur facilité d'exploitation et leur sévérité.
4. **Phase de restitution** :
    - Présentation des résultats sous forme de rapport détaillé, incluant les vulnérabilités détectées et les recommandations associées.
    - Discussion avec le client pour clarifier certains points, notamment les priorités en matière de remédiation.

Points de contact et communication :

- Un compte-rendu final a été livré sous forme de rapport détaillé, avec une synthèse managériale et une synthèse technique.

1.4 Actions de nettoyage recommandées

> FIXME: Recommendations for post-pentest cleanup actions, including removing test accounts, resetting passwords, or patching vulnerabilities.

2. Synthèse Managériale

2.1 Synthèse générale

> FIXME: General summary of the findings and their potential impact on the business.

2.2 Synthèse des risques

> FIXME: Overview of the risks identified, categorized by severity and potential impact.

2.3 Synthèse des vulnérabilités et recommandations

> FIXME: Summary of vulnerabilities found and the recommended actions to mitigate them.

2.4 Remarques

> FIXME: Any additional comments or notes for management.

3. Synthèse Technique

> FIXME: A detailed technical summary of the findings, highlighting specific vulnerabilities, misconfigurations, and security gaps in the Hackazon web application.

4. Test d'intrusion externe et applicatif

4.1 Évaluation infrastructure

4.1.1 Réseau

> FIXME: Findings from the network evaluation.

4.1.2 Services

> FIXME: Evaluation of the services exposed by the infrastructure.

4.2 Application web

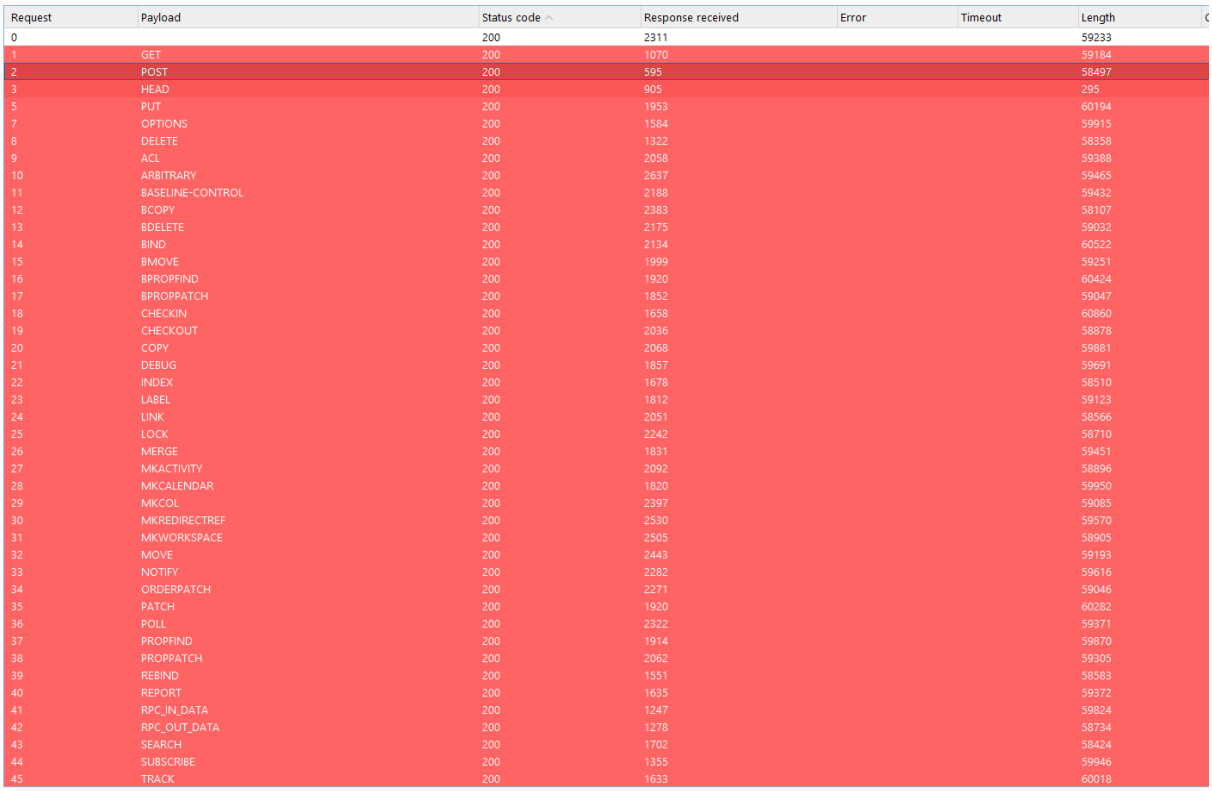
4.2.2 Évaluation application

Collecte d'informations

> FIXME: Results from the information gathering phase (e.g., recon).

