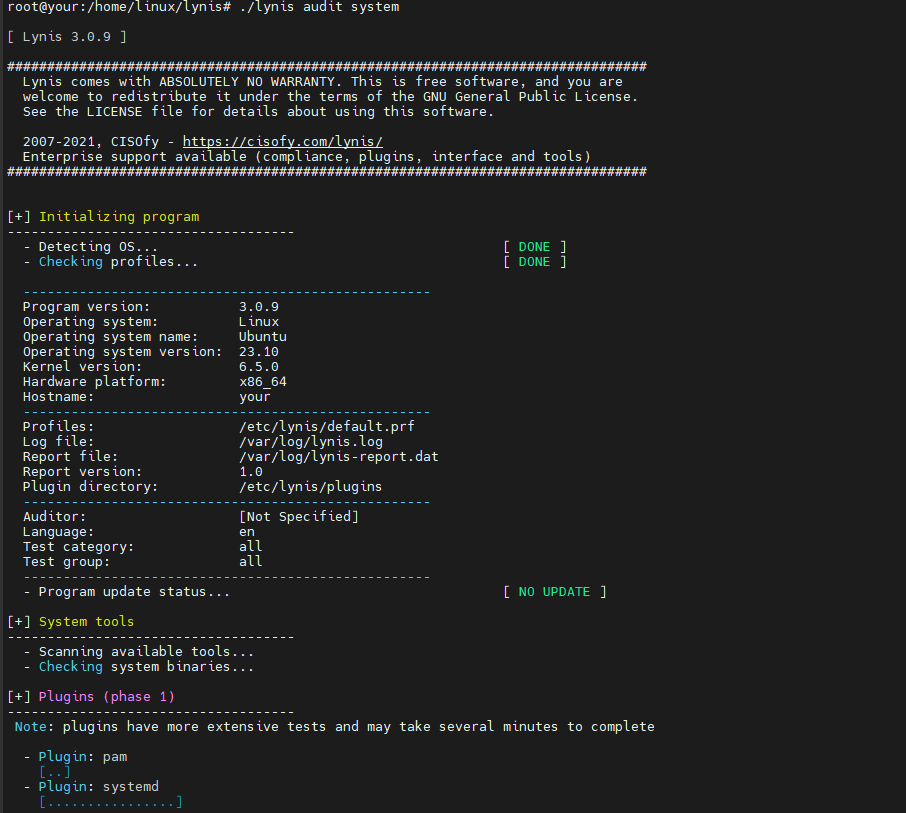
SECURITE DES SI - Projet

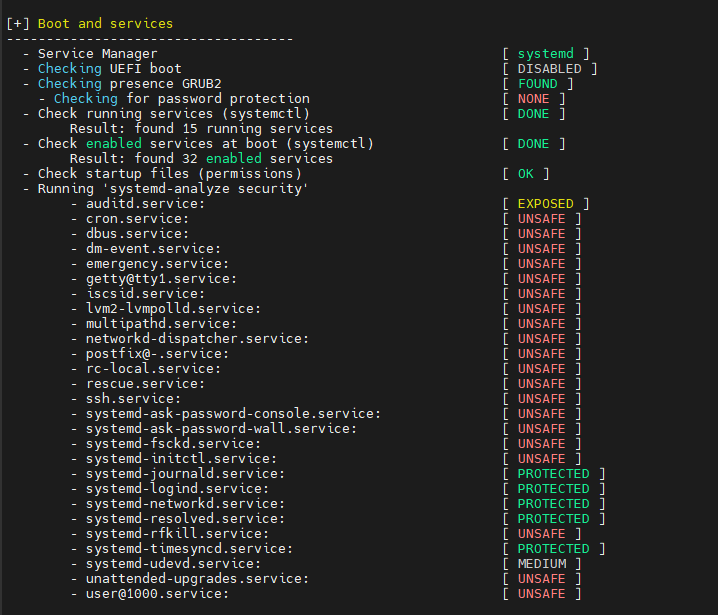
**1.Préparer une VM Linux et faites un travail de durcissement en inscrivant dans le rapport le détail de votre travail.**

J'ai commencé par choisir Ubuntu Server en version légère comme système d'exploitation pour ma machine virtuelle, visant à sécuriser et à optimiser les performances. Pour faciliter la gestion des fichiers de configuration, j'ai installé Nano, un éditeur de texte en ligne de commande simple et efficace. Conscient de l'importance de la sécurité, j'ai décidé d'ajouter Lynis à mon arsenal, un outil d'audit de sécurité reconnu pour son efficacité dans l'identification des vulnérabilités et des conseils pour renforcer la sécurité de mon système. J'ai également installé Git, un outil essentiel pour le versionnage de code, qui me permet de télécharger et de gérer facilement des scripts de sécurité ou des applications depuis des dépôts en ligne. Pour m'assurer que ma VM reste sécurisée et à jour, j'ai exécuté les commandes sudo apt update et sudo apt upgrade, actualisant ainsi les packages et appliquant les derniers correctifs de sécurité disponibles. Mon serveur Linux est maintenant en fonctionnement, équipé des outils nécessaires pour un durcissement efficace et prêt pour des configurations de sécurité plus spécifiques selon mes besoins.

J’ai donc effectuer le premiers scan Lynis en vue du durcissement:

****

**Premiere partie rien à signaler, je passe sur les boot and services :**

****

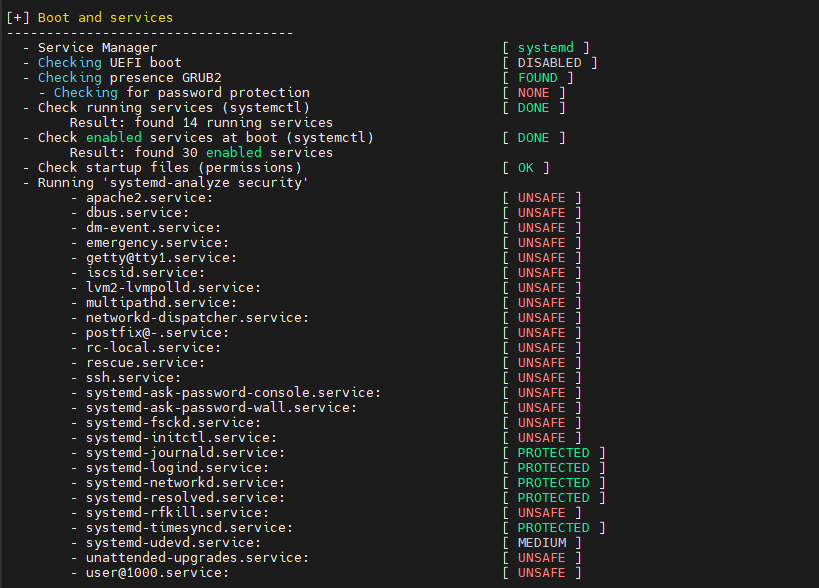
**Les services suivant sont considérer comme sécurisé :**

**- polkit.service: - systemd-journald.service: - systemd-logind.service: systemd-networkd.service: systemd-resolved.service: - systemd-udevd.service:**

**Les services dans le tableau ci dessous devront etre sécurisé**

| **SERVICE** | **Utilité** | **CORRECTIF** |
| --- | --- | --- |
| **cloud-init-hotplugd.service** | Un ensemble de scripts qui s'exécutent lors du démarrage d'une machine virtuelle dans un environnement cloud pour initialiser le système |  |
| dbus.service | Intercommunication systèmes. Permet aux applications de communiquer entre elles. | limiter les accès autorisés et surveillez les activités suspectes. |
| dm-event.service: | Surveillance LVM. Écoute les événements des volumes logiques pour gérer les changements dynamiquement. | Sécurisez-le en restreignant l'accès physique au serveur et en chiffrant les données stockées. |
| emergency.service: | Mode urgence. Fournit un shell minimal pour la récupération système. | Pour le sécuriser, utilisez des mots de passe forts pour les comptes administratifs et désactivez le démarrage automatique. |
| getty@tty1.service: | Connexion console. Gère les sessions de connexion sur les terminaux. | Sécurisez-le en limitant les accès au terminal physique et en utilisant l'authentification à deux facteurs. |
| iscsid.service: | Ce service est le démon du Initiator iSCSI, qui permet à votre serveur de se connecter à des cibles iSCSI pour l'utilisation de stockage réseau. iSCSI est largement utilisé dans les environnements d'entreprise pour connecter des serveurs à des baies de stockage réseau via le protocole IP standard.  Sécurisation/Désactivation : Si vous n'utilisez pas de stockage iSCSI, désactiver ce service réduit la surface d'attaque potentieller |  |
| lvm2-lvmpolld.service: | Ce service est utilisé par LVM (Logical Volume Manager) pour surveiller et gérer les actions asynchrones sur les volumes logiques, comme les migrations de données ou les redimensionnements de volume. |  |
| multipathd.service: | Gestion multipath. Gère les chemins multiples pour les périphériques de stockage pour améliorer la redondance et la performance. | Doit s’assurer que seuls les chemins valides et sécurisés sont utilisés, et surveillez régulièrement les configurations. |
| networkd-dispatcher.service: | networkd-dispatcher permet d'exécuter des scripts basés sur des événements réseau pour des configurations dynamiques, en travaillant avec systemd-networkd. |  |
| - packagekit.service: | PackageKit est une couche d'abstraction qui permet aux utilisateurs d'interagir avec les logiciels et les systèmes de gestion de paquets de manière unifiée, sans se soucier des détails spécifiques à chaque système de paquets | Inactif de base  sudo systemctl stop  packagekit.service  sudo systemctl disable packagekit.service  sudo apt purge packagekit |
| plymouth-start.service: | Plymouth fournit une interface graphique pendant le processus de démarrage du système, affichant un écran de démarrage animé. Il gère également le dialogue de cryptage des disques (par exemple, la saisie de mots de passe pour les disques chiffrés).  Importance sur Ubuntu Server : Sur la plupart des serveurs, surtout ceux sans interface graphique ou ceux qui ne nécessitent pas d'interaction utilisateur pendant le démarrage, Plymouth n'est pas nécessaire. Son utilité est principalement esthétique ou concerne l'interaction utilisateur lors du sudo systemctl disable plymouth-start.service | sudo systemctl disable plymouth-start.service  dpkg -l | grep plymouth  sudo apt-get purge $(dpkg -l | grep plymouth | awk '{print $2}')  sudo apt-get autoremove  apt-cache rdepends plymouth  sudo update-grub (pour vérifier si le démarrage est ok) |
| rc-local.service: | Scripts personnalisés. Exécute des scripts personnalisés au démarrage du système. |  |
| rescue.service: | Mode secours. Fournit un environnement minimal pour la réparation du système. | Protéger l'accès au mode de secours avec des mots de passe solides |
| snapd.aa-prompt-listener.service: | Très connu comme manageur de Packets | Faire la commande “snap list”  No snaps are installed yet. Pour vérifier les dependances. Ici aucun, donc je fait la commande  sudo apt-get remove --purge snapd |
| snapd.service: | DONE | Supression du manager de packet Snap car inutile |
| ssh.service: | Accès sécurisé en ssh | Utilisez des clés SSH plutôt que des mots de passe, désactivez l'accès root et limitez les adresses IP autorisées. |
| systemd-ask-password-console.service | Gestion des mots de passe | S'assurez-vous que seuls les utilisateurs autorisés peuvent répondre à ces invites et utiliser l'utilisation de l'authentification à deux facteurs. |
| systemd-ask-password-plymouth.service | Gestion des mots de passe | S'assurez-vous que seuls les utilisateurs autorisés peuvent répondre à ces invites et utiliser l'utilisation de l'authentification à deux facteurs. |
| systemd-ask-password-wall.service | Gestion des mots de passe | S'assurez-vous que seuls les utilisateurs autorisés peuvent répondre à ces invites et utiliser l'utilisation de l'authentification à deux facteurs. |
| systemd-fsckd.service | est utilisé pour fournir un démon de retour d'information pour fsck, l'utilitaire de vérification du système de fichiers. Lors du démarrage, si des vérifications du système de fichiers sont nécessaires, systemd-fsckd affiche des progrès et des messages pour informer l'utilisateur.  Comme pour systemd-initctl.service, désactiver systemd-fsckd.service n'est généralement pas recommandé car il joue un rôle important dans le processus de démarrage, surtout si des vérifications du système de fichiers sont nécessaires après un arrêt incorrect ou si le système de fichiers est marqué comme "sale". |  |
| systemd-initctl.service | Le service systemd-initctl.service est un service systemd qui fournit une compatibilité avec les systèmes SysVinit en redirigeant les appels à /dev/initctl vers systemd. Ce service permet à systemd de traiter les requêtes de contrôle du système qui étaient traditionnellement gérées par SysVinit.  Sécuriser systemd-initctl.service ou tout autre service systemd consiste généralement à s'assurer que le service est correctement configuré, ne s'exécute pas avec des privilèges inutilement élevés, et est exposé le moins possible. Voici quelques recommandations générales pour sécuriser les services systemd : | sudo systemctl disable systemd-initctl.service |
| systemd-rfkill.service | Permet aux utilisateurs de conserver leurs préférences en matière d'activation/désactivation du Wi-Fi et du Bluetooth entre les redémarrages. |  |
| thermald.service | Regule la température. Dans le cas d’une VM, le principe de minimisation s’applique | sudo systemctl stop thermald  sudo systemctl disable thermald  sudo apt-get remove --purge thermald |
| unattended-upgrades.service | Mises à jour automatiques. Installe automatiquement les mises à jour de sécurité. | A configurer pour appliquer uniquement les mises à jour de sécurité fiables et surveiller les journaux pour toute installation échouée. |
| user@1000.service | Gestion de session utilisateur. Gère les processus et services lancés par l'utilisateur avec l'ID 1000. |  |
| Apport | Un système qui collecte automatiquement des données sur les plantages des logiciels et les erreurs sur les systèmes basés sur Ubuntu et peut les envoyer à Ubuntu pour analyse. (Y compris les données sensibles) | Pas présent sur le système, sinon  sudo systemctl disable apport.service |

