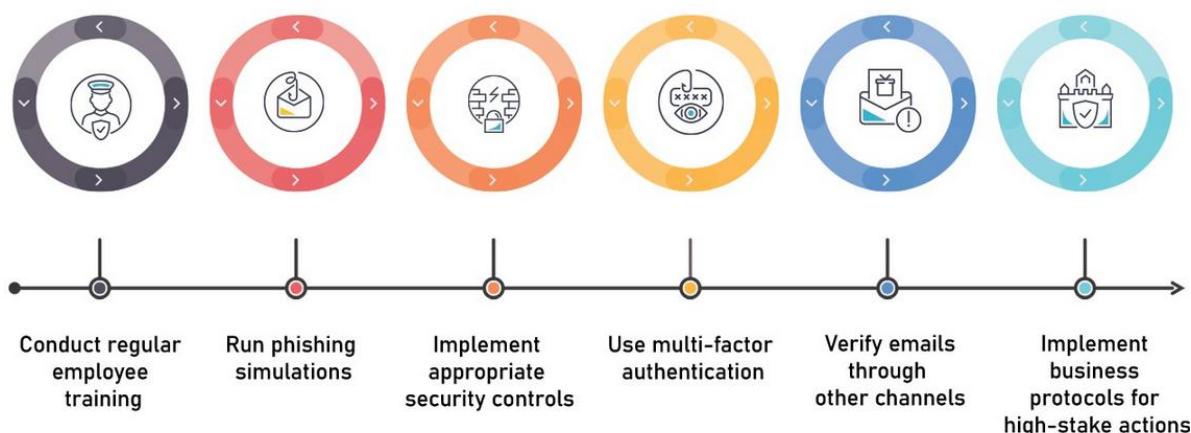


Projet d'ingénierie ICCN INE2

ICCN-INE2

Mise en place d'un service de messagerie sécurisé et développement d'un outil automatisé d'analyse des attaques par courrier électronique destiné aux analystes du SOC



Encadre par :

Mme Ayache Meryem
 Mme Hanin Charifa
 Mr Mezrioui Abdellatif
 Mme Ouaddah Aafaf

Réalisé par :

Bougalzim Imane
 Harrouche Ibtissam
 Hinaje Nezha
 Mouisset Hamza

Dédicace

Nous dédions ce travail comme un témoignage d'amour, d'affection et d'admiration :

À nos très chers parents,

Affables, honorables et aimables : vous représentez pour nous le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de nous encourager et de prier pour nous.

Vos prières et votre bénédiction nous ont été d'un grand secours pour mener à bien nos études. Aucune dédicace ne saurait être assez éloquente pour exprimer ce que vous méritez pour tous les sacrifices que vous n'avez cessé de nous donner depuis notre naissance, durant notre enfance et même à l'âge adulte.

À toute la famille et à tous nos chers amis,

Qu'ils trouvent dans ce modeste travail l'expression de notre profond respect et reconnaissance.

Remerciements

À la clôture de ce travail aussi bénéfique que profitable, nous désirons adresser nos remerciements marqués des expressions les plus distinguées de reconnaissance et de gratitude à :

Mr MEZRIOUI ABDELLATIF : pour ses directives éclaircissantes le long de ce projet. Nous avons eu le privilège d'apprécier vos qualités et vos valeurs.

Mme OUADDAH AAFAF : pour son accompagnement, son encadrement et ses conseils incitant à profiter au mieux de ce projet. Veuillez trouver ici l'expression de notre respectueuse considération et notre profonde admiration pour toutes vos qualités professionnelles et humaines.

Mme MERYEM AYACHE : pour votre soutien et vos conseils précieux. Votre expertise et votre dévouement ont été essentielle à la réalisation de ce rapport.

Mme HANIN Charifa : d'avoir accepté d'encadrer et de participer à l'élaboration de ce travail.

Permettez-nous de vous exprimer notre grand respect et notre profonde reconnaissance.

Nous remercions fortement Mesdames et Messieurs les membres du Jury del'honneur qu'ils nous font en siégeant dans ce jury. Veuillez trouver ici, Mesdames et Messieurs, l'expression de notre profond respect et de notre grande gratitude.

Il convient aussi de saluer tous les enseignants qui ont assuré notre formation au cours de cette année au sein de l'école prestigieuse INPT.

Enfin, nous adressons aussi nos sincères remerciements à tous ceux qui ont participé de près ou de loin à la réalisation de ce projet, aux personnes qui, bien que leurs noms ne figurent pas dans ce document, étaient toujours prêtes à aider et à contribuer au bon déroulement de ce travail.

Résumé

L'objectif de ce projet est de comprendre le fonctionnement de service de messagerie afin de pouvoir mettre en place un service sécurisé en implémentant les outils nécessaires.

En effet, ce projet de fin d'année est focalisé sur la conception d'un outil support aux analystes du SOC pour l'analyse des messages suspect. Pour parvenir à ses fins, nous allons décortiquer le service de messagerie afin de comprendre le fonctionnement. Ensuite faire un choix des outils adéquats pour la sécurité de ce service. Une fois l'implémentation de ces outils est achevée, nous serons amenés à comprendre les différents algorithmes d'intelligence artificielle pour détecter les attaques de phishing et de spam de manière plus efficace, réduisant ainsi la charge de travail des analystes du SOC et améliorant la rapidité de la réponse aux menaces

Nous avons entrepris une étude approfondie du fonctionnement des services de messagerie, des protocoles sous-jacents (tels que SMTP, POP, IMAP), ainsi que des principales menaces et attaques courantes (phishing, spam, virus, etc.). Nous avons également exploré les différentes techniques et outils de sécurité disponibles, notamment DKIM, SPF, DMARC, SpamAssassin, ClamAV et VirusTotal.

Le projet s'est déroulé sur 14 semaines, avec une première phase dédiée à l'étude bibliographique, suivie par la réalisation technique de la plateforme de messagerie et du développement de l'outil d'analyse. Chaque phase a été rigoureusement documentée, avec des livrables techniques détaillant l'installation, la configuration et le développement des solutions mises en place.

En conclusion, ce projet a permis de développer une solution intégrée et automatisée pour améliorer la sécurité des services de messagerie et l'efficacité des analystes du SOC dans la gestion des menaces par courriel.

Mots-clés : serveur de messagerie, sécurité, attaque , spam ,phishing,AI,SOC....

Abstract

The objective of this project is to understand how messaging services work in order to be able to set up a secure service by implementing the necessary tools.

Indeed, this end-of-year project focuses on the design of a support tool for SOC analysts for the analysis of suspicious messages. To achieve this, we will dissect the messaging service in order to understand how it works. Then choose the appropriate tools for the security of this service. Once the implementation of these tools is completed, we will be led to understand the different artificial intelligence algorithms to detect phishing and spam attacks more effectively, thus reducing the workload of SOC analysts and improving the speed of response to threats

We have undertaken an in-depth study of how email services work, the underlying protocols (such as SMTP, POP, IMAP), and the main common threats and attacks (phishing, spam, viruses, etc.). We also explored the different security techniques and tools available, including DKIM, SPF, DMARC, SpamAssassin, ClamAV, and VirusTotal.

The project took place over 14 weeks, with a first phase dedicated to the bibliographic study, followed by the technical creation of the messaging platform and the development of the analysis tool. Each phase has been rigorously documented, with technical deliverables detailing the installation, configuration and development of the solutions implemented.

In conclusion, this project made it possible to develop an integrated and automated solution to improve the security of messaging services and the effectiveness of SOC analysts in managing email threats

Keywords :

Liste des abréviations

Abréviation	Définition
SOC	Security Operations Center
POP	Post Office Protocol
IMAP	Internet Message Access Protocol
SMTP	Simple Mail Transfer Protocol
DMARC	Domain-based Message Authentication, Reporting, and Conformance
SKIM	DomainKeys Identified Mail
SPF	Sender Policy Framework
AS/AV	Anti-Spam/Anti-Virus
OSPF	Open Shortest Path First
RIP	Routing Information Protocol
NAT	Network Address Translation
PAT	Port Address Translation
MTA	Mail Transfer Agent
MDA	Mail Delivery Agent
MUA	Mail User Agent
DNS	Domain Name System
SSL	Secure Sockets Layer
TLS	Transport Layer Security
XSS	Cross-Site Scripting
S/MIME	Secure/Multipurpose Internet Mail Extensions
ML	Machine Learning
API	Application Programming Interface
BoW	Bag of Words
TF-IDF	Term Frequency-Inverse Document Frequency

Table des figures

<i>Figure 1.1 : Diagramme de composants UML</i>
<i>Figure 1.2 : Diagramme de séquences UML</i>
<i>Figure 1.3 : Diagramme Gantt</i>
<i>Figure 2.1 : Architecture matérielle des entreprises</i>
<i>Figure 2.2 : Configuration de client 1</i>
<i>Figure 2.3 : Configuration de client 2</i>
<i>Figure 2.4 : Configuration DNS de l'entreprise 1</i>
<i>Figure 2.5 : Configuration DNS de l'entreprise 2</i>
<i>Figure 2.6 : Ping allant du serveur 2 vers le serveur 1</i>
<i>Figure 2.7 : Ping allant du serveur 1 vers le serveur 2</i>
<i>Figure 2.8 : Résolution du nom de serveur de messagerie de l'entreprise 1</i>
<i>Figure 2.9 : Accès à l'internet à partir du serveur 1</i>
<i>Figure 2.10 : Accès à l'internet à partir du serveur 2</i>
<i>Figure 3.1 : Webmail Zimbra</i>
<i>Figure 3.2 : Webmail Roundcube</i>
<i>Figure 3.4 : Installation de Zimbra</i>
<i>Figure 3.5 : Installation de finale de Zimbra</i>
<i>Figure 3.6 : Webmail page de Zimbra</i>
<i>Figure 3.7 : Message envoyé sans DKIM, SPF et DMARC</i>
<i>Figure 3.8 : l'ajout de record DKIM, SPF et DMARC dans le fichier de la zone entreprise1.ma</i>
<i>Figure 3.9 : Envoi du mail avec DKIM, SPF et DMARC</i>
<i>Figure 3.10 : Infrastructure de messagerie</i>
<i>Figure 3.11 : Installation des modules nécessaires</i>

- Figure 3.12 : Vérification du statut mariadb.....*
- Figure 3.13 : Version de php.....*
- Figure 3.14 : Configuration de Postfix.....*
- Figure 3.15 : Création du groupe et utilisateur qui aura accès au répertoire var/virtual_mail_box.....*
- Figure 3.16 : Création de base de données postfixadmin_db.....*
- Figure 3.17 : Connexion entre MySQL et Postfix.....*
- Figure 3.18 : Lancement de Postfix et Dovecot.....*
- Figure 3.19 : Autorisation des ports liés à POP 3 et IMAP.....*
- Figure 3.20 : Configuration de Dovecot.....*
- Figure 3.21 : Statut de Dovecot.....*
- Figure 3.22: Installation de Postfixadmin.....*
- Figure 3.23 : Configuration du fichier postfixadmin.entreprise2.ma.conf.....*
- Figure 3.24 : Interface graphique de postfixadmin.....*
- Figure 3.25 : Configuration du fichier config.local.php.....*
- Figure 3.26 : Interface Postfixadmin après configuration.....*
- Figure 3.27 : Ajout du superadmin.....*
- Figure 3.28 : Ajout de notre propre domaine.....*
- Figure 3.29 : Ajout de Mailbox.....*
- Figure 3.30 : test de Postfix.....*
- Figure 3.31 : test de Dovecot.....*
- Figure 3.32 : Création de base données roundcube_db.....*
- Figure 3.33 : Installation de Roundcube.....*
- Figure 3.34 : Création d'utilisateur Roundcube.....*
- Figure 3.35 : Interface de Roundcube installer.....*
- Figure 3.36 : Test SMTP.....*

- Figure 3.37 : Test IMAP*
- Figure 3.38 : Message header avant openDkim*
- Figure 3.39 : Génération des clés DKIM*
- Figure 3.40 : Envoi du message après la configuration OpenDkim*
- Figure 4.1 : fonctionnement de Clamav-Amavis*
- Figure 4.2 : Mise à jour de Clamav*
- Figure 4.3 : Détection du spam par Amavis*
- Figure 4.4 : fonctionnement de Spamassassin avec Amavis*
- Figure 4.5 : Rubrique de filtres*
- Figure 4.6 : Ajout des règles graphiquement*
- Figure 4.7 : Ajout des règles en /etc/spamassassin*
- Figure 4.8 : Message détecté comme spam*
- Figure 4.9 : test des règles établis de spamassassin*
- Figure 4.10 : Règle défini sur l'entête*
- Figure 4.11 : Message détecté en tant que spam*
- Figure 4.12: API KEY Virustotal*
- Figure 4.13 : Fonctionnement de l'API de VirusTotal*
- Figure 4.14 : Installation de l 'API VirusTotal*
- Figure 4.15 : /etc/postfix/master.cf*
- Figure 4.16 : /etc/postfix/main.cf*
- Figure 4.17 : Message de L 'expéditeur*
- Figure 4.18 : Message chez le Destinataire*

<i>Figure 5.1 : Flux d'informations dans les étapes des attaques de phishing</i>
<i>Figure 5.2 : Méthodes de recherche adoptées</i>
<i>Figure 5.3 : visualisation de la distribution de type des e-mails</i>
<i>Figure 5.4 : visualisation de la distribution de type des e-mails après le undersampling ..</i>
<i>Figure 5.5 : visualisation des performances de model de ML</i>
<i>Figure 5.6 : contenu de répertoire répertoire /home/imane/Desktop/phishing-detection...</i>
<i>Figure 5.7 : test de fonctionnement de filtre de phishing base sur le modèle de ML</i>
<i>Figure 5.8 : le fichier /var/log/mail.log</i>
<i>Figure 5.9 : les lignes à insérer dans les fichiers de configuration de postfix main.cf et master.cf.....</i>
<i>Figure 5.10 : test de filtre de ML implémenté et intégré au service de messagerie</i>
<i>Figure 5.11 : /etc/postfix/main.cf.....</i>
<i>Figure 5.12 : /etc/postfix/master.cf.....</i>
<i>Figure 5.13 : résultat de l'entraînement de données.....</i>
<i>Figure 5.14 : fichier de test du model.....</i>
<i>Figure 5.15 : Résultat du test.....</i>
<i>Figure 5.16 : Résultat du test de performances du model.....</i>
<i>Figure 6.1 : Logo Rspamd.....</i>
<i>Figure 6.2 : Fonctionnement de Rspamd.....</i>
<i>Figure 6.3 : Page d'accueil de Rspamd.....</i>
<i>Figure 6.4 : Onglet « Throughput »</i>
<i>Figure 6.5 : Onglet « Configuration »</i>
<i>Figure 6.6 :Onglet « Symbols »</i>
<i>Figure 6.7 :Onglet « History »</i>
<i>Figure 6.8 : Schéma de flux pour Rspamd.....</i>
<i>Figure 6.9: Installation de Rspamd.....</i>

- Figure 6.10 : le répertoire /etc/rspamd*
- Figure 6.11 : /etc/rspamd/rspamd.conf*
- Figure 6.12 : Status du filtre Rspamd*
- Figure 6.13 : Sistatiques de Rspamd*
- Figure 6.14 : Historique de Rspamd*
- Figure 6.15 : installation de pflogsum*
- Figure 6.16 : extrait de rapport généré*
- Figure 6.17 : configuration au niveau de crontab*
- Figure 6.18 : rapport envoyé à l'administrateur*
- Figure 6.19 : exemple de statistique graphique fournit par Mailgraph*
- Figure 6.20 : Mailgraph package*
- Figure 6.21 : le fichier /usr/lib/cgi-bin/mailgraph.cgi*
- Figure 6.22 : le fichier /etc/postfix/main.cf*
- Figure 6.22 : lancement de mailgraph*
- Figure 6.22 : l'interface de Mailgraph*

Table des matières

Contents

<i>Dédicace</i>	2
Liste des abréviations	6
Table des figures	7
Table des matières	12
Introduction générale	14
Chapitre 1 : Etude du projet et spécifications des besoins	15
1.1 Sécurité de la messagerie électronique	16
1.2 Protocoles de sécurité	17
1.3 Outils de détection et de protection	18
1.4 Modélisation de notre projet	21
1.5 Gestion de projet et ordonnancement	23
Chapitre 2 : Architecture Réseau pour l'interconnexion des services de messagerie entre deux entreprises	27
2.1 Architecture des entreprises	28
2.2 Description de l'architecture	28
2.3 Configuration des deux sites	29
2.4 Tests de connectivité	32
Chapitre 3 : Réalisation de service de messagerie sécurisée	35
3.1 Choix de la plateforme messagerie :	36
3.2 Mise en œuvre des éléments de base	38
Chapitre 4 : Renforcement de la Sécurité du Service de Messagerie	56
4.1 Clamav amavis :	57
4.2 SpamAssassin	59
Explication de la règle	62
4.3 VirusTotal	64
Chapitre 5 : Intelligence Artificielle pour la Sécurité des Services de Messagerie	70
5.1 Les phases des attaques de phishing basées sur des malwares	71
5.2 Méthodologie d'étude	72
5.3 Détection de Phishing par Apprentissage Automatique	73

5.4 Détection de spam par apprentissage automatique	80
Chapitre 6 : Surveiller et protéger la boîte aux lettres	87
6.1 Pourquoi Rspamd :	88
6.2 Fonctionnalités	91
6.3 Interface d'administration	93
6.4 Mise en place avec postfix	96
6.5 Pflogsumm pour la Surveillance de l'Activité de Postfix	99
6.6 Mailgraph pour la Surveillance de serveur de messagerie	102
Conclusion générale	107
Bibliographie	108
Annexes	109
Annexe A : Modèle Bibliographie	Erreur ! Signet non défini.

Introduction générale

La messagerie électronique, souvent abrégée en e-mail, représente un pilier essentiel des échanges d'information contemporains. Elle fonctionne grâce à un ensemble de composants et de protocoles qui orchestrent le transfert efficace des messages entre expéditeurs et destinataires. Comprendre cette architecture et ces protocoles est crucial pour appréhender le fonctionnement global.

Cependant, ce service n'est pas exempt de menaces. Parmi les attaques les plus courantes figurent le phishing, où des attaquants tentent de tromper les utilisateurs pour obtenir des informations sensibles, et le spam, qui inonde les boîtes de réception de courriers indésirables. Pour contrer ces menaces, divers outils de détection sont utilisés, comme SpamAssassin pour filtrer les spams et ClamAV pour la détection de logiciels malveillants. De plus, l'intégration d'outils de machine learning dans les systèmes de sécurité opérationnelle (SOC) permet aux analystes de mieux identifier et réagir aux menaces émergentes, en offrant une analyse plus sophistiquée et proactive des attaques potentielles.

Notre objectif dans ce projet est d'essayer de configurer un service de messagerie from scratch afin de comprendre les différents composants de ce service. Puis, nous allons essayer d'intégrer les différents outils de détection de spam afin de sécuriser notre plateforme ensuite nous allons insérer des algorithmes d'intelligence artificielle afin d'automatiser les tâches d'analyse et enfin nous allons essayer d'intégrer des outils graphiques de détection pour visualiser le type des courriels envoyés.

Dans le premier chapitre, nous allons vous mettre dans le contexte général du projet en parlant de la problématique, objectif du projet, la méthodologie et le planning suivi. Ensuite, le deuxième chapitre sera consacré pour parler sur l'architecture réseau. Pour le troisième chapitre, nous allons parler de la mise en place de service de messagerie sécurisé en parlant sur la structure de service de messagerie. En ce qui concerne le quatrième chapitre, nous allons évoquer les méthodes utilisées pour renforcer la sécurité du service de messagerie. Le cinquième chapitre va porter sur l'utilisation des algorithmes d'intelligence artificielle afin de sécuriser les services de messagerie. Au final, le dernier chapitre va porter sur les outils de surveillance et détection afin de protéger la boîte aux lettres.

Chapitre 1

Etude du projet et spécifications des besoins

Afin de mettre en place un service de messagerie robuste avec un outil d'automatisation support aux SOC pour l'analyse des messages suspects, l'étude de la sécurité de messagerie et la spécification des besoins s'avèrent primordiales. L'objectif de ce chapitre est d'introduire les menaces et les protocoles de sécurité brièvement en présentant à la fois les outils de détection, la problématique générale et les objectifs du projet et enfin de donner un aperçu de la gestion du projet.

1.1 Sécurité de la messagerie électronique

De nos jours, la messagerie électronique est devenue une cible de diverses menaces et vulnérabilités. Ces attaques compromettent les triades de sécurité qui sont : la confidentialité, l'intégrité et la disponibilité des communications. Voici un aperçu des principales menaces :

- Phishing

Le phishing est une technique frauduleuse utilisant l'ingénierie sociale où les attaquants essaient de tromper les utilisateurs et obtenir des informations sensibles telles que des identifiants de connexion, des numéros de carte de crédit ou des informations personnelles. Les emails de phishing contiennent des liens vers des sites web factices. Parmi ce que nous pouvons trouver dans les textes de phishing : des erreurs grammaticales, des demandes urgentes d'action et des adresses électroniques suspectes. [1.1]

- Spam

Le spam désigne tout message non sollicité et inapproprié envoyé sur l'internet, généralement à un grand nombre d'utilisateurs, principalement à des fins de publicité, de phishing. Il est souvent envoyé en masse afin d'encombrer les boîtes de réception et réduire la productivité. La majorité des spameurs récupèrent les emails des bases de données qui sont partagées soit dans des forums soit sur des sites anodins ou des jeux concours.

- Malware et virus

Les logiciels malveillants (malware) et les virus sont des logiciels malveillants distribués via des pièces jointes ou des liens dans les courriels. Une fois ouverts, ces fichiers infectent le système de l'utilisateur et donnent accès aux attaquants à des données sensibles, de contrôler l'ordinateur à distance ou de chiffrer des fichiers pour une demande de rançon (ransomware). Les malwares sont cachés en documents légitimes afin d'inciter les utilisateurs à les ouvrir.

- Attaques de Spoofing

Les attaques de spoofing permettent de falsifier l'identité de l'expéditeur dans un courriel pour but de tromper les destinataires. Les techniques de spoofing de courriel manipulent le champ "De" ou les informations de l'en-tête du courriel afin de faire apparaître le message comme venant d'une source de confiance. Les courriels usurpés sont couramment utilisés pour mener des campagnes de phishing, distribuer des logiciels malveillants ou initier des attaques

d'ingénierie sociale. En imitant des organisations ou des individus réputés, les attaques de spoofing visent à exploiter la confiance et la crédibilité associées à l'entité usurpée.

- Compromission de courriel professionnel (BEC)

Les attaques de compromission de courriel professionnel (BEC) ciblent les organisations en usurpant l'identité de cadres supérieurs, d'employés ou de partenaires de confiance. Ces attaques visent à tromper les employés pour qu'ils initient des transactions financières non autorisées, transfèrent des fonds vers des comptes frauduleux ou révèlent des informations sensibles de l'entreprise. Les attaques BEC impliquent souvent des techniques d'ingénierie sociale, exploitant les relations de confiance au sein de l'organisation pour mener des activités frauduleuses. [1.2]

Les menaces citées ci haut exigent de mettre en place des mesures de sécurité robustes afin de protéger les systèmes de messagerie électroniques. De l'autre côté, les utilisateurs doivent être éduqués sur les risques et les meilleures pratiques de sécurité, tandis que les administrateurs doivent déployer des solutions techniques pour détecter et prévenir ces attaques.

1.2 Protocoles de sécurité

Les protocoles de sécurité visent à protéger les communications de messagerie électronique. Ils sont conçus pour assurer la confidentialité, l'intégrité et l'authenticité des courriels. De plus, ils permettent de lutter contre diverses menaces telles que l'interception de données, la falsification et l'usurpation d'identité. Citons ci-dessous quelques protocoles de sécurité :

- SSL/TLS

SSL (Secure Sockets Layer) est un protocole permettant de sécuriser les échanges sur Internet, devenu Transport Layer Security au niveau 4 du Modèle OSI. Ils permettent aux applications client /serveur de communiquer dans un réseau non sûr tout en empêchant l'écoute et la modification des messages. Dans le contexte de la messagerie électronique, SSL/TLS est principalement utilisé pour chiffrer les connexions entre les clients de messagerie et les serveurs de messagerie, ainsi qu'entre les serveurs eux-mêmes. [1.3]

- S/MIME

S/MIME (Secure/Multipurpose Internet Mail Extensions) est un standard de cryptographie pour le courrier électronique qui permet de signer numériquement et de chiffrer les messages.

MIME améliore l'email en facilitant l'échange de divers types de contenu, tels que des images, de l'audio, de la vidéo et des documents, au sein des messages email. Il permet d'encapsuler différents types de contenu garantissant la compatibilité lors de la transmission de l'expéditeur au destinataire. À la réception de l'email, le client email du destinataire interprète la structure MIME pour afficher et gérer correctement les différents types de contenu inclus dans le message.

- DKIM, SPF, DMARC

DKIM, SPF et DMARC sont des protocoles de validation d'authentification des courriels conçus pour lutter contre l'usurpation d'identité et du spam.

DKIM : Permet aux expéditeurs de signer leurs courriels avec une clé cryptographique liée à leur domaine, ce qui permet aux destinataires de vérifier que les messages proviennent réellement du domaine indiqué et n'ont pas été modifiés.

SPF : Permet aux propriétaires de domaines de spécifier quelles adresses IP sont autorisées à envoyer des courriels pour leur domaine. Les serveurs de réception peuvent vérifier cette information pour détecter et bloquer les courriels frauduleux.

DMARC : Combine les mécanismes de DKIM et SPF et permet aux propriétaires de domaines de publier des politiques sur la façon dont les courriels qui échouent aux vérifications DKIM et SPF doivent être traités. DMARC fournit également des rapports sur les tentatives de falsification de domaine, aidant les administrateurs à identifier et à réagir aux attaques.

Ensemble, DKIM, SPF, et DMARC forment une défense robuste contre le phishing et le spam, en permettant une vérification rigoureuse de l'authenticité des courriels.

L'implémentation de ces protocoles de sécurité est essentielle pour toute organisation souhaitant protéger ses communications par courriel. Ils réduisent les risques d'attaques, renforcent la confiance des utilisateurs et assurent la conformité avec les meilleures pratiques de sécurité.

1.3 Outils de détection et de protection

Pour garantir la sécurité de la messagerie électronique, nous avons utilisés des outils de détection des différentes menaces comme solution de filtrage des courriels indésirables. Citons donc :

- Filtrage anti-spam (SpamAssassin)

SpamAssassin est un exemple bien connu de logiciel de filtrage anti-spam. Il analyse les courriels entrants et attribue un score basé sur diverses règles et techniques pour déterminer la probabilité qu'un message soit du spam. Parmi les techniques utilisées par ce logiciel pour filtrer les courriels indésirables nous trouvons :

L'analyse des entêtes : SpamAssassin examine les entêtes des emails pour détecter des éléments suspects, comme une fausse adresse d'expéditeur ou un domaine discordant.

L'analyse de contenu : Le filtre analyse le contenu de l'email à la recherche de caractéristiques propres au spam, comme l'utilisation excessive de majuscules et des liens douteux.

Les listes noires et blanches : SpamAssassin vérifie l'adresse IP de l'expéditeur contre des listes noires de spameurs connus et permet aux utilisateurs de créer des listes blanches d'expéditeurs de confiance.

Le filtrage bayésien : Cette technique statistique estime la probabilité qu'un email soit un spam en fonction de son contenu et le compare à une base de données d'emails spam connus

En utilisant des techniques combinées, SpamAssassin offre une solution flexible et puissante pour réduire la quantité de spam dans les boîtes de réception.

- Antivirus (ex. ClamAV)

Clamav est un logiciel antivirus open source conçue pour la détection et la suppression des logiciels malveillants, y compris les virus, les chevaux de Troie, les vers et autres menaces.

Base de données de signatures : Clamav utilise une base de données de signatures constamment mise à jour pour détecter les logiciels malveillants. Ces signatures sont des empreintes digitales uniques qui correspondent à des codes malveillants connus.

Analyse de fichiers : Lorsqu'un fichier est soumis à Clamav pour analyse, le logiciel vérifie son contenu par rapport à sa base de données de signatures. S'il trouve une correspondance, il identifie le fichier comme étant infecté.

Heuristiques : En plus des signatures, Clamav utilise des techniques heuristiques pour détecter les logiciels malveillants basés sur leur comportement ou leurs caractéristiques. Cela lui

permet de détecter des menaces même si elles n'ont pas encore été identifiées par une signature.

Intégration avec d'autres logiciels : Clamav peut être intégré à d'autres logiciels et systèmes, tels que des serveurs de messagerie, pour analyser les pièces jointes et les e-mails entrants

Grâce à sa capacité à détecter une large gamme de menaces, ClamAV est un outil essentiel dans la protection des systèmes de messagerie contre les infections par malwares.

- **Virustotal**

VirusTotal est une plateforme en ligne qui agit comme un agrégateur et un analyseur de fichiers suspects ou potentiellement malveillants. Elle permet aux utilisateurs de soumettre des fichiers, des URL ou des adresses IP pour une analyse approfondie de la présence de logiciels malveillants ou d'autres menaces potentielles. Dans le cadre de la sécurisation de la messagerie, à l'aide de VirusTotal on peut :

Analyser les pièces jointes : Lorsqu'on reçoit un e-mail avec une pièce jointe suspecte, on peut la télécharger sur VirusTotal pour une analyse approfondie. VirusTotal utilisera ses moteurs antivirus et d'autres outils pour détecter la présence de logiciels malveillants ou d'autres menaces potentielles dans le fichier.

Vérifier les liens : Si un e-mail contient des liens suspects ou douteux, on peut également soumettre ces liens à VirusTotal pour une analyse de la réputation. VirusTotal examinera le lien pour détecter d'éventuels sites malveillants ou frauduleux.

Analyser les adresses IP : Les e-mails peuvent également contenir des adresses IP suspectes, par exemple dans les en-têtes ou les liens. VirusTotal peut analyser ces adresses IP pour identifier toute activité malveillante associée à ces adresses.

- **Solutions de sécurité basées sur la machine learning**

Les solutions de sécurité basées sur la machine learning (ML) sont de plus en plus utilisées pour détecter et prévenir les menaces avancées dans les systèmes de messagerie électronique. Nous avons opté pour Rspamd comme outil de surveillance de la messagerie qui s'appuie sur des techniques d'apprentissage automatique et une base de données de menaces constamment mise à jour. Parmi les avantages de cette solution, nous trouvons :

Analyse comportementale : Les algorithmes de ML peuvent analyser les comportements des courriels et des utilisateurs pour identifier des anomalies. Par exemple, un changement

soudain dans le volume de courriels envoyés par un utilisateur pourrait indiquer une compromission de compte.

Filtrage adaptatif : Contrairement aux filtres anti-spam traditionnels basés sur des règles fixes, les solutions ML peuvent s'adapter et évoluer en fonction des nouvelles menaces. Elles peuvent apprendre à partir de courriels marqués comme spam ou non-spam par les utilisateurs pour améliorer continuellement leur précision.

En intégrant des solutions basées sur l'intelligence artificielle, les systèmes de messagerie peuvent bénéficier d'une protection plus dynamique et proactive contre les menaces évolutives.

1.4 Modélisation de notre projet

De la même façon qu'il vaut mieux dessiner une maison avant de la construire, il vaut mieux modéliser un système avant de le réaliser. Modéliser, c'est décrire de manière visuelle et graphique les besoins, les solutions fonctionnelles et techniques du projet. Modéliser pour :

- Obtenir une modélisation de très haut niveau indépendante des langages et des environnements.
- Faire collaborer des participants de tous horizons autour d'un même document de synthèse.
- Faire des simulations avant de construire un système.

