Rapport de Test d'Intrusion

(FramaFTP)

Sommaire:

1. Introduction

Contexte

Le leader du libre FramaFTP propose un service de stockage gratuit et accessible à tous. En tant que consultant en sécurité, nous avons été chargés de mener un audit de sécurité sur leur infrastructure pour identifier et corriger les vulnérabilités potentielles.

Objectifs du test d'intrusion

L'objectif principal du test d'intrusion est de découvrir et démontrer un maximum de failles de sécurité présentes dans l'infrastructure de FramaFTP. Cela inclut l'identification des vulnérabilités, l'évaluation des risques associés et la proposition de recommandations pour renforcer la sécurité.

Périmètre du test d'intrusion

Il est essentiel de préciser que ce test de pénétration a été effectué dans un cadre strictement défini et avec l'approbation explicite de FramaFTP. Toutes les actions faites lors de ce test ont été validées en amont par les responsables de l'entreprise, garantissant ainsi que nos activités restent dans les limites autorisées. Il est important de souligner qu'aucune donnée réelle n'a été mise en danger au cours de ce processus.

Le périmètre de ce test d'intrusion inclut l'adresse IP suivante :

- 172.16.239.10

2. Synthèse Managériale

Évaluation du niveau de risque du périmètre

L'évaluation du niveau de risque de l'infrastructure FramaFTP révèle plusieurs vulnérabilités critiques et hautes. Ces vulnérabilités exposent l'infrastructure à des risques élevés de compromission, d'accès non autorisé, et de fuite de données sensibles.

Mise en avant des risques métier associés aux vulnérabilités

Les vulnérabilités découvertes peuvent avoir des impacts significatifs sur les opérations de FramaFTP :

- **Failles d'identification et d'authentification** : Mots de passe faibles, permettant un accès non autorisé.
- **Contrôle d'accès défaillant** : Serveur FTP exposé et possibilité de modification de fichiers critiques.
- **Conception peu sûre** : Exposition d'informations sensibles du serveur.
- **Inclusion de Fichiers Locaux (LFI)** : Permet à un attaquant de lire des fichiers sensibles sur le serveur.

Explication des Injections LFI

Inclusion de Fichiers Locaux (LFI) :

- **Qu'est-ce que c'est ?** L'inclusion de fichiers locaux est une vulnérabilité où un attaquant peut accéder et lire des fichiers sur le serveur en manipulant les paramètres d'URL. Cela peut exposer des informations sensibles, comme des mots de passe, des configurations, etc.
- **Comment ça marche ?** L'attaquant modifie les paramètres d'URL pour inclure un fichier local. Par exemple, en accédant à une URL telle que `http://172.16.239.10/index.php?p=../../../../../../etc/passwd`, l'attaquant peut lire le contenu du fichier `/etc/passwd`, qui contient des informations sur les utilisateurs du système.
- **Exemple découvert** : Nous avons pu accéder au fichier `passwd` en utilisant une URL modifiée, révélant des informations sensibles sur les utilisateurs du système.

Cross-Site Scripting (XSS) :

- **Qu'est-ce que c'est ?** Le Cross-Site Scripting (XSS) est une vulnérabilité de sécurité qui permet à un attaquant d'injecter des scripts malveillants dans une page web consultée par d'autres utilisateurs.
- **Comment ça marche ?** L'attaquant insère du code JavaScript malveillant dans un formulaire ou une URL. Lorsque les utilisateurs visitent la page compromise, le script s'exécute dans leur navigateur, permettant à l'attaquant de voler des cookies, de rediriger les utilisateurs vers des sites malveillants ou de manipuler le contenu de la page.

Cross-Site Request Forgery (CSRF) :

- **Qu'est-ce que c'est ?** Le Cross-Site Request Forgery (CSRF) est une attaque où un utilisateur malveillant incite l'utilisateur authentifié d'une application web à exécuter des actions non souhaitées sur cette application.
- **Comment ça marche ?** L'attaquant trompe l'utilisateur en cliquant sur un lien malveillant ou en visitant une page qui envoie des requêtes non autorisées à l'application web où l'utilisateur est déjà authentifié. Ces requêtes sont exécutées avec les privilèges de l'utilisateur authentifié, ce qui peut entraîner des actions non désirées comme des transferts d'argent, des changements de mot de passe, etc.