A la suite de ceci,

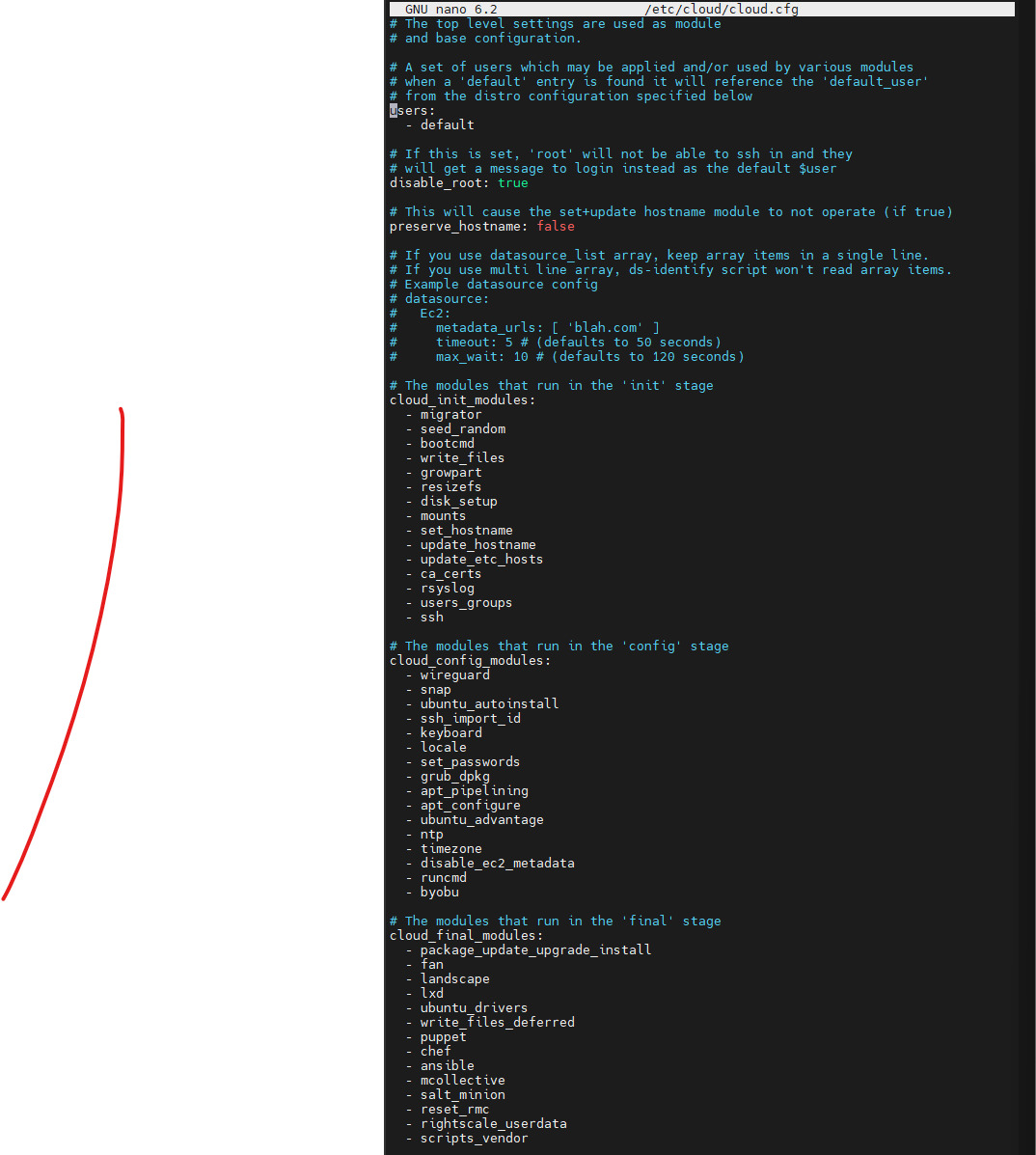
****

**J’ai considérablement réduit la surface d’attaque**

Pour améliorer la sécurité de ma configuration cloud sur le serveur, j'ai pris des mesures spécifiques pour restreindre l'accès au fichier cloud.cfg.

Tout d'abord, j'ai modifié les permissions du fichier en utilisant la commande sudo chmod 640 /etc/cloud/cloud.cfg. Cela signifie que désormais, seuls le propriétaire a le droit de lire et de modifier le fichier, tandis que les membres du groupe ne peuvent que le lire, et les autres utilisateurs ne disposent d'aucun droit d'accès.

Ensuite, pour m'assurer que seul l'utilisateur root et le groupe root ont le contrôle sur le fichier, j'ai exécuté la commande sudo chown root:root /etc/cloud/cloud.cfg. Cette étape est essentielle car elle garantit que les permissions accordées ne permettent pas aux utilisateurs non autorisés de lire ou de modifier ce fichier de configuration sensible.



J'ai également procédé à l'initialisation de cloud-init en mode local avec la commande sudo cloud-init -d init --local, ce qui permet de valider les configurations sans dépendre de ressources externes.

En conclusion, après avoir examiné les configurations de cloud-init et du service cloud-init-hotplugd.service, je me suis assuré qu'ils sont correctement sécurisés et à jour. J'ai également pris en compte la possibilité de consulter la documentation de cloud-init ou de contacter le support technique de mon fournisseur de services cloud pour obtenir des conseils supplémentaires sur la sécurisation de ces services. Si la fonctionnalité de branchement à chaud n'est pas nécessaire pour mes opérations, je pourrais envisager de désactiver ce service pour réduire encore plus les risques.

sudo chmod 640 /etc/cloud/cloud.cfg

sudo chown root:root /etc/cloud/cloud.cfg

**Sécurisation du Service ssh**

Pour sécuriser la configuration SSH, je désactive l'authentification basée sur l'hôte et les mots de passe vides, je limite l'utilisation des fichiers rhosts, je bloque l'authentification par mot de passe au profit des clés SSH, je désactive le transfert X11 et je m'assure que les configurations de PAM et de bannières sont correctes et sécurisées. Enfin, je redémarre le service SSH pour appliquer les modifications.

SSH :

- Commande :

- sudo sed -i 's/^#AllowTcpForwarding yes/AllowTcpForwarding no/' /etc/ssh/sshd\_config

- sudo sed -i 's/^#ClientAliveCountMax 3/ClientAliveCountMax 2/' /etc/ssh/sshd\_config