VULN-04 : Acceptation de méthodes HTTP excessive			
État	**Impact**	**Difficulté d'exploitation**	**Sévérité**
Avérée	**Majeur**	**Facile**	**3 / 4**

Configuration et mécanismes de déploiement De nombreuses méthodes HTTP, telles que `PATCH`, `DELETE`, `TRACE`, et d'autres, sont acceptées par l'application, ce qui élargit la surface d'attaque pour les attaquants potentiels. Cette configuration peut permettre des actions non désirées telles que la modification, la suppression, ou la découverte d'informations sensibles.



Request	Payload	Status code	Response received	Error	Timeout	Length
0		200	2311			59233
1	GET	200	1070			59184
2	POST	200	595			58497
3	HEAD	200	905			295
5	PUT	200	1953			60194
7	OPTIONS	200	1584			59915
8	DELETE	200	1322			58358
9	ACL	200	2058			59388
10	ARBITRARY	200	2637			59465
11	BASELINE-CONTROL	200	2188			59432
12	BCOPY	200	2383			58107
13	BDELETE	200	2175			59032
14	BIND	200	2134			60522
15	BMOVE	200	1999			59251
16	BPROPFIND	200	1920			60424
17	BPROPPATCH	200	1852			59047
18	CHECKIN	200	1658			60860
19	CHECKOUT	200	2036			58878
20	COPY	200	2068			59881
21	DEBUG	200	1857			59691
22	INDEX	200	1678			58510
23	LABEL	200	1812			59123
24	LINK	200	2051			58566
25	LOCK	200	2242			58710
26	MERGE	200	1831			59451
27	MKACTIVITY	200	2092			58896
28	MKCALENDAR	200	1820			59950
29	MKCOL	200	2397			59085
30	MKREDIRECTREF	200	2530			59570
31	MKWORKSPACE	200	2505			58905
32	MOVE	200	2443			59193
33	NOTIFY	200	2282			59616
34	ORDERPATCH	200	2271			59046
35	PATCH	200	1920			60282
36	POLL	200	2322			59371
37	PROPFIND	200	1914			59870
38	PROPPATCH	200	2062			59305
39	REBIND	200	1551			58583
40	REPORT	200	1635			59372
41	RPC_IN_DATA	200	1247			59824
42	RPC_OUT_DATA	200	1278			58734
43	SEARCH	200	1702			58424
44	SUBSCRIBE	200	1355			59946
45	TRACK	200	1633			60018

Figure 1: Capture d'écran de la réponse HTTP lors d'une attaque de fuzzing des méthodes HTTP.

Remediation

VULN-04 : Acceptation de méthodes HTTP excessive		
Complexité estimée : Faible	**Travail/coût estimé : Faible**	**Priorité estimée : 3 / 4**
Il est recommandé de restreindre l'acceptation des méthodes HTTP aux seules méthodes strictement nécessaires, comme 'GET' et 'POST'. Les méthodes non nécessaires comme 'DELETE', 'TRACE', et autres doivent être désactivées côté serveur pour réduire la surface d'attaque potentielle.		

VULN-05 : Absence de protection anti-malware			
État	**Impact**	**Difficulté d'exploitation**	**Sévérité**
Avérée	**Critique**	**Facile**	**4 / 4**

L'application ne dispose pas de protection contre les logiciels malveillants. Un fichier de test EICAR, utilisé pour simuler un fichier malveillant, a pu être téléchargé sans être détecté ou bloqué. Cela expose

l'application à des risques d'infection par des logiciels malveillants pouvant entraîner la compromission du serveur.

```
www-data@11eb2768af20:/var/www/html/web/user_pictures/01$ hostname
hostname
11eb2768af20
www-data@11eb2768af20:/var/www/html/web/user_pictures/01$ cat eicar.txt
cat eicar.txt
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
www-data@11eb2768af20:/var/www/html/web/user_pictures/01$ ls -la
ls -la
total 16
drwxr-xr-x   2 www-data www-data 4096 Oct 13 14:23 .
drwxr-xr-x 122 www-data www-data 4096 Oct 13 14:01 ..
-rw-r--r--   1 www-data www-data  325 Oct 13 14:01 Webshell.php
-rw-r--r--   1 www-data www-data   69 Oct 13 14:23 eicar.txt
```

Figure 2: Capture d'écran montrant le fichier EICAR téléchargé et accessible sur le serveur.

Remediation

VULN-05 : Absence de protection anti-malware		
Complexité estimée : Moyenne	**Travail/coût estimé : Moyen**	**Priorité estimée : 4 / 4**
Il est fortement recommandé d'implémenter une solution de protection anti-malware sur le serveur pour scanner les fichiers téléchargés. Cela peut inclure un antivirus tel que ClamAV, qui peut détecter et bloquer les fichiers malveillants comme le fichier de test EICAR. De plus, la surveillance régulière des fichiers et des processus sur le serveur doit être mise en place pour prévenir les infections.		