Injection SQL (SQLi) :

- **Qu'est-ce que c'est ?** L'injection SQL est une vulnérabilité de sécurité qui permet à un attaquant d'interférer avec les requêtes SQL que l'application envoie à la base de données.
- **Comment ça marche ?** L'attaquant insère du code SQL malveillant dans un champ de saisie, qui est ensuite exécuté par la base de données. Cela peut permettre à l'attaquant de visualiser, modifier ou supprimer des données sensibles, ou même d'exécuter des commandes administratives sur la base de données.

Ces vulnérabilités représentent des risques importants pour la sécurité des applications web, car elles permettent aux attaquants de compromettre la confidentialité, l'intégrité et la disponibilité des données.

3. Synthèse Technique

Liste des vulnérabilités avec le niveau de risque associé

1. **Identification et Authentification Failures** (Critique)
    - Mots de passe faibles acceptés lors de l'enregistrement.
2. **Broken Access Control** (Haut)
    - Accès et modification de fichiers via FTP sans restriction adéquate.
3. **Insecure Design** (Faible)
    - Exposition publique des informations du serveur via phpinfo().

Liste des recommandations avec la difficulté de correction

1. **Renforcer les politiques de mot de passe** (Moyen)
    - Exiger des mots de passe complexes et appliquer une politique de renouvellement régulier.
2. **Restreindre les accès FTP** (Élevé)
    - Mettre en place des contrôles d'accès stricts et utiliser des connexions sécurisées.
3. **Masquer les informations sensibles du serveur** (Faible)
    - Désactiver les pages d'information publique comme phpinfo().

Résumé chronologique et méthodologique des tests

1. **Recherche dans le domaine publique**

    Avant toutes choses, des recherches ont été faites sur internet pour trouver le plus d'informations possible sur FramaFTP (172.16.239.10)

    a. WhoIS : {"result":{"code":403,"label":"Refresh the web page"}}
    b. Pas de réseaux sociaux
    c. Tentative de certains Google Dork comme :
        i. FramaFTP Port 1..65000 : aucun de résultat
        ii. site: FramaFTP
        iii. ip: 172.16.239.10 :
        iv. (172.16.239.10 - IP address is in private non-routable range.
        v. 172.16.239.10 - IP address is in a reserved range.)
    d. DnsDumpster : aucun résultat.
    e. Github ne contient rien concernant FramaFTP.
    f. Shodan : il n'y a rien concernant FramaFTP sur ce site aussi.

2. **Reconnaissance initiale**
    a. Utilisation de nmap et patator pour découvrir les ports ouverts et tester les identifiants faibles. Nous avons pu avoir certaines informations grâce à nmap mais à ce stade avec patator.py nous n'avions rien trouvé de particulier.
3. **Accès initial**
    a. Enregistrement sur le site avec un mot de passe faible.
    b. Connexion FTP et exploration des répertoires accessibles. Via le compte que nous avions crée nous n'avions accès seulement au répertoire courant. Pas de connexion SSH disponible avec notre compte.
4. **Tests d'injection**
    a. Tentatives d'injection XSS, CSRF et SQL.
    b. Le reflected XSS est bel et bien protégé
    c. Injection dans formulaire ne marche pas avec une tentative de script
    d. Pas de vulnérabilité CRSF
    e. Exploitation de la faille LFI pour accéder à des fichiers sensibles.
    f. Accès à des fichiers tel que :
        i. http://172.16.239.10/index.php?p=../../../../../../etc/passwd
        ii. http://172.16.239.10/index.php?p=../../../../../../ftp/test/phpinfo.php
        iii. http://172.16.239.10/index.php?p=../../../../../../etc/hosts
        iv. http://172.16.239.10/index.php?p=../../../../../../home/blu/.ssh/authorized_keys
    g. Création et exécution de scripts via LFI :

        Utilisation d'un script pour exécuter des commandes système depuis le navigateur, en passant par Burp Suite. Par exemple, un script PHP a été utilisé pour exécuter des commandes système, ce qui permet de lancer un reverse shell et d'exécuter des commandes arbitraires sur le serveur cible.

