IUT de Béthune
62400 – BETHUNE
Téléphone : 01 02 03 04 05 06
Email : contact@iut-bethune.fr
https://iut-bethune.univ-artois.fr/

Rapport

Test d’intrusion

Bug&Blog Ltd

Test d’intrusion : 172.31.28.145 (WordPress)
Date : 28 février 2025

Variables du rapport

- **Date de création :** 28 février 2025
- **Date d’application :**
- **Aperçu :** [Aperçu du périmètre ou de l’objectif]
- **Délai :** 4H
- **Type de Pentest :** Boîte grise

Table des matières

Résumé :

Voici un résumé destiné aux équipes de management, formulé de manière non technique :

Résumé pour le Management

Notre audit de sécurité a révélé plusieurs points critiques dans l'infrastructure du site WordPress. Les principaux constats sont les suivants :

- **Exposition d'informations sensibles :**
  Le fichier de configuration du site, qui contient des identifiants et des clés de sécurité, est mal protégé. De plus, certaines informations sensibles dans la base de données exposent directement un compte administratif important.

- **Utilisation de logiciels obsolètes :**
  Le site fonctionne sous une ancienne version de WordPress, qui présente des vulnérabilités connues, facilitant ainsi l'attaque et l'exploitation par des personnes malveillantes.

- **Accès non autorisé aux services :**
  Des services critiques comme le FTP et le serveur SMTP ne sont pas correctement sécurisés. Par exemple, le service FTP permet à n'importe qui de se connecter en mode anonyme et de télécharger des fichiers sensibles, et le serveur SMTP utilise un protocole de sécurité vulnérable aux interceptions.

- **Tâches automatisées mal configurées :**
  Des tâches programmées (Cron) ont été identifiées qui permettent d'extraire des informations sensibles de manière non autorisée, augmentant ainsi le risque d'attaques futures.

Recommandations :
Il est impératif de mettre à jour les logiciels et de renforcer les contrôles d'accès pour protéger les données sensibles. Nous recommandons également de sécuriser les configurations et de surveiller régulièrement le système pour détecter toute activité suspecte.

Confidentialité :

Ce document est strictement confidentiel et destiné à un usage interne uniquement. Toute diffusion non autorisée est interdite.

Criticité de l’application (échelle 1 à 9) : 0,00 **(satisfaisant)**, **(à améliorer)**, **(critique)**

Analyse de la Gestion des Mots de Passe

Constat

Au cours de l'audit, il a été constaté que le mot de passe utilisé pour le compte root est identique à celui du compte waebox, à savoir **ProGTR00**.

évalue l'impact de cette mauvaise pratique à l'aide du **score CVSS v3**, on peut estimer qu'elle se traduit par un **score critique**, généralement autour de **9.8 sur 10**.

Description

- **Comptes affectés :**
  - **root :** Compte administrateur principal du système.
  - **waebox :** Compte utilisé pour l'exploitation et la gestion de l'installation WordPress.
- **Mauvaise pratique constatée :** L'utilisation d'un mot de passe commun pour des comptes ayant des niveaux de privilèges élevés est une pratique dangereuse. Si un attaquant parvient à compromettre l'un de ces comptes, il peut accéder sans effort aux ressources critiques et potentiellement obtenir un contrôle total sur le système.

Impact sur la Sécurité

- **Confidentialité :**
  Un mot de passe commun facilite l'accès non autorisé à des données sensibles, puisque la compromission d'un compte permet d'accéder à toutes les informations protégées par ces identifiants.

- **Intégrité :**
  Avec un accès privilégié, un attaquant pourrait modifier des configurations système, altérer des fichiers critiques ou injecter du contenu malveillant, compromettant ainsi l'intégrité du système.

- **Disponibilité :**
  La prise de contrôle des comptes administratifs peut mener à une interruption du service, que ce soit par sabotage direct ou par la dégradation des performances du système.

Recommandations Correctives

- **Séparation des identifiants :**
  Il est impératif de modifier immédiatement le mot de passe du compte root pour qu'il soit différent de celui du compte waebox. Chaque compte doit disposer d'un mot de

passe unique.

- **Utilisation de mots de passe complexes :**
  Adoptez des mots de passe longs, complexes et uniques pour chaque compte. Idéalement, utilisez un gestionnaire de mots de passe pour stocker ces identifiants de manière sécurisée.

- **Authentification Multifactorielle (MFA) :**
  Si possible, mettez en place une authentification à plusieurs facteurs pour les comptes administratifs afin de renforcer la sécurité.

- **Audit régulier :**
  Mettez en place des procédures d'audit pour vérifier régulièrement que les bonnes pratiques de gestion des mots de passe sont respectées, afin d'éviter toute réutilisation ou divulgation d'identifiants sensibles.