- sudo sed -i 's/^#LogLevel INFO/LogLevel VERBOSE/' /etc/ssh/sshd\_config

- sudo sed -i 's/^#MaxAuthTries 6/MaxAuthTries 3/' /etc/ssh/sshd\_config

- sudo sed -i 's/^#MaxSessions 10/MaxSessions 2/' /etc/ssh/sshd\_config

- sudo sed -i 's/^#Port 22/Port 2222/' /etc/ssh/sshd\_config

- sudo sed -i 's/^#TCPKeepAlive yes/TCPKeepAlive no/' /etc/ssh/sshd\_config

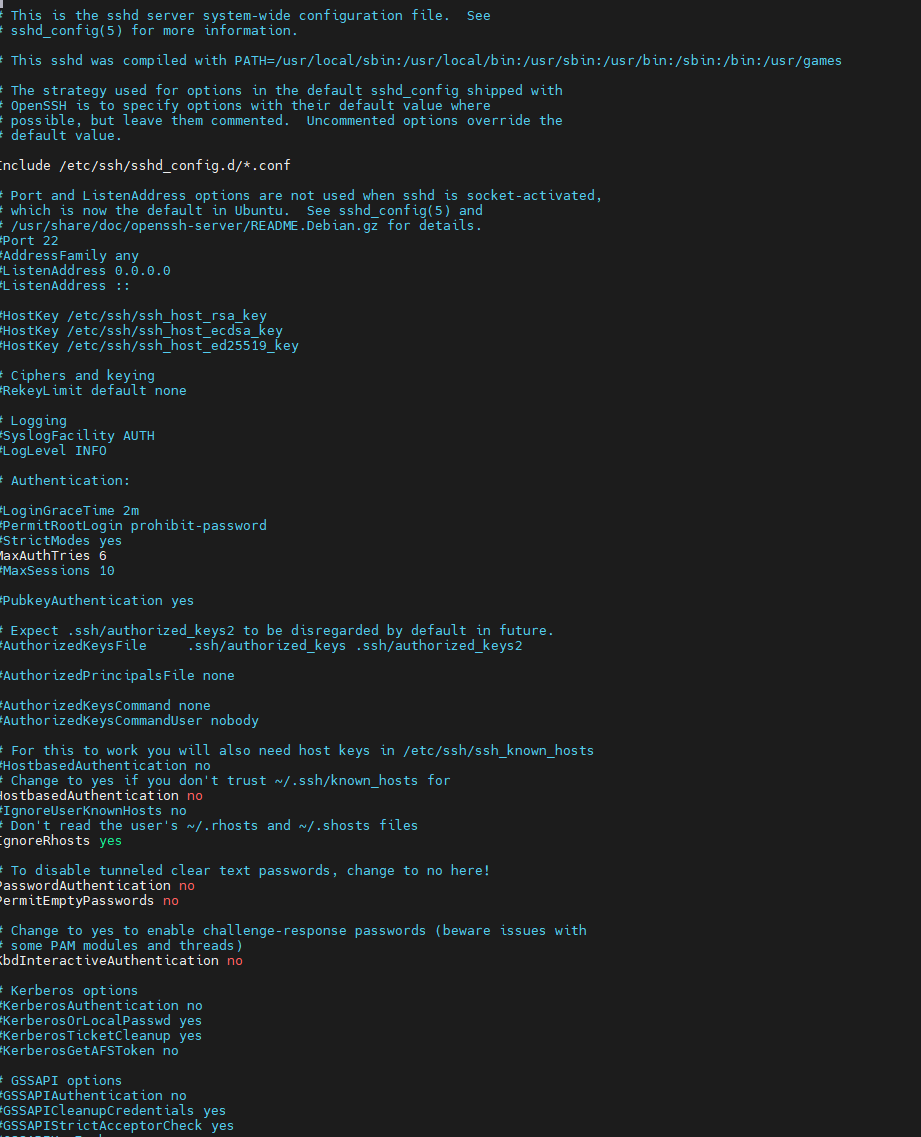
- sudo sed -i 's/^#AllowAgentForwarding yes/AllowAgentForwarding no/' /etc/ssh/sshd\_config

- sudo systemctl restart sshd

- Service : SSH (Secure Shell).

- Vulnérabilité : Configurations SSH par défaut ou laxistes.

- Attaque protégée : Protège contre les accès non autorisés, les redirections de ports malveillantes et autres exploits liés à SSH.

****

**1.2. Kernel Hardening**

J'ai identifié que la configuration par défaut de umask dans /etc/login.defs pourrait être rendue plus stricte, en utilisant par exemple une valeur de 027. Cette modification vise à renforcer la sécurité en restreignant les permissions par défaut pour les nouveaux fichiers et répertoires créés par les utilisateurs, limitant ainsi l'accès des autres utilisateurs aux fichiers personnels. Pour plus d'informations, je me suis référé à la documentation de Lynis disponible sur CISOfy.

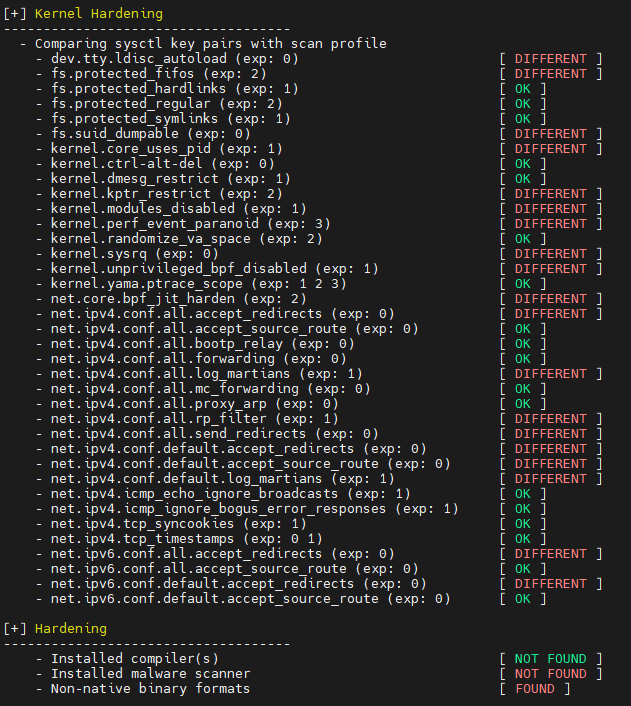
Ensuite, pour minimiser les risques liés aux services inutiles, j'ai pris la décision de masquer systemd-rfkill.service et systemd-rfkill.socket en utilisant les commandes sudo systemctl mask systemd-rfkill.service et sudo systemctl mask systemd-rfkill.socket. Cette action empêche leur démarrage automatique, ce qui réduit la surface d'attaque potentielle du système.

Enfin, je suis conscient que pour appliquer toute modification au noyau Linux, il est nécessaire d'exécuter la commande sudo sysctl -p. Cette étape est cruciale pour activer immédiatement les changements de configuration sans avoir à redémarrer le système, assurant ainsi que les ajustements de sécurité prennent effet immédiatement.ces invites et utiliser l'utilisation de l'authentification à deux facteurs.

**s**

**Rappel : pour appliquer tout modification au kernal, executer la commande**

**sudo sysctl -p**

****

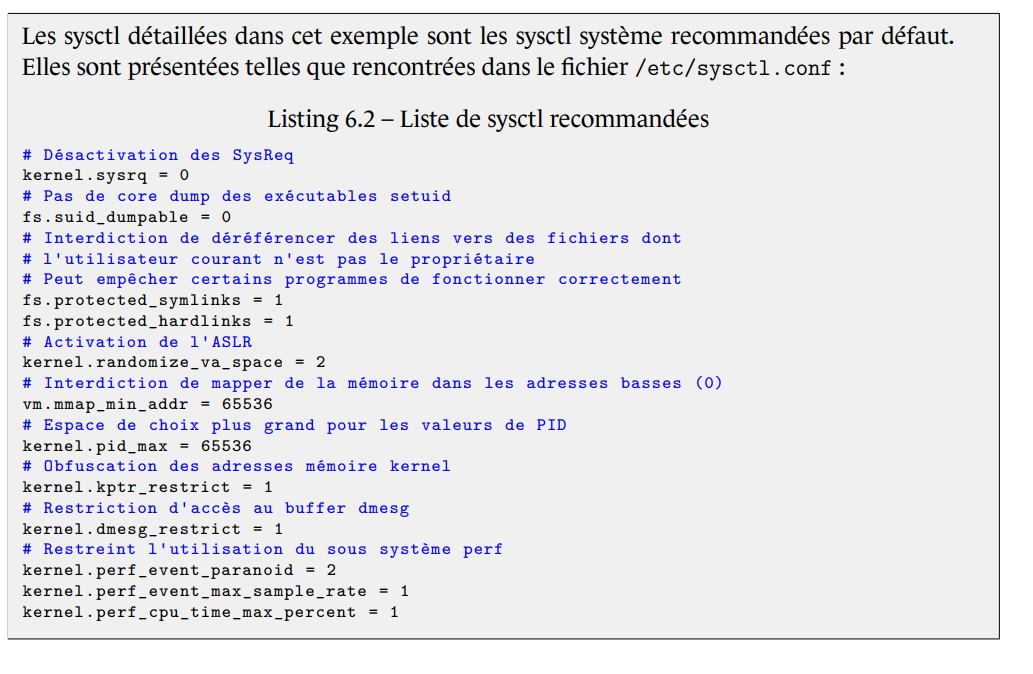
Pour garantir la conformité de ma configuration système aux recommandations de sécurité, j'ai entrepris deux étapes importantes :

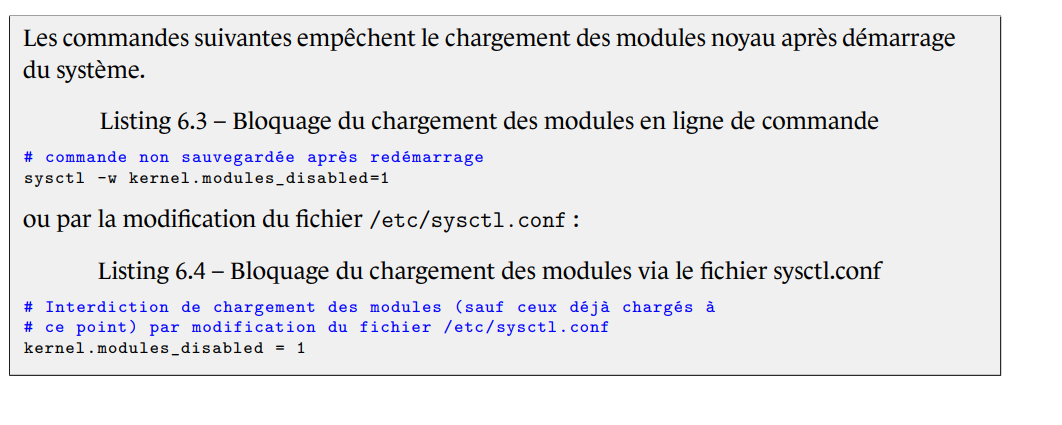
Première étape : J'ai supprimé le fichier /etc/sysctl.conf. Cette action a pour but de repartir sur une base vierge pour la configuration du noyau, me permettant ainsi d'éliminer toute directive obsolète ou non sécurisée qui pourrait y figurer.

Seconde étape : J'ai appliqué les directives spécifiques de l'ANSSI (Agence nationale de la sécurité des systèmes d'information) concernant la configuration du noyau. L'ANSSI fournit des recommandations détaillées pour renforcer la sécurité des systèmes d'information. En suivant ces directives, je m'assure que la configuration de mon système est conforme aux standards de sécurité nationaux et internationaux, optimisant ainsi la protection contre les vulnérabilités et les attaques potentielles.

Ces étapes sont essentielles pour assurer que mon système est configuré de manière à respecter les meilleures pratiques de sécurité, conformément aux exigences de l'ANSSI et aux principes généraux de durcissement des systèmes d'information.

****

****

****

**3e étape : ajuster les recommandation e Lynis**

J'ai défini kernel.core\_uses\_pid=1 pour associer les dumps de core à l'ID du processus, améliorant ainsi la traçabilité et la sécurité.

Pour kernel.perf\_event\_paranoid, j'ai opté pour une valeur de 3, ce qui restreint l'accès aux événements de performance aux utilisateurs ayant les privilèges nécessaires. Cela diverge légèrement de la recommandation de l'ANSSI qui suggère une valeur de 2, mais j'ai choisi un niveau de restriction plus élevé pour maximiser la sécurité.