1.4.1 Diagramme des composants

Au niveau de ce diagramme, nous allons essayer de montrer les différentes parties du système et leurs relations y compris les outils de détection spam/ phishing. L'utilisateur final utilise le client email tel que Zimbra ou Roundcube. Le serveur de messagerie Postfix gère l'envoi et la réception des emails.

La figure 1.1 montre comment un système de messagerie sécurisé peut être configuré en utilisant Postfix et Dovecot, avec divers outils de détection et de filtrage pour assurer la sécurité et l'intégrité des communications par email.

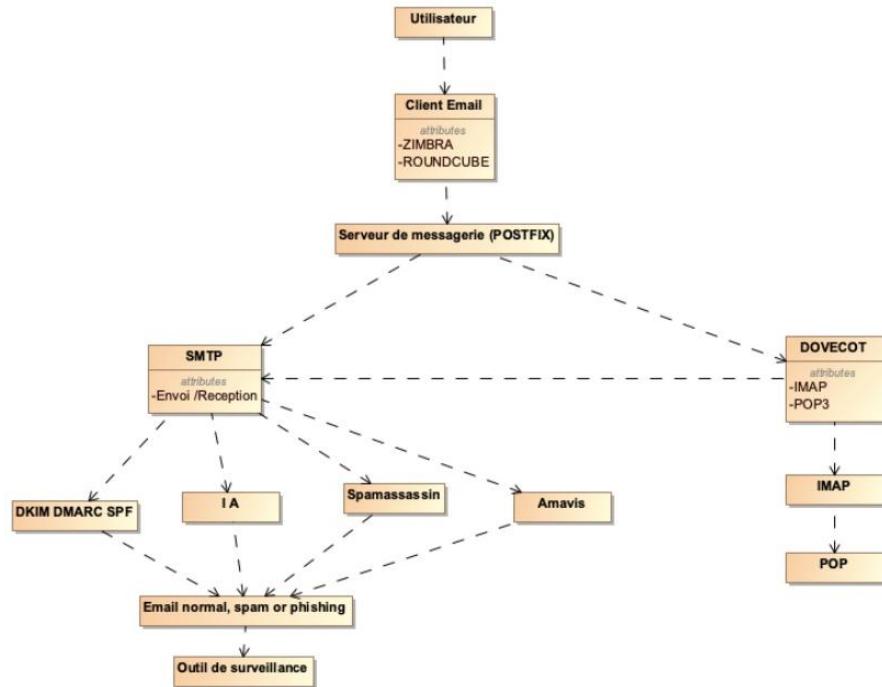


Figure 1.1 : Diagramme de composants UML

1.4.2 Diagramme de séquences

L'utilisateur envoie un email via son client de messagerie. Le client de messagerie utilise le protocole SMTP pour envoyer l'email au serveur Postfix. Ce dernier transmet l'email au destinataire via SMTP. Dovecot utilise les protocoles IMAP ou POP pour rendre l'email accessible au client de messagerie du destinataire.

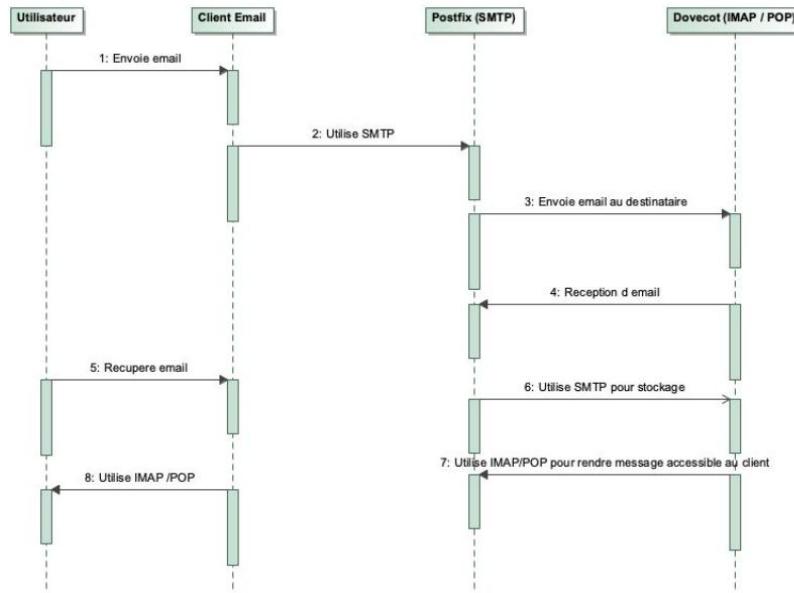


Figure 1.2 : Diagramme de séquences UML

1.5 Gestion de projet et ordonnancement

C'est l'ensemble des moyens humains et matériels pour atteindre un objectif dans un temps défini. Elle consiste à organiser et suivre chaque action du projet afin de tenir les délais, et la qualité requise. Afin de bien ordonner nos tâches, une planification s'avère nécessaire

1.5.1 Planning du projet

Les tâches du projet ont été réparties selon le programme représenté dans le tableau suivant :

Phases	Date	Description
Recherche bibliographique	Du 8/02/2024 au 19/02/2024	<ul style="list-style-type: none"> ○ Etude détaillée du service de Messagerie ○ Menaces et attaques types sur la Messagerie ○ Techniques et outils de sécurité de la messagerie
Architecture réseau	Du 21/02/2024 au 24/02/2024	<ul style="list-style-type: none"> ○ Choix d'une architecture pour lier les deux entreprises

		<ul style="list-style-type: none"> ○ Configuration des routeurs et des pcs ○ Choix des protocoles de routage pour assurer liaison entre les deux entreprises
Réalisation des services de messagerie	Du 26/02/2024 au 18/03/2024	<ul style="list-style-type: none"> ○ Installation et configuration d'outil Zimbra ○ Installation et configuration de MTA, MDA ○ Installation et configuration de Roundcube ○ Intégration de Roundcube avec MTA et MDA
Intégration des outils de sécurité	Du 20/03/2024 au 29/03/2024	<ul style="list-style-type: none"> ○ Configuration de SPF, DMARC et DKIM ○ Installation de spamassassin, Virustoatal, Clamav amavis et rspamd ○ Intégration de ces outils avec MUA : Roundcube et Zimbra
Réalisation des outils d'analyse automatiques	Du 10/04/2024 au 29/04/2024	<ul style="list-style-type: none"> ○ Choix et préparation des bases de données pour entraînement ○ Choix d'algorithme de machine learning ○ Tests de performance d'algorithme ○ Installation des outils de surveillance : Rspamd,

		mailgraph et Pflogsum
Soutenance finale	Du 03/06/2024 au 06/06/2024	<ul style="list-style-type: none"> ○ Préparation et répétition de la soutenance.

Tableau 1.1: Programme du projet

1.5.2 Logiciel de gestion de projet

Microsoft Project :(MS Project ou également MSP) est un logiciel Microsoft dédié à la gestion de projets. Il permet aux planificateurs et aux chefs de projets d'organiser et de piloter celui-ci, de gérer les ressources, le budget, l'analyse des données...

MS Project permet la planification d'un projet : il est possible à tout moment créer des tâches et des jalons, définir les liens entre chaque tâche, les hiérarchiser. MS Project a également la capacité d'estimer la durée ainsi que la charge de travail nécessaire pour accomplir une tâche définie. Microsoft Project permet aussi la création de modèles qui permet à l'utilisateur de respecter une méthodologie ou un processus quelconque. Le projet peut être représenté graphiquement de différentes manières : diagramme de Gantt, réseau des tâches... Le pilotage du projet est possible par de multiples façons telles que la définition de la planification initiale, la saisie de l'avancement des tâches ou bien la replanification.

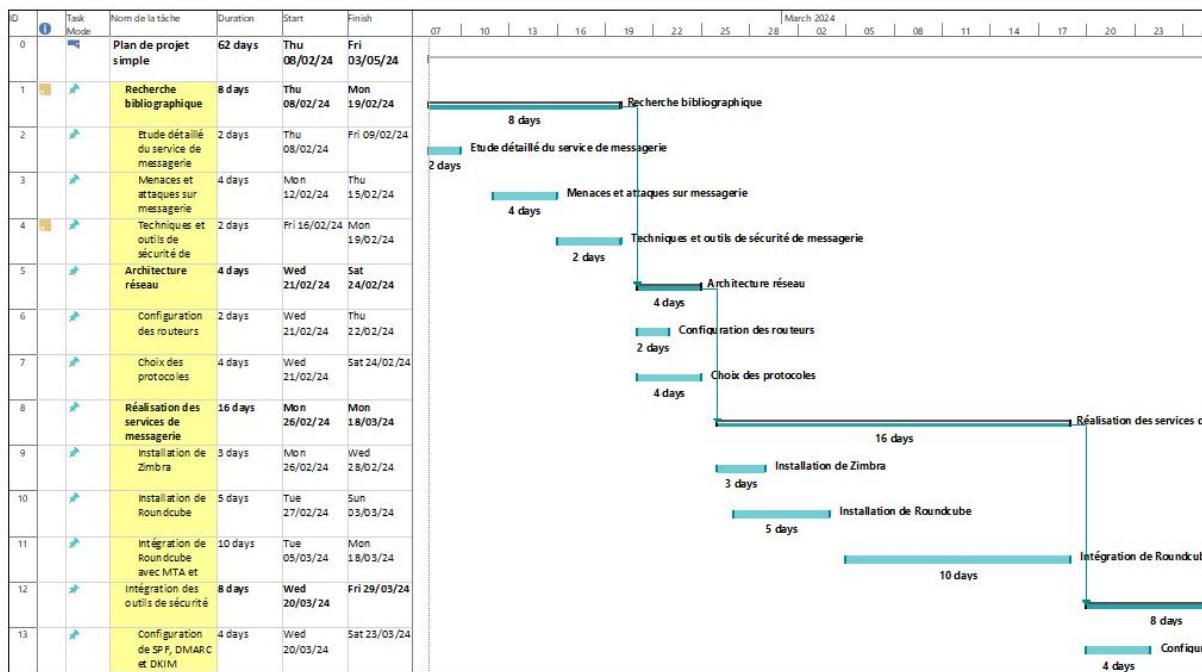


Figure 1.3 : Diagramme Gantt

Conclusion

A travers ce chapitre, nous avons pu avoir une idée globale sur le projet, ainsi que la problématique posée. Nous avons aussi eu une vision globale sur le déroulement de la conception de notre plateforme. Donc nous avons débuté par un choix de l'architecture réseau ainsi que les protocoles. Cette dernière fera l'objet du chapitre suivant.

Chapitre 2

Architecture Réseau pour l'interconnexion des services de messagerie entre deux entreprises

2.1 Architecture des entreprises

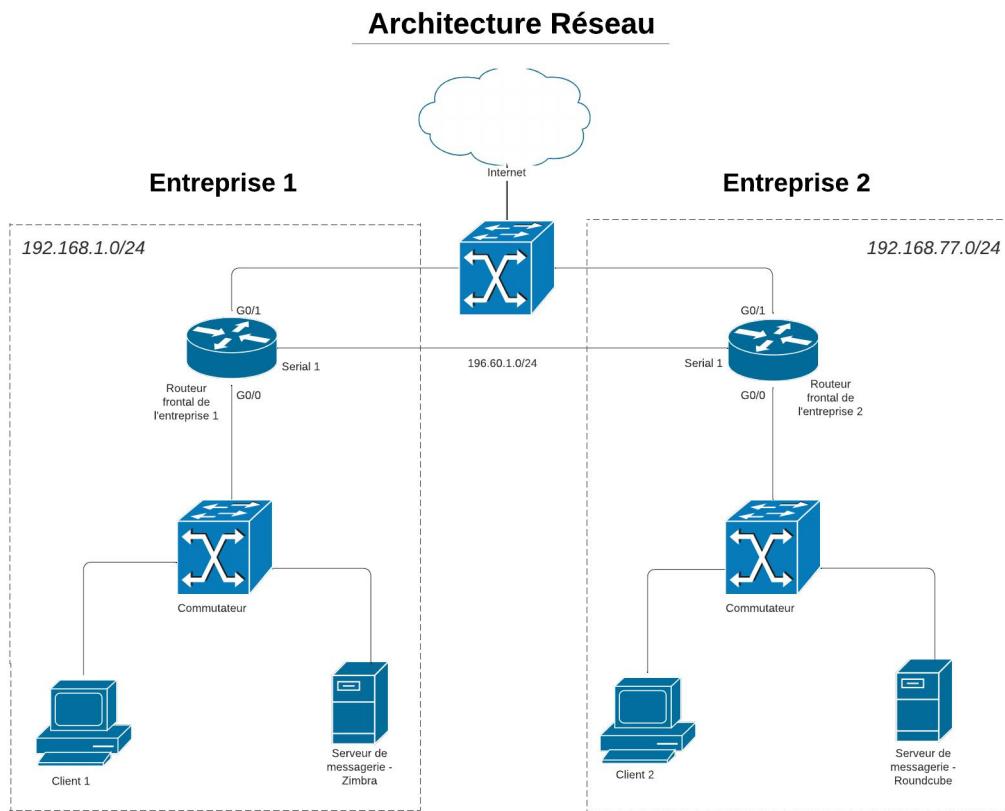


Figure 2.1 : Architecture matérielle des entreprises

2.2 Description de l'architecture

Nous avons choisi de simuler un scénario réel de la communication entre deux clients de messagerie distants. Ainsi nous avons créé deux petits réseaux de deux entreprises, chacun avec sa propre configuration. Ce scénario assure l'isolement des deux sites.

L'architecture réseau des deux entreprises, nommées Entreprise 1 et Entreprise 2, est conçue pour assurer une communication fluide et sécurisée entre leurs réseaux locaux. Chaque entreprise possède son propre sous-réseau privé. Entreprise 1 utilise la plage d'adresses IP 192.168.1.0/24, tandis qu'Entreprise 2 utilise la plage 192.168.77.0/24. Les deux entreprises sont équipées de routeurs frontaux qui gèrent la connexion entre les réseaux internes et Internet. Ces routeurs sont configurés pour diriger le trafic interne vers un commutateur qui, à son tour, connecte divers dispositifs tels que des clients (stations de travail) et des serveurs de messagerie. Entreprise 1 utilise un serveur de messagerie Zimbra, tandis qu'Entreprise 2

Chapitre 2

Architecture Réseau pour l'interconnexion des services de messagerie entre deux entreprises

utilise un serveur Roundcube. Les routeurs frontaux des deux entreprises se connectent à Internet via une adresse IP publique, facilitant ainsi l'interconnexion entre les deux réseaux. Cette configuration permet aux utilisateurs des deux entreprises d'envoyer et de recevoir des courriels et de partager des ressources de manière efficace et sécurisée.

2.3 Configuration des deux sites

A. Entreprise 1:

- Plage d'adresses IP : 192.168.1.0/24
- Routeur Frontal de l'Entreprise 1
 - Interface G0/0 : Connectée au commutateur interne.
 - Interface G0/1 : Connectée à Internet avec une adresse IP offerte par le serveur DHCP de l'INPT.
 - Interface Série 1: Connectée à l'entreprise 1
- Commutateur : Relie les dispositifs internes incluant :
 - Client 1 : Une station de travail ou un PC.

```
[root@entreprises1 mailserver1]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::6510:2207:629b:eafo prefixlen 64 scopeid 0x20<link>
          ether 00:0c:29:1a:9e:7d txqueuelen 1000 (Ethernet)
            RX packets 161457 bytes 228509009 (217.9 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 36906 bytes 2728140 (2.6 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
            RX packets 12807 bytes 17217893 (16.4 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 12807 bytes 17217893 (16.4 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figure 2.2 : Configuration de client 1

- Serveur de messagerie - Zimbra : Utilisé pour les services de messagerie électronique internes.

B. Entreprise 2 :

- Plage d'adresses IP : 192.168.77.0/24

Chapitre 2

Architecture Réseau pour l'interconnexion des services de messagerie entre deux entreprises

- Routeur Frontal de l'Entreprise 2
 - Interface G0/0 : Connectée au commutateur interne.
 - Interface G0/1 : Connectée à Internet avec une adresse IP offerte par le serveur DHCP de l'INPT.
 - Interface Série 1: Connectée à l'entreprise 2
- Commutateur : Relie les dispositifs internes incluant :
 - Client 2 : Une station de travail ou un PC.

```
root@mailserver:/home/imane# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:3a:7f:7e brd ff:ff:ff:ff:ff:ff
        altname enp2s1
        inet 192.168.77.147/24 brd 192.168.77.255 scope global ens33
            valid_lft forever preferred_lft forever
        inet6 fe80::20c:29ff:fe3a:7f7e/64 scope link
            valid_lft forever preferred_lft forever
root@mailserver:/home/imane#
```

Figure 2.3: Configuration du client 2

- Serveur de messagerie - RoundCube : Utilisé pour les services de messagerie électronique internes.

C. Routage interne:

Au sein de chaque entreprise, le routage interne est assuré par le protocole OSPF (Open Shortest Path First). Ce protocole de routage dynamique permet une convergence rapide en cas de changement de topologie, offrant ainsi une meilleure résilience du réseau interne. Les routeurs frontaux de chaque entreprise implémentent OSPF afin d'optimiser le routage des paquets entre les différents sous-réseaux et équipements internes.

D. Routage externe:

Pour interconnecter les deux entreprises via Internet, le routage externe est configuré avec le protocole RIP (Routing Information Protocol). Ce protocole de routage à vecteur de distance permet un échange simple d'informations de routage entre les routeurs frontaux des deux

Chapitre 2

Architecture Réseau pour l'interconnexion des services de messagerie entre deux entreprises
entreprises. Bien que moins évolutif que des protocoles comme OSPF et EIGRP, RIP reste une solution adaptée pour ce type de connexion Internet point-à-point.

E. Traduction des adresses IP:

Afin de permettre aux équipements internes d'accéder à Internet, un mécanisme de traduction d'adresses (PAT - Port Address Translation) est mis en place sur les routeurs frontaux de chaque entreprise. Ainsi, les adresses IP privées des clients 1 et 2 sont masquées derrière des adresses IP publiques, offrant une connectivité sécurisée et économique vers le réseau Internet.

Pour assurer la communication entre les deux entreprises, un mécanisme de traduction d'adresses statique (NAT statique) est configuré sur les routeurs frontaux. Ce procédé permet de mapper de manière fixe les adresses IP privées des équipements d'un site vers des adresses IP publiques visibles de l'autre site. Cela garantit une interconnexion stable et sécurisée entre les deux réseaux d'entreprise, facilitant ainsi les échanges d'informations et de données.

L'utilisation combinée du PAT pour l'accès Internet et du NAT statique pour la communication inter-sites permet de maintenir une architecture réseau évolutive et sécurisée, tout en optimisant l'utilisation des ressources IP publiques disponibles.

F. Configuration DNS:

Afin de permettre une résolution de noms efficace au sein de cette architecture réseau, des serveurs DNS sont déployés. Chaque entreprise dispose de son propre serveur de messagerie local, respectivement Zimbra pour l'Entreprise 1 et Roundcube pour l'Entreprise 2. Ces serveurs de messagerie sont configurés avec des enregistrements DNS correspondant à leurs adresses IP internes.

Chapitre 2

Architecture Réseau pour l'interconnexion des services de messagerie entre deux entreprises

```
GNU nano 2.3.1                                         File: /var/named/entreprise1.db

+$TTL 86400
@ IN SOA    ns1.entreprise1.ma. root.entreprise1.ma. (
                2024052501 ; Serial
                3600      ; Refresh
                1800      ; Retry
                604800    ; Expire
                86400     ; Minimum TTL
)
IN NS      ns1.entreprise1.ma.
IN MX 10   mail.entreprise1.ma.

ns1  IN A      192.168.34.146
mail IN A      192.168.34.146
www  IN A      192.168.34.146
@    IN A      192.168.34.146 ; This sets the A record for example.com
```

Figure 2.4 : Configuration DNS de l'entreprise 1

```
GNU nano 6.2                                         dns.db *

;
; BIND data file for local loopback interface
;

$TTL 604800
@ IN SOA    ns2.entreprise2.ma. admin.entreprise2.ma. (
                2          ; Serial
                604800    ; Refresh
                86400    ; Retry
                2419200  ; Expire
                604800 ) ; Negative Cache TTL

        IN NS      ns2.entreprise2.ma.

ns2  IN A      192.168.77.147
mail IN A      192.168.77.147
@    IN MX    10 mail.entreprise2.ma
```

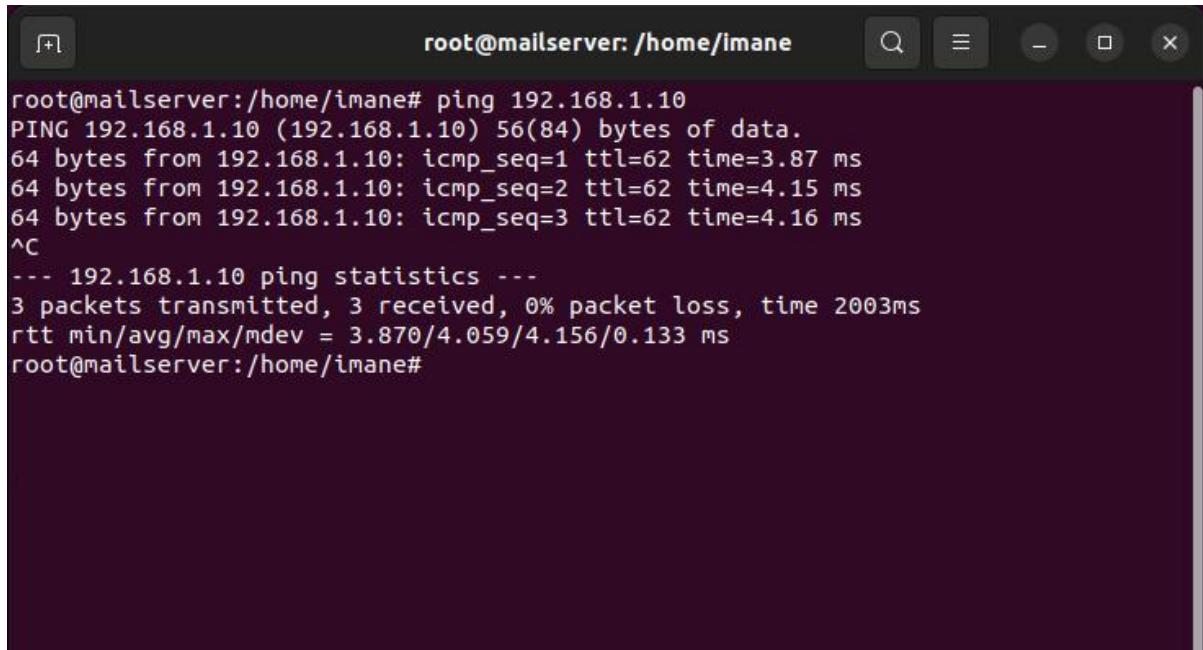
Figure 2.5 : Configuration DNS de l'entreprise 2

2.4 Tests de connectivité

G. Pings entre 2 machines distantes.

```
mailserver1@entreprise1:/home/mailserver1 - x
File Edit View Search Terminal Help
[root@entreprise1 mailserver1]# ping 192.168.77.3
PING 192.168.77.3 (192.168.77.3) 56(84) bytes of data.
64 bytes from 192.168.77.3: icmp_seq=1 ttl=126 time=4.17 ms
64 bytes from 192.168.77.3: icmp_seq=2 ttl=126 time=3.94 ms
64 bytes from 192.168.77.3: icmp_seq=3 ttl=126 time=4.28 ms
^C
--- 192.168.77.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 3.942/4.133/4.285/0.142 ms
[root@entreprise1 mailserver1]#
```

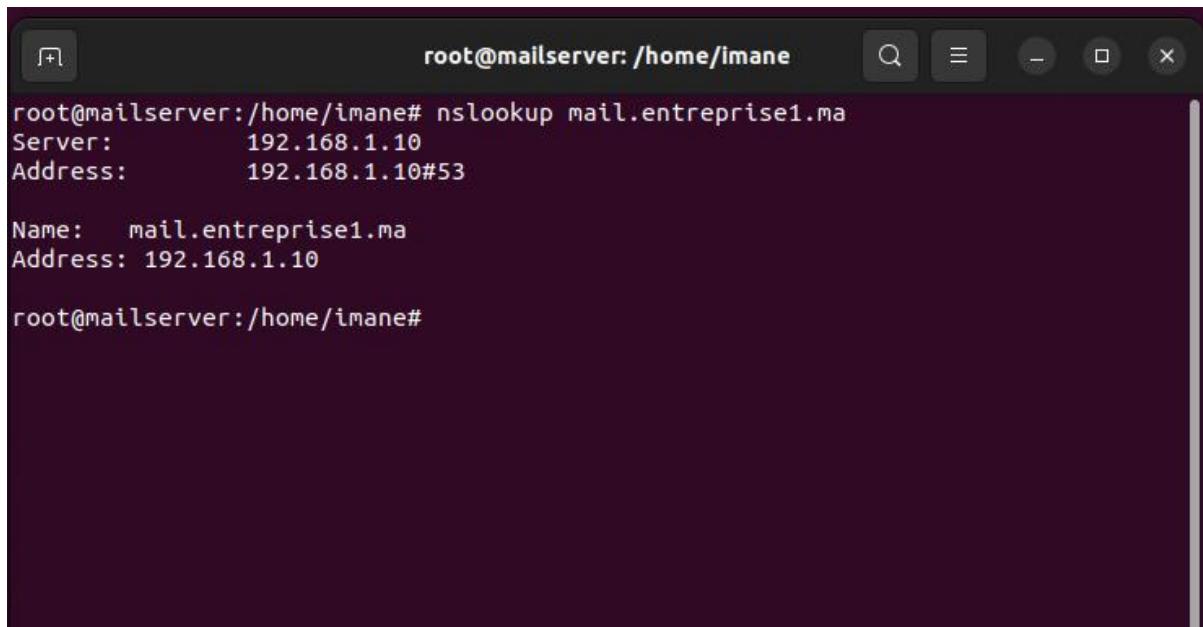
Figure 2.6 : Ping allant du serveur 2 vers le serveur 1



```
root@mailserver:/home/imane# ping 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=62 time=3.87 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=62 time=4.15 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=62 time=4.16 ms
^C
--- 192.168.1.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 3.870/4.059/4.156/0.133 ms
root@mailserver:/home/imane#
```

Figure 2.7 : Ping allant du serveur 1 vers le serveur 2

H. Test de la résolution des noms des serveurs.



```
root@mailserver:/home/imane# nslookup mail.entreprise1.ma
Server:      192.168.1.10
Address:     192.168.1.10#53

Name:   mail.entreprise1.ma
Address: 192.168.1.10

root@mailserver:/home/imane#
```

Figure 2.8 : Résolution du nom de serveur de messagerie de l'entreprise 1

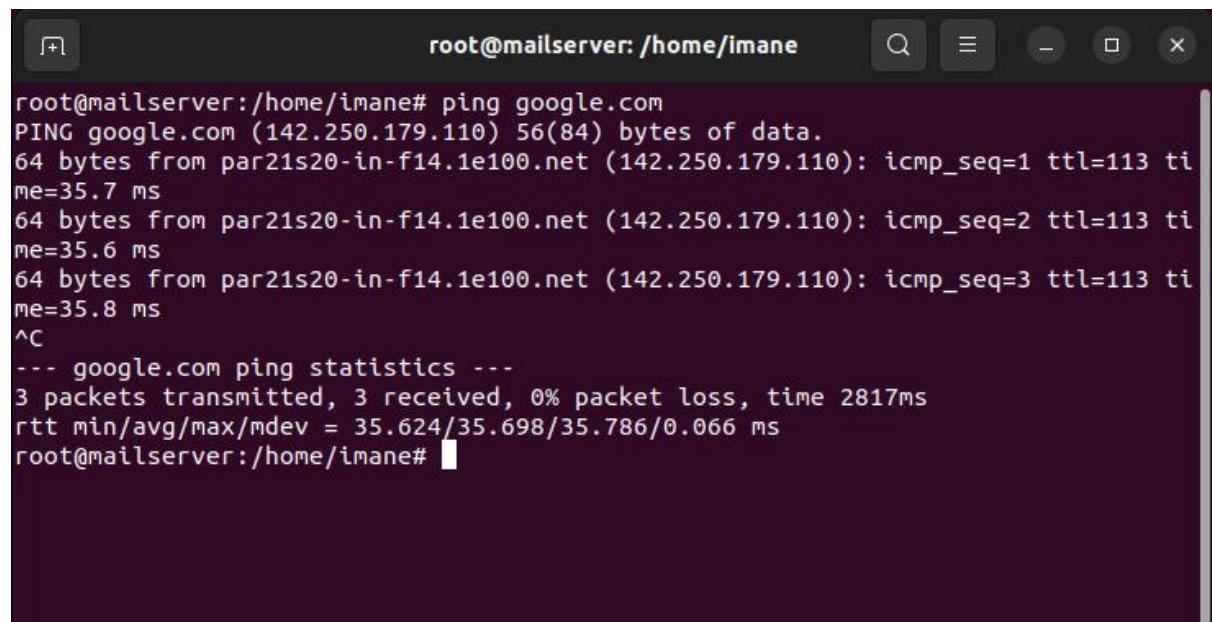
I. Accès Internet.

Chapitre 2

Architecture Réseau pour l'interconnexion des services de messagerie entre deux entreprises

```
[root@entreprise1 mailserver1]# ping google.com
PING google.com (142.250.179.110) 56(84) bytes of data.
64 bytes from par21s20-in-f14.1e100.net (142.250.179.110): icmp_seq=1 ttl=113 time=35.1 ms
64 bytes from par21s20-in-f14.1e100.net (142.250.179.110): icmp_seq=2 ttl=113 time=35.0 ms
64 bytes from par21s20-in-f14.1e100.net (142.250.179.110): icmp_seq=3 ttl=113 time=35.1 ms
64 bytes from par21s20-in-f14.1e100.net (142.250.179.110): icmp_seq=4 ttl=113 time=34.7 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 4426ms
rtt min/avg/max/mdev = 34.772/35.040/35.197/0.208 ms
[root@entreprise1 mailserver1]#
```

Figure 2.9 : Accès à l'internet à partir du serveur 1



The screenshot shows a terminal window with a dark background and light-colored text. The title bar reads "root@mailserver: /home/imane". The command entered was "ping google.com". The output shows four successful ping requests to the IP 142.250.179.110, with round-trip times ranging from 34.772 to 35.197 ms. The terminal window has standard Linux-style window controls at the top.