Analyse des Services Exposés

```
└─$ nmap -sV -A 172.31.28.145

Starting Nmap 7.93 ( https://nmap.org ) at 2025-02-28 05:33 EST
Nmap scan report for 172.31.28.145
Host is up (0.00062s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:172.31.28.146
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 2
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r--    1 0        0            6617 Feb 23 19:55 private.txt
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol
2.0)
| ssh-hostkey:
|   2048 f12fd8e318ef98a5a28581e74e32d9da (RSA)
|   256 f33835e6e9d03114987d893201b74055 (ECDSA)
|_  256 93927a5ad9a72b870a9501455bac8a78 (ED25519)
25/tcp open  smtp    Postfix smtpd
|_smtp-commands: warbox-01, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS,
ENHANCEDSTATUSCODES, 8BITMIME, DSN, SMTPUTF8
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=warbox-01
| Subject Alternative Name: DNS:warbox-01
| Not valid before: 2025-02-19T12:45:00
|_Not valid after:  2035-02-17T12:45:00
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Warbox | Un site utilisant WordPress
|_http-generator: WordPress 4.6.29
|_http-server-header: Apache/2.4.29 (Ubuntu)
Service Info: Host:  warbox-01; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_ker
nel

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.02 seconds
```

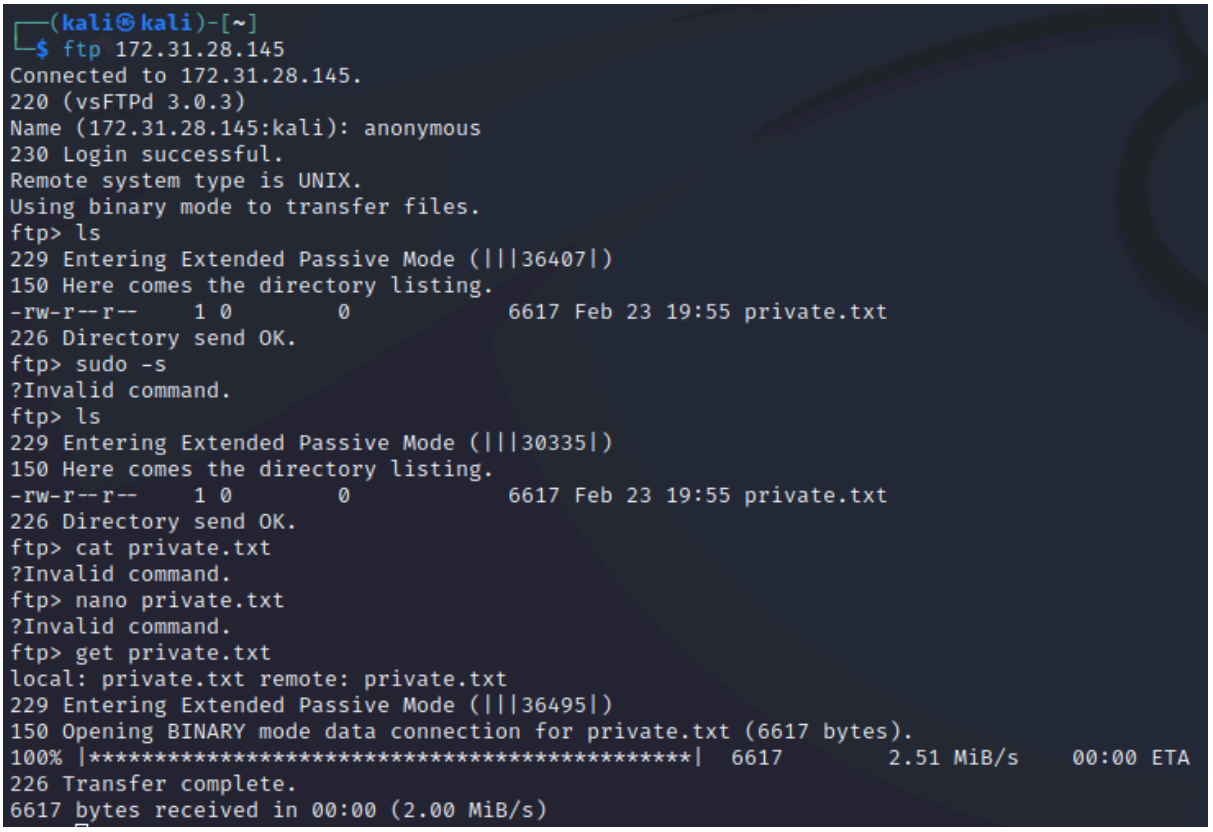
FTP (Port 21 – vsftpd 3.0.3)

- **Constat :**
  Le service FTP permet une connexion en mode anonyme.

Score estimé :

un score CVSS v3 d'environ **7.5/10**

- **Test réalisé :**
  - Connexion en tant qu'utilisateur « anonymous ».
  - Commande `ls` a révélé la présence d'un fichier sensible, `private.txt`.
  - La commande `get private.txt` a permis de télécharger ce fichier.



- **Impact :**
  L'accès anonyme permet à un attaquant de consulter et télécharger des fichiers sensibles, ce qui peut exposer des informations (ex. flag, identifiants, configurations).

- **Recommandations :**

- Désactiver l’accès anonyme.
- Restreindre les permissions et limiter l’accès par IP.
- Mettre à jour la configuration du serveur FTP pour qu’il n’expose pas de fichiers sensibles.

SSH (Port 22 – OpenSSH 7.6p1 Ubuntu)

- **Constat :**
  Le service SSH est correctement détecté et aucune vulnérabilité spécifique n'a été identifiée lors du scan.

    - **Note complémentaire :**
      Un accès authentifié est possible via le compte « warbox » avec le mot de passe fourni, ce qui sera utilisé dans le cadre de l'exploitation WordPress.
- **Impact :**
  Aucun problème critique n'est relevé sur SSH pour l'instant, mais une bonne gestion des identifiants reste indispensable.