J'ai activé net.ipv4.conf.default.log\_martians=1 pour enregistrer les paquets suspects, ce qui aide à détecter les tentatives d'accès non autorisées ou les configurations incorrectes du réseau.

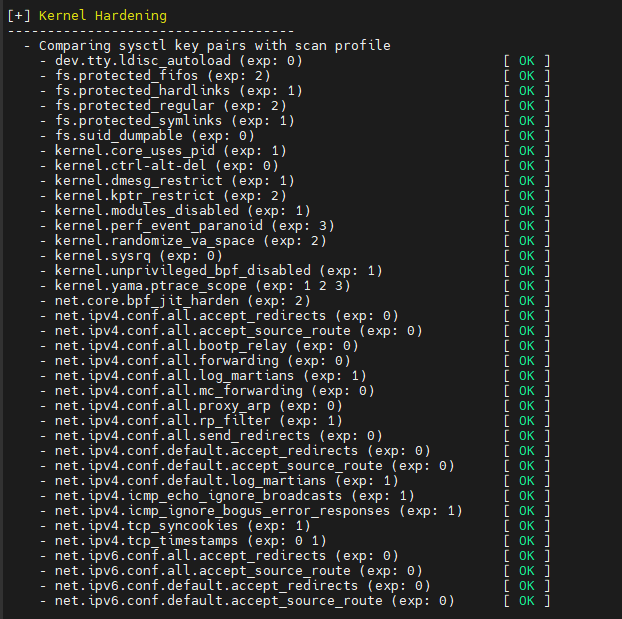
Avec dev.tty.ldisc\_autoload=0, j'ai désactivé le chargement automatique des disciplines de ligne TTY, réduisant le risque d'exploitations basées sur des modules automatiquement chargés.

J'ai configuré fs.protected\_fifos=2 et kernel.kptr\_restrict=2 pour renforcer la protection contre les écritures non autorisées dans les FIFOs et limiter l'exposition des adresses du noyau, respectivement.

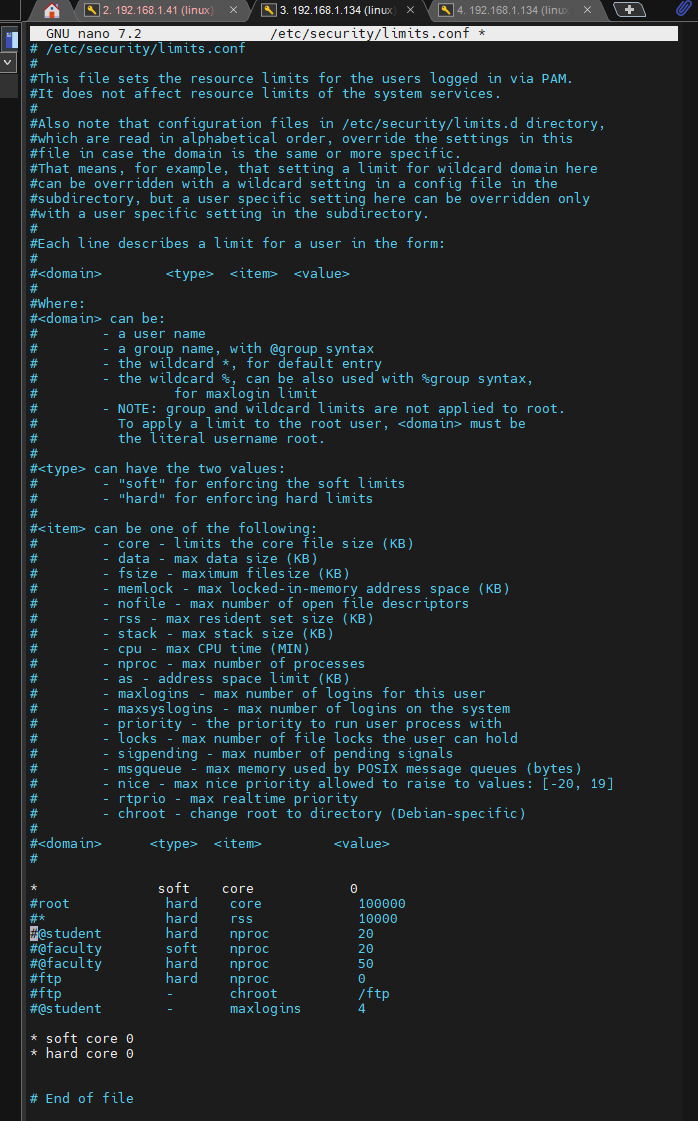
En définissant kernel.unprivileged\_bpf\_disabled=1 et net.core.bpf\_jit\_harden=2, j'ai désactivé l'utilisation des filtres BPF (Berkeley Packet Filter) par les utilisateurs non privilégiés et renforcé la sécurité du compilateur JIT BPF, réduisant ainsi le risque d'exploitations malveillantes.

Suite à ces modifications, j'ai vérifié la configuration à l'aide de Lynis et d'autres outils d'audit, confirmant que mon système est désormais conforme aux recommandations de sécurité. Toutes les modifications apportées ont pour but de durcir la sécurité du système, tout en veillant à maintenir un équilibre entre sécurité et fonctionnalité. Le résultat est satisfaisant : mon système est sécurisé et prêt à faire face aux défis de sécurité actuels.

**Le résultat Lynis pour le kernel en vue du hardening (Tout les services présent)**

****

**1.3 Hardening Guide Anssi**

****

Pour renforcer la surveillance de la sécurité de mon système et détecter la présence éventuelle de rootkits, j'ai décidé d'installer deux "chiens de garde" logiciels spécialisés dans la détection de ces menaces. Voici les commandes que j'ai utilisées

Installation de RKHunter (Rootkit Hunter) :

sudo apt-get install rkhunter

RKHunter est un outil qui scanne le système à la recherche de rootkits, backdoors et diverses vulnérabilités. Il utilise des tests de signatures ainsi que des vérifications de hash pour détecter les modifications suspectes.

Installation de Chkrootkit :

sudo apt-get install chkrootkit

Chkrootkit est un autre outil qui permet de chercher localement les signes de compromission du système par des rootkits. Il effectue plusieurs tests pour vérifier la présence de modifications suspectes ou de logiciels malveillants connus.

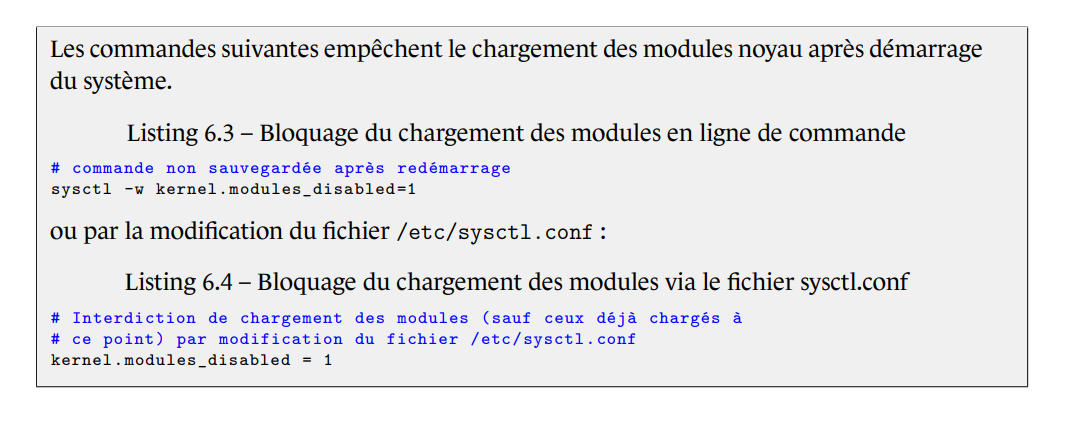
Installation de Audit D :

sudo apt-get install auditd

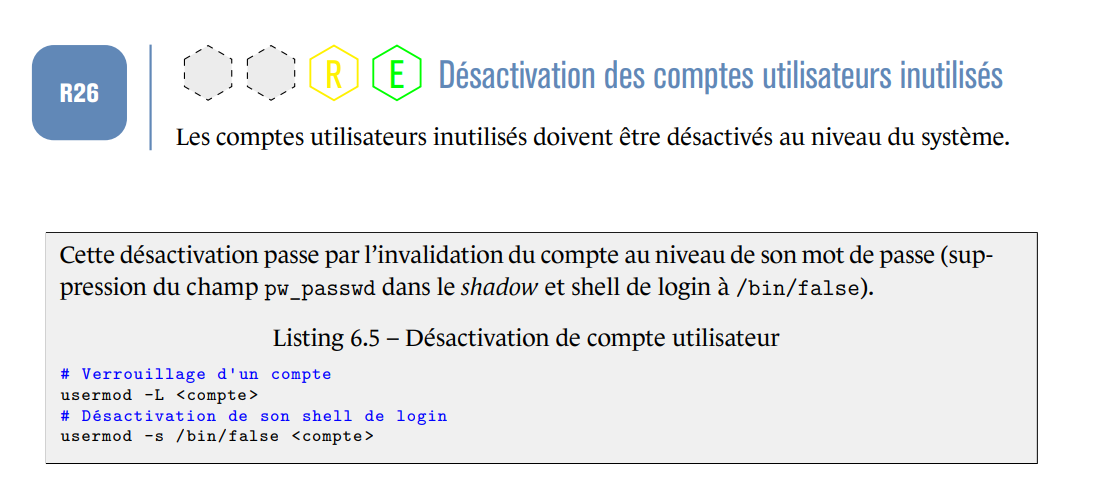
La commande sudo apt-get install auditd sert à installer le service d'audit auditd sur les systèmes Linux basés sur Debian ou Ubuntu. Elle permet de surveiller les activités de sécurité sur le système, essentielles pour détecter les activités suspectes et assurer la conformité réglementaire.

En installant ces trois outils, je m'assure une couche supplémentaire de surveillance pour détecter activement les menaces potentielles et les activités malveillantes sur mon système. L'utilisation régulière de ces outils, combinée à des mises à jour de sécurité et à d'autres pratiques de durcissement, contribue grandement à maintenir l'intégrité et la sécurité de mon système Linux.

