REPUBLIQUE DU SENEGAL



Un Peuple – Un But – Une Foi

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR DE LA RECHERCHE ET DE l'INNOVATION

UNIVERSITE GASTON BERGER DE SAINT-LOUIS



U.F.R DE SCIENCES APPLIQUEES ET DE TECHNOLOGIE

DEPARTEMENT DE PHYSIQUE APPLIQUEE

Mémoire de fin d'études pour l'obtention du

Master Ingénierie en Electronique et Télécommunications (MIETEL)

Sujet : Automatisation d'un processus de durcissement de serveur web

Présenté par :

M. Amath CAMARA

Sous la direction de :

Pr. Chérif DIALLO

Jury:

Président :	Pr. Abdou Karim DIALLO
Examinateur :	Pr. Moustapha SENE
Examinateur :	Dr. Gabriel Jean P. TEVY
Encadrant :	Pr. Cherif DIALLO

Année Académique 2022/2023

Remerciement

Au terme de ce travail, nos pensées vont à l'endroit de tous ceux qui ont participé à l'aboutissement de ce travail. Ainsi seront remerciés,

Pr. ***Chérif DIALLO*** *pour son encadrement et sa disponibilité au cours de toute la rédaction de ce mémoire.*

L'ensemble des professeurs de l'UFR SAT à l'université Gaston Berger de Saint-Louis qui ont participé de près ou de loin à ma formation.

Ma famille de m'avoir soutenu et encouragé durant toutes ces années.

Ma mère et mon père qui croient en ma modeste personne et me soutienne dans toute circonstance.

A la famille Sarr et la famille Gueye de Saint-Louis de m'avoir soutenu et encouragé.

Mes frères et sœurs et tous mes amis

Dédicaces

Je dédie ce mémoire à :

Ma mère Alima Gueye et mon père Ablaye de m'avoir donné leur amour, leur soutien et tout l'aide dont j'ai besoin dans mes études et d'avoir toujours cru en moi.

Ma sœur Fatou Camara et mon frère Bouba Camara qui m'ont soutenu et encouragé dans les périodes les plus difficiles.

Tata Khady Diouf et toute la famille Sarr de Kaolack pour leur soutien et d'avoir sorti le potentiel qui était en moi.

A mes mentors Ibou Sarr, Max Sarr, Souléye Sarr, Cheikh Sarr, vos conseils et vos encouragements m'ont été une grande aide.

Maman Sokhna Gueye et toute la famille Gueye de Saint-Louis qui m'ont fait sentir chez moi par leur amour, leurs soutiens et leurs encouragements.

A mes amis d'enfance (Abou Sylla, Mohamed Djim Kébé, Modou Ndiaye, Mamadou Ndiaye, Djibi Camara, Baba Thiam, Cheikh Thiam, Malado Diallo Malick Touré et Cheikh Diouf).

La famille Gueye en commençant par Marième Gueye, Awa Gueye, Awa Ndiaye, fadia Mare, Papa Yade, Fatou Thiam, Serigne Fallou Yade, Tapha Dia, Tapha, Modou Gueye, Ablaye Yade et Aladji Diop. Vous avez rendu mon passage à Saint-Louis exceptionnel. Un grand merci à vous !!!

Mes amis et frères de chambre 121P (Mouhamed Diallo, Mamadou Ly, Yaya Sy, Mame Baye Diouf, Assane Sao, Dame Top, Pape Demba Dieng, Cheikhouna Niang et Cheikh Ahmadou Bamba Khadim Dioum, Baillo Baldé et à Ibrahima Decarte Ndao) avec qui j'ai partagé des souvenirs merveilleux.

Tous les membres de l'association Dialégne Sa Kanam (DSK) en commençant par le président Ass Malick Camara et le secrétaire général Massar Cissé.

Tous mes camarades de classe MIETEL 2 (2022 /2023) plus particulièrement mon amie, ma sœur Ndéye Séne avec qui j'ai partagé énormément de choses.

Résumé

Cette mémoire traite de l'automatisation du processus de durcissement des serveurs web pour renforcer leur sécurité face aux cyberattaques. Le durcissement des systèmes aussi appelé renforcement des systèmes consiste à réduire les vulnérabilités en supprimant les éléments inutiles et en appliquant des configurations de sécurité adaptées. La première partie du document présente les principes et les avantages du renforcement des systèmes, ainsi que ses formes, notamment pour les serveurs, les bases de données et les systèmes d'exploitation. Des normes de sécurité, telles que celles de l'OWASP et de l'ANSSI, sont utilisées pour guider les bonnes pratiques de sécurisation. La deuxième partie détaille les étapes manuelles de sécurisation des serveurs web, incluant les audits de sécurité, la gestion des correctifs et la configuration des pares-feux. Des mesures comme le chiffrement des communications et la gestion des identités et des accès sont également mises en œuvre pour protéger les données sensibles. La troisième partie explore l'automatisation de ces tâches à l'aide d'outils comme Ansible, permettant une gestion centralisée et standardisée des configurations. Des exemples pratiques montrent comment automatiser l'installation de services, la création d'utilisateurs et la configuration de pare-feu pour un serveur web basé sur Apache sous Ubuntu. L'automatisation réduit les erreurs humaines, améliore la résilience du système et simplifie le processus de sécurisation. Ce travail démontre que l'automatisation est une solution efficace pour renforcer la sécurité des serveurs web tout en améliorant leurs performances et leur fiabilité.

Mots clés

Automatisation ; durcissement ; résilience

REPUBLIQUE DU SENEGAL.....
Remerciement i
Dédicaces ii
Liste des figures..... vi
Sigles et Abréviations viii
Chapitre 2 : Le durcissement des systèmes..... 3
Introduction 3
I Principe de durcissement des systèmes..... 3
I.1 L'importance de renforcement des systèmes..... 3
II Les différentes formes de renforcement des systèmes..... 5
II.1 Le renforcement des serveurs..... 5
II.2 Le renforcement des réseaux 6
II.3 Le durcissement des applications..... 6
II.4 Le durcissement de la base de données..... 7
II.5 Le renforcement des systèmes d'exploitation..... 7
Conclusion..... 8
Chapitre 3 : Etude de durcissement des serveurs web 9
Introduction 9
I. Principe de durcissement des serveurs web 9
I.1 **Définition et objectifs** 9
I.2. **Normes et standards utilisés**..... 10
II. Etude de durcissement des serveurs web 10
II.1 Préparation et audit initial 10
II.2. Mise à jour et gestion des correctifs..... 11
II.3 **Configuration des pares-feux et des réseaux** 12
II.4 **Gestion des identités et des accès**..... 13
II.5 **Sécurisation des services et des applications** 13
II.6 **Chiffrement des communications**..... 14
II.7 **Configuration des logs et de la surveillance** 15
II.8 **Renforcement des systèmes d'exploitation** 16
II.9 **Test, validation et maintenance continu**..... 17
Conclusion..... 17
Chapitre 4 : Automatisation d'un processus de durcissement..... 18

Introduction 18
I. Renforcement des serveurs méthode manuelle 18
I.1. **Les différents outils utilisés pour le renforcement manuel** 18
I.2 **Durcissement du serveur web apache sous Ubuntu** 19
II. Automatisation du processus de durcissement 38
II.1. **Outil d'automatisation** 38
II.2 **Automatisation du serveur web avec Ansible** 39
Conclusion 49
Conclusion Général 50
Références 51

Liste des figures

Figure 1: Architecture du réseau LAN 19
Figure 2: Compte privilégié 20
Figure 3: installation du serveur openssh 20
Figure 4: configuration de ssh 21
Figure 5: Génération de clé ssh 21
Figure 6: Copie de la clé publique dans le serveur 21
Figure 7: Connexion ssh au serveur 22
Figure 8: Liste des services disponible 22
Figure 9: Activation du pare-feu 23
Figure 10: Configuration de ufw 23
Figure 11: Statut de la table ufw 23
Figure 12: Installation de fail2band 24
Figure 13: Activavation de fail2ban 24
Figure 14: Attribution de délai à bantime findtime et maxretry 24
Figure 15 : les adresses non ignorer par fail2ban 25
Figure 16: Configuration du log de ssh dans fail2ban 25
Figure 17: Configuration des jails pour http 25
Figure 18: Vérification de l'estatut de fail2ban 25
Figure 19: Installation d'apache 26
Figure 20: Liste des profils d'application pour apache 26
Figure 21: Statut de ufw 27
Figure 22: Vérification de status de apache 27
Figure 23: répertoire pour le nom de domaine 27
Figure 24: Accueil d'apache 28
Figure 25:La page Apache avant 28
Figure 26: Masquage de la version d'apache et du système 29
Figure 27: La page après configuration 29
Figure 28: les répertoires dans apache 29
Figure 29: Masquage du répertoire 30
Figure 30: Masquage des répertoires 30
Figure 31: Activation du module apache 31
Figure 32: Activation du site apache 31
Figure 33: Installation de Openssl 31
Figure 34: Génération de certificat et sa clé 32
Figure 35: Autorisation de lecture sur le certificat et la clé 32
Figure 36:Amélioration de SSL 32
Figure 37:Activation du module ssl et du site et redémarrage d'apache 33
Figure 38: Connexion par https 33
Figure 39: Redirection de http vers https 34
Figure 40:Activation du module d'en-tête et redémarrage 34
Figure 41: Activation de HSTS pour qpache 34
Figure 42: Analyse du réseau avec Nmap 35
Figure 43: DDoS avec hping3 36
Figure 44:Copy de la clé publique 36
Figure 45: Utilisation de l'outil hydra 37
Figure 46: Vérification des logs de ufw 37
Figure 47: Vérification des configurations avec auth.log 38

Figure 48: Installation d'ansible 39
Figure 49: Inventaire 39
Figure 50:Automatisation de l'instalation et configuration de ssh 40
Figure 51: Résultats de l'automatisation de la configuration ssh 41
Figure 52: Statut du pare-feu ufw après automatisation 41
Figure 53: Ajout des utilisateurs 42
Figure 54: Résultat de l'automatisation des users 42
Figure 55: Les différentes users 42
Figure 56: Automatisation du téléchargement et configuration d'apache 44
Figure 57: Résultat de l'automatisation de la configuration d'apache 45
Figure 58: Vérification du statut d'apache 45
Figure 59: Vérification de la connexion avec apache 46
Figure 60: Automatisation de la configuration de fail2ban 47
Figure 61: Vérification du statut de fail2ban après automatisation 47
Figure 62: Résultat de l'automatisation de fail2ban 47
Figure 63: Regroupement des différents playbook 48
Figure 64: Vérification des services et ports ouverts 48
Figure 65: Utilisation de hping3 pour scanner les ports 48
Figure 66: Utilisation de hping3 pour s'informer des nœuds et les routes 49

Sigles et Abréviations

VPN: Virtual Protocol Network

DMZ : Zone démilitarisée

IDS : Système de détection des intrusions

IPS : Système de Prévention des intrusions

SGBD : Système de Gestion de Base de Données

RBAC : Contrôle d'Accès Basé sur les Rôles

OWASP: Open Application Web Security Project

CIS: Center for Internet Security

ANSSI : Agence nationale de la sécurité des systèmes d'information

API: Interface de programmation d'application

SQL*:* Structured Query Language

URL: Localisateur uniforme de ressource

IAM: Identity and Access Management

NIST: National Institute of Standards and Technology

SSL: Secure Sockets Layer

TLS: Transport Layer Security

SSH: Secure Sockets Shell

http: Hypertext Transfer Protocol

https: Hypertext Transfer Protocol Secure

hsts: Strict Transport Security

EAS: Entreprise Automation Stack

Introduction Générale

Le système d'information représente l'élément central sur lequel se concentre l'ensemble des opérations d'une entreprise ou d'une organisation contribuant directement à sa sécurité et à sa productivité. Il réunit toutes les données sensibles de l'organisation assurant ainsi le fonctionnement des applications, des serveurs, des logiciels, des bases de données, des systèmes d'exploitation… mais aussi fournit des outils nécessaires à son bon fonctionnement. Par conséquent, garantir la sécurité optimale du système d'information est d'une importance capitale et stratégique. En tenant compte que tous les systèmes présentent des faiblesses ou des vulnérabilités et que ces dernières, dans le système, pourraient rapidement conduire à des dysfonctionnements significatifs. C'est en ce sens qu'il est crucial d'appliquer des techniques de sécurité plus améliorés et plus avancés à savoir le durcissement des systèmes informatiques.

Le durcissement est un ensemble de règles et de processus de sécurité mis en place pour diminuer ou même éliminer les failles de sécurité qui pourraient occasionner la violation du système par une tierce personne. Il consiste à installer et à configurer l'ensemble des règles, des logiciels et des services dont le système a besoin pour son fonctionnement et à désactiver ou à supprimer tous ceux qui sont inutiles ou qui peuvent constituer un point d'entrée du système. Le renforcement des systèmes est une pratique de sécurité informatique qui a pour but de réduire les risques d'attaques potentielles en appliquant des configurations de sécurité adaptées aux systèmes et réseaux informatiques qui permet de réduire de manière significative les vulnérabilités.

Lutter contre la pénétration du système informatique est un défi qui est connu par tous les professionnels de sécurité informatique. Cependant pour se sentir plus en sécurité dans le réseau informatique, il est important d'étudier le renforcement des systèmes ou durcissement qui assure la sécurité, à l'intérieur qu'à l'extérieur, de l'infrastructure informatique. Notre travail portant sur l'automatisation d'un processus de renforcement de serveurs web est structuré en trois chapitres :

D'abord, dans le premier chapitre, nous étudierons le durcissement des systèmes dans le cadre général en évoquant de manière brève les différentes applications de renforcement de sécurité. Ensuite, le second chapitre nous permettra d'une part, de discuter du renforcement des serveurs web et résumera d'autre part les différentes étapes à respecter pour la sécurité desdits serveurs web.

Et enfin dans le dernier chapitre, il sera question d'étudier le processus d'automatisation du renforcement des serveurs web apache dans le système d'exploitation Ubuntu et d'expliquer les différentes étapes de renforcement des serveurs web apache dans le système d'exploitation ubuntu.

Ainsi à la fin de ce travail, nous serons en mesure de constater si notre système parvient à résister ou non aux menaces extérieures.

Chapitre 2 : Le durcissement des systèmes

Introduction

À l'heure actuelle où le numérique est en pleine expansion, les problèmes de cybersécurité affectent les organisations publiques et privées. Les violations de la sécurité cybernétique exploitent les failles et les points faibles des systèmes informatiques pour provoquer des violations, des fuites de données sensibles ou même rendre les systèmes inutilisables. L'un des principaux objectifs des attaquants est le serveur web, car il est accessible via l'internet, ce qui facilite l'accès et la recherche de faille pour les attaquants. Dès qu'une faille ou un point faible est présent dans le système, il peut être exploité pour mener des attaques. Le facteur clé contribuant à l'existence des failles est le manque de configuration de sécurité adéquate dans les systèmes, ce qui constitue l'une des dix principales vulnérabilités répertoriées par l'OWASP en 2021[1][2]. Afin de prévenir d'éventuelles attaques susceptibles de causer des dommages tels que la paralysie du système, les pertes de données sensibles, le chantage, la détérioration de la réputation ou des pertes financières [3], souvent la technique utilisée pour faire face à un éventuel dommage est d'appliquer le durcissement des systèmes dans le réseau informatique. Pour parvenir à étudier le durcissement, plusieurs étapes seront traitées à savoir : le principe de renforcement des systèmes et les différents types de renforcement de système.

I Principe de durcissement des systèmes

Les moyens de cybersécurité tels que les VPN, les DMZ, les antivirus, les IDS/IPS sont des solutions de sécurité périmétrique [3]. Le principe de durcissement des systèmes est de permettre à l'administrateur de renforcer la sécurité de son infrastructure informatique à l'intérieur comme à l'extérieur de l'entreprise ou du domaine à sécuriser. Cela implique d'appliquer des correctifs ou une configuration de sécurité beaucoup plus avancée pour maintenir la stabilité de l'infrastructure informatique.

I.1 L'importance de renforcement des systèmes

Le renforcement du système vise à réduire considérablement les risques de cyberattaques contre les organisations. Les failles de sécurité sont souvent causées par des changements de configuration qui entraînent des écarts par rapport aux normes de conformité. Chaque fois qu'un serveur ou un utilisateur est ajouté ou qu'une nouvelle application est installée, cela peut créer des points faibles sur la sécurité [4].

