



## INSTITUT DE FRANCOPHONE INTERNATIONAL

RAPPORT DE LA PARTIE THÉORIQUE DU TPE

---

# SECURISER POSTFIX

---

Réalisé par :  
VIGAN Silas

*Sous la supervision de :*  
Dr NGUYEN Hong Quang  
Dr Ho Tuong Vinh

Promotion 23 RSC

# Table des matières

<b>Introduction</b>	<b>3</b>
<b>1 Analyse de Sujet</b>	<b>3</b>
Problématique . . . . .	3
Problème à résoudre . . . . .	3
Etat Actuel . . . . .	4
Objectifs . . . . .	4
Objectifs Techniques . . . . .	4
<b>2 Etude bibliographique</b>	<b>5</b>
2.1 Les Services de Courrier Electronique : Menaces et Vulnérabilités	5
Accès non autorisé . . . . .	5
Fuite de données . . . . .	6
Le Spamming . . . . .	6
Les menaces liées aux Malwares . . . . .	6
Les menaces de Déni de Service . . . . .	7
Les menaces liées au non respect des bonnes pratiques . . . . .	7
2.2 Le serveur de messagerie Postfix . . . . .	7
Organisation du transfert de message sur Internet . . . . .	7
2.2.1 Origine et Philosophie Postfix . . . . .	8
2.2.2 Postfix : menaces et vulnérabilités . . . . .	10
2.3 Les outils d'analyse de vulnérabilité et d'audit de sécurité . . . . .	13
2.3.1 Lynis . . . . .	13
2.3.2 Nessus . . . . .	13
2.3.3 Email Security Grader [9] . . . . .	13
2.3.4 Check TLS [10] . . . . .	13
2.3.5 Spamassassin . . . . .	14
2.3.6 ClamAV . . . . .	14
<b>3 Proposition de solution</b>	<b>14</b>
3.1 Ébauche de Solution . . . . .	14
3.2 Démarche de Réalisation de la Solution, Planification et Outils . . . . .	14

<b>4</b>	<b>Implémentation de la solution proposée</b>	<b>15</b>
4.1	Proposition d'architecture . . . . .	16
4.2	Mise en oeuvre de l'architecture . . . . .	17
4.2.1	Installation du Sytème d'exploitation et des services de base	17
4.2.2	Test de niveau de sécurité du Système à son état par défaut	17
4.2.3	Script d'audit . . . . .	17
4.3	Eléments de contrôle . . . . .	20
4.3.1	Objectifs de contrôle . . . . .	20
4.3.2	Définition de la liste de contrôle . . . . .	21
4.3.3	Définition des points d'amélioration . . . . .	22
4.4	Mise en oeuvre des points d'amélioration . . . . .	23
4.5	Définition d'une politique de mot de passe . . . . .	23
4.6	Désactiver Telnetd . . . . .	24
4.7	Restreindre le redémarrage console . . . . .	24
4.8	Sécurité antispam . . . . .	24
4.8.1	Présentation des services SPF et DKIM . . . . .	25
4.8.2	Configuration de DKIM et SPF . . . . .	26
4.9	Installation d'antimalware ClamAv . . . . .	27
4.10	Configuration de TLS . . . . .	28
4.10.1	Création d'une autorité de Certification . . . . .	28
4.10.2	Configuration de TLS . . . . .	30
4.11	Configuration d'un acces à la messagerie par interfaces WEB . .	31
<b>5</b>	<b>Présentation des résultats</b>	<b>31</b>
5.1	Accès au serveur mail par Web . . . . .	32
5.2	Le protocole TLS . . . . .	33
5.3	Sécurité antimalware . . . . .	34
5.4	Perfect Forward Secrecy . . . . .	34
5.5	Evaluation des Résultats . . . . .	35
5.5.1	Appréciation des tâches réalisées . . . . .	35
5.5.2	Difficultés . . . . .	36
5.5.3	Perspectives . . . . .	36
	<b>Conclusion</b>	<b>38</b>
	<b>Références Bibliographiques</b>	<b>39</b>

# Introduction

Depuis l'envoi du premier message à travers l'Arpanet, ancêtre de l'internet en 1970, les services de courriers électroniques sont parmi les plus utilisés des services réseaux. L'usage des services de courriers électroniques s'est imposé aux utilisateurs des systèmes d'information dans tous les domaines. Tenant compte du flux de données transitant par les différents serveurs et clients de messagerie, il est d'autant plus vital de mettre en oeuvre toutes les diligences pour garder ce service sécurisé en vue de garantir la confidentialité, l'intégrité et la disponibilité des services de courriers électroniques. Pour la plupart, les services de courriers électroniques tournent en environnement Linux notamment sous le service Postfix. Il nous revient donc d'étudier le fonctionnement, l'architecture, l'environnement de déploiement d'un serveur postfix afin de le déployer en respectant les standards de sécurité.

## 1 Analyse de Sujet

### Problématique

De nos jours, la sécurité des systèmes d'informations est devenu un sujet prépondérant et s'est vu subdiviser en plusieurs branches du fait de la recrudescence des menaces et l'amélioration continue des méthodes et outils d'attaques. Malgré la prolifération des réseaux sociaux, le courrier électronique demeure le mode de communication préféré par la plupart des entreprises dans le monde et il n'existe aucune donnée indiquant un abandon progressif de ce mode de communication. Compte tenu des données transitant par les services mails, ces derniers deviennent une cible idéale pour les attaques diverses à buts malveillants. À l'instar des autres institutions de part le monde, l'IFI s'est doté depuis quelques années d'un serveur de courrier électronique. Ce serveur tourne sous postfix dans un environnement Linux. Dans le contexte des attaques récurrentes ciblant le service de courrier électronique de l'IFI, il serait salvateur d'effectuer une étude des risques auxquels sont exposés le serveur mail de l'institut. Quels pourraient donc être les mécanismes à mettre en oeuvre pour sécuriser d'avantage le serveur ? Dans cette posture, il nous revient de trouver les outils idéaux pour un audit du serveur. Quelles seraient les dispositions, règles et politiques à mettre en place pour mettre en oeuvre, suivre et maintenir un environnement de sécurité autour du serveur mail de l'institut ?

### Problème à résoudre

Le serveur mail de l'IFI est déployé depuis un certain nombre d'années. Il a dû au fil du temps devenir obsolète du point de vue de sa capacité à traiter les différentes requêtes, du point de vue de sa sécurité mais aussi du point de vue de sa conformité aux normes et exigences des standards de sécurité des services de courrier électronique.

Au cours de ce travail nous aurons donc à mettre tous les moyens et toutes les ressources en oeuvre afin d'évaluer les risques de sécurité, les écarts de conformité

du serveur dans son état actuel et ensuite étudier les possibilités de renforcement du niveau de sécurité du serveur et proposer un modèle d'architecture à même de garantir aux utilisateurs, la confidentialité, l'intégrité et la disponibilité de leur service de courriers électroniques.

## Etat Actuel

Plusieurs technologies, outils et méthodologies sont actuellement disponibles pour aider à concevoir, implémenter et auditer un environnement sécurisé de service mail. Au nombre de ceux ci, nous avons :

- Les standards : *NIST*, *PCI DSS*, *HIPAA*, *Sarbane Oxley*, qui définissent un certain nombre de normes pour régir l'encryption des mails.
- Les approches et technologies de vérification TLS
- Les technologies de Greylisting et de DNS Blacklisting
- Le Système SPF (*Sender Policy Framework*)

## Objectifs

Les objectifs de ce travail sont de deux ordres : *les objectifs théoriques et les objectifs techniques*.

### Objectifs Théoriques

Les objectifs théoriques sont :

- Définir le périmètre du travail
- Etudier l'existant
- Présenter un état des lieux
- Faire une analyse des outils et solutions adéquates pour la réalisation du travail
- Choix des outils adéquats
- Définir une démarche claire de la réalisation du travail.

### Objectifs Techniques

Les objectifs techniques sont :

- Effectuer un audit des vulnérabilités et risques de sécurité du serveur postfix
- Fournir un rapport clair et exhaustif de l'audit effectué
- Proposer les solutions de remédiation aux insuffisances relevées dans le rapport
- Mettre en oeuvre les recommandations issues du rapport
- Effectuer des tests pour valider les résultats
- Etudier une architecture sécurisée et une démarche de déploiement et de maintenance d'un nouveau serveur postfix
- Produire une Politique de sécurité et un référentiel de bonnes pratiques de sécurité.

## 2 Etude bibliographique

### 2.1 Les Services de Courrier Electronique : Menaces et Vulnérabilités

Les services mail aujourd'hui représentent l'une des principales portes d'entrée et de sortie d'un système d'information. Ils sont de ce fait sujet à plusieurs attaques exploitant les vulnérabilités de ces services. Une étude menée en 2018 par *Alexandre Marguerite*, Directeur Technique et Associé chez Devensys, une entreprise spécialisée en sécurité informatique [7], les pertes liées aux attaques ciblant les services mail s'élèvent à plusieurs millions d'euros.

L'image ci dessous présente un petit aperçu des mouvements engendrés par les attaques contre les services mail.

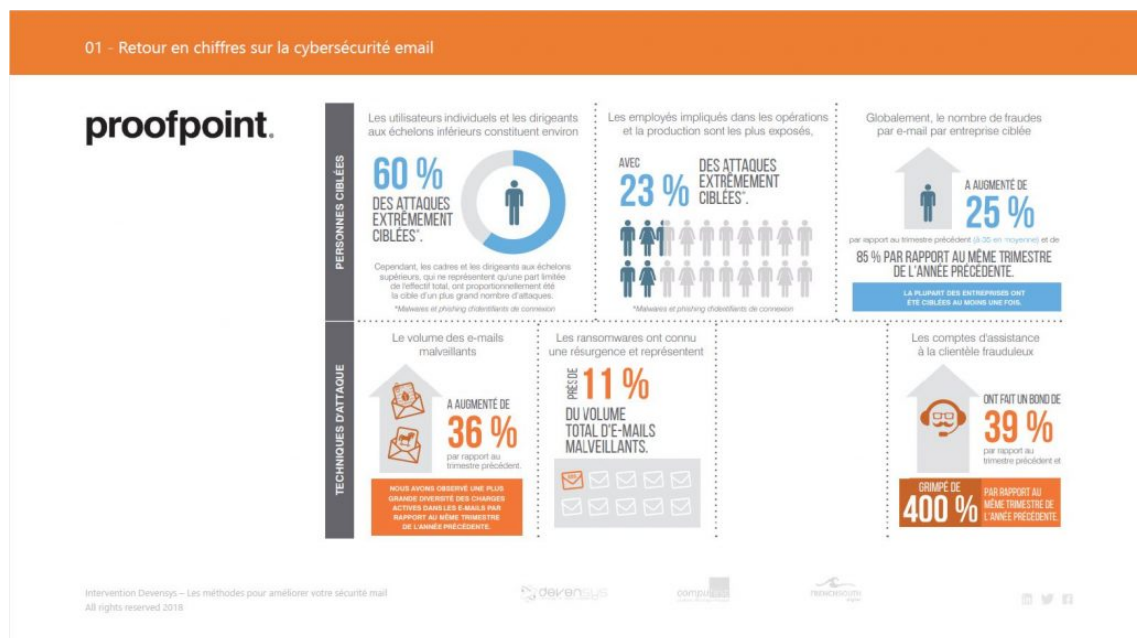


FIGURE 1 – Evolution des attaques sur les serveurs mail, tiré de la référence [7]

De notre recherche bibliographique, nous pouvons classer les principales menaces selon ce qui suit :[8]

#### Accès non autorisé

L'accès non autorisé aux données et ressources fait partie des principales menaces qui planent sur les services mail. Cet état de chose est dû à la faiblesse du niveau d'authentification. Souvent, les administrateurs définissent des mots de passe faibles, prédictibles parfois même ils laissent par défaut les mots de passe des services permettant ainsi aux individus malveillants d'accéder aux services

en utilisant les techniques d'attaques par force brute, d'attaques par dictionnaire ou encore les attaques de type Rainbow Table.