5. **Escalade de privilèges**
    a. Utilisation de fichiers SUID et scripts potentiellement exploitables.
    b. Brute force SSH avec patator.py pour accéder au compte 'michel'. Grâce à des dictionnaires de mot de passe que tout le monde peut trouver sur Github par rapport aux mots de passes les plus faibles, ou les plus utilisées par des utilisateurs. On a pu finalement trouver le mot de passe de Michel et accéder à son SSH. Nous avons remarqué des fichiers tel que healthcheck.sh qui est utile pour surveiller l'accessibilité et la disponibilité du serveur FramaFTP, et peut être exécuté régulièrement via des tâches cron pour assurer une surveillance continue.
    c. Nous avons aussi vu un fichier chez l'utilisateur "blu" qui permettait d'utiliser des crons (crons.hourly, crons.daily, crons.weekly , …) et qui, avec une insertion de script malveillant permettait de pouvoir utiliser des privilèges root sans son consentement,.

Rapport Technique Détaillé

Vulnérabilité 1 : Identification et Authentification Failures

Description : Le système accepte des mots de passe faibles lors de l'enregistrement des utilisateurs. **Détails d'exploitation** : Inscription avec un mot de passe faible comme ‘test’. **Évaluation de la criticité** : Critique **Scénario d'exploitation** : Un attaquant peut s'enregistrer avec un mot de passe faible et accéder au système.

Pour cela il faut juste ;

1. Aller sur l’url http://172.16.239.10/index.php?p=register.php
2. Puis utiliser un nom d’utilisateur avec des lettres et n’importe quel mot de passe non vide (même une lettre)

   https://watch.wave.video/NI641jmM6ZqIWtiZ (vidéo du test)

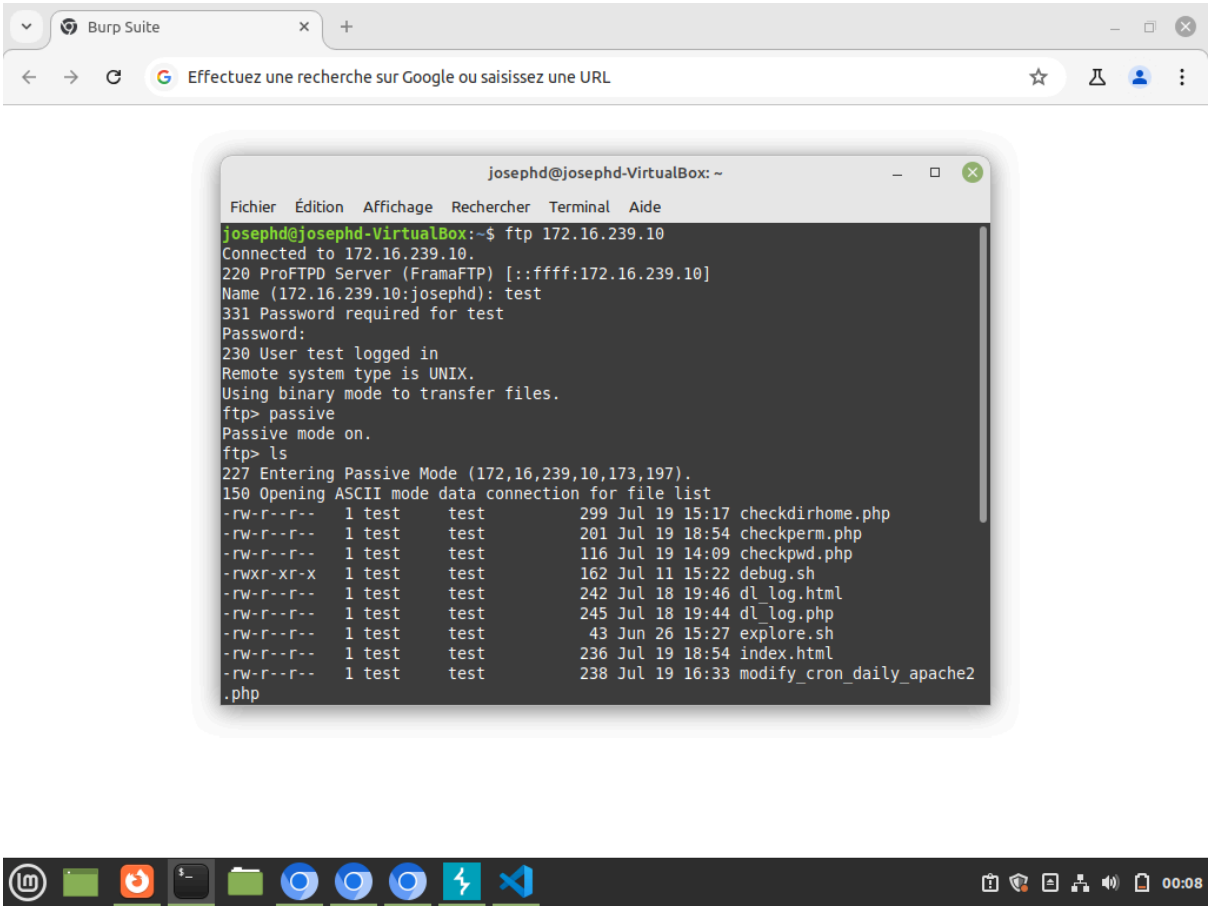
Recommandations : Implémenter une politique de mot de passe forte.