Avec les données stockées dans les serveurs de l'entreprise et aussi l'énorme moyen que les pirates ont à leur disposition, il est évident que l'on doit mettre en place des méthodes efficaces et performantes pour la mise en place d'une barrière de sécurité évolutive et préventive pour se protéger des mal intentionnés c'est-à-dire les pirates. Pour cela, nous adopterons le renforcement des systèmes ou encore le durcissement qui permet de minimiser la surface d'attaque dans l'environnement informatique.

Le renforcement des systèmes est un processus complexe par lequel tous les actifs de l'entreprise sont examinés pour avoir un aperçu sur la sécurité. Des méthodes avancées sont utilisées pour améliorer ou renforcer le système informatique.

I.2. **Avantages de renforcement des systèmes**

Le but principal de renforcer votre système, c'est de rendre votre système informatique plus résiliant et lui donner la capacité de résister aux menaces. Cela implique de réduire les risques de perdre des données, éviter que quelqu'un y accède sans permission, ou que des programmes malveillants y soient installés. En évitant les attaques, vous évitez aussi les interruptions soudaines de service. En plus, ça peut vous aider à respecter plus facilement les règles internes et externes ou la politique de sécurité de l'organisation.

Bien que le renforcement du système présente de nombreux avantages en termes de sécurité et de conformité, les principaux avantages peuvent être résumés ainsi :

- Sécurité renforcée : Grâce à des audits continus et à des corrections basées sur des normes de sécurité reconnues, toutes les vulnérabilités sont détectées et corrigées. Cela réduit les risques de piratage de données, d'infections par des logiciels malveillants et d'accès non autorisés [4].
- Amélioration des performances : En éliminant les éléments inutiles comme les programmes, services, utilisateurs et ports non utilisés, vos ordinateurs fonctionneront plus rapidement.[5]
- Facilité d'audit : Les codes clairs et compatibles avec toutes les plateformes et systèmes d'exploitation rendent les audits de sécurité plus simples, plus rapides et plus transparents. [4]
- Personnalisation : Le durcissement vous permet d'ajuster les paramètres par défaut des logiciels selon vos besoins spécifiques.
- Meilleur contrôle : La procédure de durcissement inclut généralement un audit complet du système, ce qui vous offre un meilleur contrôle sur celui-ci.

- Confidentialité : En cryptant les données, le durcissement aide à prévenir les fuites d'informations et assure leur confidentialité. [5]

II Les différentes formes de renforcement des systèmes

Le renforcement des systèmes englobe non seulement la sécurité des réseaux, des applications des logiciels et des micrologiciels d'un ordinateur mais aussi de ces bases de données et d'autres éléments susceptibles de menacer la sécurité de l'entreprise et de faciliter une éventuelle attaque.

Etant de nature très diversifié, le durcissement des réseaux informatique pourrait être décomposé en cinq catégories à savoir :

- Le renforcement des serveurs
- Le renforcement des réseaux
- Le renforcement des systèmes d'exploitation
- Le renforcement des applications
- Le renforcement des bases de données

II.1 Le renforcement des serveurs

Le processus de renforcement du serveur implique la sécurisation des données, des ports, des composants, des fonctions et des autorisations d'un serveur. Ces protocoles sont exécutés à l'échelle du système sur les couches matérielles, micrologicielles et logicielles [3]. Il est important de noter que le durcissement des serveurs est une étape importante pour une bonne marche de sécurité puisqu'il permet de minimiser la surface d'attaque et offre à l'entreprise une protection contre les ransomwares, les logiciels malveillants et d'autres menaces existantes. En suivant cette logique, il est possible de minimiser tous les points d'accès des attaquants et de corriger les vulnérabilités en matière de cybersécurité.

Le renforcement des serveurs, étant très important en ce qu'il permet de bloquer les effets d'une éventuelle attaque et de sécuriser les données et les opérations liées à la sécurité du serveur, il faut tout de même noter que le durcissement des serveurs doit être un processus continu dans le temps puisque la technologie et les moyens utilisés pour attaquer les serveurs évoluent tous les jours. Le processus de renforcement nécessite plusieurs étapes dont :

- Trouver un emplacement sécurisé du serveur
- Contrôler les autorisations d'accès

- Configurer le pare-feu
- Gérer les configurations
- Sécuriser les comptes utilisateurs
- Appliquer des correctifs faces aux vulnérabilités
- Supprimer les logiciels inutiles
- Prévoir une stratégie de sauvegarde
- Contrôler en permanence

II.2 Le renforcement des réseaux

Vu l'évolution constante de la cybersécurité dans le monde et de l'augmentation des menaces de la part des acteurs malveillants voulant exploiter les vulnérabilités présentes dans le réseau, le renforcement des réseaux informatiques prend tout son sens en ce que son objectif est de est de renforcer la sécurité de l'infrastructure du réseau en minimisant les points de vulnérabilité qui peuvent profiter à l'attaquant. Le renforcement du système doit alors être une pratique que chaque organisation ou entreprise devrait envisager avec rigueur pour améliorer leurs défenses et corriger les faiblesses potentielles.

Pour un réseau bien renforcé, il sera primordial de mettre en place soit un système de prévention des intrusions IPS soit un système de détection des intrusions. L'application de l'IPS ou de l'IDS permet de surveiller les activités suspectes sur le réseau, les signale et aide les administrateurs à empêcher les accès non autorisés. Les techniques de renforcement du réseau incluent la configuration et la sécurisation adéquate des pare-feu, l'audit des règles et des privilèges d'accès, la désactivation de ports inutilisés, le cryptage du trafic et la désactivation des services non utilisés. En combinant ces techniques avec un IPS ou un IDS, la surface d'attaque du réseau est réduite, augmentant ainsi sa résistance aux attaques.

II.3 Le durcissement des applications

La protection des applications logicielles qui s'exécutent sur le système repose sur plusieurs mesures, notamment :

- La suppression ou la désactivation des caractéristiques et fonctionnalités inutiles.
- L'application de correctifs et de mises à jour de sécurité spécifiques aux applications.
- L'adoption de pratiques de codage sécurisé et de contrôles d'accès.
- La mise en place de mécanismes d'authentification au niveau de l'application.

La sécurité des applications est cruciale car il permet aux utilisateurs de l'organisation ou de l'entreprise de bénéficier d'un environnement sûr et stable. Pour le personnel, l'application de correctifs et de mises à jour leur permet de se protéger contre les menaces connues et de mettre en œuvre des mesures préventives. Il est important de se rappeler que les utilisateurs sont souvent les points d'entrée de la cybercriminalité dans l'organisation. Parmi les techniques les plus courantes pour assurer la sécurité des applications, on peut citer :

- L'installation d'applications uniquement à partir de référentiels fiables.
- L'automatisation des correctifs pour les applications standards et tierces.
- L'installation de pare-feu, d'antivirus et de programmes de protection contre les logiciels malveillants ou espions.
- Le chiffrement des données basé sur un logiciel.
- L'utilisation d'applications de gestion et de chiffrement des mots de passe.

II.4 Le durcissement de la base de données

Le renforcement des bases de données consiste à sécuriser à la fois le contenu d'une base de données numériques et le système de gestion de base de données (SGBD), qui est l'application permettant aux utilisateurs de stocker et d'analyser les informations dans une base de données.

Ce processus implique principalement trois étapes :

- Contrôler et limiter les privilèges et l'accès des utilisateurs.
- Désactiver les services et fonctions de base de données inutiles.
- Sécuriser ou chiffrer les informations et les ressources de la base de données.

Les techniques courantes de renforcement des bases de données incluent de :

- Restreindre les privilèges et les fonctions des administrateurs.
- Chiffrer les informations de la base de données en transit et au repos.
- Adopter une politique de contrôle d'accès basée sur les rôles (RBAC).
- Effectuer des mises à jour et des correctifs réguliers du logiciel de base de données ou du SGBD.
- Désactiver les services et fonctions de base de données inutiles.
- Verrouiller les comptes de base de données en cas d'activité de connexion suspecte.
- Utiliser des mots de passe de base de données forts et complexes.

II.5 Le renforcement des systèmes d'exploitation

Le renforcement du système d'exploitation implique l'application de correctifs et la mise en œuvre de mesures de sécurité avancées pour sécuriser le système d'exploitation (OS) d'un serveur. L'un des meilleurs moyens de renforcer le système d'exploitation consiste à installer automatiquement les mises à jour, correctifs et service packs.

Le renforcement du système d'exploitation est similaire au renforcement des applications dans la mesure où le système d'exploitation est techniquement une forme de logiciel. Cependant, contrairement au renforcement des applications, qui se concentre sur la sécurisation des applications standard et tierces, le renforcement du système d'exploitation sécurise le logiciel de base permettant à ces applications de fonctionner sans aucune restriction.

Souvent, les développeurs de systèmes d'exploitation tels que Microsoft et Linux publient régulièrement des mises à jour et rappellent aux utilisateurs de les installer. Ces mises à jour fréquents, parfois ignorés, peuvent en réalité aider à maintenir votre système sécurisé et résistant aux cyberattaques.

D’autres exemples de renforcement du système d’exploitation incluent :

- Suppression des pilotes inutiles.
- Chiffrement du disque dur qui stocke et héberge votre système d'exploitation.
- Activation et configuration du démarrage sécurisé.
- Limitation et authentification des autorisations d'accès au système.
- Limitation ou suppression de la création et de la connexion de comptes d'utilisateurs.

Conclusion

Le renforcement des systèmes informatiques est essentiel pour assurer la sécurité et la résilience de l'infrastructure informatique d'une organisation. En appliquant des techniques de durcissement, on réduit significativement les risques de cyberattaques et on renforce la stabilité et la performance des systèmes. Ce processus, bien qu’exigeant, offre une protection accrue contre les menaces internes et externes, garantissant ainsi un environnement sécurisé pour les opérations de l’organisation. La mise en œuvre continue et rigoureuse de ces mesures est cruciale pour maintenir une défense efficace contre les cyberattaques en constante évolution.

Chapitre 3 : Etude de durcissement des serveurs web

Introduction

Le web est un système hypertexte public fonctionnant sur Internet. Il permet de consulter des pages accessibles sur des sites à l'aide d'un navigateur. Avec l'avancée du numérique et de l'évolution des services web, les failles de sécurités ne manqueront pas à se présenter puisque chaque avancée technologique en informatique présente ces vulnérabilités. Ces vulnérabilités des services web seront exploitées par les cybercriminels pour attaquer les serveurs. Compte tenu de l'évolution de ces services, il devient nécessaire de se soucier de la sécurité du web et de ces serveurs dans sa globalité.

Il parait évident que les serveurs web sont des cibles de choix pour les cybercriminels. Face à l'évolution incessante des menaces, renforcer la sécurité des serveurs web est une mesure cruciale pour protéger les systèmes et les données. Ceci nous pousse à nous interroger sur le renforcement des serveurs web dans un environnement informatique. Ce chapitre offre une analyse approfondie du renforcement des serveurs web, en explorant ses principes de base, les étapes clés du processus, ainsi que des exemples pratiques d'implémentation.

I. Principe de durcissement des serveurs web

Les serveurs web jouent un rôle fondamental dans le fonctionnement des applications web. Cependant il faut noter que la plupart des administrateurs les installent avec leurs configurations par défaut. Ce qui peut entrainer la présence de beaucoup de failles de sécurité et des fuites de données sensibles concernant le serveur. Les applications web sont accessibles sur internet, ce qui les rendent très vulnérables et accessibles par les cyberattaques. Pour minimiser ces risques, il est essentiel de mettre en place le durcissement des serveurs web pour lutter contre les vulnérabilités présentes dans les applications web.

I.1 Définition et objectifs

Le durcissement des serveurs est un procédé technique dont l'objectif est de garantir la protection et la sécurité [7] d'un serveur de manière à réduire sa vulnérabilité face aux différentes attaques. Ces principaux objectifs sont de réduire la surface d'attaque, de protéger les données sensibles et d'améliorer la résilience face aux menaces existantes et des menaces antérieures.

I.2. Normes et standards utilisés

Les normes et standards sont l'ensemble des principes et procédures mondialement reconnus par les professionnels de la sécurité et qui donnent un ensemble de méthodes à adopter pour une bonne pratique de sécurité. Ces différentes normes sont éditées par des organisations internationales et ces normes sont mises à jour régulièrement en fonction de l'évolution des menaces. C'est en ce sens que certaines organisations telles que OWASP répertorie, chaque année, les différentes menaces les plus connues pour la sécurité informatique.

Il faut noter que les normes et recommandations des organisations telle que OWASP, Cis et les bonnes pratiques de ANSSI sont tous à la base d'une bonne démarche de durcissement. L'ensemble des procédures de sécurité recommandées par ces organisations sont à la base de l'endurcissement des serveurs. Ces normes fournissent des lignes de commandes précises pour sécuriser les configurations des serveurs et des applications.

II. Etude de durcissement des serveurs web

II.1 Préparation et audit initial

La préparation et l'audit des serveurs web est une étape importante dans le processus de sécurisation des serveurs web, elle permet non seulement d'évaluer la robustesse du serveur mais aussi de mesurer la sécurité du serveur. Un audit de sécurité web est un processus méthodique et structuré qui se décompose en plusieurs étapes essentielles pour assurer une analyse complète et approfondie de la sécurité d'un site web [8]. Les différentes parties de ce processus peuvent être divisées en quatre parties à savoir : l'identification, l'évaluation, l'exploitation, et le rapport

- **Identification**

L'identification consiste à chercher et à sensibiliser sur les biens et les actifs de l'entreprise dans les serveurs. Ces biens sont les données personnelles, les noms de domaines, les sous-domaines, les applications web et les interfaces API [8]. La connaissance de ces différents actifs permet de savoir par où appliquer les correctifs et mieux sécuriser l'environnement du serveur.

- **L'évaluation**

Dans cette partie, l'administrateur du réseau peut avoir l'aperçu sur les actifs identifiés et pourrait éventuellement se projeter sur les menaces à venir pour ensuite détecter les vulnérabilités de son système. Cela permet de détecter les failles connues, la faiblesse des

configurations du serveur, l'analyse des codes sources, et l'évaluation des mesures de sécurité en place [8].

- Exploitation

L'exploitation consiste à identifier les vulnérabilités les plus courantes pour les serveurs web ou même les applications web. On ne peut pas parler des vulnérabilités sans parler de l'OWASP qui répertorie chaque année les 10 vulnérabilités les plus courantes et ces vulnérabilités sont [9] :

- *Injection SQL*
- Cross Site Scripting est également connu sous le nom de XSS
- Authentification brisée et gestion de session
- Références d'objets directs non sécurisés
- Contrefaçon de demande intersites
- Mauvaise configuration de la sécurité
- Stockage cryptographique non sécurisé
- Echec de la restriction de l'accès aux URL
- Protection insuffisante de la couche transport
- Redirections et transferts non validés

- Rapport

Cette partie consiste à tester et à rédiger les résultats de l'audit dans un rapport bien détaillé pour constater l'ensemble des vulnérabilités existant c'est-à-dire leurs impacts et leurs gravités dans toute l'infrastructure. C'est dans ce rapport qu'on va prescrire les recommandations par rapport à la sécurité des serveurs web et celle de l'entreprise. A travers des consignes claires et précises, le rapport permet de fournir une feuille de route pour remédier ou éradiquer les failles de sécurité et améliorer le renforcement du serveur web.

II.2. Mise à jour et gestion des correctifs

La gestion des correctifs est l'ensemble des méthodes, des techniques mis en place lors de l'application des mises à jour pour appliquer les correctifs de vulnérabilités et optimiser les performances du serveur web et les applications web. Elle est une pratique essentielle dans les environnements informatiques vu son efficacité à éviter les attaques et à corriger les failles de sécurité. La gestion des correctifs crée un processus centralisé pour appliquer de nouveaux

correctifs aux actifs informatiques. Ces correctifs peuvent améliorer la sécurité, les performances et augmenter la productivité [10].

II.3 **Configuration des pares-feux et des réseaux**

Un pare-feu pourrait être considéré comme un logiciel ou un matériel permettant de respecter ou de faire respecter la politique de sécurité du réseau. Sa configuration nous permet d'éditer ou de contrôler toutes les règles et paramètres à partir desquelles un pare-feu est configuré. Un pare-feu bien configuré et bien sécurisé permet à son tour de sécuriser les trafics réseaux entrant et sortant en tenant compte d'un ensemble de règles de sécurité. Il faut noter qu'un pare-feu pourrait être une solution informatique qui protège le réseau contre les trafics indésirables.