Plusieurs solutions sont envisageables pour palier à ce type de menace nous avons :

- l'usage de mots de passe forts pour l'accès au serveur
- l'authentification SMTP

### Fuite de données

Les données personnelles représentent la cible première des hackers ainsi, ils sont à l'affut de toute communication, toute transmission de données passant par un canal non sécurisé et non chiffré passant à travers internet. Au nombre de ces données nous avons : les messages, les identifiants, les mots de passe etc... Pour prévenir la fuite de données, il est conseillé de chiffrer toutes les communications en entrée comme en sortie (IMAP, POP3, SMTP) avec les technologies SSL/TLS

### Le Spamming

L'une des plus grandes menaces à l'endroit des services mail est celle des courriers indésirables (spam). Ces menaces se déclinent sous deux formes :

- envoi de spam à ses clients propres
- envoi de spam aux clients d'un autre serveur (au cas où le serveur joue le rôle de relai SMTP)

En effet, le spam est considéré comme un message indésiré provenant généralement d'un expéditeur inconnu dans le but de vendre un service quelconque. La plupart du temps, ce sont des messages qui conduisent vers de l'anarchie. Le spamming aujourd'hui a évolué et sert de canal pour un nouveau concept qui est le *phishing* qui consiste à amener l'utilisateur à cliquer sur un lien pour être redirigé vers une page où il saisira ses données personnelles : *login, mot de passe, données de carte bancaires, numéro de téléphone, etc...*

Pour prévenir ce genre de menace, il faut prévoir des outils de filtrage de contenu soient-ils des :

- firewall applicatifs qui est un programme permettant de filtrer les contenus entrant et sortant,
- firewall matériels c'est un hardware dédié au filtrage des paquets entrant et sortant, il permet de bloquer le contenu indésirable.

mais également prévoir des moyens de Blacklisting des serveurs de spam déjà connus.

### Les menaces liées aux Malwares

Et les serveurs et les clients sont susceptibles d'être infectés par des malwares. Les malwares sont en général des programmes malveillants qui s'avèrent être nocifs au système informatique une fois ce dernier infecté. Les malwares une fois dans un système ont la capacité de se propager dans tout le système d'information, mettant à mal ce dernier. Dans ce cas, la confidentialité et l'intégrité des données ne sont plus garanties. Pour éviter les malwares, il est important de prévoir plusieurs outils dont les antivirus en principal.

## Les menaces de Déni de Service

Les dommages qu'une attaque par déni de service peut causer à un serveur de messagerie sont difficiles à surestimer. Cela conduit à des e-mails non reçus et non envoyés, sans parler du temps passé à essayer de restaurer le service. En fin de compte, la réputation de l'ensemble de la société en souffre.

Pour éviter une telle menace, vous devez au moins limiter le nombre de connexions possibles au serveur SMTP. Afin de résoudre les problèmes de sécurité SMTP, envisagez de limiter le nombre total de connexions dans le temps, ainsi que les connexions simultanées.

Selon une étude de OVH, hébergeur français, le volume des attaques par déni de service ont augmenté en volume allant jusqu'à 1800 cas d'attaques en moyenne par jour ceci toutes cibles confondues [6].

## Les menaces liées au non respect des bonnes pratiques

L'une des meilleures pratiques les plus pertinentes consiste à ne rien stocker d'inutile sur le serveur. Le serveur doit être examiné avec soin afin de vérifier si un logiciel supplémentaire, installé sur le serveur, ne pourrait pas causer une vulnérabilité exploitable par un individu malveillant. Ensuite, vérifier chacun des ports réseau ouverts pour s'assurer qu'ils sont nécessaires (sinon, les fermer immédiatement) et protégés (par exemple, vérifier si une autorisation est requise pour envoyer des données via un port).

Il faut également mettre à jour tous les logiciels sur le serveur. Malgré les efforts des développeurs et des testeurs, aucun logiciel n'est exempt d'erreur à 100%. Il existe des listes entières de vulnérabilités logicielles facilement accessibles aux pirates informatiques. Lorsque de nouvelles vulnérabilités ou exploits sont détectés, les éditeurs de logiciels publient généralement un correctif sur plusieurs jours. Si le serveur ne reçoit pas cette mise à jour à temps, les auteurs d'infractions pourront utiliser cette vulnérabilité à leur avantage.

Le facteur humain est capital et doit être pris en compte. La disponibilité du serveur ne doit pas dépendre d'une seule personne.

## 2.2 Le serveur de messagerie Postfix

### Organisation du transfert de message sur Internet

Sur Internet, pour le transfert d'un mail de son expéditeur vers son destinataire, plusieurs agents sont sollicités. Au nombre de ces agents nous avons :

- Le **Mail User Agent (MUA)** qui est souvent l'interface client avec laquelle l'utilisateur interagit pour rédiger ou lire son message. Cela peut être par exemple : *Pine*, *Netscape Communicator*, ou *Outlook Express*.
- Le **Mail Transfer Agent (MTA)** il s'agit du serveur qui reçoit et distribue le courrier électronique. Il détermine le routage des messages et gère la réécriture possible des adresses. Les messages à livrer localement sont remis à une MDA pour la livraison finale.



- Le **Mail Delivery Agent (MDA)** C'est un programme qui gère la livraison finale des messages pour les destinataires locaux d'un système. Les MDA peuvent souvent filtrer ou catégoriser les messages à la livraison. Un MDA pourrait déterminer également si un message doit être transféré vers une autre adresse électronique.

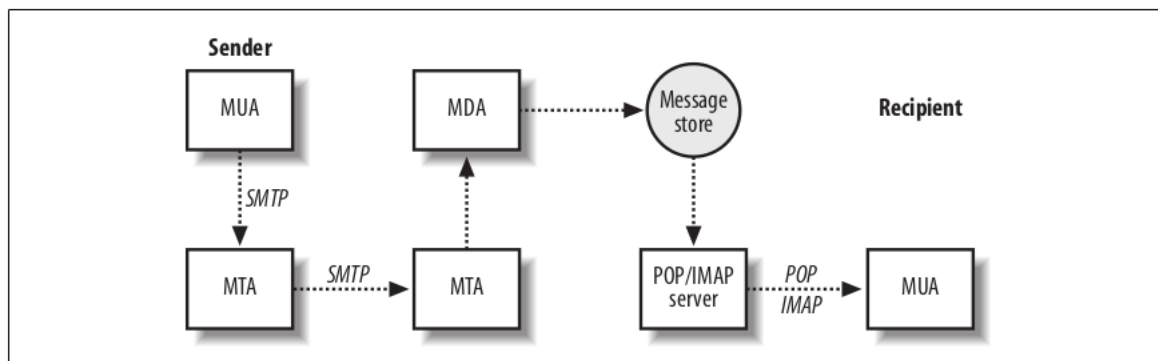


FIGURE 2 – Processus de transmission d'un mail [11]

### 2.2.1 Origine et Philosophie Postfix

Postfix a été écrit par Wietse Venema, réputé pour ses outils et documents de sécurité. Il a été rendu disponible en tant que logiciel open source en décembre 1998. IBM Research a parrainé la version initiale et a continué à soutenir son développement en cours. (IBM appelle le package Secure Mailer.)

[11]

Postfix est un MTA et gère la livraison des messages entre les serveurs et localement dans un système. Il ne gère aucune communication POP ou IMAP. La figure ci-dessous illustre un exemple simple de transmission de message où Postfix gère les responsabilités du MTA et de la livraison locale. En tant que MTA, Postfix reçoit et distribue des e-mails sur le réseau via le protocole SMTP. Pour une livraison locale, l'agent de distribution local de Postfix peut déposer des messages directement dans une banque de messages ou transmettre un message à un agent de distribution de courrier spécialisé.

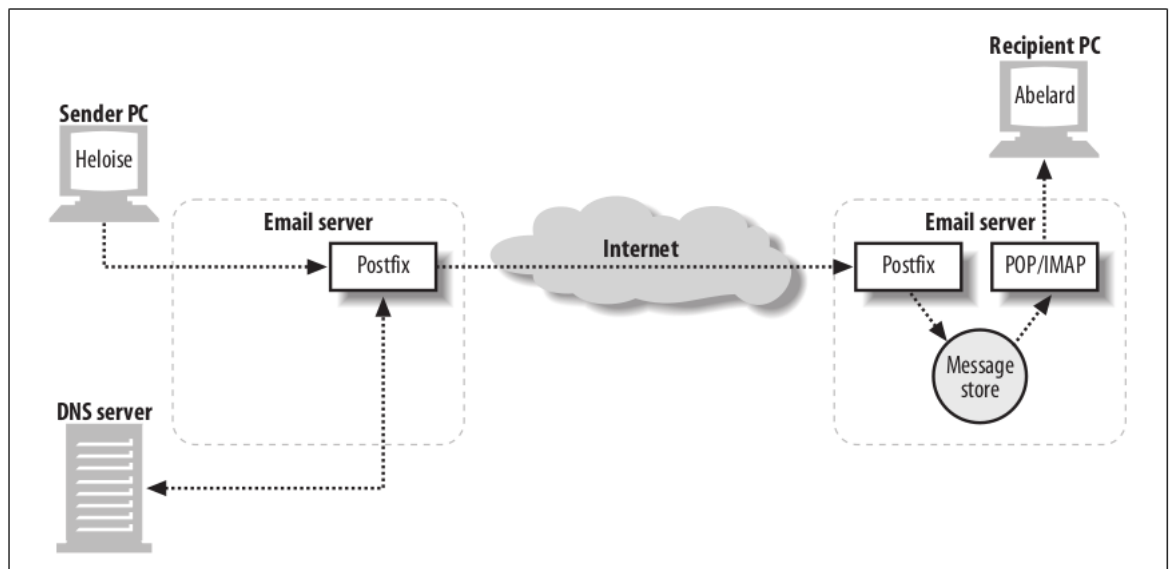


FIGURE 3 – Rôle de Postfix dans la transmission d'un message [11]

C'est ce qui a motivé la conception et le développement de Postfix c'est la recherche de :

### Fiabilité

Postfix [11] montre sa valeur réelle lorsqu'il est utilisé dans des conditions difficiles. Même dans des environnements simples, les logiciels peuvent rencontrer des conditions inattendues. Par exemple, de nombreux systèmes logiciels se comportent de manière imprévisible lorsqu'ils manquent de mémoire ou d'espace disque. Postfix détecte de telles conditions et, au lieu d'aggraver le problème, donne au système une chance de récupérer. Quels que soient les aléas rencontrés, Postfix prend toutes les précautions pour fonctionner de manière stable et fiable.

### Sécurité

Postfix suppose qu'il est en cours d'exécution dans un environnement hostile. Il utilise plusieurs couches de défense pour se protéger contre les attaquants. Le concept de sécurité du moindre privilège est utilisé dans l'ensemble du système Postfix, de sorte que chaque processus, qui peut être exécuté dans un compartiment isolé, s'exécute avec le moins de privilèges possible pour son fonctionnement. De même, les modules inutiles peuvent être désactivés, ce qui améliore la sécurité et simplifie l'installation[11].

## Performances

Postfix a été conçu dans un souci de performance. En fait, il prend des mesures pour que sa vitesse ne diminue pas. Il utilise des techniques pour limiter le nombre de nouveaux processus à créer et le nombre d'accès au système de fichiers requis pour le traitement des messages.

## Flexibilité

Le système Postfix est composé de plusieurs programmes et sous-systèmes différents. Cette approche permet une grande flexibilité. Toutes les pièces sont facilement réglables via des fichiers de configuration simples.

## Facilité d'utilisation

Postfix est l'un des paquetages de messagerie les plus faciles à configurer et à administrer, car il utilise des fichiers de configuration simples et des tables de recherche simples pour la conversion d'adresses et la conversion d'adresses. expéditeur. L'idée derrière la configuration de Postfix est la notion de moindre surprise, ce qui signifie que, dans la mesure du possible, Postfix se comporte comme prévu par la plupart des gens. Face aux choix de conception, le Dr. Venema a opté pour la décision qui semble la plus raisonnable pour la plupart des humains.