**J’ai passer la commande suivante :**

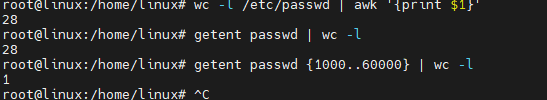
****

**6.3.1 Désactivation des comptes utilisateurs inutilisés**

****

**wc -l /etc/passwd | awk '{print $1}'**

**consiste à compter les lignes dans le fichier** /etc/passwd**, qui contient une entrée pour chaque utilisateur du système.**

****

**getent passwd | wc -l**

**getent passwd {1000..60000} | wc -l**

**Cette commande filtre les utilisateurs ayant des UID entre 1000 et 60000, ce qui devrait inclure la plupart des comptes "humains", et exclure les comptes système qui ont généralement des UID en dessous de 1000**

**Sécuriser le Répertoire /etc/sudoers.d**

**Les permissions recommandées pour le répertoire /etc/sudoers.d sont 755 (rwxr-xr-x) ou encore plus restrictives, comme 750 (rwxr-x---), et appartenant à l'utilisateur root et au groupe root. Ces permissions permettent au propriétaire (root) de lire, écrire et exécuter, tandis que les membres du groupe et les autres utilisateurs peuvent seulement lire et exécuter, sans pouvoir écrire. Si 750, seul root et les utilisateurs du groupe spécifié peuvent lire et exécuter, rendant le répertoire inaccessible aux autres utilisateurs.**

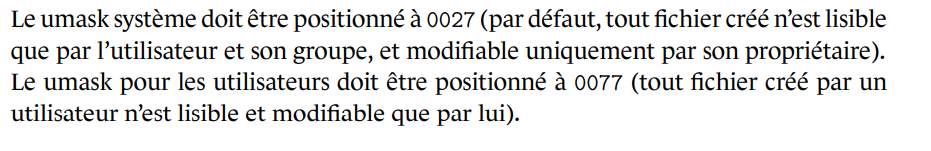
**Pour corriger les permissions et la propriété, procédez comme suit :**

**ls -ld /etc/sudoers.d**

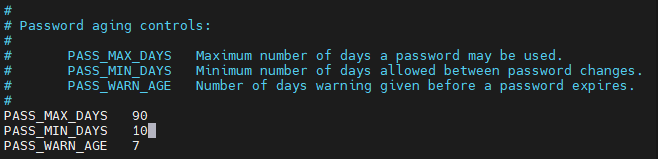
**sudo chmod 750 /etc/sudoers.d**

**udo systemctl mask systemd-rfkill.service**

**sudo systemctl mask systemd-rfkill.socket**





****

1.4 Suggestion Lynis

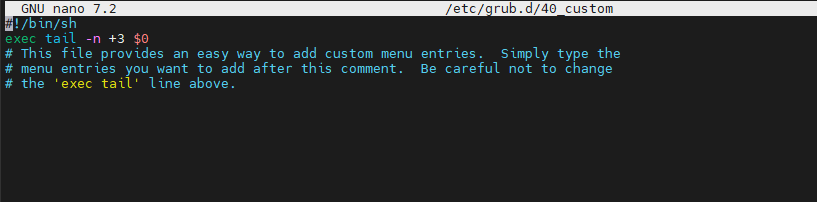
Sécurisation du démarrage avec GRUB :

- Commande : echo 'set superusers="nom\_utilisateur"' | sudo tee -a /etc/grub.d/40\_custom && echo 'password\_pbkdf2 nom\_utilisateur grub.pbkdf2.sha512.hash' | sudo tee -a /etc/grub.d/40\_custom

- Service : Chargeur de démarrage GRUB.

- Vulnérabilité : Modification non autorisée des paramètres de démarrage.

- Attaque protégée : Empêche un attaquant d'altérer le démarrage pour contourner les mécanismes de sécurité.



Gestion des utilisateurs et authentification :

- Commande :

- sudo sed -i 's/^# SHA\_CRYPT\_MAX\_ROUNDS.\*/SHA\_CRYPT\_MAX\_ROUNDS 5000/' /etc/login.defs

- sudo apt-get install libpam-cracklib

- sudo chage -M 90 linux

- Service : Authentification utilisateur.

- Vulnérabilité : Mots de passe faibles ou jamais expirés.

- Attaque protégée : Renforce la sécurité des mots de passe, empêchant les attaques par force brute ou l'utilisation indéfinie de mots de passe compromis.

Bannières et avertissements :

- Commande :

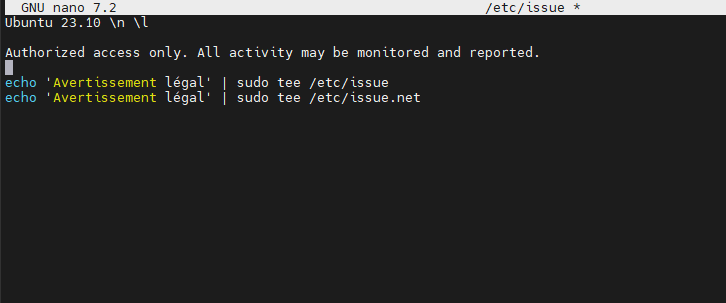
- echo 'Avertissement légal' | sudo tee /etc/issue

- echo 'Avertissement légal' | sudo tee /etc/issue.net

- Service : Connexions au système.

- Vulnérabilité : Utilisateurs non informés des politiques ou des conséquences légales.

- Attaque protégée : Décourage l'accès non autorisé en informant les utilisateurs des implications légales.



Comptabilité :

- Commande :

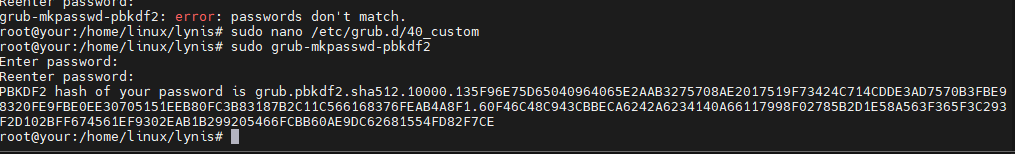
- sudo apt-get install sysstat

- sudo nano /etc/audit/audit.rules

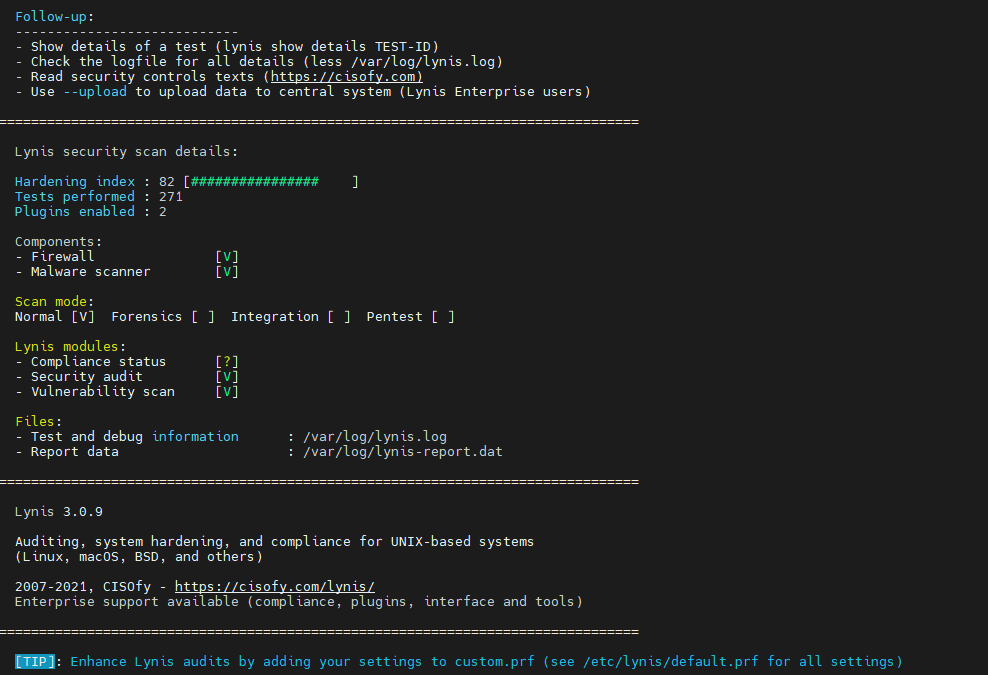
- Service : Surveillance et audit du système.

- Vulnérabilité : Manque de visibilité sur les actions système.

- Attaque protégée : Permet une analyse détaillée des événements, aidant à détecter et à enquêter sur les activités suspectes.



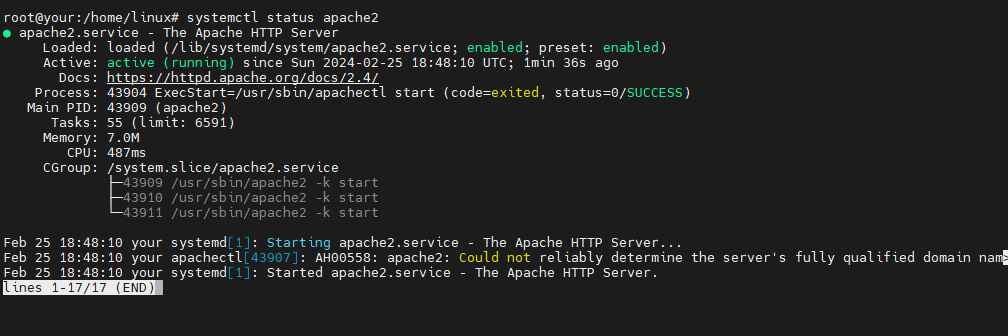
J’ai donc obenue une note de 82 sur Lynis

****

**2. Partie TLS/SSL.**

En effectuant la configuration SSL pour mon serveur Apache, j'ai rencontré quelques problèmes que j'ai réussi à résoudre. Initialement, en essayant d'activer le site par défaut pour SSL avec la commande a2ensite default-ssl, j'ai reçu un message indiquant que le dossier 'sites-enabled/' n'existait pas. J'ai donc vérifié la présence du répertoire avec ls sites-enabled/ et j'ai confirmé son existence, ce qui signifie que la commande a2ensite s'est mal exécutée à cause d'une faute de frappe ou d'un problème similaire.

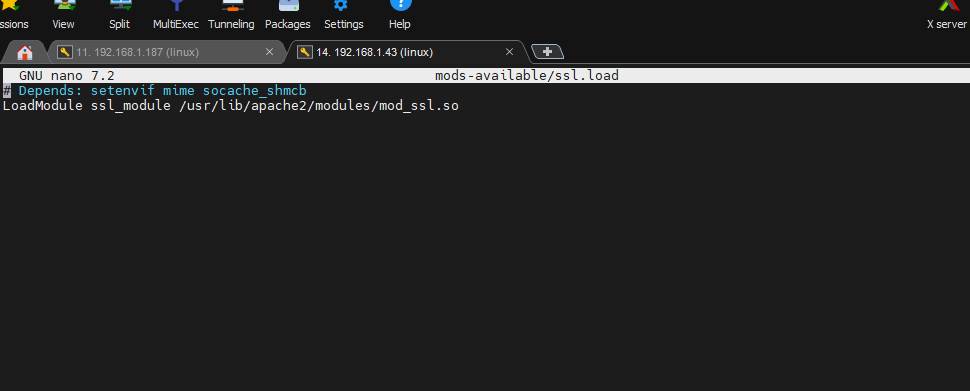
Installation du service apache2

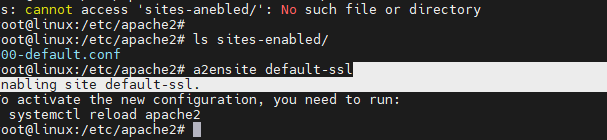


Après un examen approfondi de la configuration SSL/TLS de notre serveur, nous avons mis en œuvre des mesures de renforcement conformément aux recommandations de Mozilla pour la configuration SSL. Nous avons choisi le modèle "intermédiaire" adapté à la majorité des serveurs, qui permet une excellente compatibilité tout en assurant un haut niveau de sécurité.

