République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab, Blida
USDB.

Faculté des sciences.
Département informatique.

المكتبة

Mémoire pour l'obtention
d'un diplôme d'ingénieur d'état en informatique.
Option : SI.

Sujet :

Conception et réalisation d'un système HoneyPot-Mail.

Présenté par : Belhaoues Chafik.
Belkas Mahmoud.

Promotrice : M^me^ SOUAMI.
Encadreur : Ali Zaiter.
Karim Harmouche.
Wassim Nakadi.

Organisme d'accueil : S.A.R.L.IntraTech.

Soutenue le: date soutenance, devant le jury composé de :

Mlle Boustia, MA.	**Président**
Mr Zahar, MA	**Examinateur**
Mr Ferfera, MA	**Examinateur**

- 2006/2007 -

MIG-004-158-1

Remerciements

Tout d`abord nous remercions الله عز وجل *pour nous avoir guidé vers le*

Bon chemin de la lumière et du savoir, pour nous avoir donné du courage et

De la volonté afin de pouvoir réaliser ce mémoire.

Nous exprimons nos sincères remerciements à nos parents pour tout ce

Qu`ils ont pu nous apporter.

Nous tenons à remercier les membres du jury pour avoir eu l'amabilité

D'accepter et d`apprécier notre travail.

Nous tenons à remercier notre promotrice Mme Souami feriel pour

Son aide, sa patience, sa disponibilité, et sa compréhension.

A tous les enseignants de la faculté des sciences exactes de Blida et

Surtout les enseignants du département de l'informatique.

A tous les employés (directeur, ingénieurs, secrétaires commerciale et comptable)
De

La Société Algéro-libanaise la SARL IntraTech.

Enfin, nous remercions, de tout cœur, tous ceux qui ont contribué de prés ou

De loin à la réalisation de ce travail.

Dédicace

Je dédie ce mémoire

A mes très chers parents pour leur soutien et leur patience durant toute ma carrière d'étude,

Pour leur bienveillance, leurs efforts et encouragement durant mes études,

A mon frère et ma sœur et sa petite famille,

A tous les membres de ma famille proches et éloignés

A mon binôme Chafik et toute sa famille

A mes très chers amis

A mes collègues de travail algériens et libanais

A toute ma promotion,

A toute personne qui me connaît.

Mahmoud Belkas.

Dédicace

Ce travail est dédié :

A ma très chère Maman pour son soutien durant

toute ma carrière, et dans toute ma vie

Pour sa bienveillance, ses efforts constants dans mes études,

Pour son encouragement,

A mon frère REDHA,

A toute la famille ALLALI

(Papa, Houria, Redha japoni, mehdi, Kenza, Radhia)

A mon binôme Mahmoud et à toute sa famille,

A tous mes collègues de travail algériens et libanais

A toute ma promotion,

A tous mes amis que j'aime et qui m'aiment.

Chafik Belhaoues.

TABLE DES MATIERES

Introduction générale 1

CHAPITRE 1 : Les réseaux.

1. Introduction 3
2. Evolution des connections réseaux et internet 3
3. Topologies des réseaux 4
4. Les réseaux étendus 5
5. Intranets et extranets 6
6. La bande passante 7
7. Le standard TCP/IP 7
8. Conclusion 9

CHAPITRE 2 : Sécurité informatique.

1. Introduction 10
2. Evaluation des risques 10
   2.1 Risques humains 11
   2.2 Risques techniques 11
   2.3 Programmes malveillants 11
   2.4 Techniques d'attaque par messagerie 12
   2.5 Attaques sur les mots de passe 12
3. Critères d'évaluation 12
4. Méthodes d'évaluation 13
5. Moyens de sécurisation d'un système 13
   5.1 Conception globale 13
   5.2 Défense en profondeur 14
   5.3 Politique de sécurité 14
   5.4 Plan de continuité d'activité 14
   5.5 Moyens techniques 14
6. Acteurs 15
   6.1 Organismes officiels 15

6.2 Sociétés commerciales 15
6.3 Associations 15
6.4 Groupes de hackers 15
7. Conclusion 15

CHAPITRE 3 : Le Protocole SMTP et Mail.

1. Introduction 16
2. Le Courier Electronique – Mail 16
2.1 Contenu du mail 17
2.2 Acheminement du mail 17
2.3 Système de copie et de copie invisible 18
3. Le protocole SMTP 18
4. Les procédures SMTP 20
4.1 Ouverture et fermeture de connexion 20
4.2 Emission et postage 20
4.3 Emission du courrier 21
4.4 Réémissions (Forwarding) 23
4.5 Vérification de boites et expansion de liste de diffusion 24
4.6 Relai de messages 25
4.7 Echange des roles 26
5. Domaines 26
6. Aspects de sécurité 26
7. Conclusion 26

CHAPITRE 4 : Le Spam « pourriel ».

1. Introduction 27
2. Terminologie et Définitions 27
2.1 Définition du spam 27
2.2 Caractéristiques de spam 28
2.3 Identification du SPAM 29
3. Modèles de SPAM 30
3.1 Le SPAM anglophone 30
3.2 Le SPAM amateur 30
3.3 SPAM « erreur de destination » 30
3.4 Le Spam vicieux 31
3.5 Le spammeur 31
4. Cibles du pourriel (SPAM) 31

4.1 Par courrier électronique 31
4.2 Par message de forum de discussion 32
4.3 Par des fenêtres intruses de Windows 33
4.4 Par systèmes vocaux 33
4.5 Référencement abusif (Spamdexing) 33
4.6 Dans les blogs (splog) 34
4.7 Sur les wikis 35
5. Spam en chiffre 35
6. Profil de Spammeur 36
7. Conclusion 36

CHAPITRE 5 : Outils utilisés.

1. Introduction 37
2. Send Mail 37
2.1 Introduction 37
2.2 Historique 38
2.3 Avantages et inconvénients 38
2.4 Configuration 38
2.5 Le rôle de Send Mail 38
3. Spam assassin 39
3.1 Introduction 39
3.2 Architecture logique de Spam Assassin 39
3.3 Installation 40
3.4 Configuration 40
1. Configuration de base 40
2. Configuration de Razor 41
3. Configuration de Pyzor 41
4. Configuration de DCC 41
5. Configuration des filtres Bayesiens 42
4. Serveur FTP 42
4.1 Mode actif 43
4.2 Mode passif 43
5. Linux Redhat 46
5.1 Notions de noyau et de distribution 47
5.2 Caractéristique du système 47
5.3 Inconvénients 48

CHAPITRE 6 : Etude des HoneyPot.

1. Définition 49
2. Avantages et Inconvénients 50
2.1 Avantages 50
2.2 Inconvénients 50
3. Les types d'HoneyPot 51
3.1 Honeyd: Low-interaction HoneyPot 51
3.2 Honeynet: High-interaction HoneyPot 52
1. Contrôle des données 53
2. Capture des données 53
3. Collection des données 53
4. Génération de HoneyNet 54
3.3 HoneyPot virtuel 56
4. Les HoneyPot virtuels avec Honeyd 57
4.1 Introduction 57
4.2 Fonctionnement 57
1. Fichier de configuration 57
2. HoneyPot client 57
3. Valeurs de HoneyPot 57
5. Conclusion 58

CHAPITRE 7 : Format de message.

1. Introduction 59
2. Composants génériques 59
3. Analyse lexicale 60
3.1 Description générale 60
3.2 Les champs d'en-tête 60
3.3 Définition des champs d'en-tête 61
4. Analyse Syntaxique 61
4.1 Champs spécifiques 61
4.2 Éclaircissements 63
1. Faire suivre (Forwarding) 63
2. Reçu « received » 64
3. Le champ « From » 64
4. Le champ « replay to » 64
5. Le champ « BCC (blind carbon copy) » 65
6. Message-ID 65
5. Le cryptage 65
6. Spécifications de l'adresse 65

6.1 Syntaxe 65
6.2 Sémantique 66
6.3 Les domains 66
6.4 Spécifications d'abréviation de domaine 67
6.5 Les termes du domaine 68
6.6 Boite aux lettres multiples 68
6.7 Spécification explicite du chemin 69
6.8 Adresses réservées 69
7. Conclusion 69

CHAPITRE 8 : conception de B2SAM.

1. Introduction 70
1.1 UML 70
1.2 UP 71
1.2.1 Phase de Réalisation d'UP 72
2. résentation de l'application 72
2.1 La 1ère partie : « piège HoneyPot » 74
1. Serveurs mails virtuels volontairement vulnérables 74
2. Module de récupération d'e-mails 74
2.2 La 2ème partie : « B2SAM centrale» 75
1. Module de lecture et de décomposition 75
2. Module d'analyse 76
1. Analyse de l'adresse IP de l'expéditeur 76
2. Analyse du domaine de l'expéditeur 76
3. Analyse du sujet 77
4. Analyse de la date d'envoie des messages 77
5. Analyse du corps 78
3. Attribution du score 80
4. Module de catégorisation 81
5. Base de données temporaire 81
6. Module de mise-à-jour 81
3. Les cas d'utilisations 82
3.1 Définition des Acteurs 82
3.2 Les diagrammes de cas d'utilisation 82
1. Envoi d'E-mail 82
2. Stockage des E-mail 83
3. Ouverture d'une session ftp 84

4. Affectation dans la base de données 84
5. Analyse des E-mail 85
6. Mise à jour des Anti-spam 86
3.3 Diagramme de cas d'utilisation global 87
3.4 Digramme de transition 88
3.5 Diagramme de séquence et de communication 89
1. L'envoi d'un mail 90
2. Ouverture d'une session ftp 90
3. Module de subdivision 90
4. Module d'analyse et de catégorisation 91
5. Manipulation de RBL 93
4. Diagramme de classe 94
5. Diagramme de déploiement 95
5.1 Définition 95
5.2 Le diagramme de déploiement 95
6. Conclusion 95

CHAPITRE 9 : Implémentation et mise en œuvre.

1. Introduction 96
2. Choix du langage de programmation 96
3. Implémentation 97
3.1 Script d'ouverture d'une session ftp 97
3.2 Module de chargement des e-mails 98
3.3 Module d'analyse entre deux tables de base de données 100
3.4 Module d'analyse d'adresse IP 101
3.5 Module d'analyse du domaine de l'expéditeur 101
3.6 Module d'analyse du sujet de message 101
3.7 Module d'analyse du corps des E-mail 102
4. Réalisation de l'application 102
4.1 Modification de la configuration 103
4.2 Génération du format standard 104
5. Test effectué 104
6. Conclusion 108

Conclusion générale 109

Table des matières.

Annexe 110

Glossaire 114

Bibliographie 123

Liste des Figures.

1. **Fig. I.1** Différentes topologies physiques.....4
2. **Fig. I.2** Raccordement de différentes topologies des réseaux.....5
3. **Fig. I.3** Les réseaux étendus.....6
4. **Fig. I.4** Les intranets et les extranets.....6
5. **Fig. I.5** Modèle TCP /IP.....8
6. **Fig. I.6** Les protocoles du modèle TCP/IP.....9

7. **Fig. III.1** Acheminement du courriel.....17
8. **Fig. III.2** Dialogue client/serveur utilisant le SMTP.....18
9. **Fig. III.3** Scénario du dialogue SMTP.....19
10. **Fig. III.4** Ouverture de session SMTP.....20
11. **Fig. III.5** Fermeture de session SMTP.....20
12. **Fig. III.6** Séquence d'envoi d'un courrier.....23
13. **Fig. III.7** Utilisation des réponses SMTP.....23
14. **Fig. III.8** Vérification de nom d'utilisateur.....24
15. **Fig. III.9** Expansion d'une liste d'utilisateurs.....24
16. **Fig. III.10** Message de notification "undeliverable mail ".....25

17. **Fig. IV.1** Pourcentage de spam.....28
18. **Fig. IV.2** Exemple de spam anglophone.....30
19. **Fig. IV.3** Spam amateur.....30
20. **Fig. IV.4** Spam erreur de destination.....30
21. **Fig. IV.5** Spam vicieux.....31
22. **Fig. IV.6** Spam « ceci n'est pas un spam1 ».....31
23. **Fig. IV.7** Spam « ceci n'est pas un spam2 ».....31
24. **Fig. IV.8** comment évité le pollupostage.....32
25. **Fig. IV.9** Référencement abusif spamdexing.....34

26. **Fig. V.1** Rôle de Send Mail [usrs.cnrs France 2005].....38
27. **Fig. V.2** Architecture logique de SpamAssassin.....39
28. **Fig. V.3** Connexion Data en mode Actif. 44

29. **Fig.VI.1** Le schéma de HoneyNet.....53
30. **Fig.VI.2** 1ère génération de HoneyNet.....54
31. **Fig.VI.3** 2ème génération de HoneyNet.....55
32. **Fig.VI.4** HoneyPot Virtuel.....56
33. **Fig.VI.5** Architecture du HoneyPot Virtuel hybride.....56
34. **Fig.VI.6** Architecture de mode de fonctionnement du HoneyC.....57

35. **Fig.VII.1** Champ d'en-tête « TO » model 1.....60
36. **Fig.VII.2** Champ d'en-tête « TO » model 2.....60
37. **Fig.VII.3** Champ d'en-tête « TO » model 3.....60
38. **Fig.VII.4** Champ d'en-tête « TO » model 4.....61

39. **Fig. VIII.1** Model de cycle de vie d'un logiciel.....71
40. **Fig. VIII.2** Architecture du logiciel B2SAM.....73
41. **Fig. VIII.3** Cas d'utilisation d'envoi d'E-mail.....83
42. **Fig. VIII.4** Cas d'utilisation de stockage des E-mail dans un Répertoire.....83
43. **Fig. VIII.5** Cas d'utilisation d'ouverture de session FTP et Transfert d'E-mail.....84
44. **Fig. VIII.6** Cas d'utilisation d'affectation dans la base de données.....85
45. **Fig. VIII.7** Cas d'utilisation d'analyse des E-mail.....86
46. **Fig. VIII.8** cas d'utilisation de mise à jour des anti-spam.....86
47. **Fig. VIII.9** Diagramme de cas d'utilisation global.....87
48. **Fig. VIII.10** Diagramme de transition d'envoi d'E-mail.....88
49. **Fig. VIII.11** Diagramme de transition d'ouverture de session.....88
50. **Fig. VIII.12** Digramme de transition d'analyse d'E-mail.....89
51. **Fig. VIII.13** Diagramme de séquence d'envoi d'E-mail.....90
52. **Fig. VIII.14** Diagramme de séquence d'ouverture d'une session FTP.....90
53. **Fig. VIII.15** Diagramme de séquence de subdivision.....91
54. **Fig. VIII.16** Diagramme de séquence D'analyse d'E-mail.....92
55. **Fig. VIII.17** Diagramme de séquence d'envoi d'E-mail.....93
56. **Fig. VIII.18** Diagramme de Classe.....94
57. **Fig. VIII.19** Diagramme de déploiement.....95

58. **Fig. IX.1** Commandes d'automatisation de la tache.....97
59. **Fig. IX.2** Script d'ouverture d'une session ftp.....98
60. **Fig. IX.3** Méthode de vérification de mot de passe.....98
61. **Fig. IX.4** Code de vérification de chemin.....99
62. **Fig. IX.5** Code de chargement d'e-mails.....99
63. **Fig. IX.6** Code de découpage des e-mails.....100
64. **Fig. IX.7** Code de l'analyse entre deux tables.....100
65. **Fig. IX.8** Code de l'analyse de l'adresse IP.....101
66. **Fig. IX.9** Code de l'analyse du domaine.....101
67. **Fig. IX.10** Code de l'analyse du sujet.....101
68. **Fig. IX.11** Code de l'analyse du corps du message.....102
69. **Fig. IX.12** Interface principale de l'application.....102
70. **Fig. IX.13** Fiche de modification de la configuration.....103

71. **Fig. IX.14** Fichier de format standard généré.....104
72. **Fig. IX.15** Environnement de test du HoneyPot Mail.....105
73. **Fig. IX.16** Aperçus des messages reçus de différents domaines.....106
74. **Fig. IX.17** Table d'analyse dans un environnement de test.....107
75. **Fig. IX.18** L'invite de mot de passe.....107
76. **Fig. IX.19** Mot de passe invalide.....108
77. **Fig. A.1** Les couches du model OSI..... 111

Liste des Tables.

1. **Table IV.1** Monde : Origines du spam.....35
2. **Table IV.2** Répartition du spam par catégories en 2007.....35
3. **Table IV.3** Etats-Unis : nombre de spams quotidiens selon une étude menée en novembre 2006.....36

4. **Table VI.1** Caractéristiques des deux types d'HoneyPot.....50

5. **Table VII.1** Les Champs d'en-tête.....61
6. **Table VII.2** Les Marqueurs lexicaux. 62
7. **Table VII.3** spécification syntaxique de l'adresse. 65

8. **Table VIII.1** pourcentage des Spam en fonction de la langue utilisée.....79
9. **Table VIII.2** Définition des acteurs.....82
10. **Table VIII.3** Rôle des acteurs d'envoi d'E-mail.....82
11. **Table VIII.4** Rôle des acteurs de stockage des E-mail.....83
12. **Table VIII.5** Rôle des acteurs d'ouverture d'une session FTP.....84
13. **Table VIII.6** Rôle des acteurs d'affectation dans la base de données.....84
14. **Table VIII.7** Rôle des acteurs d'analyse des E-mail.....85
15. **Table VIII.8** Rôle des acteurs de mise à jour de l'Anti-spams.....86

Introduction générale.

The open doors are always welcome sign.

– Belhaoues Chafik.

Le courrier électronique est devenu, avec les années, un service essentiel presque pour tout le monde. Qui ne possède pas une adresse de courrier aujourd'hui?

Avec le temps, il semblait évident que de nombreuses menaces verraient le jour et qu'elles se propageraient à travers ce médium de communication. Certaines personnes, toujours prêtes à faire de l'argent, ont vu, dans le courrier, une excellente façon de rejoindre des clients potentiels. La sollicitation par courrier électronique (*spam*) était née. De plus, les auteurs de virus ont profité de ce vecteur d'attaque en l'utilisant comme tremplin pour une meilleure propagation virale. Les fraudeurs ont aussi gagné leur part du lot, surtout depuis les récentes menaces d'hammeçonage.

Dans le but de se prémunir contre de telles attaques, les administrateurs de systèmes implantèrent diverses technologies afin de protéger leurs utilisateurs contre ce type de menaces. Mais est-ce que ces programmes sont aussi sécurisés que les administrateurs le souhaiteraient? Ces bouts de codes sont-ils prêts à affronter tout le contenu malsain retrouvé aujourd'hui sur l'autoroute électronique?

La recherche qui a été menée dans le cadre de ce mémoire vise à étudier théoriquement et pratiquement un système dont l'objectif est de piéger les spammeurs, pour les dévoiler et les maitriser. Il vous sera présenté le problème lié à l'implantation de technologies d'analyse et de filtrage au niveau du courrier, ainsi que des solutions de mise à jour pour les anti-spams.

Motivations et objectifs :

Notre travail a précisément comme objectif d'étudier l'intérêt d'utiliser la technologie du HoneyPot orientée mail c.-à-d. avoir comme domaine d'étude le courrier électronique afin de proposer des solutions pour mieux sécuriser un tel système. Les objectifs que nous nous somme fixés, afin de réaliser ce travail, sont les suivants :

1. Etudier les différentes stratégies utilisées par les spammeurs.
2. Mettre en œuvre un système HoneyPot-mail basé sur des serveurs mail volontairement vulnérables afin de s'en servir comme des centres de collecte.
3. Proposer une architecture d'un système de collecte, d'analyse et de mise à jour pour les solutions anti-spams dans des plateformes hétérogènes.
4. Conception d'un système d'analyse appelé **B2SAM** pour la détection d'éventuel spam et générer des formats standards pour les solutions de mise à jour des anti-spams en utilisant UML comme approche, UP comme modèle et C++ comme langage de programmation.
5. Mise en œuvre du Système **B2SAM.**

Méthodologie suivie :

Afin d'atteindre nos objectifs, nous avons commencé par un état de l'art des notions de base sur les réseaux et la sécurité des réseaux et systèmes et les différents types d'attaques, puis nous avons détaillé le courrier électronique ainsi que le protocole SMTP, ensuite on a présenté une étude approfondie sur les SPAM afin de bien les authentifier par la suite, et les différents outils utilisés dans notre solution HoneyPot Mail, puis une étude sur les HoneyPot déjà implémenté, on a aussi spécifié le format standard du message pour se familiariser avec les messages afin de les analyser par le B2SAM suivis par une conception détaillée et une mise en œuvre en utilisant le langage C++. Finalement nous avons fait des tests pour prouver la fiabilité de notre solution.

Présentation du mémoire :

La suite du mémoire sera constituée de neufs chapitres.

Chapitre 1 : dédie aux réseaux informatiques et leurs structures.
Chapitre 2 : dans ce chapitre on met en évidence la notion de sécurité réseaux et son impacte.
Chapitre 3 : est consacré au protocole SMTP et aux différentes formes du mail.
Chapitre 4 : on a expliqué le spam et ces différentes manifestations.
Chapitre 5 : nous présentons dans ce chapitre tous les outils utilisés.
Chapitre 6 : ce chapitre décrit les différents systèmes HoneyPot.
Chapitre 7 : est dédie aux spécifications du format de message.
Chapitre 8 : nous présentons ici la conception de notre système HoneyPot mail.
Finalement le Chapitre 9 montre, en utilisant le langage de programmation C++, la possibilité de mettre en œuvre l'architecture proposée.

CHAPITRE I.

Les réseaux.

The future is the Network.

– Jonathan Schwartz.

1. Introduction

Internet est un réseau constitué d'une multitude de réseaux de plus ou moins grande taille reliés entre eux. Les ordinateurs personnels y constituent les sources et les destinations de l'information.

La connexion à Internet englobe trois éléments: physique, logique et applications. Le premier élément (physique) implique de relier l'ordinateur, par un modem ou une carte réseau, à un réseau. La connexion physique assure le transfert des signaux entre ordinateurs. L'élément logique utilise des normes (appelées protocoles). Un protocole est un ensemble formel de règles et de conventions régissant le mode de communication des unités réseau entre elles. Les protocoles TCP/IP (*Transmission Control Protocol/Internet Protocol*) sont les plus largement utilisés sur Internet. Le dernier élément est composé d'applications servant à interpréter et traiter les données (ex : un navigateur Internet, tel qu'Internet Explorer ou Netscape, est nécessaire pour afficher le code HTML sous forme de page Web).

2. Evolution des connections réseaux et Internet

Apparus au début des années 1960, les modems servaient à connecter des terminaux passifs à un ordinateur central à une vitesse de 300 bits par seconde (bits/s). Dans les années 1970, le prix des PC est devenu plus abordable et les systèmes BBS (*Bulletin Board System*) sont apparus. Ces systèmes offraient des services informatisés d'échange d'informations, auxquels les utilisateurs pouvaient se connecter notamment pour afficher des messages et y répondre. La vitesse de 300 bits/s était acceptable puisqu'elle était supérieure à la capacité de lecture à l'écran ou de frappe au clavier des utilisateurs **[ADR-2]**.

Au début des années 1980, l'utilisation des BBS a connu un développement exponentiel et la vitesse de 300 bits/s s'est rapidement avérée insuffisante pour le transfert des fichiers graphiques volumineux. Dans les années 1990, le débit des modems est passé à 9 600 bits/s. En 1998, la vitesse standard actuelle de 56 000 bits/s, ou 56 Kbits/s a été atteinte. Les services à haut débit utilisés dans les environnements d'entreprise, tels que la technologie DSL (*Digital Subscriber Line*) et les connexions par modem câble, sont ensuite devenus accessibles aux particuliers. Ces services ne requièrent plus d'équipement coûteux ni de ligne téléphonique supplémentaire. Toujours actifs, ils offrent un accès instantané à Internet et ne requièrent pas d'établir une nouvelle connexion à chaque session. Ces avancées technologiques, garantes d'une fiabilité et d'une flexibilité accrues, ont simplifié la mise en place de connexions partagées à Internet dans les réseaux domestiques ou de petites entreprises **[ADR-2].**

Les accès à Internet s'opèrent le plus souvent à travers des réseaux locaux (LAN) d'entreprise. Ces réseaux locaux au même titre que les réseaux distants, assurent la connectivité des ordinateurs. La manière dont les ordinateurs sont connectés est déterminée par la topologie réseau choisi.

3. Topologies des réseaux

La topologie réseau est définie aussi bien par la topologie physique que par la topologie logique. La topologie physique est la configuration proprement dite du câblage ou du média. La topologie logique est définie par la manière dont les ordinateurs accèdent aux médias pour envoyer des données.

Les topologies physiques sont les suivantes :

- **Topologie de bus :** un câble sert de backbone unique. Tous les hôtes se connectent directement à ce backbone.
- **Topologie en anneau :** chaque hôte est connecté à son voisin. Le dernier hôte se connecte au premier. Cette topologie crée un anneau physique de câble.
- **Topologie en étoile :** tous les câbles sont raccordés à un point central.
- **Topologie en étoile étendue :** relie des étoiles individuelles en connectant les concentrateurs ou les commutateurs.
- **Topologie hiérarchique :** similaire à une topologie en étoile étendue, elle relie les concentrateurs à un ordinateur qui contrôle le trafic sur la topologie.
- **Topologie maillée :** pour garantir une protection maximale contre l'interruption de service, on connecte tous les ordinateurs entre eux. C'est la topologie la plus coûteuse.

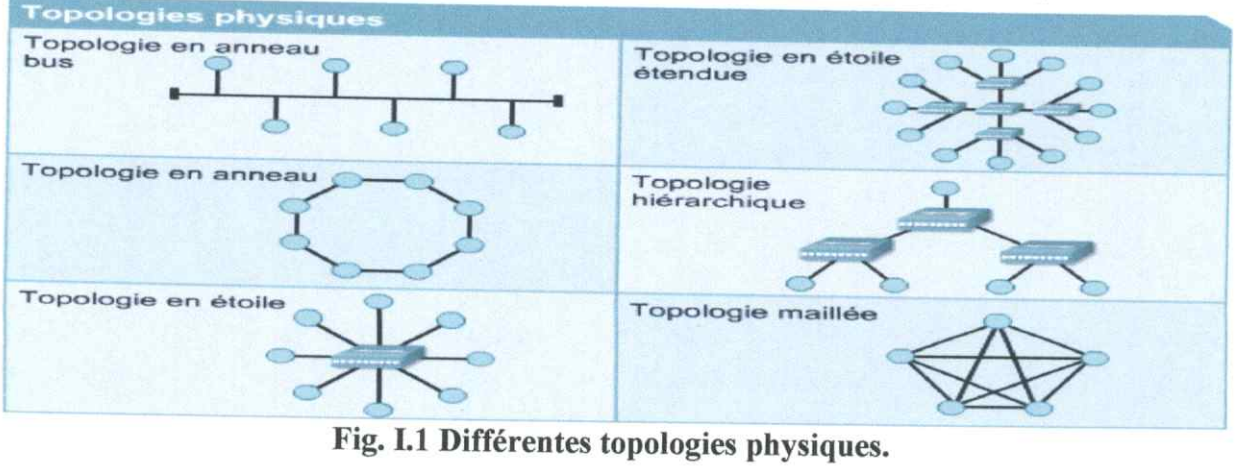


Fig. I.1 Différentes topologies physiques.

Les topologies logiques sont le **broadcast** et le **passage de jeton.** La topologie logique d'un réseau détermine de quelle façon les hôtes communiquent sur le média. L'utilisation d'une topologie de broadcast indique que chaque hôte envoie ses données à tous les autres hôtes sur le média du réseau. Pour la seconde topologie, un jeton électronique est transmis de façon séquentielle à chaque hôte. Dès qu'un hôte reçoit le jeton, cela signifie qu'il peut transmettre des données sur le réseau. Si l'hôte n'a pas de données à transmettre, il passe le jeton à l'hôte suivant et le processus est répété. Token Ring et FDDI (*Fiber Distributed Data Interface*) sont deux exemples de réseaux qui utilisent le passage de jeton. Arcnet est une variante de Token Ring et de FDDI. Il s'agit d'un passage de jeton sur une topologie de bus.

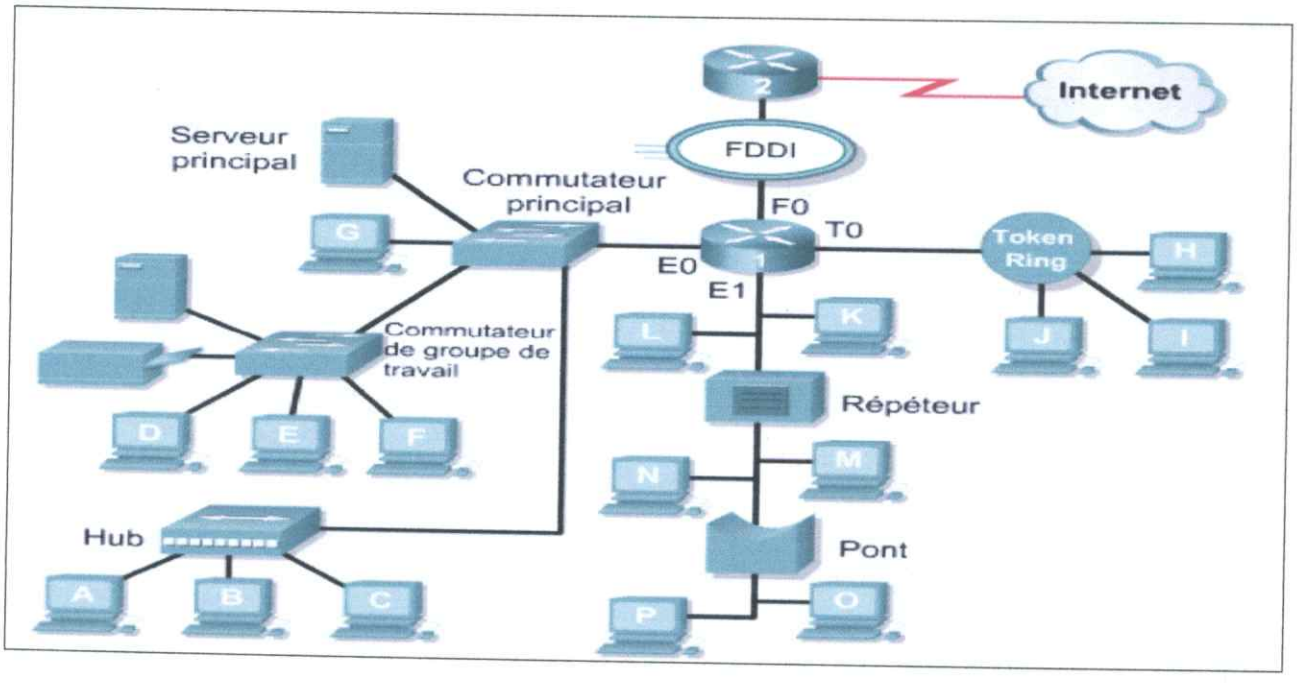


Fig. I.2 Raccordement de différentes topologies des réseaux.

Le schéma de la figure (**I.3**) illustre différentes topologies raccordées. Il représente un réseau de complexité moyenne, comme celui qu'on retrouve habituellement dans une école ou une petite entreprise. C'est un réseau LAN qui est connecté au réseau Internet. Si une entreprise se compose de bureaux distants géographiquement, elle doit pouvoir relier ses ordinateurs au travers de réseaux étendues.

4. Les réseaux étendus

Les réseaux étendus interconnectent des réseaux locaux géographiquement dispersés, ils permettent aux entreprises de communiquer sur de grandes distances. Les réseaux étendus permettent le partage d'ordinateurs, imprimantes et autres équipements raccordés.

Les réseaux étendus utilisent diverses technologies telles que :

- Modems.
- RNIS (*Réseau Numérique à Intégration de Services*).
- DSL (*Digital Subscriber Line*).
- Frame Relay.
- SONET (*Synchronous Optical Network*).

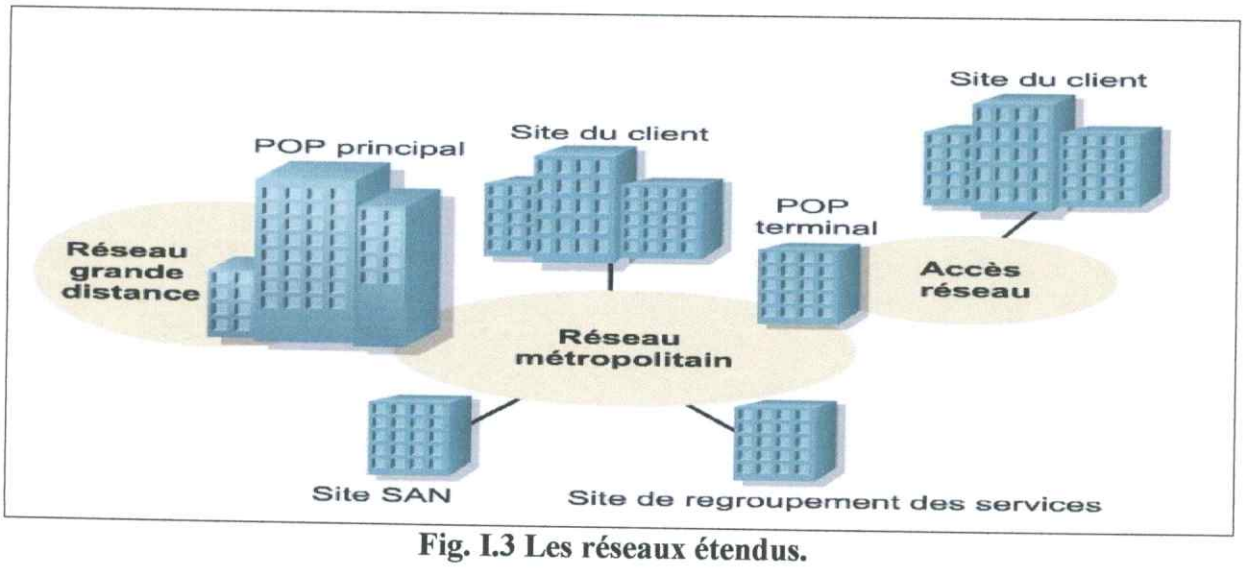


Fig. I.3 Les réseaux étendus.

5. Intranets et Extranets

Un Intranet est une configuration de réseau local conçue pour autoriser les utilisateurs ayant des privilèges d'accès à disposer des ressources internes de l'organisation. Au sein d'un intranet, les serveurs Web sont installés dans le réseau LAN permettant d'utiliser le navigateur comme interface commune d'accès aux informations, telles que les données financières par exemple.

Le terme Extranet fait référence à des applications et à des services qui sont basés sur intranet, mais qui utilisent un accès étendu et sécurisé pour les utilisateurs ou entreprises externes. Un Extranet est une extension de l'Intranet, avec une interaction sécurisée entre les entreprises participantes et leurs intranets respectifs.

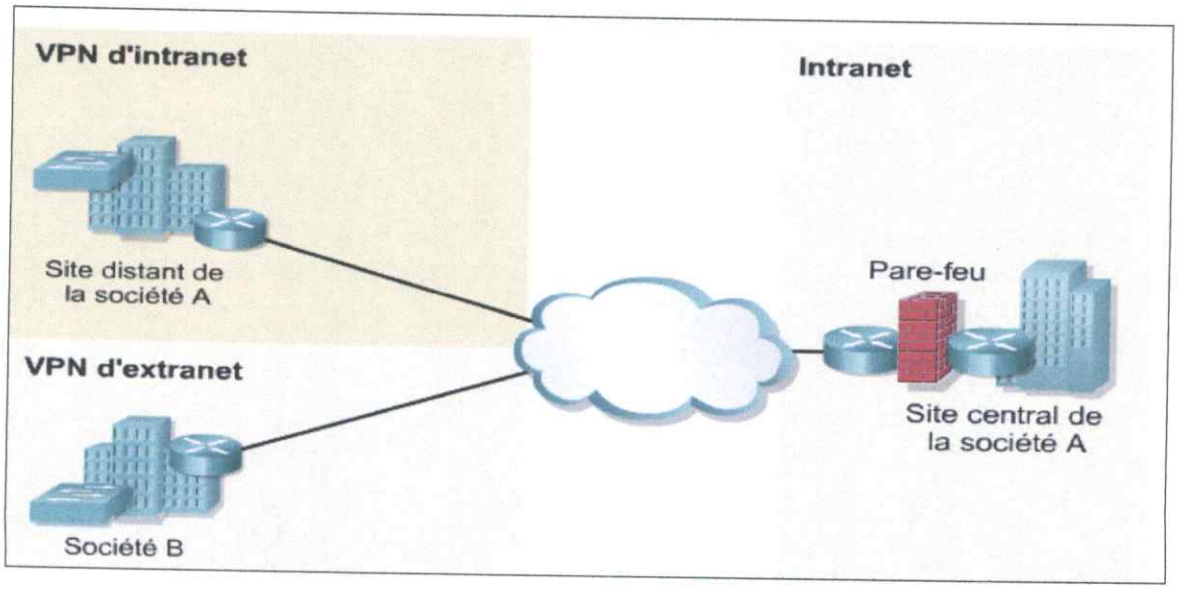


Fig. I.4 Les Intranets et les Extranets.

6. La bande passante

Pour que le partage de ressources dans un réseau (Intranet ou Extranet, ou Internet) soit utilisable, il faut que cet échange n'induise pas des retards et de l'attente. La bande passante représente la quantité d'informations qui peut transiter sur une connexion réseau en un temps donné. La bande passante est donc un élément fondateur pour les réseaux.

- La bande passante est limitée par les technologies utilisées pour placer des informations sur les médias. Par exemple, la bande passante d'un modem conventionnel est limitée à environ 56 kbits/s, à la fois par les propriétés physiques des fils téléphoniques à paires torsadées et par la technologie du modem.
- La bande passante n'est pas gratuite. Pour les connexions WAN, il est généralement nécessaire d'acquérir de la bande passante auprès d'un fournisseur de services.
- La demande en bande passante ne cesse de croître. La mise sur un réseau de contenus multimédias tels que la vidéo et l'audio en continu nécessite une quantité considérable de bande passante. Les systèmes de téléphonie IP augmentent encore le besoin en bande passante.

La disponibilité de diverses technologies assure la possibilité de connecter des ordinateurs et autres organes de transmission entre eux. Cependant, le mode de communication est tributaire des protocoles de communication et de transfert des données. Il existe deux grandes approches prédominantes aujourd'hui dans ce domaine : OSI et TCP/IP. La seconde est celle qui a été largement adoptée par le réseau des réseaux à ce jour. Cet ensemble de protocoles assure que, quel que soit le média physique, quelle que soit la structure des machines, quel que soit le système d'exploitation considéré, et quel que soit les applications de communication déployées : la communication entre machines hétérogènes est garantie.

7. Le standard TCP/IP

C'est à la demande du ministère de la Défense américain DoD qu'a été créé TCP/IP. Le Dod désirait disposer d'un réseau pouvant résister à tout, même à une guerre nucléaire. Dans un monde connecté par différents types de médias de communication tels que les fils de cuivre, micro-ondes, fibres optiques et liaisons satellite, le DoD souhaitait une transmission de paquets capable d'aboutir à coup sûr et sous n'importe quelle condition.

Ce problème de conception conduit à la création du modèle TCP/IP en tant que norme ouverte. Cela voulait dire que n'importe qui pouvait l'utiliser, ce qui contribua à accélérer le développement de TCP/IP en tant que référent.

Le modèle TCP/IP comporte les quatre couches suivantes:

- La couche application.
- La couche transport.
- La couche Internet.
- La couche d'accès au réseau.

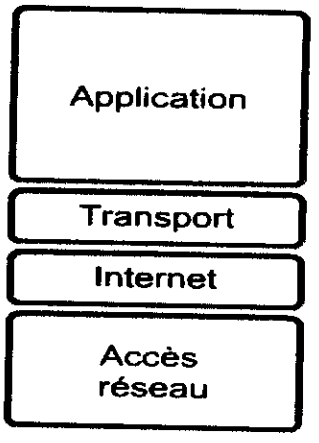


Fig. I.5 Modèle TCP /IP.

La couche application gère les questions de représentation, le code et le contrôle du dialogue.

La couche transport est chargée des questions de qualité de service touchant la fiabilité, le contrôle de flux et la correction des erreurs. L'un de ses protocoles, TCP (*Transmission Control Protocol - protocole de contrôle de transmission*), permet de créer des communications réseau fiables et présentant un taux d'erreurs peu élevé. En effet, le protocole TCP, orienté connexion[1], maintient un dialogue entre l'ordinateur source et l'ordinateur de destination pendant qu'il prépare les informations de couche application en unités, appelées segments.

La couche Internet assure l'envoi et la réception des segments TCP, de et vers n'importe quel réseau. Le protocole IP (*Internet Protocol*) gère les paquets qui arrivent ainsi au réseau de destination indépendamment du chemin qu'ils ont emprunté pour y parvenir.

La relation entre ces deux protocoles est IP et TCP est essentielle : IP pointe le chemin pour les paquets, tandis que TCP assure un transport fiable.

La couche d'accès au réseau, ou encore couche hôte-réseau concerne tous les composants, à la fois physiques et logiques, qui sont nécessaires pour créer une liaison physique. Elle comprend les technologies de réseau physique et les liaisons de données du modèle OSI.

Les protocoles de couche application les plus courants sont **[TINA-3]**:

- FTP (*File Transfer Protocol*).
- HTTP (*HyperText Transfer Protocol*).
- SMTP (*Simple Mail Transfer Protocol*).
- DNS (*Domain Name System*).
- TFTP (*Trivial File Transfer Protocol*).

[1] Un protocole orienté connexion ne signifie pas qu'il existe un circuit entre les ordinateurs en communication. Ce type de fonctionnement indique qu'il y a un échange de segments de couche *application* entre les deux ordinateurs hôtes afin de confirmer l'existence logique de la connexion pendant un certain temps.

Les protocoles les plus courants de la couche transport sont notamment:

- TCP (*Transport Control Protocol*).
- UDP (*User Datagram Protocol*).

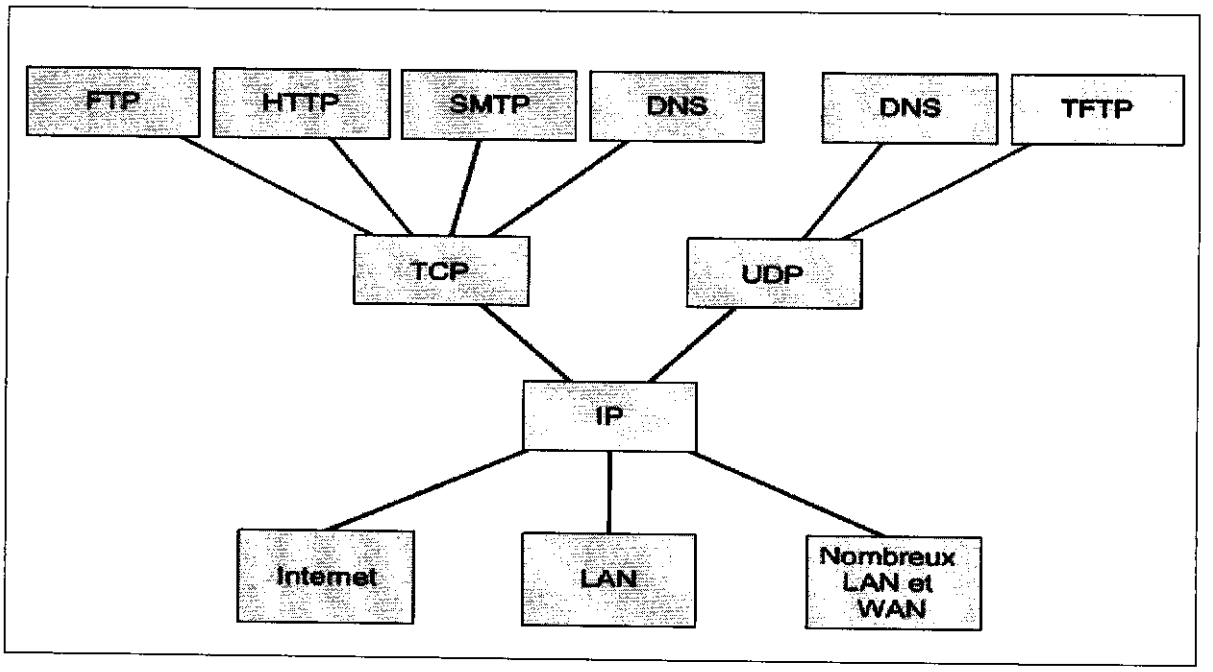
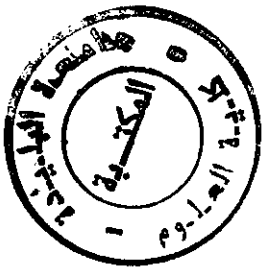


Fig. I.6 Les protocoles du modèle TCP/IP.



8. Conclusion

Comme toutes industries, les systèmes informatiques doivent êtres reliés entre eux d'une façon efficace toute en respectant certains topologies et protocoles pour mieux garder l'aspect objectif de communications entre les systèmes, et pour cela certains standards on été établis pour fournir de tels mécanismes et sans négliger quelques inconvénients non négligeable comme l'insécurité de ses systèmes.

CHAPITRE II.

Sécurité Informatique.

On the other hand,
we cannot ignore efficiency.

– Jon Bentley.

1. Introduction

La sécurité des systèmes d'information regroupe l'ensemble des moyens techniques et organisationnels nécessaires mis en place pour conserver et garantir la sécurité de l'information et du système d'information dans son ensemble.

Les systèmes d'information comprennent tous types de données. Ils sont de ce fait sensible à diverses menaces susceptibles d'altérer ou de détruire l'information (risque pour *l'intégrité de l'information*), ou de la révéler à des tiers qui ne doivent pas en avoir connaissance (risque pour la *confidentialité de l'information*), ou bien de porter atteinte à sa disponibilité (risque pour la *disponibilité du système*).

2. Évaluation des risques

Il est important de mesurer les risques, non seulement en fonction de la probabilité ou de la fréquence de leurs survenances, mais aussi en mesurant leurs effets possibles. En effet, ces effets, selon les circonstances et le moment où ils se manifestent, peuvent avoir des conséquences non négligeables ou catastrophiques :

- données irrémédiablement perdues ou altérées, ce qui les rend inexploitables.
- données ou traitements durablement indisponibles, pouvant entraîner l'arrêt d'une production ou d'un service.

- divulgation d'informations confidentielles ou erronées pouvant profiter à des sociétés concurrentes ou nuire à l'image de l'entreprise.

A l'ère de la généralisation des traitements et des échanges en masse, on imagine assez bien l'impact que pourraient avoir des événements majeurs comme, par exemple, une panne de serveurs de grande ampleur ou la saturation du réseau Internet pendant plusieurs heures.

2.1 Risques humains

Les risques humains sont importants, même s'ils sont le plus souvent ignorés ou minimisés :

- **La malveillance** : Certains utilisateurs peuvent mettre en péril le système d'information en y introduisant des virus ou de mauvaises informations. De même il est relativement aisé d'ajouter des fonctions cachées permettant de détourner à son profit de l'information.
- **L'ingénierie sociale** : (*social engineering*) est une méthode pour obtenir d'une personne des informations confidentielles. Elle consiste à se faire passer pour quelqu'un que l'on n'est pas (en général un administrateur) et de demander des informations personnelles (nom de connexion, mot de passe, etc.) en inventant un quelconque prétexte (problème dans le réseau, heure tardive, … etc.). Elle peut se faire par simple communication téléphonique, ou par mail.
- **L'espionnage** : l'espionnage, emploie les même moyens, ainsi que bien d'autres, pour obtenir des informations sur des activités concurrentes, procédés de fabrication, projets en cours, futurs produits, politique de prix, clients et prospects, etc. **[WPS-5]**.

2.2 Risques techniques

Les risques techniques sont tout simplement ceux liés aux défauts et pannes inévitables que connaissent tous les systèmes matériels et logiciels **[SIN -6]** :

- **Incidents liés au matériel** : la plupart des composants électroniques peuvent comporter des défauts et bien entendu finissent un jour ou l'autre par tomber en panne. Certaines de ces pannes sont assez difficiles à déceler car sont intermittentes ou rares.
- **Incidents liés au logiciel** : la complexité croissante des systèmes d'exploitation et des programmes génèrent inévitablement des erreurs que les meilleures méthodes de travail et les meilleurs outils de contrôle ou de test ne peuvent pas éliminer en totalité.

2.3 Programmes malveillants

Un logiciel malveillant (*malware*) est un logiciel développé dans le but de nuire à un système informatique. Les principaux types de programmes malveillants sont :

- **Le virus** : programme se dupliquant sur d'autres ordinateurs.
- **Le ver** : (*worm*) : exploite les ressources d'un ordinateur afin d'assurer sa reproduction.
- **Le wabbit** : programme qui se réplique par lui-même (n'est ni un virus, ni un ver).
- **Le cheval de Troie** (*trojan*) : programme à apparence légitime (voulue) qui exécute des routines nuisibles sans l'autorisation de l'utilisateur.
- **La porte dérobée** (*backdoor*) : ouvreur d'un accès frauduleux sur un système informatique, à distance.
- **Le logiciel espion** (*spyware*) : collecteur d'informations personnelles sur l'ordinateur d'un utilisateur sans son autorisation, et en envoyant celles-ci à un organisme tiers.

- **L'enregistreur de frappe** (*keylogger*) : programme généralement invisible installé sur le poste d'un utilisateur et chargé d'enregistrer à son insu ses frappes clavier.
- **L'exploit** : programme permettant d'exploiter une faille de sécurité d'un logiciel.
- **Le rootkit** : ensemble de logiciels permettant généralement d'obtenir les droits d'administrateur sur une machine, d'installer une porte dérobée, de truquer les informations susceptibles de révéler la compromission, et d'effacer les traces laissées par l'opération dans les journaux système.

2.4 Techniques d'attaque par messagerie

En dehors des nombreux programmes malveillants qui se propagent par la messagerie électronique, il existe aussi des attaques spécifiques :

- **Le pourriel** (*Spam*) : est un courrier électronique non sollicité, la plupart du temps de la publicité. Ils encombrent le réseau, et font perdre du temps à leurs destinataires.
- **L'hameçonnage** (*phishing*) : est un courrier électronique dont l'expéditeur se fait généralement passer pour un tiers (ex : organisme financier) et demande au destinataire de fournir des informations confidentielles.
- **Le canular informatique** (*hoax*) : est un courrier électronique qui incite généralement le destinataire à retransmettre le message à ses contacts sous divers prétextes.

2.5 Attaques sur les mots de passe

Les attaques sur les mots de passe consistent à effectuer de nombreuse tentatives jusqu'à trouver le bon mot de passe. Deux méthodes sont largement utilisées :

- L'attaque par dictionnaire (*dictionary attack*) : le mot de passe testé est pris dans une liste prédéfinie contenant les mots de passe les plus courants et aussi des variantes de ceux-ci à l'envers, avec un chiffre à la fin, *etc.*
- L'attaque par force brute (*brutal attack*) : toutes les possibilités de mot de passe sont testés jusqu'à trouver la bonne solution (par exemple de "aaaaaa" jusqu'à "ZZZZZZ" pour un mot de passe composé strictement de six caractères alphabétiques).

3. Critères d'évaluation

La vulnérabilité peut s'évaluer suivant plusieurs critères :

- **Disponibilité** : l'élément a-t-il besoin d'une disponibilité importante pour ne pas nuire au système d'information ?
- **Intégrité** : l'élément a-t-il besoin d'une intégrité importante pour ne pas nuire au fonctionnement du système d'information ?
- **Confidentialité** : l'élément a-t-il besoin d'une confidentialité importante pour ne pas nuire au système d'information ?
- **Imputabilité (*Preuve*)** : l'élément a-t-il besoin d'une gestion de la preuve importante pour ne pas nuire à l'organisation ?
- **Contrôle d'accès** : l'élément a-t-il besoin d'un contrôle d'accès important pour ne pas nuire au système d'information ?

Une fois les éléments sensibles déterminés, les risques sur chacun de ces éléments peuvent être évalués en fonction des menaces qui pèsent sur les éléments à protéger. Il faut pour cela évaluer **[SRS-7]** :

- l'impact de chaque menace sur chaque élément.
- la probabilité d'occurrence de ces impacts (*potentialité*).

4. Méthodes d'évaluation

Différentes méthodes d'évaluation des risques sur le système d'information existent, parmi elles **[SRS-7]** :

- la méthode Ebios (*Expression des besoins et identification des objectifs de sécurité*), développée par la DCSSI.
- la méthode Feros (*Fiche d'expression rationnelle des objectifs de sécurité des systèmes d'information*), développée par la DCSSI.
- la méthode Méhari (*Méthode harmonisée d'analyse des risques*), développée par le CLUSIF.
- la méthode Marion (*Méthode d'analyse de risques informatiques optimisés par niveau*), développée par le CLUSIF.
- la méthode Melisa (*Méthode d'évaluation de la vulnérabilité résiduelle des systèmes d'information*).

Paradoxalement, la définition d'indicateurs mesurables, et permettant de définir par la suite des objectifs dans le temps, raisonnables à atteindre, s'avère délicate. S'il s'agit d'indicateurs de performances, on peut désigner comme indicateurs les états d'installation d'outils ou de procédures, mais les indicateurs de résultats sont plus complexes à définir et à apprécier, se sont les « *alertes virales* ». **[SRS-7]**

5. Moyens de sécurisation d'un système

5.1 Conception globale

La sécurité d'un système d'information peut être comparée à une chaîne de maillons, plus ou moins résistants. Elle est alors caractérisée par le niveau de sécurité du maillon le plus faible. La sécurité du système d'information est ainsi abordée dans un contexte global par :

- la sensibilisation des utilisateurs aux problématiques de sécurité, ou dans certains cas « prise de conscience » (*awareness*).
- la sécurité des données, liée aux questions d'interopérabilité, et aux besoins de cohérence des données en univers réparti.
- la sécurité des réseaux.
- la sécurité des systèmes d'exploitation.
- la sécurité des télécommunications.
- la sécurité des applications (débordement de tampon), cela passe par exemple par la programmation sécurisée.
- la sécurité physique, soit la sécurité au niveau des infrastructures matérielles « stratégie de reprise ».

Pour certains, la sécurité des données est à la base de la sécurité des systèmes d'information, car tous les systèmes utilisent des données, et les données communes sont souvent très hétérogènes (format, structure, occurrences, ...).

5.2 Défense en profondeur

Tout droit sorti d'une pratique militaire ancienne et toujours d'actualité, le principe de défense en profondeur revient à sécuriser chaque sous-ensemble du système, et s'oppose à la vision d'une sécurisation du système uniquement en périphérie.

5.3 Politique de sécurité

La sécurité des systèmes d'information se résume souvent à garantir les droits d'accès aux données et ressources d'un système, en mettant en place des mécanismes d'authentification et de contrôle. Ces mécanismes permettent d'assurer que les utilisateurs des ressources possèdent uniquement les droits qui leurs ont été octroyés.