Insecure File Distribution

VULN-09 : Distribution de fichier non sécurisé			
État	**Impact**	**Difficulté d'exploitation**	**Sévérité**
Avérée	**Modéré**	**Facile**	**3 / 4**

Il a été observé que l'application distribue des fichiers APK sans aucune validation ou signature de sécurité. Le fichier téléchargé via l'URL `https://hackazon.trackflaw.com/app/hackazon.apk` n'est pas signé, ce qui peut permettre à un attaquant de distribuer des fichiers malveillants en remplacement des fichiers légitimes.

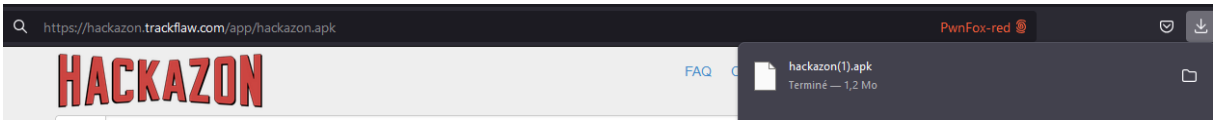


Figure 3: Capture d'écran montrant le téléchargement du fichier APK non sécurisé.

Remediation

VULN-09 : Distribution de fichier non sécurisé		
Complexité estimée : Moyenne	**Travail/coût estimé : Moyen**	**Priorité estimée : 3 / 4**
Il est recommandé de signer numériquement tous les fichiers distribués, en particulier les fichiers exécutables comme les APK. De plus, l'application doit vérifier l'intégrité des fichiers avant leur distribution pour éviter tout risque de remplacement par des fichiers malveillants.		

Public Exposure of Admin Panel

VULN-11 : Exposition du panneau d'administration public			
État	**Impact**	**Difficulté d'exploitation**	**Sévérité**
Avérée	**Critique**	**Facile**	**4 / 4**

L'application expose publiquement l'URL de connexion au panneau d'administration, accessible via `https://hackazon.trackflaw.com/admin/user/login`. Cette URL peut être exploitée par des attaquants pour tenter des attaques de force brute ou d'énumération d'utilisateurs.

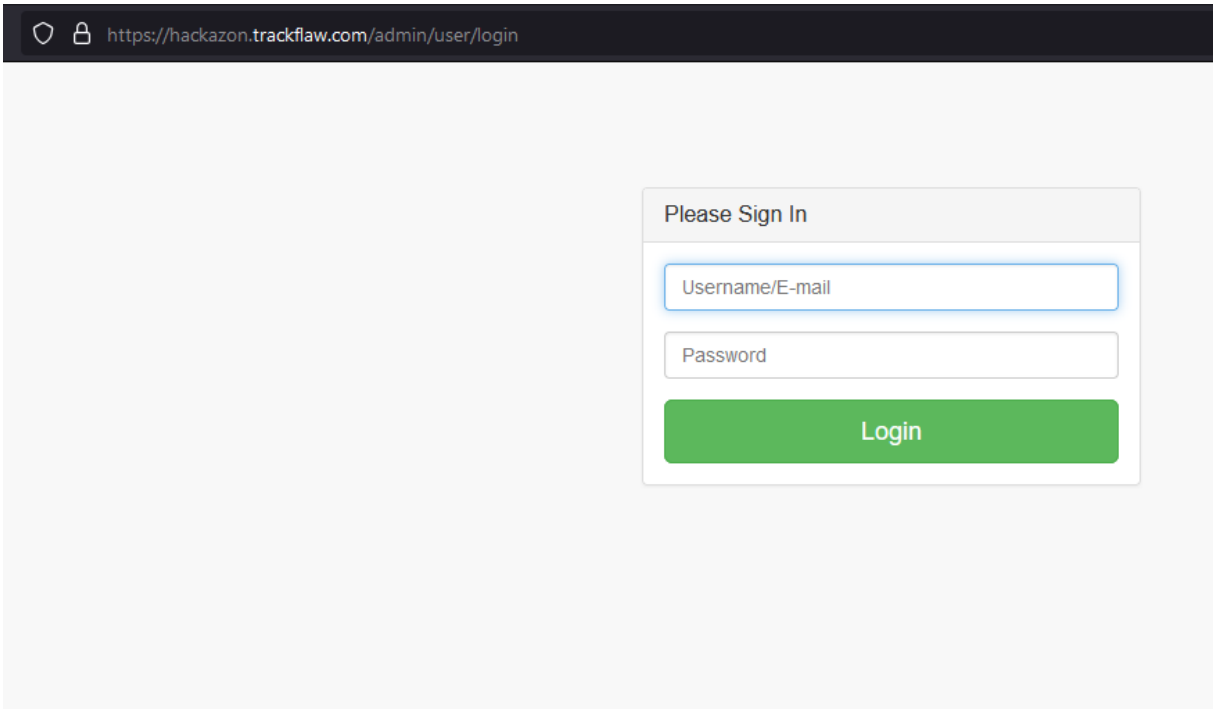


Figure 4: Capture d'écran montrant le panneau de connexion de l'administration.