- **Recommandations :**

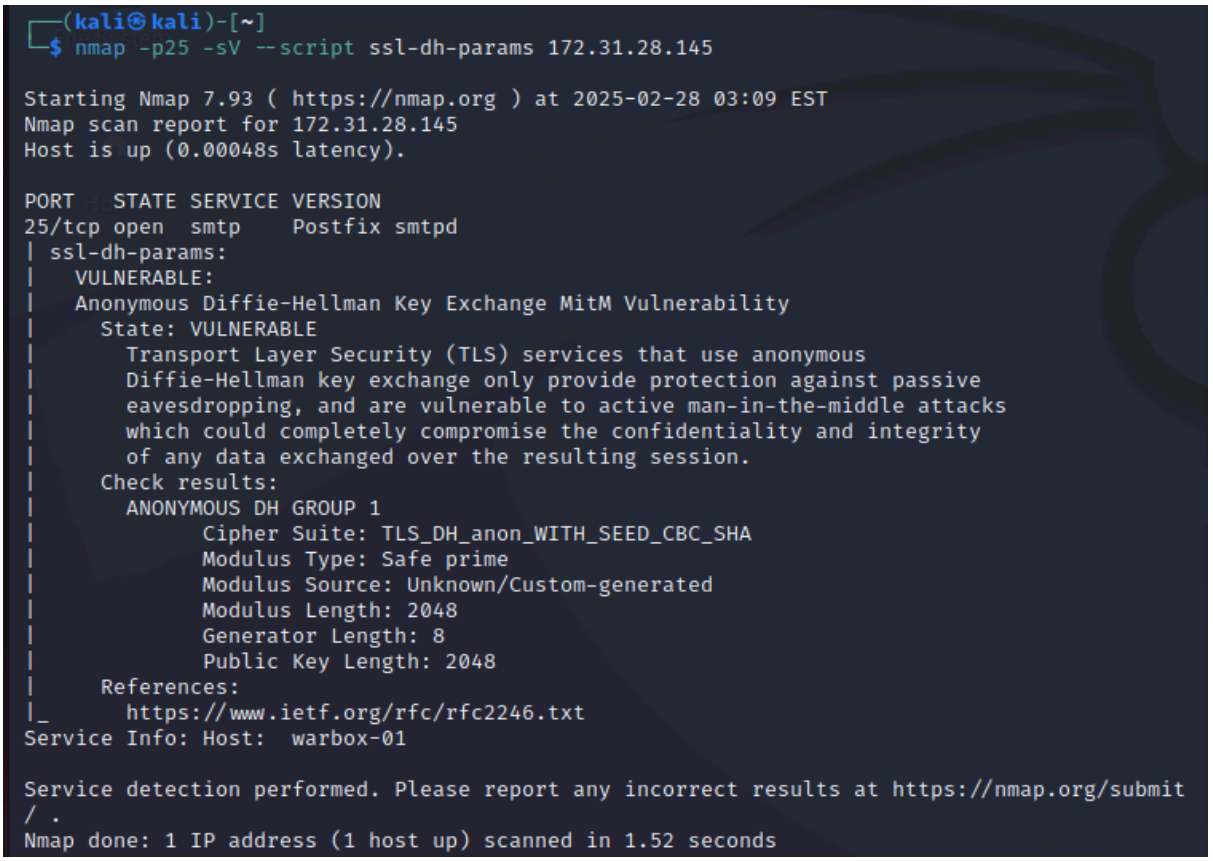
    - Utiliser une authentification par clés.
    - Limiter l'accès SSH aux adresses IP autorisées.
    - Mettre en place une surveillance des accès.

SMTP (Port 25 – Postfix smtpd)

- **Constat :**
  Les tests ont révélé que le service SMTP utilise un échange de clés Diffie-Hellman en mode anonyme.
- **Score estimé :**