```
root@mailserver:/home/imane# ping google.com
PING google.com (142.250.179.110) 56(84) bytes of data.
64 bytes from par21s20-in-f14.1e100.net (142.250.179.110): icmp_seq=1 ttl=113 time=35.7 ms
64 bytes from par21s20-in-f14.1e100.net (142.250.179.110): icmp_seq=2 ttl=113 time=35.6 ms
64 bytes from par21s20-in-f14.1e100.net (142.250.179.110): icmp_seq=3 ttl=113 time=35.8 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2817ms
rtt min/avg/max/mdev = 35.624/35.698/35.786/0.066 ms
root@mailserver:/home/imane#
```

Figure 2.10 : Accès à l'internet à partir du serveur 2

Chapitre 3

Réalisation de service de messagerie sécurisée

Dans l'ère numérique actuelle, la communication par messagerie électronique est devenue un outil indispensable. Cependant, cette popularité s'accompagne de défis majeurs, notamment la sécurité des messages et la prolifération des spams. Les spams non seulement encombrent les boîtes de réception, mais peuvent également être vecteurs de menaces telles que le phishing et les logiciels malveillants. Pour répondre à ces préoccupations, il est crucial de développer un service de messagerie sécurisée capable de détecter et de filtrer efficacement les spams.

Ce projet vise à créer une solution de messagerie sécurisée en mettant l'accent sur la détection et l'élimination des spams. En utilisant les règles de filtrage sophistiqués, ce service offrira une protection accrue contre les courriels indésirables et malveillants.

3.1 Choix de la plateforme messagerie :

3.1.1 Introduction :

Webmail est une interface informatique qui permet de lire, gérer et envoyer des courriers électroniques depuis un navigateur Internet. Un Webmail est accessible facilement à partir d'une url, et il est considéré comme un logiciel en mode SAAS (Software As A Service). [3.1]

De nombreux utilisateurs de logiciels de messagerie traditionnels configurent leurs serveurs d'envoi (SMTP) et de réception (POP3 ou IMAP) sans activer le chiffrement de la connexion. Si un Webmail est bien configuré, nous aurons deux avantages sur le plan de la sécurité qui sont : les messages ne sont pas stockés sur la machine de l'utilisateur et ils sont toujours consultable selon une connexion SSL sécurisée.

3.1.2 Critères de sélection

Plusieurs critères ont été considérés pour faire notre choix en Webmail afin de répondre aux exigences de notre cahier de charge. Le choix final était l'utilisation de Roundcube pour comprendre en détail les composants d'un système de messagerie et Zimbra en tant qu'une interface complète accessible aussi bien depuis un serveur qu'un client.

3.1.2.1 Présentation de service de messagerie Zimbra

Zimbra est un outil offrant une solution de messagerie électronique performante et polyvalente. Compatible avec les ordinateurs Windows, Apple et Linux, il s'adapte également aux appareils mobiles Android, iOS et Windows Mobile. [3.2]



Figure 3.1 : Webmail Zimbra

Parmi les principales caractéristiques de Zimbra, nous trouvons :

- Une gestion totale des mails : Les courriels peuvent être automatiquement classés en fonction de leur type, de la conversation à laquelle ils appartiennent, ou du contenu spécifique qu'ils contiennent. Les utilisateurs peuvent trier leurs courriers par date, expéditeur, sujet, ou par des critères personnalisés et ils peuvent programmer l'envoi de courriels pour une date et une heure spécifique.
- Classification par filtres réseaux sociaux et commerciaux : Les courriels commerciaux, réseaux sociaux, spams et publicités sont automatiquement triés dans des catégories dédiées, pour une boîte de réception plus claire.
- Le management de l'agenda partagé : Les utilisateurs ont droit d'accéder à un agenda partagé sur plateforme.
- Sécurité : Zimbra offre des options avancées de sécurité avec des protocoles SSL/TLS, des fonctionnalités de chiffrement des emails, une authentification multi-facteurs, et des politiques de sécurité granulaires.

3.1.2.2 Présentation de service de messagerie Roundcube

Roundcube, proposé par le géant français OVH, se distingue comme un Webmail open-source performant et apprécié par de nombreux utilisateurs. Fonctionnant sous le protocole IMAP et la technologie Ajax, il offre une expérience utilisateur fluide et intuitive. Toutes les plateformes qui supportent son langage de programmation peuvent l'ouvrir et un PHP 4.5 ou une version supérieure est recommandée pour son bon fonctionnement. [3.3]



Figure 3.2 : Webmail Roundcube

Parmi les principaux avantages de Roundcube, nous trouvons :

- Simplicité et Légèreté : Interface utilisateur simple et légère, facile à déployer et à gérer.
- Cryptage sécurisé des mails : supporte SSL/TLS pour les connexions sécurisées, intègre des mécanismes de protection contre les attaques XSS et CSRF, et prend en charge l'authentification à deux facteurs via des plugins.
- Fonctionnalités avancées de gestion des emails : Roundcube permet de faire un triage avancé, répondre automatiquement en absence de son utilisateur ainsi qu'authentifier ses emails et protéger ses communications avec des signatures numériques S/MIME.

En conclusion, nous avons choisi de travailler avec Zimbra et Roundcube comme clients de messagerie en raison de leur facilité d'intégration, de leur robustesse en matière de sécurité et de la simplicité de leurs interfaces utilisateur. Zimbra est reconnu pour ses capacités de collaboration et sa forte sécurité, tandis que Roundcube est apprécié pour son interface légère et facile à utiliser. Ces deux solutions offrent des fonctionnalités avancées tout en restant accessibles et modulables, répondant ainsi aux besoins variés des utilisateurs.

3.2 Mise en œuvre des éléments de base

Notre architecture comporte deux entreprises distinctes. Une configuré avec Roundcube et ces dépendances et l'autre avec Zimbra. En premier lieu après la mise en place de l'architecture réseau et la configuration de serveur DNS, nous allons entamer l'implémentation des services nécessaires pour chaque site.

3.2.1 Installation et configuration de zimbra

Nous allons passer à l'installation du logiciel Zimbra. Une fois l'installation terminée, nous allons extraire les fichiers avec la commande tar comme suit :

```
[root@mail mailserver1]# wget https://files.zimbra.com/downloads/8.8.15_GA/zcs-8.8.15_GA_3869.RHEL7_64.20190918004220.tgz
--2024-05-25 17:53:24-- https://files.zimbra.com/downloads/8.8.15_GA/zcs-8.8.15_GA_3869.RHEL7_64.20190918004220.tgz
Resolving files.zimbra.com (files.zimbra.com)... 18.66.115.55
Connecting to files.zimbra.com (files.zimbra.com)|18.66.115.55|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 255807770 (244M) [application/x-tar]
Saving to: 'zcs-8.8.15_GA_3869.RHEL7_64.20190918004220.tgz'

100%[=====] 255,807,770 7.61MB/s   in 28s

2024-05-25 17:53:54 (8.57 MB/s) - 'zcs-8.8.15_GA_3869.RHEL7_64.20190918004220.tgz' saved [255807770/255807770]

[root@mail mailserver1]# cd zcs-8.8.15_GA_3869.RHEL7_64.20190918004220
bash: cd: zcs-8.8.15_GA_3869.RHEL7_64.20190918004220: No such file or directory
[root@mail mailserver1]# ls
Desktop  Downloads  Pictures  Templates  zcs-8.8.15_GA_3869.RHEL7_64.20190918004220.tgz
Documents  Music  Public  Videos
[root@mail mailserver1]# tar xvf zcs-8.8.15_GA_3869.RHEL7_64.20190918004220.tgz
zcs-8.8.15_GA_3869.RHEL7_64.20190918004220/README.txt
zcs-8.8.15_GA_3869.RHEL7_64.20190918004220/docs/
zcs-8.8.15_GA_3869.RHEL7_64.20190918004220/docs/en_US/
zcs-8.8.15_GA_3869.RHEL7_64.20190918004220/docs/en_US/Migration_Exch_Admin.pdf
zcs-8.8.15_GA_3869.RHEL7_64.20190918004220/docs/en_US/MigrationWizard.pdf
zcs-8.8.15_GA_3869.RHEL7_64.20190918004220/docs/en_US/User_Instructions_for_ZCS_Import_Wizard.pdf
zcs-8.8.15_GA_3869.RHEL7_64.20190918004220/docs/en_US/RNZCS0_2005Beta.pdf
zcs-8.8.15_GA_3869.RHEL7_64.20190918004220/docs/en_US/Fedora_Server_Config.pdf
zcs-8.8.15_GA_3869.RHEL7_64.20190918004220/docs/en_US/OSmultiserverinstall.pdf
```

Figure 3.4 : Installation de Zimbra

L’installation finale est obtenue avec la commande ./install.sh

```
[root@mail zcs-8.8.15_GA_3869.RHEL7_64.20190918004220]# ./install.sh
Operations logged to /tmp/install.log.cTw8bAeM
Checking for existing installation...
zimbra-drive...NOT FOUND
zimbra-imapd...NOT FOUND
zimbra-patch...NOT FOUND
zimbra-mta-patch...NOT FOUND
zimbra-proxy-patch...NOT FOUND
zimbra-license-tools...NOT FOUND
zimbra-license-extension...NOT FOUND
zimbra-network-store...NOT FOUND
zimbra-network-modules-ng...NOT FOUND
zimbra-chat...NOT FOUND
zimbra-talk...NOT FOUND
zimbra-ldap...NOT FOUND
zimbra-logger...NOT FOUND
zimbra-mta...NOT FOUND
zimbra-dnscache...NOT FOUND
zimbra-snmp...NOT FOUND
```

Figure 3.5 : Installation finale de Zimbra

Une fois l’installation est terminée, nous avons configuré le mot de passe d’administrateur puis nous allons vérifier le statut de zimbra avec la commande « zmcontrol status » qui va donner tous les modules sont en mode running.

Pour accéder à l’interface web de Zimbra, nous allons taper sur la page web <https://mail.entreprise1.ma:7071> et nous obtenons l’interface suivante :

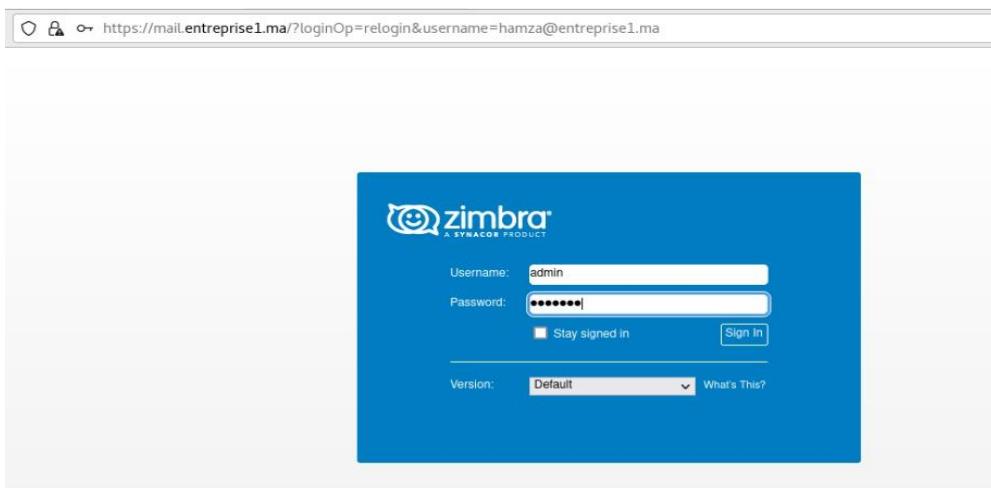


Figure 3.6 : Webmail page de Zimbra

Une fois connectée, nous avons accès à notre boite mail et nous pouvons envoyer des messages aux utilisateurs que nous avons créé. A titre indicatif, l'utilisateur hamza a envoyé à l'utilisateur nezha un message. Le mail a été scanné par le logiciel antivirus amavis sur le serveur entreprise1.ma mais au niveau de X_spam status, aucune mention de réussite ou d'échec de tests DKIM, SPF, ou DMARC. Cela signifie que l'email peut potentiellement être considéré comme moins sécurisé et plus susceptible d'être marqué comme spam ou phishing.

```

Return-Path: <hamza@entreprise1.ma>
Received: from mail.entreprise1.ma (LHL0 mail.entreprise1.ma)
          (192.168.34.146) by mail.entreprise1.ma with LMTP; Sat, 25 May 2024
          20:06:20 +0100 (WEST)
Received: from localhost (localhost [127.0.0.1])
          by mail.entreprise1.ma (Postfix) with ESMTP id A0EC4109A701
          for <nezha@entreprise1.ma>; Sat, 25 May 2024 20:06:20 +0100 (+01)
X-Spam-Flag: NO
X-Spam-Score: -1.009
X-Spam-Level:
X-Spam-Status: No, score=-1.009 required=6.6 tests=[ALL_TRUSTED=-1,
HTML_MESSAGE=0.001, T_SCC_BODY_TEXT_LINE=-0.01]
autolearn=ham autolearn_force=no
Received: from mail.entreprise1.ma ([127.0.0.1])
          by localhost (mail.entreprise1.ma [127.0.0.1]) (amavis, port 10032)
          with ESMTP id W3b8uC0Vte4D for <nezha@entreprise1.ma>;
          Sat, 25 May 2024 20:06:15 +0100 (+01)
Received: from localhost (localhost [127.0.0.1])
          by mail.entreprise1.ma (Postfix) with ESMTP id C3D4714159C9
          for <nezha@entreprise1.ma>; Sat, 25 May 2024 20:06:14 +0100 (+01)
X-Virus-Scanned: amavis at entreprise1.ma
Received: from mail.entreprise1.ma ([127.0.0.1])
          by localhost (mail.entreprise1.ma [127.0.0.1]) (amavis, port 10026)
          with ESMTP id QnEnt-nBKhEy for <nezha@entreprise1.ma>;
          Sat, 25 May 2024 20:06:10 +0100 (+01)
Received: from mail.entreprise1.ma (mail.entreprise1.ma [192.168.34.146])
          by mail.entreprise1.ma (Postfix) with ESMTP id 28355109A701
          for <nezha@entreprise1.ma>; Sat, 25 May 2024 20:06:10 +0100 (+01)
Date: Sat, 25 May 2024 20:06:09 +0100 (WEST)
From: Hamza Mouisset <hamza@entreprise1.ma>
To: Nezha Hinaje <nezha@entreprise1.ma>
Message-ID: <885464528.352.171666396925.JavaMail.zimbra@entreprise1.ma>
Subject: test
MIME-Version: 1.0
Content-Type: multipart/alternative;
boundary="=fd2d75b5-f891-4fcf-83d9-6d1e469ee867"
X-Originating-IP: [192.168.34.146]
X-Mailer: Zimbra 8.8.15_GA_4581 (ZimbraWebClient - FF115 (Linux)/8.8.15_GA_4581)
Thread-Index: m4LWt/y+z2BMwzgemb0qG/L8M10U3w==
Thread-Topic: test
--=_fd2d75b5-f891-4fcf-83d9-6d1e469ee867
Content-Type: text/plain; charset=utf-8
Content-Transfer-Encoding: 7bit
vfae
--=_fd2d75b5-f891-4fcf-83d9-6d1e469ee867
Content-Type: text/html; charset=utf-8

```

Figure 3.7 : Message envoyé sans DKIM, SPF et DMARC

Pour but de protéger notre domaine et améliorer la délivrabilité de nos emails, nous avons configuré DNS avec DKIM, SPF et Dmarc pour les raisons suivantes :

- DKIM (DomainKeys Identified Mail) : Permet à notre domaine de signer numériquement nos emails, prouvant ainsi leur authenticité et il empêche les spameurs d'usurper notre identité et d'envoyer des emails en notre nom.
- SPF (Sender Policy Framework) : Spécifie quels serveurs de messagerie sont autorisés à envoyer des emails pour notre domaine et bloque les tentatives d'usurpation d'identité en empêchant l'envoi d'emails depuis des serveurs non autorisés.
- DMARC (Domain-based Message Authentication, Reporting & Conformance): Indique aux serveurs de messagerie quoi faire avec les emails non authentifiés par DKIM ou SPF ce qui permet de réduire le spam.

```
GNU nano 2.3.1                               File: /var/named/entreprise1.db

$TTL 86400
@ IN SOA    ns1.entreprise1.ma. root.entreprise1.ma. (
                2024052501 ; Serial
                3600   ; Refresh
                1800   ; Retry
                604800 ; Expire
                86400  ; Minimum TTL
)
IN NS      ns1.entreprise1.ma.
IN MX 10  mail.entreprise1.ma.

ns1 IN A      192.168.34.146
mail IN A      192.168.34.146
www IN A      192.168.34.146
@ IN A      192.168.34.146 ; This sets the A record for example.com

@ TXT "v=spf1 a mx -all"
_dmarc TXT "v=DMARC1; pnone; fo=1; rua=mailto:authority@entreprise1.ma; ruf=mailto:authority@entreprise1.ma" 8250020E-EF81-11E8-BB5B-4520489C3827..domainkey.entreprise1.ma. IN TXT ( "v=DKS
C590E40B-1ACA-11EF-9374-4B665D00840B..domainkey IN      TXT  ( "v=DKIM1; kersa;
" "p=MIIBJANBgkqhkiG9w0BAQEFAAOCAQABMIIBgkKCAQEAvu6QoEmKcavwB50Cb6mRzYgrAfEuJfZ/d7r9LVCrzih+W9Mnkz0FeWZTV0ah+uvzVb/lJxv27ykSzqssnO125xz2lgVt1EBn0Mv+sksa/azHkQHBRgI125t6wi$" "VrqipMsj1336r5T0Bg3ufXdsXetTf8VsQmLSqJcBfJK2gdg7TDyJNNTwErfkeUqy96Jlhecz5zTuaHmpTln3+Wh0MxNbKCuT63E0nc/gIig3YasryZtGFYiwb5pN4yiFJfQIDAQAB" ) ; ----- DKIM key C590E40B-1ACAS
"
```

Figure 3.8 : l'ajout de record DKIM, SPF et DMARC dans le fichier de la zone entreprise1.ma

Nous avons de nouveau testé l'envoi de message d'utilisateur hamza vers nezha. Le message inclut des tests DKIM_VALID avec une clé de 2048 bits, ce qui renforce la crédibilité de l'email et nous obtenons :

```
Return-Path: <hamza@entreprise1.ma>
Received: from mail.entreprise1.ma (LHL0 mail.entreprise1.ma)
(192.168.34.146) by mail.entreprise1.ma with LMTP; Sat, 25 May 2024
20:58:32 +0100 (WEST)
Received: from localhost ([127.0.0.1])
by mail.entreprise1.ma (Postfix) with ESMTP id C949D14159B3
for <nezha@entreprise1.ma>; Sat, 25 May 2024 20:58:32 +0100 (+01)
X-Spam-Flag: NO
X-Spam-Score: -1.209
X-Spam-Level:
X-Spam-Status: No, score=-1.209 required=6.6 tests=[ALL_TRUSTED=-1,
DKIM_SIGNED=0.1, DKIM_VALID=-0.1, DKIM_VALID_AU=-0.1, DKIM_VALID_EF=-0.1,
HTML_MESSAGE=0.001, T_SCC_BODY_TEXT_LINE=-0.01]
autolearn-ham autolearn_force=no
Authentication-Results: mail.entreprise1.ma (amavis); dkim=pass (2048-bit key)
header.d=entreprise1.ma
Received: from mail.entreprise1.ma ([127.0.0.1])
by localhost (mail.entreprise1.ma [127.0.0.1]) (amavis, port 10032)
with ESMTP id Fax21NOUGKho for <nezha@entreprise1.ma>;
Sat, 25 May 2024 20:58:30 +0100 (+01)
Received: from localhost ([localhost [127.0.0.1]])
by mail.entreprise1.ma (Postfix) with ESMTP id A5CD41115E38
for <nezha@entreprise1.ma>; Sat, 25 May 2024 20:58:30 +0100 (+01)
DKIM-Filter: OpenDKIM Filter v2.10.3 mail.entreprise1.ma A5CD41115E38
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=entreprise1.ma;
s=C590E40B-1ACA-11EF-9374-4B665D00840B; t=1716667110;
bh=+lAGULas/QJxn/96Lnw+NNOnjH89vFkv+cNmCtIDtA=;
h=Date:From:To:Message-ID:MIME-Version;
b=JKE70457fuE687w9siRaee09R4A7]FUSqrwEAdagdu9fEamhgh27W9kSHpxMd6xpPs0
uXqb+buL5IJHFFkrRcdMVM07RcM65PKNUFMci0ioQAUfaSSGPi5E6sfELXKnMFBn5
nZNRF4tmEwhfVZ8upt0rE6HJhINboA7a9FGYr+stBo78tV+R9I0I7BnMMye7FF914
20pXL2xqNUpvUdw6SuzQl0MjNjRbsTfkUrVI00lbITmg5rPQn7g0z610D0HBr2y2
0+9mFVTdA1C65IUkBZLjo4czVwy8BPXZkoHtsDVkwI9aNpRsDhNzhlxUciXRES93B
8g+RCW3WN3BNg==

X-Virus-Scanned: amavis at entreprise1.ma
Received: from mail.entreprise1.ma ([127.0.0.1])
by localhost (mail.entreprise1.ma [127.0.0.1]) (amavis, port 10026)
with ESMTP id UpjAhU3juFei for <nezha@entreprise1.ma>;
Sat, 25 May 2024 20:58:30 +0100 (+01)
Received: from mail.entreprise1.ma (mail.entreprise1.ma [192.168.34.146])
by mail.entreprise1.ma (Postfix) with ESMTP id 2208614159B3
for <nezha@entreprise1.ma>; Sat, 25 May 2024 20:58:30 +0100 (+01)
Date: Sat, 25 May 2024 20:58:29 +0100 (WEST)
From: Hamza Mouisset <hamza@entreprise1.ma>
To: Nezha Hinaje <nezha@entreprise1.ma>
Message-ID: <15309932333.359.1716667109862.JavaMail.zimbra@entreprise1.ma>
Subject: secure test
MIME-Version: 1.0
```

Figure 3.9 : Envoi du mail avec DKIM, SPF et DMARC

3.2.2 Installation et configuration d'une solution de messagerie complète basée sur Postfix, Dovecot et Roundcube

L'infrastructure des emails constitue l'épine dorsale du système de communication par email, englobant divers composants qui travaillent ensemble pour assurer la livraison fiable et efficace des emails. Comprendre l'infrastructure des emails est essentiel pour appréhender les mécanismes sous-jacents du routage et de la livraison des emails ce qui est cité au premier paragraphe. Nous allons se basé sur cette figure pour configurer le nôtre. [3.4]

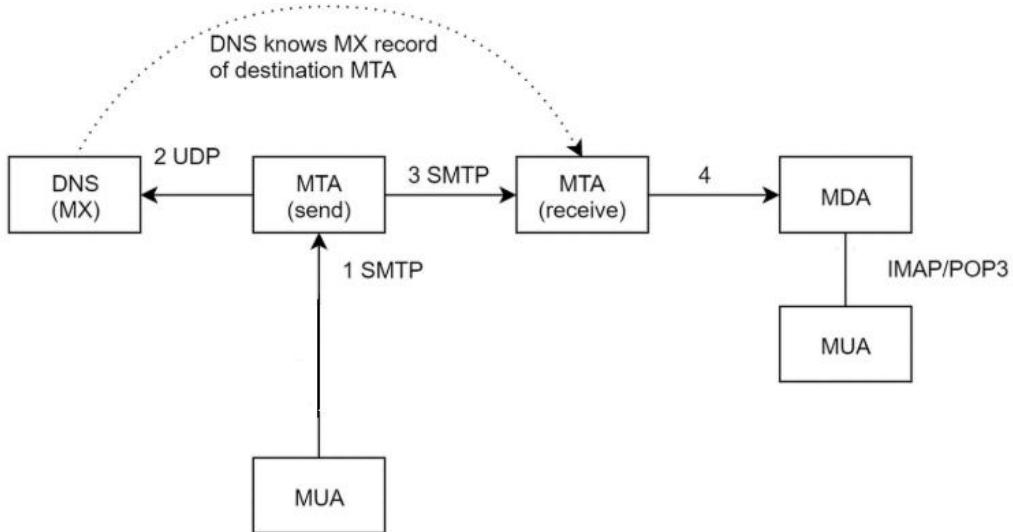


Figure 3.10 : Infrastructure de messagerie

3.2.2.1 Installation et Configuration de MTA

Postfix est un serveur de messagerie (MTA - Mail Transfer Agent) utilisé pour envoyer et recevoir des emails. Il assure les fonctions suivantes :

- Envoi d'Emails : Lorsque nous allons composer un email dans Roundcube et appuyer sur "Envoyer", Roundcube transmet cet email à Postfix, qui gère l'acheminement et la livraison de l'email au destinataire.
- Réception d'Emails : Postfix peut également être configuré pour recevoir des emails provenant d'autres serveurs de messagerie et les stocker pour que les utilisateurs puissent les lire.

En somme, Postfix est un serveur de messagerie électronique et un logiciel libre remplaçant idéalement toutes sortes de solutions moins libres grâce à ses avantages citons :

- Postfix utilise plusieurs niveaux de défense afin de protéger le système de toute intrusion. Chaque programme est enfermé dans sa cage (chrooté).
- Adapté à de gros besoins,
- Plus simple à configurer que Sendmail

- Relativement sécurisé avec anti-Spam
 - Maintenance aisée

Pour les raisons citées précédemment, nous avons fait le choix de Postfix que sendmail.

Après la configuration de DNS pour l'entreprise 2, Nous allons passer à l'installation des modules Apache2, mariadb,php Essentiels pour l'implémentations de notre solution de messagerie en exécutant la commande suivante :

```
root@mailserver:~# apt install libapache2-mod-php mariadb-server php-fpm php-imap php-mbstring php-mysql php-json php-curl php-zip php-xml php-bz2 php-intl php-gmp -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
galera-4 libapache2-mod-php8.1 libc-client2007e libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl
libdaxdbi libdbd-mysql-perl libdbi-perl libfcgi-bin libfcgi-perl libfcgioldlib libhtml-template-perl
libmariadb3 libmysqlclient21 libndctl6 libonig5 libpmm1 libsnappy1v5 liburing2 libzip4 mariadb-client-10.6
0 upgraded, 14 new, 0 to remove and 0 not upgraded.
```

Figure 3.11 : Installation des modules nécessaires

Afin de vérifier que l'installation a abouti et que mariadb a bien été installé, nous avons utilisés les commandes suivantes :

```
root@mailserver:~# systemctl status mariadb
● mariadb.service - MariaDB 10.6.16 database server
    Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
    Active: active (running) since Sun 2024-05-26 12:14:44 UTC; 1min 8s ago
      Docs: man:mariadb(8)
             https://mariadb.com/kb/en/library/systemd/
   Process: 17948 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysqld (code=exited, status=0)
   Process: 17949 ExecStartPre=/bin/sh -c systemctl unset-environment WSREP START POSITION (code=exited, status=>
```

Figure 3.12 : Vérification du statut mariadb

Pour déterminer la version de PHP en cours d'utilisation, nous exécutons la commande suivante

```
root@mailserver:~# php -v
PHP 8.1.2-1ubuntu2.17 (cli) (built: May 1 2024 10:10:07) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.17. Copyright (c). by Zend Technologies
```

Figure 3.13 : Version de php

Ensuite, nous procérons à l'installation de comme suit :

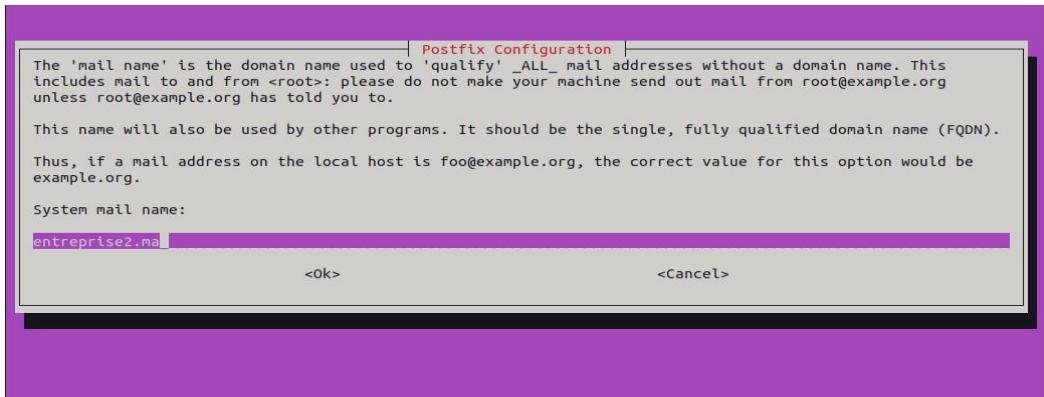


Figure 3.14 : Configuration de Postfix

Après avoir vérifier le statut de postfix , on procède à la création d'un environnement sécurisé pour la gestion des boites à lettres virtuelles dans Postfix, nous allons créer un groupe et un utilisateur entreprise 2 et nous allons créer un répertoire pour stocker ces lettres qui sera dans var/virtual_mail_box et nous allons donner les permissions de lecture et d'écriture et d'exécution que pour le groupe d'entreprise 2. Enfin nous allons changer le propriétaire de ce répertoire au groupe entreprise 2.

```
root@mailserver:~# groupadd -g 5000 entreprise2
root@mailserver:~# useradd -g entreprise2 -u 5000 entreprise2 -s /sbin/nologin
root@mailserver:~# mkdir /var/virtual_mail_box
root@mailserver:~# chmod -R 770 /var/virtual_mail_box
root@mailserver:~# chown -R entreprise2:entreprise2 /var/virtual_mail_box
chown: invalid user: 'entreprise2:entreprise2'
root@mailserver:~# useradd -g entreprise2 -u 5000 entreprise2 -s /sbin/nologin
useradd: UID 5000 is not unique
root@mailserver:~# userdel entreprise2
root@mailserver:~# useradd -g entreprise2 -u 5000 entreprise2 -s /sbin/nologin
root@mailserver:~# chown -R entreprise2:entreprise2 /var/virtual_mail_box
root@mailserver:~# ll /var/virtual_mail_box/
total 8
drwxrwx--- 2 entreprise2 entreprise2 4096 May 26 12:29 .
drwxr-xr-x 16 root      root        4096 May 26 12:29 ..
root@mailserver:~#
```

Figure 3.15 : Crédation du groupe et utilisateur qui aura accès au répertoire var/virtual_mail_box

Maintenant, nous allons procéder à la création de la base de données postfixadmin en se connectant au serveur de la base de données MySQL qui sera nommée postfixadmin_db.

```
root@mailserver:~# mysql -uroot -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.6.16-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE postfixadmin_db;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> CREATE USER postfixadmin_user@localhost IDENTIFIED BY 'postfixadmin_PWD';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> GRANT ALL ON postfixadmin_db.* TO postfixadmin_user@localhost;
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)
```

Figure 3.16 : Création de base de données postfixadmin_db

Pour assurer la connexion entre la base de données et Postfix, nous allons configurer le fichier /etc/postfix/main.cf pour spécifier les paramètres de connexion MySQL, tels que le nom d'hôte du serveur MySQL et le nom d'utilisateur MySQL et nous allons créer un répertoire query afin de configurer des mappages MySQL pour les alias virtuels, les domaines et les boîtes aux lettres dans Postfix.