La mise en place d'un pare-feu informatique est un processus délicat qui nécessite une bonne connaissance et une compréhension du réseau. La configuration du pare-feu dans le réseau peut être divisée en quatre parties [11]:

- Définir une politique de sécurité claire
- Configurer les règles d'accès entrantes et sortantes
- Mettre en place une authentification forte
- Tester la configuration dans un environnement contrôlé

En respectant ces différentes étapes et en adoptant une bonne pratique de sécurité, votre réseau serait en mesure de se protéger contre les menaces mais aussi garderait son efficacité pour les menaces. Pour une bonne pratique de sécurité, on peut procéder comme suit [11]:

- Ne pas ouvrir plus de ports que nécessaire
- Documenter et réviser régulièrement les règles de pare-feu
- Utiliser des listes de contrôle d'accès (ACL) pour un contrôle minutieux
- Implémenter une segmentation réseau pour limiter la propagation d'attaques

En appliquant toutes ces pratiques et recommandations, votre système sera en mesure de résister aux menaces et de limiter l'évolution des attaques dans le réseau.

II.4 **Gestion des identités et des accès**

La gestion des identités et des accès ou IAM (Identity and Access Management) est l'ensemble des moyens utilisés pour gérer les identités et les accès des utilisateurs. Les experts en cybersécurité peuvent définir ou modifier de manière unique les rôles de chaque utilisateur du réseau en précisant de manière claire qui peut accéder aux ressources de l'entreprise ou qui ne peut pas en accéder.

La gestion des identités et des accès améliore considérablement la sécurité en fournissant des mécanismes d'authentification robustes, des contrôles d'accès et des mesures de protection des données [12]. Une bonne gestion des accès et des identités peut faire éviter l'entreprise l'accès aux données indésirables et participer à la protection et à la sensibilisation par rapport à la protection des données sensibles de l'organisation. Il faut préciser que les données sensibles à la violation doivent être protégées par des clés d'authentification robustes mais aussi limiter l'accès aux données grâce au niveau d'accès ou des hiérarchies de l'entreprise.

La gestion des identités et des accès facilite le principe de moindre privilège, en garantissant que les utilisateurs disposent uniquement des autorisations d'accès nécessaires en fonction de leurs rôles et responsabilités, minimisant ainsi le risque des menaces internes [12]. Par ailleurs, il faut ajouter que l'IAM permet de renforcer la sécurité des données et aussi rendre efficace le système grâce à une gestion automatisée des ressources et des identités.

II.5 **Sécurisation des services et des applications**

La sécurité des applications et des services est une pratique indispensable pour se protéger et éviter les menaces dans les réseaux. Elle est un ensemble de mesures spécifiquement conçues pour lutter contre les vols de données, manipulation des données ou encore empêcher la manipulation des codes sources des applications. La sécurisation des applications permet aux entreprises de se protéger contre les attaques néfastes mais aussi de contrôler les requêtes inattendues c'est-à dire venant d'extérieur. Les professionnels de la sécurité ont tendance à suivre des procédures de sécurité conçu par des organismes internationaux (NIST). L'une des principales normes de sécurisation des applications et services conçu par NIST est une approche de sécurité qui s'exécute de manière procédurale. Il s'agit :

- **Authentification** : confirmation de la validité de l'identité d'un utilisateur ; nécessaire pour l'application de l'accès basé sur l'identité.[13]

- **Chiffrement** : conversion d'informations ou de données en code visant à empêcher tout accès non autorisé ; et peut concerner des fichiers individuels ou un projet tout entier. [13]
- **Journalisation** : analyse de l'activité des utilisateurs en vue de vérifier les incidents liés à une activité suspecte ou une compromission.[13]
- **Contrôles de validité** : contrôles visant à s'assurer que les données saisies et traitées répondent à des critères précis.[13]
- **Contrôles d'accès** : restriction de l'accès aux applications en fonction de l'adresse IP ou d'autres autorisations octroyées à certains utilisateurs.[13]

Il faut admettre que la sécurité des applications et des services est un défi très connu dans l'écosystème des sécurités informatiques. C'est en ce sens que les organisations ont la responsabilité de surveiller leur sécurité.

II.6 Chiffrement des communications

Le chiffrement est l'ensemble des méthodes et des techniques mises en place pour coder les données afin qu'elles restent cachées ou inaccessibles à une autre que son destinataire. Ces techniques permettent de protéger les informations personnelles sensibles et de renforcer la sécurité des communications entre les applications web et les serveurs web. Il existe deux principaux types de chiffrement de données qui permettent de rendre un message lisible uniquement par son destinataire : le chiffrement symétrique à clé partagée, le chiffrement asymétrique avec une paire clé publique et une autre clé privée.

- Le chiffrement symétrique : il utilise les mêmes types de clé pour l'envoi et l'écoute ou encore pour chiffrer et déchiffrer les communications. Mais cette méthode nécessite une transmission sécurisée de la clé, sans cela le message pourrait être déchiffré par une autre personne que son destinataire.
- Le chiffrement asymétrique : il utilise deux clés complétement différentes. Ces deux clés dont l'une est publique c'est-à-dire partagée avec son expéditeur et l'autre privée qui est protégée.

Le serveur web étant celui qui héberge les sites web, son chiffrement passe par des étapes essentielles pour protéger les données des utilisateurs et maintenir leur confidentialité. Ces étapes essentielles sont :

- Installer un certificat de sécurité SSL ou TLS
- Durcir la sécurité du serveur web
- Mise à jour de manière permanente le site web
- Vérifier aussi que le site web est bien sécurisé

Sécuriser la communication dans le serveur web est une méthode pour garantir la sécurité et la confidentialité dans l'entreprise, c'est pour cela qu'il est suivi de très près et régulièrement mise à jour.

II.7 **Configuration des logs et de la surveillance**

Dans l'environnement informatique, presque toutes les composantes informatiques génèrent des logs. Un log est un fichier généré par les équipements informatiques qui capte l'activité du système d'exploitation, des serveurs, des applications …. Il regroupe toutes les informations spécifiées par les administrateurs systèmes telles que : les messages, les rapports d'erreurs, les demandes de fichiers, les transferts de fichiers … Il faut noter que chaque composant du réseau enregistre au log des informations par rapport à leur utilisation que cela soit autorisée ou non. C'est ce qui permet au développeur de pouvoir connaitre l'habitude des utilisateurs pour mieux améliorer leur expérience dans le réseau.

Pour une bonne pratique de sécurité, il faut penser à la gestion des logs qui consiste à collecter, stocker, traiter, synthétiser et analyser de manière régulière ou continue les données issues des programmes, des applications et des serveurs qui présentent des défauts de conformité. Permettant ainsi d'augmenter efficacement les performances des systèmes, d'identifier les problèmes techniques, d'endurcir la sécurité, de gérer les ressources et enfin d'améliorer la conformité. La gestion des logs est composée de plusieurs parties :

- **Collecte** : les outils de gestion des logs regroupent les données du système d'exploitation, des applications, des serveurs, des utilisateurs, des Endpoints ou de toute autre source pertinente au sein de l'entreprise. [14]
- **Surveillance** : les outils de surveillance des logs retracent les événements et les activités, en indiquant le moment où ils se sont produits. [14]
- **Analyse** : les outils d'analyse des logs examinent les logs collectés sur le serveur afin d'identifier proactivement les bugs, les menaces de sécurité ou tout autre problème. [14]

- **Rétention** : cet outil indique combien de temps les données doivent être conservées dans le log. [14]
- **Indexation ou recherche** : cet outil de gestion des logs aide les équipes informatiques à filtrer, trier, analyser ou rechercher des données dans l'ensemble des logs. [14]

- **Production de rapports** : ces outils avancés automatisent la création de rapports à partir du log d'audit, qui décrit à son tour les performances opérationnelles, l'affectation des ressources, la sécurité ou la conformité réglementaire. [14]

La procédure de gestion des logs est une solution efficace et avantageuse à l'entreprise puisqu'elle est conçue pour faciliter la gestion des données et répondre plus efficacement aux besoins.

II.8 **Renforcement des systèmes d'exploitation**

Le renforcement du système d'exploitation est le processus de configuration d'un système d'exploitation pour réduire sa surface d'attaque et sa vulnérabilité aux menaces. Cela implique l'application de correctifs, la désactivation de services et de fonctionnalités inutiles, l'application d'autorisations et de politiques strictes et la mise en œuvre de contrôles et d'outils de sécurité [15]. Cela consiste en quelque sorte à faire tout son possible pour réduire les points de vulnérabilités du système connu pour ne donner aucune chance à un attaquant.

Pour réduire les niveaux de vulnérabilités, de menaces et de vecteurs d'attaques, il est nécessaire de diminuer ou même de supprimer les logiciels inutilisés ou qui n'ont aucun intérêt pour votre système, les comptes utilisateurs, les privilèges de certains utilisateurs désactivés ou de supprimer les services inutiles et les périphériques à risque. Différentes méthodes sont utilisées pour le renforcement du système d'exploitation parmi lesquelles on peut citer :

- Une application continue des correctifs et des mises à jour de sécurité
- Suppression ou désactivation de tous les ports et services utiles
- Désinstallation des packages et logiciels inutiles.
- Définir les rôles et les droits d'accès des utilisateurs et les administrateurs du réseau

II.9 **Test, validation et maintenance continu**

Dans le processus de renforcement des serveurs, plusieurs étapes interviennent et chaque étape est importante pour la réussite de ce processus. Pour assurer la pertinence ou la réussite du renforcement des serveurs, il est nécessaire de tester l'ensemble des méthodes utilisées avant de pouvoir la validé. Les tests et validations étant très cruciales dans les contrôles de sécurité réseau et l'efficacité du renforcement, les tests visent à évaluer et à documenter sur les performances des équipements, des systèmes et des processus. Les résultats de ces tests doivent être vérifiés avant la validation.

Dans un monde où les réseaux informatiques connaissent une évolution constante, une maintenance et une mise à jour continue est nécessaire pour le bon fonctionnement des serveurs et des systèmes d'exploitation. La surveillance du réseau et du sécurité de l'infrastructure doit être une tâche à laquelle les administrateurs doivent veillés.

Conclusion

Le renforcement des serveurs se repose sur des méthodes et pratiques de sécurité approuvées par des organisations telles que l'OWASP (Open Web Application Security Project) ou ANSSI (Agence Nationale de la Sécurité des Système Numérique) qui fournissent de manière régulière des techniques pour se protéger contre les menaces.

Face à ces menaces, les professionnels de sécurité utilisent des approches qui leur permettent d'assurer la sécurité des serveurs, des applications et des systèmes mais aussi de créer un environnement de confiance pour un monde numérique et des menaces en parfaite évolution.

Chapitre 4 : Automatisation d'un processus de durcissement

Introduction

Pour respecter la confidentialité des données, le renforcement des serveurs est une solution utilisée par les entreprises du secteur informatique pour diminuer les points de vulnérabilité du système. Le renforcement des serveurs fait référence au processus d'amélioration de la sécurité des serveurs en réduisant les points les plus vulnérables. Ces procédures entrainent un ensemble de modifications et de configurations qui vise à renforcer les systèmes de défense du serveur contre les attaques et les intrusions non autorisées**.**

Le renforcement des serveurs suit des règles et méthodes adoptées par des organisations telles que l'OWASP et ANSSI. L'OWASP qui est l'acronyme de : Open Web Application Security Project est un projet de sécurité des applications web qui publie chaque année les vulnérabilités les plus connues pour les applications et serveur web. L'ANSSI c'est-à-dire l'Agence Nationale de la Sécurité des Système Numérique est l'autorité en matière de cybersécurité et a comme mission de comprendre, de prévenir et de répondre aux risques d'attaque.

Dans cette partie, on va énumérer les différents outils utilisés et mettre en pratique le renforcement des serveurs de manière manuelle et enfin automatiser l'ensemble du processus de renforcement des serveurs.

I. Renforcement des serveurs méthode manuelle

I.1. Les différents outils utilisés pour le renforcement manuel

Pour le renforcement des serveurs, on a utilisé des machines virtuelles Ubuntu qui est un système d'exploitation GNU/Linux basé sur les distributions Debian. Ubuntu est un système d'exploitation open source simple et utilisé en ligne de commande. Dans la pratique, c'est ce même système qui est utilisé comme serveur web. Une machine Kali est utilisée pour tester la robustesse des configurations du serveur.

Les systèmes d'exploitation présentent souvent des vulnérabilités qui sont des points faibles que les cybercriminels peuvent exploités. Il existe de nombreuses vulnérabilités dans le noyau de Linux d'Ubuntu qui affectent Ubuntu 18.04 LTS, Ubuntu 20.04 LTS, Ubuntu 22.04 LTS et Ubuntu 22.10 mais aussi les serveurs web avec des risques tel que :

- Atteinte à la confidentialité des données [16]
- Déni de service à distance [16]
- Exécution de code arbitraire à distance [16]

Les cybercriminels peuvent exploités ces vulnérabilités pour attaquer le système ou le serveur, pour accéder aux données sensibles ou même perturber le réseau. C'est dans ce sillage que le fait de sécuriser votre serveur acquiert toute son importance.

I.2 Durcissement du serveur web apache sous Ubuntu

- Le renforcement du système d'exploitation Ubuntu
  - Architecture du réseau

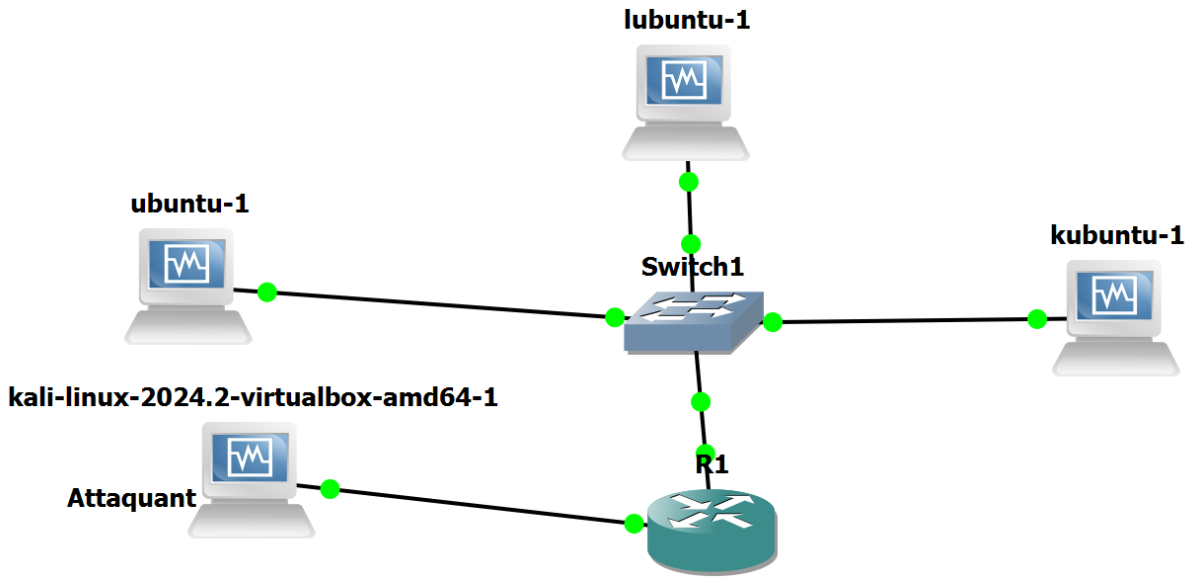


Figure 1: Architecture du réseau LAN

  - La mise à jour régulière du système

La mise à jour est une étape importante pour le renforcement du système, elle est indispensable pour la protection du système et les serveurs. Dans notre OS Ubuntu 22.04, la mise à jour se fait en implémentant :

$ sudo apt update

$ sudo apt upgrate

  - Créez des comptes privilégiés pour éviter les erreurs

La création de compte privilégié permet à l'administrateur de contrôler l'accès à root c'est-à-dire à la mode super utilisateur. Cela permet d'éviter les erreurs à la mauvaise configuration.

On a supprimé l'utilisateur root pour n'utiliser que sudo pour gérer la sécurité de telle sorte qu'une fois qu'il ait une erreur de configuration, on pourrait savoir avec précision qui a été à l'origine des erreurs de configurations et à quel moment.