Request :

```
1  GET /admin/user/login HTTP/2
2  Host: hackazon.trackflaw.com
3  Cookie: visited_products=%2C64%2C72%2C1%2C81%2C; PHPSESSID=
       XXXXXXXXXXXXXXXXXXXXX
4  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0) Gecko
       /20100101 Firefox/131.0
5  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/
       avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
6  Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
7  Accept-Encoding: gzip, deflate, br
8  Upgrade-Insecure-Requests: 1
9  Sec-Fetch-Dest: document
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-Site: none
12 Sec-Fetch-User: ?1
13 X-Pwnfox-Color: red
14 Priority: u=0, i
15 Te: trailers
```

Remediation

VULN-11 : Exposition du panneau d'administration public		
Complexité estimée : Moyenne	**Travail/coût estimé : Faible**	**Priorité estimée : 4 / 4**
Il est recommandé de restreindre l'accès au panneau d'administration via des règles de contrôle d'accès basées sur les adresses IP, ou de déplacer le panneau d'administration à une URL obscure. De plus, des mécanismes de protection contre les attaques par force brute, comme des CAPTCHA ou la limitation des tentatives de connexion, devraient être mis en place.		

Gestion des identités

> FIXME: Assessment of identity management and user roles.

VULN-08 : Détection de mots de passe utilisateur			
État	**Impact**	**Difficulté d'exploitation**	**Sévérité**
Avérée	**Critique**	**Facile**	**4 / 4**

Lors de l'analyse du mécanisme d'authentification, plusieurs mots de passe utilisateur ont été découverts à travers une attaque par force brute ou fuzzing, ce qui démontre un manque de robustesse dans la politique de gestion des mots de passe. Les mots de passe suivants ont été détectés, comme le montre la capture d'écran ci-dessous.

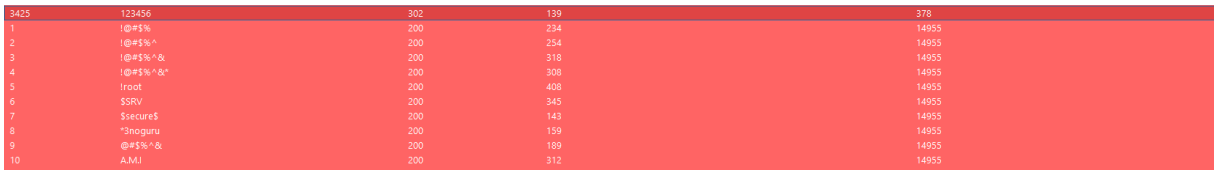


Figure 5: Capture d'écran montrant les mots de passe détectés lors du fuzzing.

Remediation

Authentification

VULN-08 : Détection de mots de passe utilisateur		
Complexité estimée : Moyenne	**Travail/coût estimé : Moyen**	**Priorité estimée : 4 / 4**
Il est fortement recommandé de mettre en œuvre une politique de mots de passe robustes comprenant des critères comme la longueur minimale, la complexité (caractères spéciaux, chiffres, lettres majuscules/minuscules), ainsi que la mise en place d'un mécanisme de limitation des tentatives de connexion pour prévenir les attaques par force brute.		

> FIXME: Review of authentication mechanisms (e.g., password policies, multi-factor authentication).

Autorisations

> FIXME: Evaluation of authorization checks and privilege separation.

VULN-03 : Absence d'expiration de session			
État	**Impact**	**Difficulté d'exploitation**	**Sévérité**
Avérée	**Majeur**	**Facile**	**3 / 4**

Gestion des sessions En vérifiant les sécurités des cookies, on peut apercevoir que les sessions n'ont pas d'expiration.