```
root@mailserver:~# cd /etc/postfix
root@mailserver:/etc/postfix# ls
dynamicmaps.cf      main.cf      makedefs.out  master.cf.proto  postfix-files.d  post-install
dynamicmaps.cf.d    main.cf.proto  master.cf    postfix-files   postfix-script  sasl
root@mailserver:/etc/postfix# mkdir query
root@mailserver:/etc/postfix# cd query
root@mailserver:/etc/postfix/query# nano mysql_virtual_alias_domain_catchall_maps.cf
root@mailserver:/etc/postfix/query# nano mysql_virtual_alias_domain_mailbox_maps.cf
root@mailserver:/etc/postfix/query# nano mysql_virtual_alias_domain_maps.cf
root@mailserver:/etc/postfix/query# nano mysql_virtual_alias_maps.cf
root@mailserver:/etc/postfix/query# nano mysql_virtual_alias_maps.cf
root@mailserver:/etc/postfix/query# nano mysql_virtual_domains_maps.cf
root@mailserver:/etc/postfix/query# nano mysql_virtual_mailbox_limit_maps.cf
root@mailserver:/etc/postfix/query# nano mysql_virtual_mailbox_maps.cf
root@mailserver:/etc/postfix/query# ■
```

Figure 3.17 : Connexion entre MySQL et Postfix

3.2.2.2 installation et Configuration de MDA

En tant que MDA (Mail Delivery Agent), Dovecot reçoit les emails à partir d'un MTA (Postfix dans notre cas) et les stocke dans les boîtes de réception des utilisateurs. Dovecot fonctionne comme un serveur IMAP/POP3. IMAP permet de gérer les courriels directement sur le serveur, offrant une synchronisation en temps réel entre plusieurs appareils, tandis que POP3 télécharge les emails sur l'appareil de l'utilisateur et les supprime du serveur par défaut.

On installe le package de dovecot avec la commande : apt install dovecot-imapd dovecot-pop3d

```

root@mailserver:~# systemctl start postfix
root@mailserver:~# systemctl start dovecot
root@mailserver:~# systemctl enable dovecot
Synchronizing state of dovecot.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable dovecot
^[[A^[[Broot@mailserver:~# systemctl enable postfix
Synchronizing state of postfix.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable postfix
root@mailserver:~# systemctl status postfix
● postfix.service - Postfix Mail Transport Agent
   Loaded: loaded (/lib/systemd/system/postfix.service; enabled; vendor preset: enabled)
     Active: active (exited) since Sun 2024-05-26 12:20:50 UTC; 2min 0s ago
       Docs: man:postfix(1)
      Main PID: 19254 (code=exited, status=0/SUCCESS)
        CPU: 3ms

May 26 12:20:50 mailserver systemd[1]: Starting Postfix Mail Transport Agent...
May 26 12:20:50 mailserver systemd[1]: Finished Postfix Mail Transport Agent.
root@mailserver:~# systemctl status dovecot
● dovecot.service - Dovecot IMAP/POP3 email server
   Loaded: loaded (/lib/systemd/system/dovecot.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2024-05-26 12:21:29 UTC; 1min 32s ago
       Docs: man:dovecot(1)
          https://doc.dovecot.org/
      Main PID: 22676 (dovecot)
        CPU: 0ms

```

Figure 3.18 : Lancement de Postfix et Dovecot

Nous allons procéder à l'ouverture des ports POP3 et IMAP afin de pouvoir accéder à nos emails depuis n'importe quel appareil connecté et avoir une synchronisation des emails. Le port standard pour POP3 est 110 et pour IMAP c'est 143.

```

root@mailserver:~# ufw allow 25
Skipping adding existing rule
Skipping adding existing rule (v6)
root@mailserver:~# ufw allow 143
Skipping adding existing rule
Skipping adding existing rule (v6)
root@mailserver:~# ufw allow 110
Skipping adding existing rule
Skipping adding existing rule (v6)
root@mailserver:~# ufw reload
Firewall reloaded
root@mailserver:~# ufw status
Status: active

To           Action    From
--           ----     ---
25           ALLOW     Anywhere
110          ALLOW     Anywhere
143          ALLOW     Anywhere
25 (v6)      ALLOW     Anywhere (v6)
110 (v6)     ALLOW     Anywhere (v6)
143 (v6)     ALLOW     Anywhere (v6)

```

Figure 3.19 : Autorisation des ports liés à POP 3 et IMAP

Après avoir vérifié que le status de Postfix est actif, nous procédons à la configuration de Dovecot en éditant les fichiers suivants et en donnant les droits au propriétaire entreprise 2 comme suit :

```

root@mailserver:/etc/dovecot/conf.d# nano 10-ssl.conf
root@mailserver:/etc/dovecot/conf.d# nano 15-lda.conf
root@mailserver:/etc/dovecot/conf.d# nano 10-master.conf
root@mailserver:/etc/dovecot/conf.d# chown -R entreprise2:dovecot /etc/dovecot
root@mailserver:/etc/dovecot/conf.d# chmod -R /etc/dovecot
chmod: missing operand after '/etc/dovecot'
Try 'chmod --help' for more information.
root@mailserver:/etc/dovecot/conf.d# chmod -R o-rwx /etc/dovecot

```

Figure 3.20 : Configuration de Dovecot

Quand la configuration de Dovecot finisse, nous le lançons :

```
root@mailserver:/etc/dovecot# systemctl restart dovecot
root@mailserver:/etc/dovecot# systemctl dovecot dovecot
Unknown command verb dovecot.
root@mailserver:/etc/dovecot# systemctl status dovecot
● dovecot.service - Dovecot IMAP/POP3 email server
   Loaded: loaded (/lib/systemd/system/dovecot.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2024-05-26 14:09:46 UTC; 28s ago
     Docs: man:dovecot(1)
           https://doc.dovecot.org/
    Main PID: 24838 (dovecot)
   Status: "v2.3.16 (7e2e900c1a) running"
      Tasks: 1 (limit: 4900)
     CGroup: /system.slice/dovecot.service
             └─24838 /usr/sbin/dovecot -c /etc/dovecot/dovecot.conf --foreground
```

Figure 3.21 : Statut de Dovecot

3.2.2.3 installation et Configuration de Postfixadmin

Nous voulons bien gérer notre serveur Postfix ce qui nous a amené à installer Postfixadmin qui fournit une interface Web pour configurer facilement les paramètres de Postfix.

```
root@mailserver:/etc/dovecot# cd /var/www
root@mailserver:/var/www# ls
html
root@mailserver:/var/www# wget -O postfixadmin.tar.gz https://github.com/postfixadmin/postfixadmin/archive/refs/tags/postfixadmin-3.3.13.tar.gz
```

Figure 3.22: Installation de Postfixadmin

Une fois le fichier installé, nous allons l'extraire et configurer le fichier config.local.php. Ensuite, nous allons configurer le fichier postfixadmin.entreprise2.ma.conf qui va servir de pont entre Apache et Postfixadmin

```
root@mailserver:/var/www# nano /etc/apache2/sites-available/postfixadmin.entreprise2.ma.conf
root@mailserver:/var/www# a2ensite postfixadmin.entreprise2.ma.conf
Enabling site postfixadmin.entreprise2.ma.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@mailserver:/var/www# a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@mailserver:/var/www# systemctl restart apache2
root@mailserver:/var/www# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2024-05-26 18:03:50 UTC; 7s ago
     Docs: https://httpd.apache.org/docs/2.4/
Process: 25324 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
```

Figure 3.23 : Configuration du fichier postfixadmin.entreprise2.ma.conf

Nous accédons à l'interface graphique en tapant l'url www.postfixadmin.entreprise2.ma

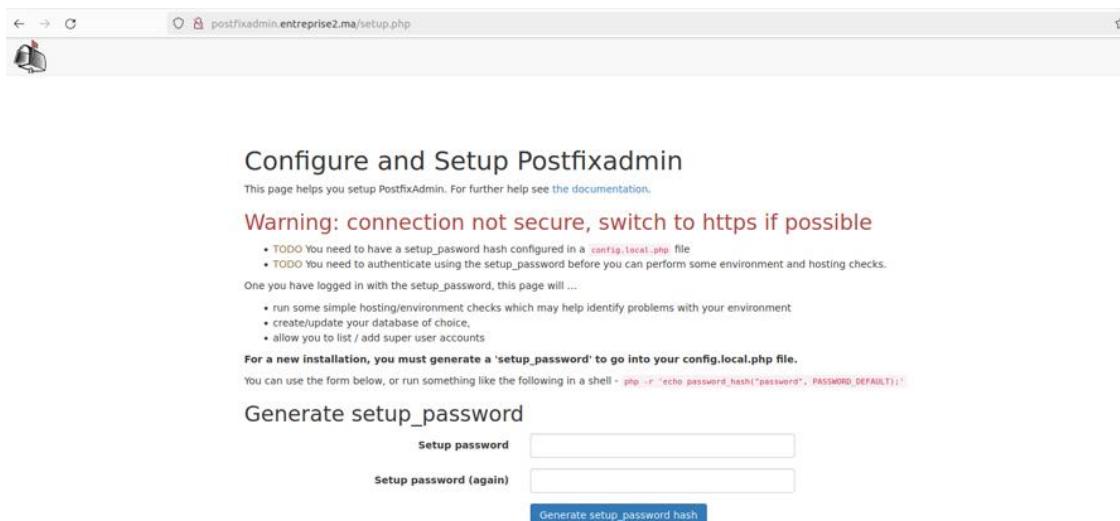


Figure 3.24 : Interface graphique de postfixadmin

Une fois le mot de passe est généré, nous l'ajoutons au fichier var/www/postfixadmin/config.local.php

```

root@mailserver:/var/www/...  x      imane@mailserver:~  x      imane@mailserver:~  x
                               GNU nano 6.2          config.local.php *
<?php
$CONF['database_type'] = 'mysqli';
$CONF['database_host'] = 'localhost';
$CONF['database_user'] = 'postfixadmin_user';
$CONF['database_password'] = 'postfixadmin_PWD';
$CONF['database_name'] = 'postfixadmin_db';
$CONF['configured'] = true;
$CONF['encrypt'] = 'md5crypt';
$CONF['setup_password'] = '$2y$10$2w2MhdCInCmT0m2IAPvd4uMbj0LjFvRkkbckV5Fwu/7lq.Hg7g3Jy>
?>
```

Figure 3.25 : Configuration du fichier config.local.php

En s'assurant que apache2 marche, nous obtenons l'interface suivante pour configurer l'admin

The screenshot shows the 'Postfix Admin - Setup' interface at the URL 192.168.77.147/setup.php. The main content area is titled 'Information' and lists various system checks with green checkmarks, indicating success. These include PHP version (8.1.2-1ubuntu2.17), Webserver (Apache/2.4.52 (Ubuntu)), Postfixadmin installation path (/var/www/postfixadmin/public), config.local.php file presence, MySQL support, password hashing methods (md5crypt and hash generation OK), database connection (using PDO, host=localhost, dbname=postfixadmin_db, charset=UTF8), database connection status (Connected OK), session support (OK), PCRE support (OK), mbstring support (OK), and optional IMAP functions (OK). Below this is a 'Warnings' section with two yellow warning icons: one for the PostgreSQL extension not found and another for SQLite support not found.

Database Update

Everything seems fine... attempting to create/update database structure
Database is up to date: 1847/1847

Add Superadmin Account

Setup password:
Admin: admin@entreprise2.ma
Password:

Figure 3.26 : Interface Postfixadmin après configuration

Nous ajoutons un superadmin comme suit :

The screenshot shows the 'Add Superadmin Account' form. It includes fields for 'Setup password' (containing '.....'), 'Admin' (containing 'admin@entreprise2.ma'), 'Password' (containing '.....'), and 'Password (again)' (containing '.....'). A blue 'Add Admin' button is at the bottom right.

Figure 3.27 : Ajout du superadmin

Ensuite, nous allons ajouter notre propre domaine d'entreprise 2

Add a new domain

Domain	entreprise2.ma
Description	ENTREPRISE2.MA
Aliases	0 -1 = disable 0 = unlimited
Mailboxes	0 -1 = disable 0 = unlimited
Mail server is backup MX	<input type="checkbox"/>
Active	<input checked="" type="checkbox"/>
Add default mail aliases	<input checked="" type="checkbox"/>
Pass expires	365 Date when password will expire
Add Domain	

Figure 3.28 : Ajout de notre propre domaine

Ensuite, nous allons ajouter deux mailbox sous le nom de iccn 2 et imane

:: Mailboxes					
	Email	To	Name	Last modified	Active
	iccn2@entreprise2.ma	Mailbox	Iccn2	2024-05-26 23:49:23	YES
	imane@entreprise2.ma	Mailbox	imane bougalzim	2024-05-26 23:50:08	YES
<input type="button" value="Add Mailbox"/> <input type="button" value="Download this list as CSV file"/>					

Figure 3.29 : Ajout de Mailbox

Pour tester les ports de Postfix et dovecot, nous utilisons telnet comme suit :

```
root@mailserver:/# telnet mail.entreprise2.ma 25
Trying 192.168.77.147...
Connected to mail.entreprise2.ma.
Escape character is '^>'.
220 mail.entreprise2.ma ESMTP Postfix (Ubuntu)
ehlo mail.entreprise2.ma
250-mail.entreprise2.ma
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-AUTH PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-SMTPUTF8
250 CHUNKING
mail from:<iccn2@entreprise2.ma>
250 2.1.0 Ok
```

Figure 3.30 : test de Postfix

```
root@mailserver:/# telnet mail.entreprise2.ma 110
Trying 192.168.77.147...
Connected to mail.entreprise2.ma.
Escape character is '^>'.
+OK Dovecot (Ubuntu) ready.
user imane@entreprise2.ma
+OK
pass Root123@
+OK Logged in.
retr 2
+OK 611 octets
Return-Path: <iccn2@entreprise2.ma>
Delivered-To: imane@entreprise2.ma
```

Figure 3.31 : test du Dovecot

3.2.2.4 Configuration du MUA

Avant d'entamer l'installation, nous allons commencer par créer la base de données `roundcube_db` comme suit :

```
MariaDB [(none)]> CREATE DATABASE roundcube_db;
Query OK, 1 row affected (0.003 sec)

MariaDB [(none)]> CREATE USER roundcube_user@localhost IDENTIFIED BY 'roundcube_PWD';
Query OK, 0 rows affected (0.023 sec)

MariaDB [(none)]> GRANT ALL ON roundcube_db.* TO roundcube_user@localhost;
Query OK, 0 rows affected (0.005 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.005 sec)

MariaDB [(none)]> exit
Bye
```

Figure 3.32 : Création de base données `roundcube_db`

Puis, nous installons Roundcube avec la commande suivante

```
root@mailserver:/var/www# wget https://github.com/roundcube/roundcubemail/releases/download/1.5.7/roundcubemail-1.5.7-complete.tar.gz
--2024-05-27 09:13:35-- https://github.com/roundcube/roundcubemail/releases/download/1.5.7/roundcubemail-1.5.7-complete.tar.gz
Resolving github.com (github.com)... 140.82.121.3
```

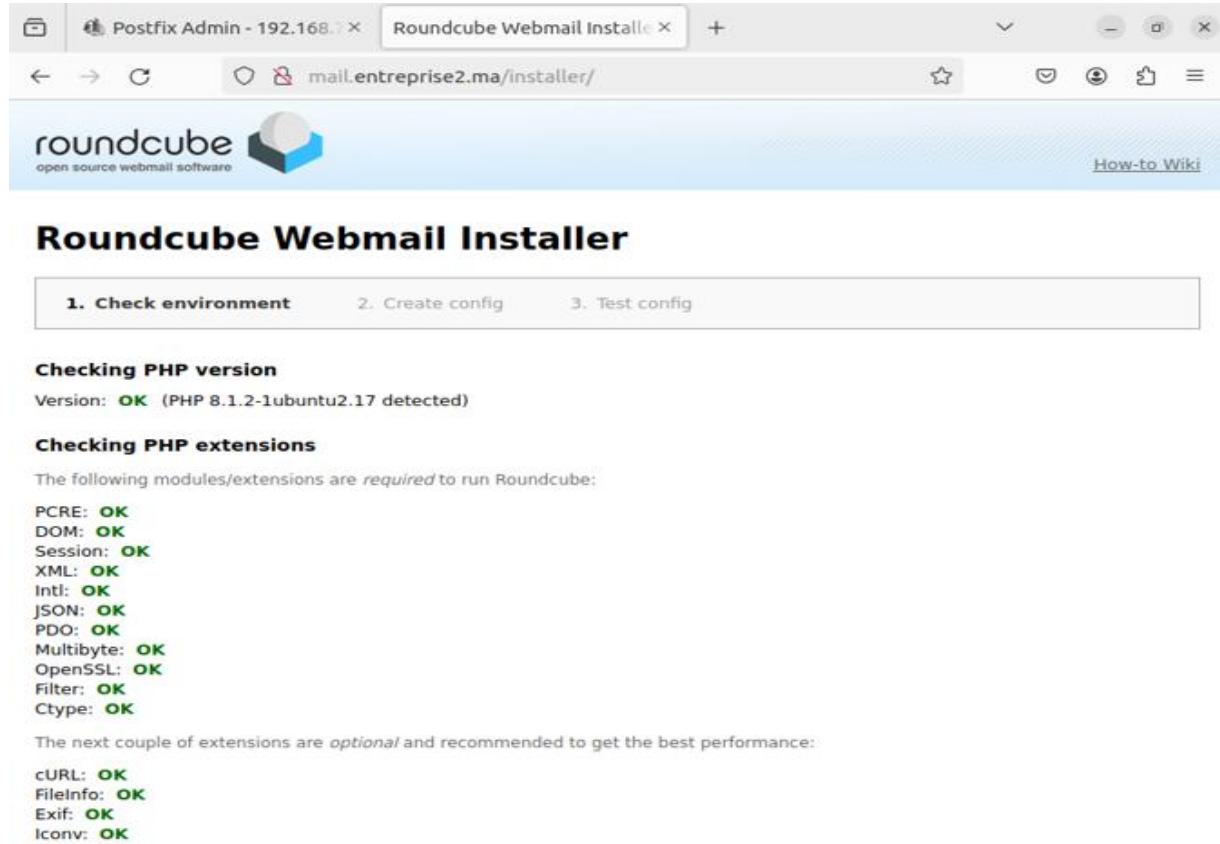
Figure 3.33 : Installation de Roundcube

Nous allons créer utilisateur de roundcube avec la commande suivante

```
root@mailserver:/var/www# mysql -uroundcube_user -proundcube_PWD roundcube_db < roundcube/SQL/mysql.initial.sql
root@mailserver:/var/www# nano /etc/apache2/sites-available/roundcubemail.conf
root@mailserver:/var/www# a2ensite roundcubemail.conf
```

Figure 3.34 : Création d'utilisateur Roundcube

Pour finaliser l'installation de Roundcube, nous accédons à <https://mail.entreprise2.ma/installer>



The screenshot shows a browser window titled "Postfix Admin - 192.168.1.10" with the URL "mail.entreprise2.ma/installer". The page is titled "Roundcube Webmail Installer" and features a navigation bar with three tabs: "1. Check environment" (selected), "2. Create config", and "3. Test config".

Checking PHP version
Version: **OK** (PHP 8.1.2-1ubuntu2.17 detected)

Checking PHP extensions
The following modules/extensions are *required* to run Roundcube:

- PCRE: **OK**
- DOM: **OK**
- Session: **OK**
- XML: **OK**
- Intl: **OK**
- JSON: **OK**
- PDO: **OK**
- Multibyte: **OK**
- OpenSSL: **OK**
- Filter: **OK**
- Ctype: **OK**

The next couple of extensions are *optional* and recommended to get the best performance:

- CURL: **OK**
- FileInfo: **OK**
- Exif: **OK**
- Iconv: **OK**

Figure 3.35 : Interface de Roundcube installer

Au niveau de cet interface, nous allons configurer l'url pour mettre notre domaine ainsi que la base de données et IMAP et SMTP. Pour vérifier, nous faisons les tests suivants :

Test SMTP config

Server	mail.entreprise2.ma
Port	25
Username	iccn2@entreprise2.ma
Password	[redacted]

Trying to send email...
SMTP send: **OK**

Figure 3.36: Test SMTP

Test IMAP config

Server	mail.entreprise2.ma
Port	143
Username	iccn2@entreprise2.ma
Password	[redacted]

Connecting to mail.entreprise2.ma...
IMAP connect: **OK** (SORT capability: yes)

Figure 3.37: Test IMAP

Nous allons envoyer le message depuis l'utilisateur iccn2 à imane avant d'intégrer les outils de sécurité. Nous visualisons le header

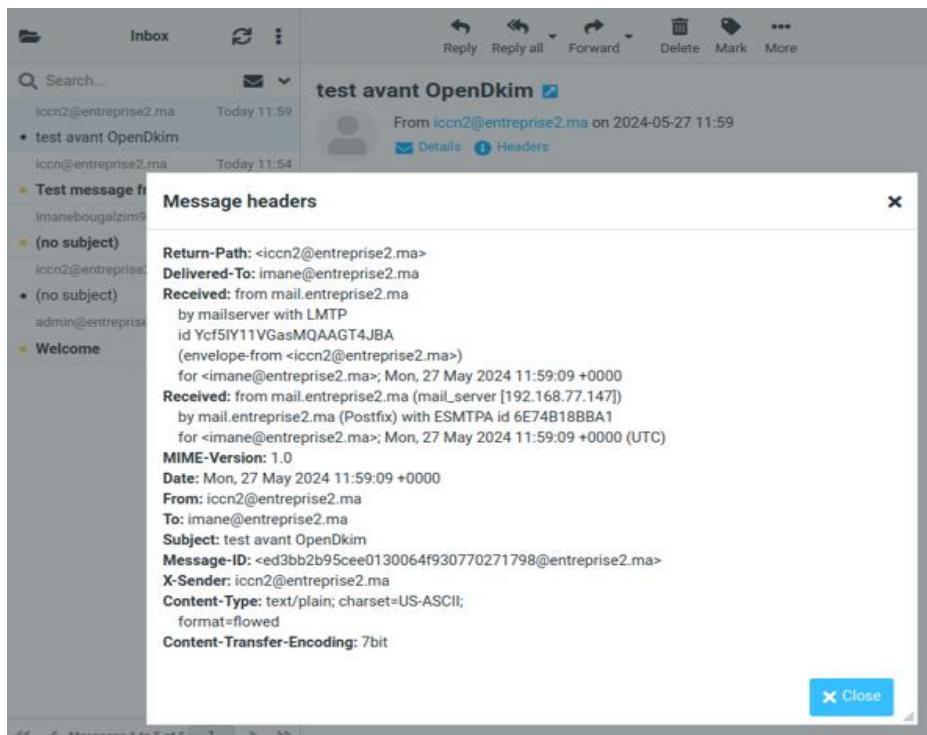


Figure 3.38 : Message header avant openDkim

Nous installons OpenDkim et nous utilisons la commande suivante opendkim-genkey afin de générer une paire de clés DKIM (DomainKeys Identified Mail) pour le domaine entreprise2.ma. DKIM est une technique d'authentification des courriers électroniques qui permet de vérifier que les courriels proviennent bien du domaine qu'ils prétendent représenter et qu'ils n'ont pas été altérés en transit.

```
root@mailserver:/etc/opendkim# opendkim-genkey -s dkimkey -d entreprise2.ma
root@mailserver:/etc/opendkim# ll
total 24
drwxr-xr-x  2 root root  4096 May 27 12:08 .
drwxr-xr-x 154 root root 12288 May 27 12:06 ..
-rw-----  1 root root  1708 May 27 12:08 dkimkey.private
-rw-----  1 root root   510 May 27 12:08 dkimkey.txt
```

Figure 3.39 : Génération des clés DKIM

Nous allons tester encore l'envoi du message après configuration de OpenDkim, nous remarquons l'ajout de Dkim signature

```
Return-Path: <iccn2@entreprise2.ma>
Delivered-To: imane@entreprise2.ma
Received: from mail.entreprise2.ma
by mailserver with LMTP
id miZpFwmLVGbnQAAAGT4JBA
(envelope-from <iccn2@entreprise2.ma>)
for <imane@entreprise2.ma>; Mon, 27 May 2024 13:30:49 +0000
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/simple; d=entreprise2.ma;
s=dkimkey; t=1716816649;
bh=0uW0cgLbqUSUWJTbqBULcMejlib6NYQcpJFGkALQlLi=;
h=Date:From:To:Subject:From;
b=oEnK05NIYbauOkPTGzGmKKqeUhZVdTXsPV+9/EGIOPDv8Tf+3IA9PqiP3t2RR/tvf
y3AGrU6uqH8wKysYqRMahjaSyYeV1J75kvJoN483JmlQ7GXxDUE5N+Hj6UL4VQrqDa
dP4FFZtSOHsRnXX8ZPQ9XCdnCuH10jUlfelhlbeC081rw+ze3urODiEF5iyicNik/YK
RWMwzGzXiWoLtssLtcj6HppWrTjlRZSM69zqio/3Nq0zFRG6cZr9YPT1rM2wl6chXt
AVWbFwa6logWxgKWWXAmayfU0Eba+g6gsGr02aP9KfJ5bqhycE2PgV4onwn6aQp+AX
veicDvEvlayUQ=
Received: from mail.entreprise2.ma (mail_server [192.168.77.147])
by mail.entreprise2.ma (Postfix) with ESMTPA id 33C1E18AE8D
for <imane@entreprise2.ma>; Mon, 27 May 2024 13:30:49 +0000 (UTC)
MIME-Version: 1.0
Date: Mon, 27 May 2024 13:30:49 +0000
From: iccn2@entreprise2.ma
To: imane <imane@entreprise2.ma>
Subject: test apres opendkim conf
Message-ID: <490f1e6cb4e96766d88ad2cea690b598@entreprise2.ma>
X-Sender: iccn2@entreprise2.ma
Content-Type: text/plain; charset=US-ASCII;
format=flowed
Content-Transfer-Encoding: 7bit
```

Figure 3.40 : Envoi du message après la configuration OpenDkim

Chapitre 4

Renforcement de la Sécurité du Service de Messagerie

Après avoir examiné dans le chapitre précédent l'implémentation d'un service de messagerie et les mesures de sécurité de base telles que SPF, DKIM et DMARC, nous allons maintenant explorer des solutions avancées pour renforcer davantage la sécurité de notre infrastructure de messagerie. Dans ce chapitre, nous aborderons l'intégration de technologies telles que ClamAV Amavis, SpamAssassin et VirusTotal, qui offrent des fonctionnalités de détection et de filtrage avancées pour protéger nos systèmes contre les menaces telles que les logiciels malveillants, le spam et les attaques ciblées. Nous discuterons également de la manière dont ces solutions peuvent être configurées et intégrées efficacement dans notre environnement de messagerie, contribuant ainsi à garantir la sécurité et l'intégrité de nos communications électroniques.

4.1 Clamav amavis :

Amavis (A Mail Virus Scanner) est une interface haute performance entre un agent de transfert de messages (MTA) tel que Postfix et des filtres de contenu. Un filtre de contenu est un programme qui analyse les en-têtes et le corps d'un message électronique, et prend généralement une certaine action en fonction de ce qu'il trouve. Les exemples les plus courants sont le scanner de virus ClamAV et SpamAssassin. [4.1]

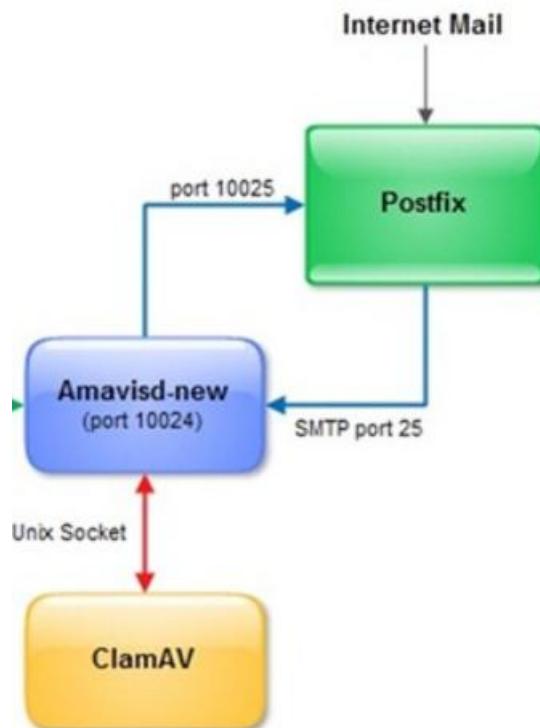


Figure 4.1 : fonctionnement de Clamav-Amavis

Les emails provenant d'Internet arrivent d'abord sur le serveur de messagerie Postfix, qui transfère les emails à Amavisd-new via le port 10024 pour le traitement. Amavisd-new agit comme un intermédiaire pour le filtrage des emails, Amavisd envoie les emails à ClamAV pour une analyse antivirus via un socket Unix. ClamAV vérifie les emails pour détecter les courriels malicieux. Après l'analyse par ClamAV, les résultats sont renvoyés à Amavisd-new qui transfère ensuite les emails de retour à Postfix via le port 10025, qui utilise le port SMTP (port 25) pour la livraison finale des emails aux destinataires.

```
root@mailserver:~# apt-get install clamav clamav-daemon -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Figure 4.2 : Installation de Clamav

Pour télécharger les dernières versions disponibles, tester leur intégrité, et mettre à jour les bases de données locales, nous avons utilisé la commande freshclam

```
root@mailserver:~# freshclam
Tue May 28 13:55:45 2024 -> ClamAV update process started at Tue May 28 13:55:45 2024
Tue May 28 13:55:45 2024 -> daily.cvd database is up-to-date (version: 27289, sigs: 2061864, f-level: 90, builder: raynman)
Tue May 28 13:55:45 2024 -> main database available for download (remote version: 62)
Time: 10.7s, ETA: 0.0s [=====
Tue May 28 13:56:04 2024 -> Testing database: '/var/lib/clamav/tmp.07be61c9b3/clamav-4ba80d4279b4478fab735b2162c61b29.tmp-main.cvd' ...
Tue May 28 13:56:26 2024 -> Database test passed.
Tue May 28 13:56:26 2024 -> main.cvd updated (version: 62, sigs: 6647427, f-level: 90, builder: sigmgr)
Tue May 28 13:56:26 2024 -> bytecode database available for download (remote version: 335)
Time: 0.4s, ETA: 0.0s [=====
Tue May 28 13:56:27 2024 -> Testing database: '/var/lib/clamav/tmp.07be61c9b3/clamav-6288d70619d7ddb6d04602d0c148878.tmp-bytecode.cvd' ...
Tue May 28 13:56:27 2024 -> Database test passed.
Tue May 28 13:56:27 2024 -> bytecode.cvd updated (version: 335, sigs: 86, f-level: 90, builder: raynman)
```

Figure 4.2 : Mise à jour de Clamav

Une fois la configuration est finie, nous avons testé en envoyant un message avec un sujet XJS*C4JDBQADN1.NSBN3*2IDNEN*GTUBE-STANDARD-ANTI-UBE-TEST-EMAIL*C.34X.

Nous remarquons que le message a été déclaré comme spam grâce au content filter de Amavis

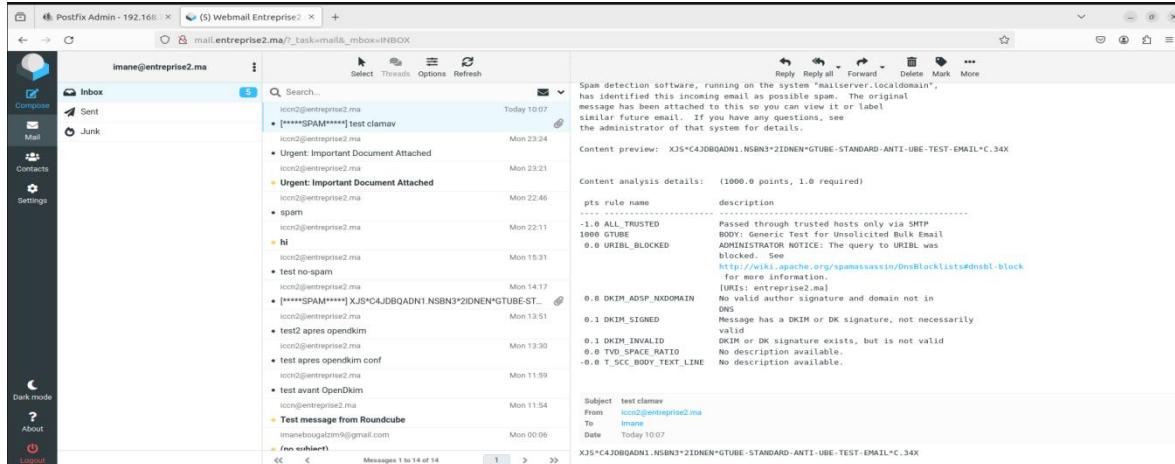


Figure 4.3 : Détection du spam par Amavis

4.2 SpamAssassin

SpamAssassin est un logiciel open source utilisé pour filtrer les courriers électroniques afin de détecter les spams (courriels indésirables). Il utilise une variété de techniques pour analyser et marquer les emails en fonction de leur probabilité d'être des spams :

- SpamAssassin utilise des algorithmes de filtrage bayésien pour analyser les schémas de mots et les caractéristiques des emails. En se basant sur des ensembles de données d'emails marqués comme spam ou non spam, il apprend à distinguer les deux catégories.
- Il applique un ensemble de règles définies à l'avance pour attribuer des scores aux emails. Ces règles peuvent inclure des tests sur les en-têtes, le contenu, les liens, et d'autres éléments des emails.
- SpamAssassin peut utiliser des listes noires (blacklists) pour bloquer les emails provenant de sources connues pour envoyer des spams, et des listes blanches (whitelists) pour permettre les emails provenant de sources fiables.
- Il vérifie la réputation des adresses IP et des domaines des expéditeurs en utilisant des bases de données externes comme les DNSBL (DNS-based Blackhole List).
- SpamAssassin peut être intégré avec d'autres outils de filtrage de courriel comme Amavis-new pour offrir une solution complète de filtrage des emails.