Vulnérabilité 2 : Broken Access Control

Description : Le serveur FTP permet l'accès et la modification de fichiers sans restrictions adéquates. **Détails d'exploitation** : Connexion FTP en mode passif, création et modification de fichiers. **Évaluation de la criticité** : Haute **Scénario d'exploitation** : Un attaquant peut modifier des fichiers critiques du système.

Pour cela il faut:

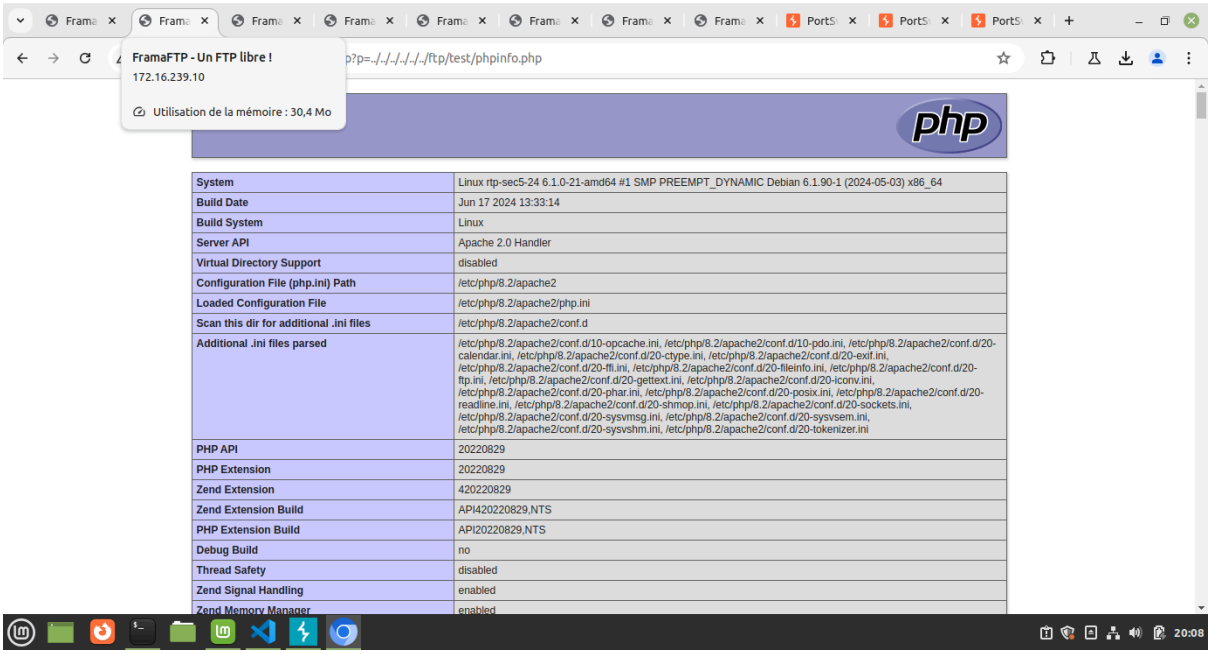
1. Créer un user (par exemple test) avec un mot de passe sur l’url de http://172.16.239.10/index.php?p=register.php
2. Aller dans un terminal et se connecter en ssh via “ftp 172.16.239.10”
3. Puis rentrer l’utilisateur test puis son mot de passe et se mettre en mode passive.