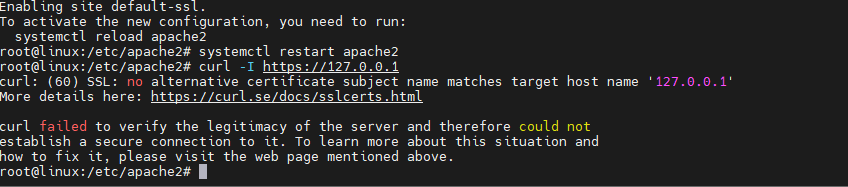
Nous avons désactivé les protocoles obsolètes et moins sécurisés tels que SSLv2, SSLv3, TLS 1.0 et TLS 1.1, garantissant ainsi que notre serveur n'offre que des protocoles modernes et sûrs tels que TLS 1.2 et TLS 1.3. De plus, nous avons appliqué une politique stricte de transport sécurisé HTTP (HSTS) pour forcer les connexions HTTPS, ce qui réduit le risque de downgrade attacks et d'autres vecteurs d'attaque associés aux connexions non sécurisées.

Après avoir corrigé l'erreur et activé le site SSL, j'ai rechargé la configuration d'Apache avec systemctl reload apache2. Pour m'assurer que le SSL fonctionnait correctement, j'ai testé l'accès à mon serveur via HTTPS en utilisant la commande curl -I https://192…

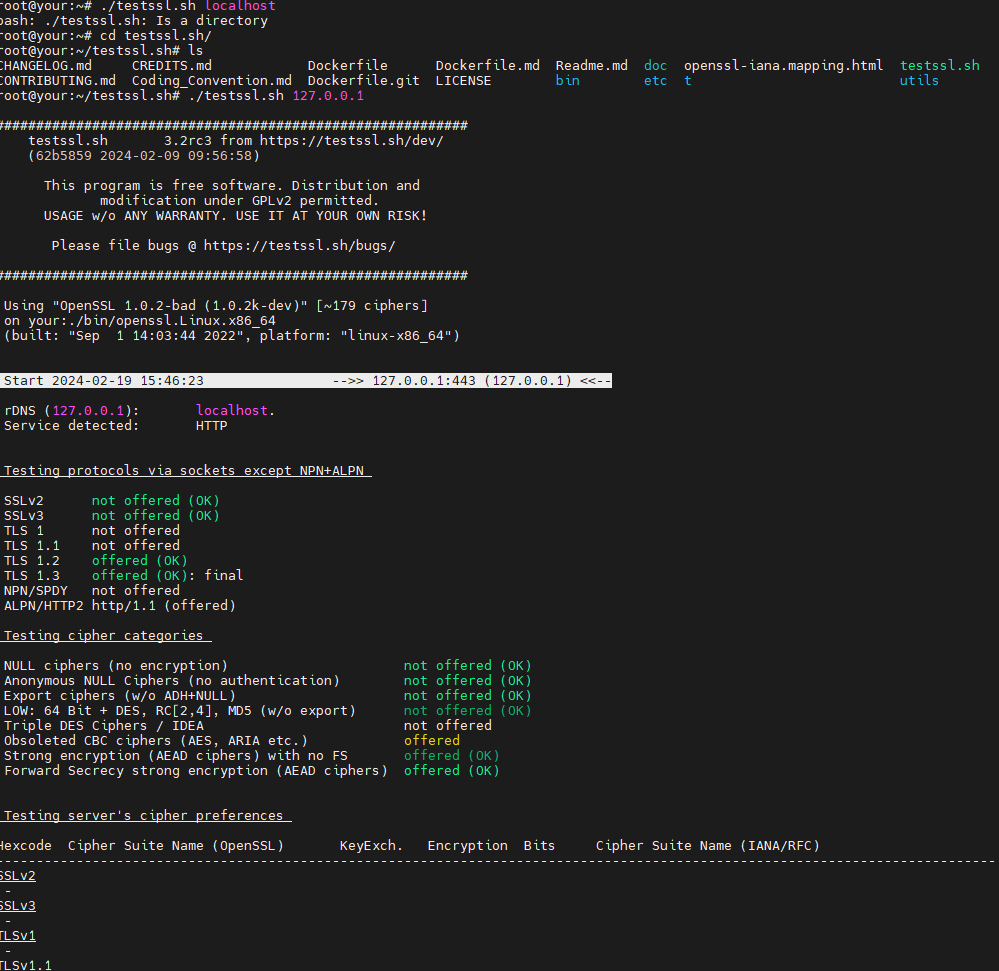




a2ensite default-ssl pour activer la configuration par défaut SSL d'Apache. Assurez-vous d'avoir le fichier default-ssl.conf dans le répertoire sites-available/.