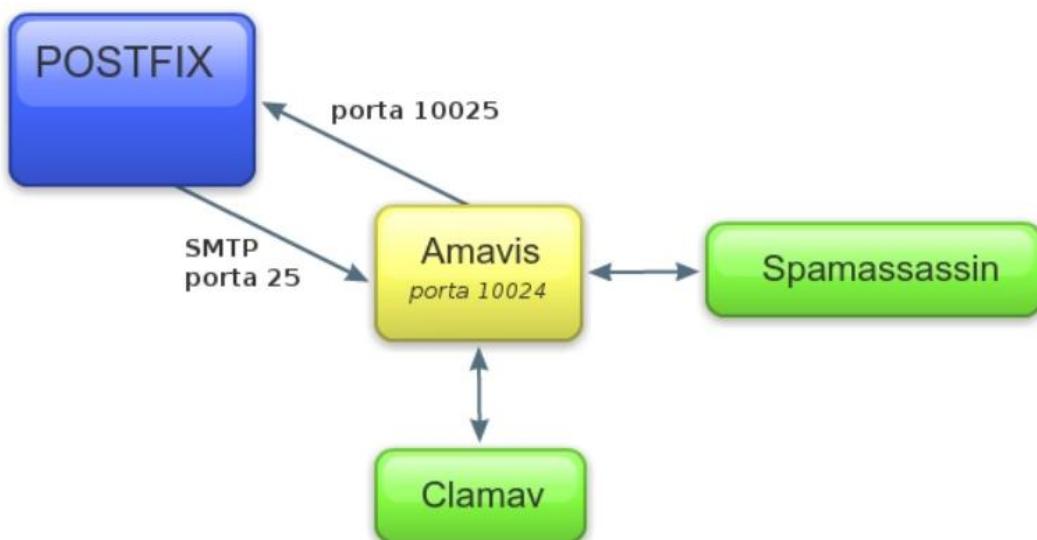


Figure 4.4 : fonctionnement de Spamassassin avec Amavis

Au niveau de notre serveur de messagerie, nous allons installer Spamassassin et nous allons essayer de l'intégrer à Roundcube afin de détecter les spams au niveau des messages. Nous allons commencer par installer Spamassassin et nous vérifions que son statut est actif.

Afin de configurer les règles de Spamassassin, nous ajoutons une rubrique de filters en ajoutant le plugin managesieve en /var/www/html/roundcube/plugins puis en actualisant la page, nous trouvons filters.

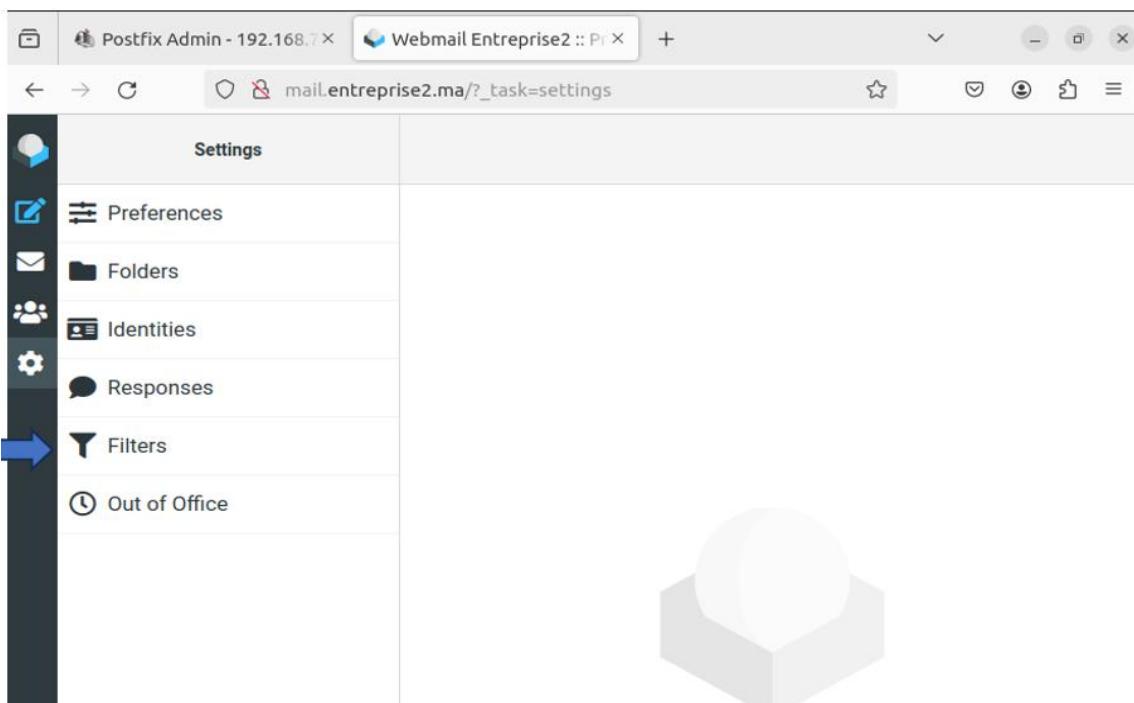


Figure 4.5 : Rubrique de filtres

L'ajout des règles de filtre peut se faire soit sur l'interface graphique soit en fichier de configuration /etc/spamassassin/local.cf comme suit

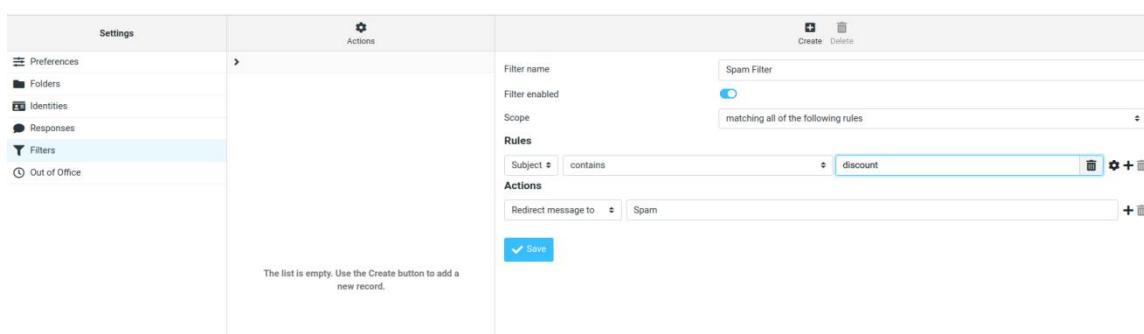
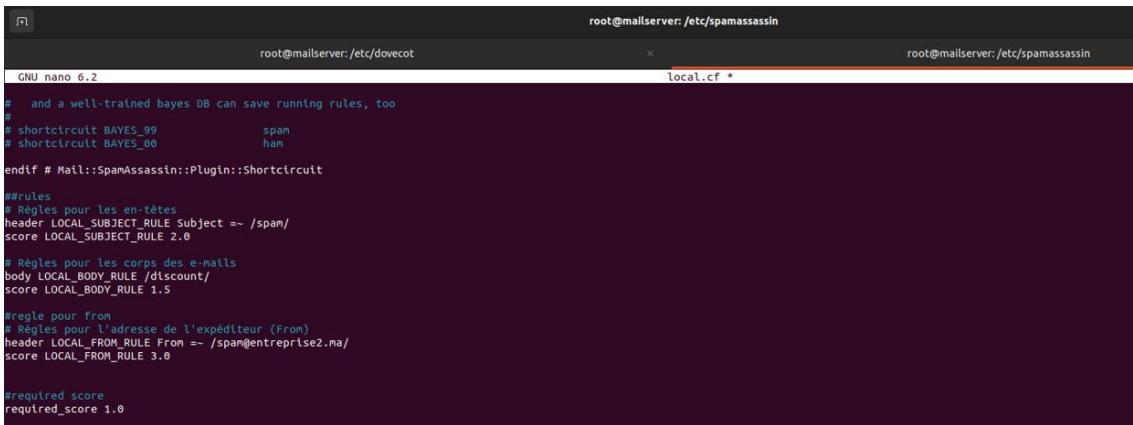


Figure 4.6 : Ajout des règles graphiquement

Dans le fichier de configuration `/etc/spamassassin`, nous pouvons ajouter des règles à appliquer sur l'en-tête, le corps et le destinataire en définissant un score requis pour indiquer à SpamAssassin quand appliquer ces règles.



```

root@mailserver:/etc/spamassassin
root@mailserver:/etc/dovecot
local.cf *

GNU nano 6.2
# and a well-trained bayes DB can save running rules, too
#
# shortcircuit BAYES_99          spam
# shortcircuit BAYES_00          ham
endif # Mail::SpamAssassin::Plugin::Shortcircuit

##rules
# Règles pour les en-têtes
header LOCAL SUBJECT_RULE Subject =~ /spam/
score LOCAL SUBJECT_RULE 2.0

# Règles pour les corps des e-mails
body LOCAL BODY RULE /discount/
score LOCAL BODY RULE 1.5

#règle pour from
# Règles pour l'adresse de l'expéditeur (From)
header LOCAL FROM_RULE From =~ /spam@entreprise2.ma/
score LOCAL FROM RULE 3.0

#required score
required_score 1.0

```

Figure 4.7 : Ajout des règles en `/etc/spamassassin`

Afin de tester, nous avons envoyé un message de imane à iccn2 avec le sujet ‘spam’ et le corps ‘discount’. Nous remarquons que le message a été détecté en tant que spam en utilisant la version de Spamassassin 3.4.6.

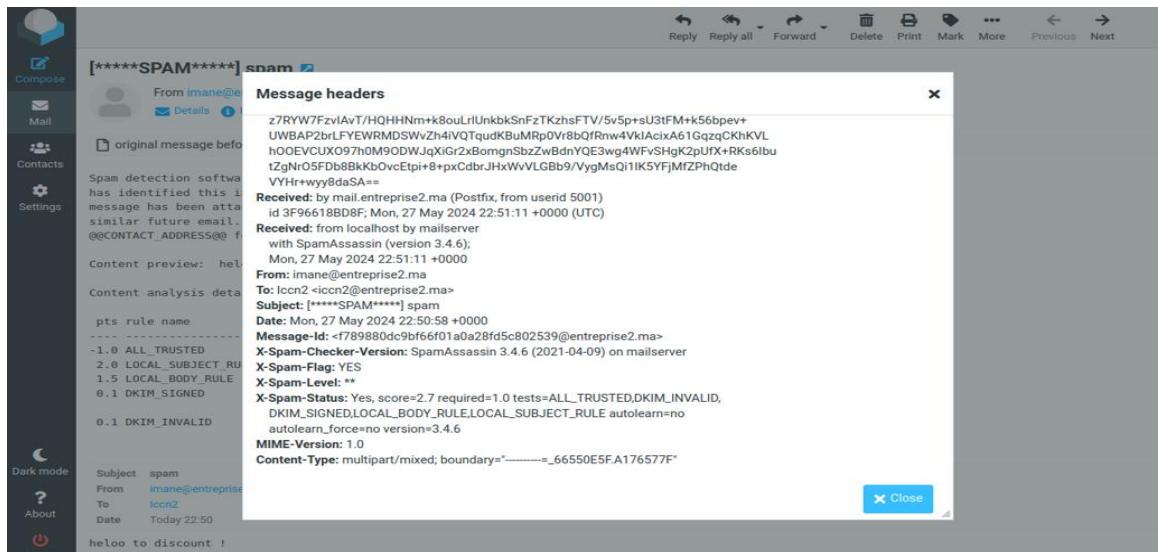


Figure 4.8 : Message détecté comme spam

Pour bien pousser le filtrage on a essayé de bien mentionner dans le fichier de configuration les mots susceptibles d'exister dans un mail de spam, parmi eux :

spam|spam|viagra|lottery|win|free|urgent|discount|offer|clickhere|exclusive|limited time|winner|prize|gift|special promotion|act now|deal|money back|best price|.....

Pour le corps de message :

```
body LOCAL_BODY_RULE /discount|free|offer|limited time offer|click
here|unsubscribe|urgent|promotion|winner|prize|http:\/\/[^\s]+|https:\/\/[^\s]+|example\.com|free-stuff\.com|cheap-products\.com/
```

Explication de la règle

- Expression régulière combinée : Cette règle utilise une seule expression régulière pour détecter une variété de mots, phrases, et liens couramment associés aux spams. L'opérateur | agit comme un "ou logique", permettant à la règle de correspondre à n'importe lequel des termes ou motifs listés.
- Score : Le score de cette règle est configuré à 5.0, mais on peut ajuster ce score en fonction des besoins et de la sensibilité désirée pour la détection des spams.

Contenu de l'expression régulière

- Mots et phrases spécifiques : discount|free|offer|limited time offer|click here|unsubscribe|urgent|promotion|winner|prize : Ces termes sont souvent utilisés dans les messages de spam pour attirer l'attention de l'utilisateur.
- Liens HTTP/HTTPS : http:\/\/[^\s]+|https:\/\/[^\s]+ :Cette partie de l'expression régulière détecte n'importe quel lien HTTP ou HTTPS, qui sont souvent utilisés dans les e-mails de phishing.
- Domaines suspects : example\.com|free-stuff\.com|cheap-products\.com : Cette section détecte les liens vers des domaines spécifiques que vous avez identifiés comme suspects ou associés à des activités de spam.

C'est le moment de faire un test de fonctionnement de ces règles :

pts rule name	description
1.0 ALL TRUSTED	Passed through trusted hosts only via SMTP
2.0 LOCAL SUBJECT RULE	No description available.
5.0 LOCAL BODY RULE	BODY: No description available.

Figure 4.9 : test des règles établis de spamassassin

Spam détecter avec succès.

4.2.1 Configuration du Spamassassin sur Zimbra :

De même, nous avons installé Spamassassin sur Zimbra afin de détecter les messages malveillants. Une fois l'installation est achevée, nous avons essayés d'intégrer des règles simples afin de s'assurer du bon fonctionnement du logiciel. Nous avons défini une règle ‘Special Offer’ pour l’entête qui sera classée en tant que spam comme suit

```
# Rule to detect "special offer" in the subject
header SPECIAL_OFFER_RULE    Subject =~ /special offer/i
score SPECIAL_OFFER_RULE    10.0
describe SPECIAL_OFFER_RULE Subject contains the phrase "special offer"
```

Figure 4.10 : Règle défini sur l’entête

Nous avons testé en envoyant un message d’utilisateur Nezha à Hamza contenant en entête ‘Special offer’ et nous remarquons que le message est classé en répertoire Junk ce qui veut dire qu’il est classé en tant que spam.

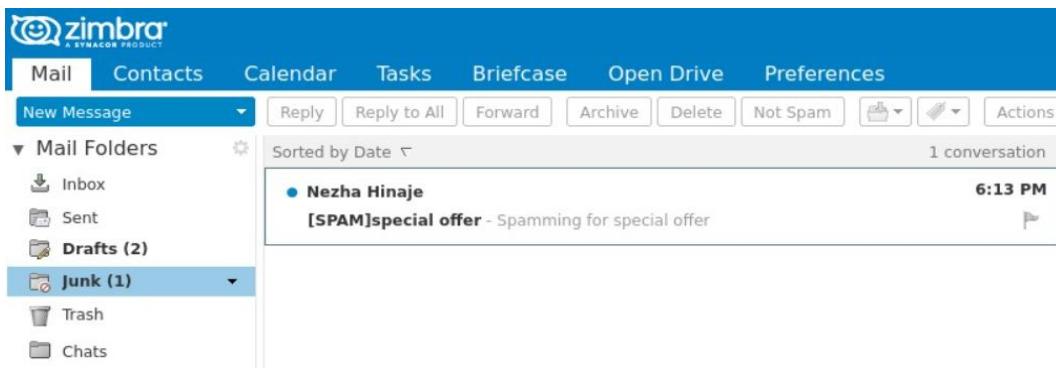


Figure 4.11 : Message détecté en tant que spam

4.3 VirusTotal

4.3.1 Description

VirusTotal est un service en ligne gratuit et puissant qui permet d'analyser des fichiers et des URL afin de détecter des virus, vers, chevaux de Troie, et d'autres types de malwares. Crée en 2004 et acquis par Google en 2012, il utilise des moteurs antivirus provenant de nombreux fournisseurs pour offrir une analyse complète et fiable. VirusTotal agit comme un agrégateur de logiciels antivirus, en intégrant des dizaines de moteurs de détection différents pour fournir une vue d'ensemble précise et détaillée sur la sécurité d'un fichier ou d'une URL. En plus de la détection de malwares, VirusTotal fournit des informations sur les comportements suspects, les menaces émergentes, et offre des statistiques globales sur les tendances de sécurité. L'outil est largement utilisé par les chercheurs en sécurité, les administrateurs réseau, et les professionnels de l'informatique pour renforcer les défenses contre les cybermenaces.

4-3-2 API VirusTotal

L'API de VirusTotal permet aux développeurs et aux professionnels de la sécurité d'intégrer les capacités de détection de VirusTotal dans leurs propres applications et systèmes. Voici une description détaillée de son fonctionnement :

4-3-2-1 Clé API

Pour utiliser l'API de VirusTotal, il est nécessaire d'obtenir une clé API. Cette clé est utilisée pour authentifier les requêtes et est fournie après l'inscription sur le site de VirusTotal.

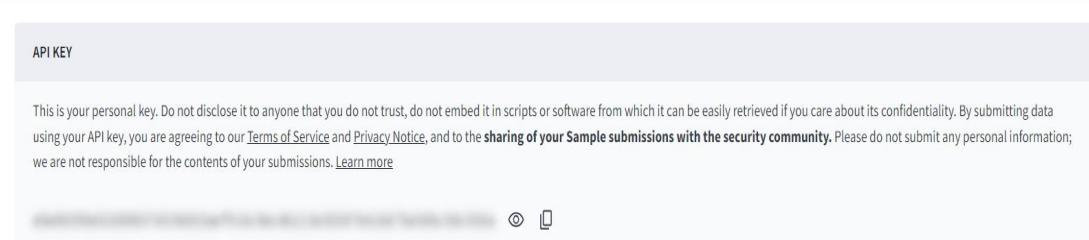


Figure 4.12: API KEY Virustotal

4.3.2.2 Principe de Fonctionnement

- **Soumission** : L'utilisateur soumet un fichier ou une URL à l'API de VirusTotal pour analyse.
- **Analyse** : VirusTotal distribue l'échantillon soumis à ses moteurs antivirus pour une analyse approfondie.
- **Rapport** : Après l'analyse, un rapport est généré. Ce rapport contient des informations détaillées sur les menaces détectées, les moteurs antivirus ayant effectué l'analyse, les dates, et les scores d'infection.
- **Récupération** : L'utilisateur peut récupérer le rapport à tout moment en utilisant l'ID de la ressource (fichier ou URL) soumis.

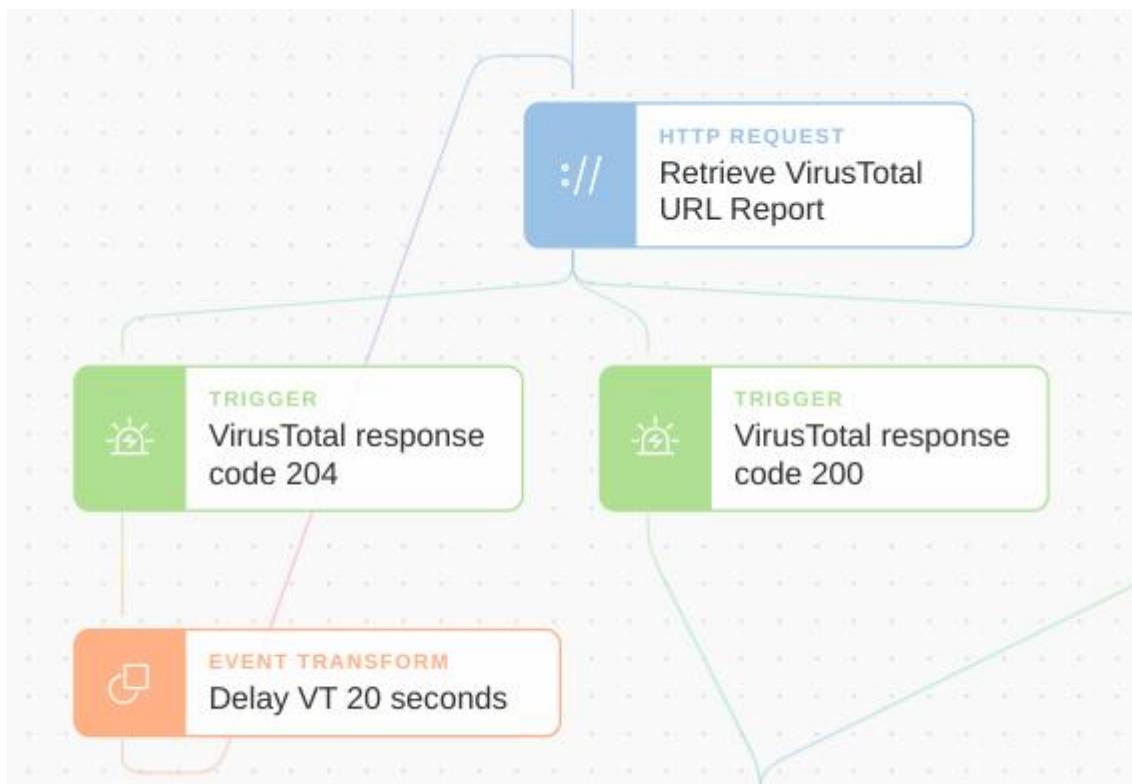


Figure 4.13 : Fonctionnement de l'API de VirusTotal

4.3.3 VirusTotal et Postfix

Dans le cadre de notre travail avec Postfix pour créer un serveur de messagerie sécurisé, nous avons intégré l'API de VirusTotal pour améliorer la détection des tentatives de phishing dans les URLs et les pièces jointes des emails. Cette intégration nous permet d'analyser automatiquement le contenu des emails en utilisant les capacités de détection avancées de VirusTotal, renforçant ainsi la sécurité de notre infrastructure de messagerie et protégeant nos utilisateurs contre les menaces malveillantes.

4.3.3.1 Installation de l'API :

Pour installer la bibliothèque virustotal-api, utilisez la commande suivante, pip install virustotal-api , Une fois installée, on intègre cette bibliothèque dans un script Python pour interagir avec l'API de VirusTotal

```
lmane@mailserver:~/part3/AI/AI$ pip3 install requests
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: requests in /home/lmane/.local/lib/python3.10/site-packages (2.32.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /home/lmane/.local/lib/python3.10/site-packages (from requests) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/lib/python3/dist-packages (from requests) (3.3)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/lib/python3/dist-packages (from requests) (1.26.5)
Requirement already satisfied: certifi>=2017.4.17 in /usr/lib/python3/dist-packages (from requests) (2020.6.20)
lmane@mailserver:~/part3/AI/AI$ pip install virustotal-api
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: virustotal-api in /usr/local/lib/python3.10/dist-packages (1.1.11)
Requirement already satisfied: requests>=2.22.0 in /home/lmane/.local/lib/python3.10/site-packages (from virustotal-api) (2.32.3)
Requirement already satisfied: idna<4,>=2.5 in /usr/lib/python3/dist-packages (from requests>=2.22.0->virustotal-api) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/lib/python3/dist-packages (from requests>=2.22.0->virustotal-api) (2020.6.20)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/lib/python3/dist-packages (from requests>=2.22.0->virustotal-api) (1.26.5)
Requirement already satisfied: charset-normalizer<4,>=2 in /home/lmane/.local/lib/python3.10/site-packages (from requests>=2.22.0->virustotal-api) (3.3.2)
lmane@mailserver:~/part3/AI/AI$
```

Figure 4.14 : Installation de l'API VirusTotal

4.3.3.2 Utilisation

À cette fin, nous avons développé un script Python utilisant la bibliothèque virustotal-api. Ce script permet d'automatiser le processus d'analyse des pièces jointes des emails entrants. Lorsqu'un email est reçu, le script extrait automatiquement les pièces jointes et les soumet à l'API de VirusTotal pour une évaluation approfondie. Une fois l'analyse terminée, le script récupère les résultats et les utilise pour décider de la marche à suivre. Cette intégration nous permet de garantir une analyse rapide et efficace de chaque fichier joint, renforçant ainsi la sécurité de notre système de messagerie électronique contre les menaces potentielles. Pour une référence détaillée du code, veuillez consulter l'annexe.

4.3.3.3 Liaison avec Postfix

Pour intégrer l'API VirusTotal à Postfix afin de scanner les emails entrants à la recherche de virus, on doit d'abord configurer Postfix pour qu'il communique avec l'API VirusTotal. Dans le fichier /etc/postfix/master.cf, on doit ajouter une entrée pour un service Postfix personnalisé qui appellera l'API VirusTotal.

```
#virustotal
virustotal_filter unix - n n - - pipe
  flags=Rq  user=vtCheck  argv=/usr/local/bin/vtCheckFilter.sh ${sender} ${recipient}
```

Figure 4.15 : /etc/postfix/master.cf

Ensuite, dans le fichier /etc/postfix/main.cf, on devra spécifier l'URL de l'API VirusTotal, ainsi que sa clé d'API personnelle. On pourra également configurer des options supplémentaires, comme le comportement à adopter lorsqu'un virus est détecté (par exemple, rejeter le message ou le placer en quarantaine). Une fois ces modifications effectuées, Postfix sera en mesure d'interroger VirusTotal pour vérifier si les pièces jointes des emails entrants sont sécurisées, renforçant ainsi la protection du système contre les menaces de malware.

```
# VirusTotal filter
content_filter =virustotal_filter:127.0.0.1:10025
```

Figure 4.16 : /etc/postfix/main.cf

4.3.4 Test et validation

Pour le test on essaie d'envoyer un mail qui contient un lien malicious et on intercepte s'il est bien détecté

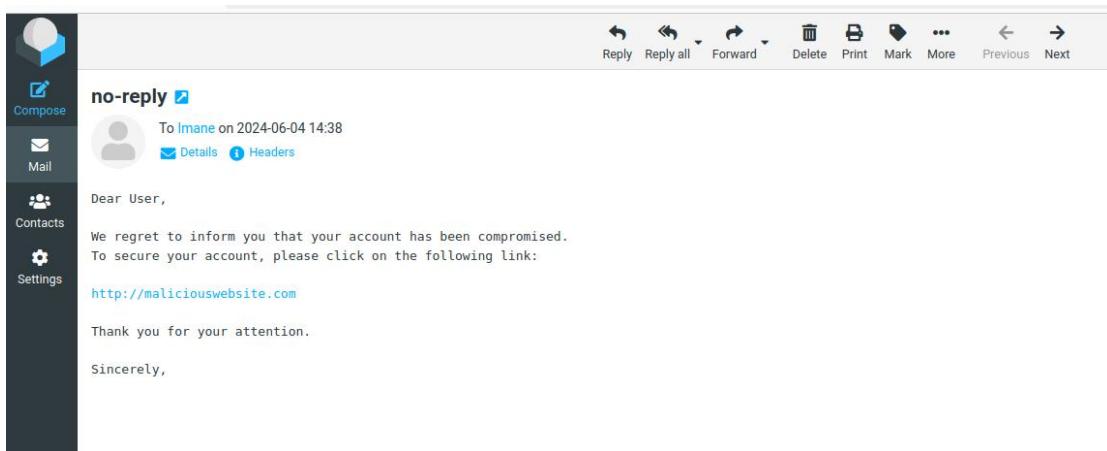


Figure 4.17 : Message de L'expéditeur

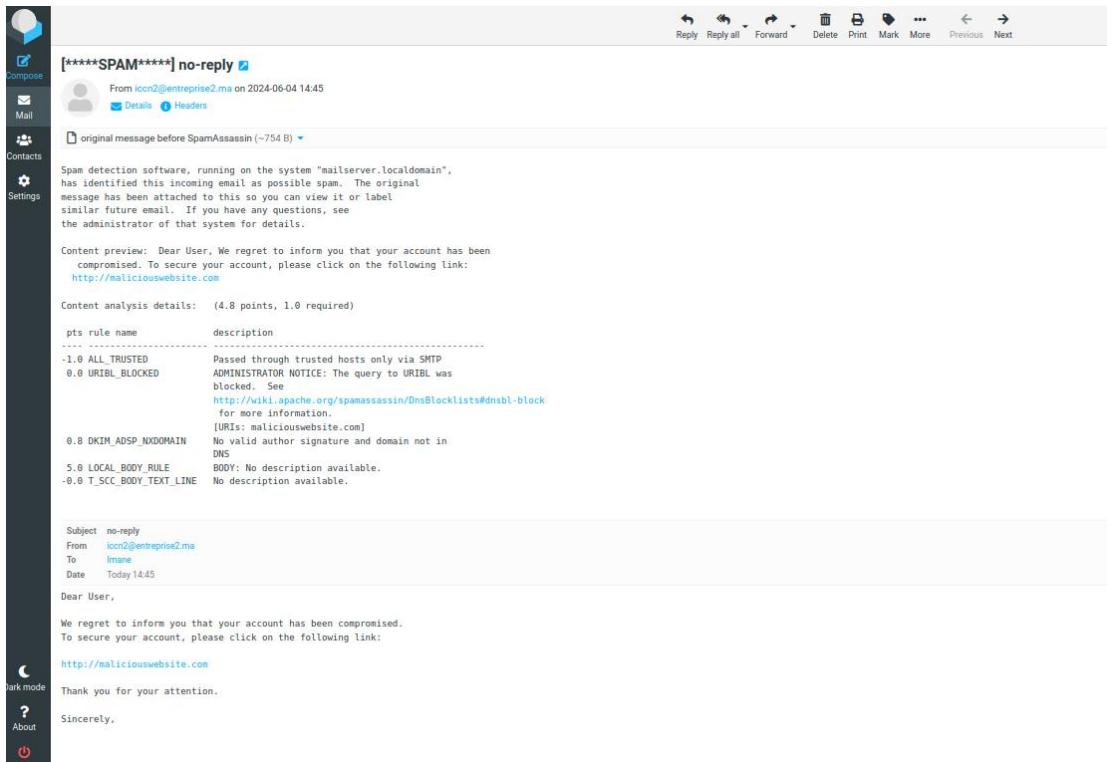


Figure 4.18 : Message chez le Destinataire

Conclusion :

En conclusion, l'implémentation d'un service de messagerie sécurisé utilisant des outils tels que SpamAssassin et VirusTotal est essentielle pour protéger les communications électroniques contre les menaces de spam et de logiciels malveillants. SpamAssassin permet une filtration efficace des courriers indésirables, tandis que VirusTotal analyse les pièces jointes pour détecter les virus et autres menaces. Ensemble, ces outils renforcent la sécurité et l'intégrité des échanges, offrant ainsi une couche de protection indispensable pour les utilisateurs et les organisations.