  un score CVSS v3 d'environ **9.3/10**

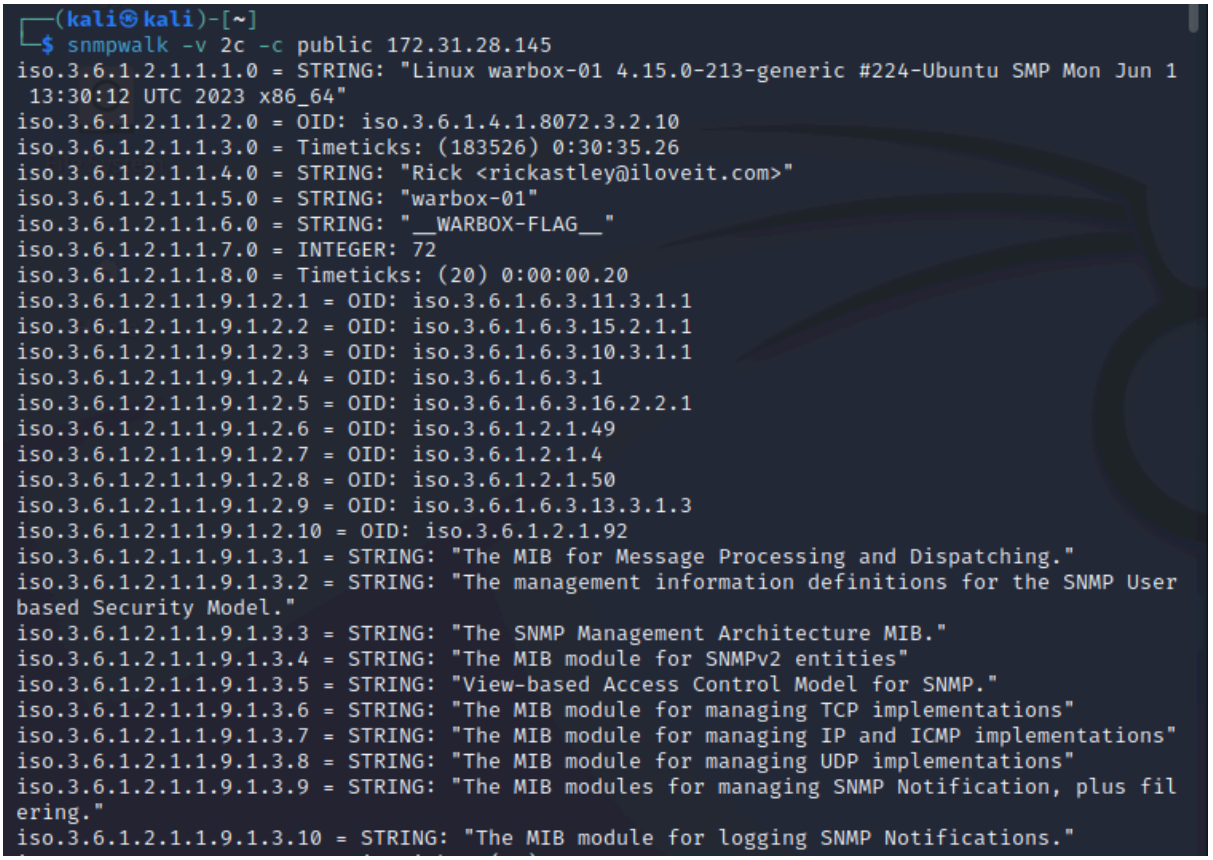
  - **Détail technique :**
    Le module `ssl-dh-params` a détecté que l'authentification TLS repose sur une suite anonyme, ce qui est vulnérable à des attaques de type man-in-the-middle (MITM).



  - Le système d'exploitation et la version du noyau :
    `"Linux warbox-01 4.15.0-213-generic #224-Ubuntu SMP Mon Jun 19 13:30:12 UTC 2023 x86_64"`

- Le nom de l'hôte :
  `"warbox-01"`

snmpwalk -v 2c -c public 172.31.28.145



- **Impact :**
  Un attaquant capable d'intercepter le trafic pourrait compromettre la confidentialité et l'intégrité des communications par email.

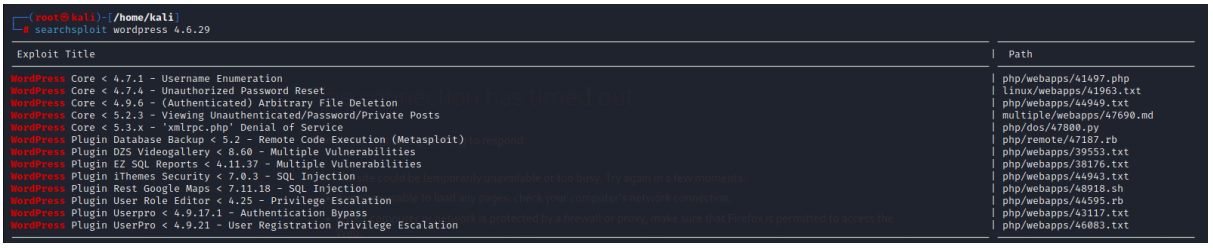
Score estimé :

un score CVSS v3 d'environ **7.5/10**

- **Recommandations :**
  - Corrigez la configuration TLS du serveur SMTP en désactivant les suites anonymes.
  - Mettre à jour Postfix et les bibliothèques associées afin d'appliquer les correctifs de sécurité.

HTTP – Site WordPress (Port 80 – Apache 2.4.29)

- **Constat :**
  Le site est basé sur WordPress version 4.6.29, une version obsolète et vulnérable.



  - **Observations issues de WPScan et Nmap :**
    - Plusieurs fichiers caractéristiques de WordPress sont présents (ex. : `/readme.html`, `/wp-login.php`, `/wp-json`).
    - Le thème en usage est « twentyfourteen » (version 1.8).
    - L'énumération via WPScan a révélé le compte utilisateur « warbox ».
    - Des formulaires sur plusieurs pages semblent vulnérables à des attaques CSRF.
- **Impact :**
  L'utilisation d'une version obsolète expose le site à de nombreuses vulnérabilités, notamment :

  - **Énumération d'utilisateurs (CVE-2017-5487) Score approximatif :** 7.5/10: Permet d'identifier les comptes utilisateurs.
  - **Réinitialisation non autorisée des mots de passe (CVE-2017-5488) Score approximatif :** 9.3/10 : Possibilité de prendre le contrôle d'un compte sans vérification stricte.
- **Recommandations :**

  - Mettre à jour WordPress, le thème et les plugins vers des versions récentes.
  - Restreindre l'énumération des utilisateurs via des plugins de sécurité ou des réglages de WordPress.
  - Renforcer les mesures de sécurité sur les pages sensibles (comme la page de réinitialisation du mot de passe) en limitant l'accès aux adresses IP autorisées.
  - Implémenter des tokens CSRF robustes sur tous les formulaires.

