Mise en place et évaluation de ClamAV

Test de détection de webshells sur un serveur Linux potentiellement compromis

Serveur de test	catelsys - Debian 11 - noyau Linux 5.10
Ressources observées	Environ 2 Go de RAM et 1 Go de swap
Périmètre	Fichiers PHP suspects déplacés dans /root/quarantine_catelsys/
Date de test	22 juin 2026

Conclusion opérationnelle

ClamAV a été installé et les bases de signatures sont valides et récentes. Le moteur a toutefois consommé environ 1 Go de RAM sur une machine de 2 Go et n'a détecté aucun des webshells PHP mis en quarantaine. Il reste utile comme couche antivirus complémentaire, mais ne répond pas seul au besoin de détection fiable de webshells sur un serveur web compromis.

1. Objectif du test

L'objectif est de vérifier si ClamAV peut identifier automatiquement des fichiers PHP malveillants récupérés sur un serveur Linux exposé. Dans le laboratoire, les fichiers suspects avaient déjà été retirés du répertoire web et placés en quarantaine afin de ne plus être exécutables. Le test consiste à contrôler si ClamAV les reconnaît comme malveillants et, à terme, à déterminer si l'outil peut renforcer la protection des serveurs exposés.

2. Fonctionnement de ClamAV

ClamAV est un antivirus open source fondé principalement sur la comparaison de fichiers avec une base de signatures. Une signature peut être une empreinte, une chaîne de caractères ou un motif associé à un malware connu. Il ne déduit donc pas systématiquement qu'un fichier est malveillant : il le détecte surtout lorsqu'une signature correspond.

Les composants utilisés sont les suivants :

- clamav-daemon (clamd) : service qui charge les signatures en mémoire et répond aux demandes de scan.
- freshclam : service chargé de récupérer les mises à jour de signatures.
- clamscan : scanner autonome lancé ponctuellement depuis la ligne de commande.
- clamdscan : client qui demande au démon clamd de réaliser le scan.

3. Installation et mise à jour initiale

Installation des paquets nécessaires :

```
apt-get update
apt-get install -y clamav clamav-daemon
```

Après installation, une mise à jour initiale est nécessaire afin de récupérer les premières bases de signatures. Le service freshclam est arrêté avant une mise à jour manuelle pour éviter un conflit sur le fichier de log ou les fichiers de base.

```
systemctl stop clamav-freshclam
freshclam
systemctl start clamav-freshclam
```

4. Cas particulier : Debian 10/11 et version ClamAV obsolète

Sur les serveurs anciens, notamment sous Debian 10 ou Debian 11, la version de ClamAV fournie par les dépôts peut être obsolète. Dans notre cas, la version 0.103.10 était signalée comme ancienne par les serveurs de mise à jour, ce qui a provoqué un refus de téléchargement des signatures (erreur 403 / blocage CDN).

Le fait de relancer freshclam plusieurs fois de suite peut entraîner un cooldown. Il faut alors attendre la fin du délai plutôt que multiplier les tentatives.

Comme contournement temporaire, les bases peuvent être mises à jour sur une machine Linux plus récente, puis transférées vers le serveur ancien par SSH :

```
scp /var/lib/clamav/*.cvd root@IP_DU_SERVEUR_CIBLE:/var/lib/clamav/
chown clamav:clamav /var/lib/clamav/*.cvd
chmod 644 /var/lib/clamav/*.cvd
```

Cette solution permet de conserver une base de signatures récente sans solliciter directement le CDN depuis le serveur ancien. Elle doit néanmoins rester temporaire : la mise à niveau du système et de ClamAV reste la solution durable.

5. Vérification des signatures transférées

Les trois bases présentes sur le serveur ont été contrôlées avec sigtool. Le résultat « Verification OK » confirme que les fichiers sont lisibles et signés correctement. La base n'est donc pas vide.

Base	Version	Signatures	État
daily.cvd	28039	355 472	Valide - mise à jour du 22/06/2026
main.cvd	63	3 287 027	Valide
bytecode.cvd	339	80	Valide

Total chargé : environ 3,64 millions de signatures. La présence des fichiers daily.cvd, main.cvd et bytecode.cvd, ainsi que la validation cryptographique effectuée par sigtool, confirment que ClamAV dispose bien d'une base exploitable.