Chapitre 5

Intelligence Artificielle pour la Sécurité des Services de Messagerie

Les outils de sécurité tels que VirusTotal, SpamAssassin, Amavis et ClamAV offrent une protection solide contre les menaces courantes comme les virus, le spam et les logiciels malveillants, ces solutions ne suffisent souvent pas à contrer les attaques de plus en plus sophistiquées et les nouvelles variantes de menaces. Pour renforcer encore davantage la sécurité de notre service de messagerie, il est crucial d'intégrer des technologies de machine learning.

Le machine learning, ou apprentissage automatique, permet de créer des modèles capables de détecter des motifs complexes et des anomalies dans les flux de messagerie, ce qui est essentiel pour identifier les spams et les tentatives de phishing qui échappent aux filtres de contenu traditionnels. En exploitant de vastes ensembles de données et en adaptant continuellement leurs algorithmes, les solutions de machine learning peuvent offrir une détection plus précise et proactive des menaces.

Dans ce chapitre, nous explorerons les différentes façons dont le machine learning peut être appliqué pour améliorer la sécurité des services de messagerie. Nous discuterons le principe de base de ces technologies, les outils disponibles et de la manière dont ils peuvent être intégrés dans notre infrastructure existante pour fournir une couche de protection supplémentaire. L'objectif est de démontrer comment l'utilisation de techniques avancées de machine learning peut renforcer notre capacité à détecter et à neutraliser les attaques, assurant ainsi une sécurité optimale pour nos communications électroniques.

5.1 Les phases des attaques de phishing basées sur des malwares

Une attaque de phishing typique comprend trois phases regroupant plusieurs étapes :

- Première phase (étapes 1, 2 et 3) : Les attaquants envoient des emails trompeurs en masse (généralement via des botnets) qui redirigent les utilisateurs vers des sites web frauduleux ou téléchargent du code malveillant pour l'installer sur leurs machines.
- Deuxième phase : Les attaquants utilisent des techniques d'obscurcissement pour cacher le code malveillant sous plusieurs couches (par exemple, masquer l'adresse IP d'un URL malveillant pour le faire ressembler à un URL légitime). Ces techniques rendent l'analyse statique des emails difficile.
- Troisième phase (étape 4) : Les attaquants créent des sites web frauduleux (hébergés sur des machines piratées) qui incitent les victimes à se rediriger vers le site de l'attaquant.
- Quatrième phase (étape 5) : L'utilisateur victime peut télécharger un cheval de Troie d'accès à distance (RAT) qui, une fois installé sur l'ordinateur du réseau, peut se propager sur l'ensemble du réseau de l'entreprise et inciter d'autres utilisateurs à fournir des informations confidentielles.
- Cinquième et sixième phase (étapes 6 et 7) : Les informations volées sont envoyées au serveur des pirates (étape 6) qui les utilisent ensuite pour pirater les données de l'utilisateur, telles que son argent (étape 7). La circulation de l'information est illustrée dans la figure suivante. [6.1]

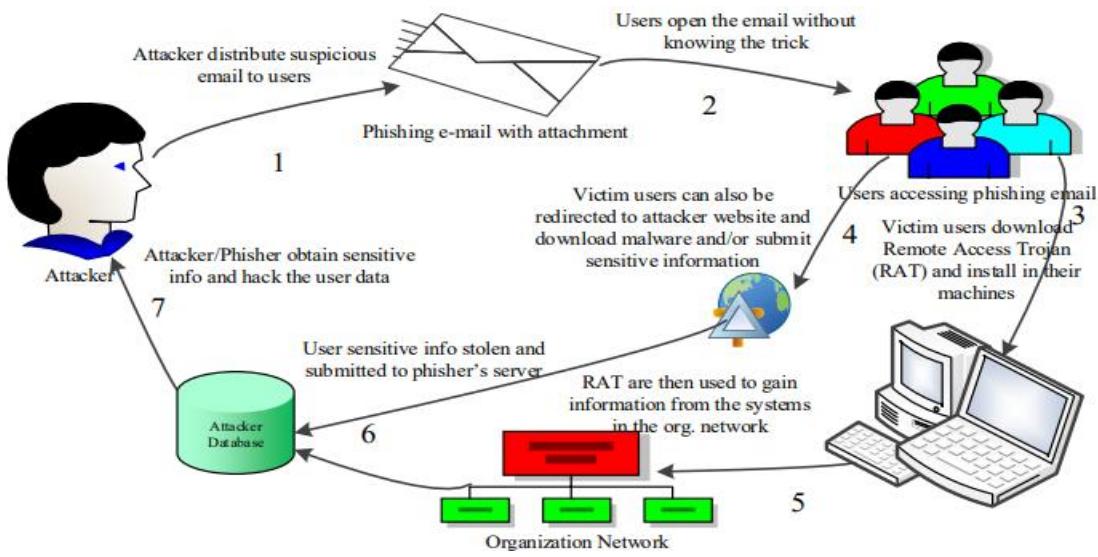


Figure 5.1 : Flux d'informations dans les étapes des attaques de phishing

5.2 Méthodologie d'étude

Les méthodes de recherche utilisées dans cette étude comprennent la collecte de données, la création d'un ensemble de données, une expérimentation pratique et l'intégration du modèle d'apprentissage automatique (ML) avec le filtre anti-spam.

Les étapes commencent par une revue de la littérature qui couvre diverses études et des caractéristiques des modèles d'apprentissage automatiques afin de fournir un contexte au sujet. Vient ensuite la collecte des données, puis la création de l'ensemble de données, car pour procéder à l'entraînement et au test du classificateur, un ensemble de données doit être en place. Le traitement des données, y compris le pré-traitement, l'évaluation du classificateur et les résultats, est examiné. En se basant sur l'algorithme d'apprentissage automatique le plus performant, le modèle ML sera amélioré et intégré au filtre anti-spam pour compléter l'étude. Afin de réaliser cette étude, une expérience d'émulation a été menée en utilisant un environnement comprenant un serveur virtuel avec des bibliothèques Python installées et des composants de serveur de messagerie tels que Dovecot, Postfix, Amavis, SpamAssassin et Webmail.

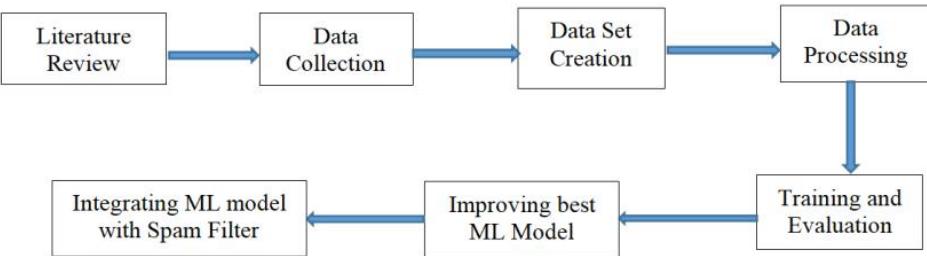


Figure 5.2 : Méthodes de recherche adoptées

5.3 Détection de Phishing par Apprentissage Automatique

5.3.1 Choix de la Base de données

Notre choix de base de données était bien exigeant afin de pouvoir bien entraîner notre modèle d'apprentissage. Nous avons sélectionné une base de données sur le site Kaggle [6.1]. Contenant plus que 18634 courriels étiquetés comme étant du phishing ou légitimes. Cette quantité importante de données est capable de détecter efficacement les emails de phishing pour les raisons suivantes :

- Détection précise du phishing : Plus un modèle d'apprentissage automatique est exposé à des exemples de phishing et d'emails légitimes, plus il est capable de faire la distinction entre les deux. Les 18634 exemples de la base données "phishing-email.csv" fournissent une matière riche pour entraîner un modèle robuste de détection de phishing.
- Amélioration de la sécurité des emails : Des modèles d'apprentissage automatique entraînés sur cette base de données peuvent être intégrés aux filtres anti-spam des messageries électroniques. Cela permet d'améliorer la protection des utilisateurs contre les attaques de phishing en identifiant et en bloquant les emails frauduleux avant qu'ils n'atteignent leur boîte de réception.
- Gain de temps et de ressources : En disposant d'un outil automatisé de détection du phishing, les entreprises peuvent gagner du temps et des

ressources qui seraient autrement consacrées à l'identification manuelle des emails frauduleux.

Cette base de données présente des extraits d'emails classifiés en fonction de leur type. Chaque entrée du tableau comprend le texte de l'email et une étiquette de classification indiquant si l'email est considéré comme sûr ou bien email de phishing.

4	software at incredibly low prices (86 % lower) . drapery seventeen term represent any sing . feet ...	Phishing Email
5	global risk management operations sally congratulations on your new role . if you were not already a...	Safe Email
6	On Sun, Aug 11, 2002 at 11:17:47AM +0100, wintermute mentioned: > > The impression I get from readin...	Safe Email

Tableau 5.1 : Extrait de notre base de données

5.3.2 Équilibrage des Classes

La base de données contient 18 634 exemples d'e-mails, répartis entre des e-mails de phishing et des e-mails sûrs, ces classes sont déséquilibrées. Le modèle peut devenir biaisé en faveur de la classe majoritaire.

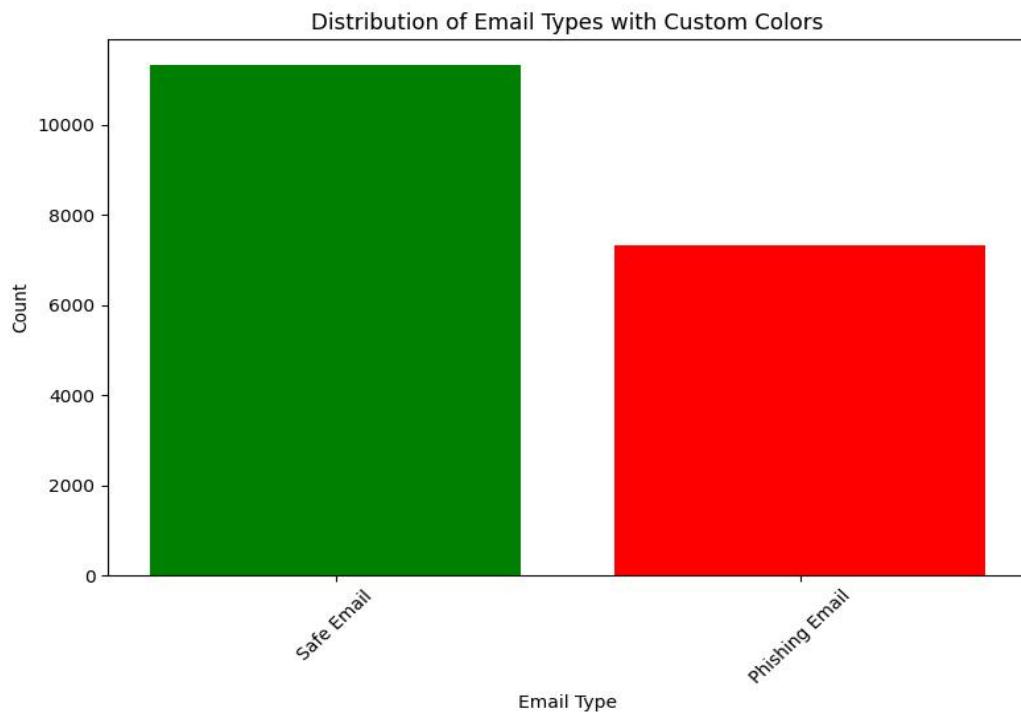


Figure 5.3 : visualisation de la distribution de type des e-mails

Cela signifie qu'il pourrait prédire la plupart des e-mails comme sûrs, même lorsqu'ils sont de phishing, ce qui est particulièrement dangereux pour des applications de sécurité. L'undersampling de la classe majoritaire (e-mails sûrs) peut aider à équilibrer les classes et à rendre le modèle plus sensible aux e-mails de phishing.

Le Sous-échantillonnage (Undersampling) : réduire le nombre d'instances dans la classe majoritaire pour correspondre à celui de la classe minoritaire, équilibrant ainsi efficacement l'ensemble de données.

```
# lets check the shape again
Safe_Email.shape,Phishing_Email.shape
:
((7312, 3), (7312, 3))
```

Figure 5.4 : visualisation de la distribution de type des e-mails après le undersampling

5.3.3 Algorithmes de Machine Learning utilisées

Dans le cadre de ce projet, nous avons exploré deux algorithmes d'apprentissage automatique pour la détection du phishing : Support Vector Machine (SVM) et Random Forest. Une comparaison approfondie de leurs performances sur notre ensemble de données a révélé que Random Forest surpassait SVM en termes de précision, atteignant un taux de 93% contre 49% pour SVM. Forts de ces résultats, nous avons opté pour Random Forest comme algorithme de détection du phishing. Ce choix se justifie par plusieurs avantages clés de Random Forest, tels que sa meilleure capacité de généralisation, sa robustesse aux et son excellente interprétabilité des résultats.

Forêt d'arbres décisionnels (Random Forest) : Il s'agit d'une application de graphe en arbres de décision permettant ainsi la modélisation de chaque résultat sur une branche en fonction des choix précédents. On prend ensuite la meilleure décision en fonction des résultats qui suivront.

```
root@mailserver:/home/imane/Desktop/phishing-detection# nano train_models.py
root@mailserver:/home/imane/Desktop/phishing-detection# python3 train_models.py
      precision    recall   f1-score   support
Phishing Email     0.91     0.96     0.94    2198
  Safe Email      0.96     0.91     0.93    2190

    accuracy         0.93    4388
   macro avg       0.94     0.93     0.93    4388
weighted avg       0.94     0.93     0.93    4388

Accuracy: 0.9348222424794895
[[2116  82]
 [ 204 1986]]
root@mailserver:/home/imane/Desktop/phishing-detection#
```

Figure 5.5 : visualisation des performances de model de ML

Le fichier `train_models.py` contient le script qui entraîne le modèle d'apprentissage automatique pour la détection du phishing sur l'ensemble de données de la base de données.

Le modèle de classification Random Forest a obtenu des résultats remarquables, avec un taux de précision de 93,1%. Il a fait preuve d'une grande capacité à classifier correctement les emails légitimes et les emails de phishing. Afin d'approfondir notre analyse de cette performance, nous allons examiner des métriques supplémentaires au-delà de la simple précision. Plus précisément, nous explorerons la précision, le

rappel et le score F1. Ces métriques nous donneront un aperçu précieux de l'efficacité du modèle dans la classification de chaque type d'email, en mettant en évidence les compromis potentiels qui peuvent exister.

5.3.4 Intégrations de modèle de machine Learning avec le service de messagerie

Après le téléchargement des bibliothèques nécessaires : numpy, panda, sklearn ... nous avons créé le répertoire /home/imane/Desktop/phishing-detection

```
root@mailserver:/home/imane/Desktop/phishing-detection#  
root@mailserver:/home/imane/Desktop/phishing-detection# ls  
file-test file-test2 filter_emails.py phishing_detection_pipeline.pkl Phishing_Email.csv random_forest_model.pkl train_models.py vectorizer.pkl  
root@mailserver:/home/imane/Desktop/phishing-detection#
```

Figure 5.6 : contenu de répertoire répertoire /home/imane/Desktop/phishing-detection

- Phishing_Email.csv : Fichier CSV qui contient la base de données utilisé pour l'entraînement.
- train_models.py : Script Python utilisé pour entraîner des modèles de machine learning pour la détection de phishing. Il inclure du code pour charger le jeu de données, prétraiter les données, entraîner les modèles et enregistrer les modèles entraînés sur le disque ainsi la generation des fichiers Pickle, fichier binaire qui contient des objets Python serialisés utilisés pour le filtrage des emails.
- vectorizer.pkl: Fichier Pickle , utilisé pour convertir des données textuelles (comme le contenu des emails) en caractéristiques numériques utilisées par les modèles de machine learning.
- filter_emails.py: le script Python responsable du filtrage des emails pour identifier et séparer les emails de phishing des emails légitimes en se basant

sur ce modèle , ce filtre sera utilisé au niveau de fichier de configuration de postfix .

Pour plus de détaille sur le code que contient ces fichiers veuillez consulter l'annexe

On teste le bon fonctionnement de ce filtre par un message de phishing que nous avons sauvegarder au niveau de fichier file-test2 comme suit :

```
root@mailserver:/home/imane/Desktop/phishing-detection# cat file-test2 | python3 filter_emails.py
Subject: [PHISHING]

Cher destinataire,

Voulez-vous gagner de l'argent rapidement et facilement ? Notre incroyable programme de gain d'argent vous permettra de gagner des milliers de dollars par semaine sans aucun effort de votre part !

Inscrivez-vous dès maintenant et commencez à gagner dès aujourd'hui !

Cliquez sur ce lien pour en savoir plus : http://your_gift.com

Ne manquez pas cette opportunité incroyable ! Rejoignez-nous maintenant !

Cordialement,
L'équipe de GainFacile
root@mailserver:/home/imane/Desktop/phishing-detection#
```

Figure 5.7 : test de fonctionnement de filtre de phishing base sur le modèle de ML

On observe que le message de phishing est correctement détecté, visualisant donc les logs pour plus de détails :

```
root@mailserver:/home/imane/Desktop/phishing-detection# tail -f /var/log/phishing_filter.log
2024-06-02 18:15:02,560 INFO Email modifié écrit vers la sortie standard
2024-06-02 18:18:59,740 INFO Pipeline chargé avec succès
2024-06-02 18:18:59,742 INFO Email lu depuis l'entrée standard
2024-06-02 18:18:59,742 INFO Corps de l'email extrait
2024-06-02 18:18:59,743 INFO Contenu du corps de l'email: Cher destinataire,

Voulez-vous gagner de l'argent rapidement et facilement ? Notre incroyable pro...
2024-06-02 18:18:59,752 INFO Résultat de la prédiction: Phishing
2024-06-02 18:18:59,753 INFO Sujet de l'email modifié pour indiquer le phishing
2024-06-02 18:18:59,754 INFO Email modifié écrit vers la sortie standard
root@mailserver:/home/imane/Desktop/phishing-detection# tail -f /var/log/mail.log
2024-06-02 18:15:02,560 INFO Email modifié écrit vers la sortie standard
2024-06-02 18:18:59,740 INFO Pipeline chargé avec succès
2024-06-02 18:18:59,742 INFO Email lu depuis l'entrée standard
2024-06-02 18:18:59,742 INFO Corps de l'email extrait
2024-06-02 18:18:59,743 INFO Contenu du corps de l'email: Cher destinataire,

Voulez-vous gagner de l'argent rapidement et facilement ? Notre incroyable pro...
2024-06-02 18:18:59,752 INFO Résultat de la prédiction: Phishing
2024-06-02 18:18:59,753 INFO Sujet de l'email modifié pour indiquer le phishing
2024-06-02 18:18:59,754 INFO Email modifié écrit vers la sortie standard
```

Figure 5.8 : le fichier /var/log/mail.log

Liant ce filtre à Postfix afin que les messages électroniques passent par ce filtre pour qu'ils soient analyser comme suit :

```
GNU nano 6.2
/etc/postfix/main.cf *
milter_protocol = 6
milter_default_action = accept
smtpd_milters = inet:127.0.0.1:12345
non_smtpd_milters = inet:127.0.0.1:12345

##Clamav
#content_filter = smtp-amavis:[127.0.0.1]:10024

##ml detect phishing
content_filter = phishfilter:dummy
receive_override_options = no_address_mappings

#postfix cyd spool argv=/usr/lib/postfix/cydia
#postfix cyd spool argv=/usr/lib/postfix/cydia

##phishing
phishfilter unix - n n - - pipe
flags=Rq user=postfix argv=/usr/local/bin/phishing_filter.sh
```

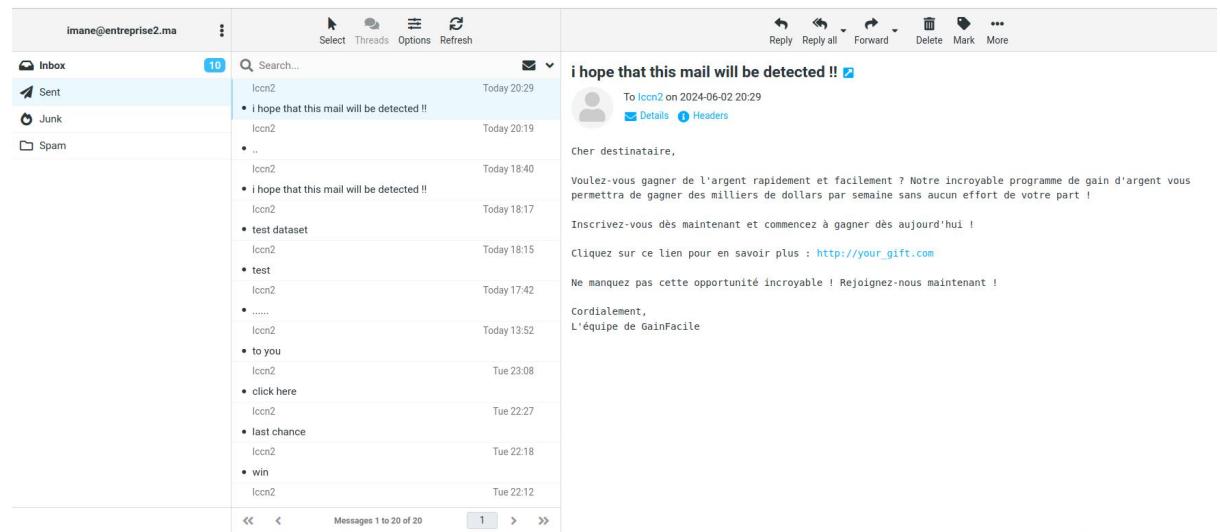
Figure 5.9 : les lignes à insérer dans les fichiers de configuration de postfix

main.cf et master.cf

Le phishing_filter.sh contient le chemin vers le filtre qu'on a défini précédemment :

/usr/bin/python3 /home/imane/Desktop/phishing-detection/filter_emails.py

On a envoyé le message de phishing que précédemment depuis imane vers iccn2, le message est bien détecté comme email de phishing :



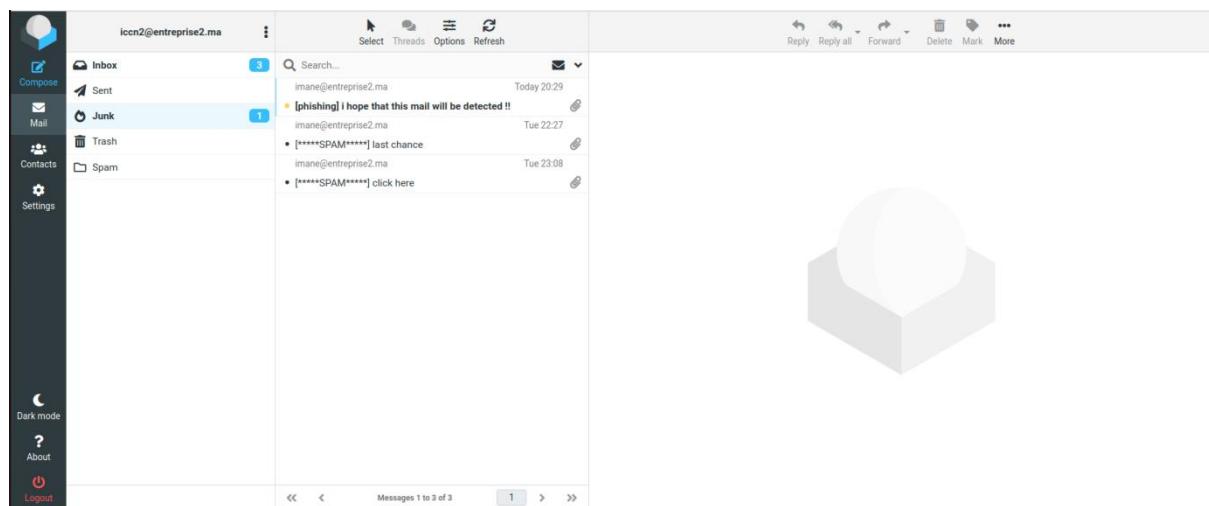


Figure 5.10 : test de filtre de ML implémenté et intégré au service de messagerie

5.4 Détection de spam par apprentissage automatique

5.4.1 Introduction au Modèle

5.4.1.1 Description de la Régression Logistique

La régression logistique est un modèle de classification linéaire qui est couramment utilisé pour résoudre des problèmes de classification binaire. Elle utilise une fonction sigmoïde pour prédire la probabilité qu'une donnée appartienne à une certaine classe. Le modèle de régression logistique estime les paramètres en maximisant la vraisemblance des données observées, ce qui permet de trouver la meilleure séparation possible entre les classes.

5.4.1.2 Pourquoi il est adapté à la détection de spam

La régression logistique est particulièrement adaptée à la détection de spam pour plusieurs raisons :

- **Simplicité et efficacité** : La régression logistique est relativement simple à implémenter et à interpréter. Elle est efficace pour les problèmes de classification binaire, tels que la détection de spam.
- **Performance** : Elle offre de bonnes performances avec des données textuelles lorsqu'elle est combinée avec des techniques de vectorisation comme le Bag of Words (BoW) et le TF-IDF.
- **Rapidement entraînable** : En comparaison avec des modèles plus complexes, la régression logistique peut être entraînée rapidement même sur des jeux de données de grande taille.
- **Généralisabilité** : Elle a une bonne capacité de généralisation, ce qui signifie qu'elle fonctionne bien sur des données qu'elle n'a pas vues auparavant.

5.4.2 Prétraitement des Données

5.4.2.1 Collecte et nettoyage des données

- **Collecte** : Les données peuvent être collectées à partir de différentes sources comme des emails, des messages instantanés, ou des commentaires sur les réseaux sociaux. Le fichier CSV `combined_data.csv` est utilisé ici comme source de données.
- **Nettoyage** : Le nettoyage des données implique la suppression des éléments inutiles comme les balises HTML, la conversion en minuscules, la suppression des stopwords, la correction orthographique, et la suppression des caractères spéciaux.

5.4.2.1 Extraction des caractéristiques (features)

- **Vectorisation** : Les caractéristiques sont extraites du texte en utilisant des techniques de vectorisation comme le Bag of Words (BoW) et le TF-IDF.

- **Bag of Words (BoW)** : Convertit le texte en une matrice de comptage des mots.
- **TF-IDF** : Convertit le texte en une matrice de scores TF-IDF, pondérant les mots en fonction de leur fréquence et de leur importance dans le corpus.

5.4.2.3 Transformation des données

- **Conversion en sac de mots** : La transformation des données en utilisant BoW ou TF-IDF vectorise le texte en une représentation numérique qui peut être utilisée comme entrée pour le modèle de régression logistique

5.4.3 Implémentation du Modèle

5.4.3.1 Algorithmes utilisés

- Régression Logistique : Utilisation de l'algorithme de régression logistique pour la classification.

5.4.3.2 Paramètres et hyperparamètres

- Paramètres : Le modèle utilise les coefficients des caractéristiques pour faire des prédictions.
- Hyperparamètres :
- max_iter : Le nombre maximum d'itérations pour la convergence de l'algorithme.
- solver : L'algorithme utilisé pour l'optimisation (par exemple, 'liblinear').

5.4.3.3 Implémentation dans le Script filtrml.py

Nous avons implémenté notre modèle de détection de spam dans un script Python nommé filtrml.py. Ce script contient le code pour charger les données, les prétraiter,

entraîner le modèle de régression logistique et sauvegarder les modèles entraînés.

- Chargement et prétraitement des données : Le script commence par charger les données à partir du fichier CSV et les vectorise en utilisant BoW et TF-IDF.
- Entraînement des modèles : Deux modèles de régression logistique sont entraînés, l'un utilisant BoW et l'autre TF-IDF.
- Sauvegarde des modèles : Les modèles entraînés et les vectoriseurs sont sauvegardés dans des fichiers .pkl pour une utilisation future

5.4.4 Intégration avec postfix

5.4.4.1 Crédation du filtre de contenu

Utiliser un filtre de contenu basé sur milter pour intercepter les emails entrants. Un milter (Mail Filter) est un programme qui peut être utilisé par Postfix pour filtrer les emails.

Le filtre de contenu appelle le script filterml.py pour analyser le contenu de chaque email et déterminer s'il s'agit de spam ou non.

5.4.4.2 Configuration de Postfix pour utiliser le filtre

Modifier le fichier de configuration de Postfix (main.cf) pour inclure le filtre de contenu.

```
#m1
content_filter = myfilter:localhost:10025

smtpd_milters = unix:/var/spool/postfix/milter.sock
non_smtpd_milters = $smtpd_milters
milter_default_action = accept
milter_protocol = 2
```

Figure 5.11 : /etc/postfix/main.cf

On doit aussi modifier dans le fichier /etc/postfix/master.cf

```
#m1
myfilter unix - n n - - pipe
  flags=F user=nobody argv=~/part3/AI/AI/filterml.py
```

Figure 5.12 : /etc/postfix/master.cf

5.4.4.3 Évaluation du Modèle

5.4.4.3.1 entraînement de données

L'entraînement de notre modèle de détection de spam repose sur l'utilisation d'une régression logistique. Nous avons commencé par collecter un ensemble de données équilibré, comprenant à la fois des e-mails spam et légitimes. Ensuite, nous avons prétraité les données en nettoyant le texte, en le normalisant et en le vectorisant à l'aide du Bag of Words (BoW) et du TF-IDF. Après avoir divisé les données en ensembles d'entraînement et de test, nous avons entraîné le modèle de régression logistique sur l'ensemble d'entraînement. Pendant l'entraînement, le modèle a ajusté ses paramètres pour minimiser l'erreur de prédiction. Enfin,

nous avons évalué les performances du modèle sur l'ensemble de test pour estimer sa capacité à généraliser et à détecter les spams efficacement.