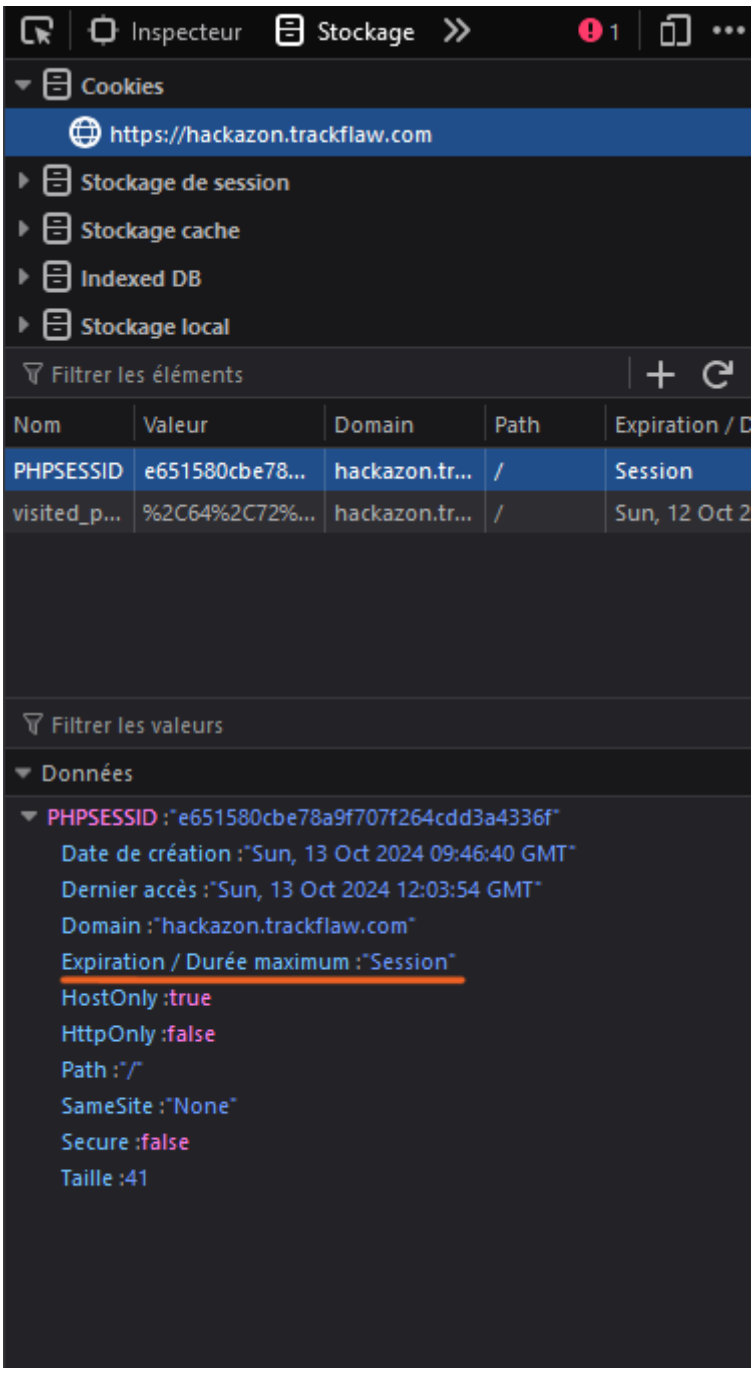


Figure 6: Screenshot des paramètres des cookies de session après un login successful.

En analysant les cookies de session de l'application, il a été constaté que le cookie de session PHP n'était pas protégé par les attributs `Secure`, `HttpOnly`, et `SameSite`. Cela rend le cookie vulnérable aux attaques telles que le vol de session via un réseau non sécurisé ou les attaques Cross-Site Scripting

VULN-03 : Absence d'expiration de session		
Complexité estimée : Faible	**Travail/coût estimé : Faible**	**Priorité estimée : 3 / 4**
Il est recommandé d'implémenter une politique d'expiration des sessions qui invalide les sessions après une période d'inactivité définie (ex : 15 minutes). Cela peut être combiné avec des techniques comme le renouvellement des cookies et des notifications d'expiration.		

VULN-07 : Cookie de Session sans Attributs Secure, HttpOnly, et SameSite			
État	**Impact**	**Difficulté d'exploitation**	**Sévérité**
Avérée	**Majeur**	**Facile**	**3 / 4**

(XSS).