Selon les résultats d'une étude comparative [5], menée en Décembre 2015, Postfix est le second service de messagerie utilisé après *MICROSOFT EXCHANGES SERVER* avec 20% d'utilisation sur le plan mondial. ***Le serveur mail de l'IFI également tourne sous le service Postfix.***

### 2.2.2 Postfix : menaces et vulnérabilités

Postfix étant un serveur de messagerie, les vulnérabilités et menaces énumérées au chapitre précédent lui sont aussi inhérentes. Mais au delà de celles ci qui sont en quelques sortes imputable à tous les serveurs de messagerie, les serveurs de messagerie tournant sous Postfix sont sujets à d'autres menaces et vulnérabilités qui sont propres à Postfix et qui sont dues à des erreurs de configurations ou parfois des défauts dans le développement d'une version *x ou y* de Postfix. Le tableau ci dessous fait une synthèse de ces menaces et vulnérabilités mais aussi des impacts de leur exploitation par des individus malveillants.

Menaces	Description des Vulnérabilités	Impact de l'exploitation	Complexité	Exigence d'authentification
Privilege Escalation	Postfix avant les versions 2.11.10, 3.0.x avant 3.0.10, 3.1.x avant 3.1.6 et 3.2.x avant 3.2.2 admet une vulnérabilité qui pourrait permettre aux utilisateurs locaux d'obtenir des privilèges en exploitant des fonctionnalités non documentées dans Berkeley DB 2.x et ses versions ultérieures, liées à la lecture des paramètres dans le fichier DB_CONFIG du répertoire courant	<b>Sur la confidentialité</b> : Partiel (la divulgation d'informations est considérable.) <b>Sur l'intégrité</b> : Impact Partiel (La modification de certains fichiers système ou de certaines informations est possible, mais l'attaquant n'a aucun contrôle sur ce qui peut être modifié, ou l'étendue de ce qu'il peut affecter est limitée.) <b>Sur la disponibilité</b> : partiel (la performance des ressources est réduite ou la disponibilité des ressources est interrompue.)	<b>Faible</b> : il n'existe pas de conditions d'accès spécialisé ni de circonstances atténuantes. Il faut très peu de connaissances ou de compétences pour pouvoir exploiter cette vulnérabilité.	Pas besoin de s'authentifier avant de pouvoir exploiter cette vulnérabilité
SQL Injection	Plusieurs vulnérabilités d'injection SQL dans Postfix Admin (alias postfixadmin) antérieures à 2.3.5 permettent aux utilisateurs authentifiés distants d'exécuter des commandes SQL arbitraires via: (1) le paramètre pw de la fonction pacrypt, lorsque mysql_encrypt est configuré ou (2) des vecteurs non spécifiés utilisés dans les fichiers de sauvegarde générés par backup.php.	<b>Sur la confidentialité</b> : Partiel (la divulgation d'informations est considérable.) <b>Sur l'intégrité</b> : Partiel (La modification de certains fichiers système ou de certaines informations est possible, mais l'attaquant n'a aucun contrôle sur ce qui peut être modifié, ou l'étendue de ce qu'il peut affecter est limitée.) <b>Sur la disponibilité</b> : partiel (la performance des ressources est réduite ou la disponibilité des ressources est interrompue.)	<b>Faible</b> : il n'existe pas de conditions d'accès spécialisé ni de circonstances atténuantes. Il faut très peu de connaissances ou de compétences pour pouvoir exploiter cette vulnérabilité.	Cette vulnérabilité nécessite la connexion d'un attaquant au système (par exemple, via une ligne de commande, une session de bureau à distance ou une interface Web).

FIGURE 4 – synthèse des menaces et vulnérabilités de Postfix

Deny of Service	Le serveur SMTP dans Postfix avant les versions 2.5.13, 2.6.x avant 2.6.10, 2.7.x avant 2.7.4 et 2.8.x avant 2.8.3, lorsque certaines méthodes d'authentification Cyrus SASL sont activées, ne gère pas bien l'échec de l'authentification du client, ce qui permet à des attaquants distants de provoquer un déni de service (corruption de la mémoire et blocage du daemon) ou éventuellement d'exécuter du code arbitraire via une commande AUTH invalide avec une méthode suivie d'une commande AUTH avec une méthode différente.	<b>Sur la confidentialité</b> : Partiel (la divulgation d'informations est considérable.) <b>Sur l'intégrité</b> : Partiel (La modification de certains fichiers système ou de certaines informations est possible, mais l'attaquant n'a aucun contrôle sur ce qui peut être modifié, ou l'étendue de ce qu'il peut affecter est limitée.) <b>Sur la disponibilité</b> : partiel (la performance des ressources est réduite ou la disponibilité des ressources est interrompue.)	<b>Moyenne</b> : Les conditions d'accès sont quelque peu spécialisées. Certaines conditions préalables doivent être satisfaites pour pouvoir être exploitées	Pas besoin de s'authentifier avant de pouvoir exploiter cette vulnérabilité
Man In The Middle Attack	L'implémentation de STARTTLS dans Postfix 2.4.x avant 2.4.16, 2.5.x avant 2.5.12, 2.6.x avant 2.6.9 et 2.7.x avant 2.7.3 ne restreint pas correctement la mise en mémoire tampon des Input / Output, qui permet aux attaquants d'insérer des commandes dans des sessions SMTP chiffrées en envoyant une commande cleartext traitée après la mise en place du protocole TLS, liée à une attaque par "injection de commande en texte clair".	<b>Sur la confidentialité</b> : Partiel (la divulgation d'informations est considérable.) <b>Sur l'intégrité</b> : Partiel (La modification de certains fichiers système ou de certaines informations est possible, mais l'attaquant n'a aucun contrôle sur ce qui peut être modifié, ou l'étendue de ce qu'il peut affecter est limitée.) <b>Sur la disponibilité</b> : partiel (la performance des ressources est réduite ou la disponibilité des ressources est interrompue.)	<b>Moyenne</b> : Les conditions d'accès sont quelque peu spécialisées. Certaines conditions préalables doivent être satisfaites pour pouvoir être exploitées	Pas besoin de s'authentifier avant de pouvoir exploiter cette vulnérabilité
Symlink Attack	Le script postfix.postinst du paquet postfix 2.5.5 de Debian GNU / Linux et Ubuntu accorde à l'utilisateur postfix un accès en écriture à	<b>Sur la confidentialité</b> : Complet (il y a divulgation totale des informations, ce qui entraîne la révélation de tous les fichiers	<b>Moyenne</b> : Les conditions d'accès sont quelque peu spécialisées. Certaines	Pas besoin de s'authentifier avant de pouvoir exploiter cette vulnérabilité

FIGURE 5 – [3] synthèse des menaces et vulnérabilités de Postfix (Suite)

	/var/spool/postfix/pid, ce qui peut permettre aux utilisateurs locaux de mener des attaques de type lien symbolique qui écrasent des fichiers de façon arbitraire	système.) <b>Sur l'intégrité</b> : Complet (L'intégrité du système est totalement compromise. La protection du système est totalement perdue, ce qui compromet l'ensemble du système.) <b>Sur la disponibilité</b> : Complet (la ressource affectée est complètement arrêtée. L'attaquant peut rendre la ressource complètement indisponible.)	conditions préalables doivent être satisfaites pour pouvoir être exploitées	
--	---	--	---	--

FIGURE 6 – synthèse des menaces et vulnérabilités de Postfix (Fin)

Postfix est un serveur dont le fonctionnement est également basé sur les services DNS (Domain Name Service). Parfois, il peut s'avérer que le bon fonctionnement de Postfix soit perturbé par la mauvaise configuration des zones DNS. Le cas le plus courant c'est la mauvaise ou non configuration du **rDNS (Reverse DNS)**. Actuellement, le serveur mail de l'IFI souffre de ce mal. En effet le service DNS permet la résolution de nom en adresse IP. Le Reverse DNS permet la résolution d'une adresse IP en nom de domaine. Le niveau de sécurité dans les services ayant augmenté de nos jours, une politique de sécurité est mise en place au

niveau des grandes entités consistant à bloquer les mails provenant de serveur dont le Reverse DNS n'est pas fonctionnel. Cet état de chose entraîne une perte des messages, ou des messages qui n'atteignent pas le destinataire et vice versa.

## **2.3 Les outils d'analyse de vulnérabilité et d'audit de sécurité**

### **2.3.1 Lynis**

Lynis est un utilitaire Open Source sous licence GPLv3 permettant d'analyser la configuration d'un serveur ainsi que les services qu'il héberge en effectuant un grand nombre de tests. A l'issue de l'analyse, un rapport est généré et permet aux administrateurs système de consulter les éventuelles failles de sécurité à résoudre pour garantir la sécurité du serveur [2].

Lynis dispose d'un module capable de relever les dysfonctionnements ou les mauvaises configurations de serveurs postfix. Ceci allant de la faiblesse de l'authentification jusqu'à la gestion de la mémoire.

### **2.3.2 Nessus**

Nessus est un outil de sécurité informatique. Il signale les faiblesses potentielles ou avérées sur les machines testées.[4] Ceci inclut, entre autres :

- les services vulnérables à des attaques permettant la prise de contrôle de la machine, l'accès à des informations sensibles (lecture de fichiers confidentiels par exemple), des dénis de service...
- les fautes de configuration (relais de messagerie ouvert par exemple)
- les patches de sécurité non appliqués, que les failles corrigées soient exploitables ou non dans la configuration testée
- les mots de passe par défaut, quelques mots de passe communs, et l'absence de mots de passe sur certains comptes systèmes. Nessus peut aussi appeler le programme externe Hydra (de) pour attaquer les mots de passe à l'aide d'un dictionnaire.
- les services jugés faibles (on suggère par exemple de remplacer Telnet par SSH) les dénis de service contre la pile TCP/IP

### **2.3.3 Email Security Grader [9]**

Email Security Grader est un outils en ligne qui permet de tester le niveau de sécurité des serveurs mail. Il permet de faire les types de test suivant :

- MX Connection Test
- Reverse DNS Test
- SMTP Plain Test Authentication Test
- DNSBL Verification Test

### **2.3.4 Check TLS [10]**

C'est un outils qui permet de tester la version de TLS implémentée sur le serveur mail. TLS c'est le protocole Transport Layer Secure. Ce protocole permet

d'assurer la sécurité des transmissions à travers l'encryption des données transmises.

### 2.3.5 Spamassassin

C'est un programme en perl développé par la fondation Apache. Ce programme regroupe plusieurs méthodes de détection de spams telles que :

- DNSBL (DNS BlackList) : blocage d'adresses IP (voir RBL plus loin) par interrogation de DNS.
- SURBL (URL BlackList) : idem mais par interrogation d'URI.
- Hashcash : Système en DoS se basant sur la consommation CPU utilisée par l'émetteur lors de l'envoi de mail.

D'autres sont disponibles via l'installation et l'activation de plug'ins.

De plus, SpamAssassin, propose la détection de spams via l'application de filtres bayesiens permettant de différencier les spams des hams (les « pas spams »).

### 2.3.6 ClamAV

ClamAV est un antimalware généralement utilisé pour la sécurité des serveurs. Il présente les avantages suivants :

- **Haute performance** : prise en charge d'un daemon de scan multithread, et utilitaires de scan de fichiers et de mises à jour de signatures
- **Versatile** : supporte plusieurs formats de fichiers
- **Open Source**

## 3 Proposition de solution

### 3.1 Ébauche de Solution

Suite à nos différentes analyses et études à propos du sujet, nous en sommes arrivés à une solution qui consisterait à protéger le serveur mail des attaques visant à mettre à mal la confidentialité, l'intégrité, la disponibilité du serveur et des utilisateurs. Mais plus précisément le protéger des attaques d'accès, des attaques de types sql injection, des attaques de type DoS, des menaces de spamming, de phishing et de malwares.

Tout ceci permettra de garantir un environnement sécurisé autour du serveur et des services qu'il fournit.

### 3.2 Démarche de Réalisation de la Solution, Planification et Outils

Le tableau ci dessous présente les différentes tâches à réaliser et la planification.