Recommandations : Mettre en place des contrôles d'accès stricts et utiliser des connexions sécurisées.

Vulnérabilité 3 : Insecure Design

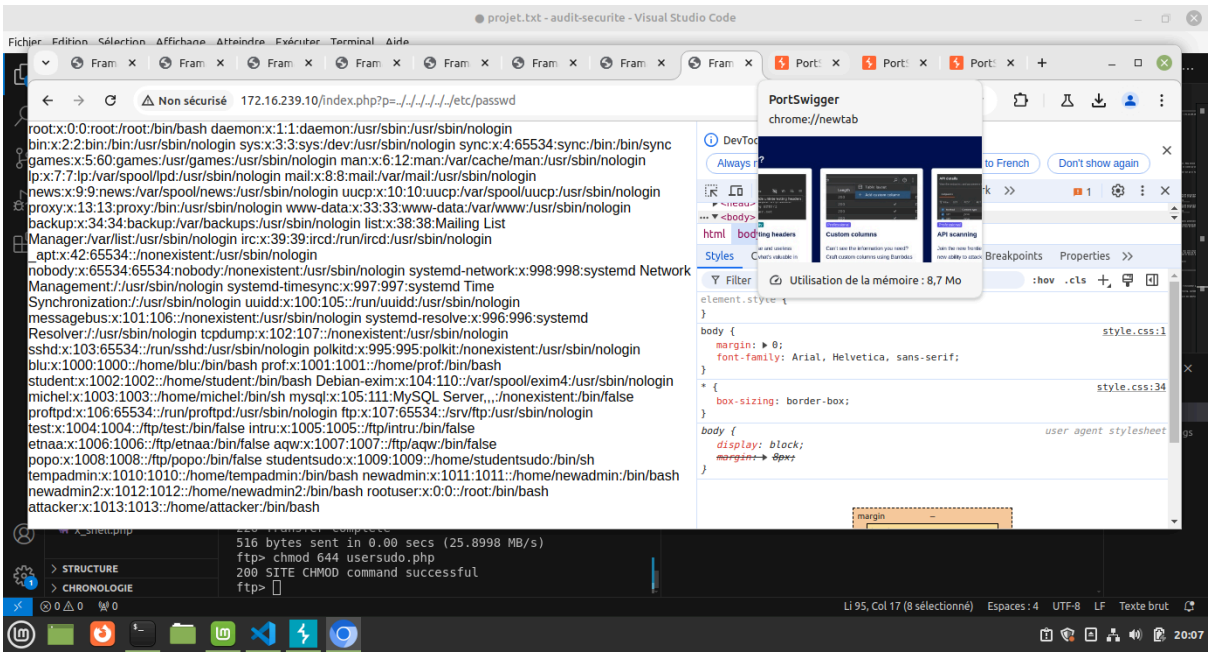
Description : Les informations du serveur sont exposées publiquement via phpinfo().
Détails d'exploitation : Accès à l'URL `http://172.16.239.10/index.php?p=../../../../../../ftp/test/phpinfo.php`. **Évaluation de la criticité** : Faible **Scénario d'exploitation** : Un attaquant peut recueillir des informations sensibles sur la configuration du serveur. Il a juste a tapé l'url selon l'utilisateur créé sur le site.



Recommandations : Désactiver les pages d'information publique comme phpinfo().

Vulnérabilité 4 : Local File Inclusion (LFI)