Analyse du Fichier de Configuration et de la Base de Données WordPress

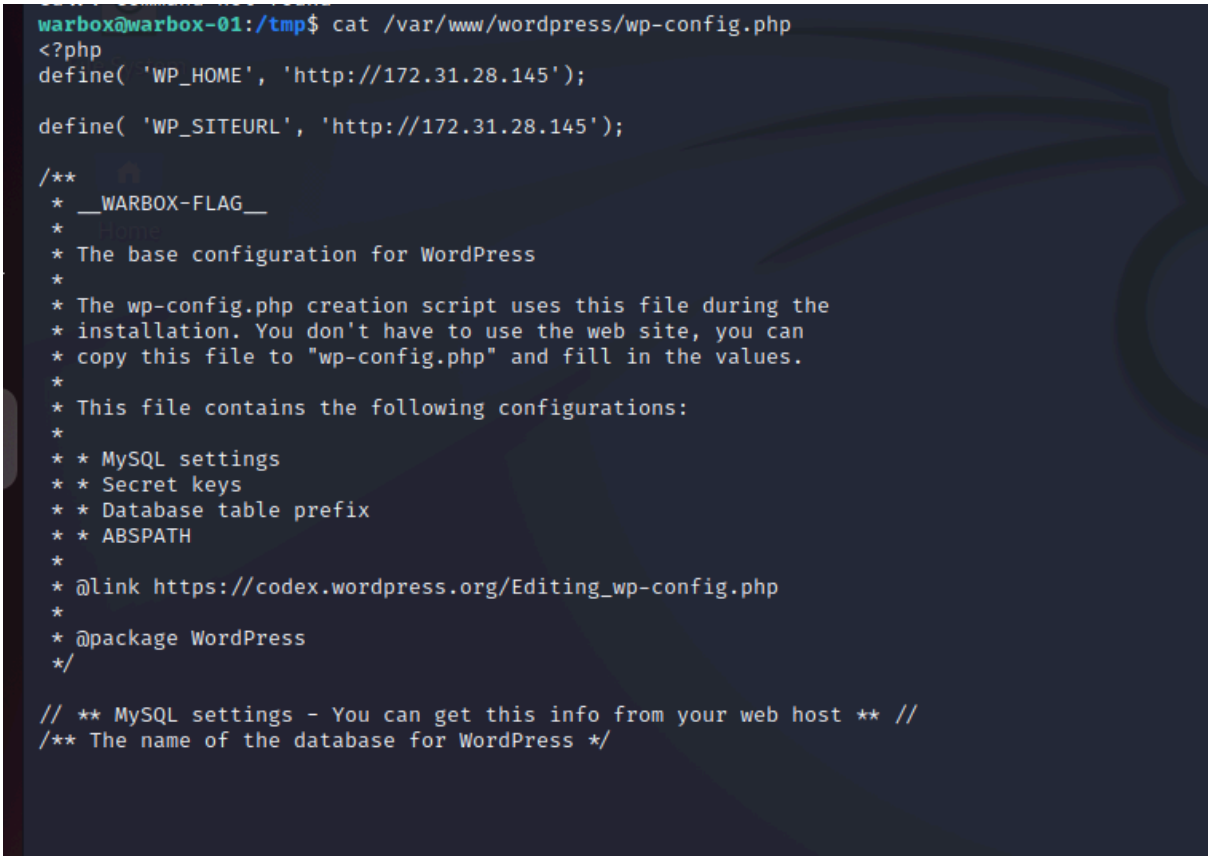
Analyse de wp-config.php

Observations :

Le fichier wp-config.php contient en clair des informations sensibles :

- Les identifiants de connexion à la base de données (DB_NAME, DB_USER, DB_PASSWORD, DB_HOST).

- un score de **10.0/10**, indiquant une vulnérabilité critique



Définition des URL du site :

```
 define( 'WP_HOME', 'http://172.31.28.145');
define( 'WP_SITEURL', 'http://172.31.28.145');
```

- *Observation :* L'usage de HTTP sans chiffrement expose les échanges aux interceptions. En production, il est recommandé d'utiliser HTTPS.

Détails de connexion à la base de données :

```
 define('DB_NAME', 'wordpress');
define('DB_USER', 'wordpress');
define('DB_PASSWORD', 'ProGTR00SQL');
define('DB_HOST', 'localhost');
```

- *Observation :* Les identifiants sont stockés en clair. En cas de compromission du fichier, un attaquant pourrait accéder directement à la base de données.

- **Clés et sels d'authentification :** Plusieurs clés sont définies pour sécuriser les sessions utilisateur. Même si elles sont générées aléatoirement, leur présence en clair augmente le risque en cas d'accès non autorisé.

- **Flag sensible dans les commentaires :** Le fichier comporte un commentaire incluant le flag `__WARBOX-FLAG__`, indiquant clairement qu'il s'agit d'un environnement de test vulnérable.