6. Premier essai avec clamscan et fichier EICAR

Un fichier EICAR a été créé pour valider le comportement général du moteur antivirus. EICAR est un fichier de test inoffensif dont le contenu est conçu pour être détecté comme une signature antivirus standard. Il permet de tester un antivirus sans introduire de malware réel.

```
echo 'X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*' > eicar.txt
clamscan --infected --remove --recursive eicar.txt
```

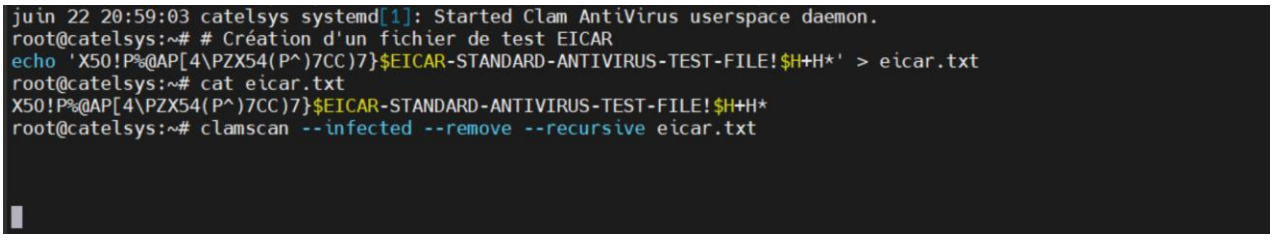

Figure 1 - Commande clamscan lancée sur le fichier EICAR. Le scan ne termine pas rapidement sur la machine de test.

Le scan via clamscan s'est avéré très lent. L'observation du processus a montré une consommation mémoire d'environ 770 Mo pour clamscan, seulement 35 Mo de mémoire disponibles, ainsi qu'un taux important d'attente disque (I/O wait). Le processus s'est retrouvé en état D, c'est-à-dire bloqué en attente d'une opération disque. Sur cette machine, clamscan recharge la totalité des signatures à chaque exécution, ce qui est trop lourd pour un scan ponctuel.

7. Choix de clamdscan pour la suite du test

Le démon clamd a été démarré avec succès. Une fois les signatures chargées, il consommait environ 1 Go de RAM. Cette charge reste importante pour un serveur doté d'environ 2 Go de mémoire, mais elle évite de recharger les bases à chaque nouveau scan.

Outil	Fonctionnement	Impact observé	Usage recommandé
clamscan	Scanner autonome ; charge les bases à chaque lancement.	Très lent sur 2 Go de RAM ; pression mémoire et attente disque.	Tests simples sur une machine disposant de ressources suffisantes.
clamdscan	Envoie la demande au démon clamd qui garde les signatures en mémoire.	Scan de fichiers individuel rapide après chargement du démon.	Serveur et scans récurrents, à condition d'accepter la consommation mémoire de clamd.

La commande utilisée pour les fichiers en quarantaine est :

```
clamdscan --fdpass /root/quarantine_catelsys/NOM_DU_FICHIER.php
```

L'option --fdpass permet de transmettre au démon les droits d'accès au fichier à scanner, ce qui est utile lorsque le client est lancé avec root alors que le démon fonctionne avec l'utilisateur clamav.

8. Résultat du test sur les webshells mis en quarantaine

Trois fichiers PHP identifiés comme suspects lors du traitement manuel du serveur ont été analysés individuellement. Ils sont restés placés dans le répertoire /root/quarantine_catelsys/.

Fichier analysé	Commande	Résultat ClamAV
cache1b4c5aindex.php	clamdscan --fdpass	OK - aucune détection
lindex.php	clamdscan --fdpass	OK - aucune détection
log55e8d1index.php	clamdscan --fdpass	OK - aucune détection

Résultat fonctionnel

Les bases de signatures étaient bien présentes et validées, mais les fichiers PHP testés ont tous été déclarés « OK ». Cela ne prouve pas qu'ils sont sains : cela signifie uniquement qu'ils ne correspondent à aucune signature ClamAV connue par le moteur installé.

9. Conclusion et recommandations

Le déploiement de ClamAV a été techniquement réussi : le démon démarre, les bases sont présentes, validées et chargées. L'outil apporte donc une protection supplémentaire contre des malwares connus et peut rester utile pour analyser des fichiers uploadés, effectuer des contrôles ponctuels ou compléter les autres mesures de sécurité.

Cependant, le test ne répond pas pleinement à l'objectif principal : ClamAV n'a détecté aucun des webshells PHP placés en quarantaine. En outre, le démon consomme une part très importante de la mémoire disponible sur ce serveur. Dans cet environnement, ClamAV ne doit pas être considéré comme une solution unique de détection de compromission web.

Recommandations :

- Conserver ClamAV comme couche complémentaire, notamment pour les fichiers connus et les pièces jointes ou uploads.

- Ne pas utiliser --remove automatiquement sur un site en production avant analyse et sauvegarde : une détection peut concerner un faux positif ou un fichier critique.
- Compléter la détection par une recherche de motifs PHP suspects : eval, base64_decode, gzinflate, shell_exec, system, passthru, assert, file_put_contents ou move_uploaded_file.
- Mettre en place un contrôle d'intégrité des fichiers (FIM) ou une surveillance des ajouts/modifications dans les répertoires web.
- Prévoir une mise à niveau de Debian et de ClamAV afin de limiter les problèmes de mises à jour via freshclam et le CDN.

Décision proposée : conserver ClamAV comme outil de contrôle complémentaire, mais rechercher une solution de détection / intégrité plus adaptée aux webshells et aux serveurs web exposés.