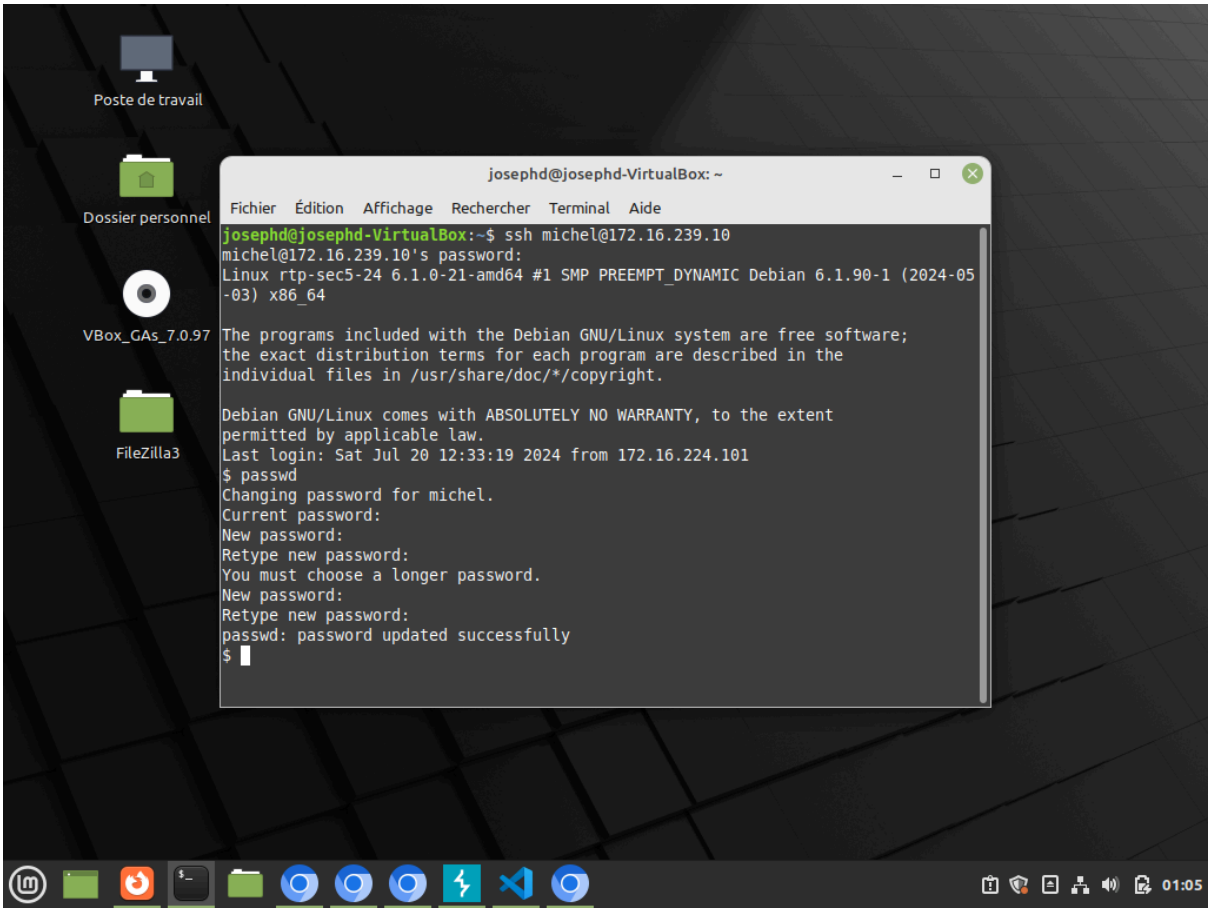
Description : Faille LFI permettant l'accès à des fichiers sensibles. **Détails d'exploitation** : Accès à l'URL `http://172.16.239.10/index.php?p=../../../../../../etc/passwd`.
Évaluation de la criticité : Critique **Scénario d'exploitation** : Un attaquant peut lire des fichiers sensibles et compromettre le système. Il a juste a tapé : http://172.16.239.10/index.php?p=../../../../../../etc/passwd



Recommandations : Mettre en place des contrôles de validation des entrées pour prévenir les inclusions de fichiers.