La sécurité informatique doit toute fois être étudiée de telle manière à ne pas empêcher les utilisateurs de développer les usages qui leur sont nécessaires, et de faire en sorte qu'ils puissent utiliser le système d'information en toute confiance. C'est la raison pour laquelle il est nécessaire de définir dans un premier temps une politique de sécurité, c'est-à-dire :

- élaborer des règles et des procédures.
- installer des outils techniques dans les différents services de l'organisation.
- définir les actions à entreprendre en cas de détection d'une intrusion.
- sensibiliser les utilisateurs aux problèmes liés à la sécurité.

La politique de sécurité est donc l'ensemble des orientations suivies par une entité en terme de sécurité. À ce titre, elle doit être élaborée au niveau de la direction de l'organisation concernée, car elle concerne tous les utilisateurs du système. Le rôle de l'administrateur informatique est donc de faire en sorte que les ressources informatiques et les droits d'accès à celles-ci soient en cohérence avec la politique de sécurité retenue. De plus, étant donné qu'il est le seul à connaître parfaitement le système, il lui revient de faire remonter les informations concernant la sécurité à sa direction, éventuellement de la conseiller sur les stratégies à mettre en œuvre, ainsi qu'être le point d'entrée concernant la communication aux utilisateurs des problèmes et recommandations en terme de sécurité. **[SRS-7]**

5.4 Plan de continuité d'activité

Puisque aucune entreprise ne peut se passer de l'outil informatique, d'où la nécessité absolue d'un plan de continuité de l'informatique. Ce plan a pour but la reprise des activités après une catastrophe et ce, de la manière la plus efficace possible tout en garantissant la survie de l'entreprise.

5.5 Moyens techniques

Contrôle les accès aux systèmes informatiques par exemple :

- Surveillance du réseau : supervision, sniffer, système de détection d'intrusion.

- Sécurité applicative : séparation des privilèges, audit de code, rétro-ingénierie.
- Emploi de technologies ad-hoc : pare-feu, anti-logiciels malveillants (antivirus, antipourriel, antiespiogiciel, antitrojan).
- Cryptographie, authentification forte, liste de control d'accès, stéganographie, infrastructure à clés publiques ...etc.

6. Acteurs

6.1 Organismes officiels

Il existe des organismes officiels chargés d'assurer des services de prévention des risques et d'assistance aux traitements d'incidents. Ces CERT (*Computer Emergency Response Team*) sont des centres d'alerte et de réaction aux attaques informatiques, destinés aux entreprises et/ou aux administrations, mais dont les informations sont généralement accessibles à tous.

6.2 Sociétés commerciales [SAH-8]

- Éditeurs d'antivirus.
- Éditeurs de pare-feux.
- Éditeurs de firewall applicatif Web.
- Éditeurs de firewall pour les Web Services (SOA, XML, SOAP).
- Éditeurs de solutions d'Authentification forte.
- Éditeurs de solutions de contrôle des périphériques (USB, Wifi, ...etc.).
- Éditeurs de solutions PKI.
- Éditeurs de solutions Anti-Spam.
- Éditeurs de solutions de chiffrement.
- Éditeurs de solutions de corrélation des événements de sécurité.
- Constructeurs d'Appliance de Sécurité pour la détection et prévension d'intrusions.
- Éditeurs de systèmes d'exploitation basés sur la sécurité : OpenBSD, SE Linux.
- SSII : Sociétés de Services en Ingénierie Informatique, spécialisées en sécurité.
- MSSP: Managed Security Services Provider.

6.3 Associations [SAH-8]

- CLUSIF (Club de la Sécurité des Systèmes d'Information Français).
- OSSIR (Observatoire de la Sécurité des Systèmes d'Information et des Réseaux).
- CLUSIS (Association suisse de la sécurité des systèmes d'information).

6.4 Groupes de hackers

Il est à noter que les hackers, spécialisés en intrusions ou tentatives d'intrusions, ne sont pas tous des *pirates*, comme certains se plaisent à le dire. Aussi, les *lamers*, *script kiddies*, *defaceurs*, et *Curious Joe*, ne sont pas des hackers, mais simplement des personnes qui prétendent l'être. Les hackers sont des passionnés d'informatique, experts des systèmes et des réseaux, capables de développer leurs propres outils, et leur motivation n'est pas forcément délictueuse.

7. Conclusion

Malgré l'impacte primordial de la sécurité sur les systemes il reste toujours des failles persistantes exploitées par la communauté des hackers même sur les systèmes hautement sécurisés, ce qui soutient l'hypothèse que la sécurité informatique à 100% n'est jamais atteinte.

CHAPITRE III.

Protocole SMTP et Mail.

The road goes ever on and on.

– Bilbo Baggins.

1. Introduction

L'utilisation du courrier électronique (mail) est devenue un élément du quotidien de l'être humain. Pour offrir une simplicité et une transparence de manipulation du courrier électronique dans le monde de l'Internet, on a besoin de définir tous les procédés que le courrier utilise pour arriver à destination sans altération.

Pour arriver à mettre en œuvre un tel système, plusieurs organisations et établissements ont, depuis l'apparition de l'Internet, élaborés des standards et des protocoles spécifiant tous les détails de ce type de trafic Internet.

2. Le Courier Electronique - Mail

Le courrier électronique a prit forme en 1965 en tant que moyen de communication entre utilisateurs d'ordinateurs à temps partagée (*time-sharing mainframe*), ce qui explique bien que le courrier électronique existait bien avant Internet. Le Q32 du SDC et le CTSS du MIT ont été les premiers systèmes de messagerie électronique **[VCE-9]**

Ces systèmes furent rapidement étendus en réseau, permettant aux utilisateurs de transmettre des messages à travers différents ordinateurs. Le système AUTODIN peut être considéré

comme le premier à autoriser l'échange de courriers entre ordinateurs en 1966, le système SAGE avait des fonctionnalités similaires quelque temps auparavant.

Ray Tomlinson fut le premier à introduire l'utilisation du signe @ pour séparer les noms d'utilisateurs et de leurs machines en 1972. Ses premiers programmes de courrier *sndmsg* et *readmail* ont joué un rôle important dans le développement du courrier électronique dont ARPANET augmenta de manière significative la popularité.

Pour recevoir et consulter des mails, il est indispensable de posséder une adresse électronique et un client de messagerie MUA (*Mail User Agent*), ce dernier pouvant être simplement accessible via un navigateur Web. Afin d'assurer l'interopérabilité, l'acheminement des courriers est régi par plusieurs standards, que ce soit pour son routage, ou pour son contenu.

2.1 Contenu du mail

À l'origine, le courrier est un document qui contient du texte. Les caractères qui peuvent être utilisés étaient d'abord ASCII sur 7 bit, puis des encodages régionaux. Aujourd'hui, certains logiciels supportent également l'UTF-8 (*Unicode Transformation Format-8*), ce qui permet d'augmenter la quantité de caractères utilisables simultanément.

Il est aujourd'hui également possible de mettre en valeur le contenu du texte d'un courrier, grâce aux fonctionnalités HTML, afin d'ajouter un formatage aux documents. Il faut noter que HTML, UTF-8 et les caractères régionaux ne sont pas toujours interopérables, en fonction du logiciel de messagerie utilisé par le destinataire et de sa localisation géographique.

2.2 Acheminement du mail

L'acheminement des courriers est régi par plusieurs standards : SMTP est dédié à l'envoi d'un message, POP et IMAP servent aussi à délivrer les messages manipulés par les serveurs pour leurs lecture.

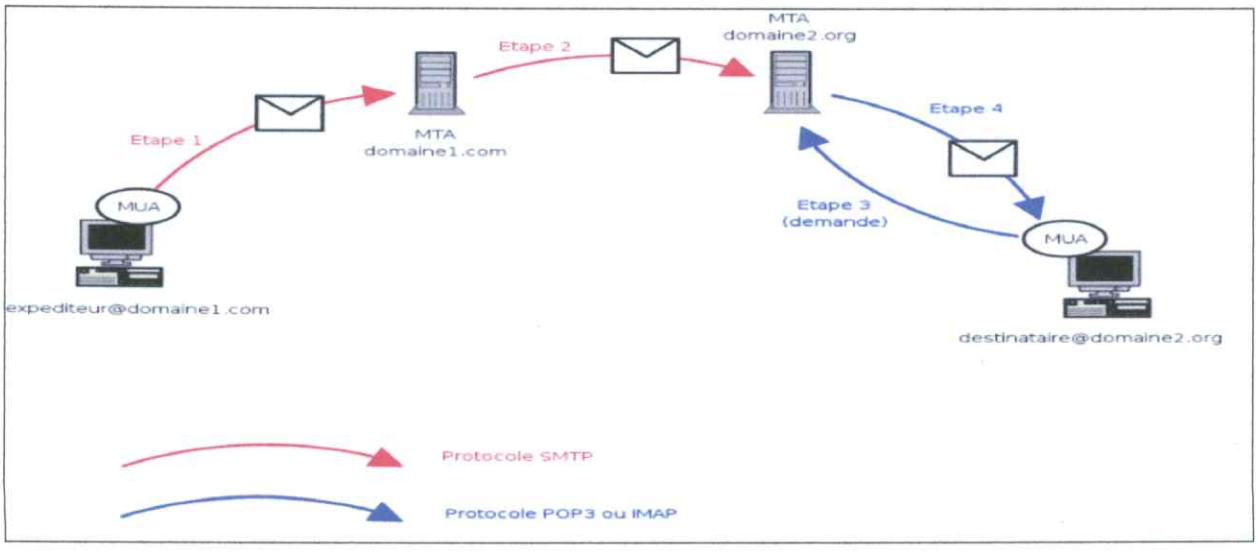


Figure. III.1 Acheminement du mail.

La circulation du courrier suit la procédure suivante :

- Le MUA de l'expéditeur envoie par SMTP le message à un serveur de courrier (celui de son fournisseur d'accès en général) ou MTA, Mail Transfert Agent
- Le premier MTA envoie le message au MTA du destinataire (celui de son domaine *domaine2.org*) qui le stocke en attente de consultation.
- Le destinataire, par l'intermédiaire de son MUA, demande à son serveur de courrier les nouveaux messages.
- Le serveur envoie le message au MUA du destinataire.

2.3 Système de copie et de copie invisible

Le courrier peut être envoyé à plusieurs destinataires :

- les destinataires principaux.
- les destinataires en copie « Cc » (copie *carbone)*
- les destinataires en copie invisible « Cci » (*Copie conforme invisible).*

Pour un envoi vers plusieurs destinataires ne se connaissant pas, il est d'usage de se servir du *Cci.* De plus, les virus et vers informatiques qui exploitent les adresses de courrier trouvées dans les carnets d'adresses des ordinateurs se trouvent bloqués par cette disposition. **[RFC-10]**

3. Le protocole SMTP (Simple Mail Transfer Protocol)

Le Simple Mail Transfer Protocol SMTP, est un protocole de communication utilisé pour transférer le courrier électronique vers les serveurs de messagerie électronique. Largement utilisé au début des années 1980, il est alors un complément à l'**UUCP** (*Unix to Unix Copy Protocol*), celui-ci étant plus adapté pour le transfert de courriers électroniques entre des machines dont l'interconnexion est intermittente. Le SMTP, de son côté, fonctionne mieux lorsque les machines qui envoient et reçoivent les messages sont interconnectées en permanence.

SMTP est un protocole assez simple : il commence par spécifier le ou les destinataires d'un message, puis l'expéditeur du message, puis après avoir vérifié leur existence, le corps du message est transféré. On peut tester un serveur SMTP en utilisant Telnet sur le port 25.

```
220 smtp.xxxx.xxxx SMTP Ready
HELO client
250 Hello client, pleased to meet you
MAIL FROM:<user@xxxx.xxxx>
250 <user@xxxx.xxxx> ... Sender ok
RCPT TO:<user2@yyyy.yyyy>
250 recipient ok.
DATA
354 Enter mail, end with "." on a line by itself
Ceci est un test pour un article de « B2SAM de la thèse de l'ingéniorat »
portant sur SMTP.
.
250 Ok?
QUIT
221 Closing connection
Connection closed by foreign host.
```

Fig. III.2 Dialogue client/serveur utilisant SMTP.

Comme le protocole SMTP utilisait du texte en ASCII (7 bits), il ne fonctionnait pas pour l'envoi de n'importe quels octets dans des fichiers binaires. Pour pallier ce problème, des standards comme **MIME** ont été développés pour permettre le codage des fichiers binaires au travers de SMTP. SMTP ne permet pas de récupérer à distance des courriels arrivés dans une boîte aux lettres sur un serveur. Les standards **POP** et **IMAP** ont été créés dans ce but.

Une fonctionnalité importante de SMTP est sa capacité à relayer des courriers dans des environnements de service de transport. Un service de transport dispose d'un environnement de communication interprocessus (**IPCE**). Un IPCE peut avoir une portée limitée à un seul réseau, étendue à plusieurs réseaux, ou réduite à un sous ensemble d'un réseau. Un processus peut communiquer directement avec un autre processus à travers toute paire d'IPCE se reconnaissant l'une l'autre. Le courrier électronique est une application, ou simplement une utilisation des mécanismes de communication interprocessus.

Le design de SMTP est basé sur le modèle suivant de communication **[WPP-12]:**

1. Suite à une requête de l'utilisateur du courrier, l'émetteur SMTP établit une communication bidirectionnelle vers un récepteur SMTP.
2. Une fois le canal de transmission établi, l'émetteur SMTP envoie une commande **MAIL** mentionnant l'émetteur d'un courrier. Si le récepteur SMTP peut accepter le courrier, il répondra par un message OK. L'émetteur SMTP envoie alors une commande **RCPT** identifiant le destinataire de ce courrier. Si le récepteur SMTP peut accepter un courrier pour ce destinataire, alors il répondra par un message OK ; sinon, il répond par un message refusant le courrier pour ce destinataire (mais n'annulant totalement pas la transaction de courrier).
3. L'émetteur SMTP et le récepteur SMTP pourront mentionner plusieurs destinataires. Une fois cette négociation effectuée, l'émetteur SMTP envoie le contenu du courrier, en le terminant par une séquence spéciale. Si le récepteur SMTP traite avec succès la

réception du contenu du courrier, il répondra par un message OK. Le dialogue est volontairement un dialogue pas à pas, à étapes verrouillées.

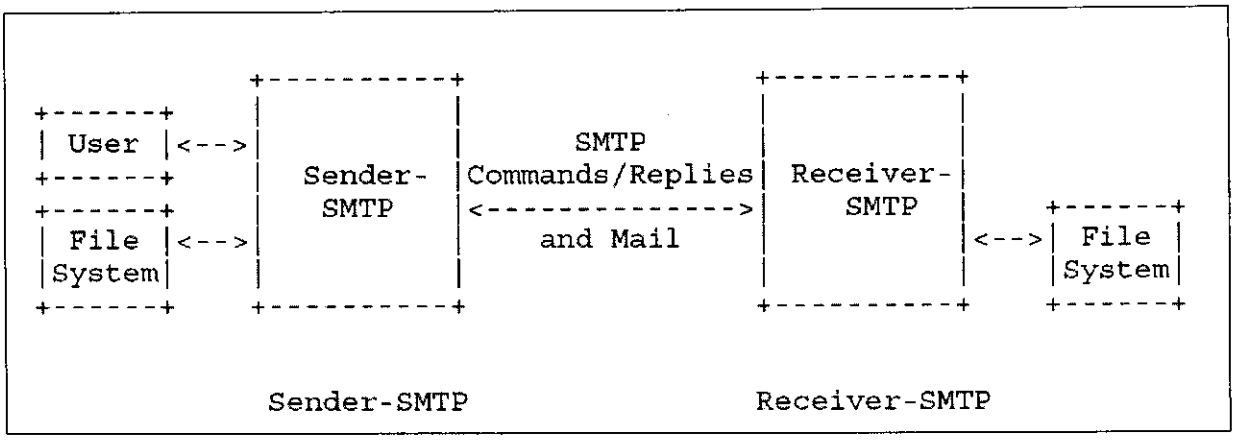


Fig. III.3 Scénario du dialogue SMTP.

L'argument associé à la commande MAIL est le chemin inverse, qui spécifie qui est l'émetteur du courrier. L'argument associé à la commande RCPT est un chemin direct, qui spécifie à qui est destiné le courrier. Le chemin direct sert à l'acheminement, tandis que le chemin inverse est utilisé pour renvoyer un message d'erreur à l'émetteur.
Les commandes et réponses du système de courrier ne tiennent pas compte de la casse et sont composées de caractères ASCII. Lorsque le service de transport utilisé permet la transmission sur une base de 8-bits (octet), tous les caractères 7-bits sont transmis justifiés à droite (dans l'octet de transmission), le huitième bit étant toujours forcé à 0.

Lorsque nous spécifierons la forme syntaxique générale des commandes et des réponses, un argument (ou un symbole "spécial") sera exprimé sous forme d'une variable (ou constante) (*métalinguistique*), par exemple :"<chaîne>" ou "<route-inverse>". Ici, les signes (*inférieur à*) et (*supérieur à*) indiquent le caractère métalinguistique de la variable. Cependant, certains arguments utilisent ces signes dans leur expression littérale. Par exemple, un chemin inverse est actuellement encapsulé par ces signes, c'est-à-dire, "<John.Smith@USC-ISI.ARPA>" est une instance de la variable générale <route-inverse> (les signes (*inférieur à*) et (*supérieur à*) sont effectivement transmis dans la commande ou la réponse). **[RFCf-13].**

4. Les procédures SMTP

4.1 Ouverture et fermeture de connexion

Dès que le canal de transmission est établi, un échange de protocole s'assure que l'hôte demandeur communique bien avec l'hôte attendu (SP est espace et CRLF est retour chariot) :

HELO <SP> <domain> <CRLF>
QUIT <CRLF>

Par la commande HELO, l'hôte émetteur s'identifie ; cette commande peut être assimilée à l'expression "Bonjour, je suis le domaine <domaine>". **[RFCa-14]**

```
R: 220 BBN-UNIX.ARPA Simple Mail Transfer Service Ready
S: HELO USC-ISIF.ARPA
R: 250 BBN-UNIX.ARPA
```

Fig. III.4 Ouverture de session SMTP.

```
S: QUIT
R: 221 BBN-UNIX.ARPA Service closing transmission channel
```

Fig. III.5 Fermeture de session SMTP.

4.2 Emission et postage

Le routage d'un message vers une boîte-aux-lettres d'un utilisateur est appelé (*postage*), l'envoi du même message sur le terminal de l'utilisateur est appelé (*transfert*). Du fait que, dans de nombreux hôtes, l'implémentation du transfert est quasiment identique à celle du postage, ces deux fonctions ont été combinées dans SMTP [RFC821].

Les utilisateurs devront avoir dans tous les cas la possibilité de contrôler l'inscription des messages entrants sur leurs écrans. La plupart des hôtes ont une fonction qui permet de désactiver l'écriture de ces messages.

Exemple1: `SEND <SP> FROM:<reverse-path> <CRLF>`

La commande SEND demande que le message soit affiché sur le terminal de l'utilisateur. Si l'utilisateur n'est pas connecté au système (ou n'accepte pas l'affichage interactif des messages sur son terminal), une réponse de code 450 sera renvoyé en réponse à la commande RCPT. La transaction est considérée comme aboutie lorsque le message est affiché sur le terminal.

Exemple2: `SOML <SP> FROM:<reverse-path> <CRLF>`

La commande « Send **Or** MaiL » demande que le message soit affiché sur le terminal utilisateur si ce dernier est connecté au système (et accepte l'affichage interactif de messages sur ce terminal). Dans le cas contraire, le message doit être redirigé vers sa boîte-aux-lettres. La transaction est considérée comme aboutie si le message est soit affiché sur le terminal, soit déposé dans la boîte-aux-lettres.

Exemple3: `SAML <SP> FROM:<reverse-path> <CRLF>`

La commande « Send **And** MaiL » demande que le message soit affiché sur le terminal utilisateur si ce dernier est connecté au système (et accepte l'affichage interactif de messages sur ce terminal). Dans tous les cas le message doit être copié dans sa boîte-aux-lettres. La transaction est considérée comme aboutie si le message a au moins pu être copié dans la boîte-aux-lettres. Les réponses à ces commandes sont les mêmes que pour la commande MAIL. **[RFCa-14]**

4.3 Emission du courrier

Une transaction SMTP se déroule en trois étapes.

1. La transaction est initiée par une commande MAIL donnant l'identification de l'émetteur.
2. Une série contenant une ou plusieurs commandes RCPT est appliquée, donnant les informations sur les destinataires.
3. Puis une commande DATA passe le contenu du message. Dans cette troisième phase, la marque de fin de données à transmettre marque la fin de la transaction, généralement un point «.» est utilisé pour marquer la fin de la commande DATA.

La première étape de la procédure est donc la commande MAIL. L'argument <route-inverse> contient le nom de la boîte aux lettres de l'émetteur.
Exemple4: `MAIL <SP> FROM:<reverse-path> <CRLF>`

Cette commande indique au récepteur SMTP qu'une nouvelle transaction de courrier débute et lui demande de réinitialiser ses tables d'états et ses tampons, y compris ses boîtes de réception et toute donnée de courrier latente.
Elle lui donne le chemin inverse qui pourra être utilisé en cas de rapport d'erreur à transmettre. Si l'émetteur accepte la commande, un code «250 OK» est renvoyé à l'émetteur.
L'argument <route-inverse> peut contenir plus d'une boîte aux lettres. On peut y inscrire une liste de plusieurs boîtes aux lettres dans des différents hôtes. La première boîte inscrite dans l'argument <route-inverse> doit néanmoins être celle désignant l'émetteur du message.

La deuxième étape de la procédure est l'émission des commandes RCPT.
Exemple5: `RCPT <SP> TO:<cheminDirect> <CRLF>`

Cette commande transmet une adresse de courrier désignant un destinataire. Si elle est acceptée, le récepteur renvoie une réponse de code « 250 OK », et mémorise le chemin d'acheminement. Si le destinataire est inconnu, le récepteur renvoie un code d'erreur « 550 Failure ». Cette seconde étape de la procédure peut être répétée autant de fois si nécessaire.

L'argument <route-directe> peut contenir plus d'une adresse de boîte aux lettres. On peut y inscrire une liste de boîtes aux lettres dans des hôtes différents. Le premier hôte à être mentionné dans l'argument <route-directe> sera l'hôte à qui est envoyé cette commande.

La troisième étape consiste en l'émission de la commande DATA.
Exemple6: `DATA < CRLF>`

Le récepteur SMTP renvoie une réponse de code "354 Intermediate" et prend en compte toutes les lignes de texte suivantes comme étant le texte du message. Lorsque la marque indiquant la fin du texte est reçue et enregistrée, le récepteur SMTP envoie une réponse de code « 250 OK ». Comme les données du courrier sont émises sur le même canal de transmission, il faudra donner un indicateur de fin de données pour que le dialogue requête-réponse puisse reprendre. SMTP indique la fin des données du message en envoyant une ligne contenant un point unique. Une procédure de "transparence" est utilisée pour éviter que celle-

ci n'interfère avec le texte de l'utilisateur. Notez que les données du message incluent un résumé d'en-tête comprenant les champs tels que Date, Subject, To, Cc, From.

L'indicateur de fin de message confirme en outre la transaction de courrier et signale au récepteur-SMTP qu'il peut désormais traiter les destinataires enregistrés pour ce message. Si les données sont acceptées, le récepteur-SMTP renvoie une réponse de code « 250 OK ». La commande DATA ne doit échouer que si la transaction de courrier s'avère incomplète (par exemple, pas de destinataires), ou si les ressources suffisantes pour traiter le courrier ne sont pas disponibles.

L'image suivante illustre l'utilisation de ces commandes dans une transaction SMTP. Elle donne un exemple de séquence SMTP montrant un courrier envoyé par Smith, utilisateur de l'hôte ARPA Alpha, à Jones, Green, et Brown utilisateurs de l'hôte ARPA Beta. Nous supposerons que l'hôte Alpha contacte l'hôte Beta directement.

Le courrier a été accepté pour Jones et Brown. Green n'avait pas de boîte aux lettres sur Beta. **[RFCa-14]**

```
S: MAIL FROM:<Smith@Alpha.ARPA>
R: 250 OK

S: RCPT TO:<Jones@Beta.ARPA>
R: 250 OK

S: RCPT TO:<Green@Beta.ARPA>
R: 550 No such user here

S: RCPT TO:<Brown@Beta.ARPA>
R: 250 OK

S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: Blah blah blah...
S: ...etc. etc. etc.
S: <CRLF>.<CRLF>
R: 250 OK
```

Fig. III.6 Séquence d'envoi d'un courrier.

4.4 Réémissions (*Forwarding*)

Il existe certains cas où l'information concernant un destinataire donnée dans la <route-directe> est incorrecte, mais le récepteur-SMTP connaît la destination exacte. Dans un tel cas, l'une des réponses suivantes pourra être émise pour permettre à l'émetteur de contacter la bonne destination **[RFCf-13].**

```
251 User not local; will forward to <forward-path>
```

Cette réponse indique que le récepteur-SMTP sait que la boîte-aux-lettres du destinataire est hébergée sur un autre hôte et indique le chemin d'accès correct de cette boîte aux lettres pour des messages futurs.

551 User not local; please try <forward-path>
Cette réponse indique que le récepteur-SMTP sait que la boîte-aux-lettres du destinataire est sur un autre hôte et signale le chemin d'accès correct à utiliser pour joindre cette boîte-aux-lettres.

```
S: RCPT TO:<Postel@USC-ISI.ARPA>
R: 251 User not local; will forward to <Postel@USC-ISIF.ARPA>

Or

S: RCPT TO:<Paul@USC-ISIB.ARPA>
R: 551 User not local; please try <Mockapetris@USC-ISIF.ARPA>
```

Fig. III.7 Utilisation des réponses SMTP.

4.5 Vérification de boites et expansion de liste de diffusion

SMTP propose à titre de fonctionnalités additionnelles, des commandes pour vérifier un nom de destinataire ou pour expanser une liste de diffusion. Ces deux opérations peuvent être menées respectivement avec les commandes **VRFY** et **EXPN**. **[SAM-15]**

Pour la commande **VRFY**, l'argument est un nom d'utilisateur, la réponse à cette commande pouvant inclure le nom complet de cet utilisateur et devant fournir une adresse complètement qualifiée de boîte-aux-lettres pour cet utilisateur.
Pour la commande **EXPN,** l'argument identifie une liste de diffusion, la réponse multi lignes à cette commande pouvant inclure le nom complet des utilisateurs et **devant** inclure l'ensemble des boîtes-aux-lettres contenues dans la liste.

```
    S: VRFY Smith
    R: 250 Fred Smith <Smith@USC-ISIF.ARPA>

Or

    S: VRFY Smith
    R: 251 User not local; will forward to <Smith@USC-ISIQ.ARPA>

Or

    S: VRFY Jones
    R: 550 String does not match anything.

Or

    S: VRFY Jones
    R: 551 User not local; please try <Jones@USC-ISIQ.ARPA>

Or

    S: VRFY Gourzenkyinplatz
    R: 553 User ambiguous.
```

Fig. III.8 Vérification de nom d'utilisateur.

```
    S: EXPN Example-People
    R: 250-Jon Postel <Postel@USC-ISIF.ARPA>
    R: 250-Fred Fonebone <Fonebone@USC-ISIQ.ARPA>
    R: 250-Sam Q. Smith <SQSmith@USC-ISIQ.ARPA>
    R: 250-Quincy Smith <@USC-ISIF.ARPA:Q-Smith@ISI-VAXA.ARPA>
    R: 250-<joe@foo-unix.ARPA>
    R: 250 <xyz@bar-unix.ARPA>

Or

    S: EXPN Executive-Washroom-List
    R: 550 Access Denied to You.
```

Fig. III.9 Expansion d'une liste d'utilisateurs.

4.6 Relais de messages

Le chemin d'acheminement des messages peut être une (*route*) de la forme :

"@UN,@DEUX:JOE@TROIS",

dans laquelle « UN », « DEUX », et « TROIS » sont des noms d'hôtes.

Cette forme est employée dans le but d'accentuer la différence formelle entre une adresse et une route. La boîte aux lettres est une adresse absolue, la route est une information permettant d'y accéder.

Dans le concept, les éléments de la route sont déplacés vers la route inverse au fur et à mesure que le message circule à travers les serveurs-SMTP intermédiaires. La route inverse est donc

construite comme la route prise en sens inverse, (c.-à-d., une route partant de la position courante et aboutissant à l'émetteur du message). Lorsque le message arrive sur un serveur-SMTP, et que le premier élément de la route (la première destination intermédiaire) ne correspond pas à la machine courante, cet élément ne doit pas être effacé de la route et est utilisé par le serveur-SMTP comme une nouvelle destination de réémission. Dans tous les cas, le serveur-SMTP courant ajoute son propre identifiant à la route inverse.

Par cette technique des routes, les récepteurs-SMTP reçoivent des messages qui doivent être relayés vers d'autres serveurs-SMTP. Le récepteur-SMTP peut accepter ou refuser de jouer ce rôle de relais de la même manière qu'il accepte ou refuse de traiter les messages de tel ou tel utilisateur local. Le récepteur-SMTP devient à ce moment un émetteur-SMTP, établit un canal de transmission vers le prochain agent SMTP mentionné dans la route, et lui envoie le message.
Si un serveur-SMTP a accepté de relayer un message, puis s'aperçoit que la route est incorrecte ou que le message ne peut pas poursuivre son chemin quelle que soit la raison, alors il devra composer un message (*undeliverable mail*) et le renvoyer à l'origine (dont il connaît la route par la route inverse).
Un message de notification d'un courrier (*undeliverable*) est montré dans la figure suivante. « Cette notification est la réponse à un message émis par JOE sur l'hôte HOSTW et envoyé via l'hôte HOSTX à l'hôte HOSTY avec des instructions pour un relais vers l'hôte HOSTZ. L'exemple montre la transaction entre l'hôte HOSTY et l'hôte HOSTX, qui constitue la première étape de la notification d'erreur ». **[SAM-15]**

```
S: MAIL FROM:<>
R: 250 ok
S: RCPT TO:<@HOSTX.ARPA:JOE@HOSTW.ARPA>
R: 250 ok
S: DATA
R: 354 send the mail data, end with .
S: Date: 23 Oct 81 11:22:33
S: From: SMTP@HOSTY.ARPA
S: To: JOE@HOSTW.ARPA
S: Subject: Mail System Problem
S:
S:    Sorry JOE, your message to SAM@HOSTZ.ARPA lost.
S:    HOSTZ.ARPA said this:
S:      "550 No Such User"
S: .
R: 250 ok
```

Fig. III.10 Message de notification "undeliverable mail".

4.7 Echange des rôles

La commande TURN peut être utilisée pour échanger les rôles des deux programmes en communication via le canal de transmission établi.
Si un programme-A est actuellement l'émetteur-SMTP, envoie la commande TURN et reçoit une réponse OK (250), alors le programme-A devient le récepteur-SMTP. Si un programme-B est actuellement en position de récepteur-SMTP, reçoit une commande TURN à laquelle il répond par une réponse OK (250), alors le programme-B devient l'émetteur-SMTP.

L'émission d'une réponse 502 indique que le récepteur refuse l'échange de rôles.
Notez toutefois que l'implémentation de cette commande est optionnelle. Elle ne devra d'ailleurs pas être utilisée en temps normal lorsque le canal de transmission est établi sous

TCP. Cependant, lorsque le "coût" d'établissement d'un canal de transmission est élevé, cette commande peut s'avérer utile. Par exemple, lorsque le canal de transmission s'appuie sur le réseau commuté public (téléphone), surtout lorsque certains hôtes font tourner une liaison vers un certain nombre d'autres hôtes pour mettre à jour les échanges de messages.

5. Domaines

Les domaines sont un concept récent dans le système de courrier de l'ARPA Internet. Et l'utilisation du système de domaines transforme l'espace d'adressage depuis un espace plat global de chaînes de caractères nommant des hôtes vers une structure hiérarchisée arborescente d'adresses globales. Le nom d'hôte est remplacé par un domaine et un identifiant d'hôte, couple représenté par une séquence d'éléments de domaines sous forme de chaînes de caractères séparés par des points, et dans un ordre conventionnellement établi du plus spécifique au plus général.
Par exemple, "USC-ISIF.ARPA", "vf.eisti.FR", et "PC7.LCS.MIT.ARPA" sont des domaines à part entière. Lorsque des noms de domaines figurent dans des messages SMTP seuls les noms officiels sont autorisés, et l'usage des pseudos ou alias n'est pas permis. **[SAM-15]**

6. Aspects de sécurité

À la base, une des limitations de SMTP vient de l'impossibilité d'authentifier l'expéditeur. Pour ceci, l'extension SMTP-AUTH a été définie. Malheureusement, l'impossibilité d'imposer largement SMTP-AUTH a rendu ce protocole impuissant face au phénomène du spam.

Le Spam est dû à un certain nombre de facteurs dont l'implémentation de MTAs ne respectant pas les standards, et les failles de sécurité dans les systèmes d'exploitations autorisant les spammeurs à contrôler à distance des PC utilisateur pour leur faire envoyer du spam.

Afin de lutter efficacement contre ce phénomène, il existe deux approches : modifier profondément SMTP ou bien lui adjoindre d'autres protocoles pour combler ses lacunes. Modifier SMTP de manière importante ne paraît pas faisable vu l'importance du réseau de serveurs déjà installé.

Une autre approche consiste à créer des systèmes visant à assister les opérations du protocole SMTP. Le ASRG (groupe de recherche anti-spam) de l'IRTF (*Internet Research Task Force*), travaille actuellement sur l'authentification des courriers électroniques dans le but de fournir un système flexible, et plus léger. L'ensemble de ces recherches ont abouti au protocole MARID en 2004 ainsi qu'au protocole Domain Keys Identified Mail en 2006. **[WPP-12].**

7. Conclusion

L'utilisation intensive du courrier électronique a amené avec lui ses lots de problèmes. Le spam en est un important. Pour maintenir ce service de manière efficace, on doit traiter ses Spams par des procédures et des mécanismes ad hoc.

CHAPITRE IV.

LE SPAM « pourriel ».

The sublime and the ridiculous
are often so nearly related that
it is difficult to class them separately.

– Tom Paine.

1. Introduction

Les spammeurs disent que le Spam est juste une forme de liberté d'expression, tandis que les anti-spammeurs disent qu'il prive les gens de cette liberté de la parole, car il maintient les utilisateurs loin des newsgroups et d'autres forums Internet de discussion.

La raison pour laquelle le Spam existe est parce qu'il fonctionne. Les statistiques reportent qu'entre 10 à 20% des messages reçus par les utilisateurs par des inconnus ont obtenus des réponses; ce qui explique bien l'évolution de cette pratique (le spamming) **[GR-17]**.

Ainsi il faut bien noter que la raison pour laquelle le Spam est si répandu est que cette technique ne coûte presque rien ! L'achat de fichiers d'adresses ou de logiciels collecteurs d'e-mails se fait à un coût dérisoire (quelques dizaines d'euros). L'envoi des courriers électroniques ne coûte pas grand chose non plus (il suffit d'avoir un ordinateur et une connexion Internet).

2. Terminologie et Définitions

2.1 Définition du spam

Le « Spam » est un terme adopté par la communauté d'Internet d'après un croquis célèbre (*python de Monty*) pour marquer des envois massif de messages non sollicités sur des groupes de discussion Usenet.

La définition technique standard proposé par « *mail-abuse.org* » et le groupe anti-Spamming au projet de Spamhaus **[DSP-16]** est qu'un message électronique est dit « spam » si :

1. l'identité et le contexte personnel des destinataires sont non pertinents parce que le message est également applicable à beaucoup d'autres destinataires potentiels.
2. le destinataire n'a pas accordé une permission vérifiable, explicite et révocable pour permettre au message d'être envoyé.
3. la transmission et la réception du message semble au destinataire donner un avantage disproportionné à l'expéditeur.

Le mot « Spam » apparaît maintenant dans les dictionnaires et il est décrit comme :

a) Un E-mail non sollicité, souvent à caractère commercial, envoyé aléatoirement aux multiples listes de mailing, aux individus, ou aux newsgroups.
b) Une autre définition employée par les compagnies qui bloquent les sources de Spam est : « Le Spam d'Internet est un ou plusieurs messages non sollicités, envoyés ou signalés en tant qu'élément d'une plus grande collection de messages, et qui ont tous un contenu identique ».

Il est considérer que, à l'heure actuelle, plus de 90% du trafic de courrier électronique est du *spam*. Voici un report du l'organisation spamhaus sur les spam :

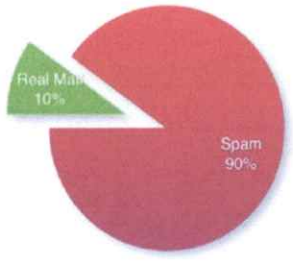


Fig. IV.1 Pourcentage de spam.

2.2 Caractéristiques de Spam

Il y a une frontière très fine entre le marketing légitime d'E-mail et le Spam. Généralement, le Spam a un ou toutes les caractéristiques suivantes :

- L'adresse de retour est invalide : Si vous essayez de répondre à l'E-mail de Spam, vous obtiendrez une erreur de la livraison.
- En-têtes forgés : Cacher l'origine de l'E-mail, conduisant des en-têtes sont habituellement forgés, qui le rend très difficile à tracer des Expéditeurs persistants.
- L'identité du destinataire est non pertinente : parce que l'E-mail est valide pour beaucoup d'autres destinataires.
- Adresse d'attaque de dictionnaire : Si le « To »ligne d'adresse est examiné, vous pouvez voir différentes variantes des adresses de l'E-mail du destinataire. Par exemple, si votre vraie adresse E-mail est JohnSmith@hotmail.com, vous verriez JohnSmith@yahoo.com, JohnSmith@aol.com, JohnnySmith@hotmail.com etc.
- Le sujet n'a pas de relation avec le contenu de l'E-mail.
- L'utilisation des programmes qui génèrent automatiquement et aléatoirement des caractères dans la ligne de « sujet » afin de dévier des filtres de Spam.
- Le contenu d'E-mail est à caractère douteux. Par exemple, certaines des matières couvriraient des arrangements, des perfectionnements de corps, devenir riches ... etc. D'autres sont de fraudes bien connues telles que de diverses formes de fraudes Nigériens d'E-mail.

- La désinscription n'est pas fonctionnelle dans le Spam : Si vous essayez de vous désinscrire du Spam, c'est souvent le cas que le lien ne fonctionne pas, ou ouvrent le site Web d'une publicité.
- Si le Spam contient le HTML, il peut contenir le Java script caché, qui peut ouvrir des sites Web et activer des fenêtres de POPUP de publicité

2.3 Identification du SPAM

Les résultats de l'opération "Boîte à Spam", menée entre juillet et octobre 2002 par la CNIL (*commission nationale de l'information et des libertés*), sont révélateurs. Sur 325 000 Spam signalés par les internautes, 85 % étaient en langue anglaise et 55 % des messages en français étaient à caractère pornographique. Mais ce n'est pas toujours le cas.

On trouve ainsi des « faux positifs » qui sont des messages légitimes classifiés comme des Spam et des « faux négatifs » qui sont des Spam qui franchissent le système de filtrage. Les filtres de Spam doivent permettre plus de faux négatifs que de faux positifs **[SMD-18]**.

Les soucis concernant les faux positifs sont l'une des raisons pour lesquelles l'utilisateur voit tellement le Spam. La crainte de perdre des informations relativement importante oblige les responsables à décider si un E-mail est un spam ou non.

Les fournisseurs de filtre de Spam mentionnent que seulement 0.001% de messages filtrés sont des faux positifs. Cependant, pour une grande compagnie qui reçoit 50.000 E-mails quotidiennement, le petit pourcentage peut faire une différence au résultat de la compagnie. Un de ces E-mails faux positifs a pu être d'importance essentielle pour la compagnie. Un autre scénario intéressant est quand il s'agit de compagnies médicales et pharmaceutiques qui reçoivent un grand nombre d'E-mails contenant des mots qui sont signalés comme des mots suspects, tel que le sexe, Viagra, la drogue etc. Il est essentiel que ces E-mails ne soient pas considérés comme des spam par le procédé de filtration.

Une méthode commune en faisant face aux faux positifs est d'expédier tous les E-mails considéré comme Spam à un dossier spécial par exemple. Les utilisateurs peuvent alors examiner ces E-mails mis en quarantaine, qui seront automatiquement supprimés après une date d'échéance. Une grande compagnie peut devoir stocker beaucoup de giga-octets de ce type d'E-mail, avec le bas coût de dispositifs de stockage, et le coût élevé de perdre un grand contrat dû à un faux positif, le choix de garder les messages filtrés est évident.

3. Modèles de SPAM

3.1 Le SPAM anglophone

Le Spam anglophone vous promet monts et merveille : vous faire gagner de l'argent ou vous procurer du Viagra **[SMD-18].** Voici à quoi cela ressemble :

Objet: Make Easy Money from Home!
Brand New - Ultimate Business Opportunity!
Just Released World Wide!
Want an Income Independent of your job?
Want To Be Your Own Boss and Own Your Own Business?
Want To Easily make $200 - $1,000 per day... paid daily?
Want To Earn Long Term Residual Income for Life?
Want Unlimited Leads to Start Earning Big Immediately!

Fig. IV.2 Spam anglophone.

3.2 Le SPAM amateur

Le responsable est généralement le web master d'un petit site amateur. Le courrier vous donne accès à une partie de la liste des spammés. **[SMD-18]**

Exemple:

From : "Spammer" <spammer@bidule.fr>
To : gp@mygale.org, grn@mygale.org, grnr@mygale.org, gnd@mygale.org, ghnr@mygale.org, gr@mygale.org, glsj@mygale.org, gnsd@mygale.org, gng@mygale.org, gr@mygale.org, gstn@mygale.org, gbrd@mygale.org, grm@mygale.org, gs@mygale.org, (...), vcs@mygale.org, vf@mygale.org, vhnd@mygale.org, vll@mygale.org, vd@mygale.org, vncnt@mygale.org, vm@mygale.org, xcss@mygale.org, xl@mygale.org, xlm@mygale.org, xl@mygale.org, m@mygale.org, mrc@mygale.org, zg@mygale.org, zrt@mygale.org, zrt92@mygale.org, zd@mygale.org, zn@mygale.org, zz@mygale.org, zrl@mygale.org

Subject : Apprendre l'anglais sur INTERNET
Nous vous invitons à essayer nos méthodes d'anglais sur Internet, à l'adresse suivante :
http://www.passerelle.com/anglais

Merci

Fig. IV.3 Spam amateur.

3.3 SPAM « erreur de destination »

La dernière tendance est d'envoyer des Spams en faisant croire qu'il s'agit de courriers qui vous été adressés par erreur. **[SMD-18].**

Objet : Tu vois que je t'oublie pas :!
Coucou c'est moi ! tu te souviens on a dial. y a deux jours sur le chat ! je suis pas sure de l'adresse mais je pense que ça doit être la bonne, donc tu me demandais ou trouver des photos gratos et des webcameras et bien je te file l'adresse c'est >> http://wwwxxxxxxx.com
comme je te l'avais dit donc il y a des photos gratos c'est ça que tu cherchais, des vidéos et aussi le dialogue ou tu fais des rencontres rapide et discrètes, moi j'y suis parfois le soir mais bon, si tu viens préviens moi et je me débrouillerais pour venir, sinon au pire tu peux m'écrire . bon alors a bientôt peut être sur http://wwwxxxxxxx.com
je te fais un gros bisous
Sabrina
nb/ si c'est pas toi, fait pas gaffe a mon message et excuse moi. Bisey !

Fig. IV.4 Spam « erreur de destination ».

3.4. Le Spam vicieux

Il vous propose dès les premières lignes un lien. Bien évidemment, vous pensez que c'est pour ne plus recevoir de messages. Bien au contraire ! C'est pour vous inscrire ! Attention au piège... **[SMD-18].**

> Just one more step! Simply click the link below to activate the subscription request you just sent us !
> http://c.postmasterdirect.com/confirm? E=buzz%40arobase.org&T=4942

Fig. IV.5 spam vicieux.

3.5. Le spammeur : plus hypocrite, tu meurs...

Le spammeur annonce que son courrier n'est pas du spam. **[SMD-18].**

> Ce message n'est pas du SPAM mais simplement une prospection pour vous faire connaitre notre site web http://www.xxxxxxx.com. Vous ne recevrez plus ce message à moins que vous n'en fassiez la demande expresse en cliquant sur le lien suivant pour vous inscrire et continuer à recevoir notre lettre.
>
> Nous avons recopié vos coordonnées "email" sur votre site ou sur un site partenaire. Ceci n'est pas du spam, et vous pouvez enlever vos coordonnées pour ne plus recevoir nos messages en cliquant ici.
>
> si ce message vous a été adresse par erreur, merci de nous le signaler

Fig. IV.6 Spam « ceci n'est pas un spam1 ».

Encore plus fort, une référence à un pseudo décret américain :

> Cet e-mail non sollicité n'est pas un S.P.A.M. parce qu'il est conforme au décret 1618 Titre III du 105e Congrès des États-Unis qui stipule que le e-mail doit contenir l'adresse d'expéditeur et avoir une facilité d'annulation pour l'adressé.

Fig. IV.7 Spam « ceci n'est pas un spam2 ».

4. Les Cibles du SPAM

Le spam peut s'attaquer à divers médias électroniques : les mails, les forums de discussion, les moteurs de recherche, les wikis, les messageries instantanées, les blogs...etc. **[WST-19].**

4.1. Par courrier électronique

Le spam par courrier électronique est le type de pollupostage le plus répandu. Le coût d'envoi d'un courrier électronique étant négligeable, il est facile d'envoyer un message à des millions de destinataires. Les destinataires assument le coût de réception et de stockage en boîte aux lettres, ce qui peut causer des coûts non négligeables aux prestataires de services, à cause du volume pris par le spam.

Contrairement aux promotions commerciales pour lesquelles les utilisateurs peuvent donner leur accord, le pourriel n'est pas sollicité. Il est souvent rédigé spécialement pour contourner les filtres anti-spam. Un mot clé tel que **Viagra** (souvent vanté dans les pourriels) peut être ainsi écrit « **vi@gr@** » ou « **v|agra** » ou « v i a g r a » de manière à tromper un filtrage automatique basé sur ce mot.

Une autre méthode employée consiste à accompagner un texte anodin d'une image sur laquelle se trouve le véritable message publicitaire, l'absence de mot compromettant en dehors de l'image rendant le filtrage de ces messages très compliqué.

Les polluposteurs redoublent d'imagination pour masquer leurs activités et ne pas être démasqués, que ce soit en falsifiant les adresses d'expéditeur ou en utilisant des serveurs SMTP non sécurisés qui permettent des envois anonymes.

Les adresses à pollupostages sont généralement collectées par robot d'indexation. Il existe un marché pour les listes d'adresses. **[WST-19].**

Pour éviter d'être polluposter, les internautes font souvent figurer leurs adresses d'une manière masquée lorsqu'elle doit apparaître dans un site Web. Par exemple :

- Jean@NOSPAM.exemple.fr pour Jean@exemple.fr.
- Jean chez exemple point fr pour Jean@exemple.fr
- Jean [at] exemple.fr pour Jean@exemple.fr (l'arobase se prononçant souvent "at").

Fig. IV.8 comment évité le pollupostage.

4.2. Par message de forum de discussion

Ce type de pollupostage est apparu sur Usenet avant celui par courrier électronique. Les forums de discussion d'Usenet sont une cible facile de pollupostage. En effet, un message envoyé à un forum touche tous les lecteurs du forum. Certains groupes de discussion ne reçoivent pratiquement plus que du pollupostage (c'est l'une des raisons pour lesquelles de nombreux forums sont modérés, c'est-à-dire surveillés par un humain ou un robot qui effectue un tri parmi les articles proposés).

D'autre part, les participants d'Usenet faisant généralement figurer leur adresse électronique dans leurs articles, les polluposteurs peuvent facilement récolter des milliers d'adresses au moyen d'un robot, puis polluposter les auteurs de ces articles par courrier électronique.

Le phénomène est rendu encore plus pénible et par la publication croisée ou la publication multiple, qui consistent respectivement à destiner un message à plusieurs groupes simultanément ou à envoyer le message dans plusieurs groupes de suite.

Dans Usenet, on parle également de pourriel lorsqu'un article, quel que soit son contenu, et même s'il n'appartient pas aux catégories usuelles de messages abusifs (publicités commerciales, escroqueries, insultes…) est publié en un nombre d'exemplaires excessif : tous les exemplaires d'un tel article peuvent être annulés par les utilisateurs (les critères numériques exacts permettant d'identifier de tels pollupostages dans la principale hiérarchie francophone sont donnés dans la documentation du forum). Les diverses hiérarchies possédant des critères différents pour identifier et annuler les articles de pollupostage, il existe une certaine incertitude quant aux traitements qui peuvent être appliqués aux pollupostages diffusés simultanément dans plusieurs de ces hiérarchies.

Une règle particulièrement simple décide du sort des escroqueries manifestes, c'est-à-dire des articles proposant de (gagner de l'argent rapidement et sans rien faire) habituellement appelés «MMF »(*Make money fast)* ces articles peuvent être annulés immédiatement par n'importe quel utilisateur.

A noter que depuis peu, le terme général de spamposting est parfois utilisé pour désigner le pollupostage apparaissant dans les blogs, forums, livre d'or...etc. **[WST-19].**

4.3. Par des fenêtres intruses de Windows

C'est ce que l'on appelle le pollupostage par messagerie instantanée.

4.4. Par systèmes vocaux

Le développement de la voix sur réseau IP (*téléphonie par Internet*) fait craindre la prochaine arrivée sur nos combinés d'un nouveau type de pourriel, le pollupostage vocal, baptisé SpIT (*Spam over Internet Telephony*). En effet, des systèmes comme « Skype » voient déjà une part notable des appels d'origine non sollicitée, même s'il s'agit encore de contacts personnels plutôt que de sollicitations commerciales. **[WST-19].**

4.5. Référencement abusif (Spamdexing)

Optimisation abusive des techniques de référencement destinée aux robots d'indexation de moteur de recherche qui consiste à modifier des pages Web en utilisant des mots-clés d'une façon abusive pour améliorer le classement dans les moteurs de recherche.

Parmi les techniques utilisées :

- La manipulation des mots clés consistant à ajouter une longue liste de mots souvent recherchés (comme « sexe » ou « piratage ») répétitivement dans une page (« sexe à UneVille », « sexe à UneAutreVille », « sexe à EncoreUneAutreVille », « sexe à UnVillageTropPetitPourÊtreUneVille », et toute autre variation possible) pour apparaître immédiatement si on fait une recherche avec ces mots. Parfois une page ne contient que les résultats d'une recherche, mis sur le Web pour être trouvé et classé par les moteurs de recherche et affiché aux usagers cherchant avec les mêmes mots.

- Le bourrage de mots clés populaires, dissimulés au visiteur mais pas au robot, soit en les imprimant blanc sur fond blanc, ou en utilisant la police de caractères la plus petite, ou encore en les utilisant en lignes de commentaires qui ne sont pas affichées à l'usager, ou encore en changeant le contenu de la page après que le logiciel du robot l'ait lu ou en modifiant le serveur pour envoyer une page au moteur de recherche et une autre aux usagers ordinaires.

- Le pollupostage de liens consiste à mettre les liens vers un site qu'on veut promouvoir dans autant d'autres sites externes que possible, incluant les forums publics et les pages de commentaires d'autres sites.

- Une ferme de liens (*link farm*) : est un site hébergeant des listes de liens vers tous les autres sites qu'on contrôle pour améliorer le classement de ces derniers en les faisant

apparaître populaires. Google compte notamment la quantité et l'importance des liens vers un site pour déterminer l'importance du site (*le PageRank*). Parfois on construit aussi les sites multiples (*simulant des sites indépendants et pas simplement des sous domaines du même site*) avec presque le même contenu ; chacun contient un tas de liens vers tous les autres pour améliorer leur classement.

- La technique de Bombardement Google en sa forme originale consiste à placer des hyperliens vers George W. Bush avec des phrases comme « l'idiot du monde » dans autant de sites Web que possible. La destination de cet hyperlien est normalement un site externe (dans cet exemple, la page de Bush). Si ce genre de lien figure dans un nombre assez grand de pages Web, une recherche pour « l'idiot du monde » va diriger l'usager immédiatement sur Bush, peu importe s'il désire ça et peu importe si ces mots figurent sur son site ou pas.

- Une autre variation est le pourriel d'affiliation (*affiliate Spam*) où une compagnie paye pour chaque visiteur ou chaque client envoyé par des liens affichés par des autres, du genre « affiliez-vous et devenez riche, mettez un lien vers www.arnaqueur.porno.example.com et pour chaque victime qui nous donne tous ses numéros de carte de crédit nous vous donnons un sou ».
- Les liens venant de ces programmes d'affiliation contiennent le code d'identification de façon :

www.arnaquer.porno.example.com/donnemoiargent?MonsieurLeSpammeur Pour laisser savoir qui doit être payé pour avoir posté tous ces liens partout

Fig. IV.9 Référencement abusif spamdexing.

Les opérateurs de sites de recherche comme Google cherchent toujours des moyens de détecter ce genre de choses et les rendre plus difficiles à utiliser effectivement ; si on a un tas de liens venant des sites du genre «www.ferme-aux-liens-inutiles.spam.example.com». **[WST-19].**

4.6 Dans les blogs (splog)

La présence de liens vers un site web est un critère important de classement dans les moteurs de recherche. Afin d'augmenter artificiellement le nombre de liens pointant vers leurs sites, certains créent des blogs, ou mettent des messages de commentaires dans des blogs préexistants, uniquement pour ajouter des liens vers un ou plusieurs sites web à promouvoir.

L'automatisation de ce genre de pollution a mené plusieurs logiciels de blog à introduire des contrôles qui rendent cette automatisation par une machine plus complexe à réaliser.

[WST-19].

4.7. Sur les wikis

De nombreux polluposteurs ou publicitaires ajoutent des liens vers leurs sites sur des wikis comme « Wikipédia », sites dont n'importe qui peut modifier les pages. **[WST-19].**

5. Spam en chiffre [SC-20]

Voici des statistiques reportant les différentes manifestations des spam dans le mode entier :

Table IV.1 Origines du Spam
(Entre avril et juin 2006)
Source : Sophos

Rang	Pays	Part des spams
1	Etats-Unis	23,2 %
2	Chine (et Hong Kong)	20,0 %
3	Corée du Sud	7,5 %
4	France	5,2 %
5	Espagne	4,8 %
6	Pologne	3,6 %
7	Brésil	3,1 %
8	Italie	3,0 %
9	Allemagne	2,5 %
10	Royaume-Uni	1,8 %
11	Taiwan	1,7 %
12	Japon	1,6 %
13	Autres pays	22,0 %

Table IV.2 Répartition du spam par catégories en 2007
Source : Clearswift

Types de spam	Avril 2007
Santé	43,86 %
Finance	37,65 %
Produits	9,06 %
Pornograhie	5,32 %
Scam (phishing)	1,41 %
Pari	0,1 %

Table IV.3 Etats-Unis : nombre de spams quotidiens selon une étude menée en novembre 2006

(En pourcentage du nombre des personnes sondées)
Source : Robert H. Smith School of Business, 2006

Nombre de messages	Pourcentage
<1	22 %
1-4	18 %
5-9	16 %
10-19	13 %
20-39	12 %
40 et plus	11 %
Ne sait pas	8 %
Moyenne	18,5 spams
Autres	2,32 %

6. Profil de Spammeur

Les noms et les adresses des spammeurs les plus prolifiques sont facilement disponibles sur Internet. Quelques sites Web anti-Spam entrent dans le grand détail du mode de fonctionnement décrivant les spammeurs, les endroits tel que noms d'emprunt qu'ils emploient, de boîte de P.O, des affaires et des adresses à la maison, des banques qu'ils emploient, des noms d'ISP et des adresses d'IP, des noms de domaine et des adresses E-mail, et les sites Web qui sont employés pour annoncer des produits et des services de Spam. Des spammeurs peuvent être classés par catégorie par les descriptions suivantes :

- spammeurs occasionnels : Ce sont les utilisateurs d'Internet qui font suivre des E-MAIL, cela est généralement considéré comme juste un ennui. C'est intéressant à noter que quelques fournisseurs, y compris Yahoo, considèrent ce type d'E-mail comme une violation de leur accord d'utilisation et pourraient avoir comme conséquence l'arrêt de compte.
- Spammeurs à échelle réduite. La grande majorité de spammeurs entrent dans cette catégorie et ils s'appellent parfois les « Spam Kiddies », généralement ils ne comprennent pas comment les relais sont ouverts et comment les logiciels de filtrages fonctionnent. Ils tendent à employer des outils de Spamming et des listes de courrier, achetées sur des CD ou téléchargées du Web.
- Spammeurs pirates : C'est les spammeurs les plus sophistiqués qui créent des logiciels pour éviter des filtres de Spam.
- Les spammeurs à grande échelle : C'est la catégorie de spammeur qui fait la plupart de dommages et crée la majeure partie du Spam sur l'Internet. Ils utilisent leurs stratégies pour des affaires rentables.