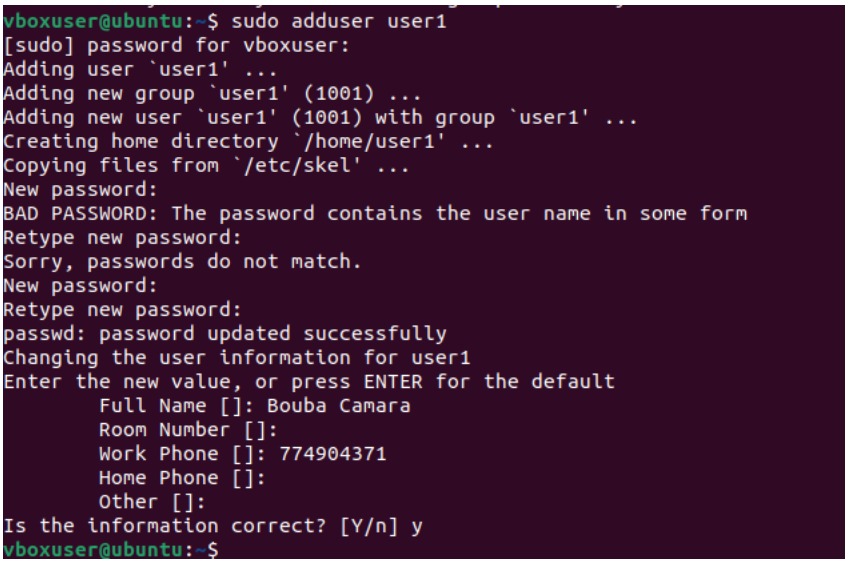


Figure 2: Compte privilégié

On a ajouté deux utilisateurs (user2 et user3) avec user2 qui a les mêmes privilèges que user1 et le même mot de passe mais User3 a des privilèges limités.

- Configuration de SSH pour accéder au serveur en toute sécurité sans mot de passe

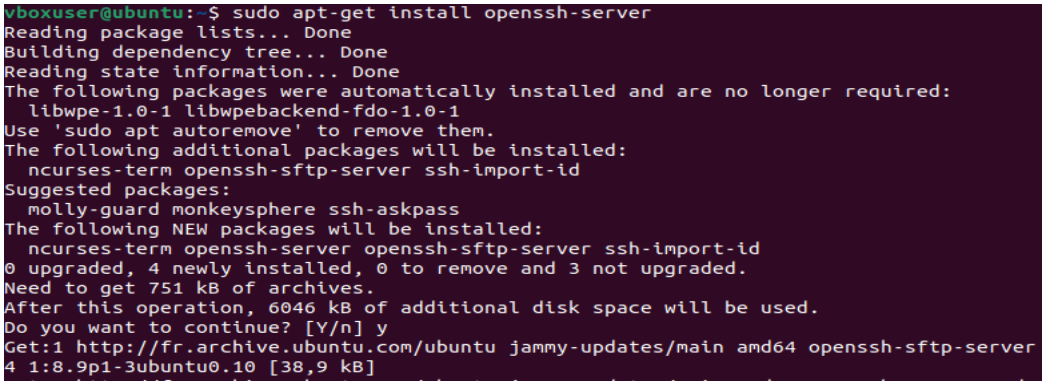


Figure 3: installation du serveur openssh

Pour la configuration de sécurité SSH l'installation de openssh est nécessaire.

Pour plus de sécurité dans notre système, on évite l'accès root à distance pour le serveur et configurer la connexion ssh en le configurant dans le répertoire /etc/ssh/sshd_config.

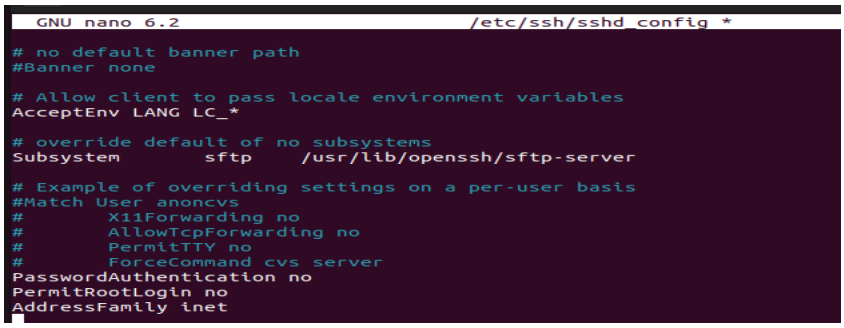


Figure 4: configuration de ssh

Pour plus de sécurité avec la connexion ssh à distance, on procède à la création des deux clés publiques et privés dans les deux machines hôtes (kubuntu et lubuntu) qui sont des clés de sécurité très robustes et difficile à attaquer par force brute.

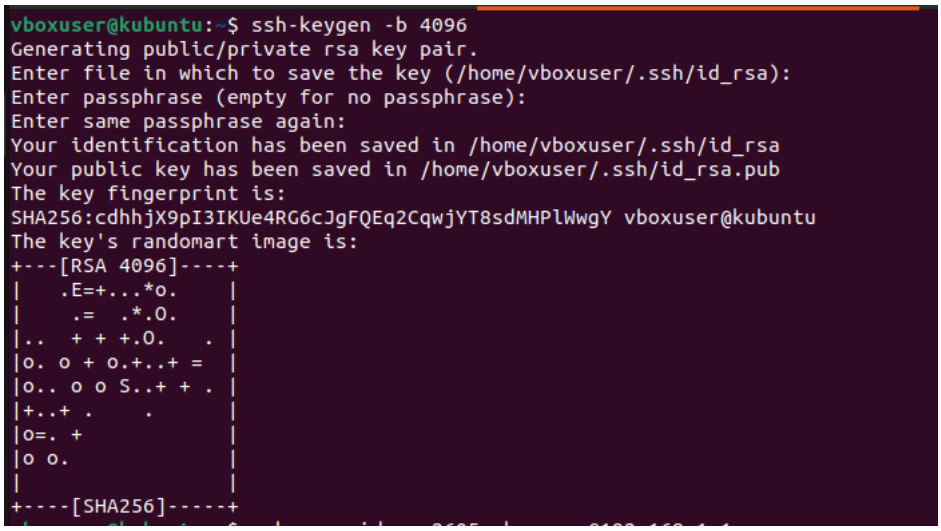


Figure 5: Génération de clé ssh

La clé publique créée dans les deux machines est copiée dans le serveur pour qu'une fois la connexion établie par ssh au serveur, on compare les deux clés pour juger si éventuellement la connexion doit être acceptée ou pas.

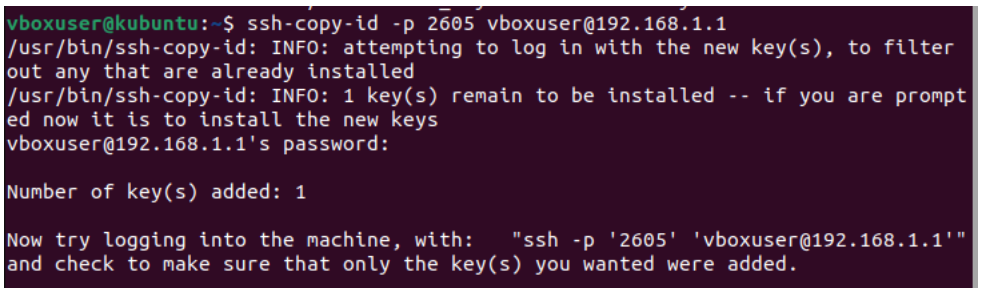


Figure 6: Copie de la clé publique dans le serveur

On constate que la connexion ssh au serveur a réussi. Donc on parvient à se connecter depuis kubuntu ou lubuntu dans notre serveur ubuntu.

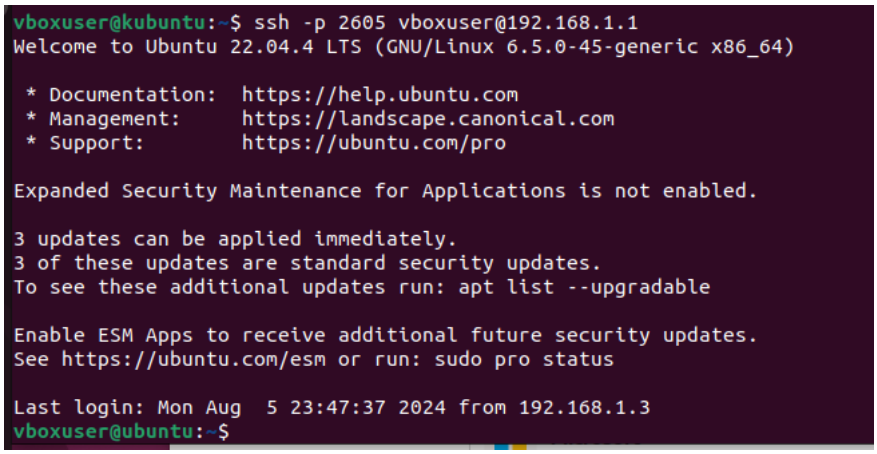


Figure 7: Connexion ssh au serveur

Pour plus de sécurité, le port par défaut de ssh est passé de 22 à 2605 pour éviter les attaques qui ciblent directement le port 22.

On redémarre les services de ssh en appliquant : $ sudo sshd service restart

- Le renforcement du serveur
  - Maintenez votre serveur Linux à jour

Toujours maintenir le système à jour pour diminuer les points de vulnérabilités.

$ sudo apt update

$ sudo apt upgrate

  - Réduisez les services inutiles pour plus de sécurité

On utilise la commande sudo ss -atpu pour lister les services disponibles

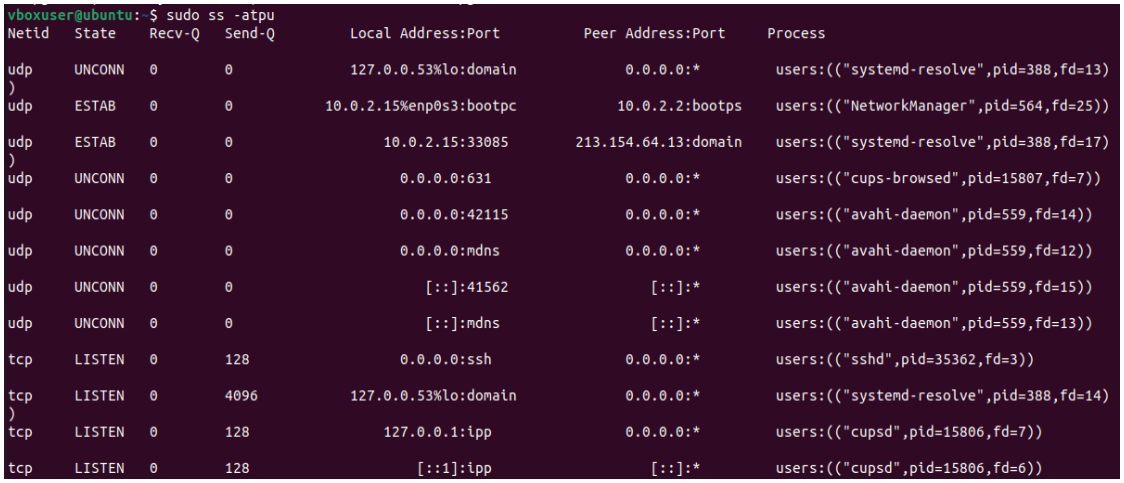


Figure 8: Liste des services disponible

UNCONN : Aucune connexion active

ESTAB : Connexion établie.

LISTEN : Le port est en mode écoute, prêt à accepter les connexions

Pour supprimer les services inutiles dans le serveur, on utilise la commande :

$ sudo apt purge nom_serveur

- Configurer un pare-feu fiable

Le pare-feu ufw est installé par défaut dans certains systèmes d'exploitation ubuntu mais si il n'est pas installé, on utilise la commande :

$ sudo apt install ufw

```
vboxuser@ubuntu:~$ sudo ufw enable
Firewall is active and enabled on system startup
vboxuser@ubuntu:~$
```

Figure 9: Activation du pare-feu

```
vboxuser@ubuntu:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
vboxuser@ubuntu:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
vboxuser@ubuntu:~$ sudo ufw allow ssh
Rule added
Rule added (v6)
vboxuser@ubuntu:~$ sudo ufw allow http
Rule added
Rule added (v6)
vboxuser@ubuntu:~$ sudo ufw allow https
Rule added
Rule added (v6)
```

Figure 10: Configuration de ufw

```
vboxuser@ubuntu:/$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
80/tcp                     ALLOW       Anywhere
443                        ALLOW       Anywhere
2605/tcp                   ALLOW       Anywhere
80/tcp (v6)                ALLOW       Anywhere (v6)
443 (v6)                   ALLOW       Anywhere (v6)
2605/tcp (v6)              ALLOW       Anywhere (v6)
```

Figure 11: Statut de la table ufw

- Utilisons Fail2ban pour plus de sécurité en configurant les journaux et log

```
vboxuser@ubuntu:~$ sudo apt install fail2ban
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  cups-browsed cups-core-drivers cups-daemon cups-server-common hplip-data libcupsimage2
  libhpmud0 libimagequant0 libraqm0 libsane-hpaio libwpe-1.0-1 libwpebackend-fdo-1.0-1
  printer-driver-postscript-hp python3-olefile python3-pil python3-renderpm
  python3-reportlab python3-reportlab-accel ssl-cert
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  python3-pyinotify whois
Suggested packages:
  mailx monit sqlite3 python-pyinotify-doc
The following NEW packages will be installed:
  fail2ban python3-pyinotify whois
0 upgraded, 3 newly installed, 0 to remove and 3 not upgraded.
Need to get 473 kB of archives.
After this operation, 2486 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://fr.archive.ubuntu.com/ubuntu jammy/universe amd64 fail2ban all 0.11.2-6 [394 kB]
Get:2 http://fr.archive.ubuntu.com/ubuntu jammy/main amd64 python3-pyinotify all 0.9.6-1.3 [24,8 kB]
Get:3 http://fr.archive.ubuntu.com/ubuntu jammy/main amd64 whois amd64 5.5.13 [53,4 kB]
Fetched 473 kB in 9s (54,1 kB/s)
Selecting previously unselected package fail2ban.
(Reading database ... 210932 files and directories currently installed.)
```

Figure 12: Installation de fail2band

Activation des services de fail2ban avec la commande : sudo systemctl enable fail2ban

```
vboxuser@ubuntu:~$ sudo systemctl enable fail2ban
[sudo] password for vboxuser:
Synchronizing state of fail2ban.service with SysV service script with /lib/systemd/systemd-
sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable fail2ban
Created symlink /etc/systemd/system/multi-user.target.wants/fail2ban.service → /lib/systemd
/system/fail2ban.service.
```

Figure 13: Activavation de fail2ban

Configuration de fail2ban : On crée un fichier de configuration à partir du fichier général en copiant le fichier /etc/fail2ban/jail.conf vers /etc/fail2ban/jail.local puis l'éditer avec vim pour ajouter les délais de contrôle.

```
vboxuser@ubuntu: ~
#
# ignorecommand = /path/to/command <ip>
ignorecommand =

# "bantime" is the number of seconds that a host is banned.
bantime  = 60m

# A host is banned if it has generated "maxretry" during the last "findtime"
# seconds.
findtime  = 15m

# "maxretry" is the number of failures before a host get banned.
maxretry = 7
```

Figure 14: Attribution de délai à bantime findtime et maxretry

La figure 14 nous montre que pendant 15 minutes on a que 7 tentatives possible au-delà de 7, l'hôte sera banni pendant 60 minutes.

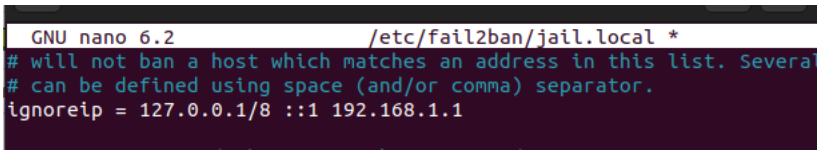


Figure 15 : les adresses non ignorer par fail2ban

La figure 14 est importante, elle permet d'éviter que fail2ban bannisse la machine locale qui est le serveur.