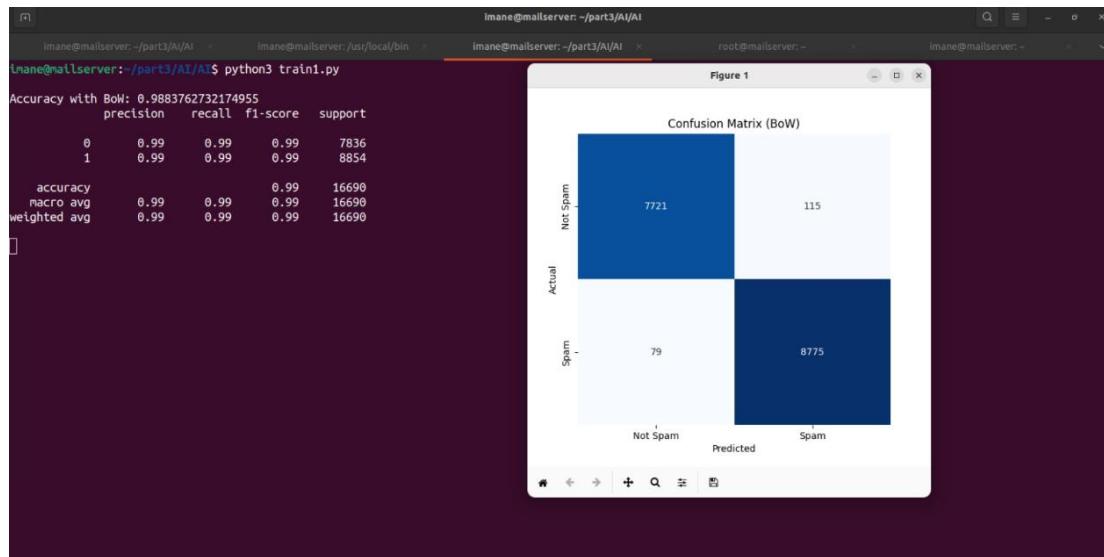


Figure 5.13 : résultat de l'entraînement de données

5.4.4.3.2 test du Model

Pour se faire nous avons crée un fichier test3.eml qui contient des mots et une url malicieuse

```
imane@mailserver:~/part3/AI/AI$ cat test3.eml
From: spammer@example.com
To: recipient@example.com
Subject: Important Information

Dear User,

We regret to inform you that your account has been compromised.
To secure your account, please click on the following link:

http://maliciouswebsite.com

Thank you for your attention.

Sincerely,
Spammer
cordialement

imane@mailserver:~/part3/AI/AI$
```

Figure 5.14 : fichier de test du model

```

Prédiction: ham
Cet email est classé comme ham
Texte de l'email:

Prédiction: ham
Cet email est classé comme ham
Texte de l'email: We regret to inform you that your account has been compromised.

Prédiction: spam
Cet email est classé comme spam
Texte de l'email: To secure your account, please click on the following link:

Prédiction: spam
Cet email est classé comme spam
Texte de l'email:

Prédiction: spam
Cet email est classé comme spam
Texte de l'email: http://maliciouswebsite.com

Prédiction: spam (URL détectée)
Cet email est classé comme spam
Texte de l'email:

Prédiction: ham
Cet email est classé comme ham
Texte de l'email: Thank you for your attention.

Prédiction: ham
Cet email est classé comme ham
Texte de l'email:

Prédiction: ham
Cet email est classé comme ham
Texte de l'email: Sincerely,

Prédiction: ham
Cet email est classé comme ham
Texte de l'email: Spammer

```

Figure 5.15 : Résultat du test

5.4.4.3.3 Evaluation du model

Les performances de notre modèle de détection de spam sont évaluées à l'aide de différentes mesures telles que la précision, le rappel et le F1 Score. Pour la représentation Bag of Words (BoW), nous obtenons une précision de 98,23%, un rappel de 95,86% et un F1 Score de 97,03%. En revanche, pour la représentation TF-IDF, nous obtenons une précision parfaite de 100%, mais un rappel de 84,48% et un F1 Score de 91,59%. Ces résultats indiquent que le modèle BoW a une légère baisse de précision mais une meilleure capacité à rappeler les spams par rapport au modèle TF-IDF, qui, bien que présentant une précision parfaite, a un rappel légèrement inférieur, ce qui pourrait indiquer une moindre capacité à détecter tous les spams dans l'ensemble de données.

```

lmane@mailserver:~/part3/AI/AI$ cat test3.eml | python3 filterml.py
/home/lmane/part3/AI/AI/filterml.py:20: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chainable
method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting
.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col]
form the operation inplace on the original object.

data['text'].fillna('', inplace=True)
BoW - Précision : 0.9823321554770318, Rappel : 0.9586206896551724, F1 Score : 0.9703315881326352
TF-IDF - Précision : 1.0, Rappel : 0.8448275862068966, F1 Score : 0.9158878504672897

```

Figure 5.16 : Résultat du test de performances du model

Chapitre 6

Surveiller et protéger la boîte aux lettres

La messagerie électronique est devenue un outil incontournable pour la communication et la collaboration au sein des entreprises. Cependant, cette omniprésence s'accompagne malheureusement de risques croissants, tels que les attaques par hameçonnage, l'infiltration de logiciels malveillants ou encore les fuites de données confidentielles. Face à ces menaces, il est primordial pour les organisations de se doter d'outils de surveillance et de sécurisation de leur système de messagerie.

C'est dans ce contexte que l'intégration de la solution Rspamd s'avère être une option particulièrement intéressante. Rspamd est un puissant outil de filtrage et d'anti-spam, capable d'analyser en profondeur le contenu des emails entrants pour détecter toute activité suspecte ou malveillante. S'appuyant sur des techniques d'apprentissage automatique et une base de données de menaces constamment mise à jour, Rspamd offre une protection renforcée contre les risques liés à la messagerie d'entreprise.

6.1 Pourquoi Rspamd :

6.1.1 : Description de l'outil :



Figure 6.1 : Logo Rspamd

Un puissant outil de filtrage et d'anti-spam pour sécuriser notre messagerie Rspamd est une solution de filtrage de messagerie avancée, qui se distingue par ses performances élevées et sa grande flexibilité. Conçu pour fonctionner à grande échelle, ce logiciel open source est particulièrement adapté aux besoins des entreprises en matière de sécurisation de la messagerie.

6.1.2: Caractéristiques de l'outil:

- Analyse de contenu avancée : Rspamd examine en profondeur le contenu des emails (pièces jointes, liens, entêtes, etc.) en utilisant des techniques d'apprentissage automatique pour détecter les menaces.
- Moteur anti-spam puissant : Grâce à ses règles de filtrage évolutives et sa base de données de menaces mise à jour en continu, Rspamd offre une protection de premier plan contre le spam, l'hameçonnage et autres tentatives d'intrusion par email.
- Haute performance : Capable de traiter des volumes de messagerie importants, Rspamd se distingue par sa grande rapidité d'exécution et sa faible consommation de ressources système.
- Modularité et extensibilité : La structure modulaire de Rspamd permet une

personnalisation poussée selon les besoins spécifiques de votre entreprise. De nombreuses extensions sont également disponibles.

- Multiplateforme : Rspamd fonctionne sur une grande variété de système d'exploitation (Linux, BSD, macOS, Windows), facilitant son intégration dans votre infrastructure existante.

6.1.3: Avantages de Rspamd :

Avantages clés pour la sécurité de la messagerie :

- Protection renforcée contre les menaces véhiculées par email (malware, hameçonnage, fuites de données, etc.)
- Réduction significative des risques liés à la messagerie d'entreprise
- Gain de temps et d'efficacité grâce à l'automatisation du filtrage
- Personnalisation et configuration fine des règles de sécurité
- Intégration transparente dans votre environnement IT existant

6.1.3 : Principe de fonctionnement de Rspamd :

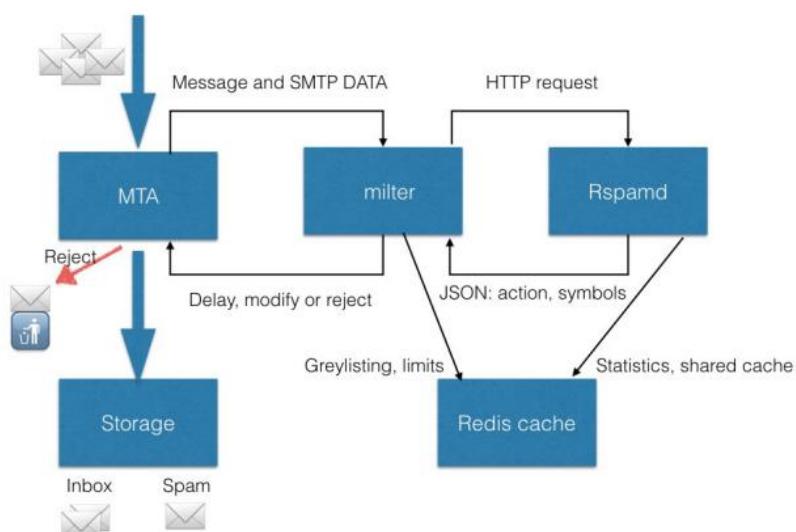


Figure 6.2 : Fonctionnement de Rspamd

Le fonctionnement de Rspamd repose sur une analyse approfondie du contenu des emails entrants, combinant plusieurs techniques avancées de détection des menaces.

Analyse du contenu :

Rspamd examine en détail les différents éléments constitutifs des emails : pièces jointes, liens, en-têtes, corps du message, etc.

Des moteurs d'analyse spécialisés sont utilisés pour identifier les signaux révélateurs de comportements malveillants, tels que la présence de code malveillant, de tentatives d'hameçonnage ou de fuites d'informations sensibles.

Apprentissage automatique :

Rspamd s'appuie sur des modèles d'apprentissage machine entraînés sur de vastes bases de données d'emails légitimes et malveillants.

Ces modèles permettent de détecter les patterns caractéristiques des messages frauduleux, en se basant sur des centaines de critères différents.

L'algorithme ajuste en permanence ses modèles grâce à l'analyse des nouveaux emails, afin d'améliorer continuellement ses capacités de détection.

Base de données de menaces :

Rspamd maintient une base de données évolutive des dernières menaces identifiées au niveau mondial.

Cette base de données, constamment mise à jour, permet de détecter les signatures connues de logiciels malveillants, d'URLs dangereuses ou d'autres indicateurs de compromission.

Règles de filtrage personnalisables :

En plus des fonctionnalités d'analyse automatique, Rspamd offre la possibilité de définir des règles de filtrage personnalisées.

Les administrateurs peuvent ainsi ajouter leurs propres critères de détection, adaptés aux spécificités de leur organisation.

Ces règles peuvent par exemple cibler certains expéditeurs, types de fichiers ou modèles d'emails suspects.

6.2 Fonctionnalités

6.2.1 Globale

- Rspamd possède une interface web d'administration que nous présentons après
- Il fonctionne avec les principaux MTA open source (Postfix, Exim, Sendmail ...).
- Rspamd intègre des centaines de règles dans le langage Lua et il est également possible d'en ajouter en s'inspirant des règles existantes.
- Et il permet l'utilisation de listes dynamiques téléchargeables en http(s) ou présents dans un fichier sans nécessiter de redémarrage.

6.2.2 Analyse de contenu

- Rspamd est capable de s'interfacer avec plusieurs antivirus : ClamAV, F-Prot, Sophos, Avira et Kaspersky.
- Le filtrage par expressions régulières offre un traitement de base des messages, de leurs parties textuelles, des en-têtes MIME et des données SMTP reçues par le MTA par rapport à un ensemble d'expressions qui inclut à la fois des expressions régulières normales et des fonctions de traitement des messages.
- Les « Fuzzy hashes » ou hachages flous sont utilisés par Rspamd pour identifier des messages similaires. Ces structures visent à masquer les petites différences entre les messages de spam afin de trouver rapidement des messages communs. Cette base peut être alimentée directement depuis des remontées utilisateurs ou via des données collectées par des honeypots. Pour l'instant, au vu de la qualité des données remontées par nos utilisateurs, nous avons préféré ne pas utiliser ce module.
- DCC est assez similaire au module précédent, mais il utilise le service externe éponyme pour vérifier si un message est envoyé en masse.
- Le module « Chartable » permet de trouver des messages spécialement conçus pour tromper les systèmes de filtrage du spam en changeant la langue du texte et en remplaçant des lettres par leurs analogues visuellement identiques. Rspamd utilise la

normalisation UTF-8 pour détecter et filtrer ces techniques couramment utilisées par de nombreux spameurs

6.2.3 Vérification de la politique de filtrage

Un ensemble de modules permet d'évaluer les messages transitant par le MTA.

- Les contrôles SPF⁴ permettent de valider la source d'un message en utilisant la politique définie dans l'enregistrement DNS du domaine de l'expéditeur
- La politique DKIM⁵ valide la signature cryptographique d'un message par rapport à une clé publique placée dans l'enregistrement DNS du domaine de l'expéditeur. Cette méthode permet de s'assurer qu'un message a été reçu du domaine spécifié sans être altéré en chemin. Rspamd permet également d'appliquer la signature DKIM pour les messages envoyés depuis notre domaine pour les utilisateurs authentifiés.
- DMARC⁶ combine les techniques DKIM et SPF pour définir des politiques plus ou moins restrictives pour certains domaines. Rspamd peut également stocker des données pour les rapports DMARC dans la base de données Redis
- Le plugin « IP reputation » permet d'ajuster progressivement la réputation des adresses IP spécifiques, des réseaux, des blocs autonomes (ASN) et même des pays en fonction de la proportion de spam reçu.
- Le module « Rate Limits » permet d'empêcher l'envoi de courriels en masse à partir d'un de nos comptes piratés. Il est également possible de définir des seuils en fonction de l'IP, de l'adresse e-mail source

6.3 Interface d'administration

Rspamd propose une interface d'administration simple permettant d'effectuer les opérations courantes. Elle est accessible via un navigateur en se connectant sur l'un des nœuds du cluster.



Figure 6.3 : Page d'accueil de Rspamd

La page d'accueil permet de connaître l'état global du cluster, de connaître la version déployée, de vérifier que chaque nœud possède la même configuration et d'avoir des statistiques globales depuis la dernière réinitialisation des compteurs.

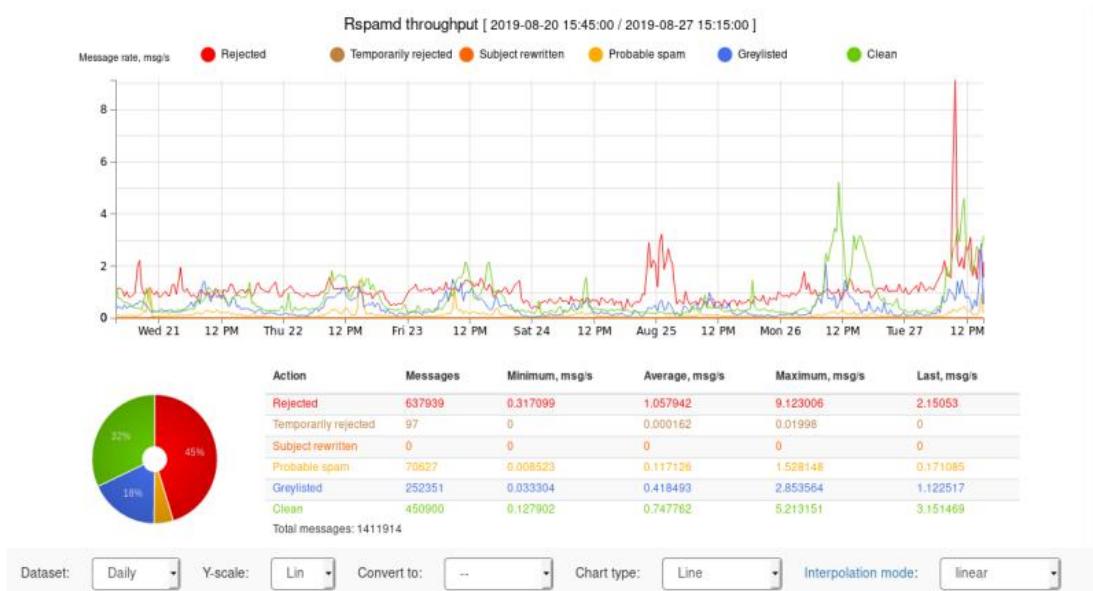


Figure 6.4 : Onglet « Throughput »

Le deuxième onglet permet d'avoir un aperçu sous forme de graphique et de tableau du nombre de messages traités, ainsi que leur répartition dans le temps. Ces données sont consultables par jour, semaine, mois ou année.

The screenshot shows the RSPAMD configuration interface. At the top, there is a navigation bar with tabs: Status, Throughput, Configuration (which is selected), Symbols, Scan/Learn, and History. There are also Refresh and Disconnect buttons.

Actions:

Action	Value
Greylist	5
Probably Spam	8
Rewrite subject	
Spam	20

Sauvegarde : Save actions, Save cluster

Lists:

Type	Path	Description
Read Write	/var/lib/rspamd/rspamd_dynamic	Dynamic configuration map
Read Write	/var/lib/rspamd/spf_whitelist.inc.local	Whitelist map for WHITELIST_SPF
Read Write	/var/lib/rspamd/dkim_whitelist.inc.local	Whitelist map for WHITELIST_DKIM
Read Write	/var/lib/rspamd/dmrc_whitelist.inc.local	Whitelist map for WHITELIST_DMARC
Read Write	/var/lib/rspamd/spf_dkim_whitelist.inc.local	Whitelist map for WHITELIST_SPF_DKIM
Read	/etc/rspamd/local.d/spamtrap.map	Spamtrap map for %s
Read Write	/var/lib/rspamd/ratelimit-user-wl.map	Ratelimit whitelist user map
Read Write	/var/lib/rspamd/ratelimit-ip-wl.map	Ratelimit whitelist ip map
Read	/etc/rspamd/maillist.inc	Exclude specific domains from MX checks

Figure 6.5 : Onglet « Configuration »

L'onglet configuration permet dans la partie haute de la page de modifier les seuils d'actions sur l'ensemble du cluster. Dans la partie basse, il est possible de modifier les listes multimap en un clic sur l'ensemble du cluster. Il est possible d'ajouter une expression régulière, une chaîne de caractère, une adresse e-mail, un domaine, une IP, un pays. Toutes ces listes seront analysées et permettront l'ajout d'un symbole et d'un score sur les messages analysés.

The screenshot shows the RSPAMD configuration interface. At the top, there is a navigation bar with tabs: Status, Throughput, Configuration (selected), Symbols (selected), Scan/Learn, and History. There are also Refresh and Disconnect buttons.

Symbols and rules:

Search: antivirus

Group	Symbol	Description	Score	Frequency	Avg. time	Save	Save in cluster
antivirus	CLAM_VIRUS_ENCRYPTED		0	0.00e-3	0.00s	<button>Save</button>	<button>Save in cluster</button>
antivirus	JUST_EICAR		0	0.00e-3	0.00s	<button>Save</button>	<button>Save in cluster</button>
antivirus	CLAM_VIRUS_FAIL		0	0.00e-3	0.00s	<button>Save</button>	<button>Save in cluster</button>
antivirus	CLAM_VIRUS		0	0.00e-3	0.00s	<button>Save</button>	<button>Save in cluster</button>

Figure 6.6 :Onglet « Symbols »

Le dernier onglet « History » permet d'avoir une vision en temps réel des messages analysés sur l'ensemble des nœuds de la plateforme anti-spam

The screenshot shows the Rspamd History interface. At the top, there are tabs for All SERVERS, Status, Throughput, Configuration, Symbols, Scan/Learn, and History. The History tab is selected. Below the tabs, there are filters for 'reject' and a search bar. The main area displays a table of messages with columns: ID, IP address, [Envelope From] From, [Envelope To] To/Cc/Bcc, Subject, Action, Score, Msg size, Scan time, Time, If Authenticated user. The table lists various messages with their analysis results, such as 'reject' actions, scores, and timestamps.

Figure 6.7 :Onglet « History »

6.4 Mise en place avec postfix

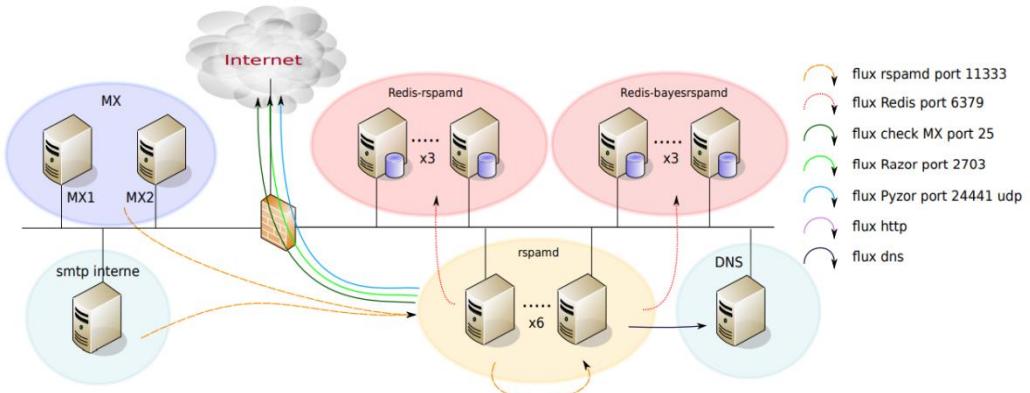


Figure 6.8 : Schéma de flux pour Rspamd

L'idée, lors de la mise en place de Rspamd, était d'obtenir la plus grande transparence possible envers les utilisateurs en ce qui concerne les messages mis en quarantaine. Lorsque le message est probablement indésirable, Rspamd modifie son en-tête afin que le store de messagerie — via une règle sieve9 globale — délivre ce message douteux dans le dossier Junk de l'utilisateur.

6.4.1 Installation de Rspamd

L'installation du composant principal de Rspamd, à savoir le démon Spamd, se fait de manière relativement simple.

```
imane@mailserver:/etc/spamassassin$ sudo apt-get install rspamd
[sudo] password for imane:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
rspamd is already the newest version (2.7-1build2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
imane@mailserver:/etc/spamassassin$
```

Figure 6.9: Installation de Rspamd

Pour vérifier que l'installation est bien faite on doit voir des fichiers qui ressemblent à Ça dans le répertoire /etc/rspamd

```
imane@mailserver:/etc/rspamd$ ls
actions.conf    composites.conf  local.d      metrics.conf  options.inc   scores.d      test.eml    worker-normal.inc
cgp.inc         dkim           logging.inc  modules.conf  override.d    settings.conf  worker-controller.inc  worker-proxy.inc
common.conf     groups.conf    maps.d      modules.d    rspamd.conf  statistic.conf  worker-fuzzy.inc
imane@mailserver:/etc/rspamd$
```

Figure 6.10 : le répertoire /etc/rspamd

La configuration principale de cette solution se fait dans le fichier rspamd.conf

```
worker "controller" {
    bind_socket = "localhost:11334";
    .include "$CONFDIR/worker-controller.inc"
    .include(try=true; priority=1,duplicate=merge) "$LOCAL_CONFDIR/local.d/worker-controller.inc"
    .include(try=true; priority=10) "$LOCAL_CONFDIR/override.d/worker-controller.inc"
}

worker "rspamd_proxy" {
    bind_socket = "localhost:11332";
    .include "$CONFDIR/worker-proxy.inc"
    .include(try=true; priority=1,duplicate=merge) "$LOCAL_CONFDIR/local.d/worker-proxy.inc"
    .include(try=true; priority=10) "$LOCAL_CONFDIR/override.d/worker-proxy.inc"
}

# Local fuzzy storage is disabled by default

worker "fuzzy" {
    bind_socket = "localhost:11335";
    count = -1; # Disable by default
    .include "$CONFDIR/worker-fuzzy.inc"
    .include(try=true; priority=1,duplicate=merge) "$LOCAL_CONFDIR/local.d/worker-fuzzy.inc"
    .include(try=true; priority=10) "$LOCAL_CONFDIR/override.d/worker-fuzzy.inc"
}
```

Figure 6.11 : /etc/rspamd/rspamd.conf

6.4.1 Rspamd et postfix

Pour se faire, on doit modifier les deux fichier /etc/postfix/main.cf et

/etc/postfix/master.cf pour intégrer Rspamd qui va surveiller notre serveur de messagerie pour que le message, une fois reçu, va être traité par le proxy de rspamd afin d'identifier s'il s'agit de spam ou non.

Après avoir faire cette configuration, on modifiant et adaptant les règles de filtrages qu'on veut mettre en place pour le traitement des emails, on peut accéder à l'interface web de rspamd <http://localhost:11334> pour voir les statistiques liées au serveur .

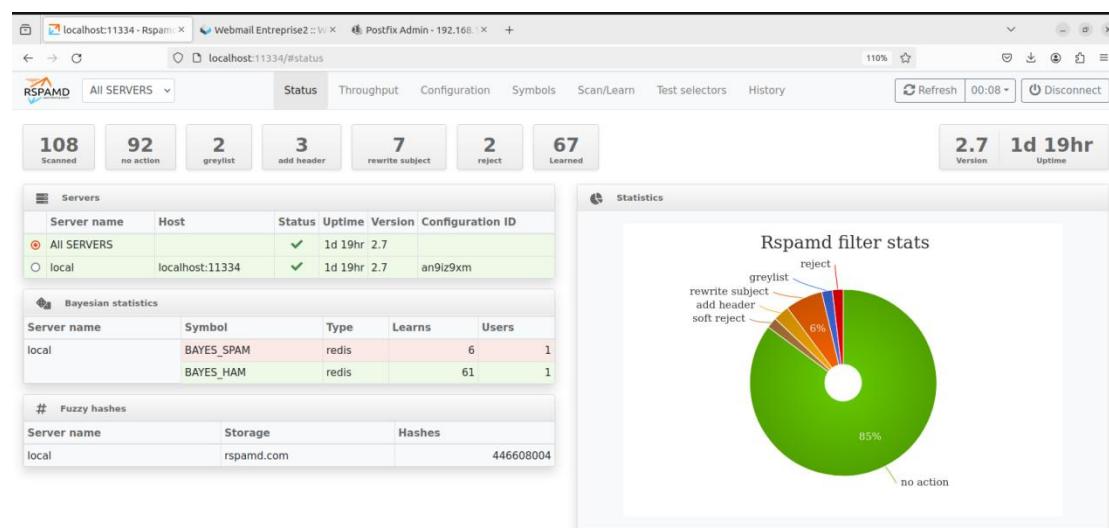


Figure 6.12 : Status du filtre Rspand

On peut même consulter les statistiques quotidiennes dans la partie Throughput

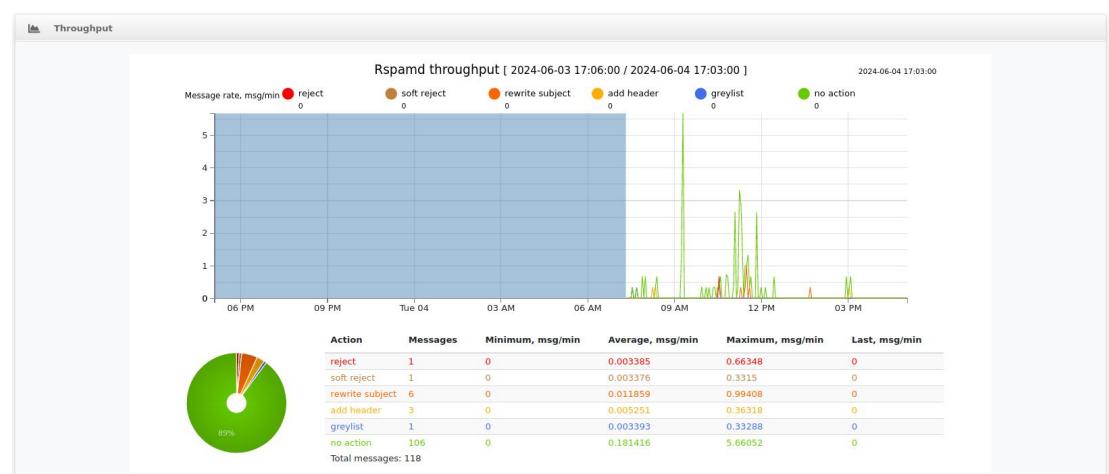


Figure 6.13 : Statistiques de Rspamd

Nous avons paramétré nos nœuds Rspamd pour qu'ils stockent dans la base Redis les informations des 500 derniers messages analysés, soit un total de 3000 messages dans le cas d'un cluster de 6 nœuds. Cela nous permet de visualiser les messages des dix dernières minutes environ. Un nombre de messages trop important rend l'onglet « History » inutilisable

ID	IP address	Action	Score	Message size	Scan time	Time	Authenticated user
- 836314c45ec3e583211a50e06725082@entre...	127.0.0.1	no action	0.00 / 10	808	0.017908	6/4/2024, 2:47:06 PM	
Symbols							
- 836314c45ec3e583211a50e06725082@entre...	192.168.187.128	no action	-0.10 / 10	332	0.017276	6/4/2024, 2:47:02 PM	imane@entreprise2.ma
Symbols							
- 79d2c0da785e8763c881e88a49b6a5e1@entre...	127.0.0.1	add header	5.00 / 10	3.23k	0.34657	6/4/2024, 2:45:36 PM	
Symbols							
+ 79d2c0da785e8763c881e88a49b6a5e1@entre...	192.168.187.128	no action	-0.10 / 10	547	0.440829	6/4/2024, 2:45:35 PM	iccn2@entreprise2.ma
+ 16ba5412443e0eeb7702c6abf8905e15@entre...	127.0.0.1	no action	0.00 / 10	989	0.423179	6/4/2024, 2:38:36 PM	
+ 16ba5412443e0eeb7702c6abf8905e15@entre...	192.168.187.128	no action	-0.10 / 10	549	0.567802	6/4/2024, 2:38:30 PM	iccn2@entreprise2.ma
+ 2024060315553.9586C18199@mail.entrepre...	127.0.0.1	rewrite subject	1.90 / 10	99.6k	0.820467	6/3/2024, 3:55:53 PM	
+ 20240602232122.BA50218199C@mail.entrepre...	127.0.0.1	no action	-0.10 / 10	44.8k	0.356394	6/2/2024, 11:21:22 PM	
+ 20240602232103.CA3FA18199C@mail.entrepre...	127.0.0.1	no action	-0.10 / 10	44.7k	0.46126	6/2/2024, 11:21:03 PM	

Figure 6.14 : Historique de Rspamd

6.5 Pflogsumm pour la Surveillance de l'Activité de Postfix

pflogsumm.pl est un outil puissant conçu pour analyser les fichiers de log de Postfix et fournir une vue d'ensemble claire et détaillée de l'activité du serveur de messagerie. Cet outil est particulièrement utile pour les administrateurs système qui souhaitent surveiller la performance, détecter les anomalies, et identifier les points chauds potentiels dans leur infrastructure de messagerie. Ce rapport décrit l'installation, la configuration et les bénéfices de l'utilisation de pflogsumm pour la surveillance de l'activité de Postfix.

6.5.1 Fonctionnement de pflogsum

En seulement quelques jours, le fichier de logs d'un serveur modeste de messagerie peut cumuler des milliers d'entrée, rendant la gestion et l'analyse de ces données complexes. Pour répondre efficacement à la question essentielle "Qu'est-ce qui se passe sur notre serveur mail?", nous avons adopté Pflogsumm (Postfix Log Summary). Pflogsumm est un outil écrit en Perl, qui permet de générer des rapports détaillés à partir des logs de Postfix. Grâce à ces rapports, nous pouvons obtenir une vue d'ensemble claire et précise de l'activité du serveur de messagerie, facilitant ainsi la surveillance, l'identification des problèmes et la prise de décisions éclairées pour la gestion de notre infrastructure de messagerie.

Pflogsumm fournit une vue d'ensemble de l'activité de Postfix avec divers détails importants :

- **Nombre total de messages envoyés et reçus** : Cela donne une idée du volume de trafic de messagerie.
- **Messages rejetés, retardés, et rebondis** : Ces métriques aident à identifier les problèmes potentiels dans le flux de messages.
- **Délai de livraison** : Le temps moyen que prend un message pour être livré, ce qui peut indiquer des problèmes de performance.
- **Top des émetteurs et destinataires** : Identifie les utilisateurs ou adresses email les plus actifs, ce qui peut être utile pour détecter un usage abusif ou une activité suspecte.
- **Analyse des erreurs** : Fournit des détails sur les types d'erreurs rencontrées, aidant à diagnostiquer des problèmes spécifiques.