7. Conclusion

Avec l'évolution exceptionnelle du spamming, les organisations « anti-abuse » ont mobilisé une quantité importante de ressources dans le but de minimiser le taux d'attaque. L'utilisation de différentes solutions de détection tel que les anti-spam et les pièges à pirates (HoneyPot) a permis d'identifier 70% des spam envoyés.

CHAPITRE V.

Les Outils Utilisés.

Il faut de tout pour faire un monde.

– proverbe romain.

1. introduction

Pour mieux profiter des avancées technologiques qui ont facilité l'utilisation des outils informatiques même pour ceux qui ne sont pas des spécialistes dans ce domaine, on a adopté quelques outils tant tôt software que hardware et qui coopèrent entre eux pour maximiser le bénéfice de notre solution et pour palier aux différentes lacunes de la solution proposée toute seule. Ainsi dans le reste de ce chapitre vous trouviez des présentations et des descriptions de différents outils qu'on a utilisés.

Certains de ces outils sont utilisés dans l'architecture de base de notre solution tandis que d'autres sont utilisés seulement dans l'environnement de test.

2. SendMail

2.1 Introduction

Sendmail est un serveur de messagerie électronique dont le code source est ouvert. Il se charge de la livraison de messages électroniques. Descendant du logiciel ARPANET DELIVERMAIL, sendmail est un programme très flexible supportant un large éventail de moyens de transfert et de livraison d'emails, incluant le protocole SMTP.

2.2 Historique

Il est banal, aujourd'hui, de dire que l'Internet connaît un succès foudroyant. Ce succès entraîne dans son sillage la messagerie basée sur le protocole SMTP. L'implémentation la plus répandue de ce protocole est sans conteste le programme sendmail conçu par « Eric Allman en 1980 ». Après plusieurs années sans maintenance claire, ayant vu apparaître plusieurs versions dérivées et divergentes, Eric en a repris depuis 1990 la maintenance et l'extension pour prendre en compte les nouveaux besoins exprimés et standardisés, comme MIME et DSN par exemple. **[SMH-21].**

2.3 Avantages et inconvénients

Sendmail est très critiqué pour sa lenteur, sa complexité et sa maintenance difficile en comparaison avec d'autres MTA tels que Qmail et Postfix. Toutefois, il reste le MTA le plus populaire sur Internet, ce qui est certainement dû à son implémentation par défaut dans les différentes variantes d'Unix. Une étude de novembre 2001 estime que 42 % des serveurs mails accessibles publiquement utilisent sendmail.
Il a été implanté dans Mac OS X avec les versions 10.0 à 10.3. **[WSM-22].**

2.4 Configuration

La configuration de SendMail a toujours été une étape difficile dans la vie d'un administrateur de messagerie. Les difficultés ont sans doute des origines multiples, mais ont généralement en commun la complexité et la diversité des situations que doit gérer SendMail.

Une nouvelle évolution s'est dessinée ces dernières années : il n'y a plus guère de configurations de SendMail écrites à la main. La plupart des configurations sont obtenues à partir de définitions de plus ou moins haut niveau, et le fichier `sendmail.cf` est généré automatiquement. **[SAM-15].**

2.5 Le rôle de Send Mail

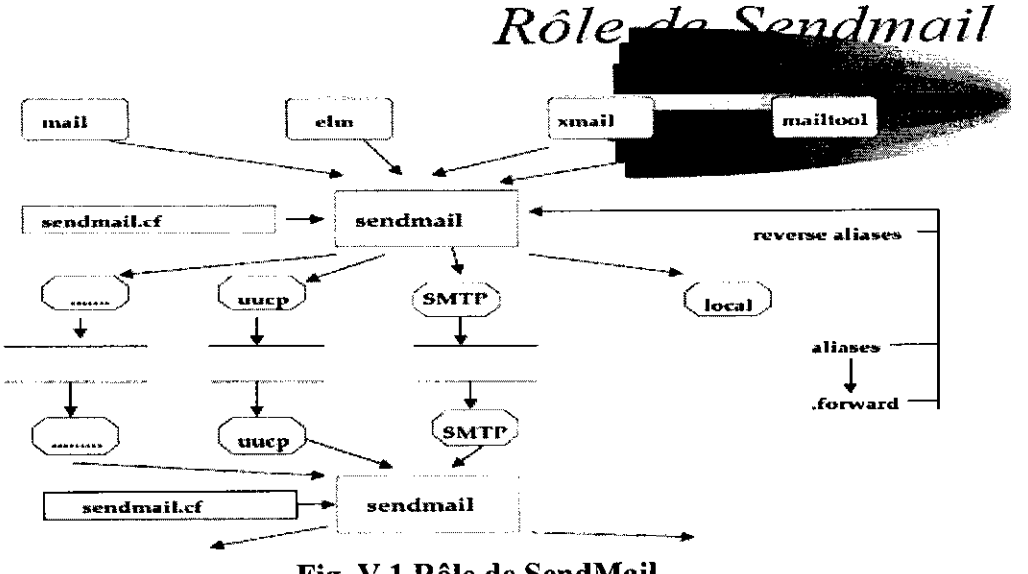


Fig. V.1 Rôle de SendMail.

3. Spam assassin

3.1 Introduction

Spamassassin est un outil anti-spam pouvant être utilisé en collaboration avec SendMail. Il attribue des "scores" à chaque message électronique reçu selon différents tests destinés à déterminer s'il s'agit ou non d'un spam. Pour cela, il effectue de nombreuses vérifications comme :

- la validité des adresses de l'expéditeur et du destinataire,
- la date des messages,
- la présence dans le corps du message d'un mot répertorié sur une liste de mots interdits,
- l'appartenance ou non d'un des serveurs expéditeurs à une liste noire,
- etc....

Chaque test positif augmente le score total d'un message. Quand le score final obtenu dépasse le seuil défini dans la configuration, le message est considéré comme étant un spam.

La base de données de SpamAssassin peut être mise à jour régulièrement grâce à la possibilité d'intégrer les outils Razor, Pyzor et DCC. SpamAssassin inclut également un algorithme de Bayes qui lui permet de reconnaître les nouveaux messages non sollicités à partir d'anciens spam. Enfin, Spamassassin peut être soit utilisé à la demande (appel de l'application chaque fois qu'un message est reçu), soit exécuté en tant que démon. Nous utiliserons, dans le cas de l'interaction avec SendMail, la configuration en mode démon. **[MSI-24].**

3.2 Architecture logique de SpamAssassin

L'architecture logique pour la transmission de serveur de messages se compose :

- d'un premier serveur de mail associé à un anti-spam situé sur la DMZ,
- d'un second serveur de mail situé sur le réseau local.

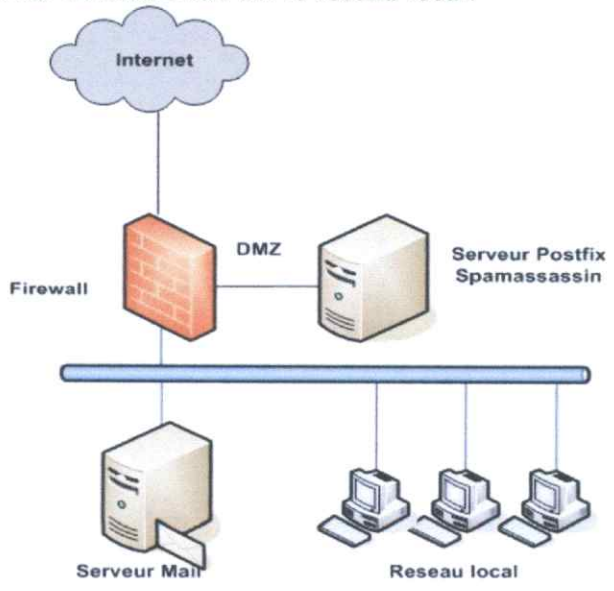


Fig. V.2 Architecture logique de Spamassassin.

3.3 Installation

Vérifiez que vous êtes logués en tant que administrateur (*root*) sur le serveur pour effectuer l'installation de SpamAssassin. Vous devez exécuter les commandes suivantes dans le répertoire /usr/ports/mail/p5-Mail-Spamassassin :

- Sinon, faire séparément :
  - #make
  - #make install
  - #make clean

L'installation a ajouté l'application (*spamc)* et le démon (*spamd)* dans le répertoire /usr/local/bin. Vous devez créer un utilisateur et un groupe « spamassassin » pour permettre l'interaction avec SendMail. **[SAB-23].**

Pour lancer le démon *spamd* au démarrage, vous devez ajouter la ligne suivante dans le fichier /etc/rc.conf : spamd_enable=« YES »

Pour lancer manuellement le démon en mode « debug », tapez la commande suivante :

- #/usr/local/bin/spamd –D –c –d –r /var/run/spamd.pid

3.4 Configuration

Le fichier de configuration local.cf de SpamAssassin se trouvent dans le répertoire /usr/local/etc/mail/spamassassin. Et vous pouvez l'utiliser pour régler et même définir différentes configurations sur Spamassassin.

3.4.1 Configuration de base

Pour faire une configuration basique, vous devez, avant tout, connaître le contexte dans lequel SpamAssassin doit fonctionner. La principale question est de savoir si la tolérance aux spam est faible, moyenne ou élevée. Nous vous conseillons de paramétrer une tolérance moyenne (required_hits 7.0) dans un premier temps et de modifier ce choix si nécessaire.

Voici, ci-dessous, les paramètres que vous devez appliquer dans le fichier local.cf :

- # combien de coups avant de considérer le message comme
- spam required_hits 7.0
- # encapsuler le spam dans un fichier attaché (O non,1 oui)
- report_safe 0
- # langues acceptées (fr : France)
- Ok_languages fr

Pour faire une configuration évolutive, il vous suffit d'intégrer les outils Razor, Pyzor, DCC et l'algorithme de Bayes. **[SAB-23].**

3.4.2 Configuration de Razor (intégré avec SpamAssassin)

Vérifiez que vous êtes logués en tant que root sur le serveur pour effectuer la configuration de Razor. Exécutez les commandes ci-après pour mettre à jour la configuration de Razor :

- #razor-client
- #razor-admin -home=/usr/local/etc/mail/spamassassin/.razor -create
- #razor-admin -home=/usr/local/etc/mail/spamassassin/.razor -discover
- #razor-admin -home=/usr/local/etc/mail/spamassassin/.razor -register

Vérifiez dans le fichier de log /usr/local/etc/mail/spamassassin/.razor/razor-agent.log que l'exécution de chacune des commandes s'est correctement réalisée.

Dans le fichier de configuration de SpamAssassin local.cf, ajoutez les lignes suivantes :

- # utiliser Razor ? oui=1 non=0
- use_razor2 1
- # chemin d'acces au fichier de configuration de razor
- razor_config /usr/local/etc/mail/spamassassin/.razor/razor-agent.conf

3.4.3 Configuration de Pyzor

Vérifiez que vous êtes logués en tant que root sur le serveur pour effectuer l'installation de Pyzor. Vous devez exécuter les commandes suivantes dans le répertoire /usr/ports/mail/pyzor

- Sinon, faire séparément :

  - #make
  - #make install
  - #make clean

Dans le fichier de configuration de SpamAssassin local.cf, ajoutez les lignes suivantes :

- # utiliser Pyzor ? oui=1 non=0
- use_pyzor 1
- # chemin d'acces au client pyzor
- pyzor_path /usr/local/bin/pyzor
- # modification de l'entête du mail par Pyzor? oui=1 non=0
- pyzor_add_header 1

[SAB-23]

3.4.4 Configuration de DCC

Vérifiez que vous êtes logués en tant que root sur le serveur pour effectuer l'installation de DCC.

Vous devez exécuter les commandes suivantes dans le répertoire /usr/ports/mail/dcc-dccd :

- Sinon, faire séparément :

```
#make
#make install
#make clean
```

L'installation est effectuée. Dans le fichier de configuration de SpamAssassin local.cf, ajoutez les lignes suivantes :

```
# utiliser DCC ? oui=1 non=0
use_dcc 1
# chemin d'acces au client DCC
dcc_path /usr/local/bin/dccproc
# modification de l'entête du mail par DCC? oui=1 non=0
dcc_add_header 1
```

3.4.5 Configuration des filtres Bayesiens

Dans le fichier de configuration de SpamAssassin local.cf, ajoutez les lignes suivantes :

```
# utiliser Bayes ? oui=1 non=0
use_bayes 1
# chemin d'acces à Bayes
bayes_path /usr/local/bin/
# auto apprentissage
bayes_auto_learn 1
#
bayes_min_ham_num 100
```

Vous devez vous procurer un fichier contenant des spam afin d'initialiser Bayes. Pour cela, procurez vous des mails de spam et générez un fichier contenant tout ces mails. Il est conseillé d'utiliser le programme « MailTest.exe » et d'envoyer la liste des mails à l'adresse IP du serveur SendMail à l'utilisateur root, le fichier concaténé de spam sera alors le fichier /var/mail/root. **[SAS-25].**

Une fois le fichier obtenu, lancez la commande :

```
#sa-learn --spam --showdots --mbox fichier_spam
```

4. Serveur FTP

Le File Transfer Protocol (*protocole de transfert de fichiers*), ou FTP, est un protocole de communication dédié à l'échange informatique de fichiers sur un réseau TCP/IP. Il permet, depuis un ordinateur, de copier des fichiers depuis ou vers un autre ordinateur du réseau, d'administrer un site web, ou encore de supprimer ou modifier des fichiers sur cet ordinateur.

La variante sécurisée de FTP avec les protocoles SSL ou TLS s'appelle FTPS.

FTP obéit à un modèle client-serveur, c'est-à-dire qu'une des deux parties, le *client*, envoie des requêtes auxquelles réagit l'autre, appelé *serveur*. En pratique, le serveur est un ordinateur sur lequel fonctionne un logiciel lui-même appelé serveur FTP, qui rend publique une arborescence de fichiers similaire à un système de fichiers Unix. Pour accéder à un serveur FTP, on utilise un logiciel client FTP (possédant une interface graphique ou en ligne de commande).

Le protocole, qui appartient à la couche session du modèle OSI et à la couche application du modèle ARPA, utilise une connexion TCP. Il peut être utilisé de deux façons différentes :

1. Mode actif

C'est le client FTP qui détermine le port de connexion à utiliser pour permettre le transfert des données. Ainsi, pour que l'échange des données puisse se faire, le serveur FTP initialisera la connexion de son port de données (port 20) vers le port spécifié par le client. Le client devra alors configurer son pare-feu pour autoriser les nouvelles connexions entrantes afin que l'échange des données soit fait. Ce mode est donc moins sécurisé que le FTP passif pour le client. De plus il peut s'avérer problématique pour les utilisateurs essayant d'accéder à des serveurs FTP lorsqu'ils sont derrière une passerelle NAT. Vu la façon dont fonctionne le NAT, le serveur FTP initie la connexion de données en se connectant à l'adresse externe de la passerelle NAT sur le port choisi. Si la passerelle NAT n'ayant pas de correspondance pour le paquet reçu dans sa table d'état le paquet sera ignoré et ne sera pas délivré au client.

2. Mode passif

Le serveur FTP détermine lui-même le port de connexion à utiliser pour permettre le transfert des données (*data connexion*) et le communique au client. Dans le cas de l'existence d'un pare-feu devant le serveur FTP celui-ci devra être configuré pour autoriser la connexion de données. L'avantage de ce mode, est que le serveur FTP n'initie aucune connexion. Dans le cas des clients FTP sur un réseau local, ce mode est beaucoup plus sécurisé que le FTP en mode actif, car le pare-feu ne devra laisser passer que les flux sortant vers internet pour permettre aux clients d'échanger des données avec le serveur. Il fonctionne donc sans problèmes avec une passerelle NAT.

Deux ports sont standardisés (*well known ports*) pour les connexions FTP : le port 21 pour les commandes et le port 20 pour les données.

Ce protocole peut fonctionner avec IPv4 et IPv6.

Pour accéder à un serveur FTP, on utilise un client ftp, en ligne de commande ou avec une interface graphique.

Les utilisateurs de GNU-Linux ou d'un Unix peuvent consulter une documentation (la plupart du temps installée par défaut), en tapant « man ftp »

La plupart des navigateurs récents autorisent les connexions FTP en utilisant une URL de type :

ftp://nom_d'utilisateur:mot_de_passe@nom_ou_adresse_du_serveur:port_ftp

En informatique anglophone on utilise login pour utilisateur et passwd ou password pour mot de passe.

Par mesure de sécurité, il est conseillé de ne pas préciser le mot de passe, le serveur le demandera. Cela évite de le laisser visible dans l'historique du navigateur, mais ne change rien au fait qu'il soit transmis en clair à travers le réseau.

Il existe de nombreux clients en ligne de commande, comme par exemple ftp, « Wget », « Curl » en téléchargement ou « Wput » pour déposer des fichiers.

La plupart des logiciels possèdent désormais une interface graphique permettant de mettre en place un serveur FTP ou de se connecter à celui-ci pour y copier des données (client FTP).

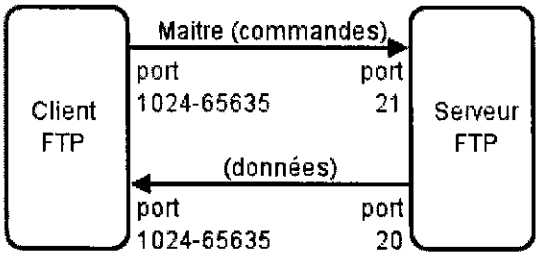


Fig. V.3 Connexion Data en mode Actif.

Le protocole utilise deux types de connexions TCP :

- Une connexion de *contrôle* initialisée par le client, vers le serveur (port 21 en général), pour transmettre les commandes de fichiers (transfert, suppression de fichiers, renommage, liste des fichiers…).
- Une connexion de *données* initialisée par le client ou le serveur pour transférer les données requises (contenu des fichiers, liste de fichiers), cette connexion utilise le protocole Telnet. Le client envoie une commande sous la forme d'une ligne de texte terminée par un retour à la ligne (CR suivi de LF, soit \r\n, ou 0D0A en hexadécimal).

Par exemple, la commande suivante demande le téléchargement du fichier "fichier.txt" :

```
RETR fichier.txt
```

N.B

Les commandes telles que GET ou PUT ne sont pas reconnues dans le protocole FTP, mais souvent utilisées par les logiciels de client FTP.

A la suite de l'envoi de la commande, le client reçoit une ou plusieurs réponses du serveur. Chaque réponse est précédée d'un code décimal permettant au client FTP de traiter la réponse qui peut comporter une ou plusieurs lignes de texte.

Pour l'exemple précédent, si le serveur trouve le fichier demandé, il envoie au client :

150 File status okay ;about to open data connection

Selon ce que le client et le serveur ont convenu, l'un des deux écoute sur le port TCP convenu, et l'autre s'y connecte pour établir la connexion de données. Puis le serveur envoie au client le contenu du fichier demandé, ferme la connexion de données, et envoie la réponse suivante sur la connexion de contrôle :

226 Closing data connection

La connexion de données est établie pour la durée de transmission de données (contenu de fichiers, ou liste de fichiers). En général, elle est établie pour le transfert de données d'une seule commande, à moins qu'un autre mode de transmission soit sélectionné et supporté par le serveur.

La commande « PASV » indique au serveur qu'il doit attendre passivement la connexion en écoutant un port TCP (en général, le port 20). Le port écouté par le serveur est indiqué dans la réponse :

277 Entering Passive Mode (h1,h2,h3,h4,p1,p2)

Où h1 à h4 sont 4 nombres entiers entre 0 et 255 représentant l'adresse IP du serveur, et p1 et p2 représentent le port TCP où le serveur attend la connexion, sous la forme de deux entiers entre 0 et 255 (port_TCP = p1 * 256 + p2).

Dans le cas contraire où le client attend la connexion sur un port TCP, il indique sous la même forme le port écouté en envoyant la commande PORT :

PORT h1,h2,h3,h4,p1,p2

Si tout se passe bien, le serveur répond :

200 Command Okay

Lors du transfert de fichier sur la connexion de données, 2 modes peuvent être utilisés:

- Le mode Binaire: le fichier est transmis tel quel.

- Le mode ASCII: uniquement destiné aux fichiers texte. Le fichier est examiné et des transformations apportées pour conserver un format correct. Par exemple, la fin de

  Ligne est représentée par le caractère <LF> sur un système UNIX, et par la paire <CR><LF> sous Windows. Une machine Windows recevant un fichier texte par FTP récupère donc au final un fichier avec des <CR><LF> en mode ASCII et des <CR> en mode binaire. Ce mode a donc ses avantages, mais peut être source de corruption de fichiers (non texte) pendant le transfert si on utilise un client ancien / en ligne de commande, incapable de s'adapter au type de fichier. Il faut alors basculer de mode (en utilisant généralement la commande BIN) avant le transfert, afin de le conserver intact.

La spécification du protocole FTP (RFC 959) précise qu'il est possible d'effectuer un transfert de fichiers directement entre deux serveurs FTP. Cette fonctionnalité est peu connue, car non disponible dans les logiciels de client FTP.

Pour cela, le client établit une connexion de contrôle par serveur. Au moment d'établir la connexion de données, le client demande à un serveur d'attendre la connexion (commande PASV), et capture l'adresse IP et le port TCP écouté. Ces deux paramètres sont envoyés à l'autre serveur en utilisant la commande PORT. A partir de là, la connexion de données est établie entre les deux serveurs.

Le client est informé de la fin du transfert en recevant une réponse positive de chaque serveur, on appelle couramment ce protocole le FXP.

5. Linux Redhat

Linus B.Torvalds est à l'origine de ce système d'exploitation entièrement libre. Au début des années 90, il voulait mettre au point son propre système d'exploitation pour son projet de fin d'étude. Linus Torvalds avait pour intention de développer une version d'UNIX pouvant être utilisé sur une architecture de type 80386. Le premier clone d'UNIX fonctionnant sur PC a été Minix, écrit par Andrew Tanenbaum, un système d'exploitation minimal pouvant être utilisé sur PC. Linus Torvalds décida donc d'étendre les possibilités de Minix, en créant ce qui allait devenir Linux. Amusées par cette initiative, de nombreuses personnes ont contribué à aider Linus Torvalds à réaliser ce système, si bien qu'en 1991 une première version du système a vu le jour. C'est en mars 1992 qu'a été diffusée la première version ne comportant quasiment aucun bug, avec le nombre croissant de développeurs travaillant sur ce système, celui-ci a rapidement pu intégrer des redéveloppements libres des outils présents sous les systèmes UNIX commerciaux. De nouveaux outils pour Linux apparaissent désormais à une vitesse vertigineuse.

L'originalité de ce système réside dans le fait que Linux n'a pas été développé dans un but commercial.

En effet aucune ligne de code n'a été copiée des systèmes UNIX originaux (en effet Linux s'inspire de nombreuses versions d'UNIX commerciales: BSD UNIX, System V.). Ainsi, tout le monde, depuis sa création, est libre de l'utiliser mais aussi de l'améliorer.

Il existe plusieurs façons de prononcer Linux :

- Dans les pays anglo-saxons deux prononciations existent :
  - Lineuxe.
  - Laïnuxe.
- En France on prononce tout simplement :
  -Linux

Bien que Linux ait été initialement conçu pour fonctionner sur plateforme PC, il a désormais été porté (c'est-à-dire adapté) vers de nombreuses autres plates formes, telles que Macintosh, stations SPARC, stations DEC Alpha, et même des plates formes telles que des assistants personnels PDA, voire des consoles de jeu vidéo !

5.1 Notions de noyau et de distribution

Linux est architecturé autour d'un **noyau** (***kernel***) chargé de prendre en charge le matériel. On appelle **distribution** l'assemblage d'un ensemble de logiciels autour d'un noyau Linux afin de fournir un système clé en main. Le noyau d'une distribution peut-être mis à jour afin de permettre la prise en compte de matériels récents, toutefois cette manipulation consistant à recompiler le noyau est délicate car elle nécessite un certain niveau de connaissance du système et du matériel. La recompilation du noyau est à réserver aux spécialistes ou bien aux utilisateurs prêts à rendre inutilisable leur système dans le but d'apprendre.

5.2 Caractéristique du système :

Linux est un système d'exploitation proche des systèmes UNIX pouvant être exécuté sur différentes plates-formes matérielles : x86 (c'est-à-dire des plates-formes à base de processeurs Intel, AMD, etc.), Sparc, PowerPC, Alpha, ARM, etc.
Linux est ainsi un système multi plate-forme. Il est également multi-utilisateurs (plusieurs personnes peuvent en même temps travailler sur le même ordinateur), mais aussi multi-tâches (plusieurs applications peuvent être lancées en même temps sans qu'aucune n'affecte les autres) à noter qu'il est aussi multiprocesseur.
Linux est considéré comme un système fiable, robuste et puissant. Il est d'ailleurs capable de fonctionner avec très peu de ressources sur des ordinateurs bas de gamme très peu puissants.
Le système Linux possède notamment les avantages suivants :

- Le support des standards de l'internet, c'est-à-dire des protocoles TCP/IP, la famille de protocoles utilisée sur Internet. Linux est donc un moyen gratuit de créer un réseau local, de se connecter à Internet et de mettre en place un serveur.
- Une sécurité accrue dû à la transparence de son code source et de la réactivité de la communauté lors des annonces de vulnérabilités.

- Un cloisonnement des espaces mémoire et de l'espace disque couplé à une gestion pointue des droits permettant de gérer un grand nombre d'utilisateurs avec un niveau de risque minimal.
- Un noyau entièrement configurable en fonction du matériel de la machine sur laquelle le système est installé afin de maximiser les performances.
- Open source (on dit libre en français), et gratuit (aucune licence à payer).
- Compatibilité multi architecturales.
- Une distribution Linux récente peut fonctionner sur des ordinateurs à faible performances (à partir des Architecture Intel 386).
- Peut s'installer et démarrer sur une clé USB (via votre BIOS).
- Peut prendre en charge plusieurs versions de noyau dans un même système.
- Est beaucoup moins gourmand en ressources systèmes.
- Accepte beaucoup plus de requêtes sollicitées à la seconde.
- Linux est un vrai système multitâche.
- Gestions de plusieurs types de partitions Linux et Windows (fat16, fat32, ntfs, ext2fs, ext3fs, ext4fs (?), reiserfs, xfs,...).
- Pas besoin de défragmenter les partitions de type Linux (ext2fs, ext3fs, reiserfs, xfs,...).
- Linux peut avoir plusieurs interfaces graphiques et de bureaux virtuels.
- Est parfaitement fonctionnelle quand Linux est installé en mode terminal (c'est-à-dire sans Interface graphique).
- L'interpréteur de commande n'est pas intégré de manière fixe au noyau.
- Les commandes dans un terminal peuvent interagir avec les programmes graphiques.
- Un crash d'un environnement graphique (comme KDE ou Gnome) n'entraîne pas Un crash du système. L'interface graphique est totalement indépendante du système.
- Possibilité de télécharger et de gérer des milliers de logiciels via les outils de mise à jour du Système (qui est personnalisable) comme l'utilitaire Synaptic pour les paquetages Debian (.deb).
- L'installation des paquetages dans Linux est plus uniformisée (voir organisée) que ceux dans Windows (.exe, .msi), ce qui contribue aussi à la stabilité de Linux et des logiciels installés.
- Les bogues critiques sont corrigés rapidement.
- Une sécurité plus sévère et efficace sur l'accessibilité des programmes.
- Dépendance partielle du BIOS de l'ordinateur car Linux possède son propre BIOS (Noyau) détections matériels.
- A une plus grande tolérance aux pannes que Windows.
- Gestion et administration de tout le système à distance.
- Existe en version Live CD parfaitement fonctionnelle.
- Le copier/coller rapide avec sélection d'un texte.

5.3 Inconvénients :

- Est plus complexe que Windows.
- Le nombre impressionnant de distributions Linux peut dérouter le débutant Linuxien.
- Pas assez de publicité pour mieux faire connaître Linux dans le monde.
- Un manque de pilotes ou de drivers propriétaires pour Linux.
- Pas suffisamment de logiciels propriétaires professionnelle.

CHAPITRE VI.

Etude des HoneyPot.

Prefer the standard to the offbeat.

– Strunk & White.

1. Définition

Les HoneyPot (*pots de miel*) sont des systèmes vulnérables déployés sur Internet et surveillés par des administrateurs et des experts de sécurité. Les HoneyPot sont utilisés pour faire de la collection d'informations sous forme de fichiers journaliers (*log files*).
Cette information est utilisée pour connaitre les nouvelles techniques, outils et motivations des hackers pour mieux parer contre les attaques sur les systèmes en production. Les HoneyPot déployés pour ce type d'utilisation sont appelés des HoneyPot de recherche.

Un HoneyPot peut protéger une organisation par trois moyens :

- La prévention des attaques : laissé un hacker jouer sur l'honeypot au lieu des systèmes de production.
- La détection d'activités malicieuses : utilisation des systèmes de détection d'intrusions (IDS).
- La réponse aux attaques : routage de tout trafic malencontreux vers des honeypots.

Lorsqu'un HoneyPot protège une organisation c'est un honeypot de production. Comparé à un honeypot de recherche, un honeypot de production génère moins de trafic. **[HP-28].**

Le tableau suivant donne un récapullatif de ces deux types de HoneyPot :

Niveau d'interaction	Type d'honeypot	Utilité de l'honeypot
Faible	Production	Protection d'une organisation
Forte	Recherche	Collecte d'informations

Table VI.1 Caractéristiques des deux types d'HoneyPot.

Théoriquement, un HoneyPot n'a pas normalement de trafic parce qu'il n'a pas d'activité légitime. Cela veut dire que, chaque interaction avec un honeypot signifie un malicieux ou une mauvaise activité. Une connexion quelconque à un HoneyPot est considérée comme une attaque ou compromise. Grâce à sa simplicité, il a énormément d'avantages et bien évidemment des inconvénients.

2. Avantages et Inconvénients

2.1 Avantages

- Le HoneyPot est une conception très simple qui donne des petits ensembles de données mais de haute valeur : le HoneyPot collecte un petit nombre d'information. Au lieu de collecté des Go par jour, il peut collecter environ un Mo de donnée par jour. Au lieu de générer 10 milles alertes par jour, il ne peut générer que 10. Mais il faut souligner que le honeypot ne capture que des mauvaises activités. Chaque interaction avec un honeypot signifie une activité dangereuse. Alors, l'HoneyPot réduit le taux de collecte, toute en gardant l'information très valable, qui ne concerne que l'attaquant. Cela signifie qu'il est plus facile (et moins chère) à analyser les données collectées pour dériver la valeur à partir de ces données.
- Ressource minimale : Un HoneyPot n'a besoin que de peu de ressources pour capturer les mauvaises activités. Un ancien ordinateur Pentium avec 128Mo de mémoire peut facilement contrôler la classe B du réseau.
- Chiffrement ou IPv6 : Différence à presque toutes les technologies de sécurité (*comme IDS système*), l'honeypot travaille bien dans l'environnement Ipv6 ou avec des données chiffrées. N'importe quelles données (instruction ou commande) qu'un attaquant envoie, sera détectée, capturée et analysée.
- Simplicité : En fin, l'honeypot est simple, il n'y a pas d'algorithme à développer et il est facile à configurer.

2.2 Inconvénients

Comme toutes les technologies, l'honeypot a aussi des faiblesses. Parce qu'il ne remplace aucune technologie courante, mais il travaille avec les technologies existantes.

- Vue limitée : l'HoneyPot ne peut détecter et capturer que des activités directement interactives avec lui.
- Risque : Toutes les technologies de sécurité ont des risques. Le pare-feu a le risque d'être pénétré, le chiffrement à risque d'être démoli. L'IDS a le risque de se tromper. L'honeypot a

aussi des risques. Il a un risque d'être pris la relève par un attaquant et d'être utilisé à faire du mal aux autres systèmes. Les risques sont variés selon chaque type d'honeypot.

3. Les types d'HoneyPot

Il y a deux catégories d'HoneyPot: *low-interaction* et *high-interaction* honeypots. Ces deux catégories nous donnent des connaissances sur quel type d'honeypot on travaille et avec qui on a une interaction, sa faiblesse et sa force. L'interaction définie le niveau d'activité qu'un honeypot permet aux attaquants. Low-interaction HoneyPot a un nombre d'interaction limité, il travaille grâce à une simulation des services et des systèmes d'exploitation. Les activités des attaquants sont limitées au niveau de la simulation par l'honeypot. Par exemple, Un service simule SMTP entendu sur le porte 25 peut simuler l'ouverture d'une session SMTP ou il est également supporter quelques autres commands SMTP. L'avantage d'une low-interaction est la simplicité. L'honeypot a tendance à déployer et maintenir facilement avec un risque minimal. Normalement, il a besoin d'installer des logiciels, de sélectionner des services ou de système d'exploitation que l'on veut simuler et surveiller. Ainsi l'attaquant n'accède jamais au système d'exploitation principal pour lui faire du mal.

L'inconvénient principal de low-interaction HoneyPot est que les informations qu'il a collectées sont limitées. Donc, il est plus facile pour un attaquant de détecter une low-interaction HoneyPot. Bien qu'un low-interaction HoneyPot soit bien configuré, un attaquant qualifié peut détecter sa présence. Les exemples d'honeypot sont: *Spector, Honeyd et KFSensor.*

High-interaction HoneyPot est différent. Il est a des solutions complexes parce qu'il a besoin d'applications réelles et d'un système d'exploitation réel. Aucun n'est simulé, on donne aux attaquants des choses réels. Si on veut un HoneyPot Linux qui lance le serveur SMTP, on construit un système Linux réel qui lance un serveur SMTP (Sendmail par exemple). Il y a deux avantages majeurs de cette solution :

- on peut capturer extensivement des informations. En donnant aux attaquants un système réel à communiquer avec, on peut suivre tous ses comportements.
- une High-interaction HoneyPot ne fait pas attention à comment un attaquant se comportera. Au lieu de cela, elle propose un environnement ouvert qui capture toutes les activités.

En générale, une High-interaction HoneyPot peut faire ce qu'un low-interaction honeypot fait et beaucoup plus. Mais, il est plus complexe à déployer et maintenir. Voici quelque HoneyPot High-interaction : *Symantec Decoy Server, Honeynet.*
Pour mieux comprendre le mode de fonctionnement des deux types de HoneyPot cités ci-dessus, on va commencer par : low-interaction HoneyPot ou bien le « *Honeyd* ». **[HHN-30].**

3.1 Low-interaction HoneyPot « Honeyd »

Honeyd est un HoneyPot low-interaction développé par *Niels Provos*. C'est un Open Source conçu pour le système Unix. Il surveille les espaces IP qui ne sont pas encore utilisés.
Quand il trouve une connexion il l'accepte et commence à communiquer avec l'attaquant en se considérant comme victime.

Par défaut, Honeyd détecte et collecte n'importe quelle connexion vers les ports TCP et UDP. En plus, on peut configurer les services simulés pour contrôler les ports spécifiques comme un serveur FTP simulé qui contrôle le port 21ou un serveur mail comme un serveur SMTP simulé qui contrôle le port 25.
Quand un attaquant connecte au service simulée, honeyd non seulement détecte et collecte les activités, mais il capture aussi les interactions de l'attaquant avec le service simulée. Dans le cas de serveur SMTP, on peut potentiellement capturer le compte que l'attaquant a utilisé, les commandes qui a introduit, ainsi que sont identité.
Quelque honeypots, comme Honeyd, peuvent non seulement simuler des services, mais simuler des systèmes d'exploitation actuels. D'autre part, Honeyd peut apparaître comme routeur Cisco, web server Win XP ou serveur DNS sous Linux. Il y a certains avantages quand on simule des systèmes d'exploitation différents.

D'abord, l'HoneyPot peut être mieux mélangé dans le réseau existant s'il a la même apparence et comportement du produit système. En suite, on peut choisir des attaquants spécifiques en leur proposant des systèmes et des services pour mieux connaitre leurs objectifs ou on proposant ceux que l'on veut étudier. La clé de toutes ces simulations est l'existence d'un élément relié au système d'exploitation simulé, c'est le service simulé.
Quand un attaquant connecte au service simulé, ce service a un comportement et apparaît comme un sous système d'exploitation spécifique. Par exemple, si on a un service simulé d'un serveur web et on veut que notre HoneyPot apparaît comme Window2000 serveur, on peut simuler son comportement à la façon qu'un IIS web server se comporte. Pour Linux, on peut le simuler à la façon d'un web server Apache ; et de même pour simuler un serveur Mail sous linux en utilisant un serveur de messagerie comme Sendmail, La plupart des honeypots simulent le système d'exploitation selon cette manière. **[HPW-27].**

3.2 High-interaction HoneyPot « Honeynet »

Honeynet est un exemple de HoneyPot High-interaction. Il n'est pas un produit, ni une solution logiciel que l'on installe sur un ordinateur. Au lieu de cela, honeynet est une architecture, une entité du réseau des ordinateurs construits pour être attaqué. L'idée est de proposer une architecture qui crée un réseau contrôlé à haut niveau où toutes les activités sont contrôlées et capturées.
Dans le réseau on met des ordinateurs intentionnellement vulnérables, ce sont des ordinateurs réels qui lancent des applications réelles. L'attaquant trouve une fuite d'attaque et entre dans les systèmes proposés. Quand il fait cela, il ne prend pas conscience qu'il est dans un Honeynet.
Toutes ses activités, de la session SSH encryptée vers email et téléchargement des fichiers, sont capturées sans aucune connaissance de l'attaquant.
Cela se fait par insertion des modules dans le noyau du système victime qui capture toutes les actions des attaquants. En même temps, l'honeynet contrôle les activités des attaquants. Il le fait en utilisant une passerelle Honeywall. Cette passerelle permet le trafic d'entrer vers le système victime, mais contrôle le trafic sortie en utilisant la technologie de prédiction des instructions. Cela donnera aux attaquants une flexibilité dans l'interaction avec le système victime. L'exemple du déploiement de honeynet est dans la figure suivante:

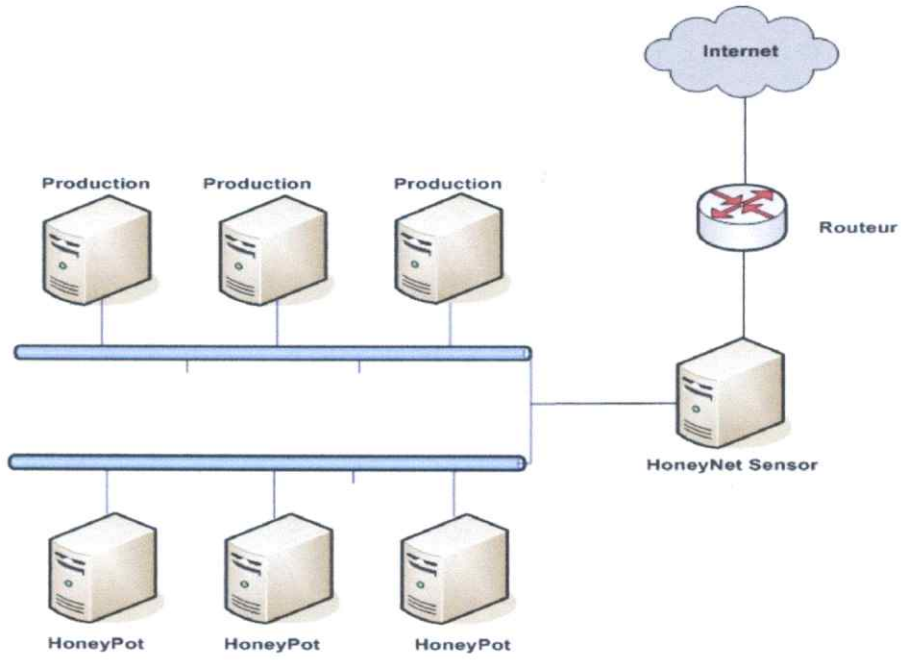


Fig.VI.1 Le schéma de HoneyNet.

3.2.1 Contrôle de données

Le HoneyNet doit contenir toute activité. S'il est compromis le honeypot ne doit pas compromettre d'autres systèmes ne faisant pas partie du réseau d'honeypot. Si le honeypot est placé derrière un pare-feu ou routeur, les connexions en sortie peuvent effectivement être contrôlées.
Le honeypot doit être souple. Pour qu'il n'y ait pas une suspicion de la part du hacker sur l'honeypot, les droits d'utilisation sur le honeypot ne doivent pas être limités uniquement à la lecture. Il faut aussi permettre l'exécution ou l'écriture des fichiers contrôlée. **[HP-28].**

3.2.2 Capture des données

Toutes les activités sur l'honeynet doivent être enregistrées. Il est important de pouvoir capturer le plus d'informations possibles sur le hacker sans qu'il le sache.
Par contre, les données enregistrées ne devraient pas être stockées uniquement sur l'honeypot. Il faut pouvoir enregistrer les données depuis plusieurs sources (fichiers de log du système, log des IDS, analyseurs réseau). Il est important aussi pour la capture et le contrôle de données, d'assurer l'intégrité de données. **[HP-28].**

3.2.3 Collection des données

Dans un environnement distribué qu'il soit un réseau physiquement distribué à l'échelle mondiale ou un réseau logiquement distribué, les données doivent être stockées dans un lieu central. La corrélation de ces données par rapport au temps permettra d'étudier les motifs d'attaque d'un seul ou d'un groupe de hackers. **[HP-28].**

3.2.4 Génération de HoneyNet

1. La 1ère génération des honeynets

Sur les HoneyNet de 1ère génération, le contrôle et la capture des données ont été implémentés via plusieurs dispositifs (routeur, pare-feu, IDS) permettant l'enregistrement et le contrôle de données en Multi-couches. **[HP-28].**

La figure suivante illustre la 1ère génération des HoneyNet :

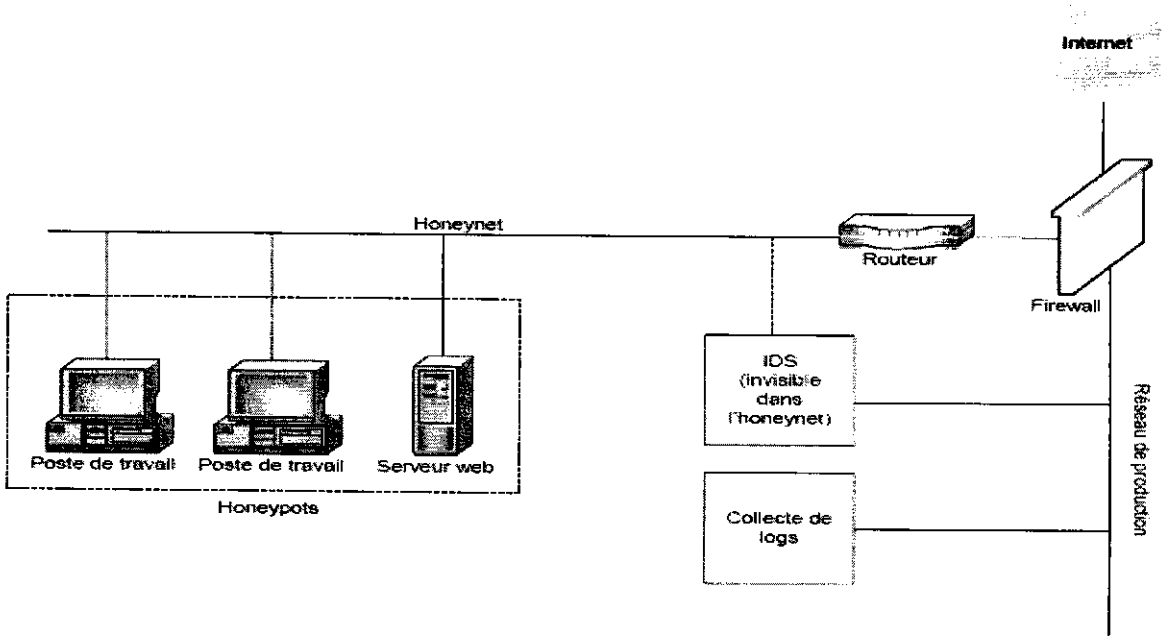


Fig.VI.2 1ère génération de HoneyNet.

L'avantage de multicouche est que si un des dispositifs tombe en panne, la surveillance et la capture de données sont encore garanties. Par contre, ce type d'honeynet est difficile à administrer dû à la complexité de l'architecture réseau.
Voici comment le contrôle et la capture des données sont implémentés sur un honeynet de première génération.

a. Le contrôle de données

- Le firewall offre la première couche de protection. Le firewall ne contrôle pas le nombre de connexions vers l'extérieur mais limite le nombre de connexions en entrée.
- Le routeur limite toute connexion vers l'Internet. Il permet aussi de cacher le firewall du réseau intérieur. C'est la deuxième couche de protection.

b. La capture de données

- Le firewall garde un journal de toutes les connexions de l'Internet et de honeynet (1ère couche de capture).
- Le système de détection d'intrusion garde aussi un log de toutes les activités du réseau d'honeynet. Il est accessible depuis le réseau de production mais non pas depuis le honeynet (2ème couche de capture).
- Sur les HoneyPot eux-mêmes, les journaux du système (système logs) peuvent être exploitables (3éme couche de capture).

2. La 2ème génération de HoneyNet

Sur un HoneyNet de deuxième génération, le contrôle et la capture de données sont implémentés sur un seul dispositif. Dans l'honeynet de première génération, l'élément de contrôle est principalement le pare-feu, mais celui-ci est remplacé par un pont (bridge) dans un honeynet de deuxième génération. Le bridge est difficilement détectable car il n'a pas d'adresse IP et il ne décrémente pas le champ TTL. **[HHN-30].**

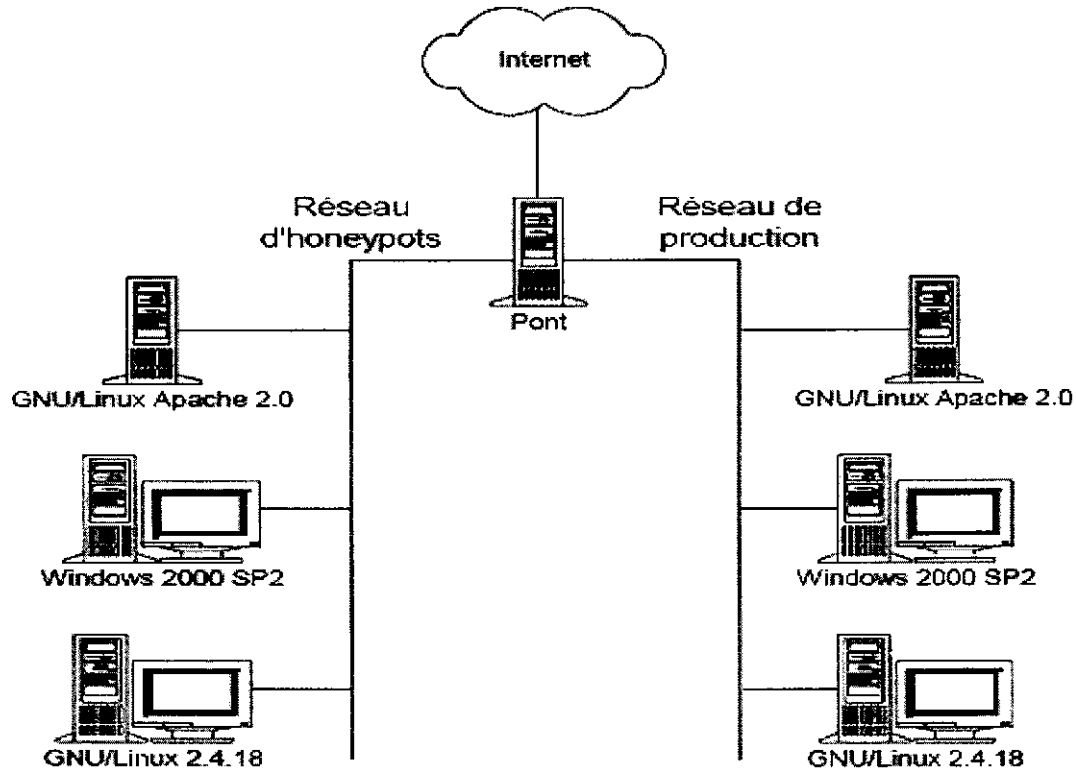


Fig.VI.3 2ème génération de HoneyNet.

Le principe de HoneyPot de 2ème génération est la détection du trafic malveillant destiné au réseau de production, puis le routage de celui-ci vers le réseau de honeypot.

3.3 HoneyPot virtuel

Un HoneyPot virtuel est une simulation d'architecture d'une machine sur une machine servant comme hôte. Cette possibilité permet de simuler un réseau complexe d'honeypots sur une seule machine. Les honeypots virtuels peuvent être classés en deux catégories:

- Dans un HoneyPot purement virtuel, les dispositifs de contrôle et capture de données ainsi que les honeypots sont déployés sur une seule machine.

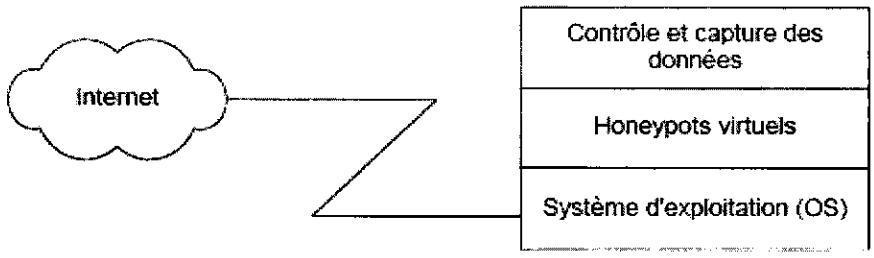


Fig.VI.4 HoneyPot Virtuel.

- Dans un HoneyPot virtuel hybride, les dispositifs de contrôle et de capture de données sont déployés sur une machine séparée.

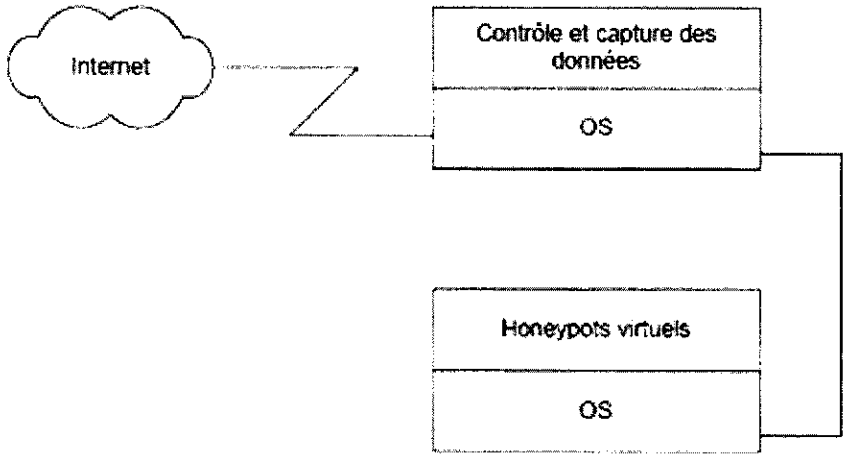


Fig.VI.5 Architecture du HoneyPot Virtuel hybride.

Un HoneyPot purement virtuel permet une administration facile sur une seule machine. Toutefois si un hacker parvient à prendre contrôle de l'hôte, tout l'honeynet est compromis, ce qui n'est pas le cas dans un honeynet virtuel hybride.

4. Les HoneyPot virtuels avec Honeyd

4.1 Introduction

Les HoneyPot virtuels permettent une mise en oeuvre facile des honeypots due à la simplicité de leur installation et leur administration qui peut se faire sur une seule machine hôte. La simulation des honeypot virtuels avec *honeyd* permet de créer une topologie de serveurs et machines vulnérables et attirantes aux hackers. De plus, des possibilités de routage offertes permettent la simulation par exemple de la décrémentation de TTL des paquets, la latence et même des paquets perdus.

4.2 Fonctionnement

4.2.1 Fichier de configuration

Honeyd est un daemon exécuté sur une machine qui répond aux paquets destinés aux honeypots virtuels qu'il simule.

4.2.2 HoneyPot client

Récemment, Christian Seifert (*Honeynet New-Zealand*) a publié un outil très intéressant. HoneyC de son nom, permet non pas d'attendre passivement les requêtes des botnets mais HoneyC via le moteur de recherche Yahoo ! Envoi des requêtes http afin de détecter des URLs hébergeant des malwares (notion de client honeypot). On note l'initiative *stopbadware.org* dont Google fait parti qui a recensé, à l'écriture de cet article, 405 URL dangereuses. Si vous ouvrez une de ces URLs après une requête de recherche sur Google, alors vous serez redirigé vers un message d'avertissement indiquant que l'URL contient un malware.

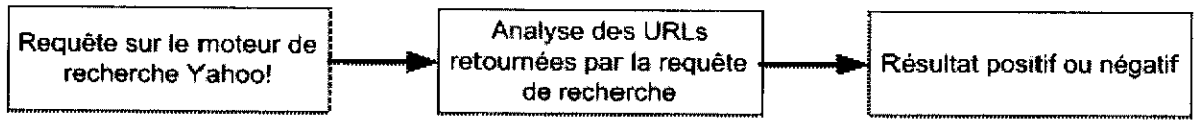


Fig.VI.6 Architecture de mode de fonctionnement du HoneyC.

4.2.3 Valeurs de HoneyPot

HoneyPot aide à prévenir par certaines façons. Premièrement, il contre les attaques automatiques. Ces attaques basent sur les outils qui scannent les entités du réseau de façon aléatoire pour chercher les vulnérabilités du système. Si une des vulnérabilités est trouvée, les outils automatiques attaqueront le système et obtiendront le droit d'accès au système.