Figure 16: Configuration du log de ssh dans fail2ban

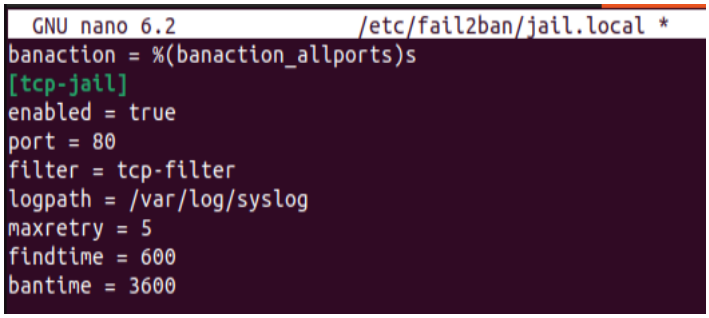


Figure 17: Configuration des jails pour http

Puis on redémarre les services de fail2ban avec : Sudo systemctl reload fail2ban.

La vérification de l'état de de fail2ban nous montre que fail2ban est active et cette vérification se fait avec la commande : sudo systemctl status fail2ban

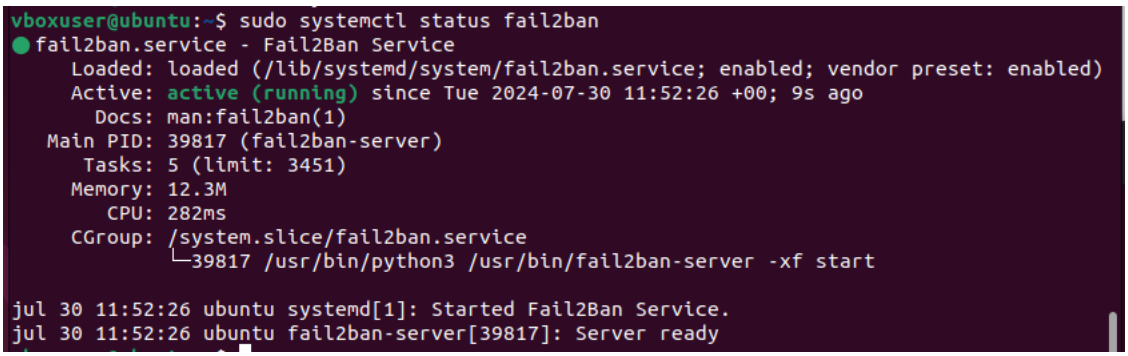


Figure 18: Vérification de l'estatut de fail2ban

- Renforcement de Apache
  - Installation et configuration du serveur apache

```
vboxuser@ubuntu:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  cups-browsed cups-core-drivers cups-daemon cups-server-common hplip-data libcupsimage2
  libhpmud0 libimagequant0 libraqm0 libsane-hpaio libwpe-1.0-1 libwpebackend-fdo-1.0-1
  printer-driver-postscript-hp python3-olefile python3-pil python3-renderpm
  python3-reportlab python3-reportlab-accel
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
0 upgraded, 8 newly installed, 0 to remove and 3 not upgraded.
Need to get 1922 kB of archives.
After this operation, 7724 kB of additional disk space will be used.
```

Figure 19: Installation d'apache

On affiche les profiles disponibles des applications dans le pare-feu avec la commande : $ sudo ufw app list

```
vboxuser@ubuntu:~$ sudo ufw app list
[sudo] password for vboxuser:
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
  OpenSSH
```

Figure 20: Liste des profils d'application pour apache

- **Apache** : ouvre le port TCP 80 pour HTTP (dans ce cas la connexion est non chiffrée)
- **Apache Full** : ouvre le port TCP 80 (HTTP, la connexion est non chiffrée) et 443 (HTTPS, la connexion est chiffrée avec TLS/SSL)
- **Apache Secure** : ouvre seulement le port HTTPS 443 pour une connexion chiffrée et sécurisé

Comme aucune application n'a encore été mis en place, nous ne pouvons activer apache full pour ouvrir les ports 80 et 443 dans apache et l'autoriser dans le pare-feu ufw avec la commande : $ sudo ufw allow apache Full

En vérifiant status de ufw on verra si l'apache Full est ajouté ou pas.

```
vboxuser@ubuntu:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
80/tcp                     ALLOW       Anywhere
443                        ALLOW       Anywhere
2605/tcp                   ALLOW       Anywhere
Apache Full                ALLOW       Anywhere
80/tcp (v6)                ALLOW       Anywhere (v6)
443 (v6)                   ALLOW       Anywhere (v6)
2605/tcp (v6)              ALLOW       Anywhere (v6)
Apache Full (v6)           ALLOW       Anywhere (v6)
```

Figure 21: Statut de ufw

Sur cette liste, on voit qu'apache est ajouté. Après, on peut vérifier l'état d'apache avec la commande : $ sudo systemctl status apache2

```
vboxuser@ubuntu:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2024-07-30 12:30:05 +00; 3h 17min ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 40890 (apache2)
      Tasks: 55 (limit: 3451)
     Memory: 4.9M
        CPU: 884ms
     CGroup: /system.slice/apache2.service
             ├─40890 /usr/sbin/apache2 -k start
             ├─40892 /usr/sbin/apache2 -k start
             └─40893 /usr/sbin/apache2 -k start

jul 30 12:30:05 ubuntu systemd[1]: Starting The Apache HTTP Server...
jul 30 12:30:05 ubuntu systemd[1]: Started The Apache HTTP Server.
```

Figure 22: Vérification de status de apache

Après vérification, on voit qu'apache est activé.

Création d'une structure de répertoire pour un domaine propre au sein de */var/www/*

```
vboxuser@ubuntu:~$ sudo systemctl restart apache2
vboxuser@ubuntu:~$ sudo mkdir /var/www/sn.com
vboxuser@ubuntu:~$
```

Figure 23: répertoire pour le nom de domaine

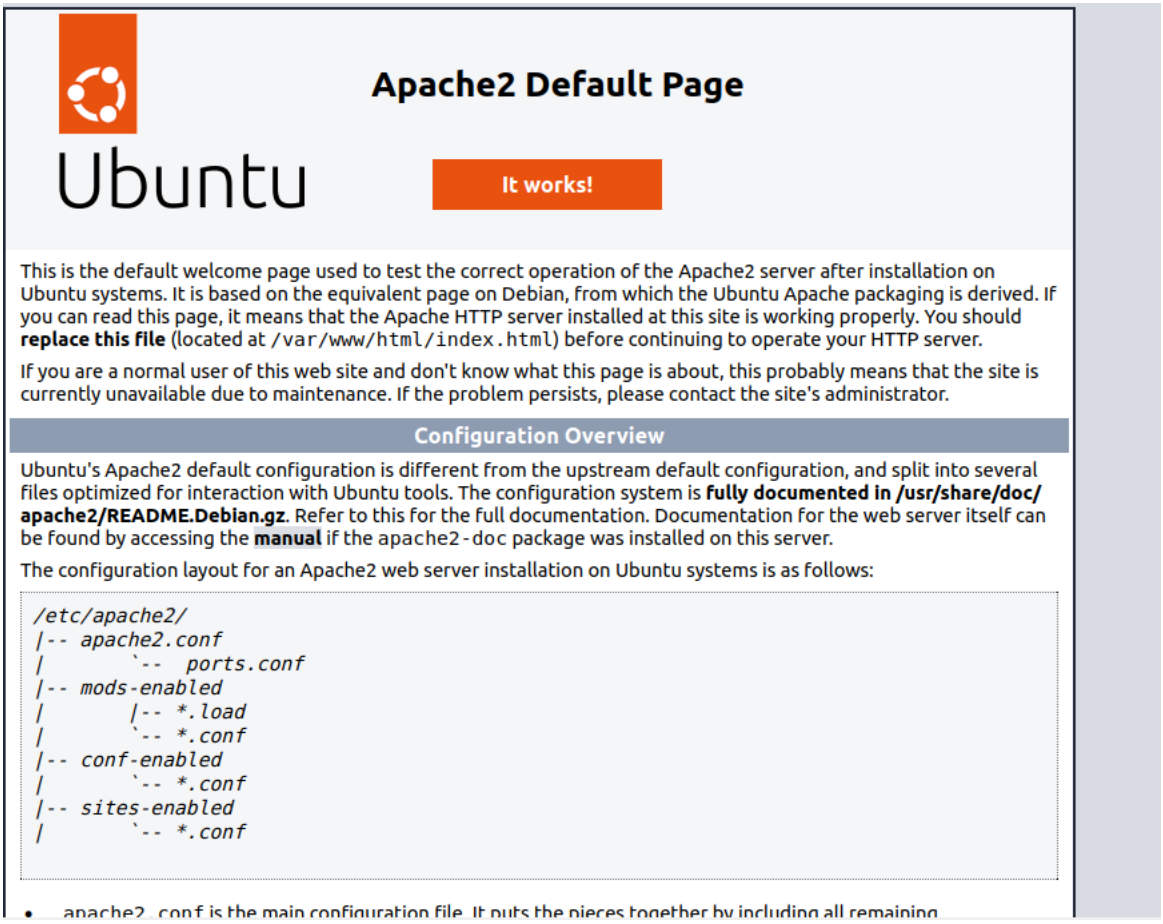


Figure 24: Accueil d'apache

- Masquer la version d'Apache et les informations sur le système d'exploitation



Figure 25:La page Apache avant

Sur cette page test d'apache, on voit que la connexion n'a pas réussi mais on peut avoir l'information sur la version d'apache, sur quel système elle a été installée et quel port a été utilisé. Ce qui peut être bénéfique pour une personne mal intentionnée. C'est pourquoi il est important de masquer ces informations qui pourrait être bénéfique au pirate.

```
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
ServerTokens Prod
ServerSignature Off
```

Figure 26: Masquage de la version d'apache et du système

Ces directives dans la figure 26 permettent de contrôler l'affichage de la version d'apache et du système sur lequel il a été installé. Après redémarrage du service apache, voici le résultat :

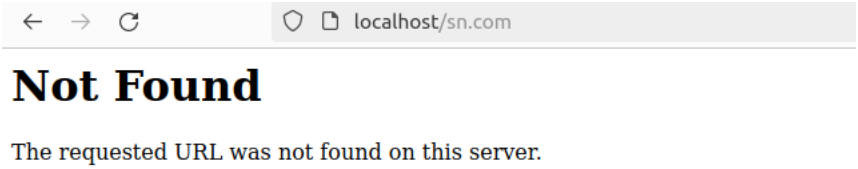


Figure 27: La page après configuration

- Désactiver la liste des répertoires dans apache

Les répertoires par défauts ont été créées dans le dossier /var/www/html/sn.com :

$ sudo mkdir -p /var/www/html/sn.com

Les répertoires créées sont : main.py et app.py :

$ sudo touch app.py main.py

Apache affiche la liste des répertoires par défaut que les visiteurs peuvent voir (voir figure) :



Figure 28: les répertoires dans apache

Ces répertoires peuvent être masquées ou désactivées de telle sorte que les visiteurs ne le verront plus.

Dans le fichier principal d'apache, on insère les directives suivantes pour masquer les répertoires et pour éviter que les visiteurs modifient le contenu des répertoires ou du serveur.

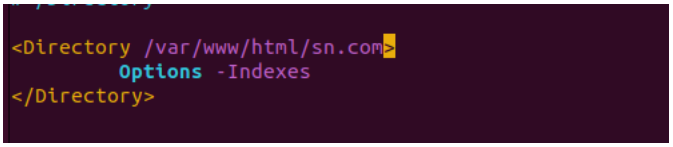


Figure 29: Masquage du répertoire

Après ceci, le navigateur vous affiche que vous n'avez pas de permission pour accéder à ces ressources.

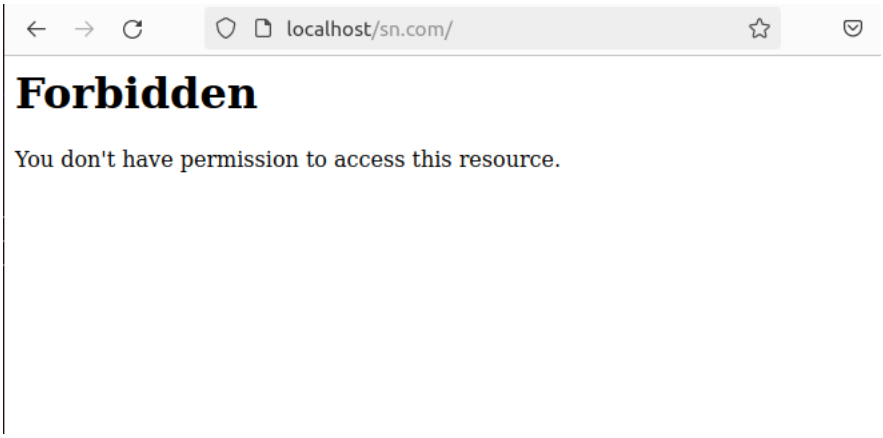


Figure 30: Masquage des répertoires

- Mise à jour régulièrement apache

La mise à jour est une étape importante dans le processus de renforcement des serveurs puisqu'elle permet au système d'avoir la dernière version des logiciels ou d'applications installées dans le système et de diminuer par la même occasion les points de vulnérabilités du système.

Ces mises à jour sont faites avec les commandes :

$ sudo apt update

$ sudo apt upgrade

- Utilisé le cryptage https sur apache

Activation du module ssl pour apache, ces modules nécessaires pour configurer le cryptage SSL/TLS

```
vboxuser@ubuntu:/$ sudo a2enmod ssl
[sudo] password for vboxuser:
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed
certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

Figure 31: Activation du module apache

```
vboxuser@ubuntu:/$ sudo a2ensite default-ssl
[sudo] password for vboxuser:
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
vboxuser@ubuntu:/$
```

Figure 32: Activation du site apache

On redémarre les services apache pour appliquer ces activations avec la commande :

$ Sudo service apache2 reload

La sécurisation des échanges entre les différentes postes clients et le serveur nécessite l'utilisation d'un certificat dans notre cas et on utilise l'application Openssl comme certificat.

```
vboxuser@ubuntu:/$ sudo apt-get install openssl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssl is already the newest version (3.0.2-0ubuntu1.16).
openssl set to manually installed.
The following packages were automatically installed and are no longer required:
  cups-browsed cups-core-drivers cups-daemon cups-server-common hplip-data libcupsimage2
  libhpmud0 libimagequant0 libraqm0 libsane-hpaio libwpe-1.0-1 libwpebackend-fdo-1.0-1
  printer-driver-postscript-hp python3-olefile python3-pil python3-renderpm
  python3-reportlab python3-reportlab-accel
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

Figure 33: Installation de Openssl

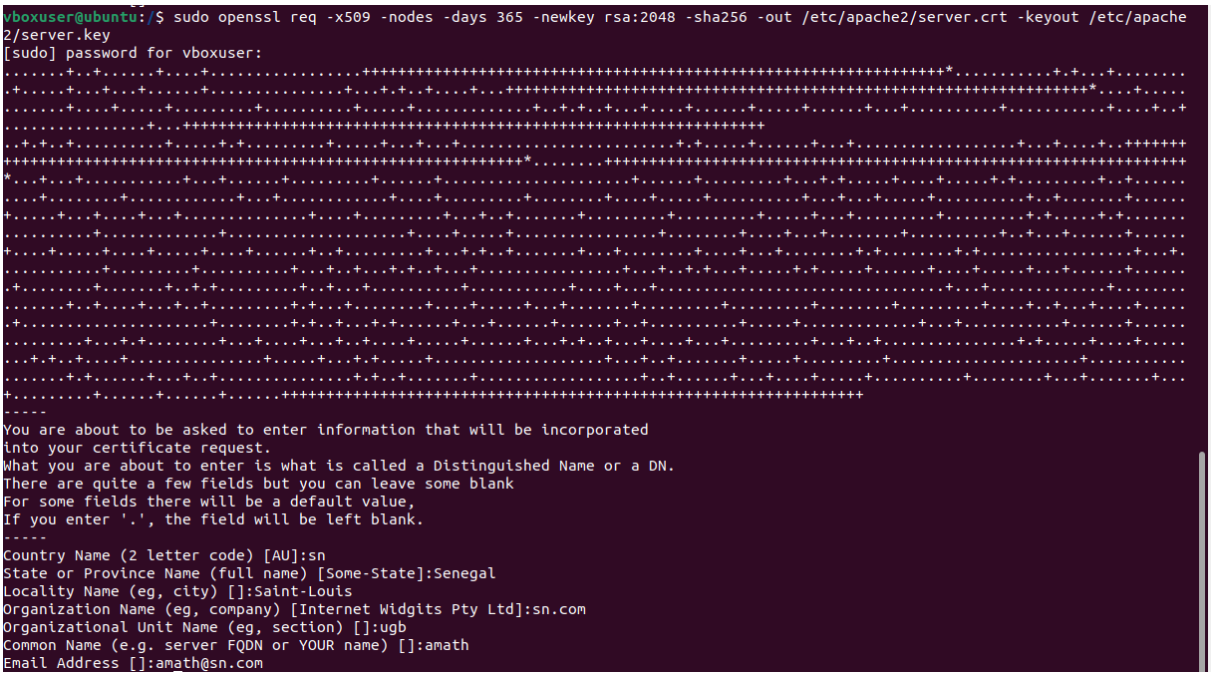


Figure 34: Génération de certificat et sa clé

Pour plus de sécurité, on modifie les permissions de lecture sur le certificat et la clé afin de ne pas autoriser la lecture par les utilisateurs mais uniquement les utilisateurs root.

```
vboxuser@ubuntu:/$ sudo chmod 440 /etc/apache2/server.crt
vboxuser@ubuntu:/$ sudo chmod 440 /etc/apache2/server.key
vboxuser@ubuntu:/$
```