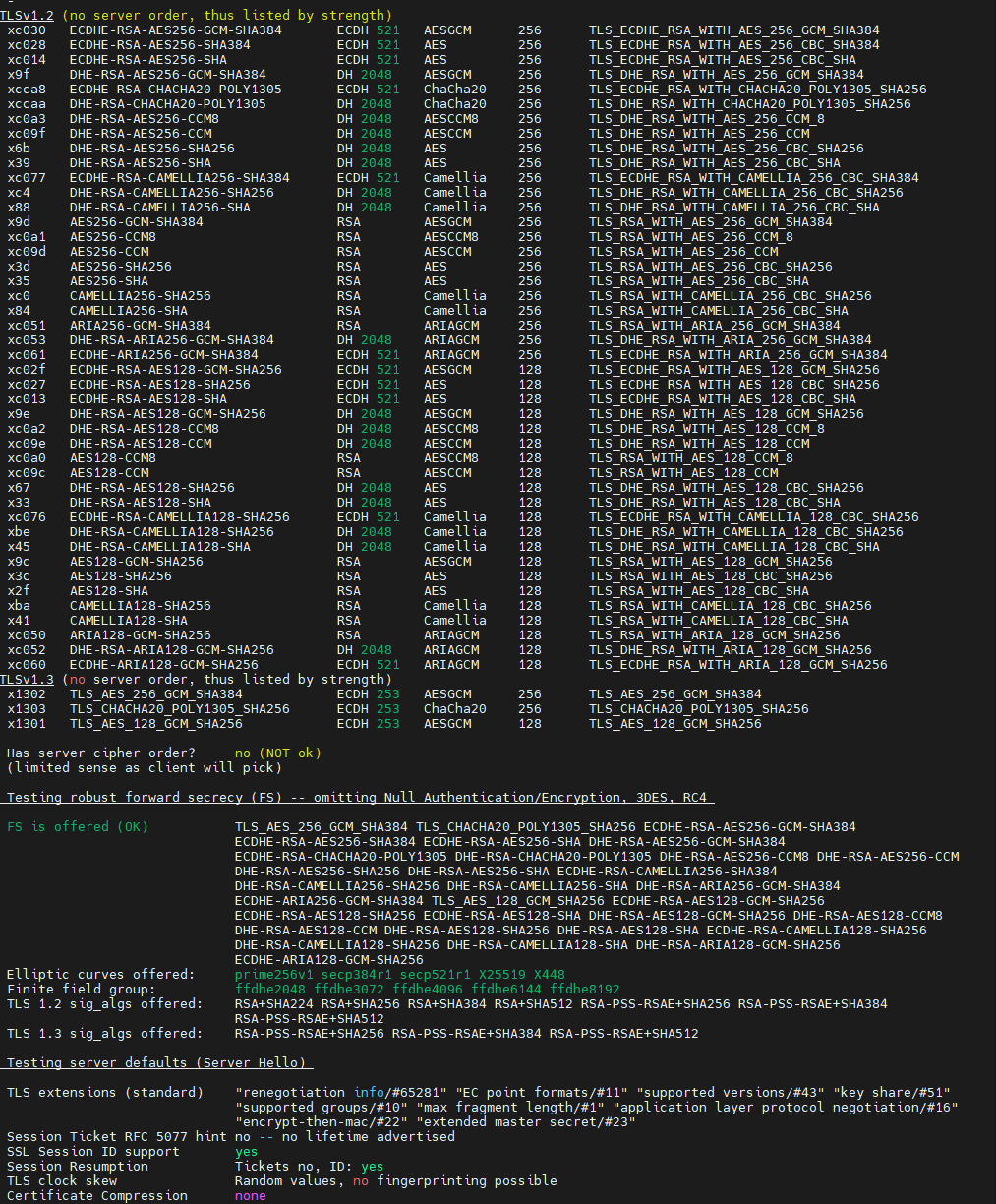


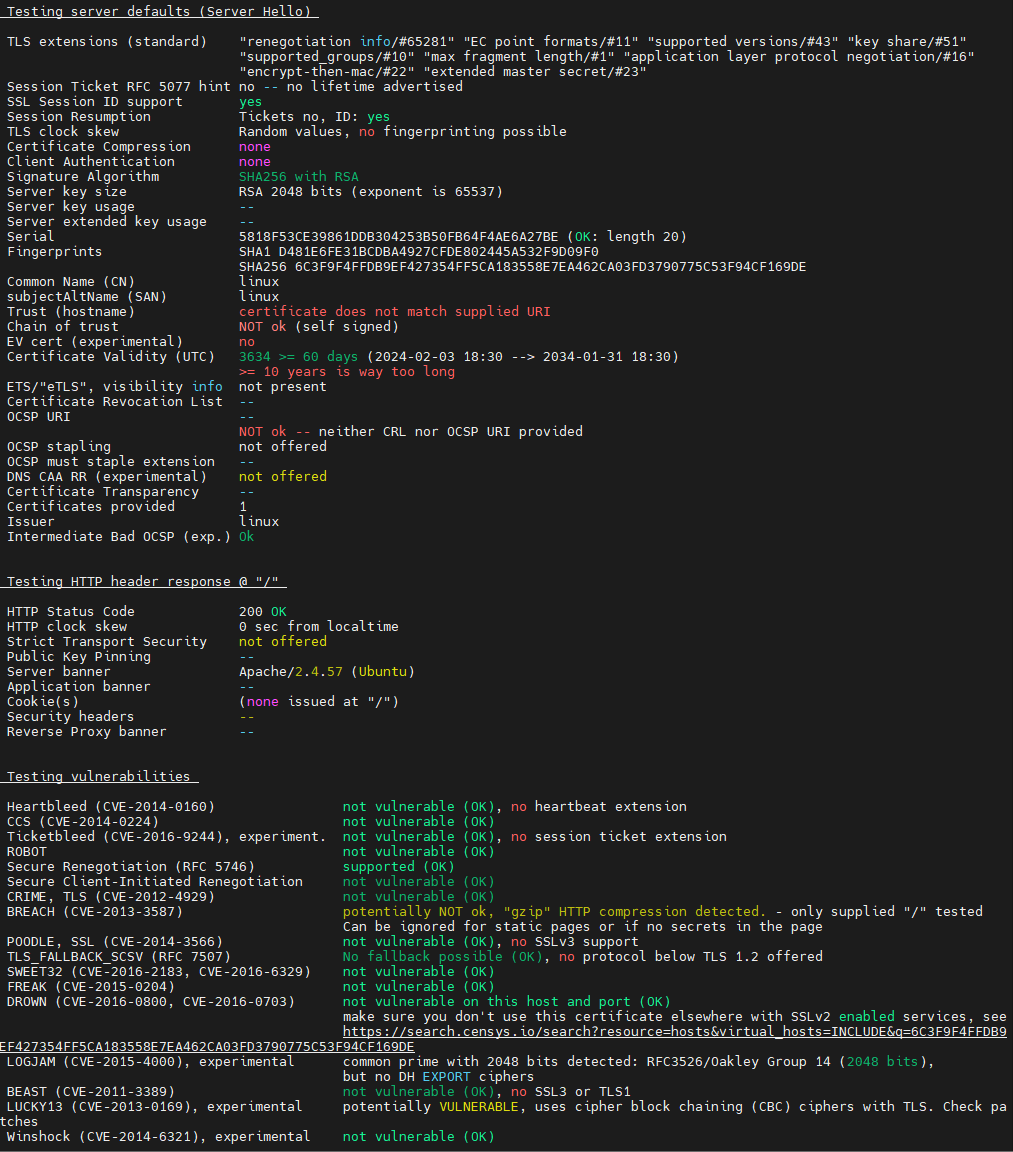
Screen du scan testssh

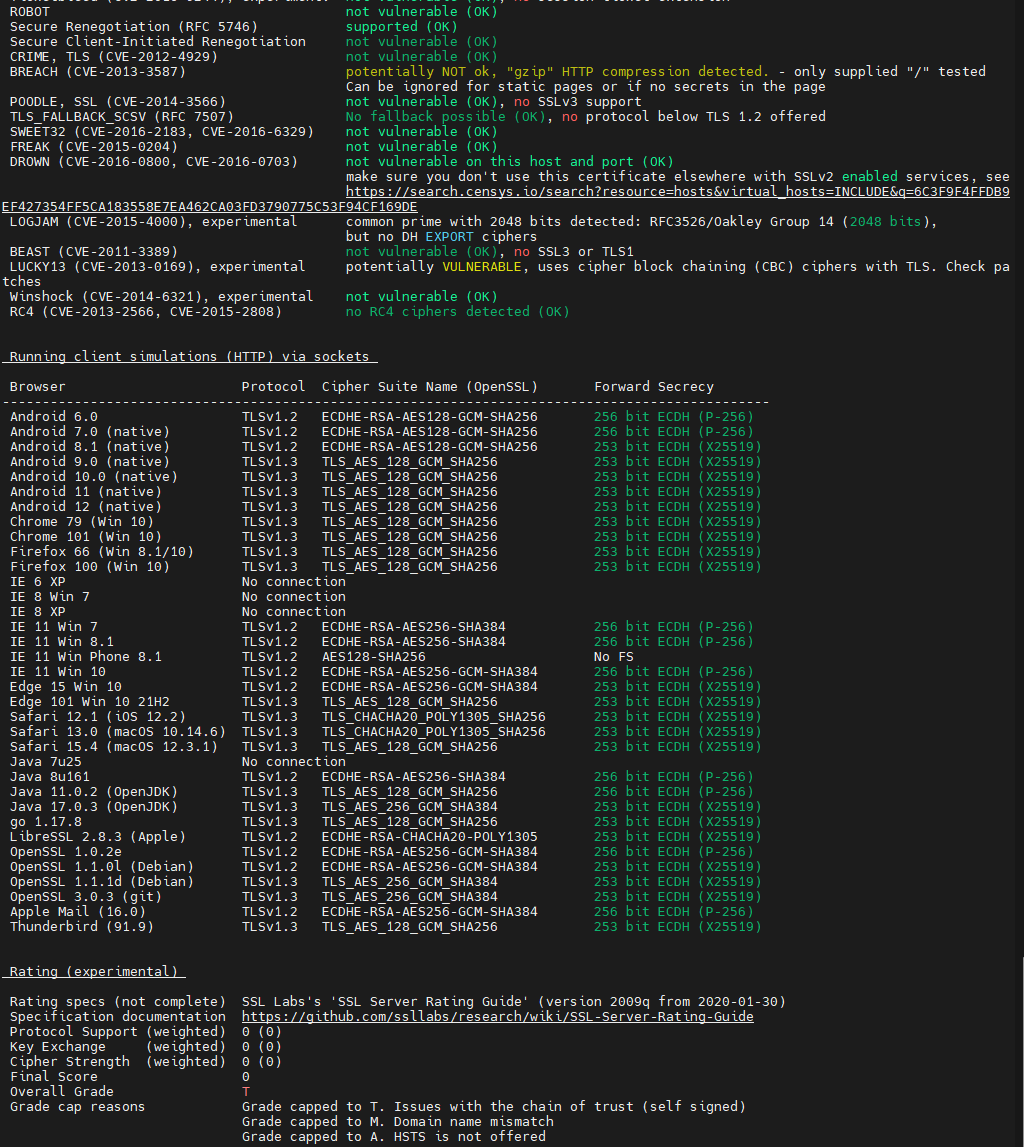


Tout est ok ici en appliquant

****

****

****

****

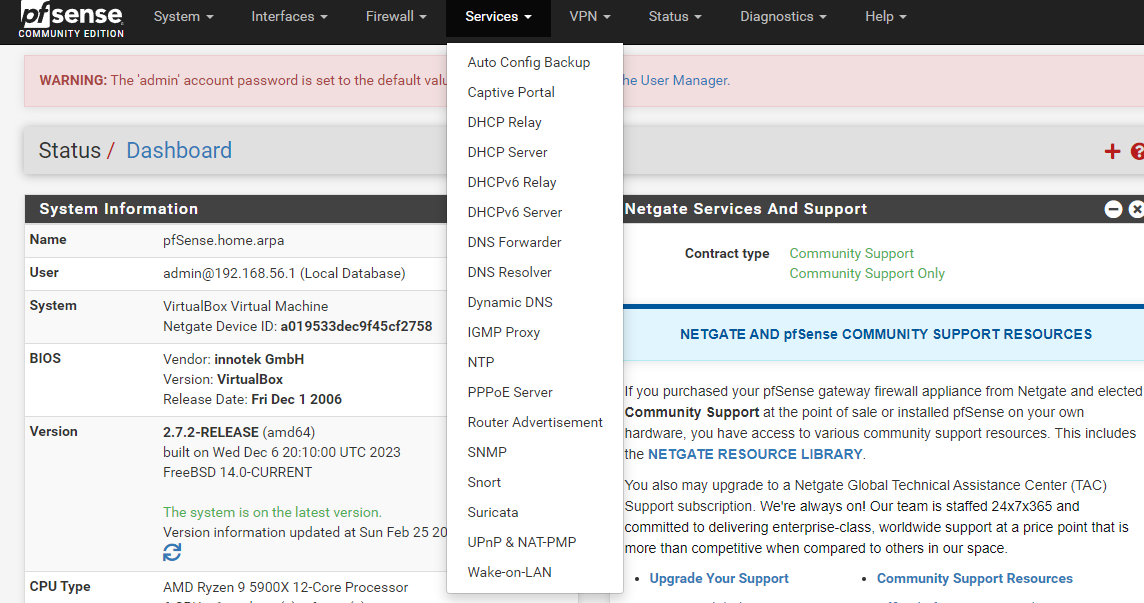
Après avoir effectué un scan SSL de mon serveur, je constate qu'il ne présente aucune vulnérabilité connue. Le scan a minutieusement vérifié les protocoles, les suites de chiffrement et d'éventuelles vulnérabilités spécifiques telles que Heartbleed, POODLE et BEAST. Heureusement, mon serveur est bien configuré pour éviter ces failles de sécurité.

**3. La partie réseau a été traitée dans le TP3. Rajouter sur le parefeu pfSense les points suivants :**

**a. Règles de filtrage qui permettent de détecter et bloquer les balayages réseau. Vous pouvez utiliser un IDS pour la détection des intrusions.**

Activation et configuration de l'IDS

Suite à notre recherche, j’ai trouvé deux IDS, Snort et Suricata. Je l’est ait correctement installé et activé sur votre système pfSense. (depuis l'interface web de pfSense) :



Pour la suite des question, je pense utilisé Suricata, qui est un très bon IDS.

2. Configuration des règles de détection de balayages réseau

Ensuite, j'ai configuré des règles spécifiques dans Suricata pour détecter les balayages réseau. Ces règles sont essentielles car elles me permettent d'identifier des tentatives d'accès non autorisées ou des scans de ports. J'ai filtré et activé des règles liées aux scans de réseau dans l'onglet Rules.

3. Règles de filtrage sur pfSense

Je me suis rendu dans Firewall > Rules pour configurer ces règles supplémentaires.