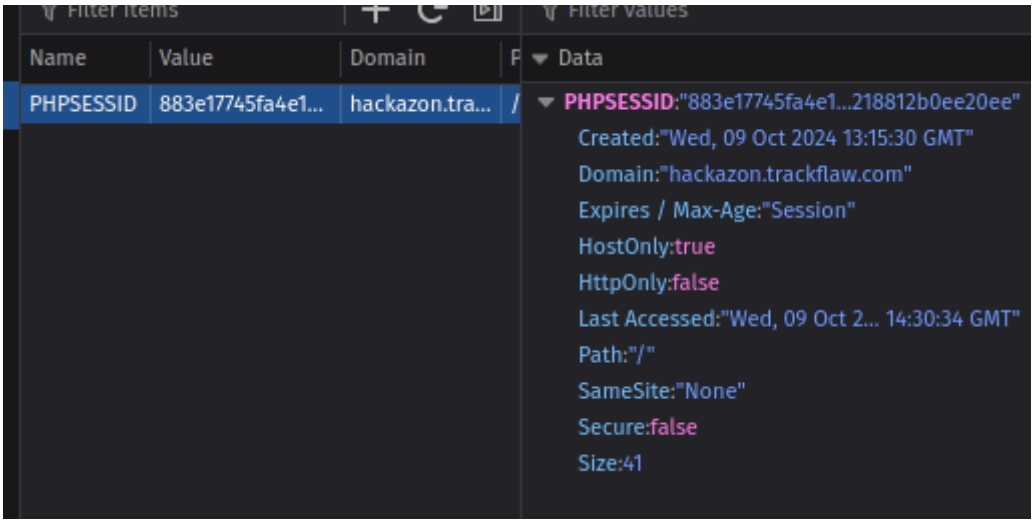


Figure 7: Capture d'écran du cookie de session sans attributs de sécurité.

Remediation

Validation des entrées utilisateurs En analysant l'URL `https://hackazon.trackflaw.com/account/documents?page=terms.html;id`, on constate que le paramètre `page` est vulnérable à une injection de commandes système. Cette vulnérabilité permet à un attaquant d'exécuter des commandes directement sur le serveur.

VULN-07 : Cookie de Session sans Attributs Secure, HttpOnly, et SameSite		
Complexité estimée : Faible	**Travail/coût estimé : Faible**	**Priorité estimée : 3 / 4**
Il est recommandé de configurer les cookies de session avec les attributs 'Secure', 'HttpOnly', et 'SameSite'. L'attribut 'Secure' garantit que les cookies ne sont envoyés que via une connexion HTTPS, 'HttpOnly' empêche les scripts côté client d'accéder aux cookies, et 'SameSite' empêche les attaques Cross-Site Request Forgery (CSRF).		

VULN-06 : Injection de commandes			
État	**Impact**	**Difficulté d'exploitation**	**Sévérité**
Avérée	**Critique**	**Facile**	**4 / 4**

Requête HTTP

```
GET /account/documents?page=terms.html;id HTTP/2
Host: hackazon.trackflaw.com
Cookie: PHPSESSID=XXXXXXXXXXXXXXXXXXXXXXXX; visited_products=%2C1%2C208
    %2C15%2C101%2C81%2C21%2C
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:131.0) Gecko/20100101
    Firefox/131.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/
    avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
X-Pwnfox-Color: blue
Priority: u=0, i
Te: trailers
```

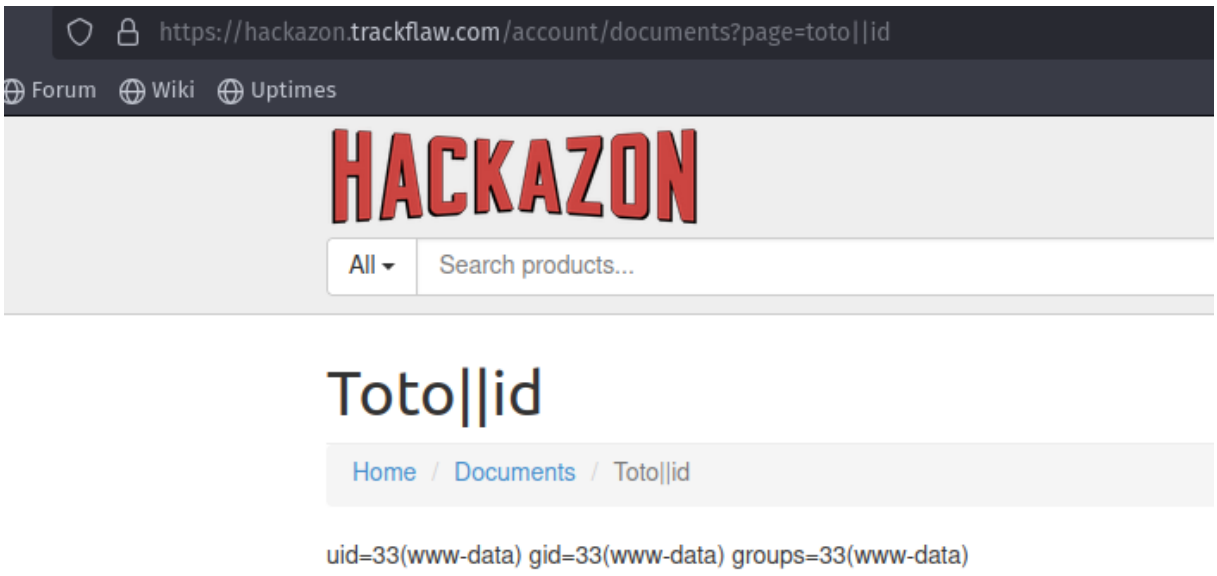


Figure 8: Capture d'écran montrant la commande exécutée avec succès.