Tâches	Etapes	Durée Estimée
Audit du Système et Production de Rapport	<ul style="list-style-type: none"> <li>- Etablissement de la Checklist</li> <li>- Scan du serveur (Online, Offline)</li> <li>- Vérification des différents points de la checklist</li> <li>- Rédaction du rapport et du TPA</li> </ul>	<b>02 semaines</b>
Mise en œuvre du TPA et Test Préliminaire	<ul style="list-style-type: none"> <li>- Installation dans un environnement de Test du serveur et mise en œuvre de toutes les recommandations (Solutions antispam, antimalware, solutions de chiffrement et de gestion de trafic, etc.)</li> <li>- Test de fonctionnement et de résistance</li> </ul>	<b>01 mois</b>
Déploiement	<ul style="list-style-type: none"> <li>- Installation et mise en œuvre de la configuration testée sur un serveur proprement dit</li> <li>- Déploiement et Test en environnement réel</li> </ul>	<b>01 mois</b>
Post Déploiement	<ul style="list-style-type: none"> <li>- Rédaction de politique de sécurité</li> <li>- Rédaction référentiel de Bonnes pratiques</li> </ul>	<b>03 semaines</b>

FIGURE 7 – Tâches à réaliser et planification

Catégorie	Outils	Fonctionnalités
OffLine Scanner	Lynis	<ul style="list-style-type: none"> <li>- Scan de la sécurité et du fonctionnement de Postfix</li> <li>- Prouction de Rapport détaillé</li> </ul>
Online Scanner	Email Security Grader	- Test online de: <b>MX Connection, Reverse DNS, DNSBL, Open Relay and Email Format, SMTP Plain Text Authentication</b>
TLS Encryption Scanner	CheckTLS	<ul style="list-style-type: none"> <li>- Test du niveau d'encryption SSL/TLS</li> <li>- Assistance à la mise au Norme</li> </ul>
Certificate Manager	Let's Encrypt	- Création et gestion de Certicat de sécurité
Content Filter	Spamassassin	<ul style="list-style-type: none"> <li>- Détection de Spamassassin</li> <li>- DNSBL</li> <li>- SURLBL</li> </ul>
Anti Malware	ClamAV	<ul style="list-style-type: none"> <li>- Command-line scanner</li> <li>- Plusieurs Mises à jour de la base Virale par jour</li> <li>- Supporte plusieurs formats de fichiers: Zip, RAR, Dmg, Tar, Gzip, Bzip2, OLE2, Cabinet, CHM, BinHex, SIS</li> <li>- Intègre module de scan des fichiers exécutables et des formats courants de documents</li> </ul>

FIGURE 8 – Les outils à utiliser pour réaliser le travail

## 4 Implémentation de la solution proposée

Dans l'objectif de sécuriser le serveur mail de l'IFI, il est impératif pour nous d'effectuer une analyse des différentes vulnérabilités du système afin de les corri-



ger. Pour ce faire notre démarche consiste à dans un premier temps installer un serveur virtuel ensuite le tester, appliquer les différents correctifs pour ensuite évaluer par la suite le serveur réel. Dans ce document nous allons présenter les différentes étapes par lesquelles nous allons passer pour aboutir aux résultats attendus.

#### 4.1 Proposition d'architecture

L'architecture que nous allons proposer pour la réalisation de ce travail est la suivante :

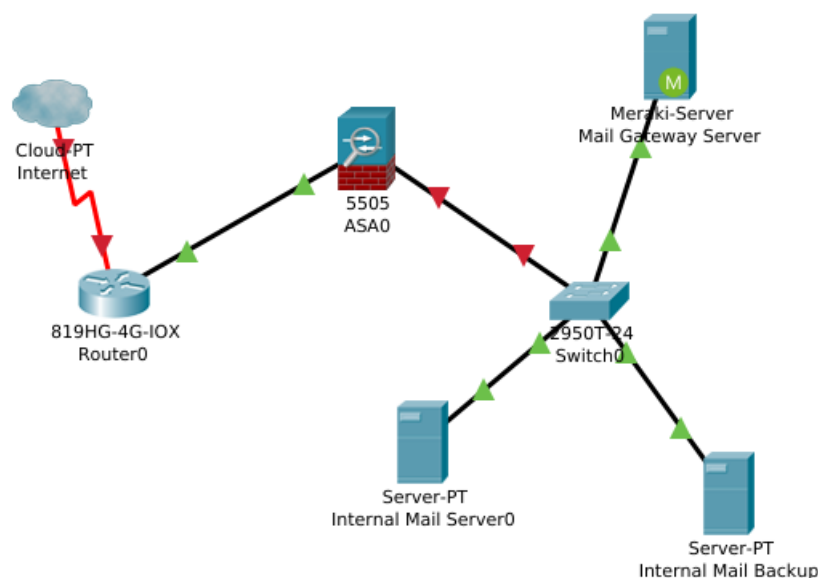


FIGURE 9 – Architecture basée sur le modèle recommandé dans le livre : **Postfix : the definitive Guide**

En effet selon ce schéma, nous aurons trois (03) différents serveurs qui vont interagir afin de permettre le bon fonctionnement de la messagerie. Il s'agit entre autre de :

- Mail Gateway Serveur : jouant le rôle de relais et de passerelle, ce serveur relié au firewall reçoit les messages provenant de l'internet et est chargé de les acheminer vers les serveurs mails à l'intérieur du réseau.
- Internal Mail Serveur0 : ce serveur est censé jouer le rôle de serveur mail interne gérant la messagerie interne de sorte que les utilisateurs à l'interne soient plus ou moins autonomes.
- Internal Mail Backup : Ce serveur est là pour servir de Backup du serveur Internal Mail Server0. Ceci permettra de balancer les charges et de

réduire le Recovery Time Object (RTO) e cas de désagrément ou d'in-disponibilité du serveur serveur principal.

Ce modèle d'architecture a pour avantage la séparation maximale des tâches, la tolérance aux pannes mais aussi la capacité à ne pas laisser une grande quantité d'information accessible à travers internet en ce sens que, le serveur visible sur internet est le Mail Gateway qui est juste chargé de transférer les messages au serveur interne.

## 4.2 Mise en oeuvre de l'architecture

Afin de mettre en oeuvre cette architecture ce faire, nous avons utilisé l'outil de virtualisation Vmware pour créer des machines virtuelles tournant sous Debian 10.

### 4.2.1 Installation du Système d'exploitation et des services de base

Pour installer le système d'installation, nous avons :

- lancé l'outils VMWare puis y avons créé une machine virtuelle
- nous avons ensuite démarré la machine virtuelle à partir d'une image iso de Debian 10 pour pouvoir démarrer l'installation.
- nous avons installé Debian en mode textuel avec quelques services de base, sans utiliser de miroir sur le réseau
- nous avons ensuite lancé les commandes :  
***apt-get update***  
***apt-get upgrade***

pour pouvoir effectuer la mise à jour du système installé.

### 4.2.2 Test de niveau de sécurité du Système à son état par défaut

Afin d'évaluer le niveau de sécurité du système par défaut, nous avons utilisé quelques outils :

- Un script bash que nous avons édité
- L'outils de scan de sécurité des serveurs Unix *Lynis*

### 4.2.3 Script d'audit

Le script bash que nous avons édité est une succession de plusieurs commandes de bases permettant d'obtenir des informations sur : la version de linux, les différents utilisateurs connectés, les interfaces réseaux, les services en fonction, les règles de sécurités, la consommation de mémoire, etc...

```

silas@mailserv: ~
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
GNU nano 3.2      LinuxAudit.sh

echo "#####"
echo "Bienvenue dans ce script d'audit des SE UNIX:"
echo "#####"
echo
echo "Le Script se chargera de la collecte de toutes les informations $
echo "Cette checklist est utile dans le processus de renforcement de l$
echo "Note: Il a ete teste pour Debian Linux Distro:"
echo
sleep 3
echo "#####"
echo "OK...$HOSTNAME..on y va...Veuillez patienter la fin:"
echo
sleep 3
echo "Script Starts ;)"
START=$(date +%s)
echo -e "\e[0;33m 1. Linux Kernel Information///// \e[0m"
uname -a
echo
echo "#####"
echo -e "\e[0;33m 2. Current User and ID information///// \e[0m"
whoami
echo
id
echo
echo "#####"
echo -e "\e[0;33m 3. Linux Distribution Information///// \e[0m"
lsb_release -a
echo
echo "#####"
echo -e "\e[0;33m 4. List Current Logged In Users///// \e[0m"

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell

```

FIGURE 10 – Fichier LinuxAudit.sh

```
silas@mailserv: ~
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
GNU nano 3.2  LinuxAudit.sh

echo
echo "#####"
echo
echo -e "\e[0;33m 5. $HOSTNAME uptime Information///// \e[0m"
uptime
echo
echo "#####"
echo
echo -e "\e[0;33m 6. Running Services///// \e[0m"
service --status-all | grep +
echo
echo "#####"
echo
echo -e "\e[0;33m 7. Active internet connections and open ports///// \e[0m"
netstat -natp
echo
echo "#####"
echo
echo -e "\e[0;33m 8. Check Available Space///// \e[0m"
df
echo
echo "#####"
echo
echo -e "\e[0;33m 9. Check Memory///// \e[0m"
free -h
echo
echo "#####"
echo
echo -e "\e[0;33m 10. History (Commands)///// \e[0m"
history
echo
echo "#####"
echo

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell
```

FIGURE 11 – Fichier LinuxAudit.sh

Ce script est entre autre composé des commandes :

- Pour afficher les informations du Kernel :  
*uname -a*
- Pour afficher les informations de l'utilisateur actuellement connecté et de l'ID  
*whoami*  
*id*
- Pour afficher les informations sur la distribution de Linux

- lsb\_release -a*
- Pour afficher la Liste des utilisateurs actuellement connectés  
*w*
- Pour afficher les informations sur le temps de démarrage de la machine  
*uptime*
- Pour afficher les services qui tournent sur la machine  
*service --status-all --grep "+"*
- Pour afficher les connexions internet actives et les ports ouverts  
*netstat -natp*
- Pour Vérifier l'espace disque disponible  
*df*
- Pour vérifier la mémoire  
*free -h*
- Pour afficher l'historique des commandes  
*history*
- Pour afficher les interfaces réseau  
*ip a*
- Pour afficher les information de iptables  
*iptables -L -n -v*
- Pour vérifier les Processus en cours  
*ps -a*
- Pour vérifier la Configuration de SSH  
*cat /etc/ssh/sshd\_config*
- Pour afficher la liste de tous les packages installés  
*apt-cache pkgnames*
- Pour afficher les paramètres de configuration Réseau  
*cat /etc/sysctl.conf*
- Pour afficher la polices des mots de passe  
*cat /etc/pam.d/common-password*
- Pour vérifier le fichier de Source List  
*cat /etc/apt/sources.list*
- Pour vérifier les dépendances  
*apt-get check*
- Pour afficher le MOTD banner message  
*cat /etc/motd*
- Pour afficher la Liste des "usernames"  
*cut -d : -f1 /etc/passwd*
- Pour afficher la table de routage  
*route -n*

## 4.3 Eléments de contrôle

### 4.3.1 Objectifs de contrôle

Les contrôles sont des mécanismes utilisés pour évaluer, mitiger les risques en éliminant les vulnérabilités dans le système. Chaque contrôle apparaissant dans la checklist plus bas à un objectif spécifique dans le processus d'amélioration du niveau de sécurité et de résilience du serveur mail et donc du système d'information. Ceci en garantissant certains principes tels que la défense en profondeur,

le principe du moindre privilège et de la séparation des tâches.

TABLE 1 – Résumé des objectifs de contrôle

Catégorie de contrôle	Objectif général
Contrôle du Processus d’audit	Assurer la réussite de l’audit et valider le périmètre d’audit, l’évaluation des risques et les différents contrôles à effectuer
Contrôle de l’environnement	Assurer que le serveur fonctionne dans un environnement sans interruption et dont l’accès physique est sécurisé
Contrôle du Réseau	Protéger le serveur des trafics réseaux peu utiles, assurer les meilleures performances réseau au serveur
Contrôle du SE Linux	Assurer que le système d’exploitation protège le serveur des trafics non désirés, assurer que le serveur peut se défendre en cas de défaillance des autres mécanismes de sécurité
Contrôle du service Postfix	Assurer que postfix dans son fonctionnement permet de sécuriser les mails et est configuré de sorte à résister à certaines attaques qui ont pu passer le réseau et le système d’exploitation
Contrôle Opérationnel	Assurer que des polices et procédures Opérationnelles sont en place afin de garantir la confiance et la sécurité dans le fonctionnement des services.

#### 4.3.2 Définition de la liste de contrôle

Plus haut nous avons classé dans le tableau 1, les objectifs que nous voulons atteindre en faisant les contrôles. Nous allons présenter dans le tableau qui suit, les différents éléments qu’il serait judicieux de contrôler dans le cadre de la réalisation de ce travail.