Deuxièmement, l'HoneyPot peut aider à contrer ces attaques par un retardement de scans, même s'il faut les arrêtés. Les HoneyPot retardent les actions des hackers. Ils les font en utilisant une variété de valeurs, comme Window taille zéro... les HoneyPot peuvent protéger votre organisation des attaquants humains. L'idée est de refuser un attaquant, lui faire perdre son temps

à communiquer avec les honetpots. Pendant ce temps là, votre organisation détecte les activités de l'attaquant et a assez de temps pour répondre ou arrêter cet attaquant. Si l'attaquant sait que votre organisation utilise l'honeypot, mais il ne sait pas quels systèmes sont les honeypots et quels systèmes sont les machines légitimes. Il a des risques d'être pris dans le système d'honeypot et il décide donc de ne pas attaquer votre organisation. L'honeypot dissuade l'attaquant. Un exemple est Déception *Toolkit,* qui est un *low-interaction HoneyPot.* **[HPF-29].**

5. Conclusion

Malgré l'existence des hackers chevronnés qui réussissent souvent à franchir les systèmes hautement sécurisés, ils peuvent êtres pris au piège par les HoneyPot et suivi dans toutes leurs activités tout cela sans se rendre compte, c'est l'objectif de la mise en place d'un tel système.

CHAPITRE VII.

Format de message.

"The time has come," the Walrus said,
"to talk of many things".

– L.Carroll.

1. Introduction

Pour pouvoir arriver à bien comprendre les systèmes les plus compliqués qui existent il faut bien comprendre les éléments les plus atomiques constituant ces gros systèmes et déterminer avec exactitude les objets manipulés, ce qui rend l'interaction avec de tels systèmes plus flexible.

2. Composants génériques

Les messages électroniques tout comme les messages postaux sont composés d'une **enveloppe** et d'un **contenu.**

L'enveloppe contient les informations nécessaires pour effectuer la transmission et la distribution. Tandis que le contenu contient l'objet à livrer au destinataire.

Cependant, quelques systèmes de messagerie peuvent utiliser une partie de l'information qui est situé dans le contenu pour créer une enveloppe. Ce standard a pour intention de faciliter l'acquisition de telles informations par programmes. Ainsi que d'autres systèmes de messagerie peuvent enregistrer les messages dans des formats qui diffèrent de celui définit par ce standard.

Les messages sont des **lignes de texte**. Aucune spécification particulière n'est prévue pour coder les dessins, les textes structurés ou tableaux. On n'a pas pris en considération les questions de compression ou d'efficacité de stockage ou de transmission et le standard tend à être libéral au niveau du nombre de bit consommé.

Une structure plus robuste permet le codage de l'information avec des fontes multiples, multicolores et multidimensionnelles. Une structure moins robuste, présente dans la plupart des machines restreindrait, avec plus de sévérité, le potentiel à ajouter des champs et de décider d'inclure des champs spécifiques. Contrairement à la communication basée sur le papier, il est intéressant de noter que le destinataire d'un message peut exercer une quantité extraordinaire de contrôle sur l'apparence du message.

3.3.1 Description générale

Un message comprend des champs d'en-tête, et facultativement un corps.
Le corps est une simple séquence de lignes contenant des caractères ASCII. Il est séparé des en-têtes par une ligne nulle (c'est à dire une ligne que rienne précède appelée un retour chariot à la ligne ou bien CRLF).

3.2 Les champs d'en-tête

Chaque champ d'en-tête peut être vu comme une simple ligne logique de caractères ASCII, comprenant un **nom** de champs et un **corps** de champs.

Par commodité, la portion du corps de champ de cette entité conceptuelle peut être découpée en une représentation multi-ligne. On appelle cela le "**plissage**" (*folding*). La règle générale est que, partout où il peut y avoir un retour à la ligne avec des espaces blancs (*linear-white-space (LWSP)*) -pas simplement un caractère LWSP- un retour chariot à la ligne (*CRLF*) immédiatement suivie par **au moins** un caractère LWSP peut être inséré à sa place.

Ainsi la simple ligne :

```
To: "Joe & J. Harvey" <user @Org>, JJV @ BBN
```

Fig. VII.1 Champ d'en-tête « TO » model 1.

peut être représenté par:

```
To: "Joe & J. Harvey" <user @ Org>,
                    JJV@BBN
```

Fig. VII.2 Champ d'en-tête « TO » model 2.

et par

```
To: "Joe & J. Harvey"
                <user @ Org>, JJV
@BBN
```

Fig. VII.3 Champ d'en-tête « TO » model 3.

et par

```
To:"Joe &
J. Harvey" <user @Org>, JJV @ BBN    .
```

Fig. VII.4 Champ d'en-tête « TO » model 4.

Le processus qui consiste à modifier cette représentation multi ligne de champs d'en-tête en une représentation mono ligne est appelé "**déplissage**". Le déplissage s'accomplit en considérant le CRLF immédiatement suivi par le caractère LWSP, comme un simple caractère LWSP.

Remarque: Alors que le standard autorise le plissage n'importe où une ligne d'espace blanc est permise, il est recommandé pour les champs structurés, tel que ceux contenant les adresses, de limiter le plissage aux coupures syntaxiques de haut niveau. Pour les champs d'adresses, il est recommandé que de tels plissages se passent entre les adresses, après les virgules (comma) de séparation.

Une fois que le champ a été déplissé, la structure peut être vue comme étant composée d'un nom de champ suivie de deux points (:) suivie par le corps du champ et terminée par un retour chariot à la ligne (CRLF). Le nom du champ doit se composer de caractère ASCII imprimable c'est à dire des caractères ayant une valeur ASCII comprise entre 33 et 126, des décimaux, exception faites aux deux points.

3.3 Définition des champs d'en-tête

Ces règles indiquent une méta-syntaxe de champ, sans chercher dans la syntaxe interne ou les types particuliers. Leur objectif est de permettre la détection des champs ; Ainsi, elles donnent, à travers une analyse syntaxique de haut niveau, une image de chaque champ sur une seule ligne. Le tableau suivant illustre ces différents champs :

champ	nom de champ":" [corps du champ] CRLF.
nom de champ	n'importe quel CHAR, à l'exclusion des CTL, SPACE, et ":".
corps de champ	contenu du corps de champ [CRLF caractère-LWSP corps de champ].
contenu du corps de champ	Les caractères ASCII créent le corps de champ, comme ce qui est défini dans les sections suivantes, qui consiste à des combinaisons d'atome, des chaînes entre guillemets, et marqueurs spéciaux, ou autre objets comprenant du texte.

Table VII.1 Les Champs d'en-tête.

4. Analyse Syntaxique

4.1 Champs spécifiques

La syntaxe indique que, quand ils sont présents, certains champs, doivent être dans un ordre particulier.

Cependant il **n'est pas** nécessaire que les champs d'en-tête soient dans un ordre particulier, excepté que le corps du message doit être **après** les en-têtes. Il est recommandé, que s'ils sont présents, les en-têtes sont envoyés dans l'ordre suivant :

"Return-Path", "Received", "Date", "From", "Subject", "Sender", "To", "cc", etc.

Le tableau suivant illustre les différents champs syntaxiquement spécifiés:

Champs	Définition	Commentaires
message	champs *(CRLF *texte)	Tout ce qui est après la ligne nulle est le corps du message.
champs	Dates Source Destination Champs optionnel	Instants de création. Auteur id & une. Adresse nécessite. D'autres champs.
source	[trace] Origine [resent]	Réseau traversé. Courrier originel. Renvoyer, fais suivre.
trace	return received	Chemin vers l'expéditeur. Marque le destinataire.
return	"Return-path" ":" route-addr	Adresse de retour à l'envoyeur.
received	"Received" ":" ["from" domaine] ["by" domaine] ["via" atome] ("with" atome) ["id" msg-id] ["for" addr-spec] ";" date-time	Un par relais Hôte expéditeur Hôte receveur Chemin physique Protocole de courrier/lien identificateur de message Forme initiale Temps de réception (time received)
originator	Authentic ["Reply-To" ":" 1#address]	Adresse authentifiée.
authentic	"From" ":" mailbox ("Sender" ":" mailbox "From" ":" 1#mailbox)	Un simple auteur. Soumissionnaire réel. Auteurs multiples ou non expéditeur.
resent	resent-authentic ["Resent-Reply-To" ":" 1#address])	

resent-authentic	"Resent-From" ":" mailbox ("Resent-Sender" ":" mailbox "Resent-From" ":" 1#mailbox)	
dates	origin-date [resent-date]	Date d'origine. Date de renvoie.
orig-date	"Date" ":" date-temps	
resent-date	"Resent-Date" ":" date-temps	
destination	"To" ":" 1#address "Resent-To" ":" 1#address "cc" ":" 1#address "Resent-cc" ":" 1#address "bcc" ":" #address "Resent-bcc" ":" #address	Destination primaire. Destination secondaire. Copie invisible (Blind carbon).
optional-field	"Message-ID" ":" msg-id "Resent-Message-ID" ":" msg-id "In-Reply-To" ":" *(phrase / msg-id) "References" ":" (phrase / msg-id) "Keywords" ":" #phrase "Subject" ":" text "Comments" ":" text "Encrypted" ":" 1#2word extension-field user-defined-field	Destinés à êtres défini et peuvent êtres pré-vidés.
msg-id	"<" addr-spec ">"	Id unique de message.
Extension Field	N'importe quel champ qui est défini dans un document publié comme une extension formelle à cette spécification.	
user-defined-field	Des champs spécifiés par l'utilisateur et les noms pour ces champs doivent être unique et peuvent être pré-vide par des extensions publiques.	

Table VII.2 Les Marqueurs lexicaux.

4.2 Éclaircissements

4.2.1 Faire suivre « *Forwarding* »

Certains systèmes permettent aux destinataires des courriers de répondre aux messages (*forwarding*), en s'appropriant l'en-tête d'origine, et en y ajoutant quelques champs nouveaux.

Généralement les champs ainsi ajoutés contiennent l'en-tête du message original inséré au début du corps du nouveau message.

4.2.2 Reçu « *received* »

Une copie de ce champ est ajoutée par le service de transport qui relaye le message. Cette information peut être assez utile pour suivre les problèmes de transport.

Les noms d'hôtes d'expédition et de réception et de date-temps de réception peuvent être spécifiés. Le paramètre "via" peut être utilisé, pour indiquer par quel mécanisme le message est envoyé tel que Arpanet ou Phonenet, et le paramètre "with" peut être utilisé pour indiquer quel niveau de protocole courrier ou connexion doit être utilisé, tel que le protocole de courrier SMTP ou le protocole de transport X.25.

Remarque: Plusieurs paramètres "with" peuvent être inclus pour désigner exactement le jeu de protocole utilisé.

Certains services transports mettent en file d'attente les courriers ; l'identificateur de message interne qui est assigné au message peut être noté en utilisant le paramètre "id". Quand le serveur hôte d'expédition utilise la spécification d'une adresse de destination que l'hôte receveur réinterprète, à travers l'extension ou la transformation, le serveur receveur pourrait enregistrer la spécification originelle, en utilisant le paramètre "for". Par exemple, quand une copie de courrier est envoyée à un membre d'une liste de télédistribution, ce paramètre peut être utilisé pour enregistrer les adresses originelles qui sont utilisé pour définir la liste.

4.2.3 Le champs « *From* »

Ce champ contient l'identité de la ou des personnes qui ont voulu que ce message soit envoyé. Le processus de création de message devrait mettre par défaut une seule adresse de machine authentifiée, indiquant l'**agent** (personne, système ou processus) qui a voulu envoyer le message. Si cela n'est pas fait, le champ expéditeur (sender) doit être présent. Si le champ "from" est mis par défaut, le champ "expéditeur" ("sender") est optionnel et est redondant avec le champ "From". Ainsi le champ «sender » a pour but d'être utilisé quand l'expéditeur n'est pas l'auteur du message ou pour indiquer parmi un groupe d'auteur expéditeur qui a réellement expédié le message.
Dans tous les cas, les adresses dans le champ "From" doivent être utilisables par la machine et ne doivent pas contenir des listes nommées (groupes).

4.2.4 Le champs « *replay to* »

Ce champ fournit un mécanisme global pour indiquer n'importe quelle boite aux lettres vers laquelle la réponse doit être envoyée. Trois utilisations classiques pour cette caractéristique peuvent être distinguées. Dans le premier cas, le(s) auteur(s) peuvent ne pas avoir de boite à lettre régulièrement basée sur un ordinateur, et qui veulent indiquer une adresse de machine alternative. Dans le second cas, un auteur veut que des personnes supplémentaires soient averties, ou soient responsables. Une utilisation quelque peu différente peut en être faite pour aider les groupes de téléconférence en mode texte équipés avec des services de télédistribution automatique : incluant l'adresse de ce service dans le champ "Reply-To" pour tous les messages soumis à la téléconférence ; Alors les participants peuvent "répondre" aux soumissions faites pendant la conférence pour garantir la distribution correcte de n'importe quelle soumission de leur part.

Remarque: Le champ "Return-Path" est ajouté par le service de transport de courrier, au moment de la distribution finale. Cela à pour but d'identifier le chemin de retour vers l'originaire du message. Le champ "Reply-To" est ajouté par l'originaire du message et s'attend à des réponses directes.

4.2.5 Le champ « *BCC (blind carbon copy)* »

Alors qu'en français cci (*copie conforme invisible*) ce champ contient l'identité des destinataires supplémentaires du message de telle sorte que si vous ajoutez le nom d'un destinataire au champ **Cci** dans un message électronique, une copie du message est envoyée au destinataire, mais le nom de celui-ci n'est pas visible pour les autres destinataires du message.

4.2.6 Identificateur de message « *Message-ID* »

Ce champ contient un identificateur unique (l'élément d'adresse de la partie locale) qui référence la version de ce message. L'unicité de l'identificateur du message est garantie par le système d'hôte qui le génère. Cet identificateur à pour but d'être lisible à la machine et non d'être nécessairement compréhensible pour les humains.
A un identificateur de message correspond uniquement une seule instance d'un message précis ; Les révisions successives du message devront chacune recevoir de nouveaux identificateurs de message.

5. Le cryptage

Parfois, la cryptographie est utilisée pour augmenter la confidentialité du contenu du message. Si le corps du message a été crypté, pour garder son contenu privé, le champ "Encrypted" peut être utilisé pour noter le fait que le message soit crypté, et pour indiquer la nature du cryptage. Le premier paramètre de type <word> indique le logiciel utilisé pour crypter le corps, et le second, optionnel de type <word> a pour intention d'aider le destinataire dans la sélection de la bonne clé de décryptage.

Remarque: Malheureusement, les en-têtes doivent contenir aussi bien l'enveloppe, que le contenu et l'information. Par conséquent, il est nécessaire qu'elles restent non cryptées, afin que le service de transport de courrier puisse y accéder. Puisque le contenu des champs de noms, d'adresses, et "Sujet" peut contenir de l'information sensible, cela limite la confidentialité totale du message.

6.6.1 Syntaxe — Spécifications de l'adresse

Address	Mailbox Group	Une adresse. Liste nommée.
Group	Phrase ":" [#mailbox] ";"	
mailbox	addresse-spec phrase route-addresse	Adresse simple. Nom et spécification d'adresse.
route-addr	"<" [route] addr-spec ">"	

route	1#("@" domain) ":"	Chemin relatif (path-relative).
addr-spec	local-part "@" domain	Adresse globale.
local-part	word *("." word)	Non interprétée. préservation de la case
sub-domain	domain-ref / domain-literal	
domain	sub-domain *("." sub-domain)	
sub-domain	domain-ref / domain-literal	
domain-ref	atom	Référence symbolique.

Table VII.3 spécification syntaxique de l'adresse.

6.2 Sémantique

Une boite aux lettres reçoit des courriers. C'est une entité conceptuelle qui n'est pas forcement en rapport avec le stockage de fichier. Par exemple, quelques sites peuvent choisir d'imprimer le courrier sur une imprimante en ligne et remettre la sortie au bureau du destinataire.
Une spécification de boite aux lettres comprend une référence à un nom de personne, de système ou de processus, une chaîne dépendant d'un domaine, et une référence à un nom de domaine. La référence au nom est optionnelle et on l'utilise généralement pour indiquer le nom humain de la personne destinataire. La référence au nom de domaine désigne une série de sous-domaines. La chaîne dépendant du domaine n'est pas interprétée, exception faite du sous domaine final ; le reste du service de courrier, la transmet simplement comme une chaîne littérale.

6.3 Les domaines

Un nom de domaine est un ensemble de noms répertoriés. Une définition de nom de domaine décide de la définition de nom de domaine subordonné ou d'une chaîne terminale dépendante d'un domaine. Par conséquent, la définition d'un domaine est extensible, permettant n'importe quels nombres de niveaux d'enregistrement.

Les noms de domaine modélisent le schéma hiérarchique, logique, et global d'adressage. Le modèle est logique, du fait que la définition d'adressage est relative au nom d'enregistrement et n'est pas obligatoirement liée au chemin de transmission. La hiérarchie du modèle est un graphe orienté, appelé arbre entrant (*in-tree*), de telle manière qu'il n'y ait qu'un seul chemin de la racine de cet arbre vers n'importe élément de la hiérarchie. Si plus d'un chemin existe réellement, on considère que l'on a affaire à des adresses différentes.
Le nœud racine est commun à toutes les adresses ; Par conséquent, il n'est pas référencé. Ces enfants constituent les noms de domaine de haut niveau. Habituellement, un service a accès à sa définition complète de nom de domaine et à tous les domaines de nom de niveau supérieur.
Le "sommet" de la hiérarchie d'adressage de domaine - un des fils de la racine - est indiqué par le champ situé le plus à gauche dans la définition d'un domaine. Ses enfants sont situés à gauche, leur fils à gauche, et ainsi de suite.
Certains groupes fournissent des services officiels d'enregistrements ; constituant des domaines de noms qui sont indépendant logiquement de machines spécifiques. En plus, les machines et les réseaux composent implicitement des domaines de noms, puisque leurs membres sont habituellement enregistrés dans des tables de noms. Dans le cas d'un

enregistrement officiel, une organisation implémente une base de données (distribuée) qui fournit un service de correspondance adresse-route pour les adresses qui sont de la forme :

person@registry.organization

Notez que "organization" est une entité logique, indépendante de tout réseau de communication.
Un mécanisme pour accéder à l'organization est disponible pour tout le monde. Ce mécanisme, en boucle, cherche une instance de l'enregistrement ; sa localisation n'est pas spécifiée sous forme d'adresse. On suppose que le système qui opère sous le nom "Organization" sait comment trouver un registre subordonné. Le registre utilisera alors la chaîne "person" pour déterminer ou envoyer la spécification du courrier. Ce dernier permet une spécification simple et directe d'adresse relative à un attachement, comme:
user@host.network

Une fois que le réseau est atteint, on s'attend à ce que le message ira directement vers l'hôte et que l'hôte résoudra le nom de l'utilisateur, le message est placé dans la boite aux lettres de l'utilisateur.

6.4 Spécifications d'abréviation de domaine

Puisqu'un nombre quelconque de niveaux est possible dans la hiérarchie d'un domaine, la spécification d'une adresse pleinement qualifiée, peut devenir un inconvénient. Ce standard permet des définitions de domaine mises sous forme abrégées pour le cas particulier :

Pour les adresses de l'expéditeur, on appelle Niveau N le sous domaine le plus à gauche. Dans une adresse d'en-tête, si tous les sous domaines situés au-dessus (c.à.d. à droite) de Niveau N sont les même que celle de l'expéditeur, alors ils n'ont pas à apparaître dans la spécification. Sinon, l'adresse doit être pleinement qualifiée (*fully qualified*).
Cette caractéristique est sujet d'approbation par les sous domaines locaux. Les sous domaines individuellement peuvent avoir besoin de leur système membre, qui donne l'origine du courrier, pour uniquement fournir une définition de domaine complète.
Quand c'est autorisé, les abréviations peuvent être présentes seulement quand le message reste à l'intérieur du sous domaine.
L'utilisation de ce mécanisme nécessite que le sous-domaine de l'expéditeur réserve les noms de tous les domaines de niveau supérieur, de cette façon ses définitions complètes peuvent être différenciées des définitions mise sous forme d'abréviations.

Par exemple, si l'adresse d'un expéditeur est :
expediteur@registry-A.registry-1.organization-X
et l'adresse d'un destinataire est :
destinataire@registry-B.registry-1.organization-X
et qu'un autre soit :
destinataire@registry-C.registry-2.organization-X

Alors ".registry-1.organization-X" n'a pas besoin d'être indiqué dans le message, mais "registry-C.registry-2" **doit** être indiqué. C'est à dire, les deux premières adresses peuvent être mises sous forme abrégée mais la troisième doit être entièrement définie.

Quand un message traverse une limite de domaine, toutes les adresses doivent être déclarées sous le format complet, en terminant par le nom de domaine de niveau supérieur sur le champ le plus à droite. C'est la responsabilité du service d'expédition de courrier de s'assurer que les adresses soient en conformité avec cette exigence. Dans le cas des adresses mises sous abréviation le serveur qui fait relais doit faire l'extension nécessaire. On doit noter que c'est souvent difficile pour de tel service de localiser toutes les occurrences des abréviations d'adresse. Par exemple, il sera impossible de trouver ces abréviations à l'intérieur du corps d'un message. Le champ de " chemin de retour " peut aider les destinataires dans la correction de ce type d'erreur.

Remarque: Quand on passe n'importe quelle portion d'une définition d'adresse à un processus qui n'interprète pas les données en accord avec ce standard (par exemple les serveurs de protocole courrier). Il ne doit pas y avoir de caractère LWSP précédant ou suivant le signe « at @» ou aucun point de délimitation ("."), tel que ce qui a été montré dans les exemples au-dessus, et seulement un espace entre les mots <word>: contiguës.

6.5 Les termes du domaine

Une référence de domaine doit être le nom officiel d'un registre, d'un réseau ou un hôte. C'est une référence symbolique, à l'intérieur d'un nom de sous domaine. A tout moment, il est nécessaire de passer outre les mécanismes du standard pour résoudre de telles références, en utilisant des informations plus primaires, tel que les adresses d'hôte de réseau plutôt que leur nom d'hôte associé.

Remarque: l'utilisation de littéral de domaine est **fortement découragée**. Cela est permit seulement comme un moyen de contourner temporairement les limitations système, tel que les tables de nom qui ne sont pas complètes.

La partie locale d'une spécification d'adresse dans une déclaration d'adresse (c.à.d. le nom d'hôte pour la boite aux lettres) est interprétée comme étant ce que le serveur receveur d'email autorise. Par exemple, certains systèmes ne comprennent pas les déclarations de boite aux lettres de la forme "P. D. Q. Bach", mais d'autres le font.
Cette norme traite les points (".") en tant que séparateurs lexicaux. En conséquence, leur présence dans les parties locales qui ne sont pas entre les chaînes mise entre guillemets, est détectée. Cependant de telles occurrences n'apportent aucune sémantique. Cela signifie, que si une partie locale avait des points dans cette partie, un analyseur d'adresse la diviserait en plusieurs marques syntaxiques, mais la séquence de ces marques serait traitée comme une seule unité ininterrompue. La séquence devra être réassemblée, quand l'adresse sera passée hors du système comme celui du service de protocole de mail.

Dans certains cas, la frontière entre la partie locale et le domaine peut être flexible. La partie locale peut être une simple chaîne de caractère, qui est utilisé pour déterminer la boite aux lettres du destinataire final. Tous les autres niveaux font par conséquent, partie du domaine.

6.6 Boite aux lettres multiples

Un individu peut avoir plusieurs boites aux lettres et aimerait recevoir du courrier dans la boite aux lettres qui serait la plus pratique à accéder pour l'expéditeur. Et par conséquent un

ensemble d'individu peut vouloir recevoir un courrier comme une seule unité (c.à.d. une liste de distribution). Le constructeur de <group> permet la spécification à la manière d'une liste. Les boites aux lettres des destinataires sont spécifiées à l'intérieur de la partie entourée de parenthèse (bracket). Une copie du message est envoyée à chacune des boites aux lettres listées. Le standard ne permet pas des spécifications récursives de groupe à l'intérieur de groupes.

Certains services de courrier peuvent fournir un mécanisme de distribution par groupe-liste, acceptant une simple référence de boite aux lettres, en développant une liste de distribution complète, et en relayant le courrier aux membres de la liste. L'utilisation d'adresse alternative <group>, qui contient à l'intérieur, une boite aux lettres, peut signifier que, soit la référence de la boite aux lettres sera développée en une liste, soit qu'il y a un groupe avec un seul membre.

6.7 Spécification explicite du chemin

Quelquefois, une personne à l'origine d'un message doit pouvoir indiquer quel chemin de transmission, le message doit suivre. On appelle cela le **routage à la source**. Le schéma normal d'adressage, utilisé dans la spécification d'adresse, est dissocie par prudence de ces informations. On spécifie la séquence de serveurs hôte et/ou les services de transmission qui sont traversés. A la fois, les références de domaines et les littéraux de domaine peuvent être utilisées.

Remarque: L'utilisation du routage à la source est découragée. A moins que l'expéditeur a un besoin spécifique pour des restrictions de route, le choix de la route de transmission devra être laissé au service de transport de courrier.

6.8 Adresses réservées

Il est souvent nécessaire d'envoyer un courrier vers un site, sans connaître aucunes adresses valides. Par exemple, il y aura peut être des dysfonctionnements du système de courrier quand un utilisateur veut connaître l'adresse correcte d'une personne sur ce site.

Des standards spécifient une simple adresse, adresse réservée de boite aux lettres (*partie locale*) qui doit être valide pour chaque site. Le courrier envoyé à cette adresse doit être routé vers une personne responsable pour le système du courrier. Le nom de la partie locale de l'adresse est : **Postmaster**, ainsi on doit avoir "Postmaster@domain" pour que ce soit valide.

7. Conclusion

La simplicité apparente de la manipulation du message est due à l'établissement de différents formats pour mieux adapter le plus efficace, cette approche a tendance de respecter les standards établis par les plus grandes organisations pour minimiser au maximum les incohérences de communications.

CHAPITRE VIII.

Conception de B2SAM.

En arrivant là; là se trouve l'embarras.
– William Shakespeare.

1. Introduction

1.1 UML

Dans ce chapitre, nous utilisons UML (*Unified Modeling Language)* pour décrire la conception de notre système HoneyPot et pour cela nous avons utilisé le modèle en V.

Le modèle en V demeure actuellement le cycle de vie le plus connu et certainement le plus utilisé. Le principe de ce modèle est qu'avec toute décomposition doit être décrite la recomposition, et que toute description d'un composant est accompagnée de tests qui permettront de s'assurer qu'il correspond à sa description. Ceci rend explicite la préparation des dernières phases (validation-vérification) par les premières (Construction du logiciel), et permet ainsi d'éviter un écueil bien connu de la spécification du logiciel :

« Énoncer une propriété qu'il est impossible de vérifier objectivement après la réalisation ». **[UML-31]**

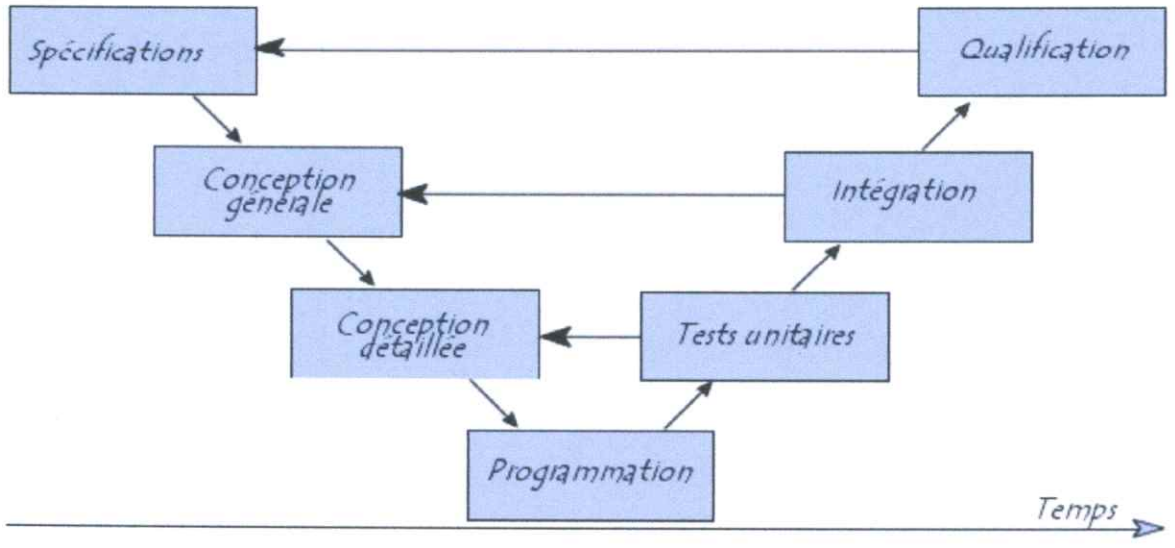


Fig. VIII.1 Model de cycle de vie d'un logiciel.

La notation UML que nous avons utilisée est née de la fusion des trois méthodes qui ont les plus influencées de la modélisation objet au milieu des années 90 : OMT, Booch et OOSE. De ce fait, elle permet de couvrir le cycle de vie d'un logiciel depuis l'analyse du besoin jusqu'au codage. Cette notation est d'une très grande richesse. Elle permet de couvrir à peu près toutes les phases du développement.

Parmi les avantages d'UML:

- Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions.
- L'aspect formel et normalisé de sa notation, limite les ambiguïtés et les incompréhensions.
- Son indépendance par rapport aux langages de programmation, aux domaines d'application et aux processus, en font un langage universel.
  [UMF-32]

1.2 UP (*Unified process*)

Le processus unifié a été élaboré par Jacobson il a particulièrement été bien adapté pour la mise en œuvre d'un processus de développement basé sur la notation UML.
Une réalisation conforme à UP, pour transformer les besoins des utilisateurs en logiciel, doit nécessairement présenter les caractéristiques suivantes :

- UP est à base de composants.
- UP utilise UML (*ensemble d'outils et de diagramme*).
- UP est piloté par les cas d'utilisation.
- UP est centré sur l'architecture.
- UP est itératif et incrémental.

1.2.1 Phase de Réalisation d'UP

- **Phase de Création**

Traduit une idée en vision de produit fini et présente une étude de rentabilité pour ce produit.
Elle répond aux questions suivantes :

- que va faire le système ?
- quelle va être l'architecture générale (cible) de ce système.
- quels vont être : les délais, les coûts, les ressources, les moyens à déployer?

- **Phase d'élaboration**

Permet de préciser la plupart des cas d'utilisation et de concevoir l'architecture du Système.
Les taches à effectuer dans la phase élaboration sont les suivantes :

- Créer une architecture de référence.
- Identifier les risques, ceux qui sont de nature à bouleverser le plan, le coût et le calendrier
- Définir les niveaux de qualité à atteindre
- Élaborer une offre abordant les questions de calendrier, de personnel et de budget

- **Phase de construction**

L'architecture de référence se métamorphose en produit complet, elle est maintenue Stable.

- **Phase de transition**

Un groupe d'utilisateurs essaye le produit et détecte les anomalies et défauts.

2. Présentation de l'application

Apres une étude approfondie sur les Spams et les technique de spamming ainsi que les HoneyPot et même les différentes techniques d'attaques sur les réseaux et systèmes d'exploitations on a essayé de mettre au point un système anti-spamming utilisant des ressources distantes, ainsi que différentes techniques de détection de spam pour préserver au mieux l'aspect objectif du mailing; et pour cela on a proposé l'architecture suivante (*Fig VIII.2*) toute en utilisant des logiciels open source définit au préalable.

Ainsi l'architecture de notre système est composée d'éléments suivants :

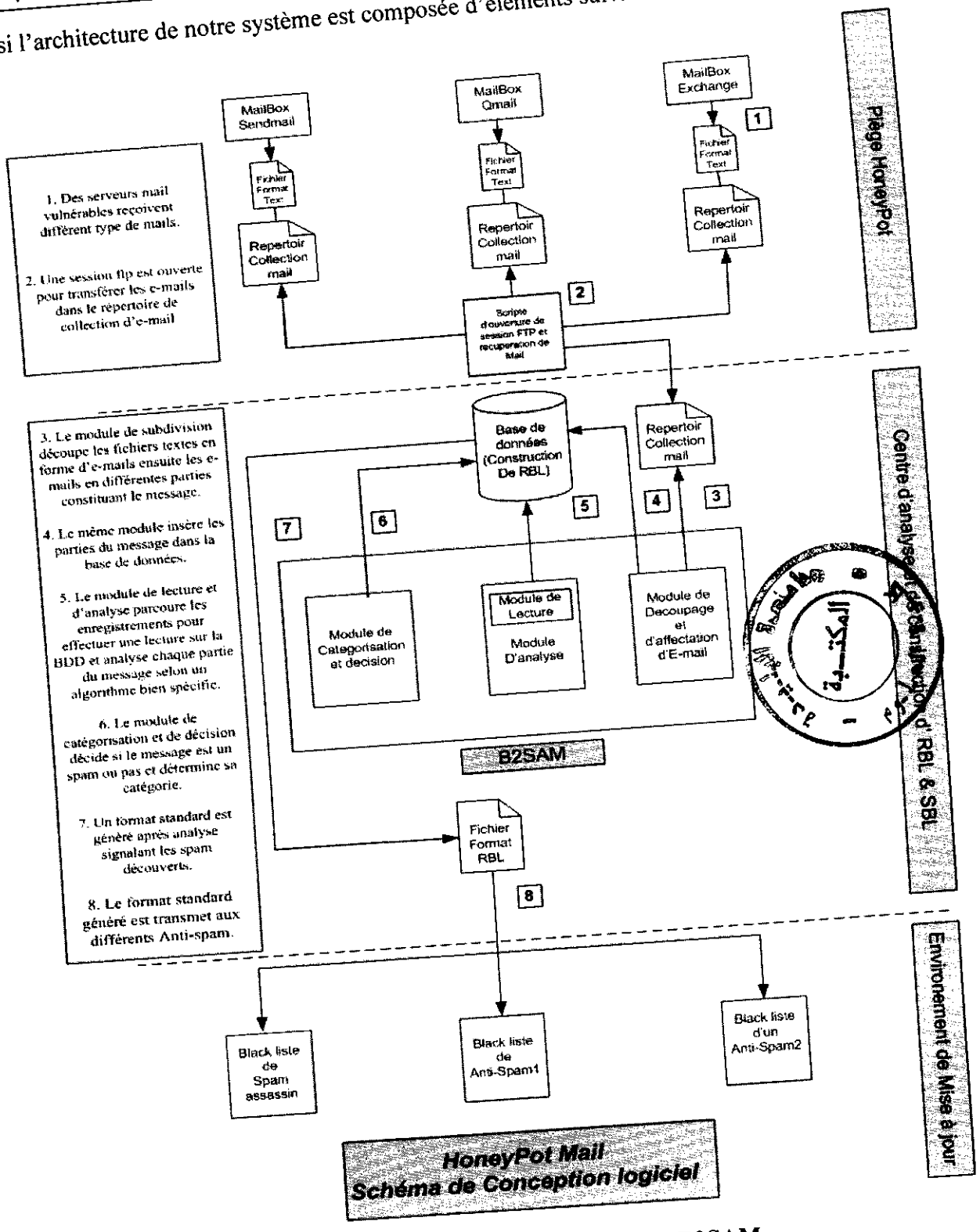


Fig. VIII.2 Architecture du système B2SAM.

2.1 La 1ère partie : « piège HoneyPot »

- **Serveurs mails virtuels volontairement vulnérables**

Dans le but de collecter le plus maximum possible d'e-mails et d'attirer les spammeurs qui cherche à bousier les serveurs mails public ou bien utiliser le mailing à des fins négatives on a utilisé plusieurs serveurs mail volontairement vulnérables et à une grande tendance d'importance, aussi on a utilisé plusieurs systèmes d'exploitation avec divers Services mails tel que LINUX RED HAT avec « Sendmail & Qmail »et WINDOWS SERVEUR 2003 avec « Exchange ».
Pour notre architecture proposée on a adopté « SendMail » sous linux Redhat Entreprise Edition(V4) pour les raison bien expliqués dans le chapitre « outils ».
Et ainsi pour définir le point de vulnérabilité on a utilisé différentes méthodes pour que nos domaines soient volontairement connus par les internautes et plus précisément les spammeurs en s'enregistrant aux forums de hacking de publicité de commerce...Etc.

- **Module de récupération d'e-mails**

Ce module fait partie de notre système d'analyse « B2SAM » qui sera détaillé ultérieurement, en utilisant un script d'ouverture de session ftp qui est programmé comme étant une tache planifiée qui s'exécute chaque semaine, il essaye de se connecter à notre système d'analyse qui contient déjà un server ftp installé avec le pack Microsoft appelé (Internet Information Services-File Transfer Protocol Service), puis après avoir ouvert la session ftp il récupère les E-mail à partir des boites aux lettres de différents serveurs mails qu'on a mis au point sous format texte, et les envois directement au répertoire qui se trouve sur le B2SAM central, cela rend les e-mails locaux pour le module d'analyse de B2SAM.

Note : Les e-mails récupérés sont sous format texte, mais il est à noter que **tous** les e-mails reçus pour un utilisateur donné sont regroupés dans un seul fichier texte pour éviter les conflits de stockage et de transfère ainsi que de consultation et de manipulation; cette technique est utilisée par défaut dans tous les systèmes d'exploitation basés sur UNIX et elle est ainsi préserver par « SendMail ».

Le fait d'automatiser cette procédure de collecte de mail, l'ouverture de session ftp et de transfère de données rend notre système plus autonome (car l'analyse est séparée de la collecte d'e-mails) d'une part, d'autre part si on décide un jour de mettre un nouveau serveur de collecte ou d'ajouter un nouveau centre de collecte on pas besoin de configuration manuel, il suffit juste d'exécuter le script une seul fois -même si on n'est pas des spécialistes- et le script fonctionnera sans aucune intervention de l'utilisateur. Ce procéder définit un point d'extensibilité très important tant au niveau de collecte d'informations qu'au niveau de compatibilité.

2.2 La 2ème partie : « B2SAM CENTRAL »

C'est le noyau de notre système ; et il a été conçu pour répondre au grand besoin de sécurisé les serveurs mail et bien respecter la notion du mailing dans un sens constructif.

Il est constitué des modules suivants :

1. Module de lecture du fichier texte, de décomposition et d'écriture sur la base de données

Apres la collecte d'e-mails et le transfert de ces derniers par une session ftp, ils deviennent locaux à notre système d'analyse, et ils se situeront exactement dans le répertoire dédié pour le stockage appelé « Rep_Centr » ; c'est la qu'intervient ce module qui doit comme première étape charger les fichiers textes ainsi transférés à partir du répertoire de stockage pour les rendre prét pour l'analyse, ensuite dans la deuxième phase il doit les décomposer localement en différents mails séparés, puis il procède a la décomposition dans la troisième étape de ces mails en différentes parties constituant le mail et les insère au fur et à mesure dans les différentes champs de la table d'analyse temporaire de la base de données, cette table qui est appelée « Mail_decomp » sera utilisée lors de l'analyse par les différents modules d'analyse qui utilisent des champs déjà existant de cette table pour remplir les autres champs de l'analyse.

Ces champs obtenus sont les composants principaux du message à qui sont appliquer les critères d'analyse et de catégorisation, et qui ont été bien détaillés dans le chapitre traitant le format du message.

Les parties des messages qui vont êtres utilisées pour l'analyse et la catégorisation sont les suivants :

1. L'adresse IP de l'expéditeur.
2. Le domaine à qui appartient l'expéditeur.
3. Le sujet du message.
4. La date de l'envoi du message.
5. Le corps du message.

Apres l'extraction de ces parties le module écrit les informations estradiers un par un dans la base de données temporaire afin de les analyser avec le module d'analyse.
On a utilisé seulement ces parties parmi tout le jeu de composants que contient le message parce que sont les parties les plus significatives dans le message et à partir de ces composants que la plus grande partie des formats standards de mise à jour pour les anti-spams sont générés, car ils présentent l'identité du message, l'identité de l'expéditeur et même le domaines à qui appartient l'expéditeur.

2. Module d'analyse

Afin de détecter qu'un e-mail est un spam ou pas, ce module utilise plusieurs méthodes d'analyse combinées décrites ci-dessous pour pouvoir porter un jugement correcte sur l'e-mail en question.

2.1 Analyse de l'adresse IP de l'expéditeur

Le module d'analyse fait un test comparatif de l'adresse IP de l'expéditeur de chaque message contre une liste d'adresses IP définit au préalable et stockée dans la base de donnée locale de l'analyse plus précisément dans la table « IP_suspect » qui contient plus de 11000 adresses IP signalées comme suspectes par les fournisseurs d'internet les plus connus au monde. Cette liste d'adresse est tirée de la dernière version (Mai 2007) des adresses IP suspectes reconnues mondialement et utilisées par le projet Spamhaus sur le site (*www.spamhaus.prg*), et sera mise à jour automatiquement après la détection de nouveau spam par d'autre modules tel que analyse sur le sujet, les mots suspects etc...., et pour préserver l'efficacité et l'informité par rapport aux standards universels on a utilisé le format SBL utilisées par les grands centres d'observation et de détection de spam à travers le monde (des statistiques furent établît par le projet « Spamhaus » reportent que plus de 85% des format standards de mise à jour sont sous format SBL, 15% sous format RBL et le reste sous plusieurs formats définit par d'autres organismes). **[ANNEXE Spamhaus]**.

Ce test comparatif à la base de fonctionnement de ce module signale si l'adresse du message fait partie de la liste des adresses IP suspectes ou pas pour attribuer un score précis au message en cours.

2.2 Analyse du domaine de l'expéditeur

En utilisant le même principe de l'analyse de l'adresse IP mais cette fois ci en utilisant une table prés définit appelée « Domaine_susp » qui contient les adresse IP des domaines suspects et les domaines correspondants en jouant le rôle d'un serveur DNS, cette liste qui a été extraite du projet Spamhaus, est considérée comme la version la plus ressente (on à utilisé une liste de plus de 4000 domaines-IP suspects), cette dernière sera met à jour automatiquement après la détection de spam par d'autre module d'analyse tout comme le premier module d'analyse d'adresse IP, le module d'analyse utilise cette liste pour la détection des domaines blacklistés en les comparant avec listes qui contiennent bien évidemment les domaines considérés comme étant potentiellement source de spam ou tout simplement blacklistés.

A l'essor de ce module un score est généré selon l'appartenance ou non du message à un domaine suspect.

2.3 Analyse du sujet

Dans cette étape le module d'analyse vérifie le sujet du message s'il respecte les spécifications standards dictées par les organismes de définition du format standard du mail et ces différentes parties notamment les RFC, et en particulier la RFC822; donc on adopté les spécifications suivantes :

- Le sujet d'un message ne doit pas être vide.
- Ne doit pas commencer par deux chiffres.
- Ne doit pas contenir que des chiffres.
- Le sujet ne doit pas commencer par au moins deux tirets (underscore).
- Le sujet ne doit pas commencer par au moins trois points.
- Ne doit pas contenir un mot suspect ou une phrase suspecte.

2.4 Analyse de la date d'envoie du message

Le module d'analyse parcourt séquentiellement tous les enregistrements puis, s'il trouve des messages envoyés par le même émetteur il calcule le laps de temps séparant les dates d'envoi des messages ; bien entendus cette analyse est basée sur un seuil c.-à-d. si la différence entre les deux date est inferieur ou égale au seuil, l'envoi sera considéré comme une attaque et l'émetteur est hautement qualifie d'être un spammeur.

Exemple : 1- expéditeur : David date : 06/15/2007 04:30 :22 PM.
2- expéditeur : David date : 06/15/2007 04:31 :11 PM.
3- expéditeur : David date : 06/15/2007 04:32 :10 PM.
4- expéditeur : David date : 06/15/2007 04:32 :50 PM.

Si par exemple le seuil est fixer à 3 minute, et le nombre de message est fixer à 4, donc après une comparaison sur l'adresse et sur l'expéditeur ainsi que sur la date et le temps on considère David comme spammeur ; et son adresse est ajoutée à la liste d'adresses suspectes sur la base de données, son domaine est ajouté à la liste des domaines suspects sur la base de données, et même le sujet de son message est ajouté à la liste des mots suspects sur la base de données. Mais ces mises à jour ne sont faites bien sure que si le score obtenu par cet e-mail atteint le score qualifiant l'e-mail comme étant un éventuel spam.

2.5 Analyse du corps

C'est la partie la plus complexe et la plus délicate car malgré toutes les techniques d'analyse même les plus avancées utilisant les algorithmes mathématiques les plus compliqués comme celle des filtres Bayaisien on arrive pas à des ajustements rigoureux de détection de spam, car il ne s'agit pas seulement d'un jeu de mots déterminé auparavant pour en arriver à classer un tel message, ce qui explique bien la délicatesse de cette tache, par exemple une société pharmaceutique, lorsqu'elle manipule le courrier électronique doit obligatoirement utiliser des mots qui sont bien considérés par la plupart des anti-spam comme des mots suspects comme par exemple les noms des produits (viagra, gélule,..Etc.) Et par la suite la plus grande partie des e-mails de cette société seront considérés comme des Spams. **[AOS-33]**

Cela dit, la méthode qu'on a adoptée pour ce module est la suivante :

Le module d'analyse parcourt tous l'e-mail (le corps de l'e-mail) un par un pour extraire les mots suspicieux en utilisant une liste de mots suspects déjà existante cette liste est construite en utilisant plusieurs méthodes de collecte de mots suspects :

- la collecte des mots suspects s'est faite en utilisant des listes définies par la plupart des anti-Spam en essayant de garder une vue logique et cohérente par rapport à toutes les listes qu'on a utilisé.
- Une analyse manuel de plusieurs Spam reçus sur des boites aux lettres utilisées spécialement pour collecté des spam et pour voir les différents formats et types (adresse1@hotmail.com, adresse2@usthb.dz.com, adresse3@gmail.com...etc. plus de 2000 spam reçu).
- Après une étude prolongée sur les techniques utilisées par les spammeurs pour échapper aux différents filtres anti-spams, on a remplacé quelques lettres par des caractères que les spammeurs utilisent par exemple : « a par @ », « i par | », « g par 6 », « s par 5 », « l'ajout des espace entre les lettre dans quelques mots ».
- Faire toutes les combinaisons possibles de ces remplacements pour avoir une liste exhaustive de tout ce qui pourra être utilisé pour dévier aux filtres anti-spams.

Cette analyse du corps basée sur les mots suspects est la première étape, dans la seconde phase on a utilisé une technique presque jamais utilisée jusqu'à présent cette technique consiste à l'analyse basée sur des **phrases** ; c.-à-d. que le module d'analyse du corps parcourt tout le contenu de chaque message en recherchant l'occurrence d'une phrase suspecte définie au préalable dans une table appelée « Phrase_susp », l'analyse basée sur des phrases est plus significative et plus efficace car c'est une comparaison basée sur le contexte et sur une certaine sémantique et non seulement sur de simples mots.

Le seul souci qui reste c'est la construction d'une table des phrases suspectes, ce qui n'est pas toujours évident, car de simples phrases utilisées dans un contexte peuvent ne pas avoir le même

sens dans un autre contexte ce qui rend la tache plus délicate et plus sensible aux variations du contexte. C'est pour cette raison qu'on a utilisé les deux méthodes (analyse par mot et par phrase) en même temps et combinées entre eux de telle sorte quelles soient les plus efficaces possible, et toute en jouant sur l'attribution des scores pour mieux garder la cohérence entre les deux méthodes car les phrases suspectes peuvent contenir des mots suspects qui vont êtres détectés seulement par l'analyse basée sur les mots et par la suite l'analyse basée sur les phrases se révèle inutile tandis que l'analyse basée sur les phrases tient compte de la **combinaison** des mots et donc du contexte ce qui rend l'analyse basée sur seulement des mots peut promotrice. Ce qui explique bien la nécessité de combiner les deux méthodes soigneusement pour en profiter des avantages de chacune d'entre elles.

Note

- toutes les tables qu'on a utilisé (table des adresses IP suspect; table des domaines suspects et ainsi que les deux tables des mots et des phrases suspectes) sont susceptibles de variations et de mise à jour soit manuellement par l'administrateur utilisant notre solution ou bien automatiquement par de nouvelles tables récupérées auprès des organisations professionnelles et des sites de recherche travaillant sur des thèmes similaires. Ceci est un point d'**extensibilité** très important car la séparation entre les modules d'analyse et les tables utilisées dans l'analyse permet d'utiliser et d'ajouter de nouvelles tables (table des adresses IP suspect ou bien table des domaines suspects...) sans se soucier du fonctionnement interne des modules ce qui rend une telle interaction plus flexible.

- Lors de la collecte de mots suspects et les phrases suspectes on a tenu compte du rapport publie par la CNIL (Commission Nationale de l'Informatique et des Libertés) :

Typologie des spams en fonction de la langue utilisée	
Spams de langue anglaise	84,8 %
Spams de langues asiatiques	8 %
Spams de langue française	7 %

Table VIII.1 pourcentage des Spams en fonction de la langue utilisée.

Ainsi on s'appuyant sur ces pourcentages notre collecte de mots suspects ainsi que les phrases suspectes est basée sur la langue anglaise beaucoup plus que d'autre langue, et on a ainsi utilisé la langue française et l'allemand pour avoir le pourcentage le plus élevé des mots et phrases et rendre l'analyse plus rigoureuse.

3. Attribution du score

Un score est donné après chaque phase d'analyse et plus exactement par chaque module d'analyse portant sur une partie bien définie, il est ajouté au score de la phase précédente pour

avoir le score final calculé par l'addition de différents scores issus de chaque module d'analyse du message.

En réglant le score donné par chaque phase d'analyse en peut assez bien combler les lacunes que chaque phase peut générer, pour y remédier on a utilisé des statistiques proposées par le projet APACHE qui nous donne les rapports les plus appropries du scorage de chaque module, voici les formules qu'on a utilisé pour déterminer les score avec exactitude :

- L'adresse IP représente 11.1% de Spams reçu.
- Le nom du domaine représente 22.19% de Spams reçu.
- Le sujet ne représente que 6.66%.
- La date représente 15.55%.
- Les mots suspects représentent 11.01%.
- Les phrases suspectes représentent 33.33%.

Note : L'administrateur de notre solution peut changer la configuration par défaut et régler les scores comme bon lui semble selon les besoins ou bien l'orientation du domaine d'études désiré.
En basant sur les statistiques données ci-dessus on a établir la stratégie de scorage suivante qui est la stratégie par défaut :

- Module d'analyse de l'adresse IP : 5 points.
- Module d'analyse du domaine : 10 points.
- Module d'analyse du sujet : 3 points
- Module d'analyse de la date : 7 points
- Module d'analyse du corps basée sur les mots : 5 points.
- Module d'analyse du corps basée sur les phrases : 15 points.

Apres avoir calculé les pourcentages appropries attribuer à chaque module et les points que chaque module peut donner on a décidé de fixer le seuil de signalisation de spam à **10** points (d'être dans le milieu des scores (max+min)/2).
Et par conséquent on a critères suivantes:

- Un message ayant un score inferieur à 10 est considéré comme non spam.
- Un message ayant un score supérieur ou égal à 10 est considéré comme spam.

Un message qui provient d'un domaine suspect ou bien contient une phrase suspecte doit être considéré comme étant un spam, et bien effectivement il est car le score est supérieur ou égale à 10. Tandis qu'un message qui contient un mot suspect ou bien le sujet est suspect ne doit pas être qualifier directement spam, il faut vérifier les autres champs pour décider; d'où provient la délicatesse d'attribution du score.

4. Module de catégorisation

Comme son nom l'indique il répertorier les Spams détectés en différentes catégories en remplissant la case « catégorie de spam » dans la base de données temporaire. Cette catégorisation nous permet d'avoir des idées claires sur les comportements des spammeurs ainsi que les différents domaines qu'ils utilisent pour piéger les internautes, elle est donnée sous forme d'un commentaire, par exemple : publicité, loterie, santé, voyance ...etc.

La catégorisation permet de détecter les nouveaux centres d'intérêts des spammeurs pour mieux les anticiper et mieux régler la configuration des anti-Spams pour qu'ils puissent êtres à jour en terme de nouvelles techniques de spamming ; et on peut aussi lui en servir aussi comme un argument de statistique.

5. Base de données temporaire

Cette base de données stocke les différentes parties du message après décomposition, ainsi que le score final et la catégorie du message; donc elle doit contenir toutes les informations nécessaires pour l'analyse du message aussi bien que les informations pour les formats standards générés(les outputs) après analyse, elle contient les colonnes suivantes :

- Le numéro séquentiel d'entrée dans la table.
- L'adresse IP expéditeur.
- L'adresse du domaine de l'expéditeur.
- Le sujet du message.
- Le corps du message.
- La date du message.
- La catégorie du message.
- Le score du message.
- Type du message (SPAM, Non spam).

6. Module de mise-à-jour

Ce module en récupérant les informations nécessaires à partir de la base de données temporaire pour générer un format standard c-à-d accepté par la plus part des anti-Spams, se connecte aux base de données de ces derniers pour les mettre à jour ou bien tout simplement dépose la liste de la mise à jour dans les base de données distribuées afin que les anti-Spams puissent l'avoir lorsqu'ils font les mise-à-jour.

3. Les cas d'utilisations

3.1 Définition des Acteurs

ACTEUR	Catégorie	Définition
Expéditeur	(externe)	➢ Envoi des E-mail
Destinataire	(interne)	➢ Réception des E-mail
Serveur Mail	(interne)	➢ Serveur mail vulnérable
Rep_Dist	(interne)	➢ Répertoire ou les E-mail sont stockés par le serveur mail (fichier texte)
Sys_Exp_loc	(interne)	➢ Le system ou toutes les analyses sont faite
Sys_Exp_dist	(interne)	➢ Le system ou il y a u serveur mail vulnérable
Rep_centr	(interne)	➢ Liste des domaines suspects (une table dans une base de données)
Domaine_susp	(interne)	➢ Liste des Domaines suspects (une table dans une base de données)
Mot_susp	(interne)	➢ Liste des mots suspects (une table dans une base de données)
IP_susp	(interne)	➢ Liste des IP suspectes (une table dans une base de données)
Anti-spam	(externe)	➢ Spamme Assassin (l'anti-spam ou on va tester notre RBL)

Table VIII.2 Définition des acteurs.

3.2 Les diagrammes de cas d'utilisation

3.2.1 Envoi d'E-mail

ACTEUR	CAS D'UTILISATION
Expéditeur	➢ Envoi un E-mail : - L'E-mail peut être un spam - L'E-mail peut être un Non spam
Destinataire	➢ Reçoit un E-mail : - L'E-mail peut être un spam - L'E-mail peut être un Non spam

Table VIII.3 Rôle des acteurs d'envoi d'E-mail.