Figure 35: Autorisation de lecture sur le certificat et la clé

En 2014, des vulnérabilités ont été observées dans la version 3 du protocole SSL permettant de sécuriser les connexions entre clients et serveurs. Elles permettent à un attaquant de provoquer une atteinte à la confidentialité des données, par exemple de récupérer un cookie de session HTTPS. Pour éviter ces genres de vulnérabilité, on écrit ces lignes de commande dans le fichier « **default-ssl** » contenant la configuration du site SSL.

```
SSLProtocol -ALL +TLSv1 +TLSv1.1 +TLSv1.2
SSLHonorCipherOrder On
SSLCipherSuite ECDHE-RSA-AES128-SHA256:AES128-GCM-SHA256:HIGH:!MD5:!aNULL:!EDH:!RC4
SSLCompression off
```

Figure 36:Amélioration de SSL

Après modification des configurations de ssl, on réactive le module et redémarre les services ssl.

```
vboxuser@ubuntu:/$ sudo a2enmod ssl
[sudo] password for vboxuser:
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
vboxuser@ubuntu:/$ a2ensite default-ssl
Site default-ssl already enabled
vboxuser@ubuntu:/$ sudo a2ensite default-ssl
Site default-ssl already enabled
vboxuser@ubuntu:/$ sudo service apache2 reload
```

Figure 37:Activation du module ssl et du site et redémarrage d'apache

Arriver à ce stade, la connexion par https passe par



Figure 38: Connexion par https

Pour plus de sécurité, il est nécessaire d'utiliser que du https. C'est en ce sens qu'on désactive le protocole http et redirige toutes les requêtes http vers https autrement dit les requêtes du port 80 vers le port 443 offrent plus de sécurité grâce au chiffrement.

Dans un premier temps, on désactive le http par la commande : $ sudo a2dissite 000-default.conf

On modifie le fichier /etc/apache2/sites-available/000-default.conf en y ajoutant la ligne suivante : Redirect permenent / https://sn.com/

```
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port t>
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn
        Redirect permanent / https://sn.com/
```

Figure 39: Redirection de http vers https

On active la redirection par la commande : $ sudo a2enmod rewrite

On termine par le redémarrage des services d'apache par la commande : $ sudo service apache2 reload

- Active HTTP Strict Transport Security (HSTS) pour Apache

```
vboxuser@ubuntu:~$ sudo a2enmod headers
[sudo] password for vboxuser:
Enabling module headers.
To activate the new configuration, you need to run:
  systemctl restart apache2
vboxuser@ubuntu:~$ systemctl restart apache2
```

Figure 40:Activation du module d'en-tête et redémarrage

On accède dans le répertoire par défaut par nano /etc/apache2/sites-available/default-ssl.conf

```
GNU nano 6.2                          /etc/apache2/sites-available/default-ssl.conf
#        useless for serving static content. So one usually enables the
#        exportation for CGI and SSI requests only.
#   o OptRenegotiate:
#        This enables optimized SSL connection renegotiation handling when SSL
#        directives are used in per-directory context.
#SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
<FilesMatch "\.(cgi|shtml|phtml|php)$">
                SSLOptions +StdEnvVars
</FilesMatch>
<Directory /usr/lib/cgi-bin>
                SSLOptions +StdEnvVars
</Directory>
 Header always set Strict-Transport-Security "max-age=31536000;includeSubDomains"
```

Figure 41: Activation de HSTS pour qpache

On redémarre apache pour appliquer la sécurité.

- Test d'intrusion dans le réseau
  - Utilisation de l'outil nmap

Nmap est un outil open source d'analyse, de scanne et d'audit réseau qui permet de voir les ports ouverts et les services réseaux qui leur sont alloués. C'est ce même outil que les

administrateurs réseaux ou les auditeurs de réseaux utilisent pour analyser le réseau et rechercher les ports ouverts et les machines vulnérables.

L'analyse avec nmap nous a permis d'avoir le résultat suivant :

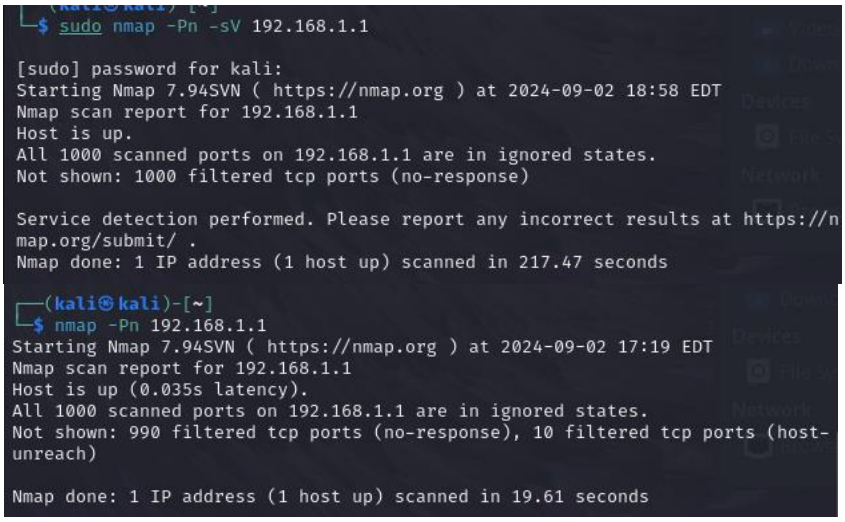


Figure 42: Analyse du réseau avec Nmap

La figure 41 nous montre que nmap a tenté d'analyse ou de scanner 1000 ports tcp mais 990 ports n'ont pas réponses ou ont était filtrés et 10 ports filtrés indiquent que l'hôte n'est pas joignable. Nous ne trouvons pas aussi les services dans les différents ports du serveur. Ces résultats montrent que le serveur a une configuration de sécurité qui bloque les tentations de connexion sur les différents ports.

- Utilisation de l'outil hping

Pour un administrateur système, comprendre les menaces qui pèsent sur le réseau devient un atout pour mieux se prémunir contre les attaques. C'est en ce sens qu'on utilise l'outil hping qui est un utilitaire simple à utiliser pour envoyer de manière régulière et répéter les paquets ICMP, UDP ou TCP puis afficher les réponses. Hping est aussi utilisé pour les attaques ddos et saturé le réseau.

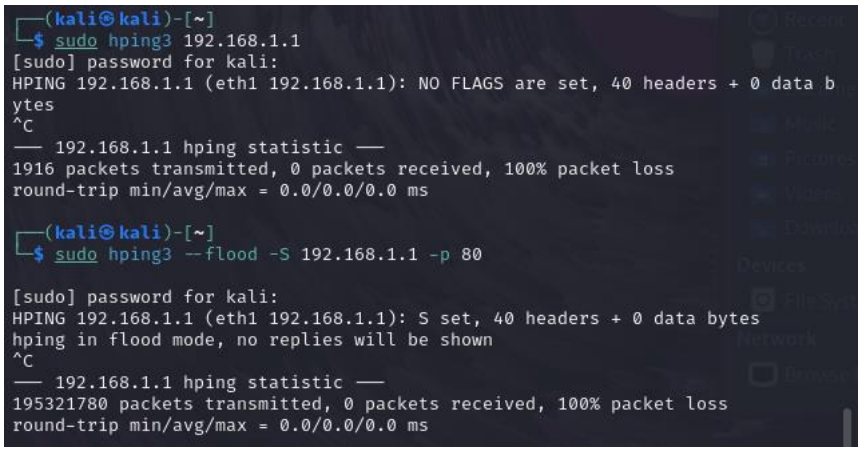


Figure 43: DDoS avec hping3

Les résultats montrent que bien qu'on a envoyé un grand nombre de requêtes au serveur 192.168.1.1. On constate qu'il n'y a pas de réponse envoyée et cela peut être dû à une configuration du pare-feu qui bloque une succession de requête envoyée où tous les ports du réseau ont été bloqués.

- Utilisation de l'outil hydra pour une attaque par force brute

Hydra est un outil de brute force puissante qui utilise une liste de nom d'utilisateur ou de mot de passe possible pour tenter de se connecter à la machine ou au réseau distant. Dans notre cas, on essaie de forcer la connexion ssh dans le serveur. Pour cela on copie la clé publique dans un dossier de Kali linux pour tenter d'avoir la clé privée.



Figure 44:Copy de la clé publique

Après avoir copier la clé publique, on essaie de récupérer les clés privées qui leur sont associées pour se connecter via ssh.

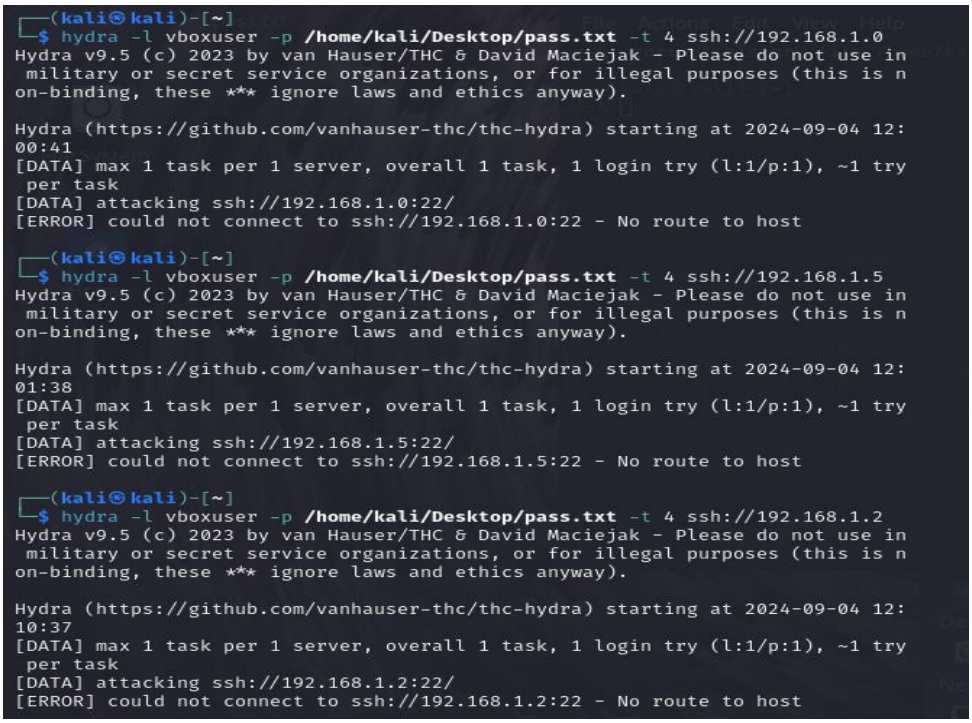


Figure 45: Utilisation de l'outil hydra

L'attaque de type ssh avec hydra utilise directement le port 22 par défaut de ssh pour se connecter. Pour forcer la connexion sur ce port, on constate que hydra a envoyé un message d'erreur indiquant (No route to host) qu'il ne peut pas se connecter sur le port 22 du ssh.

- Vérification des logs ufw et auth

Pour les logs ufw, on peut percevoir l'ensemble des connexions refusées ou les ports qui ont été filtrés. Avec la commande sudo cat /var/log/ufw.log, on peut voir l'ensemble des adresses IP et des ports filtrés dans le réseau.

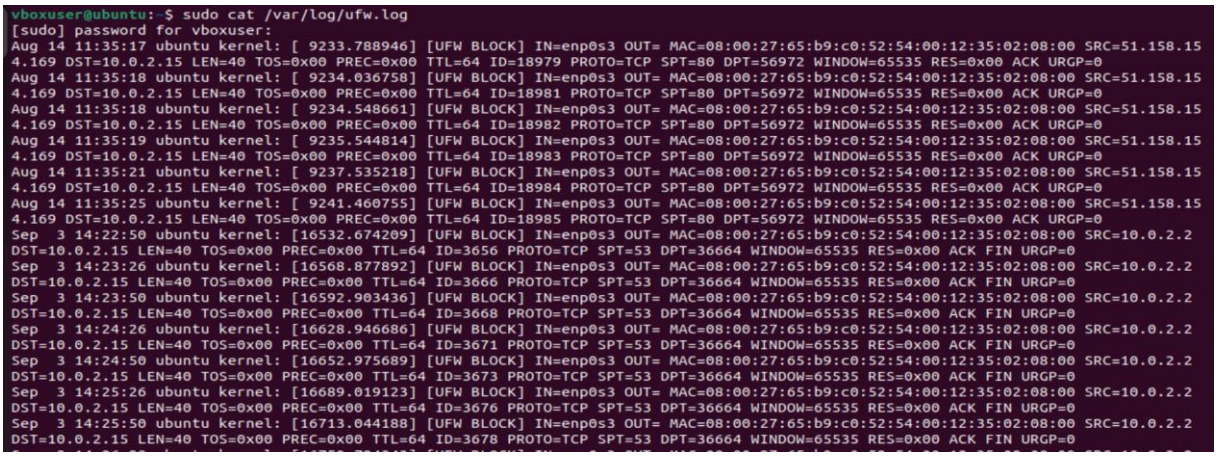


Figure 46: Vérification des logs de ufw

Pour les logs auth, on peut avoir les informations sur l'utilisation des différents utilisateurs mais aussi avoir une connaissance sur les lignes de commandes que chaque utilisateur a édité dans le système. Pour qu'une fois qu'il y aurait une erreur, on pourrait connaitre qui l'auteur de cette erreur avec la date et l'heure.

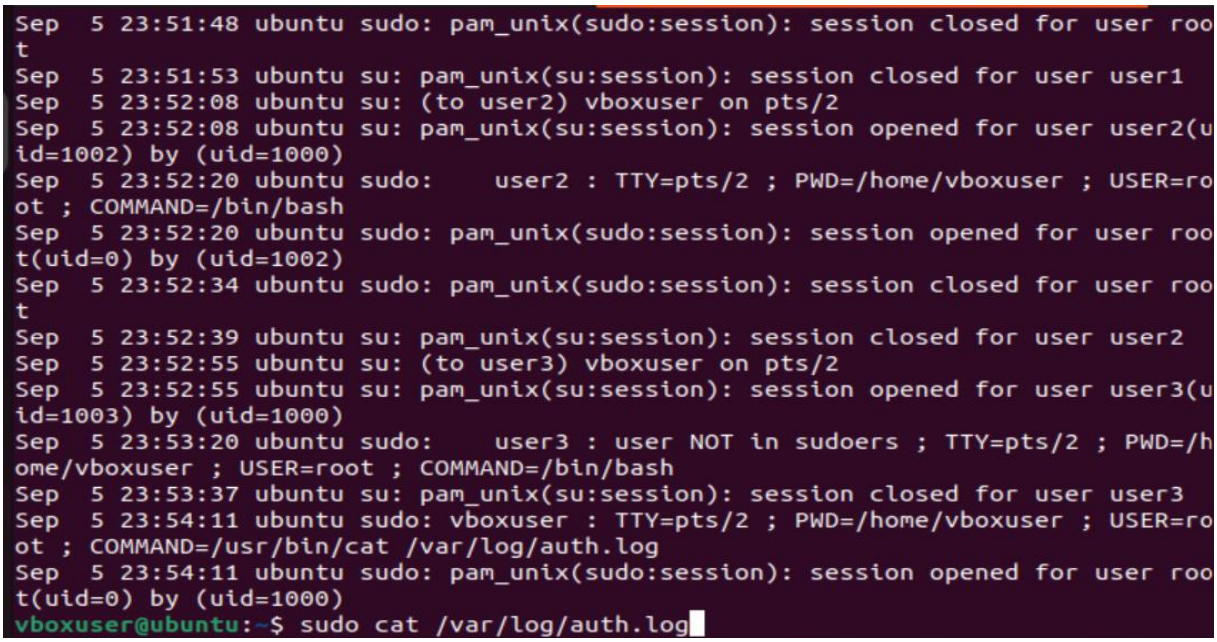


Figure 47: Vérification des configurations avec auth.log

II. Automatisation du processus de durcissement

II.1. Outil d'automatisation

L'automatisation des serveurs est une pratique très importante pour les serveurs puisqu'elle permet de gérer l'ensemble des processus de mise à jour du système et des différents logiciels impliqués dans la mise en place d'un serveur web sécurisé et durable. Les outils d'automatisation sont nombreux et variés. Parmi eux, on peut en citer : Chef (EAS) et Ansible.