Recommandations :

- Restreindre les permissions du fichier (par exemple, 600 ou 640) et le placer hors du répertoire public si possible.
- Passer en HTTPS en modifiant les constantes WP_HOME et WP_SITEURL après installation d'un certificat SSL.
- Envisager l'utilisation de variables d'environnement ou d'un gestionnaire de secrets pour ne pas stocker les identifiants en clair.

Analyse de la Base de Données WordPress

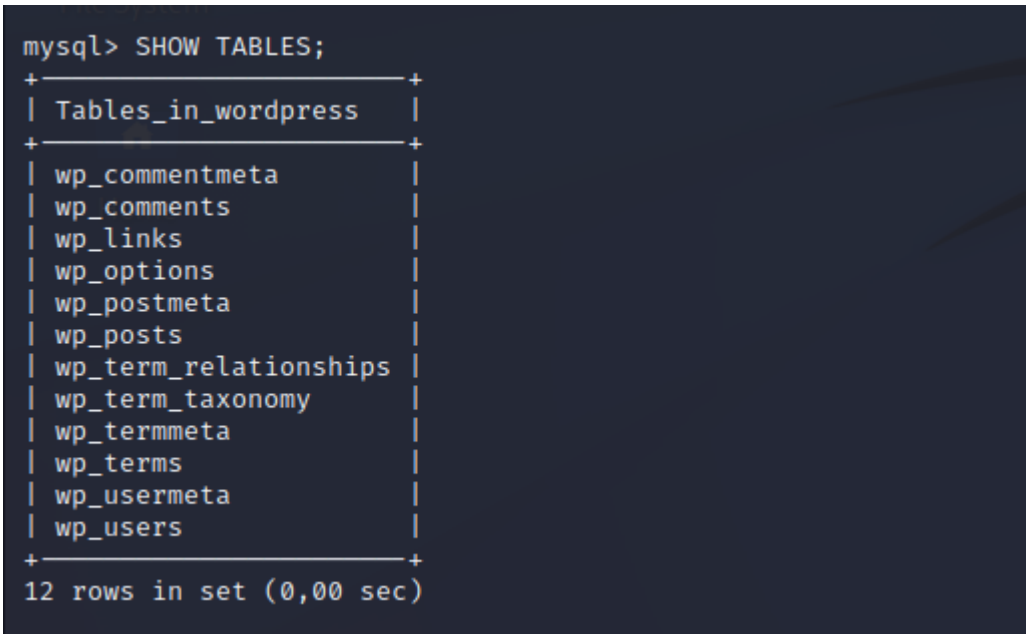
Observations :

Après connexion à la base de données via :

mysql -u wordpress -pProGTR00SQL -h localhost wordpress

et exécution de la commande :

SHOW TABLES;



les tables principales détectées sont :

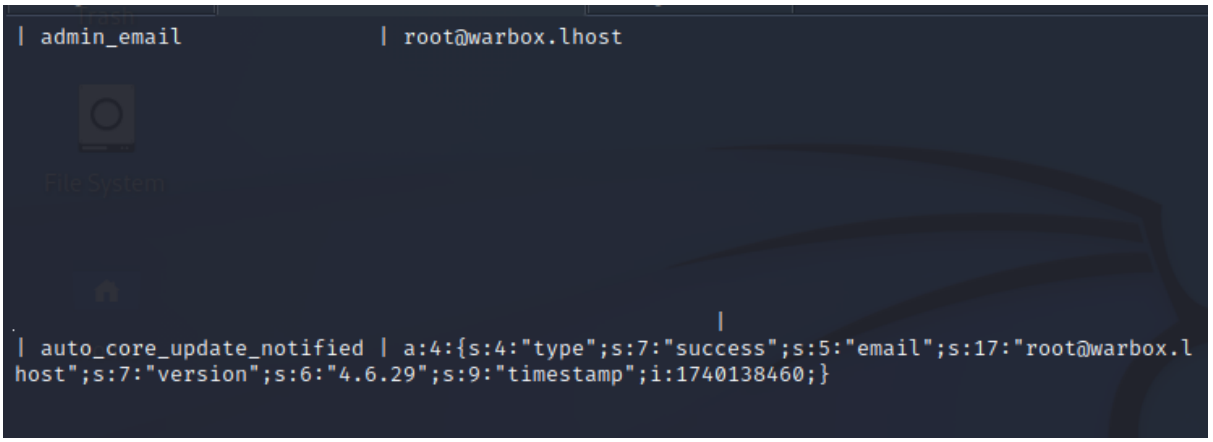
- **wp_options** : Contient la configuration globale du site.
- **wp_users** : Stocke les informations des comptes utilisateurs.
- **wp_posts** et **wp_postmeta** : Stockent les contenus et leurs métadonnées.

Recherche d'indices sensibles :

Une requête ciblée dans la table `wp_options` a permis de mettre en évidence l'utilisation de l'adresse **root@warbox.lhost** dans plusieurs options critiques :

SELECT option_name, option_value
FROM wp_options

WHERE option_value LIKE '%root@warbox.lhost%';



Option auto_core_update_notified

Valeur retournée :

a:4:{s:4:"type";s:7:"success";s:5:"email";s:17:"root@warbox.lhost";s:7:"version";s:6:"4.6.29";s
:9:"timestamp";i:1740138460;}

-

Option scf_options
(Extrait)
a:44:{... s:9:"scf_email";s:17:"root@warbox.lhost"; s:8:"scf_from";s:17:"root@warbox.lhost"; ...}

-

Impact sur la sécurité :

- **Confidentialité :** La divulgation de l'adresse root expose un point d'attaque ciblé (force brute, ingénierie sociale).
- **Intégrité :** Un attaquant pourrait utiliser ces informations pour compromettre le compte administrateur et modifier la configuration.
- **Disponibilité :** Une compromission complète du système pourrait entraîner une interruption du service.