6.5.2 installation et configuration de Pflogsum

Pour utiliser pflogsumm, il doit d'abord être installé sur le serveur de messagerie par la commande :

```
root@mailserver:/# apt-get install pflogsumm
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
libbit-vector-perl libcage-clue-perl libdata-salv-perl libdata-salv-xs-perl
```

Figure 6.15 : installation de pflogsum

Une fois le package de pflogsum est installé , nous pouvons l'exécuter manuellement pour générer des rapports à partir des fichiers de log de Postfix. Voici une commande typique pour générer un rapport sur l'activité de la journée précédente :

```
root@mailserver:/# perl /usr/sbin/pflogsumm -d today /var/log/mail.log
Postfix log summaries for Jun 2

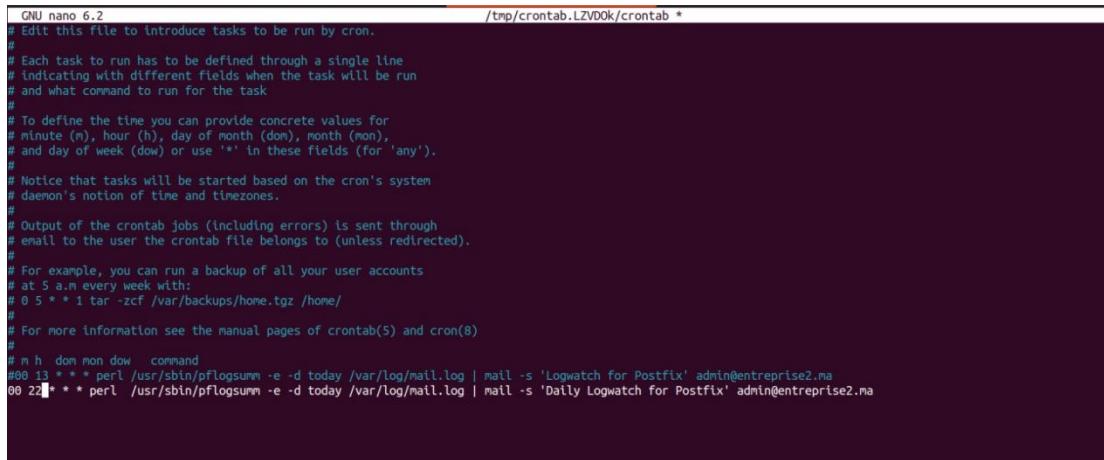
Grand Totals
-----
messages
    13 received
    12 delivered
    0 forwarded
    0 deferred
    2 bounced
    0 rejected (0%)
    0 reject warnings
    0 held
    0 discarded (0%)

113333 bytes received
14472 bytes delivered
  3 senders
   1 sending hosts/domains
   2 recipients
   1 recipient hosts/domains

Per-Hour Traffic Summary
-----
time      received  delivered  deferred  bounced  rejected
-----  -----
0000-0100      0        0        0        0        0
0100-0200      0        0        0        0        0
0200-0300      0        0        0        0        0
```

Figure 6.16 : extrait de rapport généré

Pour automatiser ce processus et recevoir des rapports quotidiens, nous pouvons configurer une tâche cron. Voici comment ajouter une entrée dans le crontab pour exécuter pflogsumm tous les jours à 22:00h et envoyer le rapport par email à l'administrateur : admin@entreprise2.ma.



```

GNU nano 6.2
/tmp/crontab.LZvD0k/crontab *
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m. every week:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
#00 13 * * * perl /usr/sbin/pflogsumm -e -d today /var/log/mail.log | mail -s 'Logwatch for Postfix' admin@entreprise2.ma
#00 22 * * * perl /usr/sbin/pflogsumm -e -d today /var/log/mail.log | mail -s 'Daily Logwatch for Postfix' admin@entreprise2.ma

```

Figure 6.17 : configuration au niveau de crontab

Par cette configuration l'administrateur reçoit chaque jour à 22.00h un rapport sur l'activité de serveur de messagerie :

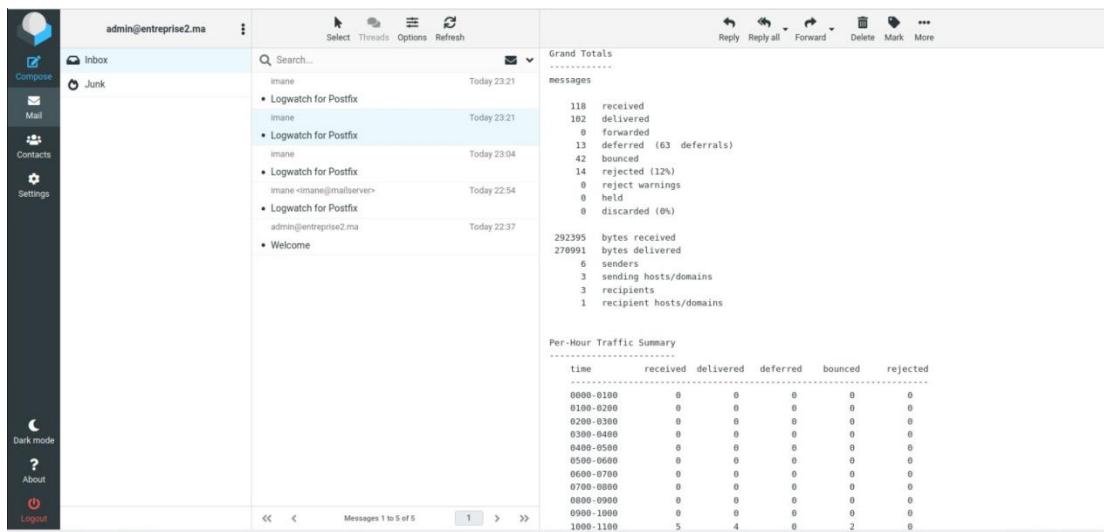


Figure 6.18 : rapport envoyé à l'administrateur

6.6 Mailgraph pour la Surveillance de serveur de messagerie

Pour compléter notre surveillance de l'activité de notre serveur de messagerie Postfix, nous avons intégré Mailgraph, un outil essentiel pour la visualisation des performances et des tendances de messagerie. Mailgraph utilise RRDtool pour créer des graphiques détaillés et en temps réel de l'activité de messagerie. Il génère des graphiques quotidiens, hebdomadaires, mensuels et annuels, fournissant ainsi une vue

d'ensemble continue de l'utilisation du serveur de messagerie.

6.6.1 Fonctionnalités de Mailgraph

Mailgraph présente plusieurs fonctionnalités :

- **Graphiques Détailés** : Mailgraph produit des graphiques qui montrent les volumes de courriels envoyés, reçus, rejetés et rebondis. Ces graphiques permettent d'identifier rapidement les tendances et les anomalies.
- **Visualisation en Temps Réel** : Les graphiques sont mis à jour en temps réel, offrant une surveillance constante de l'activité de messagerie.
- **Historique des Données** : En conservant un historique des données de messagerie, Mailgraph permet une analyse approfondie des performances à long terme. Cela aide à détecter les variations saisonnières ou les changements progressifs dans l'utilisation du serveur.
- **Facilité de Déploiement** : Mailgraph est simple à déployer et à configurer. Une fois installé, il fonctionne automatiquement en arrière-plan, collectant les données et générant les graphiques sans intervention manuelle.

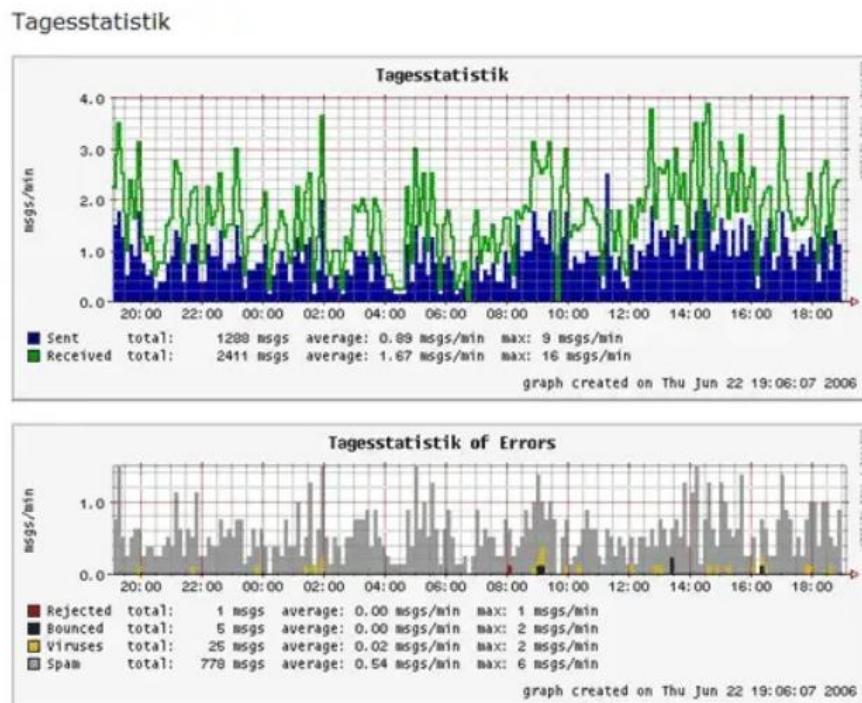


Figure 6.19 : exemple de statistique graphique fournit par Mailgraph

6.6.2 Implémentation de Mailgraph

On installe le package de Mailgraph depuis le site :

<http://mailgraph.schweikert.ch/pub/mailgraph-1.14.tar.gz> puis en la compresse et on obtient le package de Mialgraph comme suit :

```
mailgraph-1.14.tgz
root@mailserver:/# tar -xvf mailgraph-1.14.tar.gz /etc/mailgraph
tar: /etc/mailgraph: Not found in archive
tar: Exiting with failure status due to previous errors
root@mailserver:/# ls
${SAHOME}spand.log' dev index.html lib64 mailgraph-1.14
bin etc lib lib32 mailgraph-1.14.tar.gz mnt root snap sys var
boot home lib32 lost+found media opt run srv tmp visutotal.sh.save
root@mailserver:/# cd mailgraph-1.14#
root@mailserver:/# ls
CHANGES COPYING mailgraph.cgi mailgraph.css mailgraph-init mailgraph.pl README
root@mailserver:/mailgraph-1.14#
```

Figure 6.20 : Mailgraph package

Ou avec la commande apt-get install mailgraph.

Nous avons configuré Mailgraph pour qu'il surveille les fichiers de log de Postfix et qu'il enregistre les données dans un fichier RRD. La configuration a été adaptée pour s'assurer que Mailgraph collecte et affiche les données pertinentes.

```
GNU nano 6.2                               /usr/lib/cgi-bin/mailgraph.cgi

#!/usr/bin/perl -w

# mailgraph -- postfix mail traffic statistics
# copyright (c) 2000-2007 ETH Zurich
# copyright (c) 2000-2007 David Schweikert <david@schweikert.ch>
# released under the GNU General Public License

use RRDs;
use POSIX qw(uname);
my $VERSION = "1.14";

my $host = (POSIX::uname())[1];
my $scriptname = 'mailgraph.cgi';
my $xpoints = 540;
my $points_per_sample = 3;
my $points = 160;
my $points_err = 96;
my $rrd = '/var/log/mailgraph.rrd'; # path to where the RRD database is
my $rrd_virus = 'mailgraph_virus.rrd'; # path to where the Virus RRD database is
my $tmp_dir = '/tmp/mailgraph'; # temporary directory where to store the images

my @graphs = (
    { title => 'Last Day', seconds => 3600*24, },
    { title => 'Last Week', seconds => 3600*24*7, },
```

Figure 6.21 : le fichier /usr/lib/cgi-bin/mailgraph.cgi

Nous devons ainsi lier specifie dans le fichier de configuration de Postfix comme suit le chemin vers le fichier de log comme suit :

```
##log
maillog_file = /var/log/maillog
```

Figure 6.22 : le fichier /etc/postfix/main.cf

Les outils de surveillance des logs, tels que Mailgraph, pflogsumm, ou autres scripts de gestion des logs, utiliseront ce chemin pour lire et analyser les logs de messagerie.

On active Mailgraph et on redémarre postfix et apache2 par ces commandes :

```
root@mailserver:/mailgraph-1.14# a2enmod cgi
Enabling module cgi.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@mailserver:/mailgraph-1.14# systemctl restart apache2
root@mailserver:/mailgraph-1.14# nano /etc/apache2/conf-available/serve-cgi-bin.conf
root@mailserver:/mailgraph-1.14# a2enconf serve-cgi-bin
Conf serve-cgi-bin already enabled
root@mailserver:/mailgraph-1.14# systemctl restart apache2
```

Figure 6.22 : lancement de mailgraph

Après la vérification de status de Mailgraph, postfix et apache2 on peut accéder à mailgraph depuis l'interface web en tapant <http://192.168.77.147/cgi-bin/mailgraph.cgi>

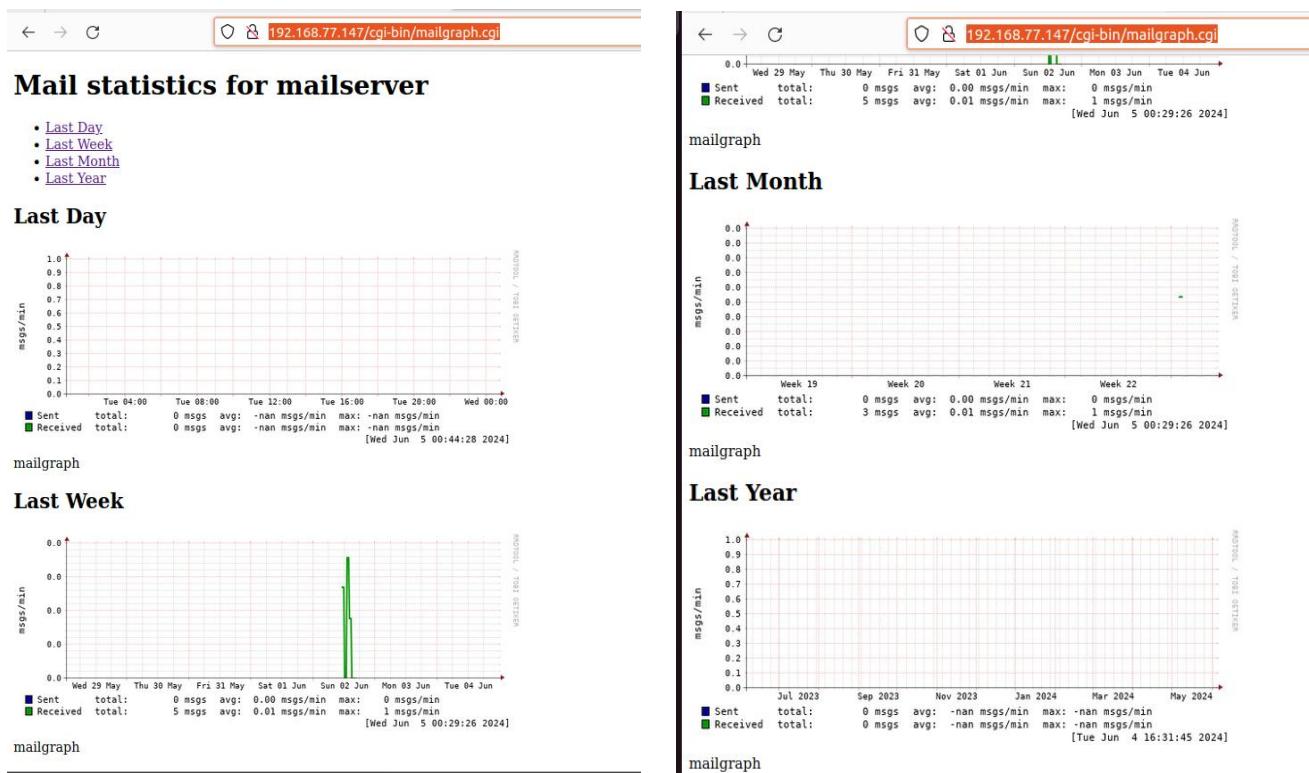


Figure 6.22 : l'interface de Mailgraph

En utilisant Mailgraph, les administrateurs peuvent maintenir une surveillance proactive et continue de l'activité de messagerie sur leur serveur Postfix, ce qui est

crucial pour assurer la performance et la sécurité du service de messagerie.

Conclusion

Rspamd, comme toute solution de sécurité, demande un peu de temps d'administration et d'apprentissage. Sa mise en place n'est pas particulièrement complexe, mais nécessite comme prérequis pour le déploiement d'un cluster, d'avoir les connaissances suffisantes pour l'installation et l'administration d'un cluster Redis. Tous les messages sont délivrés en moins de 10 secondes, en moyenne moins de 2 secondes, il est donc nécessaire de bien verrouiller les envois de messages via le SMTP authentifié, sinon les spameurs bénéficieront d'une plateforme très performante pour envoyer leur spam

Conclusion générale

La mise en place d'un service de messagerie sécurisé et performant en utilisant Postfix et Roundcube peut transformer la manière dont les communications par e-mail sont gérées au sein d'une organisation. En intégrant des technologies telles que SPF, DKIM et DMARC, nous renforçons l'authenticité et la fiabilité des e-mails, réduisant considérablement les risques liés à l'usurpation d'identité et au phishing. L'ajout de solutions robustes de détection de menaces comme VirusTotal, SpamAssassin, ClamAV et Amavis permet de filtrer efficacement les logiciels malveillants et les spams, assurant que les utilisateurs reçoivent uniquement des e-mails sûrs et pertinents.

L'utilisation de techniques de machine learning apporte une dimension évolutive et adaptative au système de messagerie, permettant de détecter et de neutraliser de manière proactive les nouvelles formes de spam et de menaces émergentes. Cette approche dynamique améliore continuellement la précision et l'efficacité des mécanismes de filtrage et de protection.

En combinant ces diverses technologies et pratiques, nous avons créé un environnement de messagerie qui non seulement répond aux exigences actuelles en matière de cybersécurité, mais qui est également capable de s'adapter et d'évoluer face à de nouvelles menaces. Ce système intégré offre une sécurité renforcée, une fiabilité accrue et une meilleure gestion des communications, contribuant ainsi à la protection des données sensibles et à la continuité des opérations au sein de l'organisation. En conclusion, l'implémentation de ce service de messagerie avancé constitue une étape majeure vers la création d'une infrastructure de communication résiliente et sécurisée, essentielle pour toute entreprise moderne.

Bibliographie

[1.1] <https://www.altospam.com/glossaire/phishing/>

[1.2] https://assets.barracuda.com/assets/docs/dms/Barracuda-eBook_13-email-threats_may2020.pdf

[1.3] Cours de deuxième année Protocoles et services de sécurité Mme OUADDAH Aafaf

[3.1] <https://www.1min30.com/dictionnaire-du-web/webmail>

[3.2] <https://www.objetconnecte.com/zimbra-guide-complet/>

[3.3] <https://www.gentlemans-shop.com/tout-savoir-sur-le-service-roundcube/>

[3.4]

http://m.inpt.ac.ma/pluginfile.php/62587/mod_resource/content/1/PFE%20Messagerie%20.pdf

[4.1] : <https://www.linuxbabe.com/mail-server/postfix-amavis-spamassassin-clamav-ubuntu>

[6.1] : <https://www.kaggle.com/>

paper23_article_rev4778_20191008_144207.pdf

efaidnbmnnibpRapport-activite-2022.pdf

<https://www.kaggle.com/datasets/abdallahwagih/spam-emails>

<https://rspamd.com/doc/configuration/>

Annexes

Annexe A

Fichier traind_models.py :

```
****traind_models.py****

import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
import joblib
import matplotlib.pyplot as plt
# Lire les données
df = pd.read_csv("/home/imane/Desktop/phishing-detection/Phishing_Email.csv")
df = df.dropna()

# Balancer les données
Safe_Email = df[df["Email Type"] == "Safe Email"]
Phishing_Email = df[df["Email Type"] == "Phishing Email"]
Safe_Email = Safe_Email.sample(Phishing_Email.shape[0])
Data = pd.concat([Safe_Email, Phishing_Email], ignore_index=True)
# lets check the sahpe again
print("Lets check the shape of each type after undaresampling")
```

```

print(Safe_Email.shape,Phishing_Email.shape)

X = Data["Email Text"].values
y = Data["Email Type"].values
# Count the occurrences of each E-mail type.
email_type_counts = df['Email Type'].value_counts()
#visualisation apres underslmaping
# Create the bar chart
# Create a list of unique email types
unique_email_types = email_type_counts.index.tolist()

# Define a custom color map
color_map = {
    'Phishing Email': 'red',
    'Safe Email': 'green',}

# Map the colors to each email type
colors = [color_map.get(email_type, 'gray') for email_type in unique_email_types]

# Create the bar chart with custom colors
plt.figure(figsize=(8, 6))
plt.bar(unique_email_types, email_type_counts, color=colors)
plt.xlabel('Email Type')
plt.ylabel('Count')
plt.title('Distribution of Email Types with Custom Colors')
plt.xticks(rotation=45)

# Show the chart
plt.tight_layout()
plt.show()
#count
email_type_counts = df['Email Type'].value_counts()
print(email_type_counts)

# lets check the sahpe again
Safe_Email.shape,Phishing_Email.shape

```

```

# Split des données
X_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

# Créer le pipeline
classifier = Pipeline([
    ("tfidf", TfidfVectorizer()),
    ("classifier", RandomForestClassifier(n_estimators=10))
])

# Entrainer le modèle
classifier.fit(X_train, y_train)

# Sauvegarder le pipeline complet
joblib.dump(classifier, '/home/imane/Desktop/phishing-
detection/phishing_detection_pipeline.pkl')

# Faire des prédictions pour évaluer le modèle
y_pred = classifier.predict(x_test)
# Évaluer le modèle
print(classification_report(y_test, y_pred))
print(f"Accuracy: {accuracy_score(y_test, y_pred)}")
print(confusion_matrix(y_test, y_pred))

```

Fichier traind_models.py :

```

#!/usr/bin/env python3
import sys
import joblib
import logging
from email import policy
from email.parser import BytesParser

# Configurer la journalisation
logging.basicConfig(filename='/var/log/phishing_filter.log', level=logging.INFO,
format='%(asctime)s %(levelname)s %(message)s')

```

```

# Charger le pipeline
try:
    pipeline = joblib.load('/home/imane/Desktop/phishing-
detection/phishing_detection_pipeline.pkl')
    logging.info("Pipeline chargé avec succès")
except Exception as e:
    logging.error(f"Erreur lors du chargement du pipeline: {e}")
    sys.exit(1)

# Lire l'email depuis l'entrée standard
try:
    msg = BytesParser(policy=policy.default).parse(sys.stdin.buffer)
    logging.info("Email lu depuis l'entrée standard")
except Exception as e:
    logging.error(f"Erreur lors de la lecture de l'email: {e}")
    sys.exit(1)

# Lire l'email depuis l'entrée standard
try:
    msg = BytesParser(policy=policy.default).parse(sys.stdin.buffer)
    logging.info("Email lu depuis l'entrée standard")
except Exception as e:
    logging.error(f"Erreur lors de la lecture de l'email: {e}")
    sys.exit(1)

# Extraire le corps de l'email
email_body = ""
try:
    if msg.is_multipart():
        for part in msg.iter_parts():
            if part.get_content_type() == 'text/plain':
                charset = part.get_content_charset()
                charset = charset if charset is not None else 'utf-8'
                email_body += part.get_payload(decode=True).decode(charset,
errors='replace')
    else:
        charset = msg.get_content_charset()

```

```

charset = charset if charset is not None else 'utf-8'
email_body = msg.get_payload(decode=True).decode(charset, errors='replace')
logging.info("Corps de l'email extrait")
except Exception as e:
    logging.error(f"Erreur lors de l'extraction du corps de l'email: {e}")
    sys.exit(1)
email_body = ""
try:
    if msg.is_multipart():
        for part in msg.iter_parts():
            if part.get_content_type() == 'text/plain':
                charset = part.get_content_charset()
                charset = charset if charset is not None else 'utf-8'
                email_body += part.get_payload(decode=True).decode(charset,
errors='replace')
            else:
                charset = msg.get_content_charset()
                charset = charset if charset is not None else 'utf-8'
                email_body = msg.get_payload(decode=True).decode(charset, errors='replace')
                logging.info("Corps de l'email extrait")
except Exception as e:
    logging.error(f"Erreur lors de l'extraction du corps de l'email: {e}")
    sys.exit(1)

# Vérifier le contenu du corps de l'email
logging.info(f"Contenu du corps de l'email: {email_body[:100]}...") # Limite à 100
caractères pour le journal

# Prédire si l'email est un phishing
try:
    is_phishing_rfc = pipeline.predict([email_body])[0]
    logging.info(f"Résultat de la prédiction: {'Phishing' if is_phishing_rfc else 'Non-Phishing'}")
except Exception as e:
    logging.error(f"Erreur lors de la prédiction: {e}")
    sys.exit(1)

```

```
# Si le modèle détecte un phishing, ajouter [PHISHING] à l'objet du mail
```

```
try:
```

```
    if is_phishing_rfc == 'Phishing Email':
```

```
        if 'Subject' in msg:
```

```
            msg.replace_header('Subject', '[PHISHING] ' + msg['Subject'])
```

```
        else:
```

```
            msg['Subject'] = '[PHISHING]'
```

```
            logging.info("Sujet de l'email modifié pour indiquer le phishing")
```

```
except Exception as e:
```

```
    logging.error(f"Erreur lors de la modification du sujet de l'email: {e}")
```

```
    sys.exit(1)
```

```
# Sortir l'email modifié
```

```
try:
```

```
    sys.stdout.buffer.write(bytes(msg))
```

```
    logging.info("Email modifié écrit vers la sortie standard")
```

```
except Exception as e:
```

```
    logging.error(f"Erreur lors de l'écriture de l'email modifié: {e}")
```

```
    sys.exit(1)
```

Annexe B.

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer

from sklearn.linear_model import LogisticRegression

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

import pickle

# Load dataset

df = pd.read_csv("combined_data.csv")

df.head()

# Split dataset

X = df["text"].values

y = df["label"].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=87)

# Bag of Words (BoW) vectorizer

count_vectorizer = CountVectorizer()

X_train_bow = count_vectorizer.fit_transform(X_train)

X_test_bow = count_vectorizer.transform(X_test)

# Logistic Regression with BoW

lr_bow = LogisticRegression(max_iter=len(y_train))
```

```

lr_bow.fit(X_train_bow, y_train)

y_pred_bow = lr_bow.predict(X_test_bow)

# Evaluate BoW model

print(f"Accuracy with BoW: {accuracy_score(y_test, y_pred_bow)}")

print(classification_report(y_test, y_pred_bow))

conf_matrix_bow = confusion_matrix(y_test, y_pred_bow)

# Plot confusion matrix for BoW

plt.figure(figsize=(6, 6))

sns.heatmap(conf_matrix_bow, annot=True, cmap='Blues', fmt='g', cbar=False,
            xticklabels=['Not Spam', 'Spam'], yticklabels=['Not Spam', 'Spam'])

plt.title('Confusion Matrix (BoW)')

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.show()

# Save BoW model and vectorizer

with open("lr_bow.pkl", "wb") as model_file:
    pickle.dump(lr_bow, model_file)

with open("count_vectorizer.pkl", "wb") as vectorizer_file:
    pickle.dump(count_vectorizer, vectorizer_file)

# TF-IDF vectorizer

tfidf_vectorizer = TfidfVectorizer()

X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)

```

```

X_test_tfidf = tfidf_vectorizer.transform(X_test)

# Logistic Regression with TF-IDF

lr_tfidf = LogisticRegression(max_iter=len(y_train))

lr_tfidf.fit(X_train_tfidf, y_train)

y_pred_tfidf = lr_tfidf.predict(X_test_tfidf)

# Evaluate TF-IDF model

print(f"Accuracy with TF-IDF: {accuracy_score(y_test, y_pred_tfidf)}")

print(classification_report(y_test, y_pred_tfidf))

conf_matrix_tfidf = confusion_matrix(y_test, y_pred_tfidf)

# Plot confusion matrix for TF-IDF

plt.figure(figsize=(6, 6))

sns.heatmap(conf_matrix_tfidf, annot=True, cmap='Blues', fmt='g', cbar=False,
            xticklabels=['Not Spam', 'Spam'], yticklabels=['Not Spam', 'Spam'])

plt.title('Confusion Matrix (TF-IDF)')

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.show()

# Save TF-IDF model and vectorizer

with open("lr_tfidf.pkl", "wb") as model_file:
    pickle.dump(lr_tfidf, model_file)

with open("tfidf_vectorizer.pkl", "wb") as vectorizer_file:
    pickle.dump(tfidf_vectorizer, vectorizer_file)

```

```
import pandas as pd
import re
import os
import sys
import requests
import json

API_KEY = "e0e89390e92609657d53bb53aef916c9ec4b213e35597b41b675e50bc58c556a"
vt = Virustotal(API_KEY)

# Fonction pour extraire l'URL du texte
def extract_url(text):
    if isinstance(text, str):
        url_pattern = re.compile(r'https?://\S+')
        match = url_pattern.search(text)
        return match.group(0) if match else None
    return None

# Fonction pour enrichir l'URL en utilisant VirusTotal
def enrich_url(url):
    if url:
        try:
            response = vt.request("url/report", params={"resource": url})
            if response.status_code == 200:
                return response.json()
            else:
                return {'error': response.json().get('verbose_msg', 'Unknown error')}
        except Exception as e:
            return {'error': str(e)}
    return {'error': 'No URL found'}

# Fonction pour scanner les fichiers avec VirusTotal
def scan_file(file_path):
```

```

try:
    files = {'file': open(file_path, 'rb')}
    response = vt.request("file/scan", files=files)
    if response.status_code == 200:
        return response.json()
    else:
        return {'error': response.json().get('verbose_msg', 'Unknown error')}
except Exception as e:
    return {'error': str(e)}

def get_file_report(resource):
    try:
        response = vt.request("file/report", params={"resource": resource})
        if response.status_code == 200:
            return response.json()
        else:
            return {'error': response.json().get('verbose_msg', 'Unknown error')}
    except Exception as e:
        return {'error': str(e)}

# Fonction principale
def main():
    attachment_paths = sys.argv[1:]
    for attachment_path in attachment_paths:
        if os.path.isfile(attachment_path):
            print(f"Scanning file: {attachment_path}")
            scan_result = scan_file(attachment_path)
            if 'error' not in scan_result:
                resource = scan_result.get('resource')
                report = get_file_report(resource)
                if report.get('positives', 0) > 0:
                    print("Malware detected!")
                else:
                    print("No malware detected.")
            print(json.dumps(report, indent=4))

```

```

else:
    print(f'Error: {scan_result["error"]}')
else:
    print(f'File not found: {attachment_path}')

if __name__ == "__main__":
    main()

#!/usr/bin/env python3.10
import pymilter
from pymilter import Milter
import pickle
import re

# Charger le modèle et le vectoriseur
with open('model_tfidf.pkl', 'rb') as model_file:
    model_tfidf = pickle.load(model_file)

with open('tfidf_vectorizer.pkl', 'rb') as vectorizer_file:
    tfidf_vectorizer = pickle.load(vectorizer_file)

def contains_url(text):
    url_pattern = re.compile(r'http[s]?://\S+|www\.\S+')
    return re.search(url_pattern, text) is not None

def is_spam(email_text, model, vectorizer):
    if contains_url(email_text):
        print(f"Texte de l'email: {email_text}\nPrédiction: spam (URL détectée)")
        return True

    email_tfidf = vectorizer.transform([email_text])
    prediction = model.predict(email_tfidf)
    print(f"Texte de l'email: {email_text}\nPrédiction: {prediction[0]}")
    return prediction[0] == 'spam'

```

```
class SpamFilterMilter(Milter):
    def __init__(self):
        self.email_content = ""

    def envfrom(self, mailfrom, *str):
        self.email_content = ""
        return Milter.CONTINUE

    def header(self, key, val):
        self.email_content += f'{key}: {val}\n'
        return Milter.CONTINUE

    def body(self, chunk):
        self.email_content += chunk
        return Milter.CONTINUE

    def eom(self):
        if is_spam(self.email_content, model_tfidf, tfidf_vectorizer):
            print("Cet email est classé comme spam")
            return Milter.REJECT
        else:
            print("Cet email est classé comme ham")
            return Milter.ACCEPT

    def run():
        pymilter.runmilter("spamfilter", SpamFilterMilter, "inet:127.0.0.1:12345")

if __name__ == "__main__":
    run();
```