Remediation

VULN-06 : Injection de commandes		
Complexité estimée : Moyenne	**Travail/coût estimé : Élevé**	**Priorité estimée : 4 / 4**
Il est recommandé de valider et d'assainir strictement tous les paramètres passés dans les URLs, en particulier ceux qui interagissent avec des commandes système. Des mécanismes comme l'utilisation de bibliothèques sécurisées ou l'échappement des caractères spéciaux devraient être appliqués pour éviter toute injection.		

User Enumeration

VULN-10 : Énumération d'utilisateur			
État	**Impact**	**Difficulté d'exploitation**	**Sévérité**
Avérée	**Modéré**	**Facile**	**3 / 4**

L'application divulgue des informations sensibles lors de l'enregistrement ou de la récupération de mot de passe pour les utilisateurs. Lorsqu'un utilisateur tente de s'inscrire ou de récupérer un mot

de passe, l'application révèle si l'adresse email ou le nom d'utilisateur existe déjà, permettant à un attaquant de cartographier les comptes existants.

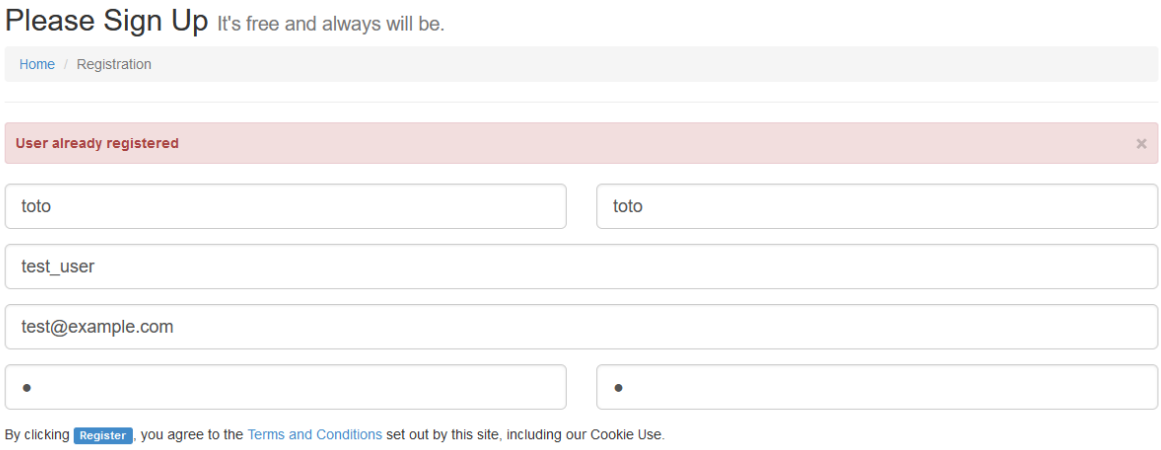


Figure 9: Capture d'écran montrant la divulgation des informations d'enregistrement et de récupération.

```
POST /user/register HTTP/2
Host: hackazon.trackflaw.com
Cookie: visited_products=%2C64%2C72%2C1%2C81%2C; PHPSESSID=
    XXXXXXXXXXXXXXXXXX
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0) Gecko
    /20100101 Firefox/131.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/
    avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 109
Origin: https://hackazon.trackflaw.com
Referer: https://hackazon.trackflaw.com/user/register
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
X-Pwnfox-Color: red
Priority: u=0, i
Te: trailers

first_name=toto&last_name=toto&username=test_user&email=test%40example.
    com&password=1&password_confirmation=1
```