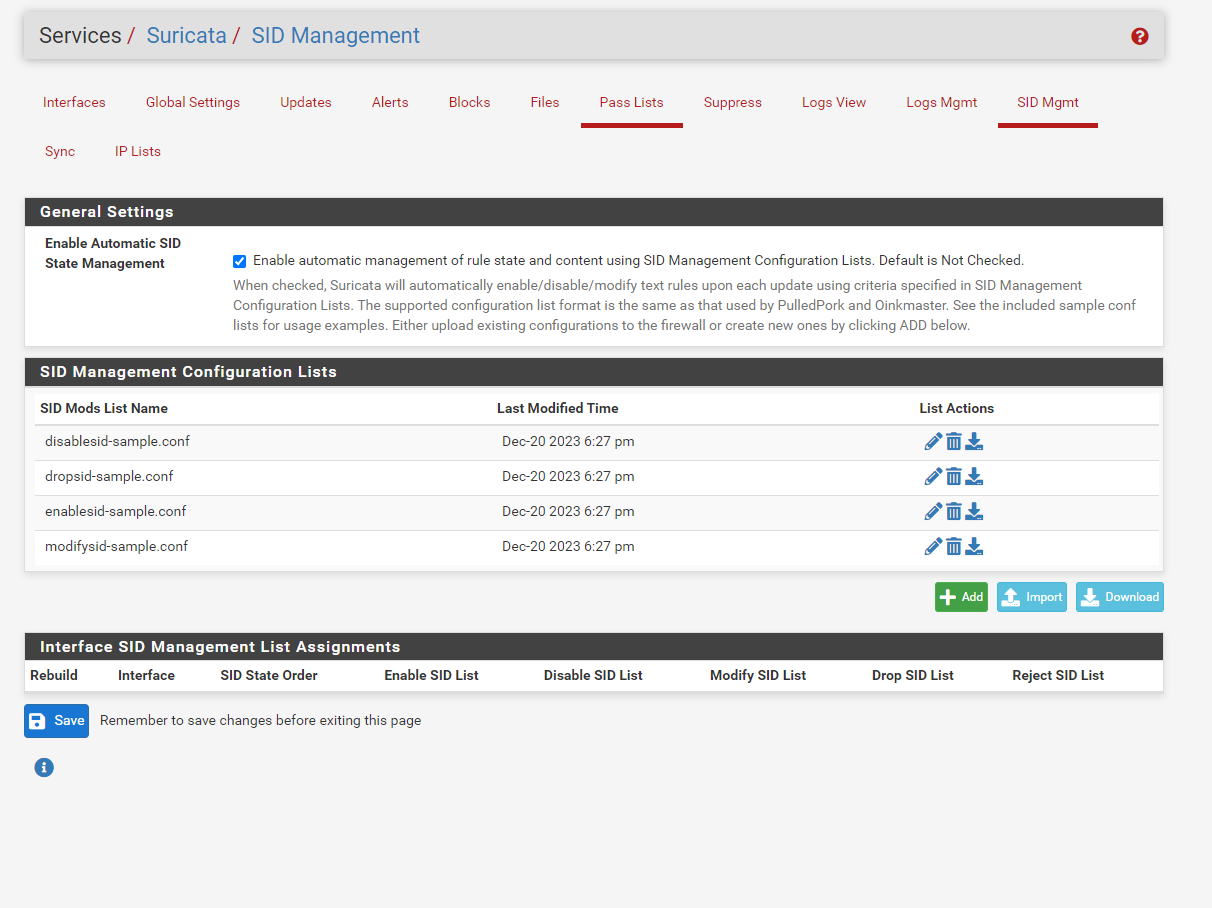
En plus de l'IDS, configurez des règles de filtrage sur pfSense pour bloquer les adresses IP connues pour leurs activités de balayage réseau.

Accédez à Firewall > Rules.

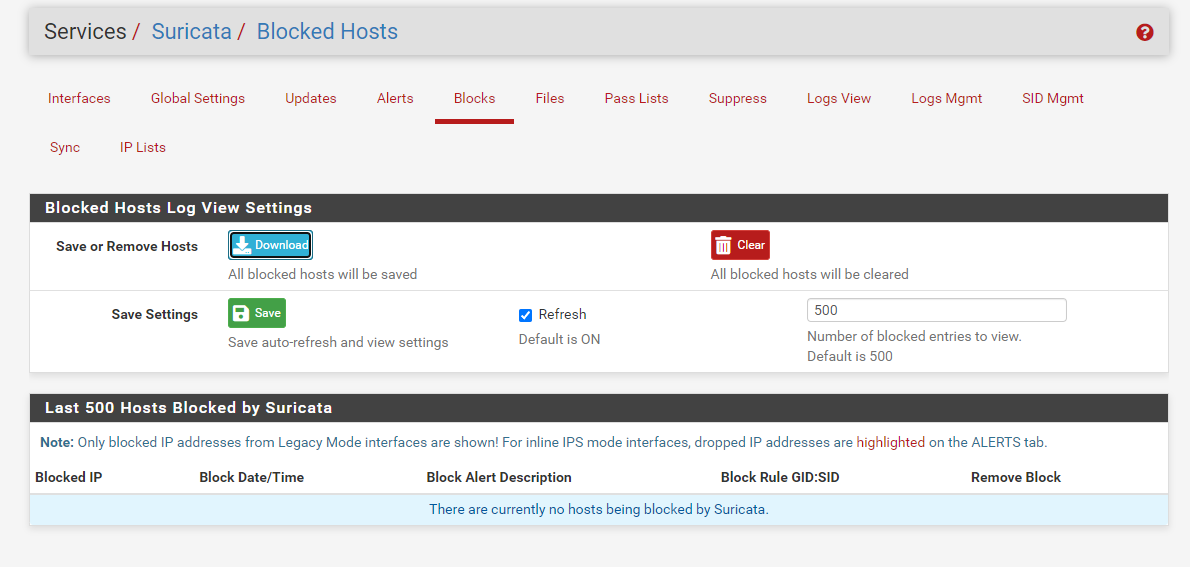
J'ai configuré Suricata pour qu'il génère des alertes ou bloque automatiquement le trafic suspect, en m'assurant que ces actions sont bien reflétées dans les journaux de pfSense.

**B. Ajouter les règles qui permettent de protéger les parefeux et les services internes des attaques DoS. Expliquer votre méthode.**

Je commence par identifier et sélectionner des règles spécifiques au sein de Suricata qui sont conçues pour détecter les comportements associés aux attaques DoS. Sur l'interface de pfSense, je navigue dans Services > Suricata puis SID Management. Ici, je peux filtrer et activer des règles qui sont spécifiquement conçues pour détecter les attaques DoS. Des exemples de ces règles incluent celles qui surveillent les tentatives de connexion répétitives et rapides ou les volumes anormaux de trafic en peu de temps.

****

Dans Services > Suricata > Blocked Hosts, j'active l'option de blocage automatique des adresses IP qui déclenchent les règles de détection de DoS. Cela permet de bloquer immédiatement les sources d'attaque potentielles sans intervention manuelle.

****

Après avoir configuré les règles et les seuils, je teste la configuration pour m'assurer qu'elle fonctionne correctement. Je peux simuler des attaques DoS en utilisant des outils comme hping3

sudo hping3 -S --scan 1-1000 scanme.nmap.org

**C. Ecrire des règles de protection contre les attaques par usurpation d’adresse IP**

Pour écrire des règles de protection contre ce type d'attaque dans Suricata, voici ce que je ferais :

Identifier le trafic légitime : Je commencerais par déterminer quelles sont les plages d'adresses IP légitimes pour mon réseau. Toute adresse source ne faisant pas partie de ces plages et tentant de communiquer avec mon réseau interne serait suspecte.

Ecrire des règles de détection : J'ai écrit des règles dans Suricata pour détecter les paquets avec des adresses IP sources qui ne devraient pas être routées sur Internet ou qui ne font pas partie des plages d'adresses légitimes pour mon réseau. Par exemple, si mon réseau interne utilise la plage 192.168.1.0/24, une règle génère une alerte pour tout trafic entrant prétendant provenir de cette plage, car ce trafic devrait normalement être interne et non routé sur Internet.

Implémenter l'ingénierie de trafic : Pour les réseaux où je connais les chemins habituels du trafic, je pourrais écrire des règles qui alertent ou bloquent le trafic provenant de chemins inattendus, ce qui pourrait indiquer une usurpation.

Voici la règle Suricata pour détecter une tentative d'usurpation d'adresse IP :

**alert ip ![192.168.1.0/24] any -> 192.168.56.70 any (msg:"Tentative d'usurpation d'adresse IP détectée"; sid:1000001; rev:1;)**

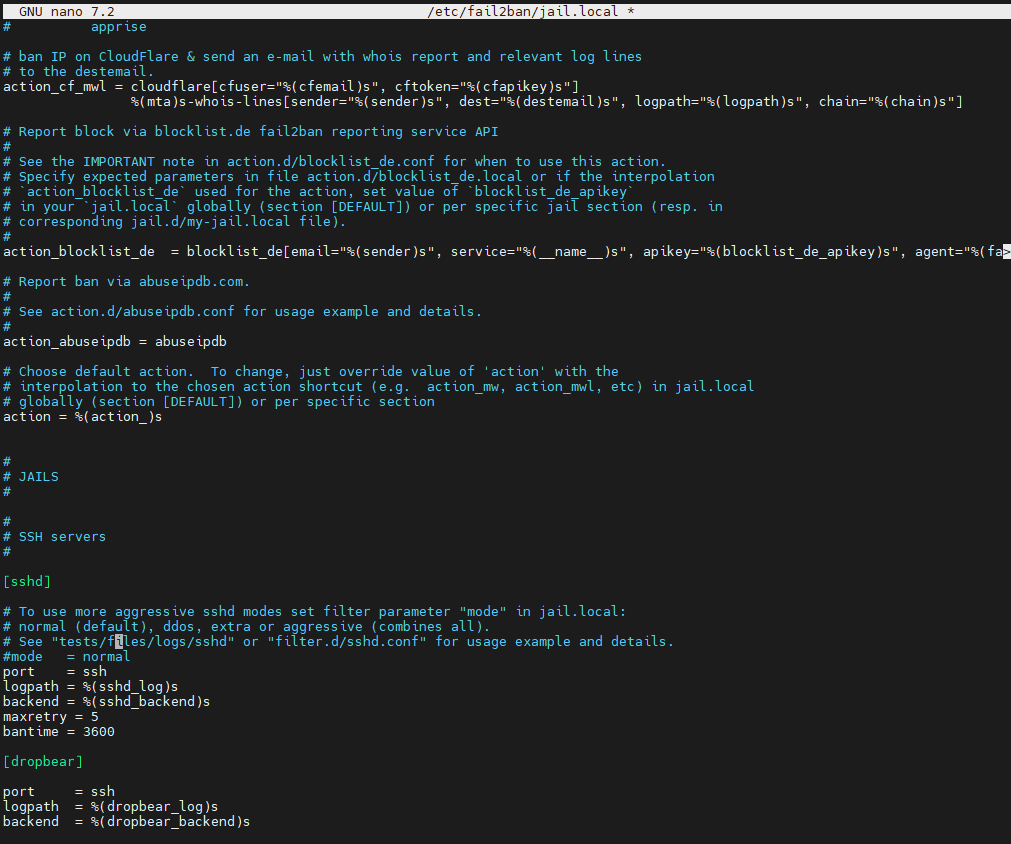
**D. Le parefeu est accessible par SSH. Faites le nécessaire afin de bloquer une**

**adresse IP pour une heure après 5 tentatives de connexions SSH échouées.**

Pour bloquer une adresse IP pendant une heure après 5 tentatives de connexion SSH échouées, je pourrais utiliser un outil comme fail2ban (Que j’ai précédemment installer) sur le pare-feu. fail2ban est un logiciel qui analyse les fichiers de journalisation (logs) pour détecter des motifs d'échecs de connexion et applique des règles de pare-feu pour bloquer l'adresse IP source correspondante. Voici les étapes générales que je suivrais :

Installer fail2ban : Si fail2ban n'est pas déjà installé sur le pare-feu, je me connecterais via SSH et utiliserais le gestionnaire de paquets du système pour l'installer. Par exemple, sur un système basé sur Debian, je pourrais utiliser apt-get install fail2ban.

Configurer fail2ban : Je créerais ou éditerais un fichier de configuration pour fail2ban dans /etc/fail2ban/jail.local. Voici un exemple de configuration :



logpath = par default

maxretry = 5

bantime = 3600 (bantime) à 3600 secondes (1 heure).

J’ai pu tester la configuration pour m'assurer que fail2ban bloque effectivement une adresse IP après 5 tentatives de connexion SSH échouées. (En ayant taper de mauvais identifiant…)