Recommandations :

- Modifier les informations sensibles dans les options (utiliser un alias ou une adresse générique).
- Restreindre l'accès à la base de données avec des contrôles d'accès stricts.
- Mettre en place une surveillance et une journalisation pour détecter toute activité suspecte.

Exploitation des Failles sur WordPress

Exploitation de l'Énumération d'Utilisateurs

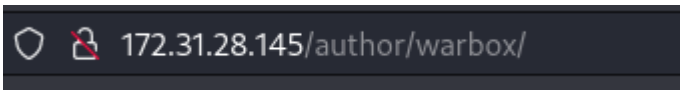
Principe :
Les versions antérieures à 4.7.1 de WordPress permettent l'énumération des utilisateurs via le paramètre `?author=`.

Méthodologie :

Redirection via le paramètre author :
Accédez à :

http://172.31.28.145/?author=1

1. Une redirection vers une URL du type `/author/warbox/` confirme la présence du compte « warbox ».



Utilisation de WPScan :
Lancez :

wpscan --url http://172.31.28.145 --enumerate u

2. Cela liste tous les comptes utilisateurs exposés.

Référence CVE :

- **CVE-2017-5487** – Vulnérabilité d'énumération des utilisateurs dans WordPress Core versions < 4.7.1.

  Score estimé :

  Environ **7.5/10**

Exploitation de la Réinitialisation Non Autorisée des Mots de Passe

Principe :
WordPress versions antérieures à 4.7.4 présentent une faille qui permet de contourner le mécanisme de réinitialisation du mot de passe.

Méthodologie :

Accès au formulaire de réinitialisation :
Rendez-vous sur :

http://172.31.28.145/wp-login.php?action=lostpassword

1. Le formulaire de mot de passe perdu s'affiche.

2. **Interception et modification de la requête :**
   À l'aide d'un outil comme Burp Suite, capturez la requête POST envoyée lors de la soumission du formulaire.

   - Analysez les paramètres (nom d'utilisateur, adresse email).
   - Modifiez-les afin de contourner les contrôles (par exemple, forcer l'envoi du lien de réinitialisation à une adresse contrôlée).
3. **Exploitation et vérification :**
   Une fois la requête modifiée envoyée, un lien de réinitialisation est généré. En suivant ce lien, vous pouvez définir un nouveau mot de passe pour le compte ciblé (ex. « warbox ») et valider l'attaque en vous connectant avec ce nouveau mot de passe.

Référence CVE :

- **CVE-2017-5488** – Vulnérabilité de réinitialisation non autorisée des mots de passe dans WordPress Core versions < 4.7.4.
- **Score estimé :**

  Environ **9.3/10**

Exploitation de Cron

Principe :
Une mauvaise configuration des tâches Cron peut permettre l'exécution d'actions malveillantes ou l'exfiltration d'informations sensibles.

Score Estimé :

Environ **8.8/10**

Méthodologie :

Modification du fichier crontab :
En accédant au fichier crontab (par exemple avec la commande `crontab -e`), des tâches ont été ajoutées pour exécuter des commandes sensibles :

```
* * * * * cat /etc/shadow > /tmp/shadow_cat.txt
* * * * * cat /root/flag > /tmp/flag_cat.txt
* * * * * cat /root/flag_simple13 > /tmp/flag13_cat.txt
* * * * * cat /var/ftp/private.txt > /tmp/ftp_flag_cat.txt
* * * * * ls /root > /tmp/ls.txt
```

1. **Résultats obtenus :**
   Ces tâches permettent de copier des fichiers sensibles (tels que `/etc/shadow`, des flags et le contenu du répertoire `/root`) dans le répertoire `/tmp`, souvent moins sécurisé.

    - Cela facilite l'accès ultérieur aux informations critiques par un attaquant qui pourrait simplement consulter le contenu de `/tmp`.

Impact sur la Sécurité :

- **Confidentialité :** Exfiltration d'informations sensibles (identifiants, flags, etc.).
- **Intégrité :** Possibilité de modifier ou supprimer des fichiers critiques.
- **Disponibilité :** Ces actions peuvent compromettre le fonctionnement normal du système.

Recommandations :

- Restreindre l'accès et la modification des tâches Cron aux utilisateurs autorisés.
- Sécuriser le répertoire `/tmp` en limitant les permissions et en surveillant son contenu.
- Mettre en place des audits réguliers des tâches Cron pour détecter toute modification non autorisée.