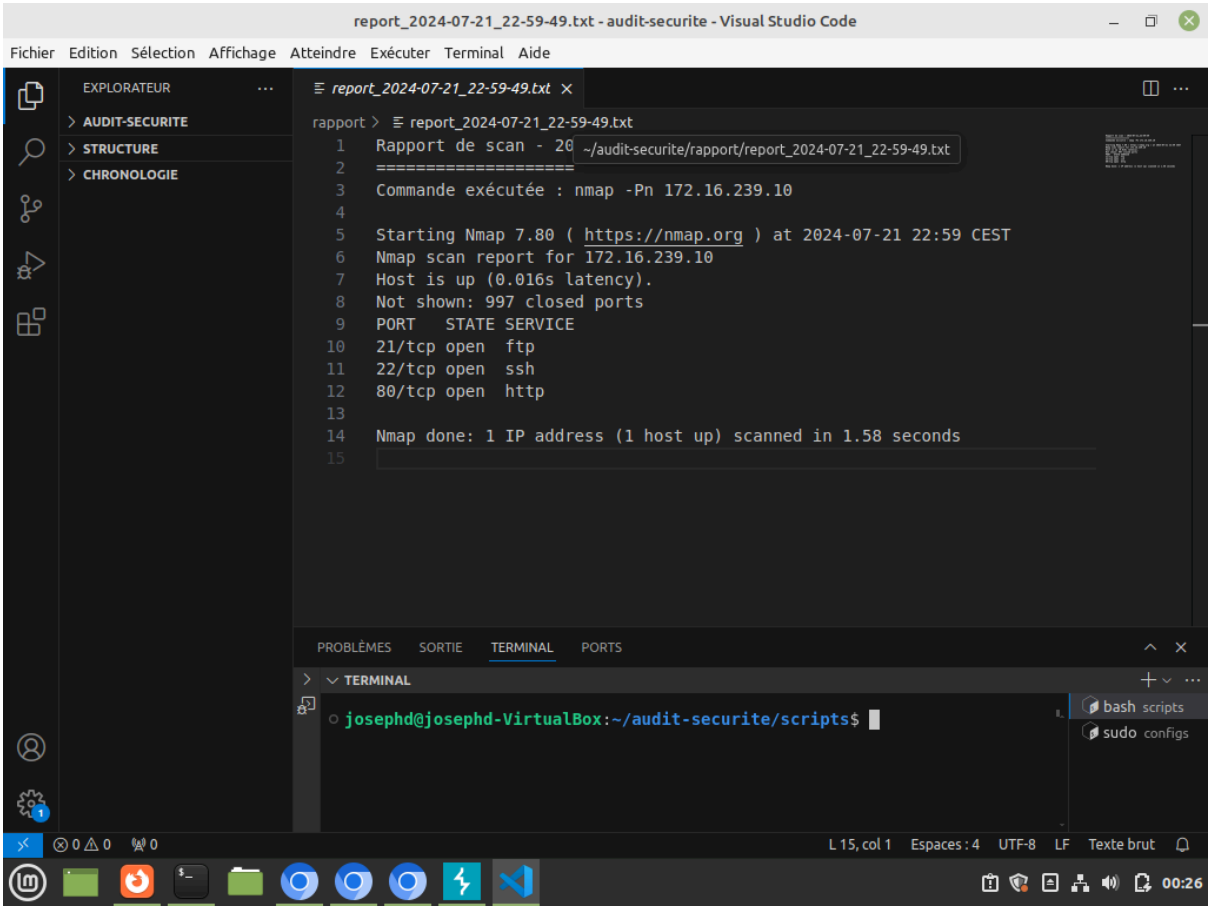
Vulnérabilité : Privilege Escalation

- **Description** : Une faille de sécurité a permis de changer le mot de passe de l'utilisateur `michel` via SSH sans autorisation appropriée.
- **Détails d'exploitation** : L'attaquant a pu changer le mot de passe de l'utilisateur `michel` en utilisant certaines commandes disponibles sur le système.
- **Évaluation de la criticité** : Critique
- **Scénario d'exploitation** : Un attaquant peut obtenir un accès complet au système en modifiant les privilèges des utilisateurs existants.
- **Recommandations** : Restreindre les permissions des utilisateurs, utiliser des outils de gestion des privilèges, et auditer régulièrement les permissions des fichiers et des utilisateurs.



Vulnérabilités autre:

Scan de ports avec Nmap : Pour obtenir une vue d'ensemble des services exposés par la cible, un scan de ports avec Nmap a été réalisé. Voici les résultats :



Recommandations pour chaque service découvert :