- Chef (EAS)

Chef Enterprise Automation Stack (EAS) est une plate-forme d'automatisation permettant aux équipes DevSecOps de créer, de déployer, de gérer et de sécuriser n'importe quelle application exécutée sur n'importe quelle infrastructure [17]. Chef, en tant qu'outil d'automatisation, fonctionne avec un agent. Ce qui implique l'ajout d'un logiciel sur chaque machine et ce qui rend un peu son déploiement compliqué.

- Ansible

Ansible est un outil de configuration et de déploiement de logiciels open source. Il permet de définir et de gérer l'état d'un système informatique de manière simple et automatisée. Ansible utilise YAML un langage de configuration basé sur Python, facile à apprendre et conçu pour les administrateurs et ces commandes exécutées sur le serveur et la configuration des ordinateurs distants se fait via SSH.

II.2 Automatisation du serveur web avec Ansible

- Téléchargement de Ansible

```
vboxuser@ubuntu1:~$ sudo apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  ieee-data python-babel-localedata python3-argcomplete python3-babel
  python3-distutils python3-dnspython python3-jinja2 python3-jmespath
  python3-kerberos python3-libcloud python3-netaddr python3-ntlm-auth
  python3-packaging python3-pycryptodome python3-requests-kerberos
  python3-requests-ntlm python3-requests-toolbelt python3-selinux
  python3-simplejson python3-winrm python3-xmltodict
Suggested packages:
  cowsay sshpass python3-sniffio python3-trio python-jinja2-doc ipython3
  python-netaddr-docs
The following NEW packages will be installed:
  ansible ieee-data python-babel-localedata python3-argcomplete python3-babel
  python3-distutils python3-dnspython python3-jinja2 python3-jmespath
  python3-kerberos python3-libcloud python3-netaddr python3-ntlm-auth
```

Figure 48: Installation d'ansible

Après le téléchargement d'ansible, on passe à l'inventaire. L'inventaire nous permet de spécifier les différentes hôtes (clients) et le serveur que notre outil d'automatisation ansible doit configurer. Il est nécessaire de signaler que Ansible fonctionne grâce au protocole ssh pour assurer l'exécution des différentes playbooks.

```
  GNU nano 6.2                    /etc/ansible/hosts
[servers]

web-server ansible_host=192.168.1.1 ansible_user=vboxuser ansible_port=22

[clients]

clients1 ansible_host=192.168.1.3 ansible_user=vboxuser ansible_port=22

clients2 ansible_host=192.168.1.5 ansible_user=vboxuser ansible_port=22
```

Figure 49: Inventaire

Une fois l'inventaire terminé, on passe à une configuration d'un playbook écrit en YAML qui nous permet de faire plusieurs tâches telles que la mise à jour du système, l'installation, la configuration et l'exécution des service de ssh et en même temps l'ouverture d'un port de ssh dans le pare-feu ufw.

L’exécution du playbook se fait par la commande suivante : ansible-playbook -i /etc/ansible/hosts instalation_configuration_ssh. yml -- ask-become-pass

```
  GNU nano 6.2                                    instalation_configuration_ssh.yml *
- name: Mise a jours et telechargement et configuration de ssh dans les machine
  hosts: all
  become: yes
  tasks:
    - name: Mise a jours du systeme
      apt:
        update_cache: yes


    - name: installation de openssh
      apt:
        name: openssh-server
        state: present

    - name: Configuration du port ssh
      lineinfile:
        path: /etc/ssh/sshd_config
        regexp: '^#?Port'
        line: 'Port 22'
        state: present
        backup: yes

    - name: Execution des services ssh
      systemd:
        name: ssh
        state: started
        enabled: yes
    - name: Ouverture des port ssh dans ufw
      ufw:
        rule: allow
        port: '22'
        proto: tcp
    - name: Redemarrage des services ssh
      systemd:
        name: ssh
        state: restarted

    - name: Verification si ssh fonctionne avec le port 22
      shell: ss -tiln | grep :22
      register: ssh_port_check
      ignore_errors: yes
```

Figure 50:Automatisation de l’instalation et configuration de ssh

```
vboxuser@ubuntu1:~$ ansible-playbook -i /etc/ansible/hosts instalation_configuration_ssh.yml --ask-become-pass
BECOME password:

PLAY [Mise a jours et telechargement et configuration de ssh dans les machine] *************************************************
********

TASK [Gathering Facts] *********************************************************************************************************
********
ok: [clients2]
ok: [web-server]
ok: [clients1]

TASK [Mise a jours du systeme] *************************************************************************************************
********
changed: [clients2]
changed: [clients1]
changed: [web-server]

TASK [installation de openssh] *************************************************************************************************
********
ok: [clients1]
ok: [clients2]
ok: [web-server]

TASK [Configuration du port ssh] ***********************************************************************************************
********
ok: [web-server]
ok: [clients1]
ok: [clients2]

TASK [Execution des services ssh] **********************************************************************************************
********
ok: [clients2]
ok: [clients1]
ok: [web-server]
```

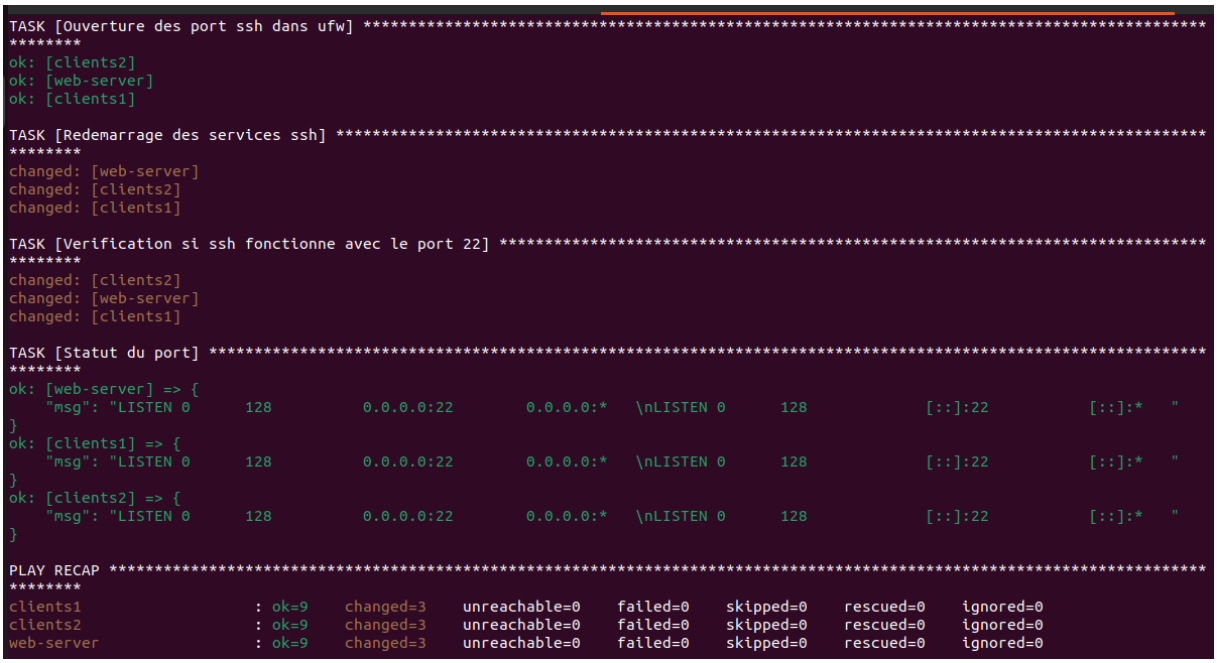


Figure 51: Résultats de l'automatisation de la configuration ssh

Cette capture nous permet de voir que l'exécution du playbook se passe bien et les services ssh et ufw sont bien installés et activés.

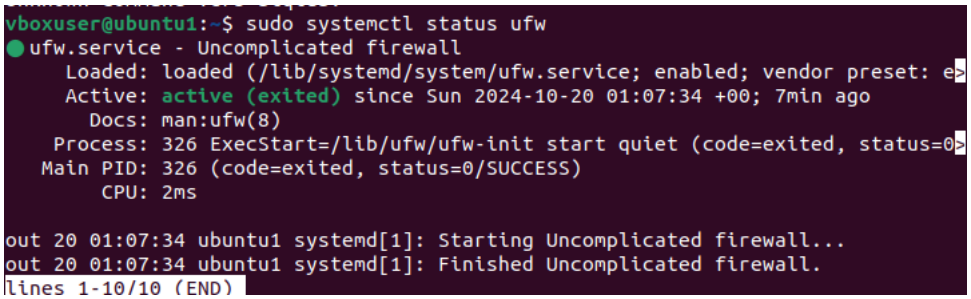


Figure 52: Statut du pare-feu ufw après automatisation

Après la mise à jour et la configuration de ssh, on ajoute trois autres utilisateurs avec des privilèges différents (user1 et user2 peuvent utilisés sudo et user3 sans les privilèges sudo) avec comme mot de passe : Passer123!. Mais ce mot de passe est crypté par Openssh par la commande : openssl passwd -6 'Passer123!'

```
 GNU nano 6.2                                    utilisateur.yml
---
- hosts: web-server
  become: yes
  vars:
   encrypted_password: "$6$ge5zYH0vb5PbUnTW$pE/dW7btUD6ACu1vv0pWmfaQTf2vQeo8EBv/DcEwIxkwWYBNsGHhY4vYDWcSo0Eokrxk3Xl54c>
  tasks:
   - name: Creation des utilisateurs avec acces sudo
     user:
       name: "{{ item }}"
       state: present
       group: "sudo"
       shell: /bin/bash
       password: "{{ encrypted_password }}"
     loop:
       - "user1"
       - "user2"

   - name: Creation de l'utilisateur sans acces sudo
     user:
       name: "user3"
       state: present
       group: "users"
       shell: /bin/bash
       password: "{{ encrypted_password }}"
```

Figure 53: Ajout des utilisateurs

Avec l'exécution de ce playbook, on voit que les trois utilisateurs sont créés avec des privilèges différents.

```
vboxuser@ubuntu1:~$ ansible-playbook -i /etc/ansible/hosts utilisateur.yml --ask-become-pass
BECOME password:

PLAY [web-server] **************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [web-server]

TASK [Creation des utilisateurs avec acces sudo] *******************************************************************
changed: [web-server] => (item=user1)
changed: [web-server] => (item=user2)

TASK [Creation de l'utilisateur sans acces sudo] *******************************************************************
changed: [web-server]

PLAY RECAP *********************************************************************************************************
```

Figure 54: Résultat de l'automatisation des users



Figure 55: Les différentes users

- Téléchargement et configuration d'apache (http et https) avec ansible

La configuration de ce playbook nous permet de télécharger apache, de configurer la connexion http et https, d'ajouter les ports dans le pare-feu ufw et en même temps de masquer les informations sur la version d'apache et du système d'exploitation sur lequel apache est installé.

```
---
- hosts: web-server
  become: yes
  tasks:
    - name: install apache2
      apt:
        name: apache2
        update_cache: yes
        state: latest
    - name: enable mod_rewrite
      apache2_module:
        name: rewrite
        state: present
      notify:
        - restart apache2
    - name: Configuration de apache pour masquer les information sur la version
      lineinfile:
        dest: /etc/apache2/conf-available/security.conf
        regexp: '^ServerSignature'
        line: 'ServerSignature Off'
      notify:
        - restart apache2

    - name: Activation de la configuration de securite apache
      command: a2enconf security
      notify:
        - restart apache2

  - name: apache2 ecouter dans le port 80
    lineinfile:
      dest: /etc/apache2/ports.conf
      regexp: "^Listen"
      line: "Listen 80"
    notify:
      - restarted apache2
  - name: apache2 virtualhost on port 80
    lineinfile:
      path: /etc/apache2/sites-available/000-default.conf
      regexp: "^<VirtualHost"
      line: "<VirtualHost *:80>"
    notify:
      - restart apache2
  - name: Ajout du module ssl
    apache2_module:
      name: ssl
      state: present
    notify:
      - restart apache2
```

```
- name: Generation du certificat auto-signe
  command: >
    openssl req -x509 -nodes -days 365 -newkey rsa:2048
    -keyout /etc/ssl/private/apache-selfsigned.key
    -out /etc/ssl/certs/apache-selfsigned.crt
    -subj "/C=FR/ST=Paris/L=Paris/O=Mycompany/OU=IT/CN=localhost"
- name: Configuration de la page https
  copy:
    dest: /etc/apache2/sites-available/default-ssl.conf
    content: |
      <IfModule mod_ssl.c>
        <VirtualHost *:443>
          ServerAdmin webmaster@localhost
          ServerName sn.com
          DocumentRoot /var/www/html
          DocumentRoot /var/www/sn


          SSLEngine on
          SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
          SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

          ErrorLog ${APACHE_LOG_DIR}/error.log
          CustomLog ${APACHE_LOG_DIR}/access.log combined


          <Directory /var/www/html>
            AllowOverride All
          </Directory>
        </VirtualHost>
      </IfModule>
  notify:
    - restart apache2
```

```
- name: Activation de la configuration
  command: a2ensite default-ssl.conf
  notify:
    - restart apache2

- name: Activation des ports http et https dans ufw
  ufw:
    rule: allow
    port: "80,443"
    proto: tcp


handlers:
  - name: restart apache2
    service:
      name: apache2
      state: restarted
    ignore_errors: yes
```

Figure 56: Automatisation du téléchargement et configuration d'apache

Ce scripte cache la version d'apache utilisée permet de se prémunir contre les attaques qui touchent directement la version d'apache.