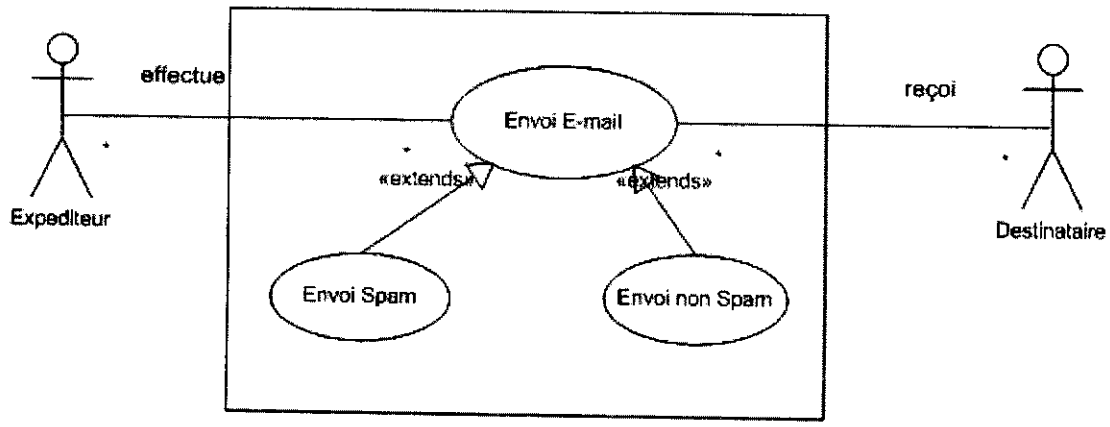


Fig. VIII.3 Cas d'utilisation d'envoi d'E-mail.

3.2.2 Stockage des E-mail

ACTEUR	CAS D'UTILISATION
Serveur Mail	➢ Apres la réception des E-mail il les stocks dans un répertoire local sous format de fichier Texte.
Répertoire Distant	➢ les E-mail sont stockés dans ce répertoire en format Texte.

Table VIII.4 Rôle des acteurs de stockage des E-mail.

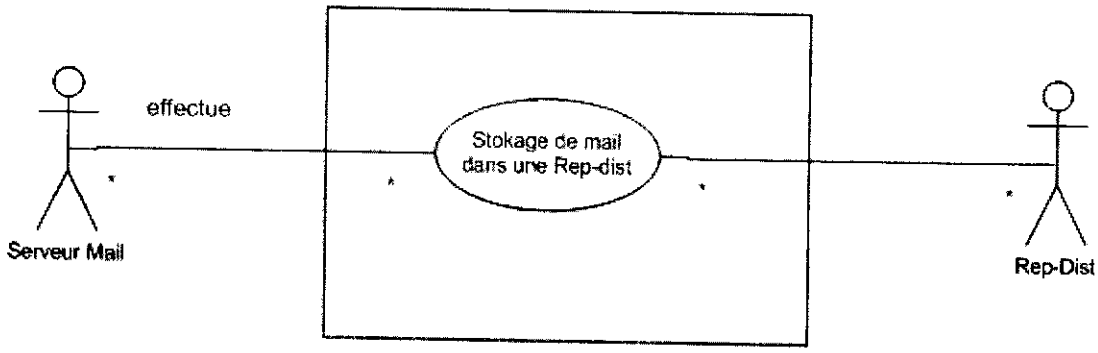


Fig. VIII.4 Cas d'utilisation de stockage des E-mail dans un Répertoire.

3.2.3 Ouverture d'une session ftp

ACTEUR	CAS D'UTILISATION
Système d'exploitation local	➢ Une session FTP est ouverte sur le serveur.
Système D'exploitation distant	➢ Ouvre une session FTP sur le système d'exploitation local. ➢ Procède à un dépôt de fichier sur le Répertoire central.
Répertoire Central	➢ Les fichiers sont stockés dans ce répertoire.

Table VIII.5 Rôle des acteurs d'ouverture d'une session FTP.

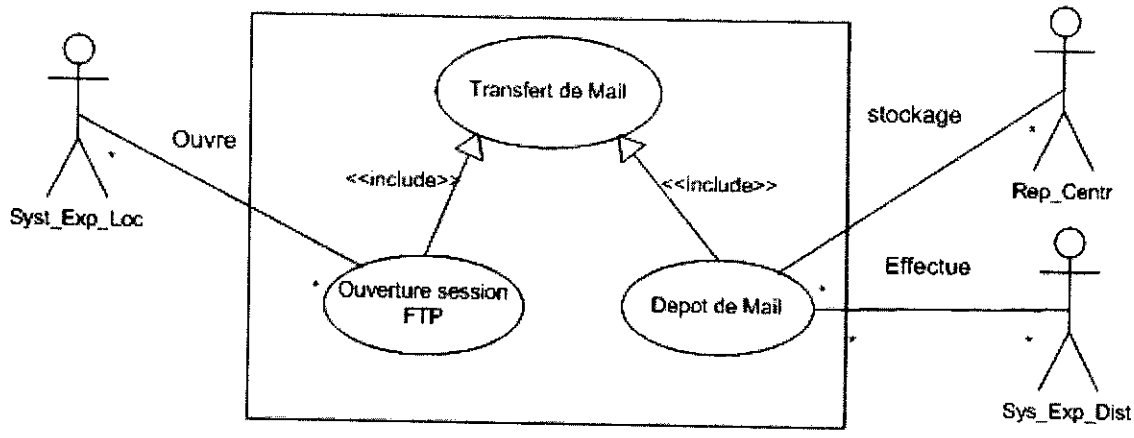


Fig. VIII.5 Cas d'utilisation d'ouverture de session FTP et Transfert d'E-mail.

3.2.4 Affectation dans la base de données

ACTEUR	CAS D'UTILISATION
Répertoire Central	➢ Une lecture est effectue sur le Répertoire central. ➢ Les fichiers inclues dans le répertoire sont découpés par le module de découpage. ➢ Apres chaque extraction d'une partie de L'E-mail il est affecté dans une table de la base de données

Table VIII.6 Rôle des acteurs d'affectation dans la base de données.

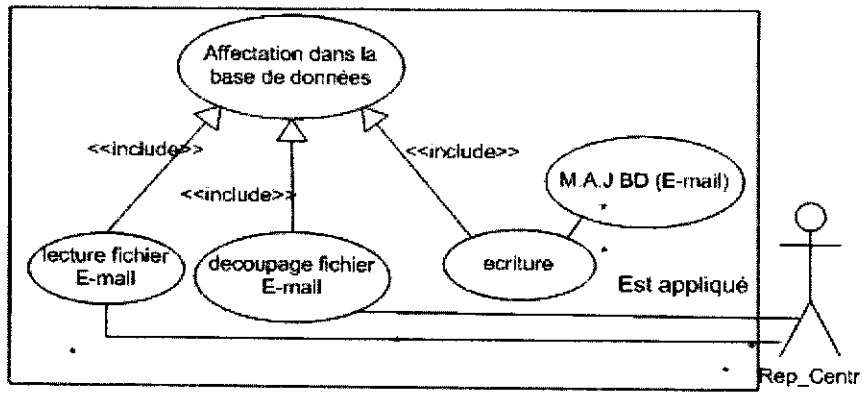


Fig. VIII.6 Cas d'utilisation d'affectation dans la base de données.

3.2.5 Analyse des E-mail

ACTEUR	CAS D'UTILISATION
Mot_susp	➢ Quand l'analyse et lancer le module « analyse Mot » consulte la base de données plus précisément la table des mots suspects.
Domaine_susp	➢ Quand l'analyse et lancer le module « analyse domaine » consulte la base de données plus précisément la table des domaines suspects.
IP_susp	➢ Quand l'analyse et lancer le module « analyse IP » consulte la base de données plus précisément la table des adresses IP suspects.

Table VIII.7 Rôle des acteurs d'analyse des E-mail.

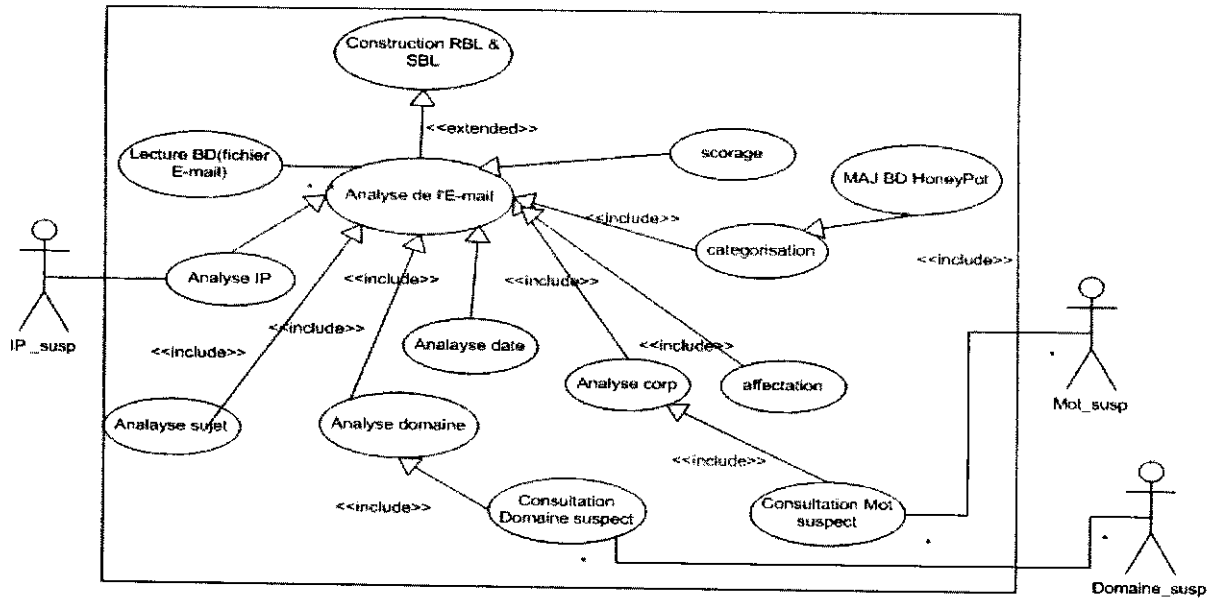


Fig. VIII.7 Cas d'utilisation d'analyse des E-mail.

3.2.6 Mise à jour des Anti-spam :

ACTEUR	CAS D'UTILISATION
Système d'exploitation distant	➢ Ouvre une session sur la base de données du répertoire central
Anti-spam	➢ Récupère le fichier en format RBL ou SBL on consultant la base de données.

Table VIII.8 Rôle des acteurs de mise à jour de l'Anti-spam.

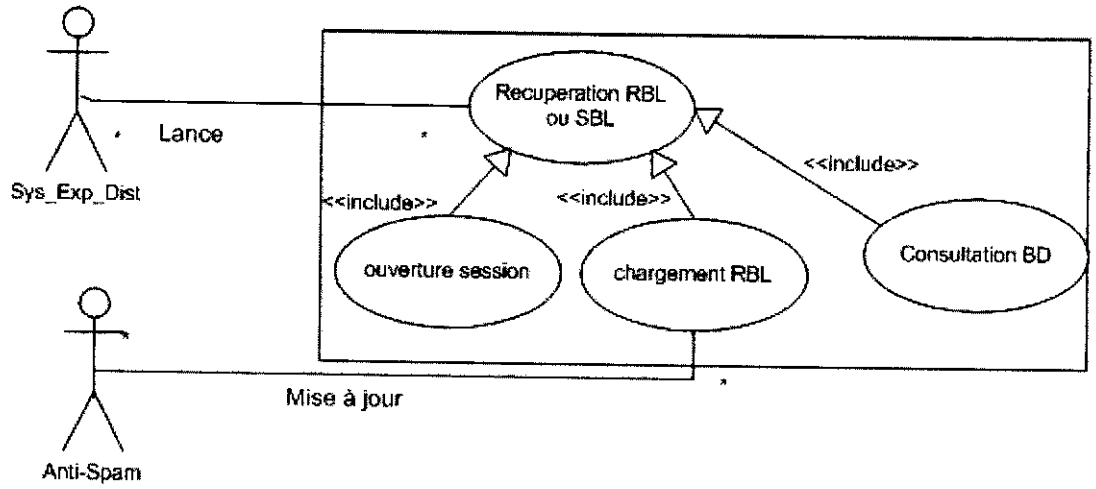


Fig. VIII.8 cas d'utilisation de mise à jour des anti-spam.

3.3 Diagramme de cas d'utilisation global

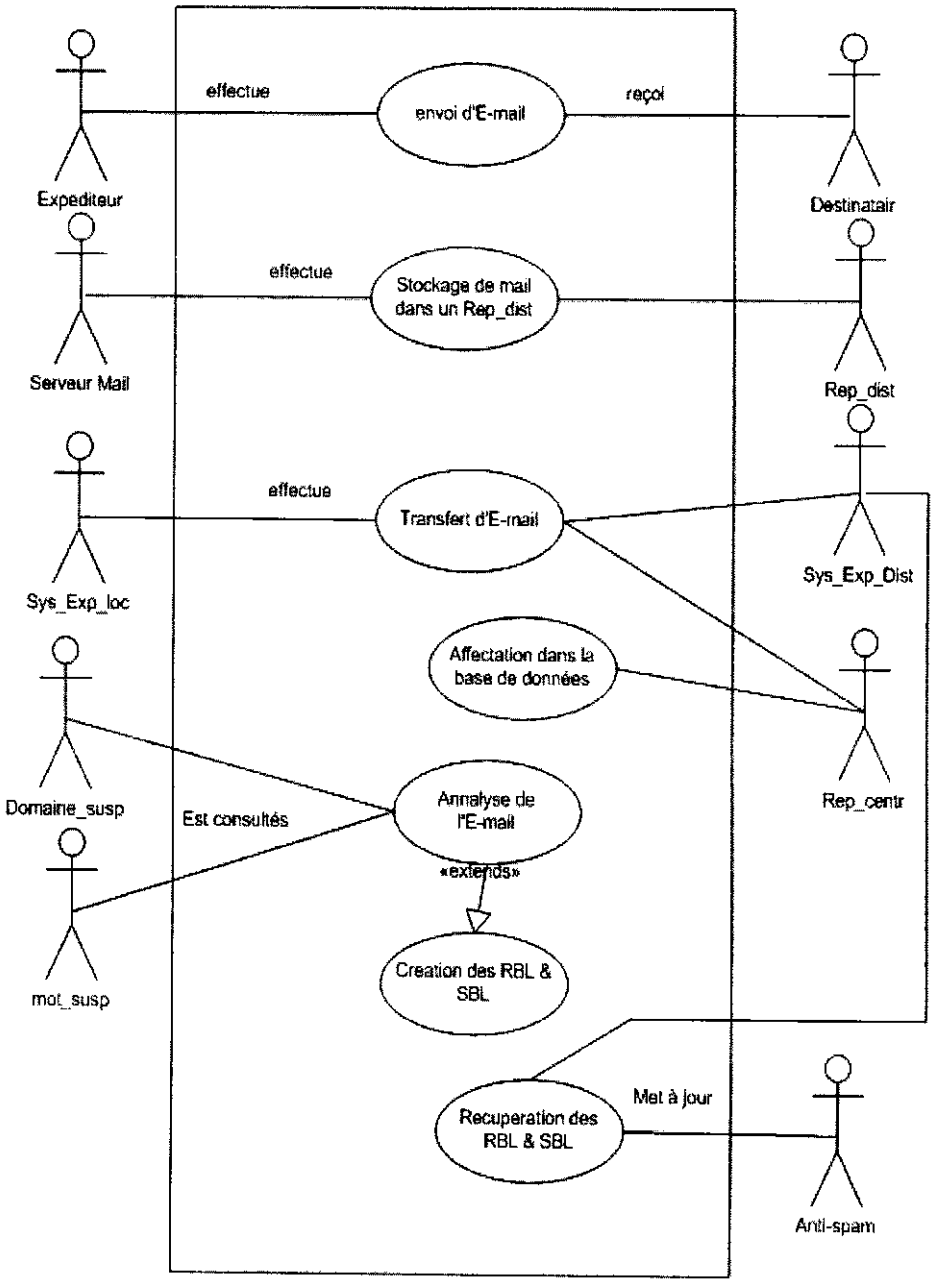


Fig. VIII.9 Diagramme de cas d'utilisation global.

3.4 Digramme de transition

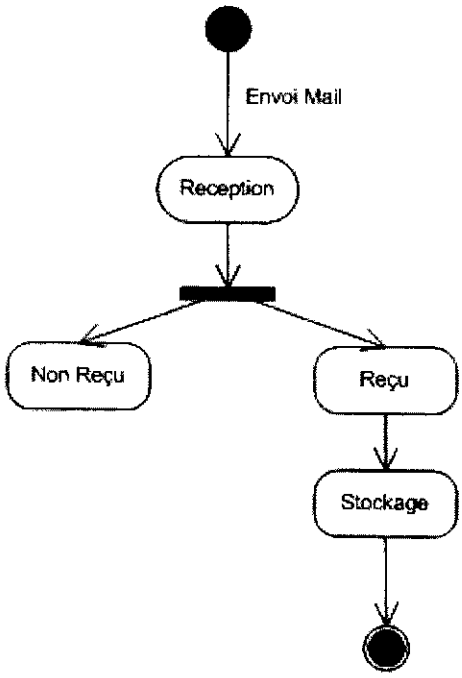


Fig. VIII.10 Diagramme de transition d'envoi d'E-mail.

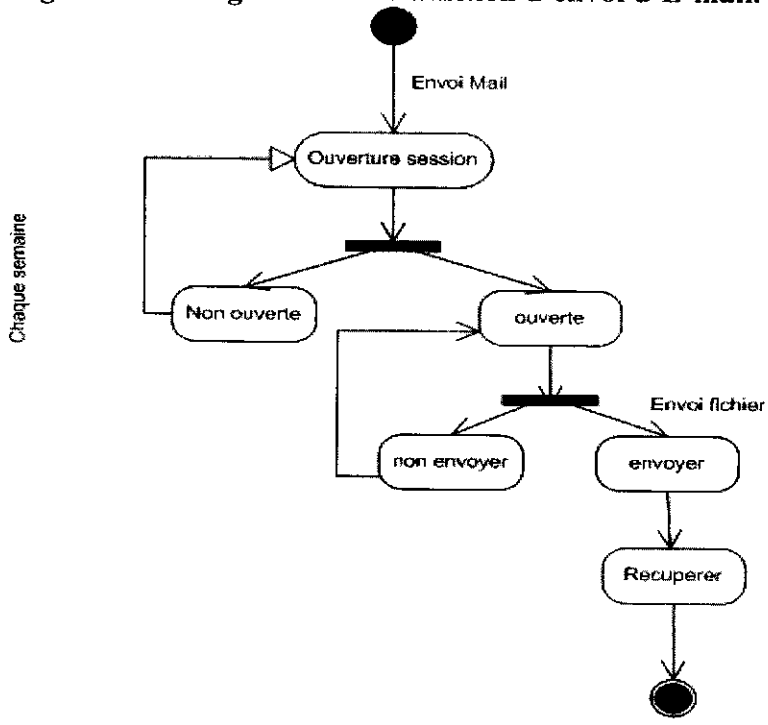


Fig. VIII.11 Diagramme de transition d'ouverture de session.

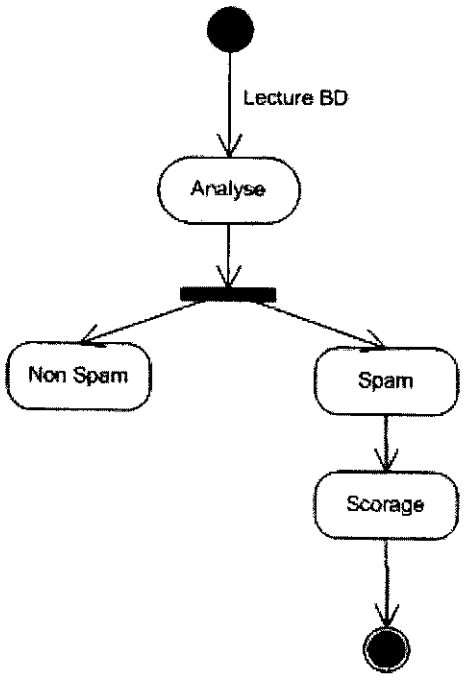


Fig. VIII.12 Digramme de transition d'analyse d'E-mail.

3.5 Diagramme de séquence et de communication

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur. Il indique les objets que l'acteur va manipuler et les opérations qui font passer d'un objet à l'autre. On peut représenter les mêmes opérations par un diagramme de communication graphe dont les nœuds sont des objets et les arcs (numérotés selon la chronologie) les échanges entre objets.

En fait, diagramme de séquence et diagramme de communication sont deux vues différentes mais logiquement équivalentes (on peut construire l'une à partir de l'autre) d'une même chronologie. Ce sont des diagrammes d'interaction.

3.5.1 L'envoi d'un mail

1. un expéditeur envoie un ou plusieurs mails à un ou plusieurs serveurs.
2. le serveur mail stocke les emails dans le rep_temp_dest.

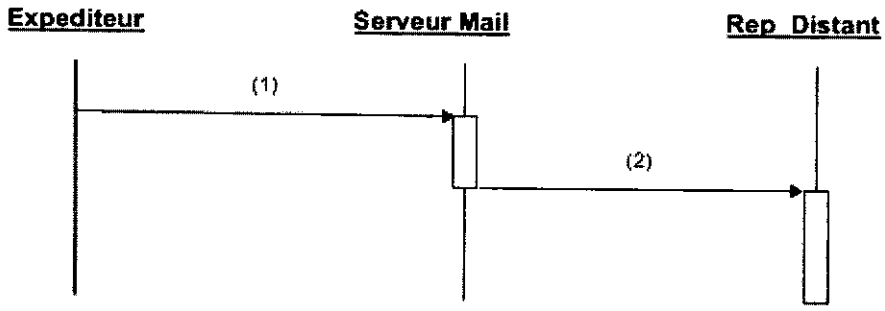


Fig. VIII.13 Diagramme de séquence d'envoi d'E-mail.

3.5.2 Ouverture d'une session ftp

1. le module d'ouverture de session ftp ouvre une session ftp sur le système distant.
2. une confirmation d'ouverture est envoyée au module d'ouverture de session.
3. le module transfère le contenu du rep_dist au rep_temp_central.

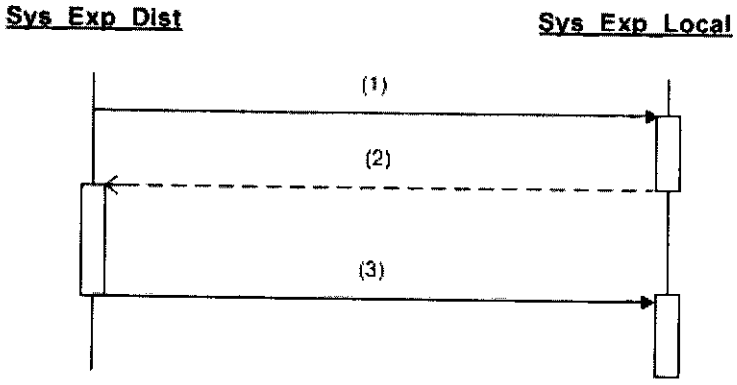


Fig. VIII.14 Diagramme de séquence d'ouverture d'une session FTP.

3.5.3 Module de subdivision

1. Le module de Découpage lit et décompose le fichier à partir Fichier_mail.
2. Le fichier est Décomposé en différentes mails.
3. Le module de Découpage décompose le mail en différentes champs.
4. Le mail est décomposé en différents champs.
5. Récupère chaque champ.
6. Le champ est récupéré.
7. Insère les différents champs dans mail_décomp (BD).
8. Tous les champs d'E-mail sont stockés dans la base de données .

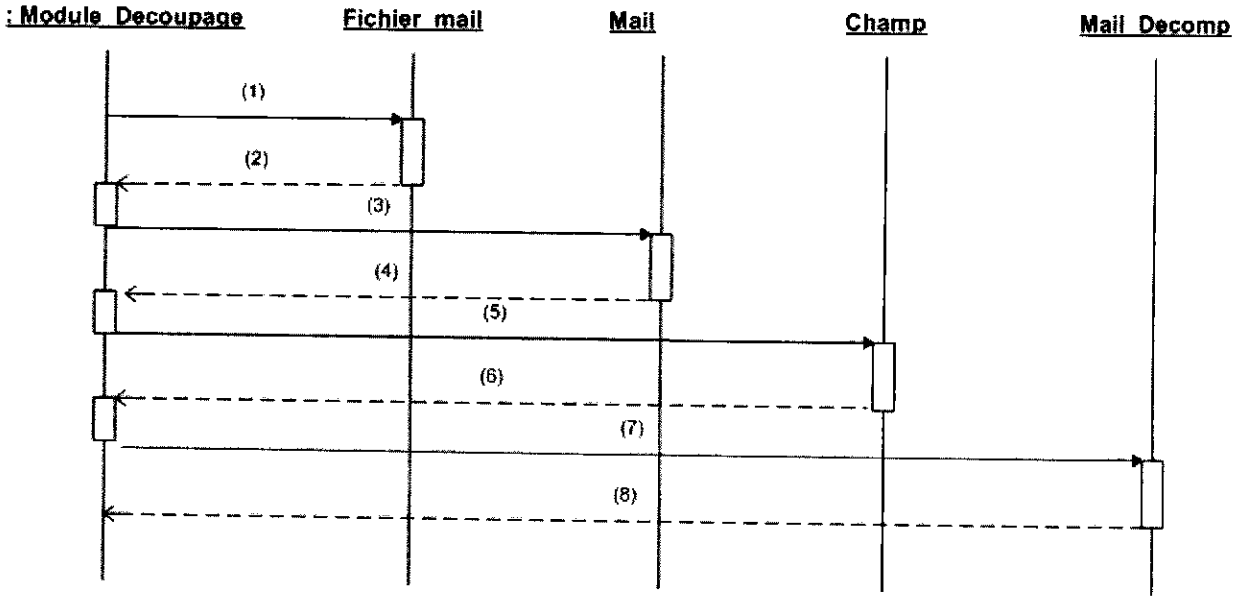


Fig. VIII.15 Diagramme de séquence de subdivision.

3.5.4 Module d'analyse et de catégorisation

1. Le 1er algorithme vérifie le **[sujet]** ; puis il donne un score d'évaluation selon qu'il est suspect ou pas, il ajoute le score ainsi obtenu au score précédemment donné au mail.
2. La fin du 1er algorithme déclenche le début du 2eme algorithme.

3. Le 2eme algorithme vérifie le champ **[adresse IP]** ; puis il donne un score d'évaluation selon qu'elle est black-listée ou pas, il ajoute le score ainsi obtenu au score précédemment donné au mail.
4. La fin du 2eme algorithme déclenche le début du 3eme algorithme.

5. Le 3eme algorithme vérifie le champ **[domain-Sender]** ; puis il donne un score d'évaluation selon qu'il est black-listé ou pas, ou bien s'il est trusted ou pas, il ajoute le score ainsi obtenu au score précédemment donné au mail.
6. La fin du 3eme algorithme déclenche le début du 4eme algorithme.

7. Le 4eme algorithme vérifie la **[date]** ; puis il donne un score d'évaluation selon le temps séparant deux réceptions du message du même Sender, il ajoute le score ainsi obtenu au score précédemment donné au mail.
8. La fin du 4eme algorithme déclenche le début du 5eme algorithme.

9. Le 5eme algorithme **analyse** le **[corps du message]** ; puis il donne un score d'évaluation selon les mots utilisés ou bien les liens insérés, il ajoute le score ainsi obtenu au score précédemment donné au mail.
10. La fin du 5eme algorithme déclenche le début de l'algorithme de catégorisation.

11. L'algorithme de **catégorisation** selon des critères bien évalués détermine la catégorie du mail et l'écrit dans la case dédiée.
12. La fin de l'algorithme de catégorisation déclenche le début de l'algorithme de décision.

13. L'algorithme de **décision** selon le score et la catégorisation du message décide s'il est spam ou non en utilisant un mécanisme de décision bien adopté.

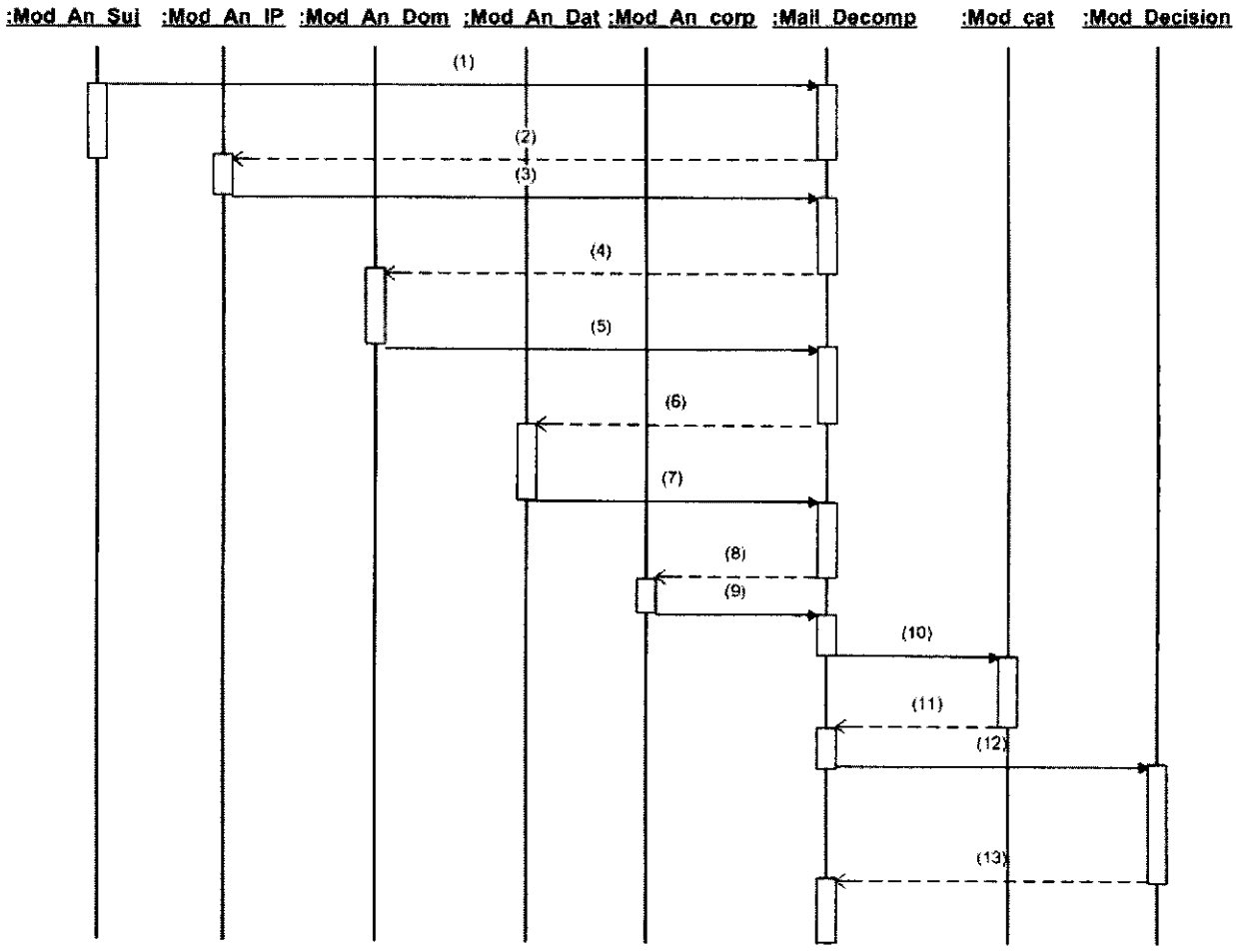


Fig. VIII.16 Diagramme de séquence D'analyse d'E-mail.

3.5.5 Manipulation de RBL

1. à partir de la table mail_décomp un module de génération de format standard génère un format standard (RBL ou SBL).
2. Un module de mise-à-jour communique la table RBL ainsi générée aux Anti-spam pour les mettre à jour.

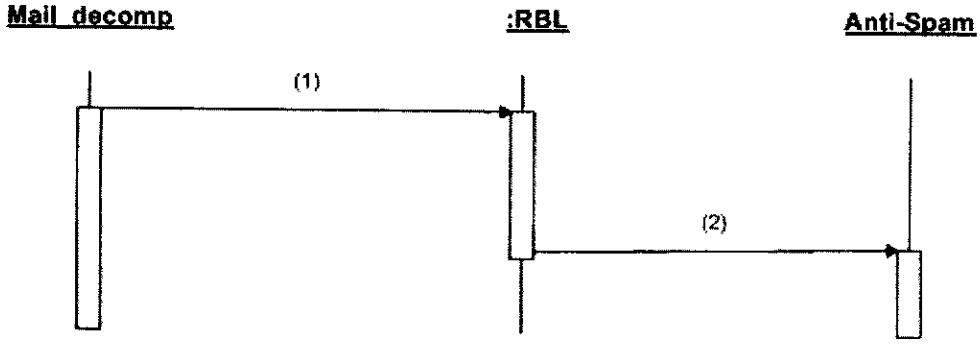


Fig. VIII.17 Diagramme de séquence d'envoi d'E-mail.

4. Diagramme de classe

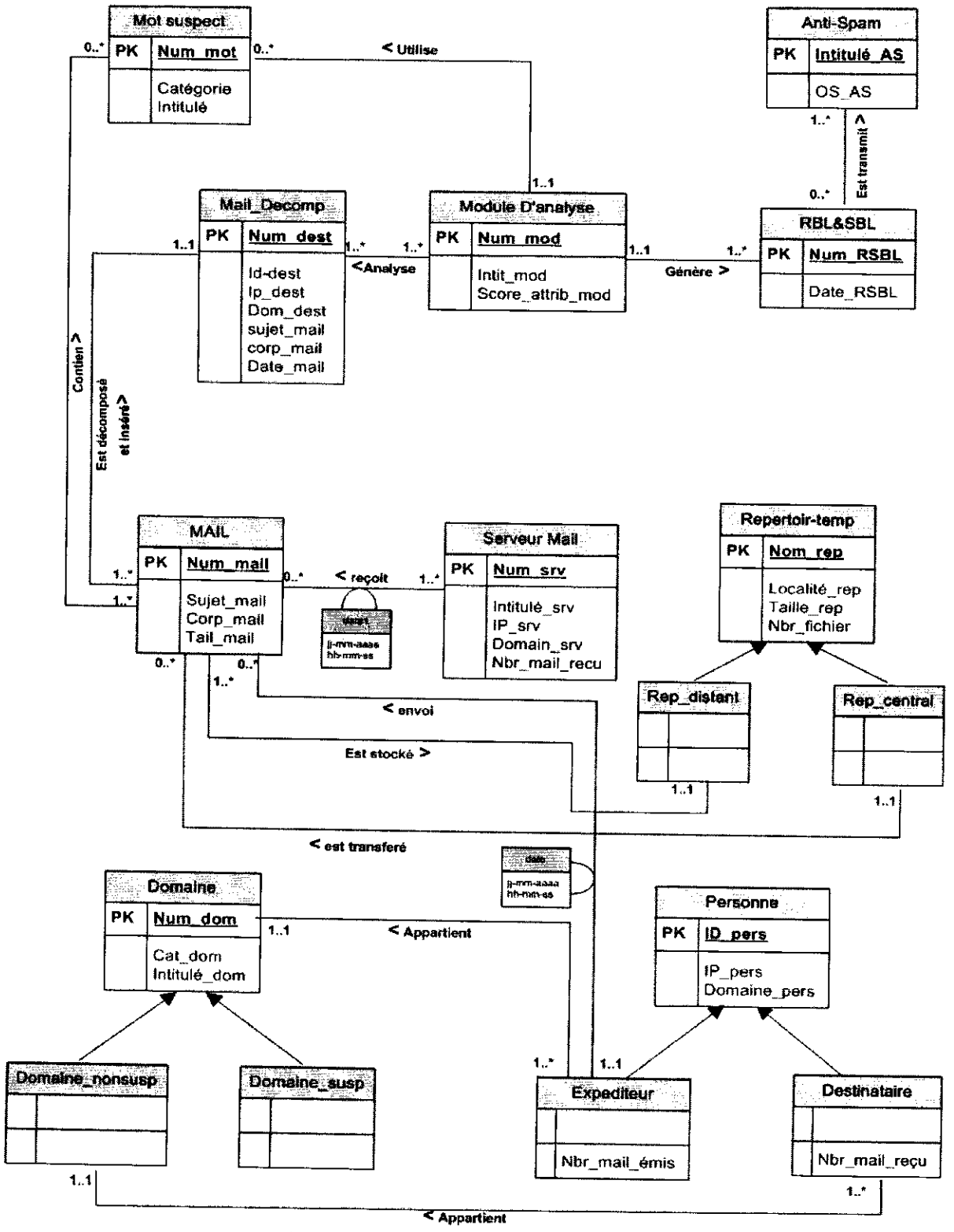


Fig. VIII.18 Diagramme de Classe.

5. Diagramme de déploiement

5.1 Définition

Les diagrammes de déploiement montrent la disposition physique des matériels qui composent le système et la répartition des composants sur ces matériels.
Les ressources matérielles sont représentées sous forme de nœuds. Les nœuds sont connectés Entre eux, à l'aide d'un support de communication. La nature des lignes de communication et leurs caractéristiques peuvent être précisées. Les diagrammes de déploiement peuvent montrer des instances de nœuds (un matériel précis), ou des classes de nœuds. **[UMD-34].**

5.2 Le diagramme de déploiement

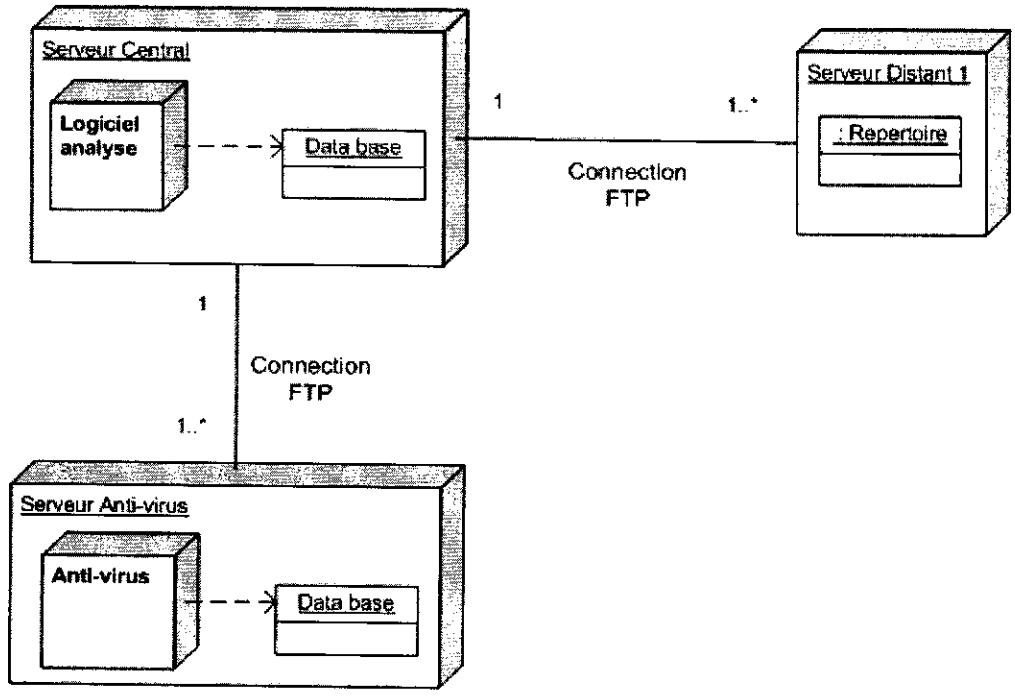


Fig. VIII.19 Diagramme de déploiement.

6. Conclusion

Dans ce chapitre, nous avons présenté une conception de l'application et une conception du système B2SAM.
De plus, par l'intermédiaire de plusieurs diagrammes, nous avons essayé de bien expliquer la solution proposée tout en illustrant l'ensemble de ses composants et leur manière d'agir, après la modélisation de la solution proposée, on passe maintenant à la réalisation, ce qui est proposé dans le chapitre suivant.

CHAPITRE IX.

Implémentation

Et mise en œuvre.

La ou il y a une volonté il y a un chemin.
– Belkas Mahmoud.

1. Introduction

Dans ce chapitre nous présentons une proposition d'une mise en œuvre de notre architecture déjà expliquée dans le chapitre décrivant la conception du « B2SAM », ainsi que l'explication illustrée de tous les modules d'analyse et de leurs interactions, en exposant tous les aspects importants de l'implémentation. Et on terminera par exposé les résultats expérimentaux de tests de notre système avec différents types de situations.

2. Choix du langage de programmation

L'efficacité des applications les plus avancées repose sur l'utilisation des langages les plus robustes pour garantir une meilleure interaction avec les différentes plateformes, le langage C++ est l'un des langages les plus appropriés pour le développement des applications même les plus complexes aux niveaux conceptuels.

Ainsi parmi les avantages de C++ qui on favorisé notre choix les suivants :

- C++ est un langage orienté objet qui préserve l'aspect de la programmation modulaire.
- Est le langage le plus utilisé dans le développement des Systèmes d'exploitation.
- La mobilité de l'exécutable vers différentes machines.
- Flexibilité de manipulation.
- Meilleur interaction avec le système d'exploitation et meilleur gestion des API, ainsi que l'efficacité de gestion des exceptions.

Pour IDE « Integrated Development Environment - environnement de développement intégré » on a choisi le « Borland Builder C++ Version 6 Entreprise Edition » grâce à sa meilleure gestion d'interface graphique la plus approprié pour l'utilisateur, et qui rend l'interaction avec un tel environnement plus flexible et plus efficace, ainsi que l'utilisation du débuggeur avancé qui est totalement compatible avec le système d'exploitation même au niveau le plus basic, aussi l'intégration de la technologie Microsoft ADO (*Active Data Object*) pour la gestion des base de données, et le regroupement d' un jeux de composants VCL intéressent.

3. Implémentation

Dans cette section nous allons exposer les modules utilisés par notre système dans le cadre de développement logiciel, et qui ont été spécifiés dans l'architecture proposées du système, présentées dans le chapitre « conception du B2SAM ».

3.1 Script d'ouverture d'une session ftp

Le script utilisé ici est un script écrit pour un système d'exploitation UNIX et ces dérivés en utilisant le Shell appelé « BASH » sous Linux RedHat pour ouvrir une session ftp. L'ouverture de cette session ftp est programmée chaque semaine pour récupérer les fichiers d'e-mails des systèmes distants on les déposant dans un répertoire spécifique et dédies à cette effet, ce répertoire est par défaut installé dans le « C:\ », et le fichier des e-mails est appelé « ANALYSE.txt ».

L'image suivante montre les instructions Shell utilisées pour programmer la tache d'ouverture de la session ftp sous Linux :

```
EDITOR = vi
export EDITOR
crontab -e
  10 21 * * 0 /etc/init.d/ftpagent
```

Fig. IX.1 Commandes d'automatisation de la tache.

Et voici le script approprie :

```
#!/bin/sh
#shell script to open an ftp session and get mails
#
 ftp
 open "name of local_domain"
 anonymous
 user@domain.com
 if [ftp session fails];then  loop
 else
     lcd /var/spool/mail
     binary
     put root
 fi
 bye
```

Fig. IX.2 Script d'ouverture d'une session ftp.

3.2 Module de chargement des e-mails

Ce module ouvre le répertoire par défaut (celui spécifié dans le script) et charge le fichier appropries dans le système pour l'analyser; ainsi si le répertoire par défaut n'existe pas seul l'administrateur pourra changer de répertoire pour charger les fichiers de l'analyse, c'est pour cette raison on a distingué entre l'utilisateur normale du système et l'administrateur qui possède le mot de passe approprie pour effectuer toutes les modifications nécessaires.

Le code suivant montre les méthodes utilisées pour la vérification de mot de passe ainsi que le code source du module de vérification de chemin et du chargement d'e-mails :

```
void __fastcall Password_check (TObject *Sender)
{
   // Vérifier si le mot de passe entré est valide :
 if (MaskEdit1->Text == Password)
  {
   pass = true;
   ...
  }
else
  {
  MessageDlg ("\n Mot de passe invalide, Veuillez consulter votre
   administrateur.", mtWarning, TMsgDlgButtons () << mbOK, 0);
  ...
  ...
  }
}
```

Fig. IX.3 Méthode de vérification de mot de passe.

```
String __fastcall path_ctrl (String path)
{
if (!FileExists(path))
  {
   if (MessageBox(Handle,"Le dossier source n'existe pas. ",
      "Erreur de chemin", MB_YESNO) == IDYES)
     {
      Password_Form->ShowModal();
      if (Password_Form->pass == true)
        {
         .
         .
         .
        }
     }
  }
  return path;
}
```

Fig. IX.4 Code de vérification de chemin.

```
void __fastcall chargement_Module (TObject *Sender)
{
String source_folder = "C:\\ANALYSE\\Analyse.txt";
try
   {
    source_folder = path_ctrl (source_folder);
    TStringList *Liste = new TStringList;
    Liste->LoadFromFile (source_folder);
    .
    .
    .
    delete Liste;
   }
catch (...) {}
}
```

Fig. IX.5 Code de chargement d'e-mails.

La figure suivante présente le code source du découpage des e-mails en utilisant un **automate** pour la gestion du procédé de découpage :

```
void __fastcall decoupage_Module (TObject *Sender)
{
// Module de découpage des e-mails chargés.

if (!Text_source->Text.IsEmpty())
  {
  for (int i=0; i <= Text_source->Lines->Count-1; i++)
     {
      .
      .   // Insertion des fragments des messages dans la table
      .   // d'analyse.
     }
  }
```

Fig. IX.6 Code de découpage des e-mails.

3.3 Module d'analyse entre deux tables de base de données :

```
void __fastcall Analyser (TADOTable *Table_work, TADOTable
   *Table_model, String champ_work, String champ_model, float
   score)
{
Table_work->DisableControls();
Table_model->DisableControls();
try
{
  Table_work->First();
  while (!Table_work->Eof)
  {
    Table_model->First();
    while (!Table_model->Eof)
    {
     .
     .
     .
    }
     Table_model->Next();
    }
    Table_work->Next();
  }
}
__finally
{
  Table_work->EnableControls();
  Table_model->EnableControls();
}
}
```

Fig. IX.7 Code de l'analyse entre deux tables.

3.4 Module d'analyse d'adresse IP

```
Analyser (Mail_decomp_table, IP_table, "IP_Field", "Intitule",
   score_IP);
```

Fig. IX.8 Code de l'analyse de l'adresse IP.

3.5 Module d'analyse du domaine de l'expéditeur

```
Analyser (Mail_decomp_table, Domain_table, " Domain_Field ",
   "Intitule", score_ Domain);
```

Fig. IX.9 Code de l'analyse du domaine.

3.6 Module d'analyse du sujet de message

```
void __fastcall Subject_analyser (TObject *Sender)
{
Mail_decomp_table->DisableControls();
String subject_buffer = "";
try
{
  Mail_decomp_table->First();
  while (!Mail_decomp_table->Eof)
  {
    if(   subject_buffer.IsEmpty() ||
          .
          .
          .
          !subject_buffer.SubString(1,2).AnsiCompare("__")
      )
      {
       ... // l'ajout du score.
      }

    Mail_decomp_table->Next();
  }
}
__finally
 {
  Mail_decomp_table->EnableControls();
 }
}
```

Fig. IX.10 Code de l'analyse du sujet.

3.7 Module d'analyse du corps des E-mail

```
void __fastcall Body_analyser (TObject *Sender)
{
Analyser (Mail_decomp_table, Sword_table, "Corp_mail",
   "Intitule", score_mot);      // analyse basée sur les mots.
Analyser (Mail_decomp_table, Phrase_table, "Corp_mail",
   "intitule", score_phrase);  // analyse basée sur les phrases.
}
```

Fig. IX.11 Code de l'analyse du corps du message.

4. Réalisation de l'application

L'application réalisée est une interface graphique manipulant notre système de traitement des e-mails jusqu'à la génération du format standard de mise à jour.
Ainsi la figure suivante présente l'interface principale de notre logiciel :

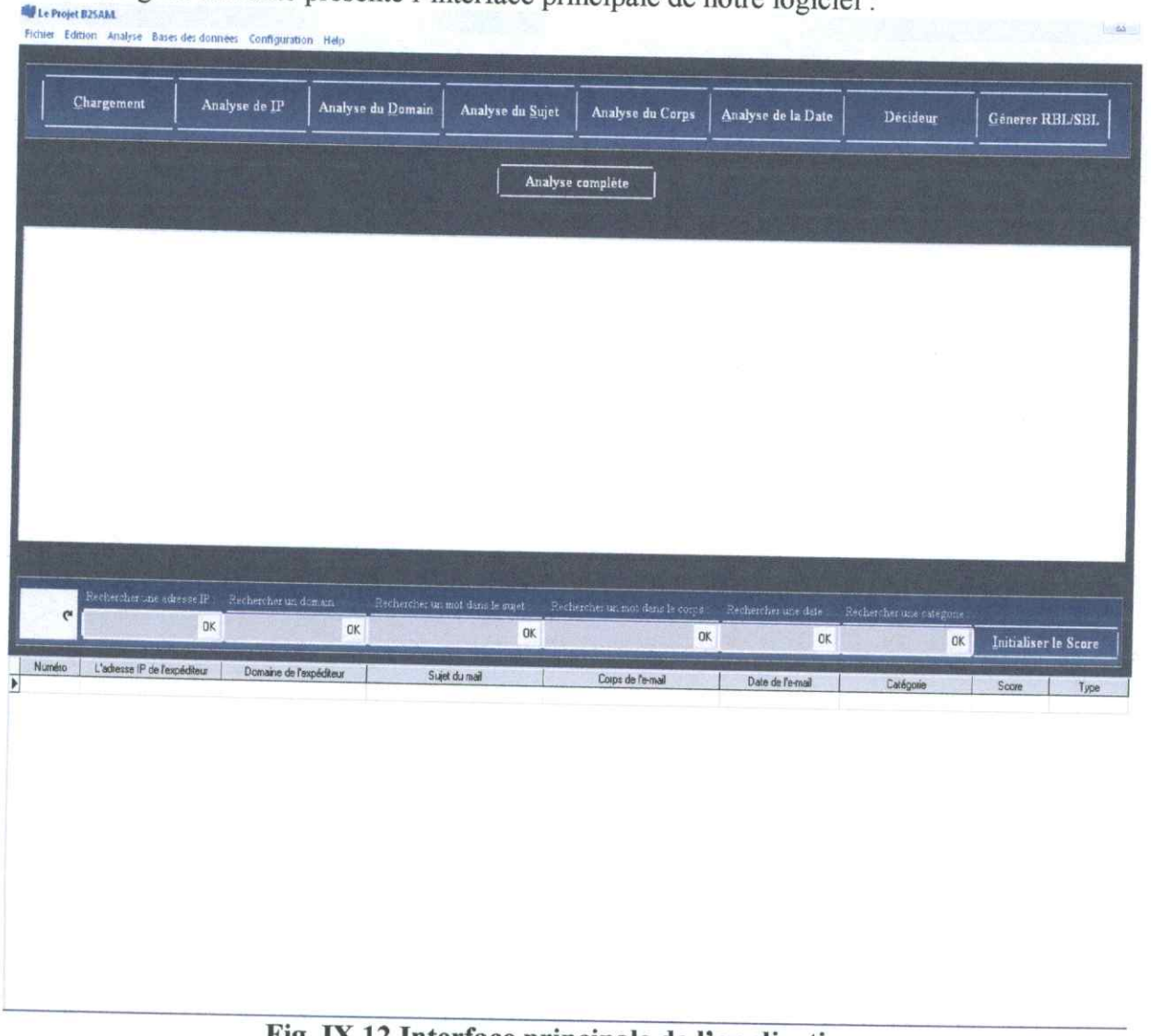


Fig. IX.12 Interface principale de l'application.

4.1 Modification de la configuration

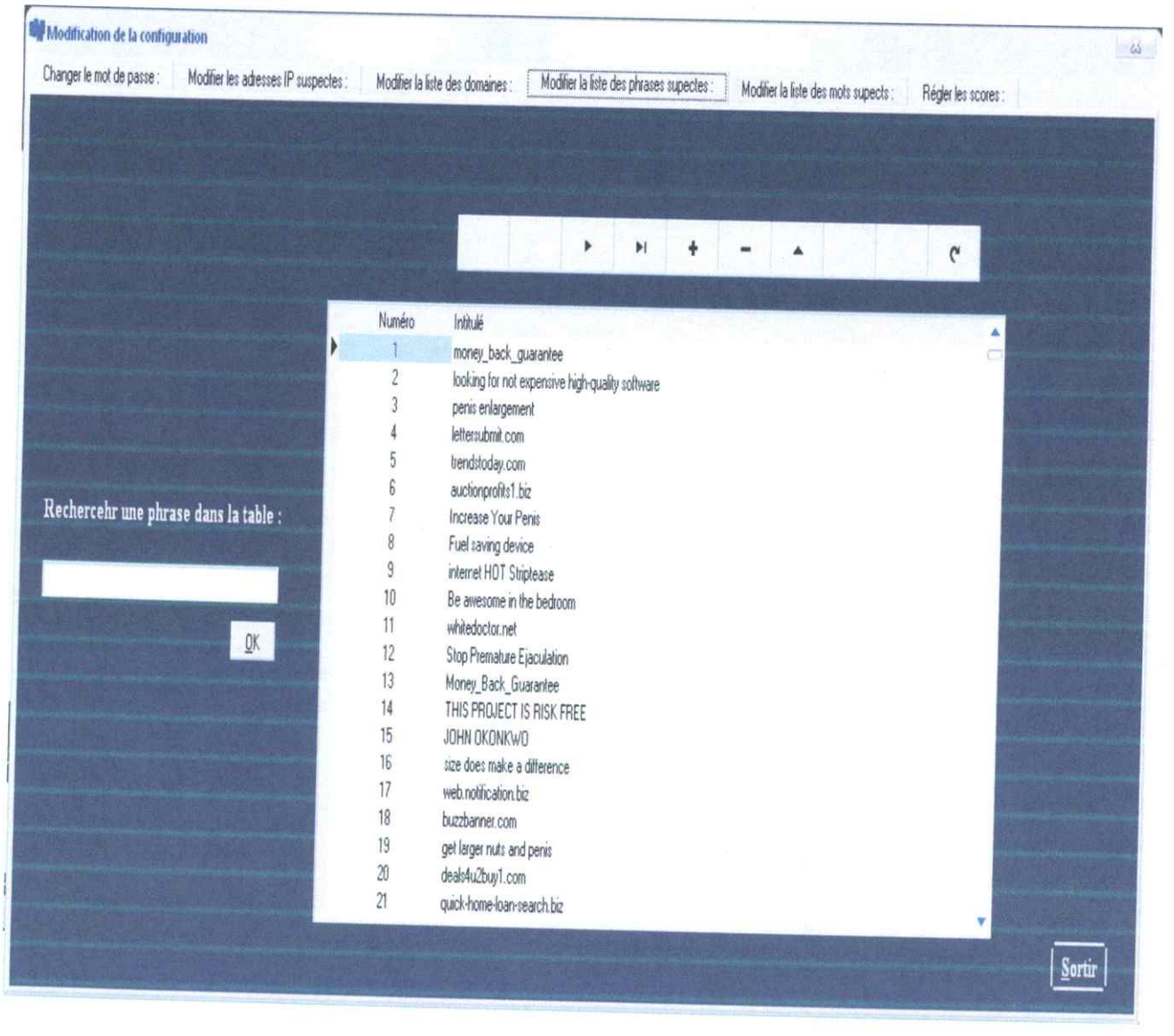


Fig. IX.13 Fiche de modification de la configuration.