TABLE 2 – Listing des éléments de contrôle

Etapas	Contrôle
1	Vérifier la réaction du serveur au redémarrage
2	Contrôle des versions de linux et de postfix
3	Déterminer si l'accès physique est bien contrôlé
4	Déterminer si l'environnement physique est adéquat
5	Test des entrées réseau avec Nmap
6	Test de contrôle des trafics indésirables avec Nessus
7	Vérification du bon fonctionnement du service DNS
8	Vérification du fonctionnement du logging au niveau du serveur mail
9	Vérification des Patches du système d'exploitation
10	Vérification de la configuration du firewall sur le serveur mail
11	Vérification du fonctionnement uniquement des services nécessaires sur le serveur
12	Vérification de l'installation et de la configuratuion de OpenSSH
13	Vérification de la protection effective de tous les comptes d'utilisateurs par mot de passe avec complexité
14	Vérification de la version de OpenSSL
15	Vérification de la version de TLS
16	Vérification du fonctionnement effectif des restrictions implémentées dans Postfix
17	Vérifier que les RBL-based message sont bloqués
18	Vérification du fonctionnement du blocage des messages basé sur une analyse de header
19	Vérification des fonctionnalités antispam
20	Vérification des fonctionnalités antimalware
21	Vérification de la manière dont postfix réagit aux messages à contenus non pris en charge

#### 4.3.3 Définition des points d'amélioration

En vue de renforcer la sécurité, après les contrôles effectués et selon les standards de sécurisation des services de messagerie électronique, il nous faut améliorer un certain nombre de points se trouvant listés dans le tableau qui suit.

TABLE 3 – Tableau des Points d’Amélioration

Etapes	Contrôle
1	Choisir un mot de passe BIOS
2	Créer une partition distincte pour le répertoire des mails
3	Définir un mot de passe fort pour le compte root
4	Activer le chiffrement des mots de passe
5	Désactiver les services qui ne sont pas nécessaires
6	Désactiver Telnetd
7	Appliquer les mises à jour de sécurité du système
8	Restreindre les connexions consoles et les redémarrage système depuis la console
9	Limiter l’accès aux informations d’autres utilisateurs
10	Configuration et Gestion efficace des Logs
11	Protéger le système contre le dépassement de tampon
12	Installation et Configuration d’antivirus
13	Configuration de Pare feu
14	Configuration de IDS
15	Sécuriser le service de courrier au moyen du protocole TLS
16	Garantir la délivrance d’un certificat numérique à tous les utilisateurs
17	Assurer la Confidentialité Persistante (Perfect Forward Secrecy)
18	En cas d’accès Web, Garantir la sécurité en utilisant le protocole HTTPS
19	Protection contre les attaques XSS et les Session Hijacking

#### 4.4 Mise en oeuvre des points d’amélioration

En vue de Réaliser les éléments cités dans le tableau précédent, nous allons suivre les différentes étapes de **4.5 à 4.11**

#### 4.5 Définition d’une politique de mot de passe

Pour ce faire, il nous faut configurer le PAM (Pluggable Authentication Module). Pour ce faire, nous avons

- installé le module PAM en exécutant la commande :  
`sudo apt-get install libpam-cracklib`
- nous avons ensuite édité le fichier `/etc/pam.d/common-password`  
 Nous avons donné la valeur de taille minimale d’un mot de passe et le nombre de caractères qui doivent changer lorsqu’on change un mot de passe.



```
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
password      requisite          pam_cracklib.so minlen=8 difok=3
password      [success=1 default=ignore] pam_unix.so obscure use_authtok try_first_pass sha512
# here's the fallback if no module succeeds
```

FIGURE 12 – Modification du fichier `/etc/pam.d/common-password`

## 4.6 Désactiver Telnetd

En fonction des vulnérabilités que présente le protocole telnet, l'idéal est de le désactiver afin d'empêcher son utilisation ultérieure. Pour cela, nous avons exécuté la commande :

```
sudo apt-get remove telnetd
```

En plus de cela nous avons écrit une règle de firewall iptables pour bloquer toute tentative de communication vers le port 23 par défaut utilisé pour Telnet

```
iptables -A INPUT -p tcp -dport 23 -j REJECT
```

## 4.7 Restreindre le redémarrage console

En effet, les systèmes UNIX permettent un redémarrage grâce à la combinaison *Ctrl+Alt+Sup* au clavier. Ceci dit, toute personne ayant accès physique au serveur pourrait facilement lancé un redémarrage. Aussi, par un accès réseau, un utilisateur pourrait par fausse manipulation envoyer cette commande, ce qui provoquerait un redémarrage du serveur. Pour empêcher cela, nous devons désactiver le redémarrage par la commande *Ctrl+Alt+Sup*.

Pour ce faire, nous devons :

- supprimer le fichier permettant l'exécution de cette commande :  
*sudo rm /lib/systemd/system/ctrl-alt-del.target*
- ensuite nous allons créer un lien symoblique qui pointe sur nul afin de ne pas provoquer un crash du système lorsqu'on appuiera sur la combinaison *Ctrl+Alt+Sup* :  
*sudo ln -s /dev/null /lib/systemd/system/ctrl-alt-del.target*
- nous allons ensuite recharger le daemon pour qu'il prenne en compte nos modifications  
*sudo systemctl daemon-reload*

## 4.8 Sécurité antispam

Afin de garantir la sécurité, il faudrait penser à un mécanisme de filtrage anti-spam. Pour ce faire nous allons commencer par installer **Spamassassin** qui est un outil permettant de filtrer les mails et détecter les spam.

Afin de l'installer et de le configurer :

- nous allons installer l'outil avec la commande  
*sudo apt-get install spamassassin*
  - ensuite nous allons créer l'utilisateur et le groupe qui permettront de faire fonctionner l'outil  
*sudo adduser spamassassin*  
*sudo addgroupe spamassassin*
- Il nous faut ensuite éditer la configuration du daemon */etc/default/spamassassin* comme ci-dessous :
- ```
ENABLED=1
OPTIONS="-username spamassassin -nouser-config -max-children 2 -
helper-home-dir $SAHOME -socketowner=spamassassin -socketgroup=spamassassin
-socketmode=0660"
PIDFILE="/var/run/spamassassin/spamd.pid"
CRON=1
```
- Nous allons à présent ajouter des modules et les configurer :
- Module Amavisd  
Pour ça nous allons commencer par éditer le fichier */etc/postfix/master.cf*

```
127.0.0.1:10025 inet      n      -      y      -      -      smtpd
        -o content_filter=
        -o local_recipient_maps=
        -o relay_recipient_maps=
        -o smtpd_restriction_classes=
        -o smtpd_helo_restrictions=
        -o smtpd_sender_restrictions=
        -o smtpd_recipient_restrictions=permit_mynetworks,reject
        -o mynetworks=127.0.0.0/8
        -o strict_rfc821_envelopes=yes
```

FIGURE 13 – Modification du fichier */etc/postfix/master.cf* pour activer un service de listening pour le content filter

#### 4.8.1 Présentation des services SPF et DKIM

La sécurisation des serveurs de mails (MTA) et la lutte contre l'usurpation d'email (spoofing en anglais), le phishing et SPAM n'est pas forcément chose facile. Une des problématiques étant que le protocole Simple Mail Transfer Protocol (SMTP) utilisé pour le transfert du courrier électronique sur Internet ne prévoit pas de mécanisme de vérification de l'expéditeur. C'est-à-dire qu'il est facile d'envoyer un courrier avec une adresse d'expéditeur factice, voire usurpée.

Afin de limiter ces problèmes, des normes sont apparues pour tenter de limiter ces usurpations.

##### — DKIM

DKIM fonctionne par signature cryptographique du corps du message ou d'une partie de celui-ci et d'une partie de ses en-têtes DKIM et d'ajouter une signature à l'en-tête de chaque message envoyé. Cette signature est propre au domaine, elle est générée par une clé privée. La clé publique qui y correspond est ajoutée au domaine, dans un enregistrement DNS.

— **SPF**

Le protocole SPF vise à réduire les possibilités d’usurpation en publiant, dans le DNS, un enregistrement (de type TXT) indiquant quelles adresses IP sont autorisées ou interdites à envoyer du courrier pour le domaine considéré.

#### 4.8.2 Configuration de DKIM et SPF

##### SPF

Afin de configurer ce service, il nous a fallu faire la déclaration ci-après dans le fichier de zone DNS

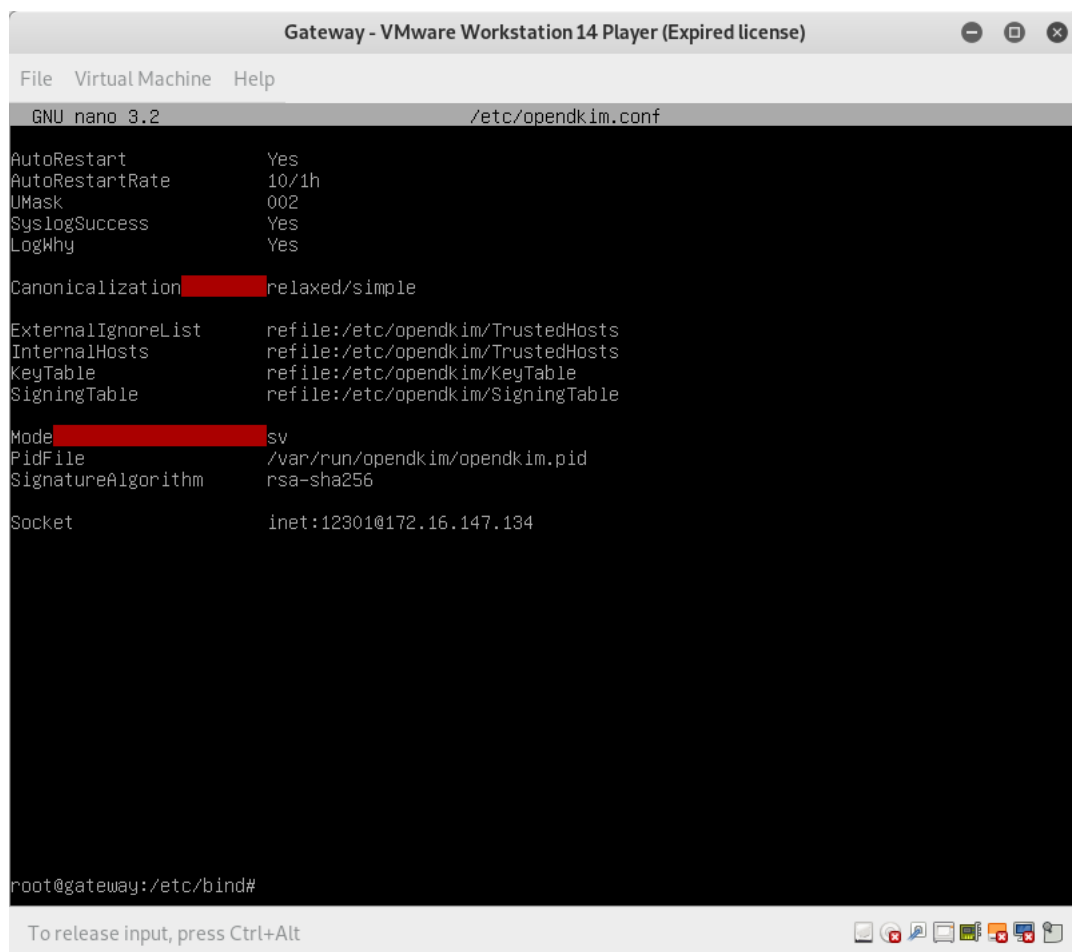
```
@ 10800 IN TXT "v=spf1 ptr :ifi.edu.local all"
```

Cette déclaration permet d’autoriser tous les serveurs mail du domaine à pouvoir envoyer un mail.