Permettant aussi d'utiliser une méthode de communication sûre grâce à l'utilisation de https. La configuration du pare-feu ufw nous permet d'autoriser certains ports et d'en refuser pour le reste.

```
vboxuser@ubuntu1:~$ ansible-playbook -i /etc/ansible/hosts apache.yml --ask-become-pass
BECOME password:

PLAY [web-server] ***************************************************************************************

TASK [Gathering Facts] **********************************************************************************
ok: [web-server]

TASK [install apache2] **********************************************************************************
ok: [web-server]

TASK [enable mod_rewrite] *******************************************************************************
ok: [web-server]

TASK [Configuration de apache pour masquer les information sur la version] ******************************
changed: [web-server]

TASK [Activation de la configuration de securite apache] ************************************************
changed: [web-server]

TASK [apache2 ecouter dans le port 80] ******************************************************************
ok: [web-server]

TASK [apache2 virtualhost on port 80] *******************************************************************
ok: [web-server]

TASK [Ajout du module ssl] ******************************************************************************
ok: [web-server]

TASK [Generation du certificat auto-signe] **************************************************************
changed: [web-server]

TASK [Configuration de la page https] *******************************************************************
ok: [web-server]

TASK [Activation de la configuration] *******************************************************************
changed: [web-server]

TASK [Activation des ports http et https dans ufw] ******************************************************
ok: [web-server]

RUNNING HANDLER [restart apache2] ***********************************************************************
changed: [web-server]

PLAY RECAP **********************************************************************************************
web-server                 : ok=13   changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Figure 57: Résultat de l'automatisation de la configuration d'apache

```
vboxuser@ubuntu1:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
     Active: active (running) since Sun 2024-10-20 01:07:39 +00; 12min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 595 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUC>
   Main PID: 699 (apache2)
      Tasks: 55 (limit: 2271)
     Memory: 10.0M
        CPU: 261ms
     CGroup: /system.slice/apache2.service
             ├─699 /usr/sbin/apache2 -k start
             ├─702 /usr/sbin/apache2 -k start
             └─703 /usr/sbin/apache2 -k start

out 20 01:07:38 ubuntu1 systemd[1]: Starting The Apache HTTP Server...
out 20 01:07:39 ubuntu1 apachectl[627]: AH00112: Warning: DocumentRoot [/var/ww>
out 20 01:07:39 ubuntu1 systemd[1]: Started The Apache HTTP Server.
vboxuser@ubuntu1:~$
```

Figure 58: Vérification du statut d'apache

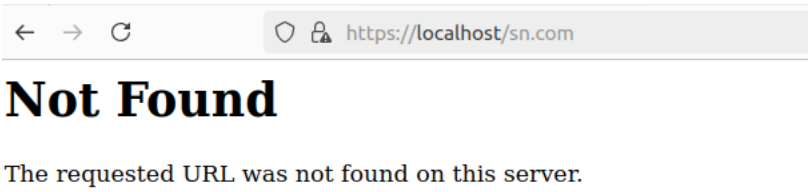


Figure 59: Vérification de la connexion avec apache

Ceci montre que les résultats de la scripte d'automatisation d'apache marchent. On peut alors se connecter avec https pour plus sécurité mais aussi une fois connecté, on ne peut pas voir ni la version d'apache ni sur quel système d'exploitation il a été installé.

Après la configuration d'apache, on passe au téléchargement et à la configuration de fail2ban pour filtrer les connexion indésirées http et https d'apache, bloquer les tentatives d'attaque par force brute en filtrant et en piégeant l'adresse IP de l'attaquant dans un prison ou dans le jail d'apache badbots. Le Jail apache-auth permet de gérer les erreurs dues à la mauvaise connexion http et https et le Jail par défaut permet lui aussi de ne pas bloquer les tentatives de connexion venant de 127.0.0.1/8 vers n'importe quel autre adresse IP.

```
  GNU nano 6.2                                          fail2ban.yml
---
- hosts: web-server
  become: yes
  tasks:


    - name: Installation de Fail2Ban
      apt:
        name: fail2ban
        state: present
        update_cache: yes
    - name: Configuration des jails ou prisons de fail2ban
      copy:
        dest: /etc/fail2ban/jail.local
        content: |
          [DEFAULT]
          ignoreip = 127.0.0.1/8 ::1
          bantime = 600
          findtime = 600
          maxretry = 5
          backend = systemd



          [apache-auth]
          enabled = true
          port = http,https
          filter = apache-auth
          logpath = /var/log/apache2/error.log
          maxretry = 3
          bantime = 600
```

```
        [apache-badbots]
        enabled = true
        port = http,https
        filter = apache-badbots
        logpath = /var/log/apache2/access.log
        maxretry = 5
        bantime = 86400
    notify:
      - restart fail2ban
  - name: Creation de filtre apache
    copy:
      dest: /etc/fail2ban/filter.d/apache-auth.conf
      content: |
        [Definition]
        failregex = ^<HOST> -.*"GET.*HTTP/.*" 401
        ignoreregex =
    notify:
      - restart fail2ban
handlers:
  - name: restart fail2ban
    service:
      name: fail2ban
      state: restarted
```

Figure 60: Automatisation de la configuration de fail2ban

L'exécution de la du playbook de fail2ban montre qu'ansible a bien automatisé le téléchargement de fail2ban, la configuration de tous les Jails et le redémarrage des services fail2ban.

```
vboxuser@ubuntu1:~$ sudo systemctl status fail2ban.service
● fail2ban.service - Fail2Ban Service
     Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor pres>
     Active: active (running) since Mon 2024-12-09 12:06:30 -01; 5s ago
       Docs: man:fail2ban(1)
   Main PID: 2319 (fail2ban-server)
      Tasks: 9 (limit: 2271)
     Memory: 23.5M
        CPU: 282ms
     CGroup: /system.slice/fail2ban.service
             └─2319 /usr/bin/python3 /usr/bin/fail2ban-server -xf start
```

Figure 61: Vérification du statut de fail2ban après automatisation

L'ensemble des configurations sont faites sur quatre scriptes différents et le tout peut-être importé dans un playbook qui nous permet d'exécuter l'ensemble des scriptes avec une seule ligne de commande : ansible-playbook -i /etc/ansible/hosts principal.yml –ask-become-pass

```
vboxuser@ubuntu1:~$ ansible-playbook -i /etc/ansible/hosts fail2ban.yml --ask-become-pass
BECOME password:

PLAY [web-server] ***************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [web-server]

TASK [Installation de Fail2Ban] *************************************************************************************
changed: [web-server]

TASK [Configuration des jails ou prisons de fail2ban] ***************************************************************
changed: [web-server]

TASK [Creation de filtre apache] ************************************************************************************
changed: [web-server]

RUNNING HANDLER [restart fail2ban] **********************************************************************************
changed: [web-server]

PLAY RECAP **********************************************************************************************************
web-server                 : ok=5    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Figure 62: Résultat de l'automatisation de fail2ban

```
 GNU nano 6.2                                          principal.yml
---
- import_playbook: instalation_configuration_ssh.yml
- import_playbook: utilisateur.yml
- import_playbook: apache.yml
- import_playbook: fail2ban.yml
```

Figure 63: Regroupement des différents playbook

- Test et vérification des résultats de l'automatisation

Après la configuration du système avec les scripts ansible, on passe à des tests pour regarder la robustesse du système. C'est en ce sens qu'on a passé un test de reconnaissance avec l'outil nmap pour regarder les ports qui sont ouverts pour une éventuelle attaque.

```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.1.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-05 21:23 EDT
Note: Host seems down. If it is really up, but blocking our ping probes, try
-Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.07 seconds

┌──(kali㉿kali)-[~]
└─$ nmap -Pn 192.168.1.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-05 21:31 EDT
Nmap scan report for 192.168.1.1
Host is up (0.047s latency).
All 1000 scanned ports on 192.168.1.1 are in ignored states.
Not shown: 990 filtered tcp ports (no-response), 10 filtered tcp ports (host-
unreach)

Nmap done: 1 IP address (1 host up) scanned in 18.95 seconds
```

Figure 64: Vérification des services et ports ouverts

Bien que nmap envoie 1000 requêtes pour faire la reconnaissance, on voit que nmap n'a pas trouvé de port ouvert.

Après nmap, on répète le même processus de scanne avec hping3 en ciblant le scanne dans les ports 80 et 443 avec l'option know qui permet le scanne dans les ports connus.

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 --scan 80,443,known -S 192.168.1.1
Scanning 192.168.1.1 (192.168.1.1), port 80,443,known
264 ports to scan, use -V to see all the replies
+----+------------+--------+---+-----+-----+-----+
|port| serv name  |  flags |ttl| id  | win | len |
+----+------------+--------+---+-----+-----+-----+
All replies received. Done.
Not responding ports: (1 tcpmux) (2 nbp) (4 echo) (6 zip) (7 echo) (9 discard
) (11 systat) (13 daytime) (15 netstat) (17 qotd) (19 chargen) (20 ftp-data)
(21 ftp) (22 ssh) (23 telnet) (25 smtp) (37 time) (43 whois) (49 tacacs) (53
domain) (67 bootps) (68 bootpc) (69 tftp) (70 gopher) (79 finger) (80 http) (
88 kerberos) (102 iso-tsap) (104 acr-nema) (106 poppassd) (110 pop3) (111 sun
rpc) (113 auth) (119 nntp) (123 ntp) (135 epmap) (137 netbios-ns) (138 netbio
s-dgm) (139 netbios-ssn) (143 imap2) (161 snmp) (162 snmp-trap) (163 cmip-man
) (164 cmip-agent) (174 mailq) (177 xdmcp) (179 bgp) (199 smux) (209 qmtp) (2
10 z3950) (213 ipx) (319 ptp-event) (320 ptp-general) (345 pawserv) (346 zser
v) (369 rpc2portmap) (370 codaauth2) (371 clearcase) (389 ldap) (427 svrloc)
(443 https) (444 snpp) (445 microsoft-d) (464 kpasswd) (465 submissions) (487
 saft) (500 isakmp) (512 exec) (513 login) (514 shell) (515 printer) (517 tal
k) (518 ntalk) (520 route) (538 gdomap) (540 uucp) (543 klogin) (544 kshell)
(546 dhcpv6-clie) (547 dhcpv6-serv) (548 afpovertcp) (554 rtsp) (563 nntps) (
587 submission) (607 nqs) (623 asf-rmcp) (628 qmqp) (631 ipp) (636 ldaps) (64
```

Figure 65: Utilisation de hping3 pour scanner les ports

Hping3 nous donne le même résultat que nmap puisqu'il ne nous montre pas de port ouvert.

Comme aucun des ports scannés n'a répondu, on peut penser que les ports sont soient inaccessibles soient bloqués. Ces résultats peuvent être dus au pare-feu ufw installé dans notre serveur et que la fonction première de ufw est de bloquer toutes les requêtes entrantes et d'autoriser toutes les requêtes sortantes. Donc on peut penser que ces résultats pourraient être dus au pare-feu.

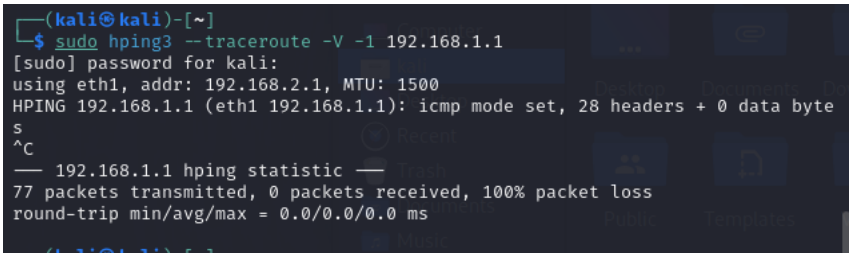


Figure 66: Utilisation de hping3 pour s'informer des nœuds et les routes

Toujours avec hping3 on tente de tracer le chemin des paquets icmp vers l'adresse 192.168.1.1. Cependant tous les paquets sont perdus à 100%. On peut conclure que les paquets perdus sont dus au pare-feu qui rejette toutes les requêtes entrant au serveur.

Conclusion

Avec le développement accru de l'internet, les risques de se faire pirater augmentent de jour en jour. Les serveurs web sont des cibles potentielles dans un réseau puisque le serveur est souvent joignable sur internet. C'est en ce sens que renforcer la sécurité du serveur pour protéger les données devient vital pour les organisations.

Conclusion Général

Le renforcement des serveurs web constitue une nécessité dans un environnement numérique en constante évolution où les menaces et les problèmes de sécurité ne cessent de se multiplier. Ce travail nous a permis de mettre en évidence l'importance de la sécurité des serveurs web face aux menaces cybernétiques en adoptant des règles et recommandations reconnues par les organisations telles que OWASP et ANSSI. La sécurité des systèmes d'information repose sur des principes qui se basent sur la réduction des surfaces d'attaque, la protection des données sensibles, l'assurance des mises à jour réguliers et l'application des configurations de sécurité stricte.

L'audit initial, la gestion des correctifs, la configuration des pares-feux, la gestion des identités et des accès et le chiffrement des communications constituent des étapes clés pour le processus de renforcement des serveurs. Ces différentes étapes contribueront à améliorer la sécurité et la résilience des systèmes face aux menaces interne et externe. A cela s'ajoute la gestion des logs et la surveillance permettant de détecter rapidement les anomalies et assurer une défense efficace et continue.

L'automatisation des processus de renforcement des systèmes joue un rôle primordial dans les infrastructures informatiques puisqu'elle permet de réduire les erreurs humaines, améliorer l'efficacité et la rapidité du déploiement de sécurité. L'utilisation des outils tels qu'Ansible par les entreprises permet de configurer et de sécuriser les différents serveurs tout en restant sur place et respecter les bonnes pratiques.

Ainsi, l'ensemble des mesures décrites offre une protection robuste contre les menaces actuelles et ultérieurs. Il est important de maintenir une surveillance constante qui intègre la technologie et en adaptant de manière continue les stratégies de défense et de résilience. Le renforcement des serveurs web est une activité continue dans le temps et qui permet de créer un environnement de confiance pour les utilisateurs et à garantir la sécurité des données critique.

Bien que l'automatisation réduise les erreurs humaines, elle accélère le déploiement des configurations et améliore la robustesse des systèmes. Mais il faut noter que l'automatisation seule ne garantit pas une sécurité absolue sans une surveillance continue. C'est pourquoi intégrer des outils SIEM (Systèmes de gestion des informations et des évènements de sécurité) pour détecter et réagir aux menaces en temps réel.

Références

[1] Améliorer la sécurité du système du serveur Web grâce au renforcement automatique du système : https://easychair.org/publications/preprint_english/WGHg

[2] A05 Mauvaise configuration de sécurité : https://owasp.org/Top10/fr/A05_2021-Security_Misconfiguration/

[3] https://www.scirp.org/journal/paperinformation?paperid=125184

[4] https://www.chef.io/blog/system-hardening-with-chef-for-a-secure-it-infrastructure

[5] https://www.chef.io/blog/system-hardening-with-chef-for-a-secure-it-infrastructure

[6] Renforcement des serveurs web apache : https://beaglesecurity.com/blog/article/apache-web-server-hardening.html

[7] En quoi consiste le durcissement d'un serveur http://www.cncrgroup.com/en-quoi-consiste-le-processus-de-durcissement-dun-serveur/ Consulté le 04/07/2024 à 22h 50 min

[8] Audit de sécurité Web Audit Sécurité Site Web : Testez et Renforcez votre Protection (akaoma.com) Visité le 15/07/2024 13h 20

[9] 10 vulnérabilités de sécurité web 10 vulnérabilités de sécurité Web les plus courantes (guru99.com) Visité le 15/07/2024 à 20h 25

[10] Qu'est-ce que la gestion des correctifs : Qu'est-ce que la gestion des correctifs ? | IBM

[11] Comprendre et configurer les pares-feux pour une protection optimal Comprendre et Configurer les Pare-Feux pour une Protection Optimale - CyberInstitut

[12] Comprendre la gestion des identités et des accès (IAM) : un guide complet Le guide complet de la gestion des identités et des accès : avantages, technologies et pratiques d'excellence | Veritas

[13] Sécurité des applications : menaces, outils et techniques Sécurité des applications : menaces, outils et techniques (crowdstrike.fr) Visité le 18/07/2024 à 22h 45

[14] Qu'est-ce que la **gestion des logs** ? L'importance de la journalisation et des bonnes pratiques Qu'est-ce que la gestion des logs ? Bonnes pratiques - CrowdStrike Visité le 25/07/2024 à 16h 12 min

[15] Quels sont les défis courants du renforcement et de l'audit du système d'exploitation ?

https://www.linkedin.com/advice/0/what-common-challenges-os-hardening-auditing-xvrvf?lang=fr&originalSubdomain=fr

[16] Avis du CERT FR https://www.cert.ssi.gouv.fr/avis/CERTFR-2023-AVI-0244/

[17] Automatisation complète de la fourniture d'applications, de sécurité et d'infrastructures avec Chef EAS https://www.chef.io/products/enterprise-automation-stack

[18] Comment sécuriser votre serveur Linux : un guide détaillé

https://www.plesk.com/blog/various/how-to-secure-your-linux-server-a-detailed-guide/?utm_campaign=remarketing&utm_source=google&utm_medium=ppc&utm_term=&gad_source=1&gclid=CjwKCAjwqf20BhBwEiwAt7dtdWThC5MNdMUGpn28T9fZCsYQIBXhGAymfQLo7achdslmU9bwPM4hGxoCBawQAvD_BwE

[19] Comment sécuriser votre serveur Linux : un guide détaillé :

https://net-security.fr/securite/durcissement-gnulinux-1/#durcissement-du-serveur-ssh

[20] Durcissement d'un système GNU/Linux : Gestion des utilisateurs & des connexions : https://net-security.fr/securite/durcissement-gnulinux-1/

[21] 18 conseils de sécurité et de renforcement du serveur Web Apache

https://www.tecmint.com/apache-security-tips/

[22] Qu'est-ce que la **gestion des logs** ? L'importance de la journalisation et des bonnes pratiques

[23] Configurer le ssl avec apache2 https://www.it-connect.fr/configurer-le-ssl-avec-apache-2%EF%BB%BF/