4.2 Génération du format standard

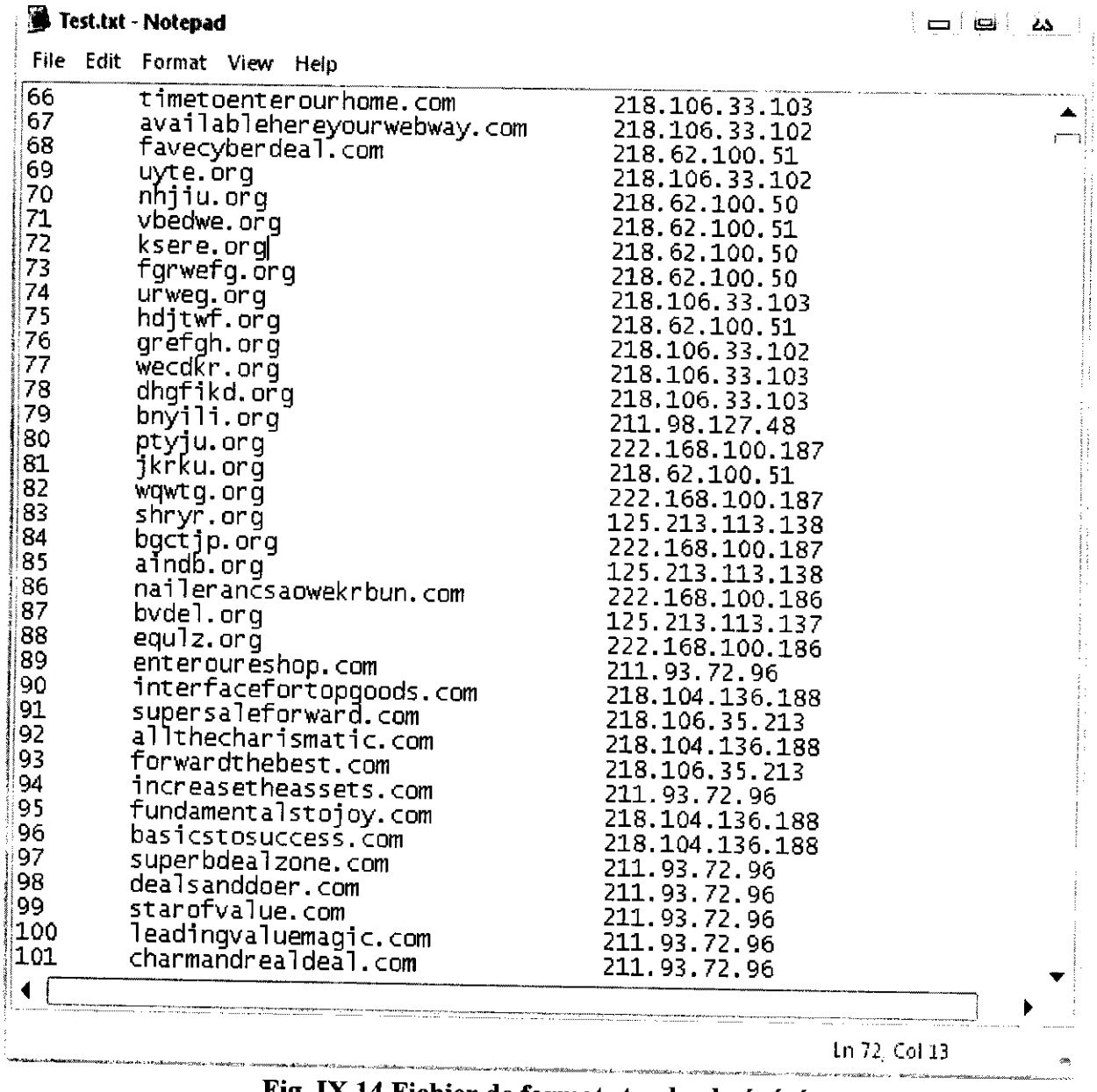


Fig. IX.14 Fichier de format standard généré.

5. Test effectué

Apres l'implémentation et la réalisation du système d'analyse B2SAM, et dans le but de tester son efficacité et son mise en œuvre on a mis au point une architecture réseau bien spécifique pour bien visualiser les étapes d'acheminement des E-mail à partir de centre de collecte d'E-mail jusqu'à la création du format standard RBL et mise à jour des anti-spams en passant par la moulinette de l'analyse, la figure suivante présente l'architecture de test globale du système HoneyPot Mail, avec le serveur mail « SendMail » et le system d'exploitation Linux RedHat qui on été utilisés dans le but de la collecte des E-mail, l'ouverture d'une session FTP et l'envoi du fichier des E-mail en format texte au système local de l'analyse dans un répertoire bien spécifié ainsi que la mise à jour de l'anti-spam « *Spamassassin* » sous Linux et avec un serveur mail « SendMail ».

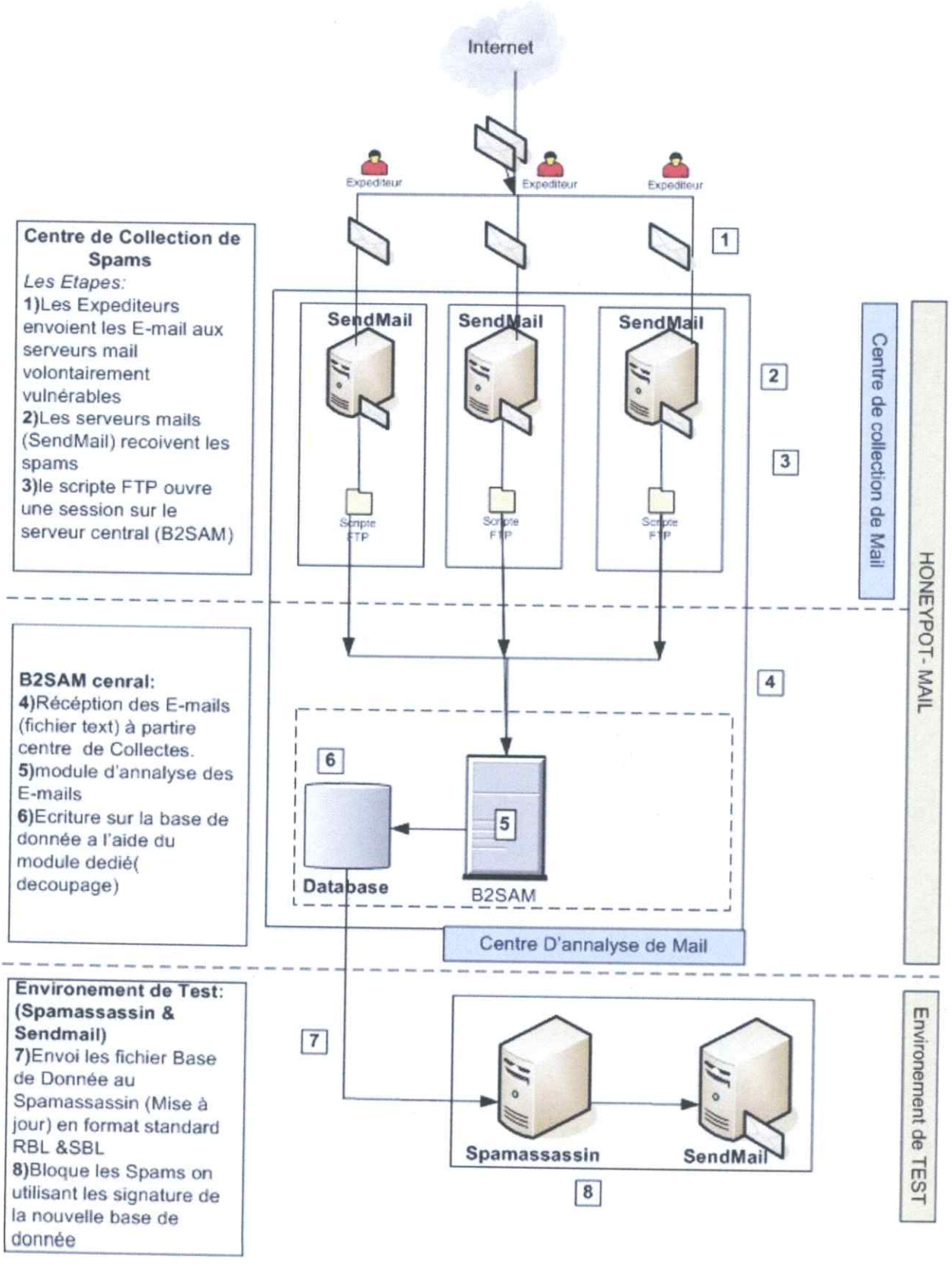


Fig. IX.15 Environnement de test du HoneyPot Mail.

L'utilisation des serveurs mail demande l'acquisition de **domaines** pour qu'ils puissent êtres publiés et par conséquence attaqués ; ce qui n'est pas dans le cadre de cette dissertation, donc on a simulé la mise en place de ces serveurs mail dans le but de tester l'efficacité de notre solution HoneyPot.

La figure suivante représente le test effectué en recevant des e-mails dans un domaine local entre deux machines de test possédant deux noms de domaines différents, en simulant le domaine (suspect) de l'émetteur et le domaine local du destinataire :

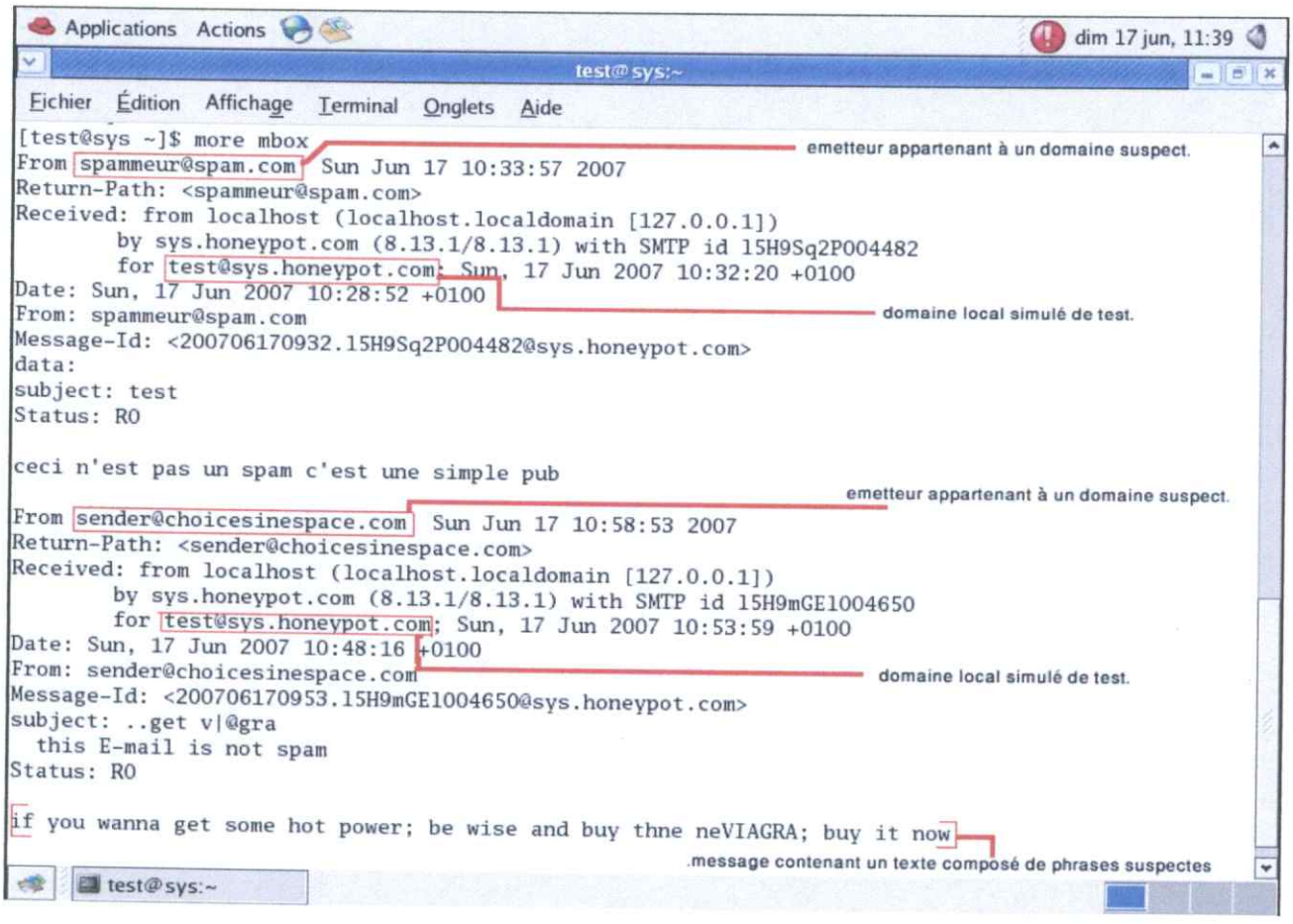


Fig. IX.16 Aperçus des messages reçus de différents domaines.

A cause de l'autonomie du fonctionnement de notre système c-à-d que la collecte des e-mails est indépendante de l'analyse, le déroulement de ce test commence par le lancement de notre application avec la configuration par défaut contenant le mot de passe « admin » et chemin de recherche de fichiers « C:\ANALYSE\ANALYSE.txt », on cliquant sur le bouton « chargement » le fichier par défaut est chargé dans l'éditeur de texte pour visualiser tous les e-mails qui ont été récupérés auprès les serveurs de collecte, puis l'analyse sera déclenchée par le bouton « analyse complète » et qui se déroule sur 6 étapes :

1. Découpage des e-mails et insertion dans la table de l'analyse.

2. Les analyses consécutives de l'adresse IP, du domaine, de sujet, et de la date et le corps du message sont exécutées une après l'autre et chaque analyse donne un score spécifique à elle.
3. Le module de catégorisation détermine la catégorie de chaque message pour mis à jour le champ « catégorie ».
4. Le module de décision en analysant les scores obtenus décide si le message est spam ou pas et écrit dans la case « type » la décision élue.
5. Le module de génération de format standard parcoure la table de l'analyse puis génère un format standard en incluant les messages qui ont été signalés comme spam ; ce format standard est sous forme texte et sauvegardé dans le fichier spécifique « C:\RBL/SBL\fichier1 ».
6. Le format standard ainsi généré est contacté aux anti-spams pour les mettre à jour et spécialement « *spam assassin* » dans la base de données locale de ce dernier.

La figure suivante montre la table d'analyse complète avec tous les enregistrements remplis :

Rechercher une adresse IP : GO | Rechercher un domain : GO | Rechercher un mot dans le sujet : GO | Rechercher un mot dans le corps : GO | Rechercher une date : GO | Rechercher une catégorie : GO | Initialiser le Score

Numéro	L'adresse IP de l'expéditeur	Domaine de l'expéditeur	Sujet du mail	Corps de l'e-mail	Date de l'e-mail	Catégorie	Score	Type
1	41.201.160.5	77-mail.net	_take care	take care of your self.....	5/12/2007 4:22:50 PM	Escroquerie	18	SPAM
2	64.233.171.83	gmail.com	urgent	bonjour je vous pris.....	5/12/2007 6:11:02 AM		0	Non spam
3	41.252.156.30	orlawyers.com	CAN YOU IMAGINE THAT YOU ARE HE.	LegalaRxMedication drug-shop propose a	5/13/2007 11:03:11 AM	Sante	23	SPAM
4	216.130.199.25	ns1.aisleymore.com	summer is almost here,be ready	take adventage of the chance--	5/13/2007 11:06:10 AM	Tourisme	28	SPAM
5	222.60.80.191	coldustesalandit.com	get out of the obese crowd	do not waste opportunity! anarim, The...	5/13/2007 11:56:20 AM	pornographie/escroquerie	25	SPAM
6	218.62.100.51	supersaleessential.com	stop the painful craving fot\r more food	do your counte up all the cases when you	5/16/2007 2:56:20 AM	pornographie/escroquerie	30	SPAM
7	218.62.100.51	favecybermarket.com	stop the painful craving fot\r more food	do your counte up all the cases when you	5/16/2007 2:57:10 AM	pornographie/escroquerie	30	SPAM
8	218.62.100.51	favecybermarket.com	stop the painful craving fot\r more food	do your counte up all the cases when you	5/16/2007 2:57:22 AM	pornographie/escroquerie	30	SPAM
9	218.62.100.51	favecybermarket.com	stop the painful craving fot\r more food	do your counte up all the cases when you	5/16/2007 2:58:54 AM	pornographie/escroquerie	30	SPAM
10	218.62.100.51	favecybermarket.com	stop the painful craving fot\r more food	do your counte up all the cases when you	5/16/2007 2:59:03 AM	pornographie/escroquerie	30	SPAM
11	64.4.33.7	hotmail.com	invitation	la societe TechRepublic organise.....	5/18/2007 4:32:20 PM		5	Non spam

Fig. IX.17 Table d'analyse dans un environnement de test.

Si l'utilisateur désire changer la configuration ou bien modifier la table des mots suspects et ou bien la table des phrases suspectes il invité à fournir le mot de passe administrateur comme illustre la figure suivante :

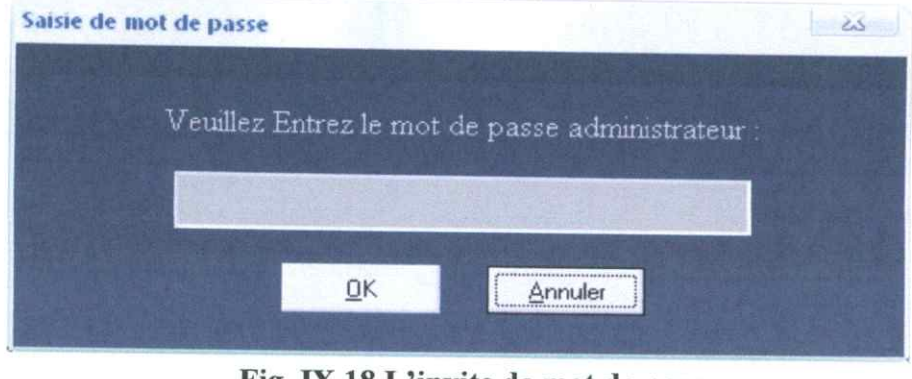


Fig. IX.18 L'invite de mot de passe.

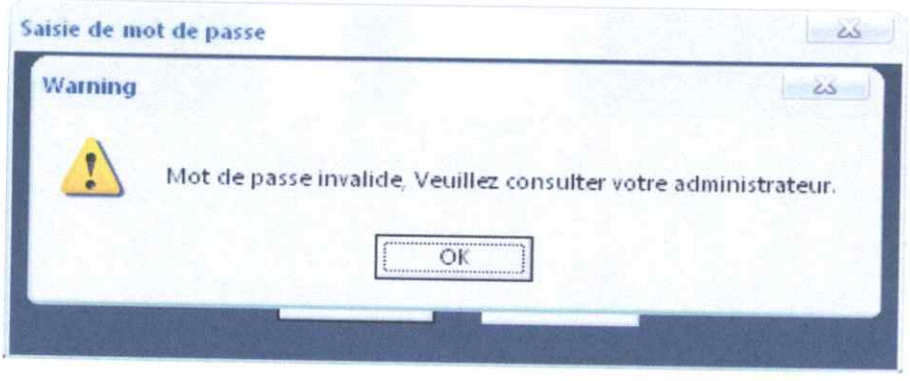


Fig. IX.19 Mot de passe invalide.

Si le mot de passe est valide il peut personnaliser la configuration comme il est montré dans la **Fig. IX.13.**

6. Conclusion

Dans ce Chapitre, nous avons présenté. Une proposition d'une mise en œuvre possible de notre architecture décrit auparavant, à base d'un mélange de techniques et de méthodes pour récupérer les emails et faire les analyses correspondantes pour enfin générer un format standard de mis à jour.

Le coté conversationnel a été possible grâce à l'utilisation de techniques de programmation les plus avancées et les API de manipulation des systèmes pour formaliser les interactions nécessaires.

Notre système a donné des résultats satisfaisants, mais des améliorations restent encore à faire, notamment en ce qui peut survenir pour les techniques d'analyse surtout avec le développement de la technique d'analyse des images.

Conclusion générale.

Le gain de notre étude
c'est en être devenu
meilleur et plus sage.
-Michel Eyquem.

L'objectif primordial du présent travail consistait à mettre en œuvre une solution distribuée dans le but de piéger les spammeurs et connaitre leurs méthodes de procéder pour mieux protéger les systèmes de messagerie en offrant des formats standards de mise à jour pour les anti-spams.

La motivation qui sous-tend ce travail réside principalement dans le fait que les pertes liées aux Spams sont considérables comme la limitation de la bande passante ou la perte du temps et de l'espace quelle soit du coté serveur de messagerie ou du coté utilisateur de mailing.

Au cours de notre projet nous avons développé un Système dans le but est de collecter le maximum de Spams en s'appuyant sur des serveurs mail volontairement vulnérables ce qui est bien le « HoneyPot mail » pour dévoiler le profile du spammeur et mieux l'anticiper, en masquant tous les obstacles liés aux hétérogénéités des systèmes.

Perspectives :

En perspective, nous proposons quelques extensions du système, qui se résument dans ce qui suit :

- L'ajout de module d'analyse basé sur l'analyse des images et de reconnaissance des formes.
- L'adoption de nouveaux formats de mise à jour basé sur les formats de base de données et non seulement les formats texte.

Annexe.

1. Le modèle OSI

1.1 Historique

À ses débuts, le développement des réseaux était désorganisé à de nombreux égards. Le début des années 1980 a été marqué par une croissance exceptionnelle du nombre et de la taille des réseaux. Pour profiter des avantages de l'utilisation de la technologie réseau, les sociétés se mirent à installer et à étendre des réseaux, à un rythme presque aussi rapide que l'introduction des nouvelles technologies.

Vers le milieu des années 80, ces sociétés commencèrent à rencontrer des problèmes, justement en raison de cette extension rapide. Tout comme il est difficile pour les gens qui ne parlent pas la même langue de communiquer, il devenait difficile pour des réseaux qui utilisent des spécifications et des implémentations différentes d'échanger des informations. Les sociétés qui développaient des solutions de réseau privé ou propriétaire furent confrontées au même problème. Le terme « propriétaire » signifie ici qu'une entreprise ou un petit groupe d'entreprises contrôle entièrement l'utilisation de la technologie. Les technologies réseau qui suivaient strictement des règles propriétaires ne pouvaient pas communiquer avec des technologies qui respectaient des règles propriétaires différentes.

Pour résoudre le problème de l'incompatibilité des réseaux, l'OSI (*International Organization for Standardization*) examina les modèles réseau tels que DECnet (*Digital Equipment Corporation net*), SNA (*Systems Network Architecture*) et TCP/IP afin de trouver un ensemble de règles applicables de façon générale à tous les réseaux. Sur la base de ces recherches, l'ISO a mis au point un modèle de réseau pour aider les fournisseurs à créer des réseaux compatibles avec d'autres réseaux.

Le modèle de référence OSI (*Open System Interconnection*) publié en 1984 fut le modèle descriptif de réseau créé par l'ISO. Ce modèle propose aux fournisseurs un ensemble de normes assurant une compatibilité et une interopérabilité accrues entre divers types de technologies réseau produites par de nombreuses entreprises à travers le monde.

Le modèle de référence OSI s'est imposé comme le principal modèle pour les communications réseau. Bien qu'il existe d'autres modèles, la plupart des constructeurs de réseau se fondent sur le modèle de référence OSI pour la conception de leurs produits. Cela est particulièrement vrai lorsqu'ils souhaitent éduquer les utilisateurs à l'utilisation de leurs produits. Ce modèle est considéré comme le meilleur outil disponible pour décrire l'envoi et la réception de données sur un réseau.

Les avantages du modèle OSI :

- Il réduit la complexité.
- Il uniformise les interfaces.
- Il facilite la conception modulaire.
- Il assure l'interopérabilité de la technologie.
- Il accélère l'évolution.
- Il simplifie l'enseignement et l'acquisition des connaissances.

1.2 Les couches du modèle OSI

Le modèle de référence OSI constitue un cadre qui aide à comprendre comment les informations circulent dans un réseau. Ce modèle explique comment les paquets transitent à travers les différentes couches vers un autre équipement du réseau, même si l'émetteur et le destinataire utilisent des types de média réseau différents.

Le modèle de référence OSI comporte sept couches numérotées, chacune illustrant une fonction réseau bien précise. Le découpage du réseau en sept couches présente les avantages suivants:

- Il permet de diviser les communications sur le réseau en éléments plus petits et plus gérables.
- Il uniformise les éléments du réseau afin de permettre le développement et le soutien multi constructeur.
- Il permet à différents types de matériel et de logiciel réseau de communiquer entre eux.
- Les modifications apportées à une couche n'affectent pas les autres couches.
- Il divise les communications sur le réseau en éléments plus petits, ce qui permet de les comprendre plus facilement.

La figure suivante illustre les différentes couches du modèle OSI :

Fig. A.I Les couches du modèle OSI.

Les différentes couches du modèle sont :

1. Couche physique

Assure une transmission binaire à l'aide des fils,connecteur, tension et débit.

2. Couche liaison de données

Elle fait un contrôle de liaison directe pour l'acces au media on assurant un transfert fiable des données par le media, et une connectivité et selection du chemin entre les système hotes.

3. Couche Réseau

Donne une adresse réseau et cherche le chemin le plus fiable pour le transfert des données on assurant un transfert fiable des données par le media, et une connectivité et selection du chemin entre les système hotes ainsi que l'adressage logique et acheminement au mieux.

4. Couche transport

Est une connexion de bout en bout, elle concerne les problèmes de transport entre les hôtes, on assurant :

- une fiabilité de transport de données.
- Etablissement, maintien et terminaison des circuits virtuels.
- Une détection des pannes et reprise du contrôle du flux d'information.

5. Couche session :

Elle assure une communication entre les hôtes, elle établit gère et ferme les sessions entre les applications.

6. Couche Présentation :

Elle assure :

- La lisibilité des données pour le système de destination.
- Format des données.
- Structure des données.
- Négocie la syntaxe de transfert des données pour la couche application.

7. Couche application :

Elle fournit des services aux processus d'application tels que le courrier électronique, le transfert de fichiers et l'émulation de terminal. **[AMO-4].**

2. Le Projet SPAMHAUS

Le projet Spamhaus est une organisation internationale à but non lucratif dont la mission est de dépister les troupes du Spam de l'Internet, d'assurer la protection sûre d'anti-Spam à temps réel pour des réseaux Internet, pour travailler avec des agences d'application de loi pour identifier et pour poursuivre des Spammeurs dans le monde entier, et pour inciter des gouvernements pour la législation efficace d'anti-Spam.

Fondé en 1998, Spamhaus est basé à Genève, en Suisse et à Londres, il est couru par une équipe consacrée de 25 investigateurs et spécialistes situés dans 9 pays.
Spamhaus édite le registre des opérations connues de Spam (ROKSO) – contient plus de 200 base de donnée qui stock les plus mauvaises troupes connues de Spam dans le monde entier.

D'après des statistiques dévoilées le 9 juin 2007 les listes de bloc de Spamhaus protègent 645.112.000 boîtes aux lettres d'utilisateur.
Spamhaus édite un certain nombre de bases de données de Spam-blocage en temps réel, y compris la liste de bloc de Spamhaus (SBL), la liste de bloc d'exploits (XBL) et la liste de bloc de politique (PBL).

Il est constitué d'un réseau de 40 serveurs de DNS dans 17 pays, les blocklists (SBL) de Spamhaus sont employés par plusieurs services fournisseurs d'internet principaux, des sociétés, des universités, des gouvernements et des réseaux militaires. L'infrastructure internationale est placée et contrôlée par une organisation séparée de logistique qui fournit des services utilisant des technologies de Spamhaus à l'industrie de filtre de Spam, y compris un service de synchronisation de blocklists de Spam assuré à de grands réseaux Internet et à organismes commerciaux de filtre de Spam. **[SHP- 32]**

Glossaire.

A

- **ASCII** (American Standard Code for Information Inter change) :
  Est la norme de codage de caractères en informatique la plus connue et la plus largement compatible. C'est également la variante américaine du codage de caractères ISO/CEI 646. Elle contient les caractères nécessaires pour écrire en anglais. Elle a été inventée par l'américain Bob Bemer en 1961. Elle est à la base de nombreuses autres normes (Unicode, ISO 8859-1, Windows-1252), qui l'étendent.
- **ASCII HT** (ASCII Horizontal tabulation).
- **ASRG** (Groupe de Recherche Anti-Spam).
- **ARPANET**(Advanced Research Projects Agency Network):
  Est le premier réseau à transfert de paquets développé aux États-Unis par la DARPA, le projet fut lancé en 1967 et la première démonstration officielle date d'octobre 1972. Il est le prédécesseur de l'Internet.
- **AUTODIN** (Automatic Digital Network) :
  Est un service de communications de données dans le département des Etats-Unis de la défense.

B

- **B2SAM** (Belkas Belhaoues System Analyses Module).
- **BAL** (Boîte Aux Lettres)
- **BBS** (Bulletin Board System):
  Consiste en un serveur équipé d'un logiciel offrant les services d'échange de messages, de stockage et d'échanges de fichiers, de jeux via un ou plusieurs modems reliés à des lignes téléphoniques.
- **Bcc** (Blind Carbon Copy) :
  Utilisée dans les courriels signifie en anglais « copie cachée ».Littéralement, il signifie "copie carbone aveugle".
- **BIOS** (Basic Input Output System):
  Un ensemble de fonctions, contenu dans la mémoire morte (ROM) de la carte mère servant à faire des opérations basiques (écrire un caractère à l'écran, lire un secteur sur un disque, etc. ...).
- **Booch :**
  L'une des méthodes d'analyse et de conception orientées objet à l'origine d'UML.
  Cette méthode permet de faciliter l'implémentation de programmes dans des langages de programmation orientée objet, ainsi que de représenter les différentes phases du développement d'un projet. Pour les programmeurs, l'avantage est d'obtenir une modélisation d'une solution qui soit indépendante de son implémentation dans un langage particulier (par exemple Java, C++, PHP...).

- **BST** (Basic Security Theorem):
  Théorème de sécurité des systèmes informatiques.

C

- **Cc** (copie carbone) :
  Etait à l' origine une méthode utilisant du papier carbone pour produire simultanément une ou plusieurs copies d'un document.
- **Cci** (Copie carbone invisible).
- **CERT** (Computer Emergency Response Team):
  Sont des centres d'alerte et de réaction aux attaques informatiques, destinés aux entreprises et/ou aux administrations, mais dont les informations sont généralement accessibles à tous.
- **CLUSIF** (Club de la sécurité de l'information français) :
  Est une association française d'entreprises et de collectivités réunis en groupes de réflexion autour de différents domaines de la sécurité de l'information : gestion des risques, droit, intelligence économique…
- **CLUSIS** (Association suisse de la sécurité des systèmes d'information).
- **CNIL** (Commission nationale de l'informatique et des libertés) :
  Est une autorité administrative indépendante française chargée de veiller à la protection des données à caractère personnel et de la vie privée. Elle a été créée par la loi n° 78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés.
- **CRLF LWP**(Chariot Return Linefeed) :
  Un retour chariot à la ligne.
- **CSMA/CD** (abréviation de Carrier Sense Multiple Access /Collision Detection) :
  Un protocole qui gère le partage de l'accès physique au réseau Ethernet.
- **CTSS** (Compatible Time-Sharing System) :
  Est l'un des premiers systèmes d'exploitation à temps partagé.

D

- **DCSSI** (Direction centrale de la Sécurité des systèmes d'information) :
  L'organisme interministériel officiel définissant les normes de la sécurité des systèmes d'information en particulier les normes sur l'évaluation et la certification des systèmes d'information.
- **DECnet** (Digital Equipment Corporation net):
  Le DECnet est un groupe de produits de communications de données, incluant une suite de protocole, développée et soutenue par la Digital Equipment Corporation (Digital). La première version du DECnet, 1975.
- **DNS** (Domain Name System) :
  Un système permettant d'établir une correspondance entre une adresse IP et un nom de domaine et, plus généralement, de trouver une information à partir d'un nom de domaine.
- **DOD** (Department Of Defense) :
  Est le ministère du gouvernement des États-Unis chargé de commander aux forces militaires du pays, en temps de guerre et en temps de paix.

- **DSL** (Digital Subscribe line):
  Est un mode de codage des données.

E

- **EBIOS** (Expression des Besoins et Identification des Objectifs de Sécurité) :
  Est une méthode d'évaluation des risques en informatique, développée par la Direction Centrale de la Sécurité des Systèmes d'Information.
- **EXIM** :
  Est un serveur de messagerie électronique (ou *Mail Transfer Agent* en anglais) utilisé sur de nombreux systèmes de type UNIX. La première version a été écrite en 1995 par Philip Hazel pour le service informatique de l'Université de Cambridge.

F

- **FAI** (Fournisseur d'accès à Internet) :
  Est un organisme (généralement une entreprise) offrant une connexion au réseau informatique Internet. Le terme anglais désignant un FAI est *Internet Service Provider* (ISP).
- **FDDI** (Fiber Distributed Data Interface):
  Est un type de réseau informatique LAN ou MAN permettant d'interconnecter plusieurs LAN à une vitesse de 100 Mbit/s sur de la fibre optique (ce qui lui permet d'atteindre une distance maximale de 200 km).
- **FEROS** (Fiche d'Expression Rationnelle des Objectifs de Sécurité des Systèmes d'information) :
  Est une méthode d'évaluation des risques développée par la Direction Centrale de la Sécurité des Systèmes d'Information.
- **FTP** (File Transfer Protocol):
  Est un protocole de communication dédié à l'échange informatique de fichiers sur un réseau TCP/IP.
- **FTPS** (Secure File Transfert Protocol) **:**
  Il permet au visiteur de vérifier l'identité du serveur auquel il accède grâce à un certificat d'authentification. Il permet également de chiffrer la communication.
- **FXP** (File Exchange Protocol):
  Est une variante non officielle du protocole FTP. Il permet le transfert direct de fichiers entre deux serveurs FTP sans passer par le client.

H

- **HLU**(Harrison, Ruzzo, Ullman model) :
  Modèle de sécurité qui fournit des politiques pour des droits d'accès changeants et des droites pour la création et la suppression des sujets et des objets. On le considère généralement comme l'un des modèles plus complexes de sécurité.

- **HTML** (Hyper Text Mark-Up Language):
  Est le langage informatique qui sert à écrire les pages Web.

I

- **ID** (Identificateur)
- **IDS** (Intrusion Detection System) :
  Est un mécanisme destiné à repérer des activités anormales ou suspectes sur la cible analysée (un réseau ou un hôte). Il permet ainsi d'avoir une action de prévention sur les risques d'intrusion.
- **IEEE** (Institute of Electrical and Electronics Engineers):
  Est l'organisation professionnelle qui compte le plus de membres, et possède différentes branches dans plusieurs parties du monde. Elle est constituée d'ingénieurs électriciens, d'informaticiens, de professionnels du domaine des télécommunications, etc.
  L'organisation a pour but de promouvoir la connaissance dans le domaine de l'ingénierie électrique.
- **IMAP** (Internet Message Access Protocol) :
  Est un protocole utilisé par les serveurs de messagerie électronique, fonctionnant pour la réception.
  Ce protocole permet de laisser les e-mails sur le serveur dans le but de pouvoir les consulter de différents clients e-mails ou web mail.
- **IPCE** (Interprocessus Communication Environnement) :
  Regroupent un ensemble de mécanismes permettant à des processus concurrents (ou distants) de communiquer. Ces mécanismes peuvent être classés en trois catégories :
  - les outils permettant aux processus de s'échanger des données ;
  - les outils permettant de synchroniser les processus, notamment pour gérer les sections critiques ;
  - les outils offrant directement les caractéristiques des deux premiers (ie : permettant d'échanger des données **et** de synchroniser des processus).
- **IPv4** (Internet Protocol version 4) :
  Est la première version d'IP à avoir été largement déployée, et forme encore la base (en 2007) de l'Internet.
  Elle utilise une adresse IP sur 32 bits, ce qui est un facteur limitant à l'expansion d'Internet puisque "seulement" 4 294 967 296 adresses sont possibles. Cette limitation conduit à la transition d'IPv4 vers IPv6, actuellement en cours de déploiement, qui devrait progressivement le remplacer.
- **IPv6** (Internet Protocol version 6) :
  Est le successeur du protocole IPv4 ; mais IPv4 est encore utilisé presque exclusivement sur Internet.
- **IRTF** (Internet Research Task Force)
- **ISO** (International Standards Organization) :
  Est un organisme de normalisation international composé de représentants d'organisations nationales de normalisation de 158 pays. Cette organisation crée en 1947 a pour but de produire des normes internationales dans les domaines industriels et commerciaux appelées normes ISO.

L

- **LAN** (Local Area Network) :
  Ce terme désigne un réseau informatique d'échelle géographique restreinte.
- **LWSP** (Linear White Space).

M

- **MAN** (Metropolitan Area Network) :
  Désigne un réseau étendu d'ordinateurs habituellement utilisé dans les campus ou dans les villes. Le réseau utilise généralement des fibres optiques .
- **Marion** (Méthode d'analyse de risques informatiques optimisée par niveau) :
  Une méthode d'audit, proposée depuis 1983 par le CLUSIF, visant à évaluer le niveau de sécurité informatique d'une entreprise. L'objectif est double:

  - situer l'entreprise auditée par rapport à un niveau jugé correct, et par rapport au niveau atteint par les entreprises similaires
  - identifier les menaces et vulnérabilités à contrer

- **Melisa** (Méthode d'évaluation de la vulnérabilité résiduelle des systèmes d'information) :
  Une méthode d'audit, proposée depuis 1985 par le CLUSIF, visant à évaluer le niveau de sécurité informatique d'une entreprise.
- **Méhari** (Méthode harmonisée d'analyse des risques) :
  A été développée et est proposée par le CLUSIF, visant à la sécurisation informatique d'une entreprise ou d'un organisme.
- **MIME** (Multipurpose Internet Mail Extensions) **:**
  Est un standard internet qui étend le format de données des courriels pour supporter des textes en différents codage de caractères autres que l'ASCII, des contenus non textuels, des contenus multiples, et des informations d'en-tête en d'autres codages que l'ASCII. Les courriels étant généralement envoyés via le protocole SMTP au format MIME, ces courriels sont souvent appelés courriels *SMTP/MIME*.
- **MIT** (Massachusetts Institute of Technology) :
  Est une institution de recherche et une université américaine, spécialisée dans les domaines de la science et de la technologie.
- **MMF** (Make money fast)
- **MSSP** (Managed Security Services Provider):
  Une société extérieure ou une consultation qui fournissent la conception de sécurité, le développement et les services de gestion.
- **MTA** (Mail Transfert Agent) :
  Est un agent de programme machine ou de logiciel qui transfère des messages de courrier électronique à partir d'un ordinateur à l'autre
- **MUA** (Mail User Agent) :
  Un client de messagerie est un logiciel qui sert à lire et envoyer des courriers électroniques. Les protocoles de communication utilisés pour la réception sont POP ou

IMAP. Le protocole SMTP est utilisé par la plupart des serveurs de courriel pour envoyer les messages.

N

- **NAT (Network Address Translation)** :
C'est un mécanisme qui est implémenté sur un routeur, permet de faire correspondre les adresses IP internes non-unique et souvent non routables d'un domaine vers un ensemble d'adresses externes uniques et routables. Ce mécanisme permet de pallier la carence d'adresses IPv4 sur Internet, le protocole IPv6 dispose d'un espace d'adressage plus important.

O

- **OMT** (Object Modeling Technique) :
Ce langage est l'œuvre de James Rumbaugh, qui est aussi l'un des pères d'UML (avec Grady Booch, fondateur du langage Booch, et Ivar Jacobson, fondateur du langage OOSE).
- **OOSE** (Object Oriented Software Engineering):
Est un langage de modélisation objet, comme OMT et Booch, avec lesquels il a fusionné pour donner UML.
- **OpenBSD** (Berkeley Software Distribution):
Désigne en informatique une famille de systèmes d'exploitation Unix, développés à l'université de Berkeley par un groupe de programmeurs : Bill Joy, Marshall Kirk McKusick, etc.
- **OQLF** (Office québécois de la langue française) :
est une institution publique du Québec créée le 24 mars 1961, au même moment que le Ministère des affaires culturelles du Québec (aujourd'hui le Ministère de la culture et des communications du Québec).
- **OSI** (Open System Interconnection):
Le modèle d'interconnexion en réseau des systèmes ouverts de l'ISO est un modèle de communications entre ordinateurs. Il décrit les fonctionnalités nécessaires à la communication et l'organisation de ces fonctions.
- **OSSIR** (Observatoire de la Sécurité des Systèmes d'Information et des Réseaux) :
Est une association française existant depuis 1996 et qui regroupe les utilisateurs intéressés par la sécurité des systèmes d'information et des réseaux.

P

- **PC** (Personal Computer) :
Le terme PC est aujourd'hui aussi utilisé pour désigner un ordinateur personnel, par opposition aux ordinateurs de tailles différentes.
- **PDA** (Personal Digital Assistant) :
Comme les agendas électroniques et les Ordinateurs de poches.
- **PKI** (Public Key Infrastructure) :
Est un ensemble de composants physiques (des ordinateurs, des équipements cryptographiques ou des cartes à puces), de procédures humaines (vérifications, validation) et de logiciels (système et application) en vue de gérer le cycle de vie des certificats numériques ou certificats électroniques.

- **POP** (Post Office Protocol) :
  Un protocole qui permet de récupérer les courriers électroniques situés sur un serveur
  De messagerie électronique. Cette opération nécessite une connexion à un réseau TCP/IP. Le port utilisé est le 110 .

R

- **RBL** (Real-time block list):
  Est une liste avec un format standard, elle contient des IP et Domaine suspect ainsi que des ID d'utilisateur (expéditeurs).
- **RNIS** (Réseau Numérique à Intégration de Services) :
  Réseau entièrement numérique dont le débit est plus important que celui des lignes téléphoniques classiques.
- **RSSI** (Responsable de la sécurité des systèmes d'information).

S

- **SAGE** (schéma d'aménagement et de gestion des eaux) :
  Est une unité hydrographique ou un système aquifère les grandes orientations définies par le SDAGE.
- **SBL** (Spamhaus Block List) :
  Est une base de données en temps réel des adresses IP des sources vérifiées de Spam et des opérations de Spam.
- **SDC** (System Development Corporation):
  Basé en Santa Monica, la Californie, était discutablement la première compagnie de logiciel d'ordinateur du monde.
- **SMTP** (Simple Mail Transfer Protocol) :
  Est un protocole de communication utilisé pour transférer le courrier électronique vers les serveurs de messagerie électronique
- **SNA** (Systems Network Architecture) :
  Est une architecture réseau en couches définie par IBM en 1974. SNA est désormais largement remplacé par TCP/IP
- **SOA** (Service-Oriented Architecture) :
  Est une forme d'architecture de médiation qui est un modèle d'interaction applicative qui met en œuvre des services (composants logiciels) :
  2- avec une forte cohérence interne (par l'utilisation d'un format d'échange pivot, le plus souvent XML).
  3- et des couplages externes « lâches » (par l'utilisation d'une couche d'interface interopérable, le plus souvent un service web.
- **SOAP** (Simple Object Access Protocol):
  Est un protocole de RPC orienté objet bâti sur XML.Il permet la transmission de messages entre objets distants, ce qui veut dire qu'il autorise un objet à invoquer des méthodes d'objets physiquement situés sur un autre serveur. Le transfert se fait le plus souvent à l'aide du protocole HTTP, mais peut également se faire par un autre

protocole, comme SMTP.

- **SONET** (Synchronous Optical Network) :
  Est un modèle de norme de transmission optique. C'est un protocole de couche 1 du modèle OSI.
- **SPARC** (Scalable Processor Architecture):
  Est une architecture d'ensemble d'instruction de microprocesseur de RISC à l'origine conçue en 1985 par Sun Microsystems.

- **SPIT** (Spam over Internet Telephony):
  Est le problème qui a suscité beaucoup d'attention des acheteurs et de commerçants. Quelques pandits ont commencé à se rapporter à lui. Comme les systèmes finis d'IP, (l'E-mail et d'autres applications d'Internet), sont susceptibles de l'abus par les parties malveillantes qui lancent des communications non sollicitées et non désirées. Les téléprospecteurs, et d'autres trompeurs de système de téléphone sont susceptibles de viser des systèmes de VoIP de plus en plus, en particulier si VoIP tend à supplanter la téléphonie conventionnelle.
- **SSH** (Secure Shell) :
  Est à la fois un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite toutes les trames sont chiffrées. Il devient donc impossible d'utiliser un sniffer pour voir ce que fait l'utilisateur. Le protocole **SSH** a été conçu avec l'objectif de remplacer les différents programmes rlogin, telnet et rsh.
- **SSI** (sécurité des systèmes informatiques) :
  Est l'ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires et mis en place pour conserver, rétablir, et garantir la sécurité de l'information et du système d'information.
- **SSII** (Sociétés de Services en Ingénierie Informatique, spécialisées en sécurité) :
  Est une société de services spécialisée en informatique.
- **SSL** (Secure Socket Layer) :
  Est un protocole de sécurisation des échanges sur Internet, développé à l'origine par Netscape.

T

- **TCP/IP** (Transmission Control Protocol/Internet Protocol) :
  La suite des protocoles Internet est l'ensemble des protocoles qui constituent la pile de protocole utilisés utilisée par Internet.
- **TCSEC** (Trusted Computer System Evaluation Criteria):
  Sont un ensemble de critères énoncés par le Département de la Défense, et permettant d'évaluer la fiabilité de systèmes informatiques centralisés.
- **TFTP** (Trivial File Transfer Protocol):
  Est un protocole simplifié de transfert de fichiers.
- **TLL** (Time To Live) :
  Le Time-to-live d'un paquet d'information transitant sur un réseau représente le nombre de secondes pendant lesquelles il est autorisé à passer dans les divers routeurs ponctuant sa route.
- **TLS** (Transport Layer Security) :
  Nouvelle nomination de SSL.

U

- **UDP** (User Datagram Protocol) :
  Un des principaux protocoles de télécommunication utilisé par Internet. Il fait partie de la couche transport de la pile de protocole TCP/IP.
- **UML** (Unified Modeling Language) :
  Un langage graphique de modélisation des données et des traitements.
- **USA** (United States of America)
- **USB** (Universal Serial Bus) :
  Est un bus informatique *plug-and-play* servant à brancher des périphériques informatiques à un ordinateur pour communiquer en série.
- **UTF-8** (Unicode Transformation Format-8) :
  Est un format de codage de caractères défini pour les caractères Unicode (UCS). Chaque caractère est codé sur une suite d'un à quatre octets. il a été conçu pour être compatible avec certains logiciels originellement prévus pour traiter des caractères d'un seul octet.
- **UUCP** (Unix to Unix Copy Protocol):
  Est un ensemble de programmes qui permettent à deux machines d'échanger des fichiers et d'exécuter des commandes sur la machine distante en passant par une ligne téléphonique (modem), mais aussi sur une couche TCP/IP (souvent à travers Ssh), voire via un câble série direct (null modem). Le mode modem reste cependant le cas de figure le plus utilisé.

V

- **VPN** (Virtual Private Network):
  Est une extension des réseaux locaux qui procure une norme de sécurité en télécommunications.

W

- **WAN** (Wide Area Network) :
  Un réseau informatique couvrant une grande zone géographique, typiquement à l'échelle d'un pays, d'un continent, voire de la planète entière. Le plus grand WAN est le réseau Internet.

X

- **XML** (Extensible Markup Language) :
  Est un langage informatique de balisage générique. Le World Wide Web Consortium (W3C), organisme qui promeut la compatibilité des technologies du World Wide Web en éditant des recommandations à valeur de standards industriels, recommande XML pour exprimer des langages de balisages spécifiques.

BIBLIOGRAPHIES.

1. **[CCNA1-1]**: "CCNA1", CISCO, Cisco Academy Program, V3.1.1.
2. **[ADR-2]** : "Architecture des réseaux", Didier Donzes, Université de valencienne, 2003
3. **[TINA-3]** : "TCP/IP Network administration", Graig HUNT, O'Reilly, 2000.
4. **[AMO-4]** : "Architecture Du Modèle OSI", Guy Pujolle, Université de Versailles.2002.
5. **[WPS-5]** : "Sécurité informatique", Wikipedia, http://fr.wikipedia.org ,2005.
6. **[SIN -6]** : "Sécurité informatique", www.vulnerability.com.2005.
7. **[SRS-7]** : "Sécurité réseaux et system", Jean François Pillot, DUNOD, 2006.
8. **[SAH-8]** : "Stratégies anti-hackers", Ryan Russell, Eyrolles. 2eme edition.2005.
9. **[VCE-9]** : "Vocabulaire du courriel électronique", Commission générale de Terminologie et de néologie, 22 aout 2003.
10. **[RFC-10]** : "Standard Pour le format des messages texte internet ARPA", DavidH.Crocker, Dept. EEUD, 13 Aout 1982.
11. **[WPC-11]** : "le courriel électronique", Wikipedia, http://fr.wikipedia.org/wiki/Courrier,2000.
12. **[WPP-12]** : "Le protocole SMTP", Wikipedia, http://fr.wikipedia.org/wiki/SMTP, 2001.
13. **[RFCf-13]** : "SIMPLE MAIL TRANSFER PROTOCOL", J.Postel/ISI, August 1982.fr.
14. **[RFCa-14]** : "SIMPLE MAIL TRANSFER PROTOCOL", J.Postel/ISI, August 1982.en.
15. **[SAM-15]** : "Linux Sendmail administration", Graig Hunt, Linux Edition, February 2001
16. **[DSP-16]** : "Definition of spam", http://www.mail-abuse.org.
17. **[GR-17]** : "Gartner Research", Joyce Graff& Maurene Grey, http://www.itap.purdue.edu/presentations,2003.
18. **[SMD-18]** : "Spam Definition", http://www.arobase.org/spam/comprendre.htm.
19. **[WST-19]** : "Terminologies de Spam", http://fr.wikipedia.org/wiki/Spam#Terminologie.
20. **[SC-20]** : "Le spam en chiffres", internet monde, http://www.journaldunet.com/cc/03 /spam.shtml.
21. **[SMH-21]** : "SendMail Historique", Redhat, http://www.europe.redhat.com/ Documentation/sendmail.php3.
22. **[WSM-22]** : "SendMail Definition", http://fr.wikipedia.org/wiki/Sendmail.

23. **[SAB-23]** : "Spam Assassin", Alan Schwartz, O'Reilly, July 2004.
24. **[MSI-24]** : "Messagerie d'internet", Jean Paul Gauthier & Philipe leca, UREC, 1999.
25. **[SAS-25]** : "Apache Project", Rudy Montier, http://www.supinfo projects.com/ Spamassassin, 2005.
26. **[PMG-26]** : "Proventia M10 User Guide", Internet Security System, Mars 2006.
27. **[HPW-27]** : "Honey Pot For Linux", Roger Grimes, Apress, June 2005.
28. **[HP-28]** : "Honey Pot", M Dizon, EIG, December 2003.
29. **[HPF-29]** : "Honey Pot Farms", Franck Veysset, EUROSEC, 2004.
30. **[HHN-30]** : "Honey Pot & Honey Net", http://www.authsecu.com/honeypots-honeynet.
31. **[LRH-31]** : "Linux Redhat Entreprise Edition", http://www.commentcamarche.net/linuxR.
32. **[SHP- 32]** : "Spamhaus Project", http://www.spamhaus.org/organization/index.lasso.

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab, Blida
USDB.

Faculté des sciences.
Département informatique.

المكتبة

Mémoire pour l'obtention
d'un diplôme d'ingénieur d'état en informatique.
Option : SI.

Sujet :

Conception et réalisation d'un système HoneyPot-Mail.

Présenté par : Belhaoues Chafik.
Belkas Mahmoud.

Promotrice : M^me^ SOUAMI.
Encadreur : Ali Zaiter.
Karim Harmouche.
Wassim Nakadi.

Organisme d'accueil : S.A.R.L.IntraTech.

Soutenue le: date soutenance, devant le jury composé de :

Mlle Boustia, MA.	**Président**
Mr Zahar, MA	**Examinateur**
Mr Ferfera, MA	**Examinateur**

- 2006/2007 -

MIG-004-158-1

Remerciements

Tout d`abord nous remercions الله عز وجل *pour nous avoir guidé vers le*

Bon chemin de la lumière et du savoir, pour nous avoir donné du courage et

De la volonté afin de pouvoir réaliser ce mémoire.

Nous exprimons nos sincères remerciements à nos parents pour tout ce

Qu`ils ont pu nous apporter.

Nous tenons à remercier les membres du jury pour avoir eu l'amabilité

D'accepter et d`apprécier notre travail.

Nous tenons à remercier notre promotrice Mme Souami feriel pour

Son aide, sa patience, sa disponibilité, et sa compréhension.

A tous les enseignants de la faculté des sciences exactes de Blida et

Surtout les enseignants du département de l'informatique.

A tous les employés (directeur, ingénieurs, secrétaires commerciale et comptable)
De

La Société Algéro-libanaise la SARL IntraTech.

Enfin, nous remercions, de tout cœur, tous ceux qui ont contribué de prés ou

De loin à la réalisation de ce travail.

Dédicace

Je dédie ce mémoire

A mes très chers parents pour leur soutien et leur patience durant toute ma carrière d'étude,

Pour leur bienveillance, leurs efforts et encouragement durant mes études,

A mon frère et ma sœur et sa petite famille,

A tous les membres de ma famille proches et éloignés

A mon binôme Chafik et toute sa famille

A mes très chers amis

A mes collègues de travail algériens et libanais

A toute ma promotion,

A toute personne qui me connaît.

Mahmoud Belkas.

Dédicace

Ce travail est dédié :

A ma très chère Maman pour son soutien durant

toute ma carrière, et dans toute ma vie

Pour sa bienveillance, ses efforts constants dans mes études,

Pour son encouragement,

A mon frère REDHA,

A toute la famille ALLALI

(Papa, Houria, Redha japoni, mehdi, Kenza, Radhia)

A mon binôme Mahmoud et à toute sa famille,

A tous mes collègues de travail algériens et libanais

A toute ma promotion,

A tous mes amis que j'aime et qui m'aiment.

Chafik Belhaoues.

TABLE DES MATIERES

Introduction générale 1

CHAPITRE 1 : Les réseaux.

1. Introduction 3
2. Evolution des connections réseaux et internet 3
3. Topologies des réseaux 4
4. Les réseaux étendus 5
5. Intranets et extranets 6
6. La bande passante 7
7. Le standard TCP/IP 7
8. Conclusion 9

CHAPITRE 2 : Sécurité informatique.

1. Introduction 10
2. Evaluation des risques 10
   2.1 Risques humains 11
   2.2 Risques techniques 11
   2.3 Programmes malveillants 11
   2.4 Techniques d'attaque par messagerie 12
   2.5 Attaques sur les mots de passe 12
3. Critères d'évaluation 12
4. Méthodes d'évaluation 13
5. Moyens de sécurisation d'un système 13
   5.1 Conception globale 13
   5.2 Défense en profondeur 14
   5.3 Politique de sécurité 14
   5.4 Plan de continuité d'activité 14
   5.5 Moyens techniques 14
6. Acteurs 15
   6.1 Organismes officiels 15

6.2 Sociétés commerciales 15
6.3 Associations 15
6.4 Groupes de hackers 15
7. Conclusion 15

CHAPITRE 3 : Le Protocole SMTP et Mail.