##### DKIM

Pour configurer DKIM, nous avons installé les paquets : *opendkim*, *opendkim-tools* et *spamass-milter*, après quoi nous avons édité le fichier */etc/opendkim.conf* afin de définir les différentes propriétés de filtrage comme entre autres :

- **AutoRestart** : Redémarre le filtre en cas de plantage.
- **AutoRestartRate** : Indique le ratio de redémarrage minimum et maximum. Si le nombre de redémarrage est plus rapide que ce ratio, ce dernier va être arrêté.
- **UMask** : indique les permissions et ID utilisateur/groupe.
- **Syslog**, **SyslogSuccess**, **\*LogWhy** : activé les log syslog
- **Canonicalization** : méthode canonical de signature des messages,  
**simple** : autorise aucune modification  
**relaxed** : autorise des modifications mineures comme changer les espaces.  
**relaxed/simple** : L’en-tête du mail utilisera la méthode relaxed et le corps du message utilisera la méthode simple.
- **ExternalIgnoreList** : la liste des des hôtes par lequel les mails peuvent passer sans signatures.
- **InternalHosts** : liste des hôtes à ne pas vérifier et signer : signature sans vérification.
- **KeyTable** : Le chemin des tables de clés de signatures.
- **SigningTable** : Liste des tables de signatures pour lesquels seront basés les champs from des mails.
- **PidFile** : Le fichier PID (process identification number)
- **SignatureAlgorithm** : indique l’algorithme de signatures à utiliser
- **UserID** : Le UserID du processus opendkim
- **Socket** : Le socket d’écoute de opendkim. Postfix va envoyer les messages à opendkim pour les vérification et signatures à travers ce socket.



```
Gateway - VMware Workstation 14 Player (Expired license)
File Virtual Machine Help
GNU nano 3.2 /etc/openssl.conf
AutoRestart Yes
AutoRestartRate 10/1h
UMask 002
SyslogSuccess Yes
LogWhy Yes
Canonicalization relaxed/simple
ExternalIgnoreList refile:/etc/openssl/TrustedHosts
InternalHosts refile:/etc/openssl/TrustedHosts
KeyTable refile:/etc/openssl/KeyTable
SigningTable refile:/etc/openssl/SigningTable
Mode sv
PidFile /var/run/openssl/openssl.pid
SignatureAlgorithm rsa-sha256
Socket inet:12301@172.16.147.134
root@gateway:/etc/openssl#
```

FIGURE 14 – Fichier /etc/openssl.conf

Après quoi nous avons édité la liste des hôtes de confiance de notre domaine et généré les clés publiques et privées. Une fois cette étape passée, nous avons effectué la déclaration dans le fichier de zon DNS

## 4.9 Installation d’antimalware ClamAv

Pour installer l’antivirus ClamAV, nous avons eu à :

- installer les paquets : *clamav-daemon*, *clamav*, *clamsmtp*
- après avoir installé ces paquets, nous allons configurer le fichier */etc/clamsmtpd.conf* afin d’autoriser l’utilisation du header de clamav.
- nous avons ensuite configuré postfix pour interagir avec l’antivirus en modifiant respectivement les fichiers */etc/postfix/main.cf* et */etc/postfix/master.cf*

## 4.10 Configuration de TLS

Le protocole TLS est l'un des plus prisés en terme de sécurisation de la messagerie électronique. En effet, TLS permet de sécuriser le transfert des messages par un mécanisme. Pour le configurer, il est nécessaire de disposer d'une paire de clés générée par un tiers de confiance. Dans le cas de notre travail, ne pouvant souscrire à ce stade auprès d'une autorité publique de certification, nous avons dû créer une autorité de certification locale. Ceci est dû au fait que notre serveur n'est pas encore accessible sur internet.

### 4.10.1 Création d'une autorité de Certification

En vue de garantir la confiance entre les différents clients de notre système et en vue de garantir un échange chiffré des données à travers SSL et à travers TLS, il est important de disposer de certificats numériques. Pour ce faire nous allons installer une autorité de certification qui se chargera de certifier les différents clients.

#### Démarche

- En premier nous devons préparer notre environnement.

```
mkdir -p Autorite/certs
mkdir -m 700 -p Autorite/AC/private,newcerts,crl
chmod 700 Autorite/AC/private
touch Autorite/AC/index.txt
echo 01 > AC/serial
cp /etc/ssl/openssl.cnf Autorite/
```

Ces commandes nous permettent de créer notre propre configuration afin de ne pas utiliser la configuration par défaut de *openssl*.

- **Personnalisation du fichiers de configuration de *openssl***

```

[ ca ]
default_ca = AC_locale

[ AC_locale ]

dir = ./AC # Where everything is kept
certs = $dir/certs # Where the issued certs are kept
crl_dir = $dir/crl # Where the issued crl are kept
database = $dir/index.txt # database index file.
#unique_subject = no # Set to 'no' to allow creation of
# several certificates with same subject.
new_certs_dir = $dir/newcerts # default place for new certs.

certificate = $dir/ca-local.crt # The CA certificate
serial = $dir/serial # The current serial number
crlnumber = $dir/crlnumber # the current crl number. must be
# commented out to leave a V1 CRL
crl = $dir/ca-lille3.crl # The current CRL
private_key = $dir/private/ca.key # The private key
#RANDFILE = $dir/private/.rand # private random number file
RANDFILE = /dev/random # private random number file

x509_extensions = usr_cert # The extensions to add to the cert

name_opt = ca_default # Subject Name options
cert_opt = ca_default # Certificate field options

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crlnumber must also be commented out to leave a V1 CRL.
crl_extensions = crl_ext

default_days = 1095 # how long to certify for
default_crl_days = 30 # how long before next CRL
default_md = sha1 # which md to use.
preserve = no # keep passed DN ordering

# On utilise la politique policy_match définie ci-dessous, qui oblige à faire
# correspondre countryName, stateOrProvinceName et organizationName, nécessite

```

FIGURE 15 – Personnalisation fichier openssl.cnf

```

# correspondre countryName, stateOrProvinceName et organizationName, nécessite
# un commonName et supporte en option organizationalUnitName et emailAddress
policy = policy_match

# For the CA policy
[ policy_match ]
countryName = match
stateOrProvinceName = match
stateOrProvinceName = optional
organizationName = match
# utiliser ça si on veut être moins strict :
countryName = optional
stateOrProvinceName = optional
organizationName = optional
organizationalUnitName = optional
commonName = supplied
emailAddress = optional

[ req ]
default_bits = 1024
#default_keyfile = privkey.pem
default_keyfile = new.key
distinguished_name = req_distinguished_name
#attributes = req_attributes
x509_extensions = v3_ca # The extensions to add to the self signed cert
string_mask = nombstr
req_extensions = v3_req # The extensions to add to a certificate request

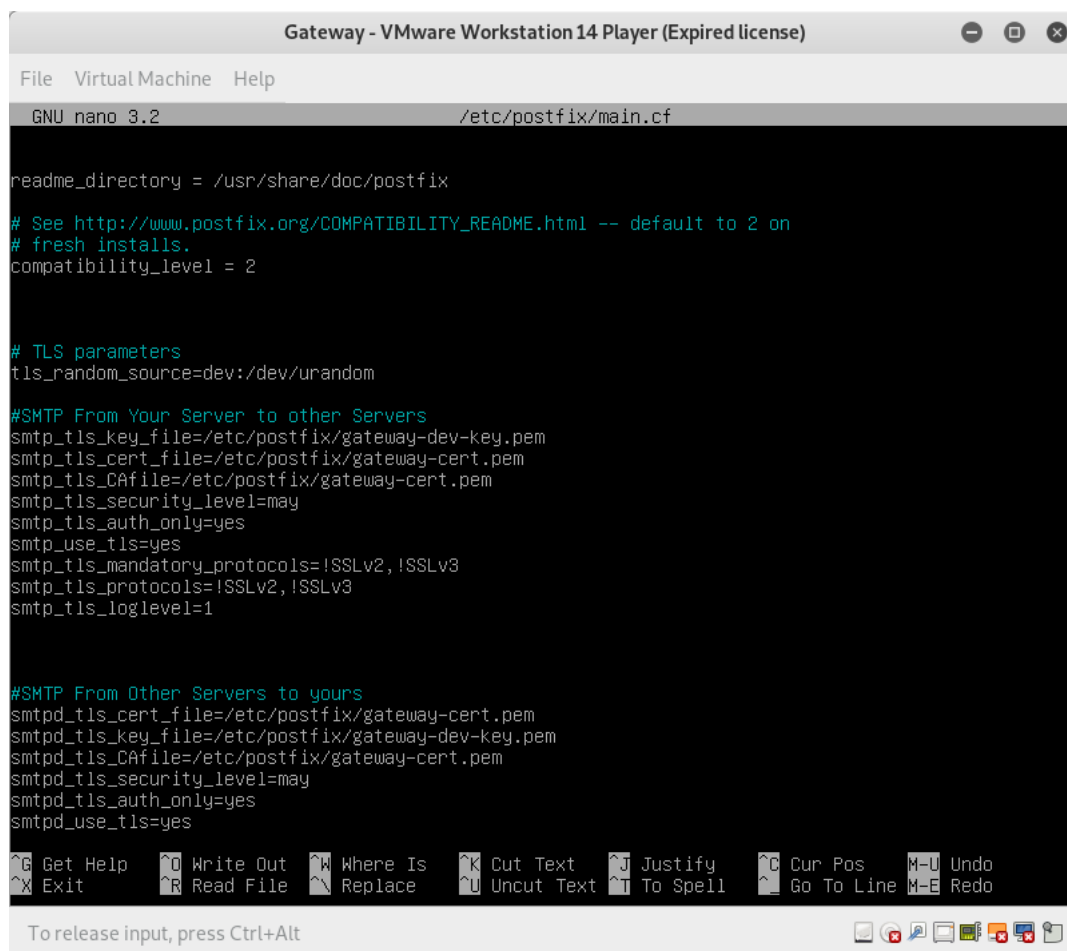
```

FIGURE 16 – Personnalisation fichier openssl.cnf

- Exportation de la configuration effectuée  
*export OPENSSL\_CONF=/etc/Autorite/openssl.cnf*
- Génération de la clé privé de l'autorité de certification  
*openssl genrsa -out /etc/Autorite/AC/private/cakey.pem 1024*
- Génération du certificat Autosigné  
*openssl req -new -x509 -key Autorite/private/cakey.pem -out Autorite/certs/cacert.pem -days 1460*

#### 4.10.2 Configuration de TLS

Pour maintenant activer TLS, il faut modifier le fichier */etc/postfix/main.cf*. Dans le fichier, nous allons d'abord autoriser l'utilisation de TLS puis définir le chemin vers la clé de chiffrement à utiliser.



The screenshot shows a window titled "Gateway - VMware Workstation 14 Player (Expired license)". Inside, the GNU nano 3.2 text editor is open, editing the file */etc/postfix/main.cf*. The file content is as follows:

```

readme_directory = /usr/share/doc/postfix

# See http://www.postfix.org/COMPATIBILITY_README.html -- default to 2 on
# fresh installs.
compatibility_level = 2

# TLS parameters
tls_random_source=dev:/dev/urandom

#SMTP From Your Server to other Servers
smtp_tls_key_file=/etc/postfix/gateway-dev-key.pem
smtp_tls_cert_file=/etc/postfix/gateway-cert.pem
smtp_tls_CAfile=/etc/postfix/gateway-cert.pem
smtp_tls_security_level=may
smtp_tls_auth_only=yes
smtp_use_tls=yes
smtp_tls_mandatory_protocols=!SSLv2, !SSLv3
smtp_tls_protocols=!SSLv2, !SSLv3
smtp_tls_loglevel=1

#SMTP From Other Servers to yours
smtpd_tls_cert_file=/etc/postfix/gateway-cert.pem
smtpd_tls_key_file=/etc/postfix/gateway-dev-key.pem
smtpd_tls_CAfile=/etc/postfix/gateway-cert.pem
smtpd_tls_security_level=may
smtpd_tls_auth_only=yes
smtpd_use_tls=yes

```

The nano editor interface includes a menu bar with options like Get Help, Write Out, Where Is, Cut Text, Justify, Cur Pos, M-U Undo, Exit, Read File, Replace, Uncut Text, To Spell, Go To Line, and M-E Redo. At the bottom, it says "To release input, press Ctrl+Alt".

FIGURE 17 – Configuration de TLS dans le fichier */etc/postfix/main.cf*

## 4.11 Configuration d'un accès à la messagerie par interfaces WEB

Afin de faciliter l'accès aux utilisateurs, nous avons configuré un client de messagerie WEB qui agira ainsi en qualité de Mail User Agent. Pour ça nous avons donc installé *Rainloop Webmail*. Cet outil fonctionne avec le serveur Web Apache 2. Après l'avoir installé, nous l'avons fait interagir avec le serveur mail par le protocole IMAP pour recevoir les messages et le protocole SMTP pour envoyer les messages.