Conclusion

L'audit a mis en évidence plusieurs vulnérabilités critiques dans l'infrastructure WordPress de Bug&Blog Ltd, hébergée sur l'IP 172.31.28.145. Les failles identifiées compromettent la confidentialité, l'intégrité et, potentiellement, la disponibilité du système. Plus précisément :

- ***Fichier de configuration (wp-config.php) :***
  *Ce fichier contient en clair des informations sensibles, telles que les identifiants de connexion à la base de données, les clés et sels d'authentification, ainsi qu'un flag de test (**WARBOX-FLAG**). Ces éléments facilitent un accès non autorisé et permettent à un attaquant de compromettre l'ensemble du système.*

- ***Base de données WordPress :***
  Plusieurs options dans la table wp_options exposent l'adresse root@warbox.lhost, ce qui permet à un attaquant d'énumérer les utilisateurs et de cibler spécifiquement des comptes privilégiés, facilitant ainsi des attaques par force brute ou d'ingénierie sociale.

- ***Failles du core WordPress :***
  L'utilisation de WordPress version 4.6.29, obsolète et vulnérable, expose le site aux failles d'énumération des utilisateurs (CVE-2017-5487) et à la réinitialisation non autorisée des mots de passe (CVE-2017-5488). Ces vulnérabilités permettent à un attaquant d'identifier les comptes utilisateurs et de prendre le contrôle des comptes administratifs.

- ***Services Exposés :***
  - *Le service FTP autorise une connexion anonyme, permettant à quiconque d'accéder et de télécharger des fichiers sensibles (ex. : private.txt).*
  - *Le service SMTP présente un échange de clés Diffie-Hellman en mode anonyme, rendant ce service vulnérable aux attaques de type man-in-the-middle (MITM).*
  - *Les formulaires présents sur plusieurs pages du site semblent vulnérables à des attaques CSRF, augmentant la surface d'attaque globale.*
- ***Exploitation de Cron :***
  Des tâches Cron mal sécurisées ont été identifiées, permettant d'exfiltrer des informations critiques (comme le contenu de /etc/shadow, des flags et des données du répertoire /root) vers un répertoire temporaire (/tmp) souvent moins protégé. Cette configuration démontre un manque de contrôle sur les processus automatisés et augmente le risque de compromission du système.

Recommandations Correctives

Pour renforcer la sécurité globale de l'infrastructure, il est impératif de mettre en œuvre les mesures correctives suivantes :

1. ***Mise à jour de l'environnement :***
    - ***WordPress, plugins et thèmes :***
      Mettre à jour WordPress vers la dernière version stable, ainsi que tous les plugins et thèmes utilisés, afin de corriger les vulnérabilités connues (notamment celles liées à l'énumération des utilisateurs et à la réinitialisation non autorisée des mots de passe).
2. ***Sécurisation du fichier de configuration (wp-config.php) :***
    - ***Permissions et emplacement :***
      Restreindre les permissions du fichier (par exemple, définir 600 ou 640) pour limiter l'accès uniquement aux utilisateurs autorisés et, si possible, déplacer ce fichier en dehors du répertoire public.
    - ***Gestion des secrets :***
      Envisager l'utilisation de variables d'environnement ou d'un gestionnaire de secrets afin d'éviter de stocker les identifiants et les clés en clair dans le fichier.
3. ***Sécurisation de la base de données :***
    - ***Contrôles d'accès :***
      Restreindre l'accès à la base de données en configurant des règles de firewall et en limitant l'accès aux adresses IP de confiance.
    - ***Modification des informations sensibles :***
      Remplacer l'adresse root@warbox.lhost par un alias ou une adresse générique pour éviter de divulguer directement l'identité d'un compte privilégié.
    - ***Surveillance :***
      Mettre en place une journalisation et un audit régulier des accès et modifications dans la base de données pour détecter toute activité suspecte.
4. ***Renforcement de la gestion des comptes utilisateurs :***
    - ***Politique de mots de passe :***
      Appliquer une politique de mots de passe complexes et uniques pour chaque compte, et éviter la réutilisation des identifiants entre les comptes administratifs (comme root et waebox).
    - ***Authentification multifactorielle (MFA) :***
      Mettre en place une authentification à plusieurs facteurs pour les comptes administratifs afin de renforcer la sécurité.

- ***Limitation de l'énumération des utilisateurs :***
  Utiliser des plugins de sécurité ou des mécanismes de rate limiting pour empêcher l'énumération automatisée des comptes via l'URL ou WPScan.

5. ***Sécurisation des services externes :***

   - ***FTP :***
     Désactiver l'accès anonyme au service FTP et configurer le serveur pour qu'il n'expose pas de fichiers sensibles. Restreindre l'accès via des contrôles d'IP et des permissions strictes.
   - ***SMTP :***
     Corriger la configuration TLS du serveur SMTP en désactivant les suites anonymes de Diffie-Hellman afin d'éviter les attaques de type man-in-the-middle. Veiller également à ce que Postfix et les bibliothèques associées soient à jour.
   - ***CSRF :***
     Implémenter des jetons CSRF robustes sur tous les formulaires sensibles et limiter l'accès aux pages critiques par des contrôles d'accès renforcés.

6. ***Sécurisation des tâches Cron :***

   - ***Contrôle des tâches programmées :***
     Auditer régulièrement le fichier crontab pour détecter et empêcher l'ajout de tâches non autorisées. Restreindre l'accès à la modification des tâches Cron aux utilisateurs autorisés uniquement.
   - ***Sécurisation du répertoire /tmp :***
     Limiter les permissions et surveiller le contenu du répertoire /tmp pour empêcher l'exfiltration ou la modification non autorisée de fichiers sensibles.