1. Introduction 16
2. Le Courier Electronique – Mail 16
2.1 Contenu du mail 17
2.2 Acheminement du mail 17
2.3 Système de copie et de copie invisible 18
3. Le protocole SMTP 18
4. Les procédures SMTP 20
4.1 Ouverture et fermeture de connexion 20
4.2 Emission et postage 20
4.3 Emission du courrier 21
4.4 Réémissions (Forwarding) 23
4.5 Vérification de boites et expansion de liste de diffusion 24
4.6 Relai de messages 25
4.7 Echange des roles 26
5. Domaines 26
6. Aspects de sécurité 26
7. Conclusion 26

CHAPITRE 4 : Le Spam « pourriel ».

1. Introduction 27
2. Terminologie et Définitions 27
2.1 Définition du spam 27
2.2 Caractéristiques de spam 28
2.3 Identification du SPAM 29
3. Modèles de SPAM 30
3.1 Le SPAM anglophone 30
3.2 Le SPAM amateur 30
3.3 SPAM « erreur de destination » 30
3.4 Le Spam vicieux 31
3.5 Le spammeur 31
4. Cibles du pourriel (SPAM) 31

4.1 Par courrier électronique 31
4.2 Par message de forum de discussion 32
4.3 Par des fenêtres intruses de Windows 33
4.4 Par systèmes vocaux 33
4.5 Référencement abusif (Spamdexing) 33
4.6 Dans les blogs (splog) 34
4.7 Sur les wikis 35
5. Spam en chiffre 35
6. Profil de Spammeur 36
7. Conclusion 36

CHAPITRE 5 : Outils utilisés.

1. Introduction 37
2. Send Mail 37
2.1 Introduction 37
2.2 Historique 38
2.3 Avantages et inconvénients 38
2.4 Configuration 38
2.5 Le rôle de Send Mail 38
3. Spam assassin 39
3.1 Introduction 39
3.2 Architecture logique de Spam Assassin 39
3.3 Installation 40
3.4 Configuration 40
1. Configuration de base 40
2. Configuration de Razor 41
3. Configuration de Pyzor 41
4. Configuration de DCC 41
5. Configuration des filtres Bayesiens 42
4. Serveur FTP 42
4.1 Mode actif 43
4.2 Mode passif 43
5. Linux Redhat 46
5.1 Notions de noyau et de distribution 47
5.2 Caractéristique du système 47
5.3 Inconvénients 48

CHAPITRE 6 : Etude des HoneyPot.

1. Définition 49
2. Avantages et Inconvénients 50
2.1 Avantages 50
2.2 Inconvénients 50
3. Les types d'HoneyPot 51
3.1 Honeyd: Low-interaction HoneyPot 51
3.2 Honeynet: High-interaction HoneyPot 52
1. Contrôle des données 53
2. Capture des données 53
3. Collection des données 53
4. Génération de HoneyNet 54
3.3 HoneyPot virtuel 56
4. Les HoneyPot virtuels avec Honeyd 57
4.1 Introduction 57
4.2 Fonctionnement 57
1. Fichier de configuration 57
2. HoneyPot client 57
3. Valeurs de HoneyPot 57
5. Conclusion 58

CHAPITRE 7 : Format de message.

1. Introduction 59
2. Composants génériques 59
3. Analyse lexicale 60
3.1 Description générale 60
3.2 Les champs d'en-tête 60
3.3 Définition des champs d'en-tête 61
4. Analyse Syntaxique 61
4.1 Champs spécifiques 61
4.2 Éclaircissements 63
1. Faire suivre (Forwarding) 63
2. Reçu « received » 64
3. Le champ « From » 64
4. Le champ « replay to » 64
5. Le champ « BCC (blind carbon copy) » 65
6. Message-ID 65
5. Le cryptage 65
6. Spécifications de l'adresse 65

6.1 Syntaxe 65
6.2 Sémantique 66
6.3 Les domains 66
6.4 Spécifications d'abréviation de domaine 67
6.5 Les termes du domaine 68
6.6 Boite aux lettres multiples 68
6.7 Spécification explicite du chemin 69
6.8 Adresses réservées 69
7. Conclusion 69

CHAPITRE 8 : conception de B2SAM.

1. Introduction 70
1.1 UML 70
1.2 UP 71
1.2.1 Phase de Réalisation d'UP 72
2. résentation de l'application 72
2.1 La 1ère partie : « piège HoneyPot » 74
1. Serveurs mails virtuels volontairement vulnérables 74
2. Module de récupération d'e-mails 74
2.2 La 2ème partie : « B2SAM centrale» 75
1. Module de lecture et de décomposition 75
2. Module d'analyse 76
1. Analyse de l'adresse IP de l'expéditeur 76
2. Analyse du domaine de l'expéditeur 76
3. Analyse du sujet 77
4. Analyse de la date d'envoie des messages 77
5. Analyse du corps 78
3. Attribution du score 80
4. Module de catégorisation 81
5. Base de données temporaire 81
6. Module de mise-à-jour 81
3. Les cas d'utilisations 82
3.1 Définition des Acteurs 82
3.2 Les diagrammes de cas d'utilisation 82
1. Envoi d'E-mail 82
2. Stockage des E-mail 83
3. Ouverture d'une session ftp 84

4. Affectation dans la base de données 84
5. Analyse des E-mail 85
6. Mise à jour des Anti-spam 86
3.3 Diagramme de cas d'utilisation global 87
3.4 Digramme de transition 88
3.5 Diagramme de séquence et de communication 89
1. L'envoi d'un mail 90
2. Ouverture d'une session ftp 90
3. Module de subdivision 90
4. Module d'analyse et de catégorisation 91
5. Manipulation de RBL 93
4. Diagramme de classe 94
5. Diagramme de déploiement 95
5.1 Définition 95
5.2 Le diagramme de déploiement 95
6. Conclusion 95

CHAPITRE 9 : Implémentation et mise en œuvre.

1. Introduction 96
2. Choix du langage de programmation 96
3. Implémentation 97
3.1 Script d'ouverture d'une session ftp 97
3.2 Module de chargement des e-mails 98
3.3 Module d'analyse entre deux tables de base de données 100
3.4 Module d'analyse d'adresse IP 101
3.5 Module d'analyse du domaine de l'expéditeur 101
3.6 Module d'analyse du sujet de message 101
3.7 Module d'analyse du corps des E-mail 102
4. Réalisation de l'application 102
4.1 Modification de la configuration 103
4.2 Génération du format standard 104
5. Test effectué 104
6. Conclusion 108

Conclusion générale 109

Table des matières.

Annexe 110

Glossaire 114

Bibliographie 123

Liste des Figures.

1. **Fig. I.1** Différentes topologies physiques.....4
2. **Fig. I.2** Raccordement de différentes topologies des réseaux.....5
3. **Fig. I.3** Les réseaux étendus.....6
4. **Fig. I.4** Les intranets et les extranets.....6
5. **Fig. I.5** Modèle TCP /IP.....8
6. **Fig. I.6** Les protocoles du modèle TCP/IP.....9

7. **Fig. III.1** Acheminement du courriel.....17
8. **Fig. III.2** Dialogue client/serveur utilisant le SMTP.....18
9. **Fig. III.3** Scénario du dialogue SMTP.....19
10. **Fig. III.4** Ouverture de session SMTP.....20
11. **Fig. III.5** Fermeture de session SMTP.....20
12. **Fig. III.6** Séquence d'envoi d'un courrier.....23
13. **Fig. III.7** Utilisation des réponses SMTP.....23
14. **Fig. III.8** Vérification de nom d'utilisateur.....24
15. **Fig. III.9** Expansion d'une liste d'utilisateurs.....24
16. **Fig. III.10** Message de notification "undeliverable mail ".....25

17. **Fig. IV.1** Pourcentage de spam.....28
18. **Fig. IV.2** Exemple de spam anglophone.....30
19. **Fig. IV.3** Spam amateur.....30
20. **Fig. IV.4** Spam erreur de destination.....30
21. **Fig. IV.5** Spam vicieux.....31
22. **Fig. IV.6** Spam « ceci n'est pas un spam1 ».....31
23. **Fig. IV.7** Spam « ceci n'est pas un spam2 ».....31
24. **Fig. IV.8** comment évité le pollupostage.....32
25. **Fig. IV.9** Référencement abusif spamdexing.....34

26. **Fig. V.1** Rôle de Send Mail [usrs.cnrs France 2005].....38
27. **Fig. V.2** Architecture logique de SpamAssassin.....39
28. **Fig. V.3** Connexion Data en mode Actif. 44

29. **Fig.VI.1** Le schéma de HoneyNet.....53
30. **Fig.VI.2** 1ère génération de HoneyNet.....54
31. **Fig.VI.3** 2ème génération de HoneyNet.....55
32. **Fig.VI.4** HoneyPot Virtuel.....56
33. **Fig.VI.5** Architecture du HoneyPot Virtuel hybride.....56
34. **Fig.VI.6** Architecture de mode de fonctionnement du HoneyC.....57

35. **Fig.VII.1** Champ d'en-tête « TO » model 1.....60
36. **Fig.VII.2** Champ d'en-tête « TO » model 2.....60
37. **Fig.VII.3** Champ d'en-tête « TO » model 3.....60
38. **Fig.VII.4** Champ d'en-tête « TO » model 4.....61

39. **Fig. VIII.1** Model de cycle de vie d'un logiciel.....71
40. **Fig. VIII.2** Architecture du logiciel B2SAM.....73
41. **Fig. VIII.3** Cas d'utilisation d'envoi d'E-mail.....83
42. **Fig. VIII.4** Cas d'utilisation de stockage des E-mail dans un Répertoire.....83
43. **Fig. VIII.5** Cas d'utilisation d'ouverture de session FTP et Transfert d'E-mail.....84
44. **Fig. VIII.6** Cas d'utilisation d'affectation dans la base de données.....85
45. **Fig. VIII.7** Cas d'utilisation d'analyse des E-mail.....86
46. **Fig. VIII.8** cas d'utilisation de mise à jour des anti-spam.....86
47. **Fig. VIII.9** Diagramme de cas d'utilisation global.....87
48. **Fig. VIII.10** Diagramme de transition d'envoi d'E-mail.....88
49. **Fig. VIII.11** Diagramme de transition d'ouverture de session.....88
50. **Fig. VIII.12** Digramme de transition d'analyse d'E-mail.....89
51. **Fig. VIII.13** Diagramme de séquence d'envoi d'E-mail.....90
52. **Fig. VIII.14** Diagramme de séquence d'ouverture d'une session FTP.....90
53. **Fig. VIII.15** Diagramme de séquence de subdivision.....91
54. **Fig. VIII.16** Diagramme de séquence D'analyse d'E-mail.....92
55. **Fig. VIII.17** Diagramme de séquence d'envoi d'E-mail.....93
56. **Fig. VIII.18** Diagramme de Classe.....94
57. **Fig. VIII.19** Diagramme de déploiement.....95

58. **Fig. IX.1** Commandes d'automatisation de la tache.....97
59. **Fig. IX.2** Script d'ouverture d'une session ftp.....98
60. **Fig. IX.3** Méthode de vérification de mot de passe.....98
61. **Fig. IX.4** Code de vérification de chemin.....99
62. **Fig. IX.5** Code de chargement d'e-mails.....99
63. **Fig. IX.6** Code de découpage des e-mails.....100
64. **Fig. IX.7** Code de l'analyse entre deux tables.....100
65. **Fig. IX.8** Code de l'analyse de l'adresse IP.....101
66. **Fig. IX.9** Code de l'analyse du domaine.....101
67. **Fig. IX.10** Code de l'analyse du sujet.....101
68. **Fig. IX.11** Code de l'analyse du corps du message.....102
69. **Fig. IX.12** Interface principale de l'application.....102
70. **Fig. IX.13** Fiche de modification de la configuration.....103

71. **Fig. IX.14** Fichier de format standard généré.....104
72. **Fig. IX.15** Environnement de test du HoneyPot Mail.....105
73. **Fig. IX.16** Aperçus des messages reçus de différents domaines.....106
74. **Fig. IX.17** Table d'analyse dans un environnement de test.....107
75. **Fig. IX.18** L'invite de mot de passe.....107
76. **Fig. IX.19** Mot de passe invalide.....108
77. **Fig. A.1** Les couches du model OSI..... 111

Liste des Tables.

1. **Table IV.1** Monde : Origines du spam.....35
2. **Table IV.2** Répartition du spam par catégories en 2007.....35
3. **Table IV.3** Etats-Unis : nombre de spams quotidiens selon une étude menée en novembre 2006.....36

4. **Table VI.1** Caractéristiques des deux types d'HoneyPot.....50

5. **Table VII.1** Les Champs d'en-tête.....61
6. **Table VII.2** Les Marqueurs lexicaux. 62
7. **Table VII.3** spécification syntaxique de l'adresse. 65

8. **Table VIII.1** pourcentage des Spam en fonction de la langue utilisée.....79
9. **Table VIII.2** Définition des acteurs.....82
10. **Table VIII.3** Rôle des acteurs d'envoi d'E-mail.....82
11. **Table VIII.4** Rôle des acteurs de stockage des E-mail.....83
12. **Table VIII.5** Rôle des acteurs d'ouverture d'une session FTP.....84
13. **Table VIII.6** Rôle des acteurs d'affectation dans la base de données.....84
14. **Table VIII.7** Rôle des acteurs d'analyse des E-mail.....85
15. **Table VIII.8** Rôle des acteurs de mise à jour de l'Anti-spams.....86

Introduction générale.

The open doors are always welcome sign.

– Belhaoues Chafik.

Le courrier électronique est devenu, avec les années, un service essentiel presque pour tout le monde. Qui ne possède pas une adresse de courrier aujourd'hui?

Avec le temps, il semblait évident que de nombreuses menaces verraient le jour et qu'elles se propageraient à travers ce médium de communication. Certaines personnes, toujours prêtes à faire de l'argent, ont vu, dans le courrier, une excellente façon de rejoindre des clients potentiels. La sollicitation par courrier électronique (*spam*) était née. De plus, les auteurs de virus ont profité de ce vecteur d'attaque en l'utilisant comme tremplin pour une meilleure propagation virale. Les fraudeurs ont aussi gagné leur part du lot, surtout depuis les récentes menaces d'hammeçonage.

Dans le but de se prémunir contre de telles attaques, les administrateurs de systèmes implantèrent diverses technologies afin de protéger leurs utilisateurs contre ce type de menaces. Mais est-ce que ces programmes sont aussi sécurisés que les administrateurs le souhaiteraient? Ces bouts de codes sont-ils prêts à affronter tout le contenu malsain retrouvé aujourd'hui sur l'autoroute électronique?

La recherche qui a été menée dans le cadre de ce mémoire vise à étudier théoriquement et pratiquement un système dont l'objectif est de piéger les spammeurs, pour les dévoiler et les maitriser. Il vous sera présenté le problème lié à l'implantation de technologies d'analyse et de filtrage au niveau du courrier, ainsi que des solutions de mise à jour pour les anti-spams.

Motivations et objectifs :

Notre travail a précisément comme objectif d'étudier l'intérêt d'utiliser la technologie du HoneyPot orientée mail c.-à-d. avoir comme domaine d'étude le courrier électronique afin de proposer des solutions pour mieux sécuriser un tel système. Les objectifs que nous nous somme fixés, afin de réaliser ce travail, sont les suivants :

1. Etudier les différentes stratégies utilisées par les spammeurs.
2. Mettre en œuvre un système HoneyPot-mail basé sur des serveurs mail volontairement vulnérables afin de s'en servir comme des centres de collecte.
3. Proposer une architecture d'un système de collecte, d'analyse et de mise à jour pour les solutions anti-spams dans des plateformes hétérogènes.
4. Conception d'un système d'analyse appelé **B2SAM** pour la détection d'éventuel spam et générer des formats standards pour les solutions de mise à jour des anti-spams en utilisant UML comme approche, UP comme modèle et C++ comme langage de programmation.
5. Mise en œuvre du Système **B2SAM.**

Méthodologie suivie :

Afin d'atteindre nos objectifs, nous avons commencé par un état de l'art des notions de base sur les réseaux et la sécurité des réseaux et systèmes et les différents types d'attaques, puis nous avons détaillé le courrier électronique ainsi que le protocole SMTP, ensuite on a présenté une étude approfondie sur les SPAM afin de bien les authentifier par la suite, et les différents outils utilisés dans notre solution HoneyPot Mail, puis une étude sur les HoneyPot déjà implémenté, on a aussi spécifié le format standard du message pour se familiariser avec les messages afin de les analyser par le B2SAM suivis par une conception détaillée et une mise en œuvre en utilisant le langage C++. Finalement nous avons fait des tests pour prouver la fiabilité de notre solution.

Présentation du mémoire :

La suite du mémoire sera constituée de neufs chapitres.

Chapitre 1 : dédie aux réseaux informatiques et leurs structures.
Chapitre 2 : dans ce chapitre on met en évidence la notion de sécurité réseaux et son impacte.
Chapitre 3 : est consacré au protocole SMTP et aux différentes formes du mail.
Chapitre 4 : on a expliqué le spam et ces différentes manifestations.
Chapitre 5 : nous présentons dans ce chapitre tous les outils utilisés.
Chapitre 6 : ce chapitre décrit les différents systèmes HoneyPot.
Chapitre 7 : est dédie aux spécifications du format de message.
Chapitre 8 : nous présentons ici la conception de notre système HoneyPot mail.
Finalement le Chapitre 9 montre, en utilisant le langage de programmation C++, la possibilité de mettre en œuvre l'architecture proposée.

CHAPITRE I.

Les réseaux.

The future is the Network.

– Jonathan Schwartz.

1. Introduction

Internet est un réseau constitué d'une multitude de réseaux de plus ou moins grande taille reliés entre eux. Les ordinateurs personnels y constituent les sources et les destinations de l'information.

La connexion à Internet englobe trois éléments: physique, logique et applications. Le premier élément (physique) implique de relier l'ordinateur, par un modem ou une carte réseau, à un réseau. La connexion physique assure le transfert des signaux entre ordinateurs. L'élément logique utilise des normes (appelées protocoles). Un protocole est un ensemble formel de règles et de conventions régissant le mode de communication des unités réseau entre elles. Les protocoles TCP/IP (*Transmission Control Protocol/Internet Protocol*) sont les plus largement utilisés sur Internet. Le dernier élément est composé d'applications servant à interpréter et traiter les données (ex : un navigateur Internet, tel qu'Internet Explorer ou Netscape, est nécessaire pour afficher le code HTML sous forme de page Web).

2. Evolution des connections réseaux et Internet

Apparus au début des années 1960, les modems servaient à connecter des terminaux passifs à un ordinateur central à une vitesse de 300 bits par seconde (bits/s). Dans les années 1970, le prix des PC est devenu plus abordable et les systèmes BBS (*Bulletin Board System*) sont apparus. Ces systèmes offraient des services informatisés d'échange d'informations, auxquels les utilisateurs pouvaient se connecter notamment pour afficher des messages et y répondre. La vitesse de 300 bits/s était acceptable puisqu'elle était supérieure à la capacité de lecture à l'écran ou de frappe au clavier des utilisateurs **[ADR-2]**.

Au début des années 1980, l'utilisation des BBS a connu un développement exponentiel et la vitesse de 300 bits/s s'est rapidement avérée insuffisante pour le transfert des fichiers graphiques volumineux. Dans les années 1990, le débit des modems est passé à 9 600 bits/s. En 1998, la vitesse standard actuelle de 56 000 bits/s, ou 56 Kbits/s a été atteinte. Les services à haut débit utilisés dans les environnements d'entreprise, tels que la technologie DSL (*Digital Subscriber Line*) et les connexions par modem câble, sont ensuite devenus accessibles aux particuliers. Ces services ne requièrent plus d'équipement coûteux ni de ligne téléphonique supplémentaire. Toujours actifs, ils offrent un accès instantané à Internet et ne requièrent pas d'établir une nouvelle connexion à chaque session. Ces avancées technologiques, garantes d'une fiabilité et d'une flexibilité accrues, ont simplifié la mise en place de connexions partagées à Internet dans les réseaux domestiques ou de petites entreprises **[ADR-2].**

Les accès à Internet s'opèrent le plus souvent à travers des réseaux locaux (LAN) d'entreprise. Ces réseaux locaux au même titre que les réseaux distants, assurent la connectivité des ordinateurs. La manière dont les ordinateurs sont connectés est déterminée par la topologie réseau choisi.

3. Topologies des réseaux

La topologie réseau est définie aussi bien par la topologie physique que par la topologie logique. La topologie physique est la configuration proprement dite du câblage ou du média. La topologie logique est définie par la manière dont les ordinateurs accèdent aux médias pour envoyer des données.

Les topologies physiques sont les suivantes :

- **Topologie de bus :** un câble sert de backbone unique. Tous les hôtes se connectent directement à ce backbone.
- **Topologie en anneau :** chaque hôte est connecté à son voisin. Le dernier hôte se connecte au premier. Cette topologie crée un anneau physique de câble.
- **Topologie en étoile :** tous les câbles sont raccordés à un point central.
- **Topologie en étoile étendue :** relie des étoiles individuelles en connectant les concentrateurs ou les commutateurs.
- **Topologie hiérarchique :** similaire à une topologie en étoile étendue, elle relie les concentrateurs à un ordinateur qui contrôle le trafic sur la topologie.
- **Topologie maillée :** pour garantir une protection maximale contre l'interruption de service, on connecte tous les ordinateurs entre eux. C'est la topologie la plus coûteuse.

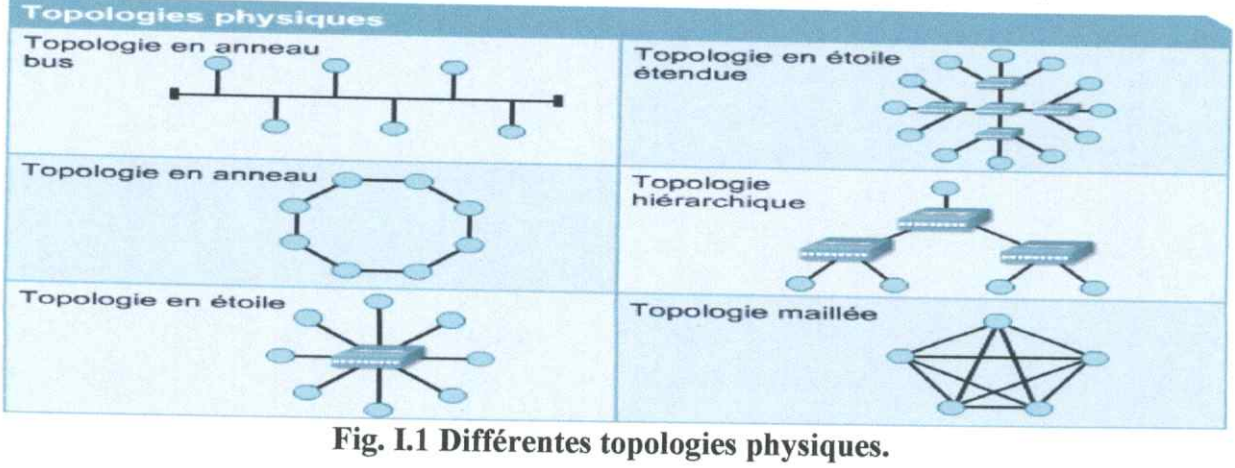


Fig. I.1 Différentes topologies physiques.

Les topologies logiques sont le **broadcast** et le **passage de jeton.** La topologie logique d'un réseau détermine de quelle façon les hôtes communiquent sur le média. L'utilisation d'une topologie de broadcast indique que chaque hôte envoie ses données à tous les autres hôtes sur le média du réseau. Pour la seconde topologie, un jeton électronique est transmis de façon séquentielle à chaque hôte. Dès qu'un hôte reçoit le jeton, cela signifie qu'il peut transmettre des données sur le réseau. Si l'hôte n'a pas de données à transmettre, il passe le jeton à l'hôte suivant et le processus est répété. Token Ring et FDDI (*Fiber Distributed Data Interface*) sont deux exemples de réseaux qui utilisent le passage de jeton. Arcnet est une variante de Token Ring et de FDDI. Il s'agit d'un passage de jeton sur une topologie de bus.

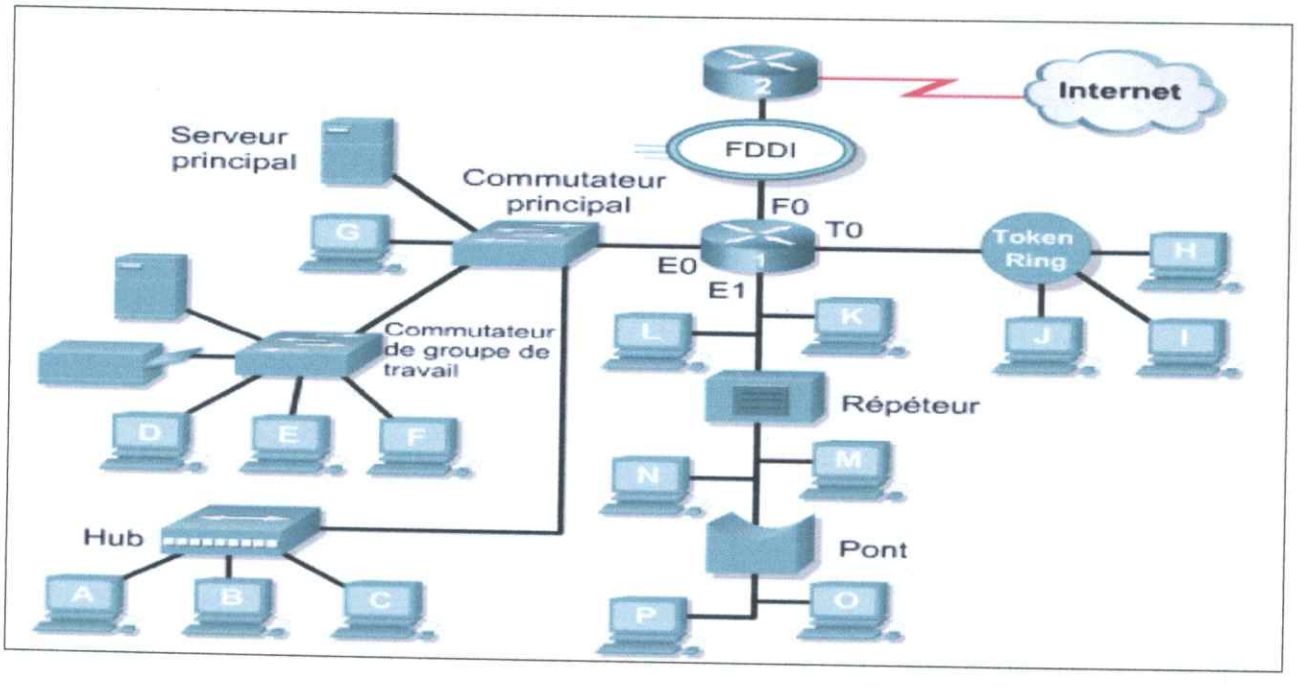


Fig. I.2 Raccordement de différentes topologies des réseaux.

Le schéma de la figure (**I.3**) illustre différentes topologies raccordées. Il représente un réseau de complexité moyenne, comme celui qu'on retrouve habituellement dans une école ou une petite entreprise. C'est un réseau LAN qui est connecté au réseau Internet. Si une entreprise se compose de bureaux distants géographiquement, elle doit pouvoir relier ses ordinateurs au travers de réseaux étendues.

4. Les réseaux étendus

Les réseaux étendus interconnectent des réseaux locaux géographiquement dispersés, ils permettent aux entreprises de communiquer sur de grandes distances. Les réseaux étendus permettent le partage d'ordinateurs, imprimantes et autres équipements raccordés.

Les réseaux étendus utilisent diverses technologies telles que :

- Modems.
- RNIS (*Réseau Numérique à Intégration de Services*).
- DSL (*Digital Subscriber Line*).
- Frame Relay.
- SONET (*Synchronous Optical Network*).

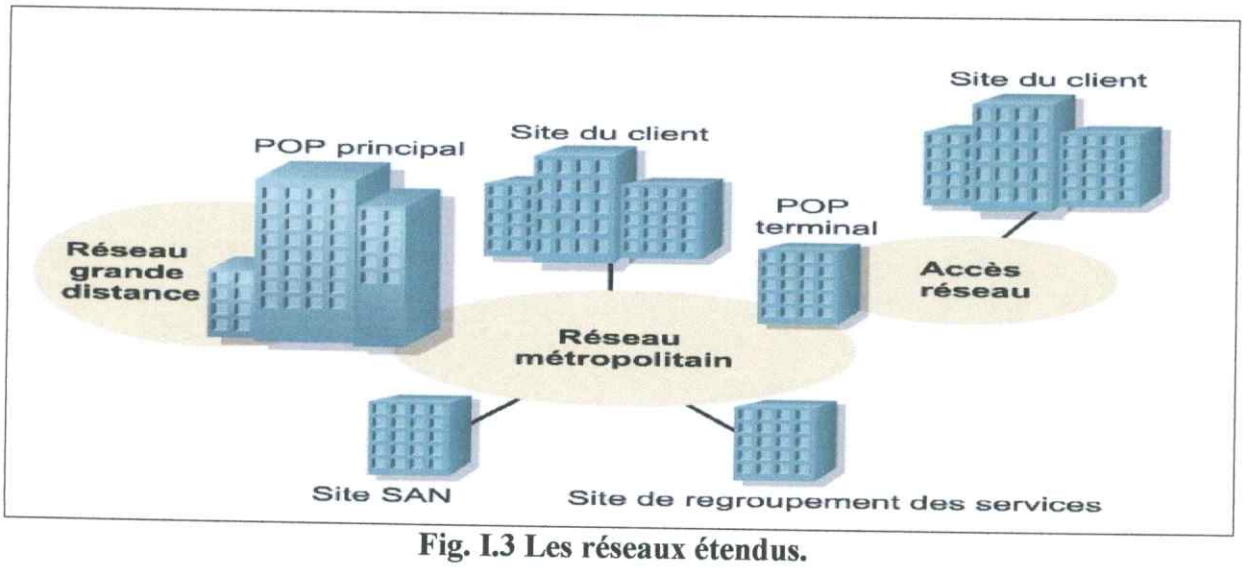


Fig. I.3 Les réseaux étendus.

5. Intranets et Extranets

Un Intranet est une configuration de réseau local conçue pour autoriser les utilisateurs ayant des privilèges d'accès à disposer des ressources internes de l'organisation. Au sein d'un intranet, les serveurs Web sont installés dans le réseau LAN permettant d'utiliser le navigateur comme interface commune d'accès aux informations, telles que les données financières par exemple.

Le terme Extranet fait référence à des applications et à des services qui sont basés sur intranet, mais qui utilisent un accès étendu et sécurisé pour les utilisateurs ou entreprises externes. Un Extranet est une extension de l'Intranet, avec une interaction sécurisée entre les entreprises participantes et leurs intranets respectifs.

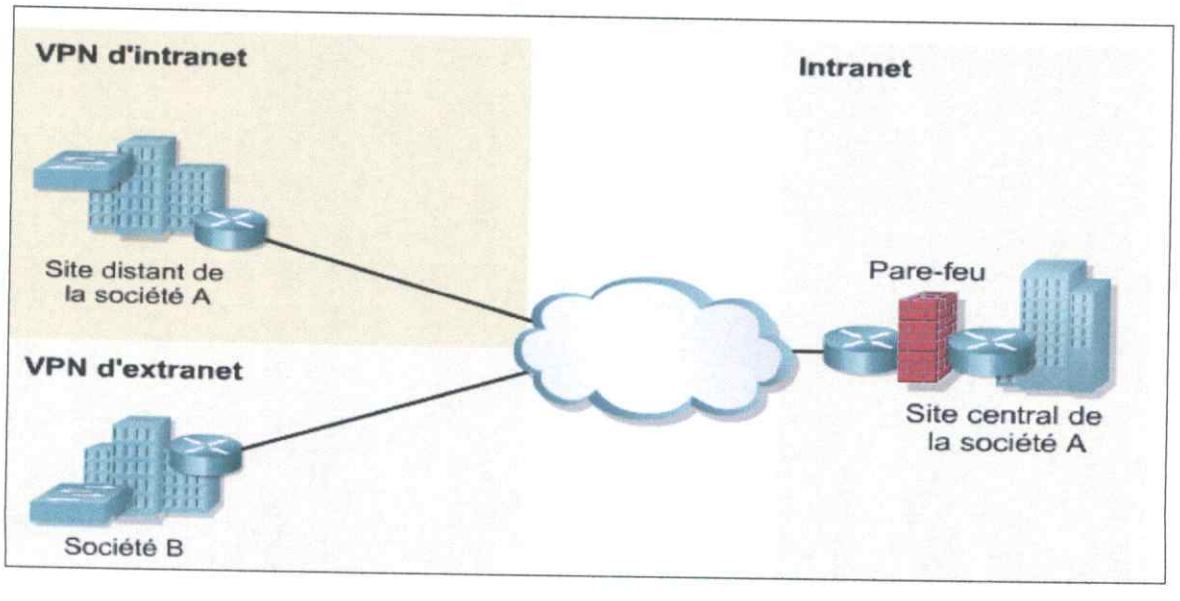


Fig. I.4 Les Intranets et les Extranets.

6. La bande passante

Pour que le partage de ressources dans un réseau (Intranet ou Extranet, ou Internet) soit utilisable, il faut que cet échange n'induise pas des retards et de l'attente. La bande passante représente la quantité d'informations qui peut transiter sur une connexion réseau en un temps donné. La bande passante est donc un élément fondateur pour les réseaux.

- La bande passante est limitée par les technologies utilisées pour placer des informations sur les médias. Par exemple, la bande passante d'un modem conventionnel est limitée à environ 56 kbits/s, à la fois par les propriétés physiques des fils téléphoniques à paires torsadées et par la technologie du modem.
- La bande passante n'est pas gratuite. Pour les connexions WAN, il est généralement nécessaire d'acquérir de la bande passante auprès d'un fournisseur de services.
- La demande en bande passante ne cesse de croître. La mise sur un réseau de contenus multimédias tels que la vidéo et l'audio en continu nécessite une quantité considérable de bande passante. Les systèmes de téléphonie IP augmentent encore le besoin en bande passante.

La disponibilité de diverses technologies assure la possibilité de connecter des ordinateurs et autres organes de transmission entre eux. Cependant, le mode de communication est tributaire des protocoles de communication et de transfert des données. Il existe deux grandes approches prédominantes aujourd'hui dans ce domaine : OSI et TCP/IP. La seconde est celle qui a été largement adoptée par le réseau des réseaux à ce jour. Cet ensemble de protocoles assure que, quel que soit le média physique, quelle que soit la structure des machines, quel que soit le système d'exploitation considéré, et quel que soit les applications de communication déployées : la communication entre machines hétérogènes est garantie.

7. Le standard TCP/IP

C'est à la demande du ministère de la Défense américain DoD qu'a été créé TCP/IP. Le Dod désirait disposer d'un réseau pouvant résister à tout, même à une guerre nucléaire. Dans un monde connecté par différents types de médias de communication tels que les fils de cuivre, micro-ondes, fibres optiques et liaisons satellite, le DoD souhaitait une transmission de paquets capable d'aboutir à coup sûr et sous n'importe quelle condition.

Ce problème de conception conduit à la création du modèle TCP/IP en tant que norme ouverte. Cela voulait dire que n'importe qui pouvait l'utiliser, ce qui contribua à accélérer le développement de TCP/IP en tant que référent.

Le modèle TCP/IP comporte les quatre couches suivantes:

- La couche application.
- La couche transport.
- La couche Internet.
- La couche d'accès au réseau.

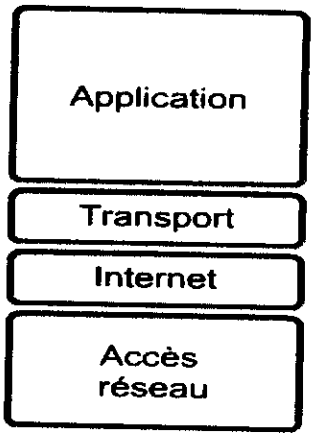


Fig. I.5 Modèle TCP /IP.

La couche application gère les questions de représentation, le code et le contrôle du dialogue.

La couche transport est chargée des questions de qualité de service touchant la fiabilité, le contrôle de flux et la correction des erreurs. L'un de ses protocoles, TCP (*Transmission Control Protocol - protocole de contrôle de transmission*), permet de créer des communications réseau fiables et présentant un taux d'erreurs peu élevé. En effet, le protocole TCP, orienté connexion[1], maintient un dialogue entre l'ordinateur source et l'ordinateur de destination pendant qu'il prépare les informations de couche application en unités, appelées segments.

La couche Internet assure l'envoi et la réception des segments TCP, de et vers n'importe quel réseau. Le protocole IP (*Internet Protocol*) gère les paquets qui arrivent ainsi au réseau de destination indépendamment du chemin qu'ils ont emprunté pour y parvenir.

La relation entre ces deux protocoles est IP et TCP est essentielle : IP pointe le chemin pour les paquets, tandis que TCP assure un transport fiable.

La couche d'accès au réseau, ou encore couche hôte-réseau concerne tous les composants, à la fois physiques et logiques, qui sont nécessaires pour créer une liaison physique. Elle comprend les technologies de réseau physique et les liaisons de données du modèle OSI.

Les protocoles de couche application les plus courants sont **[TINA-3]**:

- FTP (*File Transfer Protocol*).
- HTTP (*HyperText Transfer Protocol*).
- SMTP (*Simple Mail Transfer Protocol*).
- DNS (*Domain Name System*).
- TFTP (*Trivial File Transfer Protocol*).

[1] Un protocole orienté connexion ne signifie pas qu'il existe un circuit entre les ordinateurs en communication. Ce type de fonctionnement indique qu'il y a un échange de segments de couche *application* entre les deux ordinateurs hôtes afin de confirmer l'existence logique de la connexion pendant un certain temps.

Les protocoles les plus courants de la couche transport sont notamment:

- TCP (*Transport Control Protocol*).
- UDP (*User Datagram Protocol*).

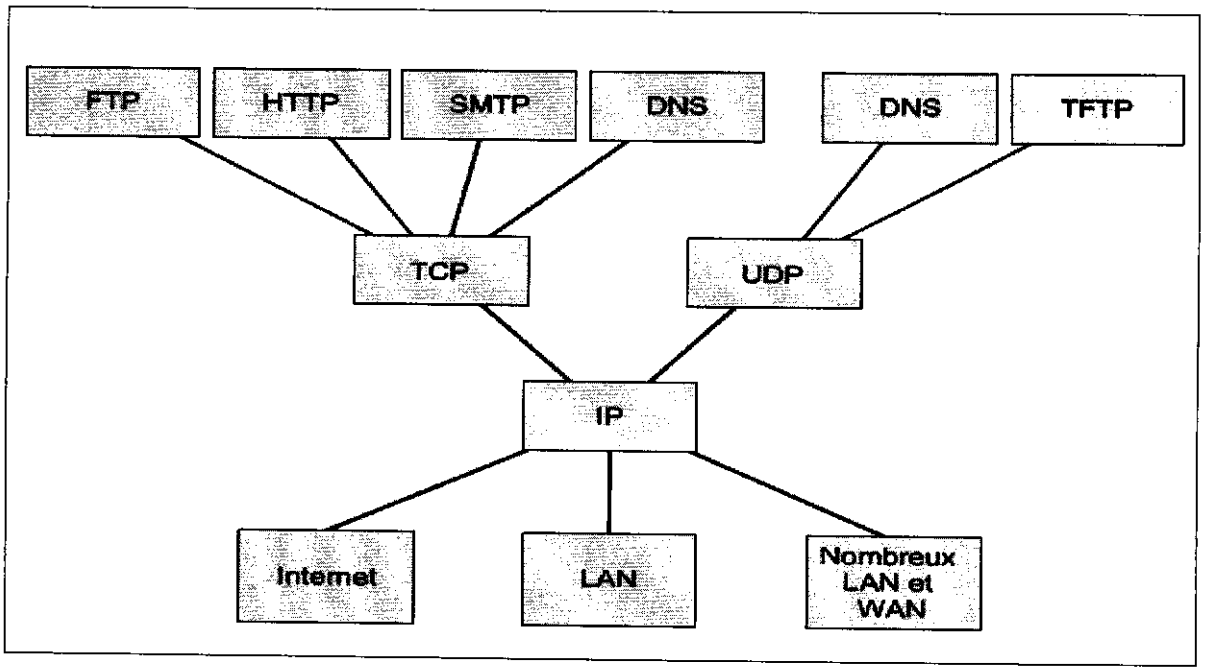
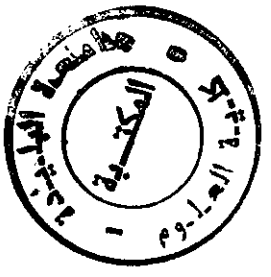


Fig. I.6 Les protocoles du modèle TCP/IP.



8. Conclusion

Comme toutes industries, les systèmes informatiques doivent êtres reliés entre eux d'une façon efficace toute en respectant certains topologies et protocoles pour mieux garder l'aspect objectif de communications entre les systèmes, et pour cela certains standards on été établis pour fournir de tels mécanismes et sans négliger quelques inconvénients non négligeable comme l'insécurité de ses systèmes.

CHAPITRE II.

Sécurité Informatique.

On the other hand,
we cannot ignore efficiency.

– Jon Bentley.

1. Introduction

La sécurité des systèmes d'information regroupe l'ensemble des moyens techniques et organisationnels nécessaires mis en place pour conserver et garantir la sécurité de l'information et du système d'information dans son ensemble.

Les systèmes d'information comprennent tous types de données. Ils sont de ce fait sensible à diverses menaces susceptibles d'altérer ou de détruire l'information (risque pour *l'intégrité de l'information*), ou de la révéler à des tiers qui ne doivent pas en avoir connaissance (risque pour la *confidentialité de l'information*), ou bien de porter atteinte à sa disponibilité (risque pour la *disponibilité du système*).

2. Évaluation des risques

Il est important de mesurer les risques, non seulement en fonction de la probabilité ou de la fréquence de leurs survenances, mais aussi en mesurant leurs effets possibles. En effet, ces effets, selon les circonstances et le moment où ils se manifestent, peuvent avoir des conséquences non négligeables ou catastrophiques :

- données irrémédiablement perdues ou altérées, ce qui les rend inexploitables.
- données ou traitements durablement indisponibles, pouvant entraîner l'arrêt d'une production ou d'un service.

- divulgation d'informations confidentielles ou erronées pouvant profiter à des sociétés concurrentes ou nuire à l'image de l'entreprise.

A l'ère de la généralisation des traitements et des échanges en masse, on imagine assez bien l'impact que pourraient avoir des événements majeurs comme, par exemple, une panne de serveurs de grande ampleur ou la saturation du réseau Internet pendant plusieurs heures.

2.1 Risques humains

Les risques humains sont importants, même s'ils sont le plus souvent ignorés ou minimisés :

- **La malveillance** : Certains utilisateurs peuvent mettre en péril le système d'information en y introduisant des virus ou de mauvaises informations. De même il est relativement aisé d'ajouter des fonctions cachées permettant de détourner à son profit de l'information.
- **L'ingénierie sociale** : (*social engineering*) est une méthode pour obtenir d'une personne des informations confidentielles. Elle consiste à se faire passer pour quelqu'un que l'on n'est pas (en général un administrateur) et de demander des informations personnelles (nom de connexion, mot de passe, etc.) en inventant un quelconque prétexte (problème dans le réseau, heure tardive, … etc.). Elle peut se faire par simple communication téléphonique, ou par mail.
- **L'espionnage** : l'espionnage, emploie les même moyens, ainsi que bien d'autres, pour obtenir des informations sur des activités concurrentes, procédés de fabrication, projets en cours, futurs produits, politique de prix, clients et prospects, etc. **[WPS-5]**.

2.2 Risques techniques

Les risques techniques sont tout simplement ceux liés aux défauts et pannes inévitables que connaissent tous les systèmes matériels et logiciels **[SIN -6]** :

- **Incidents liés au matériel** : la plupart des composants électroniques peuvent comporter des défauts et bien entendu finissent un jour ou l'autre par tomber en panne. Certaines de ces pannes sont assez difficiles à déceler car sont intermittentes ou rares.
- **Incidents liés au logiciel** : la complexité croissante des systèmes d'exploitation et des programmes génèrent inévitablement des erreurs que les meilleures méthodes de travail et les meilleurs outils de contrôle ou de test ne peuvent pas éliminer en totalité.

2.3 Programmes malveillants

Un logiciel malveillant (*malware*) est un logiciel développé dans le but de nuire à un système informatique. Les principaux types de programmes malveillants sont :

- **Le virus** : programme se dupliquant sur d'autres ordinateurs.
- **Le ver** : (*worm*) : exploite les ressources d'un ordinateur afin d'assurer sa reproduction.
- **Le wabbit** : programme qui se réplique par lui-même (n'est ni un virus, ni un ver).
- **Le cheval de Troie** (*trojan*) : programme à apparence légitime (voulue) qui exécute des routines nuisibles sans l'autorisation de l'utilisateur.
- **La porte dérobée** (*backdoor*) : ouvreur d'un accès frauduleux sur un système informatique, à distance.
- **Le logiciel espion** (*spyware*) : collecteur d'informations personnelles sur l'ordinateur d'un utilisateur sans son autorisation, et en envoyant celles-ci à un organisme tiers.

- **L'enregistreur de frappe** (*keylogger*) : programme généralement invisible installé sur le poste d'un utilisateur et chargé d'enregistrer à son insu ses frappes clavier.
- **L'exploit** : programme permettant d'exploiter une faille de sécurité d'un logiciel.
- **Le rootkit** : ensemble de logiciels permettant généralement d'obtenir les droits d'administrateur sur une machine, d'installer une porte dérobée, de truquer les informations susceptibles de révéler la compromission, et d'effacer les traces laissées par l'opération dans les journaux système.

2.4 Techniques d'attaque par messagerie

En dehors des nombreux programmes malveillants qui se propagent par la messagerie électronique, il existe aussi des attaques spécifiques :

- **Le pourriel** (*Spam*) : est un courrier électronique non sollicité, la plupart du temps de la publicité. Ils encombrent le réseau, et font perdre du temps à leurs destinataires.
- **L'hameçonnage** (*phishing*) : est un courrier électronique dont l'expéditeur se fait généralement passer pour un tiers (ex : organisme financier) et demande au destinataire de fournir des informations confidentielles.
- **Le canular informatique** (*hoax*) : est un courrier électronique qui incite généralement le destinataire à retransmettre le message à ses contacts sous divers prétextes.

2.5 Attaques sur les mots de passe

Les attaques sur les mots de passe consistent à effectuer de nombreuse tentatives jusqu'à trouver le bon mot de passe. Deux méthodes sont largement utilisées :

- L'attaque par dictionnaire (*dictionary attack*) : le mot de passe testé est pris dans une liste prédéfinie contenant les mots de passe les plus courants et aussi des variantes de ceux-ci à l'envers, avec un chiffre à la fin, *etc.*
- L'attaque par force brute (*brutal attack*) : toutes les possibilités de mot de passe sont testés jusqu'à trouver la bonne solution (par exemple de "aaaaaa" jusqu'à "ZZZZZZ" pour un mot de passe composé strictement de six caractères alphabétiques).

3. Critères d'évaluation

La vulnérabilité peut s'évaluer suivant plusieurs critères :

- **Disponibilité** : l'élément a-t-il besoin d'une disponibilité importante pour ne pas nuire au système d'information ?
- **Intégrité** : l'élément a-t-il besoin d'une intégrité importante pour ne pas nuire au fonctionnement du système d'information ?
- **Confidentialité** : l'élément a-t-il besoin d'une confidentialité importante pour ne pas nuire au système d'information ?
- **Imputabilité (*Preuve*)** : l'élément a-t-il besoin d'une gestion de la preuve importante pour ne pas nuire à l'organisation ?
- **Contrôle d'accès** : l'élément a-t-il besoin d'un contrôle d'accès important pour ne pas nuire au système d'information ?

Une fois les éléments sensibles déterminés, les risques sur chacun de ces éléments peuvent être évalués en fonction des menaces qui pèsent sur les éléments à protéger. Il faut pour cela évaluer **[SRS-7]** :

- l'impact de chaque menace sur chaque élément.
- la probabilité d'occurrence de ces impacts (*potentialité*).

4. Méthodes d'évaluation

Différentes méthodes d'évaluation des risques sur le système d'information existent, parmi elles **[SRS-7]** :

- la méthode Ebios (*Expression des besoins et identification des objectifs de sécurité*), développée par la DCSSI.
- la méthode Feros (*Fiche d'expression rationnelle des objectifs de sécurité des systèmes d'information*), développée par la DCSSI.
- la méthode Méhari (*Méthode harmonisée d'analyse des risques*), développée par le CLUSIF.
- la méthode Marion (*Méthode d'analyse de risques informatiques optimisés par niveau*), développée par le CLUSIF.
- la méthode Melisa (*Méthode d'évaluation de la vulnérabilité résiduelle des systèmes d'information*).

Paradoxalement, la définition d'indicateurs mesurables, et permettant de définir par la suite des objectifs dans le temps, raisonnables à atteindre, s'avère délicate. S'il s'agit d'indicateurs de performances, on peut désigner comme indicateurs les états d'installation d'outils ou de procédures, mais les indicateurs de résultats sont plus complexes à définir et à apprécier, se sont les « *alertes virales* ». **[SRS-7]**

5. Moyens de sécurisation d'un système

5.1 Conception globale

La sécurité d'un système d'information peut être comparée à une chaîne de maillons, plus ou moins résistants. Elle est alors caractérisée par le niveau de sécurité du maillon le plus faible. La sécurité du système d'information est ainsi abordée dans un contexte global par :

- la sensibilisation des utilisateurs aux problématiques de sécurité, ou dans certains cas « prise de conscience » (*awareness*).
- la sécurité des données, liée aux questions d'interopérabilité, et aux besoins de cohérence des données en univers réparti.
- la sécurité des réseaux.
- la sécurité des systèmes d'exploitation.
- la sécurité des télécommunications.
- la sécurité des applications (débordement de tampon), cela passe par exemple par la programmation sécurisée.
- la sécurité physique, soit la sécurité au niveau des infrastructures matérielles « stratégie de reprise ».

Pour certains, la sécurité des données est à la base de la sécurité des systèmes d'information, car tous les systèmes utilisent des données, et les données communes sont souvent très hétérogènes (format, structure, occurrences, ...).

5.2 Défense en profondeur

Tout droit sorti d'une pratique militaire ancienne et toujours d'actualité, le principe de défense en profondeur revient à sécuriser chaque sous-ensemble du système, et s'oppose à la vision d'une sécurisation du système uniquement en périphérie.

5.3 Politique de sécurité

La sécurité des systèmes d'information se résume souvent à garantir les droits d'accès aux données et ressources d'un système, en mettant en place des mécanismes d'authentification et de contrôle. Ces mécanismes permettent d'assurer que les utilisateurs des ressources possèdent uniquement les droits qui leurs ont été octroyés.

La sécurité informatique doit toute fois être étudiée de telle manière à ne pas empêcher les utilisateurs de développer les usages qui leur sont nécessaires, et de faire en sorte qu'ils puissent utiliser le système d'information en toute confiance. C'est la raison pour laquelle il est nécessaire de définir dans un premier temps une politique de sécurité, c'est-à-dire :

- élaborer des règles et des procédures.
- installer des outils techniques dans les différents services de l'organisation.
- définir les actions à entreprendre en cas de détection d'une intrusion.
- sensibiliser les utilisateurs aux problèmes liés à la sécurité.

La politique de sécurité est donc l'ensemble des orientations suivies par une entité en terme de sécurité. À ce titre, elle doit être élaborée au niveau de la direction de l'organisation concernée, car elle concerne tous les utilisateurs du système. Le rôle de l'administrateur informatique est donc de faire en sorte que les ressources informatiques et les droits d'accès à celles-ci soient en cohérence avec la politique de sécurité retenue. De plus, étant donné qu'il est le seul à connaître parfaitement le système, il lui revient de faire remonter les informations concernant la sécurité à sa direction, éventuellement de la conseiller sur les stratégies à mettre en œuvre, ainsi qu'être le point d'entrée concernant la communication aux utilisateurs des problèmes et recommandations en terme de sécurité. **[SRS-7]**

5.4 Plan de continuité d'activité

Puisque aucune entreprise ne peut se passer de l'outil informatique, d'où la nécessité absolue d'un plan de continuité de l'informatique. Ce plan a pour but la reprise des activités après une catastrophe et ce, de la manière la plus efficace possible tout en garantissant la survie de l'entreprise.

5.5 Moyens techniques

Contrôle les accès aux systèmes informatiques par exemple :

- Surveillance du réseau : supervision, sniffer, système de détection d'intrusion.

- Sécurité applicative : séparation des privilèges, audit de code, rétro-ingénierie.
- Emploi de technologies ad-hoc : pare-feu, anti-logiciels malveillants (antivirus, antipourriel, antiespiogiciel, antitrojan).
- Cryptographie, authentification forte, liste de control d'accès, stéganographie, infrastructure à clés publiques ...etc.

6. Acteurs

6.1 Organismes officiels

Il existe des organismes officiels chargés d'assurer des services de prévention des risques et d'assistance aux traitements d'incidents. Ces CERT (*Computer Emergency Response Team*) sont des centres d'alerte et de réaction aux attaques informatiques, destinés aux entreprises et/ou aux administrations, mais dont les informations sont généralement accessibles à tous.

6.2 Sociétés commerciales [SAH-8]

- Éditeurs d'antivirus.
- Éditeurs de pare-feux.
- Éditeurs de firewall applicatif Web.
- Éditeurs de firewall pour les Web Services (SOA, XML, SOAP).
- Éditeurs de solutions d'Authentification forte.
- Éditeurs de solutions de contrôle des périphériques (USB, Wifi, ...etc.).
- Éditeurs de solutions PKI.
- Éditeurs de solutions Anti-Spam.
- Éditeurs de solutions de chiffrement.
- Éditeurs de solutions de corrélation des événements de sécurité.
- Constructeurs d'Appliance de Sécurité pour la détection et prévension d'intrusions.
- Éditeurs de systèmes d'exploitation basés sur la sécurité : OpenBSD, SE Linux.
- SSII : Sociétés de Services en Ingénierie Informatique, spécialisées en sécurité.
- MSSP: Managed Security Services Provider.

6.3 Associations [SAH-8]

- CLUSIF (Club de la Sécurité des Systèmes d'Information Français).
- OSSIR (Observatoire de la Sécurité des Systèmes d'Information et des Réseaux).
- CLUSIS (Association suisse de la sécurité des systèmes d'information).