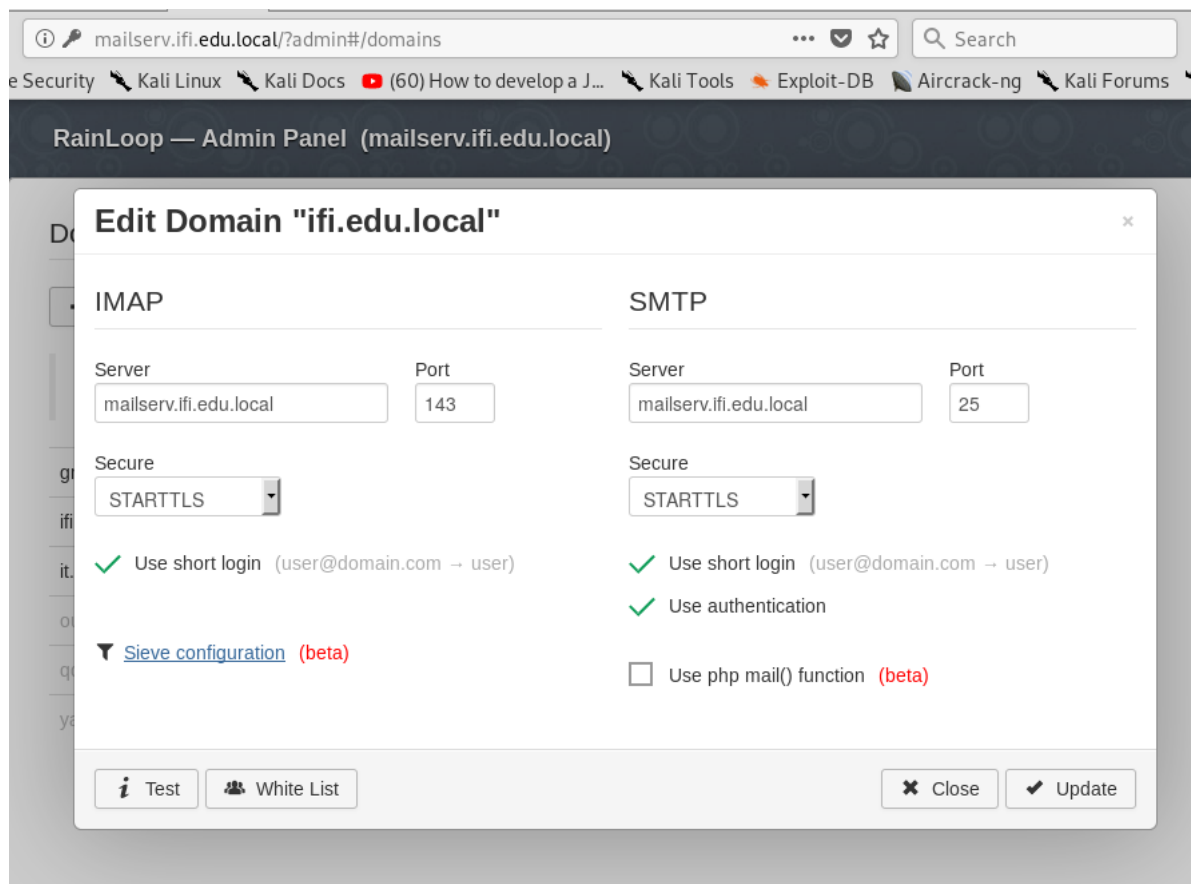


FIGURE 18 – Configuration de la Liaison entre Rainloop et le serveur Mail

## 5 Présentation des résultats

A l'issue des différents travaux effectués, nous avons obtenus les résultats suivants



## 5.1 Accès au serveur mail par Web

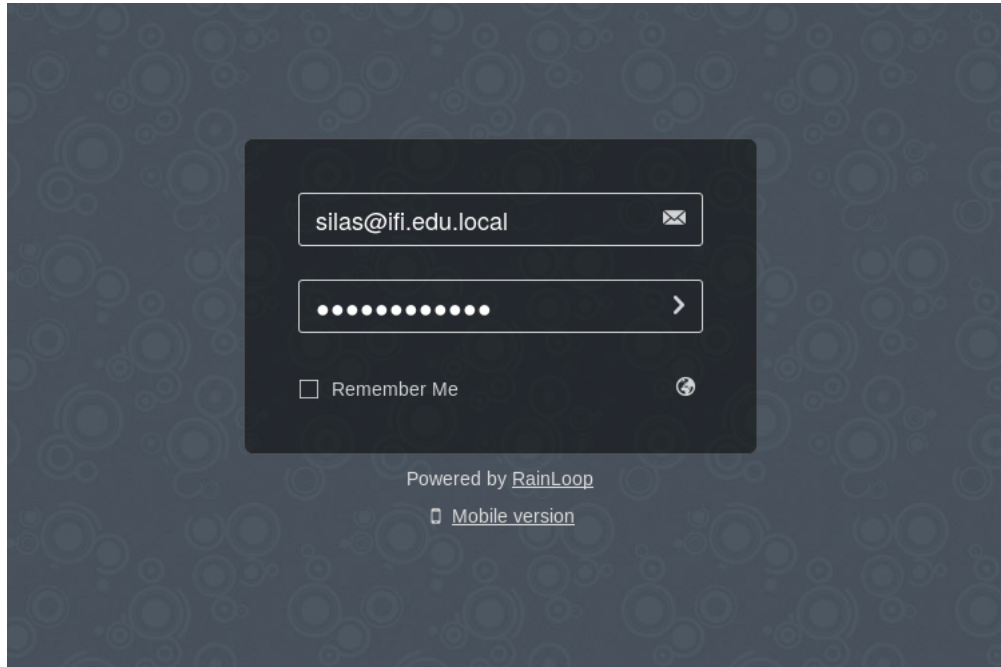


FIGURE 19 – Authentification au niveau du Webmail

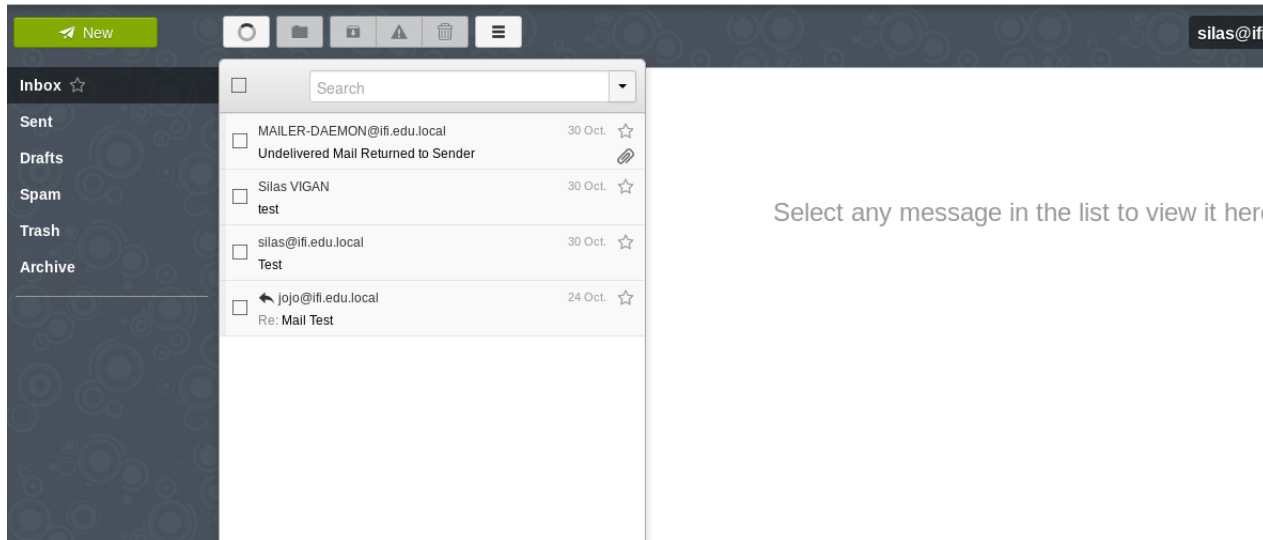


FIGURE 20 – Boite de Messagerie

## 5.2 Le protocole TLS

Nous pouvons constater que notre travail a permis de mettre en oeuvre la dernière version du protocole TLS Comparativement à la version qui est actuellement en utilisation sur le serveur mail de l'IFI.

```

root@mail: ~/Téléchargements/testssl.sh-3.0rc5
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
Start 2019-11-26 18:23:11 -->> 112.137.140.41:587 (smtp.ifi.edu.vn) <<--

rDNS (112.137.140.41): --
Service set:          STARTTLS via SMTP

Testing protocols via sockets

SSLv2      not offered (OK)
SSLv3      offered (NOT ok)
TLS 1      offered
TLS 1.1    STARTTLS handshake failed
Fixme: unexpected value around line 5029, rerun w DEBUG>=2 or --ssl-native
TLS 1.2    STARTTLS handshake failed
./testssl.sh: connect: Aucun chemin d'accès pour atteindre l'hôte cible
./testssl.sh: ligne 9925: /dev/tcp/112.137.140.41/587: Aucun chemin d'accès pour atteindre l'hôte cible
Oops: TCP connect problem

Unable to open a socket to 112.137.140.41:587. Fixme: unexpected value around line 5117, rerun w DEBUG>=2 or --ssl-native
TLS 1.3    not offered

Testing cipher categories

NULL ciphers (no encryption)          STARTTLS handshake failed
not offered (OK)
Anonymous NULL Ciphers (no authentication) offered (NOT ok)
Export ciphers (w/o ADH+NULL)         not offered (OK)
LOW: 64 Bit + DES, RC[2,4] (w/o export) offered (NOT ok)
Triple DES Ciphers / IDEA             offered (NOT ok)
Average: SEED + 128+256 Bit CBC ciphers offered
Strong encryption (AEAD ciphers)      not offered

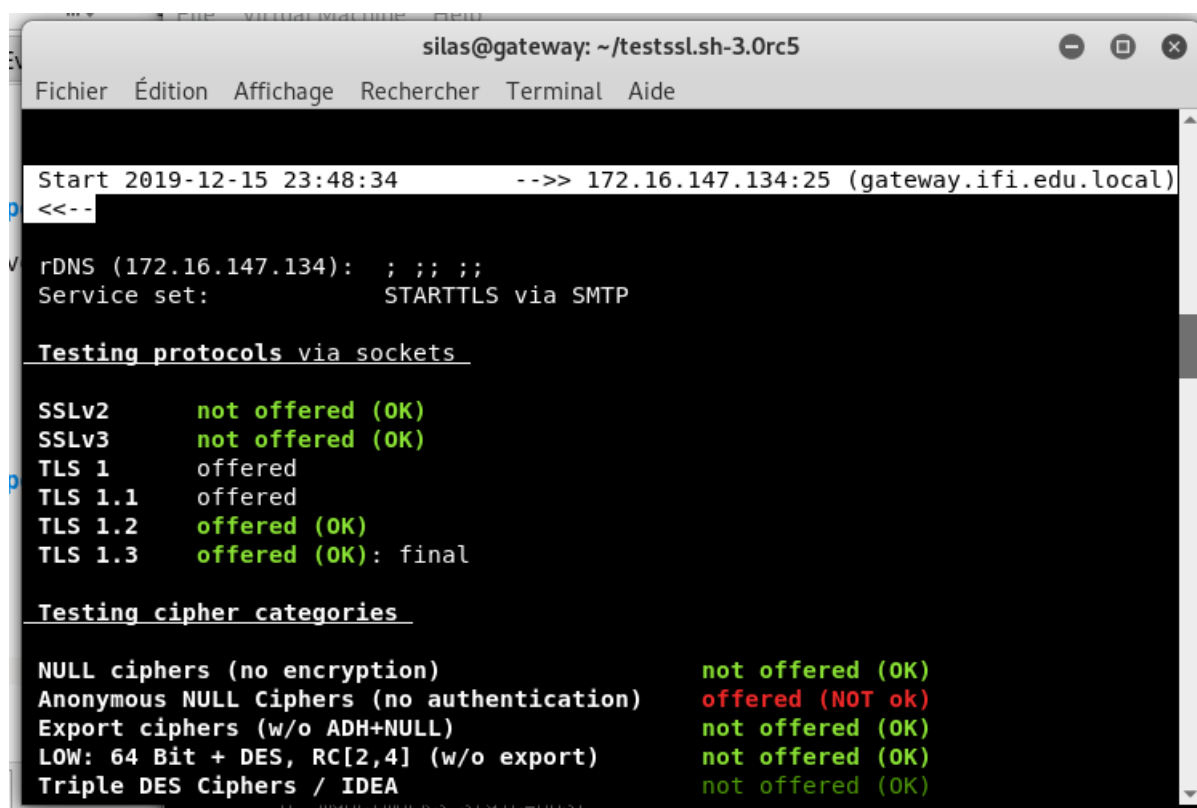
Testing robust (perfect) forward secrecy, (P)FS -- omitting Null Authentication/Encryption, 3DES, RC4

PFS is offered (OK)                   DHE-RSA-AES256-SHA DHE-RSA-AES128-SHA
DH group offered:                     Postfix (1024 bits)

Testing server preferences