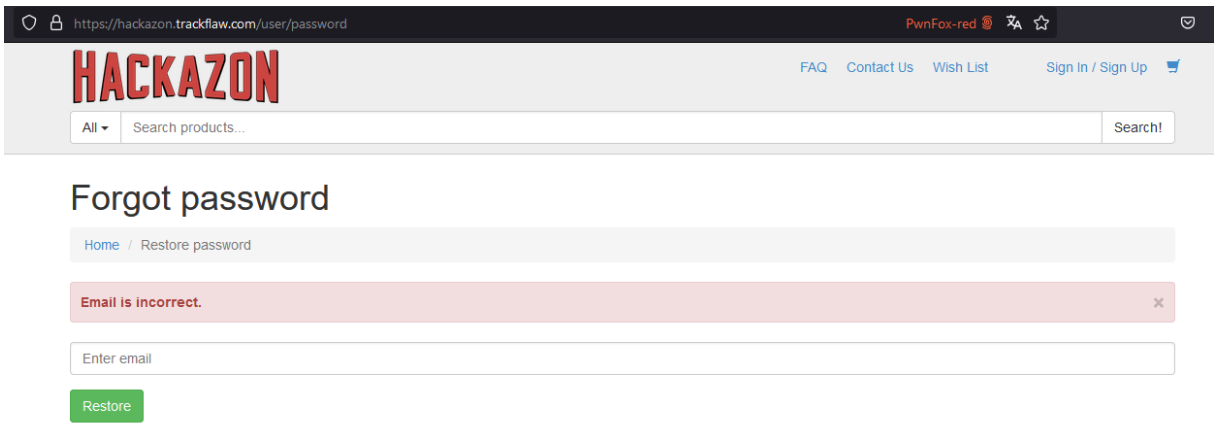


Figure 10: Capture d'écran montrant la divulgation des informations d'enregistrement et de récupération.

```
POST /user/password HTTP/2
Host: hackazon.trackflaw.com
Cookie: visited_products=%2C64%2C72%2C1%2C81%2C; PHPSESSID=
    XXXXXXXXXXXXX
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0) Gecko
    /20100101 Firefox/131.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/
    avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 18
Origin: https://hackazon.trackflaw.com
Referer: https://hackazon.trackflaw.com/user/password
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
X-Pwnfox-Color: red
Priority: u=0, i
Te: trailers

email=toto@tata.fr
```

Remediation

VULN-10 : Énumération d'utilisateur		
Complexité estimée : Moyenne	**Travail/coût estimé : Faible**	**Priorité estimée : 3 / 4**
Il est recommandé de ne pas révéler d'informations spécifiques concernant l'existence d'un utilisateur. En cas de tentative d'enregistrement ou de récupération de mot de passe, le message d'erreur doit être générique (ex : "Une erreur est survenue."). Cela permet de protéger la vie privée des utilisateurs et de rendre plus difficile la cartographie des comptes existants.		

Open Redirect

Gestion des erreurs

> FIXME: Review of how the application handles errors (e.g., verbose error messages).

Cryptographie

> FIXME: Use of cryptographic methods (e.g., encryption, hashing).

Processus métier

> FIXME: Assessment of business logic flaws.

Côté client

> FIXME: Client-side vulnerabilities (e.g., JavaScript security, DOM-based XSS).

5. Annexe

5.1 Présentation de la démarche

La démarche adoptée pour ce test d'intrusion s'inscrit dans une méthodologie de sécurité éprouvée, basée sur les bonnes pratiques en matière de tests de pénétration. Ce test a été réalisé en suivant une approche **boîte noire**, simulant un attaquant sans connaissance préalable des infrastructures internes de l'application Hackazon.

Le test s'est déroulé en plusieurs étapes, chaque phase étant conçue pour identifier et exploiter les vulnérabilités potentielles dans l'infrastructure et l'application web.

Méthodologie suivie :

1. **Collecte d'informations** (*Reconnaissance*) :
    - L'objectif de cette première phase est d'acquérir le maximum d'informations sur l'infrastructure et l'application ciblée. Des techniques de reconnaissance passive et active ont été employées pour découvrir les technologies utilisées, les points d'entrée potentiels, ainsi que les services exposés.
    - Outils utilisés : **Nmap**, **Whois**, **Google Dorking**, et divers outils de reconnaissance open-source.
2. **Analyse des vulnérabilités** (*Scanning*) :
    - Cette phase consiste à identifier les vulnérabilités potentielles sur les services exposés et les points d'entrée de l'application web. Un audit approfondi a été réalisé pour détecter des failles telles que les injections SQL, les failles XSS, les mauvaises configurations de serveur, ou encore la gestion incorrecte des sessions.
    - Outils utilisés : **Burp Suite**, **OWASP ZAP**, **SQLMap**.
3. **Exploitation des vulnérabilités** (*Exploitation*) :
    - Lors de cette étape, les vulnérabilités détectées sont exploitées afin de démontrer leur impact réel. Cela inclut l'extraction de données sensibles, la compromission de comptes utilisateurs, ou encore le contournement des mécanismes de sécurité.
    - Des preuves de concept (PoC) ont été fournies pour les vulnérabilités les plus critiques afin de montrer leur faisabilité.
4. **Post-exploitation et recommandations** :
    - Une fois les vulnérabilités exploitées, une analyse plus approfondie est réalisée pour déterminer l'étendue des dommages potentiels. Cette phase permet également de formuler des recommandations précises sur les correctifs à apporter pour chaque vulnérabilité identifiée.
    - Outils utilisés : **SQLMap** pour la récupération des bases de données, **Burp Suite** pour l'analyse des réponses serveur.

Limites et contraintes :

- Le test a été réalisé dans des conditions de temps limitées, ce qui a restreint l'exploration exhaustive de toutes les fonctionnalités de l'application.

- L'approche **boîte noire** ne permet pas d'explorer certaines vulnérabilités internes ou logicielles, qui auraient pu être visibles avec un accès direct au code source ou aux environnements de développement.

5.2 Présentation des résultats

> FIXME: Additional detailed results, if necessary.

5.3 Terminologie des risques

> FIXME: Glossary of risk-related terms used in the report.