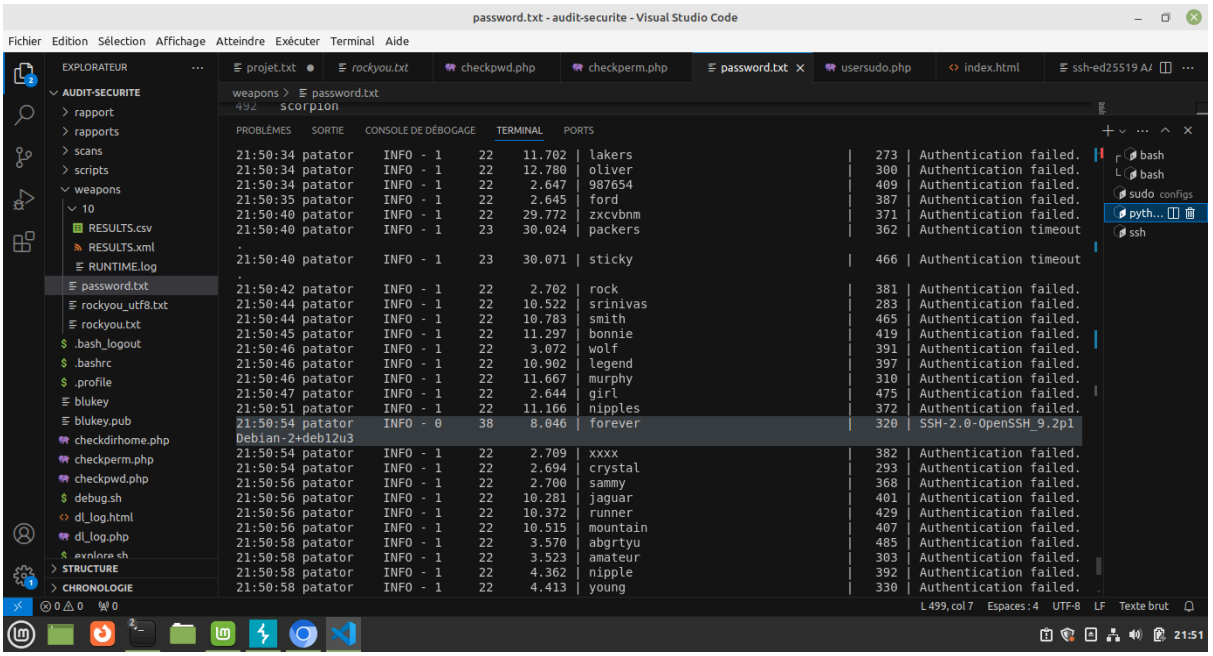
FTP (ProFTPD) :

- **Description** : Le port 21 est ouvert, exposant le service FTP.
- **Risques** : Les attaquants peuvent tenter de bruteforcer les identifiants FTP ou exploiter des vulnérabilités de ProFTPD.

- **Recommandations** :
  - **Utiliser FTPS ou SFTP** : Préférez des protocoles sécurisés pour le transfert de fichiers.
  - **Limiter les accès** : Restreindre l'accès FTP aux seules adresses IP de confiance.
  - **Mettre à jour ProFTPD** : Assurez-vous que le logiciel est à jour pour éviter les vulnérabilités connues.

SSH (OpenSSH) :

- **Description** : Le port 22 est ouvert, exposant le service SSH.
- **Risques** : Les attaquants peuvent tenter de bruteforcer les identifiants SSH.



- **Recommandations** :
  - **Utiliser des clés SSH** : Préférez l'authentification par clé plutôt que par mot de passe.
  - **Configurer Fail2Ban** : Utilisez des outils comme Fail2Ban pour bloquer les tentatives de bruteforce.
  - **Limiter les accès** : Restreindre l'accès SSH aux seules adresses IP de confiance.
  - **Désactiver les connexions root** : Empêchez les connexions SSH en tant que root.

HTTP (Apache) :

- **Description** : Le port 80 est ouvert, exposant le service HTTP.
- **Risques** : Les attaquants peuvent exploiter des vulnérabilités dans Apache ou les applications web.
- **Recommandations** :
  - **Forcer HTTPS** : Utiliser SSL/TLS pour sécuriser les communications et forcer les redirections HTTP vers HTTPS.
  - **Mettre à jour Apache** : Assurez-vous que le serveur Apache est à jour.
  - **Configurer les en-têtes de sécurité** : Implémentez des en-têtes de sécurité HTTP comme Content Security Policy (CSP), X-Frame-Options, etc.
  - **Restreindre l'accès aux fichiers sensibles** : Configurez Apache pour limiter l'accès aux fichiers sensibles.

Conclusion

Le test d'intrusion sur l'infrastructure de FramaFTP a révélé plusieurs vulnérabilités critiques et hautes qui exposent le système à des risques importants. Il est recommandé de corriger ces vulnérabilités rapidement en suivant les recommandations fournies pour renforcer la sécurité et protéger les données sensibles.