```

FIGURE 21 – Test effectué sur le serveur de l'IFI avec une version obsolète de TLS



The screenshot shows a terminal window titled 'silas@gateway: ~/testssl.sh-3.0rc5'. The window contains the output of a testssl.sh script. At the top, it shows the start time '2019-12-15 23:48:34' and the target '172.16.147.134:25 (gateway.ifi.edu.local)'. Below this, it shows 'rDNS (172.16.147.134): ; ; ; ;' and 'Service set: STARTTLS via SMTP'. The next section is 'Testing protocols via sockets', which lists the following results: SSLv2 (not offered (OK)), SSLv3 (not offered (OK)), TLS 1 (offered), TLS 1.1 (offered), TLS 1.2 (offered (OK)), and TLS 1.3 (offered (OK): final). The final section is 'Testing cipher categories', which lists the following results: NULL ciphers (no encryption) (not offered (OK)), Anonymous NULL Ciphers (no authentication) (offered (NOT ok)), Export ciphers (w/o ADH+NULL) (not offered (OK)), LOW: 64 Bit + DES, RC[2,4] (w/o export) (not offered (OK)), and Triple DES Ciphers / IDEA (not offered (OK)).

```
silas@gateway: ~/testssl.sh-3.0rc5
Fichier  Édition  Affichage  Rechercher  Terminal  Aide

Start 2019-12-15 23:48:34      -->> 172.16.147.134:25 (gateway.ifi.edu.local)
<<--

rDNS (172.16.147.134):  ; ; ; ;
Service set:           STARTTLS via SMTP

Testing protocols via sockets

SSLv2      not offered (OK)
SSLv3      not offered (OK)
TLS 1      offered
TLS 1.1    offered
TLS 1.2    offered (OK)
TLS 1.3    offered (OK): final

Testing cipher categories

NULL ciphers (no encryption)      not offered (OK)
Anonymous NULL Ciphers (no authentication)  offered (NOT ok)
Export ciphers (w/o ADH+NULL)      not offered (OK)
LOW: 64 Bit + DES, RC[2,4] (w/o export)  not offered (OK)
Triple DES Ciphers / IDEA          not offered (OK)
```

FIGURE 22 – Version TLS sur notre Serveur configuré

### 5.3 Sécurité antimalware

Grâce au module ClamAV installé, nous garantissons une sécurité antispam à travers le module clam-milter et les protocoles SPF et DKIM, nous garantissons une sécurité antispam.

### 5.4 Perfect Forward Secrecy

Nous avons pu garantir la confidentialité à travers le PFS

```

silas@gateway: ~/testssl.sh-3.0rc5
Fichier  Édition  Affichage  Rechercher  Terminal  Aide

Testing robust (perfect) forward secrecy, (P)FS -- omitting Null Authentication
/Encryption, 3DES, RC4

PFS is offered (OK)

TLS_AES_256_GCM_SHA384
TLS_CHACHA20_POLY1305_SHA256
ECDHE-RSA-AES256-GCM-SHA384
ECDHE-RSA-AES256-SHA384 ECDHE-RSA-AES256-SHA
DHE-RSA-AES256-GCM-SHA384
ECDHE-RSA-CHACHA20-POLY1305
DHE-RSA-CHACHA20-POLY1305 DHE-RSA-AES256-CCM8
DHE-RSA-AES256-CCM DHE-RSA-AES256-SHA256
DHE-RSA-AES256-SHA ECDHE-RSA-CAMELLIA256-SHA384
DHE-RSA-CAMELLIA256-SHA256
DHE-RSA-CAMELLIA256-SHA
DHE-RSA-ARIA256-GCM-SHA384
ECDHE-ARIA256-GCM-SHA384 TLS_AES_128_GCM_SHA256
ECDHE-RSA-AES128-GCM-SHA256
ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA
DHE-RSA-AES128-GCM-SHA256 DHE-RSA-AES128-CCM8
DHE-RSA-AES128-CCM DHE-RSA-AES128-SHA256
DHE-RSA-AES128-SHA ECDHE-RSA-CAMELLIA128-SHA256
DHE-RSA-CAMELLIA128-SHA256 DHE-RSA-SEED-SHA

```

FIGURE 23 – Garantie du Perfect Forward Secrecy

## 5.5 Evaluation des Résultats

### 5.5.1 Appréciation des tâches réalisées

Dans le cadre de ce travail, en se basant sur le Tableau des points d'amélioration que nous avons présenté plus haut, nous avons pu réaliser une bonne partie des tâches. Malheureusement il demeure certaines tâches que nous n'avons pas pu effectuer. Le tableau ci-dessous présente un résumé de ces tâches. Les cases en blanc représentent les tâches que nous n'avons pas pu aborder. Les cases en jaune représentent les tâches que nous avons réalisées mais dont nous ne jugeons pas les résultats satisfaisants. Les cases en rouge représentent les tâches que nous avons démarrées mais que nous n'avons pas pu terminer. Les cases en vert représentent les tâches réalisées avec succès.

| Numéro | Elément de Sécurité                                                              | Importance                                                                                                          | Statut |
|--------|----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|--------|
| 1      | Protection du BIOS par un Strong Password                                        | Permet de restreindre les risques de modifications des configurations du BIOS                                       |        |
| 2      | Créer une partition distincte pour le répertoire des mails                       | Evite que l'espace grandissant occupé par les services mails empêchent le système de bien fonctionner dans le futur |        |
| 3      | Définir un mot de passe fort pour le compte root                                 | Sécuriser le compte root                                                                                            |        |
| 4      | Activer le chiffrement des mots de passe                                         | Sécuriser le stockage des mots de passe                                                                             |        |
| 5      | Désactiver les services qui ne sont pas nécessaires                              | Réduire le nombre de vulnérabilités à maîtriser                                                                     |        |
| 6      | Désactiver inetd et Telnet                                                       |                                                                                                                     |        |
| 7      | Appliquer les mises à jour de sécurité du système                                | Permet d'appliquer les patchs de sécurités et tenir les services à jour                                             |        |
| 8      | Restreindre les connexions consoles et les redémarrage système depuis la console | Réduire les impacts sur la disponibilité                                                                            |        |
| 9      | Configuration et Gestion efficace des Logs                                       | Permet de garder une traçabilité et maintenir une corrélation entre les événements                                  |        |
| 10     | Protéger le système contre le dépassement de tampon                              | Efficace pour prévenir les DoS                                                                                      |        |
| 11     | Installation et Configuration d'antivirus                                        | Spamassassin installé, ClamAv installé                                                                              |        |
| 12     | Configuration de Pare feu                                                        |                                                                                                                     |        |
| 13     | Configuration de IDS                                                             | Snort installé (Reste à ajouter certaines règles spécifiques)                                                       |        |
| 14     | Sécuriser le service de courrier au moyen du protocole TLS                       |                                                                                                                     |        |
| 15     | Garantir la délivrance d'un certificat numérique à tous les utilisateurs         |                                                                                                                     |        |
| 16     | Assurer la Confidentialité Persistante (Perfect Forward Secrecy)                 |                                                                                                                     |        |
| 17     | En cas d'accès Web, Garantir la sécurité en utilisant le protocole HTTPS         |                                                                                                                     |        |
| 18     | Protection contre les attaques XSS et les Session Hijacking                      |                                                                                                                     |        |
| 19     | Création d'une autorité de certification                                         |                                                                                                                     |        |

FIGURE 24 – Point des Tâches réalisées

### 5.5.2 Difficultés

- Dans la réalisation de ce travail, nous avons eu du mal à appréhender les différents aspects liés à notre sujet
- Nous avons eu du mal à définir le périmètre du travail.
- Maîtrise des différents outils à utiliser
- Réaliser les différentes configurations et tester leur résistance
- Non disponibilité des équipements adéquats
- Difficultés à obtenir une adresse IP Publique afin de mettre le serveur configuré en ligne
- Difficultés à obtenir un certificat de sécurité auprès d'une autorité de certification reconnue

### 5.5.3 Perspectives

Etant donné que nous n'avons pas pu réaliser toutes les tâches et qu'aucune oeuvre ne peut être jugée de parfaite nous avons en Perspectives de :

- Réaliser l'entièreté de la solution proposée
- Mettre en ligne et tester la robustesse de la solution

- Mettre en oeuvre une interface d'administration sécurisée du serveur accessible via Web
- Maintenir un environnement sécurisé autour du serveur mail de l'IFI

## Conclusion

Au cours de ce travail, nous avons pu cerner les différents contours de notre projet. Nous avons d'autant plus essayé de faire une recherche sur les travaux similaires, ceci nous a permis de repertorier les failles et vulnérabilités des serveurs mail en général et des serveurs postfix en particulier. A partir de cette liste de vulnérabilités, et par rapport à l'environnement de l'IFI, nous avons pu définir une solution à implémenter à travers une série d'actions sus-citées. Les tâches réalisées nous ont permis de mieux apprécier les aspects techniques liés à la sécurisation d'un serveur mail. Les différentes difficultés auxquelles nous avons été confrontés nous ont permis de développer des aptitudes à trouver des solutions optimales et à devenir proactif. Dans une perspective d'amélioration continue du travail, nous voudrions bien croire que notre contribution pourrait permettre une amélioration du niveau de sécurité du serveur mail de l'IFI.

## Lien GitHub pour accéder aux Appliances virtuelles

Lien GitHub[1]

## Références

- [1] Lien Github . <https://github.com/Silasdaniel/TPE/>.
- [2] Lynis Enterprise. <https://cisofy.com/lynis-enterprise/>.
- [3] Postfix : Security Vulnerabilities . [https://www.cvedetails.com/vulnerability-list/vendor\\_id-8450/Postfix.html](https://www.cvedetails.com/vulnerability-list/vendor_id-8450/Postfix.html).
- [4] Présentation de l'outils Nessus. [https://fr.wikipedia.org/wiki/Nessus\\_\(logiciel\)](https://fr.wikipedia.org/wiki/Nessus_(logiciel)).
- [5] Etude Comparative de l'utilisation des services de messagerie. <https://lamdaouarmohamed.files.wordpress.com/2015/10/comparative-serveur-et-client-messagerie.pdf>, Décembre 2015.
- [6] <https://www.ovh.com/fr/blog/rapport-attaques-ddos-observees-par-ovh-en-2017/>. <https://blog.devensys.com/ameliorer-la-securite-des-emails/>, Decembre 2017.
- [7] Les méthodes pour améliorer votre sécurité email. <https://blog.devensys.com/ameliorer-la-securite-des-emails/>, Janvier 2019.
- [8] Mail Server Security : Potential Vulnerabilities and Protection Methods. <https://www.apriorit.com/qa-blog/428-mail-server-security-testing>, Mars 2019.
- [9] Online Mail Server Check. <https://emailsecuritygrader.com/results?id=220087>, Juillet 2019.
- [10] Online TLS Check. <https://www.cdn77.com/tls-test>, Juillet 2019.
- [11] Kyle Dent. *Postfix : The Definitive Guide A Secure and Easy-to-Use MTA for UNIX*. O'Reilly Media, 2009.