6.4 Groupes de hackers

Il est à noter que les hackers, spécialisés en intrusions ou tentatives d'intrusions, ne sont pas tous des *pirates*, comme certains se plaisent à le dire. Aussi, les *lamers*, *script kiddies*, *defaceurs*, et *Curious Joe*, ne sont pas des hackers, mais simplement des personnes qui prétendent l'être. Les hackers sont des passionnés d'informatique, experts des systèmes et des réseaux, capables de développer leurs propres outils, et leur motivation n'est pas forcément délictueuse.

7. Conclusion

Malgré l'impacte primordial de la sécurité sur les systemes il reste toujours des failles persistantes exploitées par la communauté des hackers même sur les systèmes hautement sécurisés, ce qui soutient l'hypothèse que la sécurité informatique à 100% n'est jamais atteinte.

CHAPITRE III.

Protocole SMTP et Mail.

The road goes ever on and on.

– Bilbo Baggins.

1. Introduction

L'utilisation du courrier électronique (mail) est devenue un élément du quotidien de l'être humain. Pour offrir une simplicité et une transparence de manipulation du courrier électronique dans le monde de l'Internet, on a besoin de définir tous les procédés que le courrier utilise pour arriver à destination sans altération.

Pour arriver à mettre en œuvre un tel système, plusieurs organisations et établissements ont, depuis l'apparition de l'Internet, élaborés des standards et des protocoles spécifiant tous les détails de ce type de trafic Internet.

2. Le Courier Electronique - Mail

Le courrier électronique a prit forme en 1965 en tant que moyen de communication entre utilisateurs d'ordinateurs à temps partagée (*time-sharing mainframe*), ce qui explique bien que le courrier électronique existait bien avant Internet. Le Q32 du SDC et le CTSS du MIT ont été les premiers systèmes de messagerie électronique **[VCE-9]**

Ces systèmes furent rapidement étendus en réseau, permettant aux utilisateurs de transmettre des messages à travers différents ordinateurs. Le système AUTODIN peut être considéré

comme le premier à autoriser l'échange de courriers entre ordinateurs en 1966, le système SAGE avait des fonctionnalités similaires quelque temps auparavant.

Ray Tomlinson fut le premier à introduire l'utilisation du signe @ pour séparer les noms d'utilisateurs et de leurs machines en 1972. Ses premiers programmes de courrier *sndmsg* et *readmail* ont joué un rôle important dans le développement du courrier électronique dont ARPANET augmenta de manière significative la popularité.

Pour recevoir et consulter des mails, il est indispensable de posséder une adresse électronique et un client de messagerie MUA (*Mail User Agent*), ce dernier pouvant être simplement accessible via un navigateur Web. Afin d'assurer l'interopérabilité, l'acheminement des courriers est régi par plusieurs standards, que ce soit pour son routage, ou pour son contenu.

2.1 Contenu du mail

À l'origine, le courrier est un document qui contient du texte. Les caractères qui peuvent être utilisés étaient d'abord ASCII sur 7 bit, puis des encodages régionaux. Aujourd'hui, certains logiciels supportent également l'UTF-8 (*Unicode Transformation Format-8*), ce qui permet d'augmenter la quantité de caractères utilisables simultanément.

Il est aujourd'hui également possible de mettre en valeur le contenu du texte d'un courrier, grâce aux fonctionnalités HTML, afin d'ajouter un formatage aux documents. Il faut noter que HTML, UTF-8 et les caractères régionaux ne sont pas toujours interopérables, en fonction du logiciel de messagerie utilisé par le destinataire et de sa localisation géographique.

2.2 Acheminement du mail

L'acheminement des courriers est régi par plusieurs standards : SMTP est dédié à l'envoi d'un message, POP et IMAP servent aussi à délivrer les messages manipulés par les serveurs pour leurs lecture.

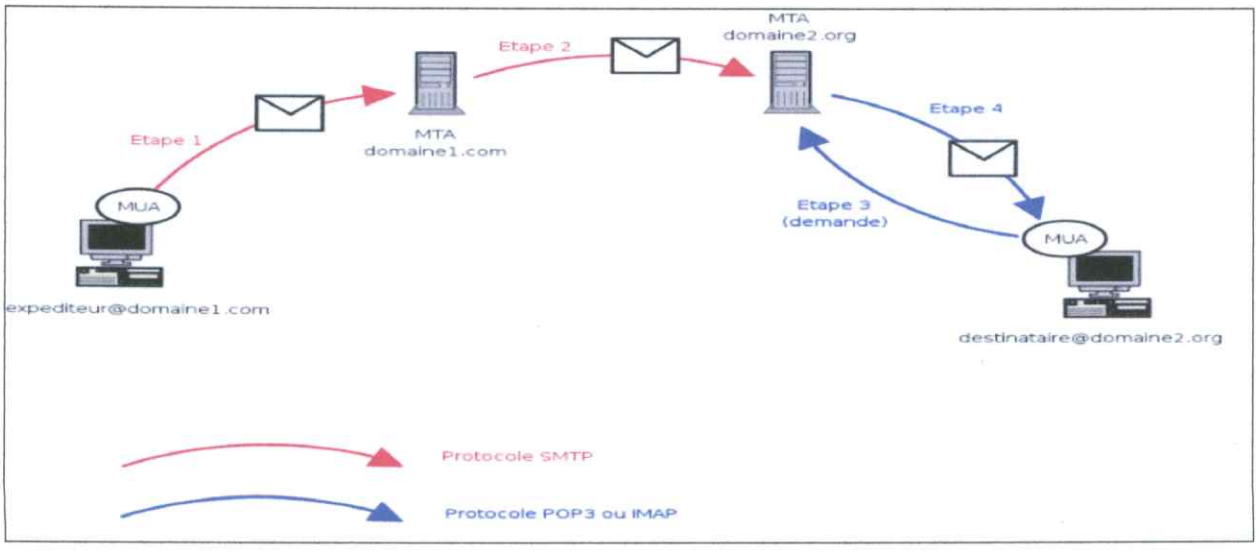


Figure. III.1 Acheminement du mail.

La circulation du courrier suit la procédure suivante :

- Le MUA de l'expéditeur envoie par SMTP le message à un serveur de courrier (celui de son fournisseur d'accès en général) ou MTA, Mail Transfert Agent
- Le premier MTA envoie le message au MTA du destinataire (celui de son domaine *domaine2.org*) qui le stocke en attente de consultation.
- Le destinataire, par l'intermédiaire de son MUA, demande à son serveur de courrier les nouveaux messages.
- Le serveur envoie le message au MUA du destinataire.

2.3 Système de copie et de copie invisible

Le courrier peut être envoyé à plusieurs destinataires :

- les destinataires principaux.
- les destinataires en copie « Cc » (copie *carbone)*
- les destinataires en copie invisible « Cci » (*Copie conforme invisible).*

Pour un envoi vers plusieurs destinataires ne se connaissant pas, il est d'usage de se servir du *Cci.* De plus, les virus et vers informatiques qui exploitent les adresses de courrier trouvées dans les carnets d'adresses des ordinateurs se trouvent bloqués par cette disposition. **[RFC-10]**

3. Le protocole SMTP (Simple Mail Transfer Protocol)

Le Simple Mail Transfer Protocol SMTP, est un protocole de communication utilisé pour transférer le courrier électronique vers les serveurs de messagerie électronique. Largement utilisé au début des années 1980, il est alors un complément à l'**UUCP** (*Unix to Unix Copy Protocol*), celui-ci étant plus adapté pour le transfert de courriers électroniques entre des machines dont l'interconnexion est intermittente. Le SMTP, de son côté, fonctionne mieux lorsque les machines qui envoient et reçoivent les messages sont interconnectées en permanence.

SMTP est un protocole assez simple : il commence par spécifier le ou les destinataires d'un message, puis l'expéditeur du message, puis après avoir vérifié leur existence, le corps du message est transféré. On peut tester un serveur SMTP en utilisant Telnet sur le port 25.

```
220 smtp.xxxx.xxxx SMTP Ready
HELO client
250 Hello client, pleased to meet you
MAIL FROM:<user@xxxx.xxxx>
250 <user@xxxx.xxxx> ... Sender ok
RCPT TO:<user2@yyyy.yyyy>
250 recipient ok.
DATA
354 Enter mail, end with "." on a line by itself
Ceci est un test pour un article de « B2SAM de la thèse de l'ingéniorat »
portant sur SMTP.
.
250 Ok?
QUIT
221 Closing connection
Connection closed by foreign host.
```

Fig. III.2 Dialogue client/serveur utilisant SMTP.

Comme le protocole SMTP utilisait du texte en ASCII (7 bits), il ne fonctionnait pas pour l'envoi de n'importe quels octets dans des fichiers binaires. Pour pallier ce problème, des standards comme **MIME** ont été développés pour permettre le codage des fichiers binaires au travers de SMTP. SMTP ne permet pas de récupérer à distance des courriels arrivés dans une boîte aux lettres sur un serveur. Les standards **POP** et **IMAP** ont été créés dans ce but.

Une fonctionnalité importante de SMTP est sa capacité à relayer des courriers dans des environnements de service de transport. Un service de transport dispose d'un environnement de communication interprocessus (**IPCE**). Un IPCE peut avoir une portée limitée à un seul réseau, étendue à plusieurs réseaux, ou réduite à un sous ensemble d'un réseau. Un processus peut communiquer directement avec un autre processus à travers toute paire d'IPCE se reconnaissant l'une l'autre. Le courrier électronique est une application, ou simplement une utilisation des mécanismes de communication interprocessus.

Le design de SMTP est basé sur le modèle suivant de communication **[WPP-12]:**

1. Suite à une requête de l'utilisateur du courrier, l'émetteur SMTP établit une communication bidirectionnelle vers un récepteur SMTP.
2. Une fois le canal de transmission établi, l'émetteur SMTP envoie une commande **MAIL** mentionnant l'émetteur d'un courrier. Si le récepteur SMTP peut accepter le courrier, il répondra par un message OK. L'émetteur SMTP envoie alors une commande **RCPT** identifiant le destinataire de ce courrier. Si le récepteur SMTP peut accepter un courrier pour ce destinataire, alors il répondra par un message OK ; sinon, il répond par un message refusant le courrier pour ce destinataire (mais n'annulant totalement pas la transaction de courrier).
3. L'émetteur SMTP et le récepteur SMTP pourront mentionner plusieurs destinataires. Une fois cette négociation effectuée, l'émetteur SMTP envoie le contenu du courrier, en le terminant par une séquence spéciale. Si le récepteur SMTP traite avec succès la

réception du contenu du courrier, il répondra par un message OK. Le dialogue est volontairement un dialogue pas à pas, à étapes verrouillées.

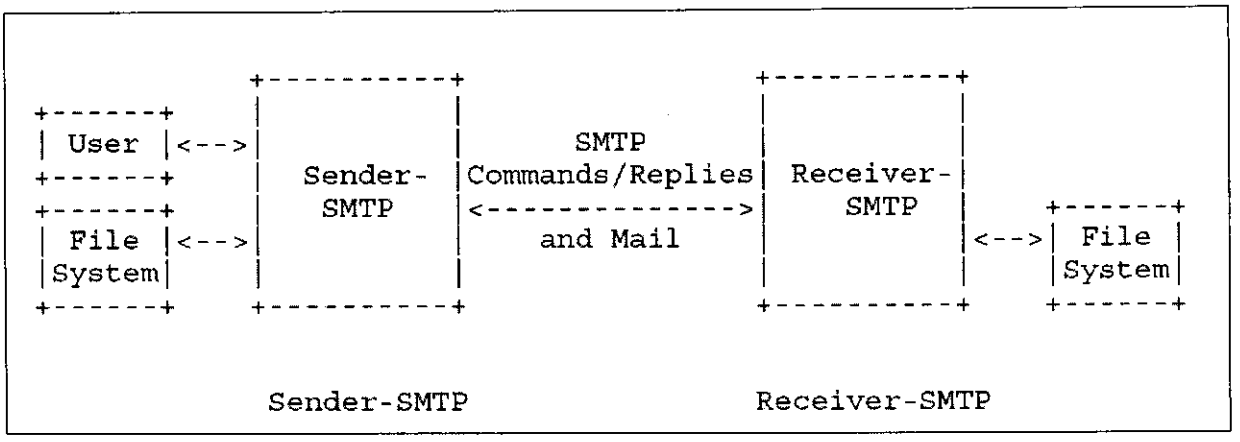


Fig. III.3 Scénario du dialogue SMTP.

L'argument associé à la commande MAIL est le chemin inverse, qui spécifie qui est l'émetteur du courrier. L'argument associé à la commande RCPT est un chemin direct, qui spécifie à qui est destiné le courrier. Le chemin direct sert à l'acheminement, tandis que le chemin inverse est utilisé pour renvoyer un message d'erreur à l'émetteur.
Les commandes et réponses du système de courrier ne tiennent pas compte de la casse et sont composées de caractères ASCII. Lorsque le service de transport utilisé permet la transmission sur une base de 8-bits (octet), tous les caractères 7-bits sont transmis justifiés à droite (dans l'octet de transmission), le huitième bit étant toujours forcé à 0.

Lorsque nous spécifierons la forme syntaxique générale des commandes et des réponses, un argument (ou un symbole "spécial") sera exprimé sous forme d'une variable (ou constante) (*métalinguistique*), par exemple :"<chaîne>" ou "<route-inverse>". Ici, les signes (*inférieur à*) et (*supérieur à*) indiquent le caractère métalinguistique de la variable. Cependant, certains arguments utilisent ces signes dans leur expression littérale. Par exemple, un chemin inverse est actuellement encapsulé par ces signes, c'est-à-dire, "<John.Smith@USC-ISI.ARPA>" est une instance de la variable générale <route-inverse> (les signes (*inférieur à*) et (*supérieur à*) sont effectivement transmis dans la commande ou la réponse). **[RFCf-13].**

4. Les procédures SMTP

4.1 Ouverture et fermeture de connexion

Dès que le canal de transmission est établi, un échange de protocole s'assure que l'hôte demandeur communique bien avec l'hôte attendu (SP est espace et CRLF est retour chariot) :

HELO <SP> <domain> <CRLF>
QUIT <CRLF>

Par la commande HELO, l'hôte émetteur s'identifie ; cette commande peut être assimilée à l'expression "Bonjour, je suis le domaine <domaine>". **[RFCa-14]**

```
R: 220 BBN-UNIX.ARPA Simple Mail Transfer Service Ready
S: HELO USC-ISIF.ARPA
R: 250 BBN-UNIX.ARPA
```

Fig. III.4 Ouverture de session SMTP.

```
S: QUIT
R: 221 BBN-UNIX.ARPA Service closing transmission channel
```

Fig. III.5 Fermeture de session SMTP.

4.2 Emission et postage

Le routage d'un message vers une boîte-aux-lettres d'un utilisateur est appelé (*postage*), l'envoi du même message sur le terminal de l'utilisateur est appelé (*transfert*). Du fait que, dans de nombreux hôtes, l'implémentation du transfert est quasiment identique à celle du postage, ces deux fonctions ont été combinées dans SMTP [RFC821].

Les utilisateurs devront avoir dans tous les cas la possibilité de contrôler l'inscription des messages entrants sur leurs écrans. La plupart des hôtes ont une fonction qui permet de désactiver l'écriture de ces messages.

Exemple1: `SEND <SP> FROM:<reverse-path> <CRLF>`

La commande SEND demande que le message soit affiché sur le terminal de l'utilisateur. Si l'utilisateur n'est pas connecté au système (ou n'accepte pas l'affichage interactif des messages sur son terminal), une réponse de code 450 sera renvoyé en réponse à la commande RCPT. La transaction est considérée comme aboutie lorsque le message est affiché sur le terminal.

Exemple2: `SOML <SP> FROM:<reverse-path> <CRLF>`

La commande « Send **Or** MaiL » demande que le message soit affiché sur le terminal utilisateur si ce dernier est connecté au système (et accepte l'affichage interactif de messages sur ce terminal). Dans le cas contraire, le message doit être redirigé vers sa boîte-aux-lettres. La transaction est considérée comme aboutie si le message est soit affiché sur le terminal, soit déposé dans la boîte-aux-lettres.

Exemple3: `SAML <SP> FROM:<reverse-path> <CRLF>`

La commande « Send **And** MaiL » demande que le message soit affiché sur le terminal utilisateur si ce dernier est connecté au système (et accepte l'affichage interactif de messages sur ce terminal). Dans tous les cas le message doit être copié dans sa boîte-aux-lettres. La transaction est considérée comme aboutie si le message a au moins pu être copié dans la boîte-aux-lettres. Les réponses à ces commandes sont les mêmes que pour la commande MAIL. **[RFCa-14]**

4.3 Emission du courrier

Une transaction SMTP se déroule en trois étapes.

1. La transaction est initiée par une commande MAIL donnant l'identification de l'émetteur.
2. Une série contenant une ou plusieurs commandes RCPT est appliquée, donnant les informations sur les destinataires.
3. Puis une commande DATA passe le contenu du message. Dans cette troisième phase, la marque de fin de données à transmettre marque la fin de la transaction, généralement un point «.» est utilisé pour marquer la fin de la commande DATA.

La première étape de la procédure est donc la commande MAIL. L'argument <route-inverse> contient le nom de la boîte aux lettres de l'émetteur.
Exemple4: `MAIL <SP> FROM:<reverse-path> <CRLF>`

Cette commande indique au récepteur SMTP qu'une nouvelle transaction de courrier débute et lui demande de réinitialiser ses tables d'états et ses tampons, y compris ses boîtes de réception et toute donnée de courrier latente.
Elle lui donne le chemin inverse qui pourra être utilisé en cas de rapport d'erreur à transmettre. Si l'émetteur accepte la commande, un code «250 OK» est renvoyé à l'émetteur.
L'argument <route-inverse> peut contenir plus d'une boîte aux lettres. On peut y inscrire une liste de plusieurs boîtes aux lettres dans des différents hôtes. La première boîte inscrite dans l'argument <route-inverse> doit néanmoins être celle désignant l'émetteur du message.

La deuxième étape de la procédure est l'émission des commandes RCPT.
Exemple5: `RCPT <SP> TO:<cheminDirect> <CRLF>`

Cette commande transmet une adresse de courrier désignant un destinataire. Si elle est acceptée, le récepteur renvoie une réponse de code « 250 OK », et mémorise le chemin d'acheminement. Si le destinataire est inconnu, le récepteur renvoie un code d'erreur « 550 Failure ». Cette seconde étape de la procédure peut être répétée autant de fois si nécessaire.

L'argument <route-directe> peut contenir plus d'une adresse de boîte aux lettres. On peut y inscrire une liste de boîtes aux lettres dans des hôtes différents. Le premier hôte à être mentionné dans l'argument <route-directe> sera l'hôte à qui est envoyé cette commande.

La troisième étape consiste en l'émission de la commande DATA.
Exemple6: `DATA < CRLF>`

Le récepteur SMTP renvoie une réponse de code "354 Intermediate" et prend en compte toutes les lignes de texte suivantes comme étant le texte du message. Lorsque la marque indiquant la fin du texte est reçue et enregistrée, le récepteur SMTP envoie une réponse de code « 250 OK ». Comme les données du courrier sont émises sur le même canal de transmission, il faudra donner un indicateur de fin de données pour que le dialogue requête-réponse puisse reprendre. SMTP indique la fin des données du message en envoyant une ligne contenant un point unique. Une procédure de "transparence" est utilisée pour éviter que celle-

ci n'interfère avec le texte de l'utilisateur. Notez que les données du message incluent un résumé d'en-tête comprenant les champs tels que Date, Subject, To, Cc, From.

L'indicateur de fin de message confirme en outre la transaction de courrier et signale au récepteur-SMTP qu'il peut désormais traiter les destinataires enregistrés pour ce message. Si les données sont acceptées, le récepteur-SMTP renvoie une réponse de code « 250 OK ». La commande DATA ne doit échouer que si la transaction de courrier s'avère incomplète (par exemple, pas de destinataires), ou si les ressources suffisantes pour traiter le courrier ne sont pas disponibles.

L'image suivante illustre l'utilisation de ces commandes dans une transaction SMTP. Elle donne un exemple de séquence SMTP montrant un courrier envoyé par Smith, utilisateur de l'hôte ARPA Alpha, à Jones, Green, et Brown utilisateurs de l'hôte ARPA Beta. Nous supposerons que l'hôte Alpha contacte l'hôte Beta directement.

Le courrier a été accepté pour Jones et Brown. Green n'avait pas de boîte aux lettres sur Beta. **[RFCa-14]**

```
S: MAIL FROM:<Smith@Alpha.ARPA>
R: 250 OK

S: RCPT TO:<Jones@Beta.ARPA>
R: 250 OK

S: RCPT TO:<Green@Beta.ARPA>
R: 550 No such user here

S: RCPT TO:<Brown@Beta.ARPA>
R: 250 OK

S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: Blah blah blah...
S: ...etc. etc. etc.
S: <CRLF>.<CRLF>
R: 250 OK
```

Fig. III.6 Séquence d'envoi d'un courrier.

4.4 Réémissions (*Forwarding*)

Il existe certains cas où l'information concernant un destinataire donnée dans la <route-directe> est incorrecte, mais le récepteur-SMTP connaît la destination exacte. Dans un tel cas, l'une des réponses suivantes pourra être émise pour permettre à l'émetteur de contacter la bonne destination **[RFCf-13].**

```
251 User not local; will forward to <forward-path>
```

Cette réponse indique que le récepteur-SMTP sait que la boîte-aux-lettres du destinataire est hébergée sur un autre hôte et indique le chemin d'accès correct de cette boîte aux lettres pour des messages futurs.

551 User not local; please try <forward-path>
Cette réponse indique que le récepteur-SMTP sait que la boîte-aux-lettres du destinataire est sur un autre hôte et signale le chemin d'accès correct à utiliser pour joindre cette boîte-aux-lettres.

```
S: RCPT TO:<Postel@USC-ISI.ARPA>
R: 251 User not local; will forward to <Postel@USC-ISIF.ARPA>

Or

S: RCPT TO:<Paul@USC-ISIB.ARPA>
R: 551 User not local; please try <Mockapetris@USC-ISIF.ARPA>
```

Fig. III.7 Utilisation des réponses SMTP.

4.5 Vérification de boites et expansion de liste de diffusion

SMTP propose à titre de fonctionnalités additionnelles, des commandes pour vérifier un nom de destinataire ou pour expanser une liste de diffusion. Ces deux opérations peuvent être menées respectivement avec les commandes **VRFY** et **EXPN**. **[SAM-15]**

Pour la commande **VRFY**, l'argument est un nom d'utilisateur, la réponse à cette commande pouvant inclure le nom complet de cet utilisateur et devant fournir une adresse complètement qualifiée de boîte-aux-lettres pour cet utilisateur.
Pour la commande **EXPN,** l'argument identifie une liste de diffusion, la réponse multi lignes à cette commande pouvant inclure le nom complet des utilisateurs et **devant** inclure l'ensemble des boîtes-aux-lettres contenues dans la liste.

```
    S: VRFY Smith
    R: 250 Fred Smith <Smith@USC-ISIF.ARPA>

Or

    S: VRFY Smith
    R: 251 User not local; will forward to <Smith@USC-ISIQ.ARPA>

Or

    S: VRFY Jones
    R: 550 String does not match anything.

Or

    S: VRFY Jones
    R: 551 User not local; please try <Jones@USC-ISIQ.ARPA>

Or

    S: VRFY Gourzenkyinplatz
    R: 553 User ambiguous.
```

Fig. III.8 Vérification de nom d'utilisateur.

```
    S: EXPN Example-People
    R: 250-Jon Postel <Postel@USC-ISIF.ARPA>
    R: 250-Fred Fonebone <Fonebone@USC-ISIQ.ARPA>
    R: 250-Sam Q. Smith <SQSmith@USC-ISIQ.ARPA>
    R: 250-Quincy Smith <@USC-ISIF.ARPA:Q-Smith@ISI-VAXA.ARPA>
    R: 250-<joe@foo-unix.ARPA>
    R: 250 <xyz@bar-unix.ARPA>

Or

    S: EXPN Executive-Washroom-List
    R: 550 Access Denied to You.
```

Fig. III.9 Expansion d'une liste d'utilisateurs.

4.6 Relais de messages

Le chemin d'acheminement des messages peut être une (*route*) de la forme :

"@UN,@DEUX:JOE@TROIS",

dans laquelle « UN », « DEUX », et « TROIS » sont des noms d'hôtes.

Cette forme est employée dans le but d'accentuer la différence formelle entre une adresse et une route. La boîte aux lettres est une adresse absolue, la route est une information permettant d'y accéder.

Dans le concept, les éléments de la route sont déplacés vers la route inverse au fur et à mesure que le message circule à travers les serveurs-SMTP intermédiaires. La route inverse est donc

construite comme la route prise en sens inverse, (c.-à-d., une route partant de la position courante et aboutissant à l'émetteur du message). Lorsque le message arrive sur un serveur-SMTP, et que le premier élément de la route (la première destination intermédiaire) ne correspond pas à la machine courante, cet élément ne doit pas être effacé de la route et est utilisé par le serveur-SMTP comme une nouvelle destination de réémission. Dans tous les cas, le serveur-SMTP courant ajoute son propre identifiant à la route inverse.

Par cette technique des routes, les récepteurs-SMTP reçoivent des messages qui doivent être relayés vers d'autres serveurs-SMTP. Le récepteur-SMTP peut accepter ou refuser de jouer ce rôle de relais de la même manière qu'il accepte ou refuse de traiter les messages de tel ou tel utilisateur local. Le récepteur-SMTP devient à ce moment un émetteur-SMTP, établit un canal de transmission vers le prochain agent SMTP mentionné dans la route, et lui envoie le message.
Si un serveur-SMTP a accepté de relayer un message, puis s'aperçoit que la route est incorrecte ou que le message ne peut pas poursuivre son chemin quelle que soit la raison, alors il devra composer un message (*undeliverable mail*) et le renvoyer à l'origine (dont il connaît la route par la route inverse).
Un message de notification d'un courrier (*undeliverable*) est montré dans la figure suivante. « Cette notification est la réponse à un message émis par JOE sur l'hôte HOSTW et envoyé via l'hôte HOSTX à l'hôte HOSTY avec des instructions pour un relais vers l'hôte HOSTZ. L'exemple montre la transaction entre l'hôte HOSTY et l'hôte HOSTX, qui constitue la première étape de la notification d'erreur ». **[SAM-15]**

```
S: MAIL FROM:<>
R: 250 ok
S: RCPT TO:<@HOSTX.ARPA:JOE@HOSTW.ARPA>
R: 250 ok
S: DATA
R: 354 send the mail data, end with .
S: Date: 23 Oct 81 11:22:33
S: From: SMTP@HOSTY.ARPA
S: To: JOE@HOSTW.ARPA
S: Subject: Mail System Problem
S:
S:    Sorry JOE, your message to SAM@HOSTZ.ARPA lost.
S:    HOSTZ.ARPA said this:
S:      "550 No Such User"
S: .
R: 250 ok
```

Fig. III.10 Message de notification "undeliverable mail".

4.7 Echange des rôles

La commande TURN peut être utilisée pour échanger les rôles des deux programmes en communication via le canal de transmission établi.
Si un programme-A est actuellement l'émetteur-SMTP, envoie la commande TURN et reçoit une réponse OK (250), alors le programme-A devient le récepteur-SMTP. Si un programme-B est actuellement en position de récepteur-SMTP, reçoit une commande TURN à laquelle il répond par une réponse OK (250), alors le programme-B devient l'émetteur-SMTP.

L'émission d'une réponse 502 indique que le récepteur refuse l'échange de rôles.
Notez toutefois que l'implémentation de cette commande est optionnelle. Elle ne devra d'ailleurs pas être utilisée en temps normal lorsque le canal de transmission est établi sous

TCP. Cependant, lorsque le "coût" d'établissement d'un canal de transmission est élevé, cette commande peut s'avérer utile. Par exemple, lorsque le canal de transmission s'appuie sur le réseau commuté public (téléphone), surtout lorsque certains hôtes font tourner une liaison vers un certain nombre d'autres hôtes pour mettre à jour les échanges de messages.

5. Domaines

Les domaines sont un concept récent dans le système de courrier de l'ARPA Internet. Et l'utilisation du système de domaines transforme l'espace d'adressage depuis un espace plat global de chaînes de caractères nommant des hôtes vers une structure hiérarchisée arborescente d'adresses globales. Le nom d'hôte est remplacé par un domaine et un identifiant d'hôte, couple représenté par une séquence d'éléments de domaines sous forme de chaînes de caractères séparés par des points, et dans un ordre conventionnellement établi du plus spécifique au plus général.
Par exemple, "USC-ISIF.ARPA", "vf.eisti.FR", et "PC7.LCS.MIT.ARPA" sont des domaines à part entière. Lorsque des noms de domaines figurent dans des messages SMTP seuls les noms officiels sont autorisés, et l'usage des pseudos ou alias n'est pas permis. **[SAM-15]**

6. Aspects de sécurité

À la base, une des limitations de SMTP vient de l'impossibilité d'authentifier l'expéditeur. Pour ceci, l'extension SMTP-AUTH a été définie. Malheureusement, l'impossibilité d'imposer largement SMTP-AUTH a rendu ce protocole impuissant face au phénomène du spam.

Le Spam est dû à un certain nombre de facteurs dont l'implémentation de MTAs ne respectant pas les standards, et les failles de sécurité dans les systèmes d'exploitations autorisant les spammeurs à contrôler à distance des PC utilisateur pour leur faire envoyer du spam.

Afin de lutter efficacement contre ce phénomène, il existe deux approches : modifier profondément SMTP ou bien lui adjoindre d'autres protocoles pour combler ses lacunes. Modifier SMTP de manière importante ne paraît pas faisable vu l'importance du réseau de serveurs déjà installé.

Une autre approche consiste à créer des systèmes visant à assister les opérations du protocole SMTP. Le ASRG (groupe de recherche anti-spam) de l'IRTF (*Internet Research Task Force*), travaille actuellement sur l'authentification des courriers électroniques dans le but de fournir un système flexible, et plus léger. L'ensemble de ces recherches ont abouti au protocole MARID en 2004 ainsi qu'au protocole Domain Keys Identified Mail en 2006. **[WPP-12].**

7. Conclusion

L'utilisation intensive du courrier électronique a amené avec lui ses lots de problèmes. Le spam en est un important. Pour maintenir ce service de manière efficace, on doit traiter ses Spams par des procédures et des mécanismes ad hoc.

CHAPITRE IV.

LE SPAM « pourriel ».

The sublime and the ridiculous
are often so nearly related that
it is difficult to class them separately.

– Tom Paine.

1. Introduction

Les spammeurs disent que le Spam est juste une forme de liberté d'expression, tandis que les anti-spammeurs disent qu'il prive les gens de cette liberté de la parole, car il maintient les utilisateurs loin des newsgroups et d'autres forums Internet de discussion.

La raison pour laquelle le Spam existe est parce qu'il fonctionne. Les statistiques reportent qu'entre 10 à 20% des messages reçus par les utilisateurs par des inconnus ont obtenus des réponses; ce qui explique bien l'évolution de cette pratique (le spamming) **[GR-17]**.

Ainsi il faut bien noter que la raison pour laquelle le Spam est si répandu est que cette technique ne coûte presque rien ! L'achat de fichiers d'adresses ou de logiciels collecteurs d'e-mails se fait à un coût dérisoire (quelques dizaines d'euros). L'envoi des courriers électroniques ne coûte pas grand chose non plus (il suffit d'avoir un ordinateur et une connexion Internet).

2. Terminologie et Définitions

2.1 Définition du spam

Le « Spam » est un terme adopté par la communauté d'Internet d'après un croquis célèbre (*python de Monty*) pour marquer des envois massif de messages non sollicités sur des groupes de discussion Usenet.

La définition technique standard proposé par « *mail-abuse.org* » et le groupe anti-Spamming au projet de Spamhaus **[DSP-16]** est qu'un message électronique est dit « spam » si :

1. l'identité et le contexte personnel des destinataires sont non pertinents parce que le message est également applicable à beaucoup d'autres destinataires potentiels.
2. le destinataire n'a pas accordé une permission vérifiable, explicite et révocable pour permettre au message d'être envoyé.
3. la transmission et la réception du message semble au destinataire donner un avantage disproportionné à l'expéditeur.

Le mot « Spam » apparaît maintenant dans les dictionnaires et il est décrit comme :

a) Un E-mail non sollicité, souvent à caractère commercial, envoyé aléatoirement aux multiples listes de mailing, aux individus, ou aux newsgroups.
b) Une autre définition employée par les compagnies qui bloquent les sources de Spam est : « Le Spam d'Internet est un ou plusieurs messages non sollicités, envoyés ou signalés en tant qu'élément d'une plus grande collection de messages, et qui ont tous un contenu identique ».

Il est considérer que, à l'heure actuelle, plus de 90% du trafic de courrier électronique est du *spam*. Voici un report du l'organisation spamhaus sur les spam :

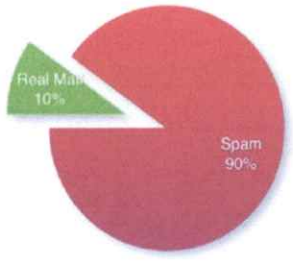


Fig. IV.1 Pourcentage de spam.

2.2 Caractéristiques de Spam

Il y a une frontière très fine entre le marketing légitime d'E-mail et le Spam. Généralement, le Spam a un ou toutes les caractéristiques suivantes :

- L'adresse de retour est invalide : Si vous essayez de répondre à l'E-mail de Spam, vous obtiendrez une erreur de la livraison.
- En-têtes forgés : Cacher l'origine de l'E-mail, conduisant des en-têtes sont habituellement forgés, qui le rend très difficile à tracer des Expéditeurs persistants.
- L'identité du destinataire est non pertinente : parce que l'E-mail est valide pour beaucoup d'autres destinataires.
- Adresse d'attaque de dictionnaire : Si le « To »ligne d'adresse est examiné, vous pouvez voir différentes variantes des adresses de l'E-mail du destinataire. Par exemple, si votre vraie adresse E-mail est JohnSmith@hotmail.com, vous verriez JohnSmith@yahoo.com, JohnSmith@aol.com, JohnnySmith@hotmail.com etc.
- Le sujet n'a pas de relation avec le contenu de l'E-mail.
- L'utilisation des programmes qui génèrent automatiquement et aléatoirement des caractères dans la ligne de « sujet » afin de dévier des filtres de Spam.
- Le contenu d'E-mail est à caractère douteux. Par exemple, certaines des matières couvriraient des arrangements, des perfectionnements de corps, devenir riches ... etc. D'autres sont de fraudes bien connues telles que de diverses formes de fraudes Nigériens d'E-mail.

- La désinscription n'est pas fonctionnelle dans le Spam : Si vous essayez de vous désinscrire du Spam, c'est souvent le cas que le lien ne fonctionne pas, ou ouvrent le site Web d'une publicité.
- Si le Spam contient le HTML, il peut contenir le Java script caché, qui peut ouvrir des sites Web et activer des fenêtres de POPUP de publicité

2.3 Identification du SPAM

Les résultats de l'opération "Boîte à Spam", menée entre juillet et octobre 2002 par la CNIL (*commission nationale de l'information et des libertés*), sont révélateurs. Sur 325 000 Spam signalés par les internautes, 85 % étaient en langue anglaise et 55 % des messages en français étaient à caractère pornographique. Mais ce n'est pas toujours le cas.

On trouve ainsi des « faux positifs » qui sont des messages légitimes classifiés comme des Spam et des « faux négatifs » qui sont des Spam qui franchissent le système de filtrage. Les filtres de Spam doivent permettre plus de faux négatifs que de faux positifs **[SMD-18]**.

Les soucis concernant les faux positifs sont l'une des raisons pour lesquelles l'utilisateur voit tellement le Spam. La crainte de perdre des informations relativement importante oblige les responsables à décider si un E-mail est un spam ou non.

Les fournisseurs de filtre de Spam mentionnent que seulement 0.001% de messages filtrés sont des faux positifs. Cependant, pour une grande compagnie qui reçoit 50.000 E-mails quotidiennement, le petit pourcentage peut faire une différence au résultat de la compagnie. Un de ces E-mails faux positifs a pu être d'importance essentielle pour la compagnie. Un autre scénario intéressant est quand il s'agit de compagnies médicales et pharmaceutiques qui reçoivent un grand nombre d'E-mails contenant des mots qui sont signalés comme des mots suspects, tel que le sexe, Viagra, la drogue etc. Il est essentiel que ces E-mails ne soient pas considérés comme des spam par le procédé de filtration.

Une méthode commune en faisant face aux faux positifs est d'expédier tous les E-mails considéré comme Spam à un dossier spécial par exemple. Les utilisateurs peuvent alors examiner ces E-mails mis en quarantaine, qui seront automatiquement supprimés après une date d'échéance. Une grande compagnie peut devoir stocker beaucoup de giga-octets de ce type d'E-mail, avec le bas coût de dispositifs de stockage, et le coût élevé de perdre un grand contrat dû à un faux positif, le choix de garder les messages filtrés est évident.

3. Modèles de SPAM

3.1 Le SPAM anglophone

Le Spam anglophone vous promet monts et merveille : vous faire gagner de l'argent ou vous procurer du Viagra **[SMD-18].** Voici à quoi cela ressemble :

Objet: Make Easy Money from Home!
Brand New - Ultimate Business Opportunity!
Just Released World Wide!
Want an Income Independent of your job?
Want To Be Your Own Boss and Own Your Own Business?
Want To Easily make $200 - $1,000 per day... paid daily?
Want To Earn Long Term Residual Income for Life?
Want Unlimited Leads to Start Earning Big Immediately!

Fig. IV.2 Spam anglophone.

3.2 Le SPAM amateur

Le responsable est généralement le web master d'un petit site amateur. Le courrier vous donne accès à une partie de la liste des spammés. **[SMD-18]**

Exemple:

From : "Spammer" <spammer@bidule.fr>
To : gp@mygale.org, grn@mygale.org, grnr@mygale.org, gnd@mygale.org, ghnr@mygale.org, gr@mygale.org, glsj@mygale.org, gnsd@mygale.org, gng@mygale.org, gr@mygale.org, gstn@mygale.org, gbrd@mygale.org, grm@mygale.org, gs@mygale.org, (...), vcs@mygale.org, vf@mygale.org, vhnd@mygale.org, vll@mygale.org, vd@mygale.org, vncnt@mygale.org, vm@mygale.org, xcss@mygale.org, xl@mygale.org, xlm@mygale.org, xl@mygale.org, m@mygale.org, mrc@mygale.org, zg@mygale.org, zrt@mygale.org, zrt92@mygale.org, zd@mygale.org, zn@mygale.org, zz@mygale.org, zrl@mygale.org

Subject : Apprendre l'anglais sur INTERNET
Nous vous invitons à essayer nos méthodes d'anglais sur Internet, à l'adresse suivante :
http://www.passerelle.com/anglais

Merci

Fig. IV.3 Spam amateur.

3.3 SPAM « erreur de destination »

La dernière tendance est d'envoyer des Spams en faisant croire qu'il s'agit de courriers qui vous été adressés par erreur. **[SMD-18].**

Objet : Tu vois que je t'oublie pas :!
Coucou c'est moi ! tu te souviens on a dial. y a deux jours sur le chat ! je suis pas sure de l'adresse mais je pense que ça doit être la bonne, donc tu me demandais ou trouver des photos gratos et des webcameras et bien je te file l'adresse c'est >> http://wwwxxxxxxx.com
comme je te l'avais dit donc il y a des photos gratos c'est ça que tu cherchais, des vidéos et aussi le dialogue ou tu fais des rencontres rapide et discrètes, moi j'y suis parfois le soir mais bon, si tu viens préviens moi et je me débrouillerais pour venir, sinon au pire tu peux m'écrire . bon alors a bientôt peut être sur http://wwwxxxxxxx.com
je te fais un gros bisous
Sabrina
nb/ si c'est pas toi, fait pas gaffe a mon message et excuse moi. Bisey !

Fig. IV.4 Spam « erreur de destination ».

3.4. Le Spam vicieux

Il vous propose dès les premières lignes un lien. Bien évidemment, vous pensez que c'est pour ne plus recevoir de messages. Bien au contraire ! C'est pour vous inscrire ! Attention au piège... **[SMD-18].**

> Just one more step! Simply click the link below to activate the subscription request you just sent us !
> http://c.postmasterdirect.com/confirm? E=buzz%40arobase.org&T=4942

Fig. IV.5 spam vicieux.

3.5. Le spammeur : plus hypocrite, tu meurs...

Le spammeur annonce que son courrier n'est pas du spam. **[SMD-18].**

> Ce message n'est pas du SPAM mais simplement une prospection pour vous faire connaitre notre site web http://www.xxxxxxx.com. Vous ne recevrez plus ce message à moins que vous n'en fassiez la demande expresse en cliquant sur le lien suivant pour vous inscrire et continuer à recevoir notre lettre.
>
> Nous avons recopié vos coordonnées "email" sur votre site ou sur un site partenaire. Ceci n'est pas du spam, et vous pouvez enlever vos coordonnées pour ne plus recevoir nos messages en cliquant ici.
>
> si ce message vous a été adresse par erreur, merci de nous le signaler

Fig. IV.6 Spam « ceci n'est pas un spam1 ».

Encore plus fort, une référence à un pseudo décret américain :

> Cet e-mail non sollicité n'est pas un S.P.A.M. parce qu'il est conforme au décret 1618 Titre III du 105e Congrès des États-Unis qui stipule que le e-mail doit contenir l'adresse d'expéditeur et avoir une facilité d'annulation pour l'adressé.

Fig. IV.7 Spam « ceci n'est pas un spam2 ».

4. Les Cibles du SPAM

Le spam peut s'attaquer à divers médias électroniques : les mails, les forums de discussion, les moteurs de recherche, les wikis, les messageries instantanées, les blogs...etc. **[WST-19].**

4.1. Par courrier électronique

Le spam par courrier électronique est le type de pollupostage le plus répandu. Le coût d'envoi d'un courrier électronique étant négligeable, il est facile d'envoyer un message à des millions de destinataires. Les destinataires assument le coût de réception et de stockage en boîte aux lettres, ce qui peut causer des coûts non négligeables aux prestataires de services, à cause du volume pris par le spam.

Contrairement aux promotions commerciales pour lesquelles les utilisateurs peuvent donner leur accord, le pourriel n'est pas sollicité. Il est souvent rédigé spécialement pour contourner les filtres anti-spam. Un mot clé tel que **Viagra** (souvent vanté dans les pourriels) peut être ainsi écrit « **vi@gr@** » ou « **v|agra** » ou « v i a g r a » de manière à tromper un filtrage automatique basé sur ce mot.

Une autre méthode employée consiste à accompagner un texte anodin d'une image sur laquelle se trouve le véritable message publicitaire, l'absence de mot compromettant en dehors de l'image rendant le filtrage de ces messages très compliqué.

Les polluposteurs redoublent d'imagination pour masquer leurs activités et ne pas être démasqués, que ce soit en falsifiant les adresses d'expéditeur ou en utilisant des serveurs SMTP non sécurisés qui permettent des envois anonymes.

Les adresses à pollupostages sont généralement collectées par robot d'indexation. Il existe un marché pour les listes d'adresses. **[WST-19].**

Pour éviter d'être polluposter, les internautes font souvent figurer leurs adresses d'une manière masquée lorsqu'elle doit apparaître dans un site Web. Par exemple :

- Jean@NOSPAM.exemple.fr pour Jean@exemple.fr.
- Jean chez exemple point fr pour Jean@exemple.fr
- Jean [at] exemple.fr pour Jean@exemple.fr (l'arobase se prononçant souvent "at").

Fig. IV.8 comment évité le pollupostage.

4.2. Par message de forum de discussion

Ce type de pollupostage est apparu sur Usenet avant celui par courrier électronique. Les forums de discussion d'Usenet sont une cible facile de pollupostage. En effet, un message envoyé à un forum touche tous les lecteurs du forum. Certains groupes de discussion ne reçoivent pratiquement plus que du pollupostage (c'est l'une des raisons pour lesquelles de nombreux forums sont modérés, c'est-à-dire surveillés par un humain ou un robot qui effectue un tri parmi les articles proposés).

D'autre part, les participants d'Usenet faisant généralement figurer leur adresse électronique dans leurs articles, les polluposteurs peuvent facilement récolter des milliers d'adresses au moyen d'un robot, puis polluposter les auteurs de ces articles par courrier électronique.

Le phénomène est rendu encore plus pénible et par la publication croisée ou la publication multiple, qui consistent respectivement à destiner un message à plusieurs groupes simultanément ou à envoyer le message dans plusieurs groupes de suite.

Dans Usenet, on parle également de pourriel lorsqu'un article, quel que soit son contenu, et même s'il n'appartient pas aux catégories usuelles de messages abusifs (publicités commerciales, escroqueries, insultes…) est publié en un nombre d'exemplaires excessif : tous les exemplaires d'un tel article peuvent être annulés par les utilisateurs (les critères numériques exacts permettant d'identifier de tels pollupostages dans la principale hiérarchie francophone sont donnés dans la documentation du forum). Les diverses hiérarchies possédant des critères différents pour identifier et annuler les articles de pollupostage, il existe une certaine incertitude quant aux traitements qui peuvent être appliqués aux pollupostages diffusés simultanément dans plusieurs de ces hiérarchies.

Une règle particulièrement simple décide du sort des escroqueries manifestes, c'est-à-dire des articles proposant de (gagner de l'argent rapidement et sans rien faire) habituellement appelés «MMF »(*Make money fast)* ces articles peuvent être annulés immédiatement par n'importe quel utilisateur.

A noter que depuis peu, le terme général de spamposting est parfois utilisé pour désigner le pollupostage apparaissant dans les blogs, forums, livre d'or...etc. **[WST-19].**

4.3. Par des fenêtres intruses de Windows

C'est ce que l'on appelle le pollupostage par messagerie instantanée.

4.4. Par systèmes vocaux

Le développement de la voix sur réseau IP (*téléphonie par Internet*) fait craindre la prochaine arrivée sur nos combinés d'un nouveau type de pourriel, le pollupostage vocal, baptisé SpIT (*Spam over Internet Telephony*). En effet, des systèmes comme « Skype » voient déjà une part notable des appels d'origine non sollicitée, même s'il s'agit encore de contacts personnels plutôt que de sollicitations commerciales. **[WST-19].**

4.5. Référencement abusif (Spamdexing)

Optimisation abusive des techniques de référencement destinée aux robots d'indexation de moteur de recherche qui consiste à modifier des pages Web en utilisant des mots-clés d'une façon abusive pour améliorer le classement dans les moteurs de recherche.

Parmi les techniques utilisées :

- La manipulation des mots clés consistant à ajouter une longue liste de mots souvent recherchés (comme « sexe » ou « piratage ») répétitivement dans une page (« sexe à UneVille », « sexe à UneAutreVille », « sexe à EncoreUneAutreVille », « sexe à UnVillageTropPetitPourÊtreUneVille », et toute autre variation possible) pour apparaître immédiatement si on fait une recherche avec ces mots. Parfois une page ne contient que les résultats d'une recherche, mis sur le Web pour être trouvé et classé par les moteurs de recherche et affiché aux usagers cherchant avec les mêmes mots.

- Le bourrage de mots clés populaires, dissimulés au visiteur mais pas au robot, soit en les imprimant blanc sur fond blanc, ou en utilisant la police de caractères la plus petite, ou encore en les utilisant en lignes de commentaires qui ne sont pas affichées à l'usager, ou encore en changeant le contenu de la page après que le logiciel du robot l'ait lu ou en modifiant le serveur pour envoyer une page au moteur de recherche et une autre aux usagers ordinaires.

- Le pollupostage de liens consiste à mettre les liens vers un site qu'on veut promouvoir dans autant d'autres sites externes que possible, incluant les forums publics et les pages de commentaires d'autres sites.

- Une ferme de liens (*link farm*) : est un site hébergeant des listes de liens vers tous les autres sites qu'on contrôle pour améliorer le classement de ces derniers en les faisant

apparaître populaires. Google compte notamment la quantité et l'importance des liens vers un site pour déterminer l'importance du site (*le PageRank*). Parfois on construit aussi les sites multiples (*simulant des sites indépendants et pas simplement des sous domaines du même site*) avec presque le même contenu ; chacun contient un tas de liens vers tous les autres pour améliorer leur classement.

- La technique de Bombardement Google en sa forme originale consiste à placer des hyperliens vers George W. Bush avec des phrases comme « l'idiot du monde » dans autant de sites Web que possible. La destination de cet hyperlien est normalement un site externe (dans cet exemple, la page de Bush). Si ce genre de lien figure dans un nombre assez grand de pages Web, une recherche pour « l'idiot du monde » va diriger l'usager immédiatement sur Bush, peu importe s'il désire ça et peu importe si ces mots figurent sur son site ou pas.

- Une autre variation est le pourriel d'affiliation (*affiliate Spam*) où une compagnie paye pour chaque visiteur ou chaque client envoyé par des liens affichés par des autres, du genre « affiliez-vous et devenez riche, mettez un lien vers www.arnaqueur.porno.example.com et pour chaque victime qui nous donne tous ses numéros de carte de crédit nous vous donnons un sou ».
- Les liens venant de ces programmes d'affiliation contiennent le code d'identification de façon :

www.arnaquer.porno.example.com/donnemoiargent?MonsieurLeSpammeur Pour laisser savoir qui doit être payé pour avoir posté tous ces liens partout

Fig. IV.9 Référencement abusif spamdexing.

Les opérateurs de sites de recherche comme Google cherchent toujours des moyens de détecter ce genre de choses et les rendre plus difficiles à utiliser effectivement ; si on a un tas de liens venant des sites du genre «www.ferme-aux-liens-inutiles.spam.example.com». **[WST-19].**

4.6 Dans les blogs (splog)

La présence de liens vers un site web est un critère important de classement dans les moteurs de recherche. Afin d'augmenter artificiellement le nombre de liens pointant vers leurs sites, certains créent des blogs, ou mettent des messages de commentaires dans des blogs préexistants, uniquement pour ajouter des liens vers un ou plusieurs sites web à promouvoir.

L'automatisation de ce genre de pollution a mené plusieurs logiciels de blog à introduire des contrôles qui rendent cette automatisation par une machine plus complexe à réaliser.

[WST-19].

4.7. Sur les wikis

De nombreux polluposteurs ou publicitaires ajoutent des liens vers leurs sites sur des wikis comme « Wikipédia », sites dont n'importe qui peut modifier les pages. **[WST-19].**

5. Spam en chiffre [SC-20]

Voici des statistiques reportant les différentes manifestations des spam dans le mode entier :

Table IV.1 Origines du Spam
(Entre avril et juin 2006)
Source : Sophos

Rang	Pays	Part des spams
1	Etats-Unis	23,2 %
2	Chine (et Hong Kong)	20,0 %
3	Corée du Sud	7,5 %
4	France	5,2 %
5	Espagne	4,8 %
6	Pologne	3,6 %
7	Brésil	3,1 %
8	Italie	3,0 %
9	Allemagne	2,5 %
10	Royaume-Uni	1,8 %
11	Taiwan	1,7 %
12	Japon	1,6 %
13	Autres pays	22,0 %

Table IV.2 Répartition du spam par catégories en 2007
Source : Clearswift

Types de spam	Avril 2007
Santé	43,86 %
Finance	37,65 %
Produits	9,06 %
Pornograhie	5,32 %
Scam (phishing)	1,41 %
Pari	0,1 %

Table IV.3 Etats-Unis : nombre de spams quotidiens selon une étude menée en novembre 2006

(En pourcentage du nombre des personnes sondées)
Source : Robert H. Smith School of Business, 2006

Nombre de messages	Pourcentage
<1	22 %
1-4	18 %
5-9	16 %
10-19	13 %
20-39	12 %
40 et plus	11 %
Ne sait pas	8 %
Moyenne	18,5 spams
Autres	2,32 %

6. Profil de Spammeur

Les noms et les adresses des spammeurs les plus prolifiques sont facilement disponibles sur Internet. Quelques sites Web anti-Spam entrent dans le grand détail du mode de fonctionnement décrivant les spammeurs, les endroits tel que noms d'emprunt qu'ils emploient, de boîte de P.O, des affaires et des adresses à la maison, des banques qu'ils emploient, des noms d'ISP et des adresses d'IP, des noms de domaine et des adresses E-mail, et les sites Web qui sont employés pour annoncer des produits et des services de Spam. Des spammeurs peuvent être classés par catégorie par les descriptions suivantes :

- spammeurs occasionnels : Ce sont les utilisateurs d'Internet qui font suivre des E-MAIL, cela est généralement considéré comme juste un ennui. C'est intéressant à noter que quelques fournisseurs, y compris Yahoo, considèrent ce type d'E-mail comme une violation de leur accord d'utilisation et pourraient avoir comme conséquence l'arrêt de compte.
- Spammeurs à échelle réduite. La grande majorité de spammeurs entrent dans cette catégorie et ils s'appellent parfois les « Spam Kiddies », généralement ils ne comprennent pas comment les relais sont ouverts et comment les logiciels de filtrages fonctionnent. Ils tendent à employer des outils de Spamming et des listes de courrier, achetées sur des CD ou téléchargées du Web.
- Spammeurs pirates : C'est les spammeurs les plus sophistiqués qui créent des logiciels pour éviter des filtres de Spam.
- Les spammeurs à grande échelle : C'est la catégorie de spammeur qui fait la plupart de dommages et crée la majeure partie du Spam sur l'Internet. Ils utilisent leurs stratégies pour des affaires rentables.

7. Conclusion

Avec l'évolution exceptionnelle du spamming, les organisations « anti-abuse » ont mobilisé une quantité importante de ressources dans le but de minimiser le taux d'attaque. L'utilisation de différentes solutions de détection tel que les anti-spam et les pièges à pirates (HoneyPot) a permis d'identifier 70% des spam envoyés.

CHAPITRE V.

Les Outils Utilisés.

Il faut de tout pour faire un monde.

– proverbe romain.

1. introduction

Pour mieux profiter des avancées technologiques qui ont facilité l'utilisation des outils informatiques même pour ceux qui ne sont pas des spécialistes dans ce domaine, on a adopté quelques outils tant tôt software que hardware et qui coopèrent entre eux pour maximiser le bénéfice de notre solution et pour palier aux différentes lacunes de la solution proposée toute seule. Ainsi dans le reste de ce chapitre vous trouviez des présentations et des descriptions de différents outils qu'on a utilisés.

Certains de ces outils sont utilisés dans l'architecture de base de notre solution tandis que d'autres sont utilisés seulement dans l'environnement de test.

2. SendMail

2.1 Introduction

Sendmail est un serveur de messagerie électronique dont le code source est ouvert. Il se charge de la livraison de messages électroniques. Descendant du logiciel ARPANET DELIVERMAIL, sendmail est un programme très flexible supportant un large éventail de moyens de transfert et de livraison d'emails, incluant le protocole SMTP.

2.2 Historique

Il est banal, aujourd'hui, de dire que l'Internet connaît un succès foudroyant. Ce succès entraîne dans son sillage la messagerie basée sur le protocole SMTP. L'implémentation la plus répandue de ce protocole est sans conteste le programme sendmail conçu par « Eric Allman en 1980 ». Après plusieurs années sans maintenance claire, ayant vu apparaître plusieurs versions dérivées et divergentes, Eric en a repris depuis 1990 la maintenance et l'extension pour prendre en compte les nouveaux besoins exprimés et standardisés, comme MIME et DSN par exemple. **[SMH-21].**

2.3 Avantages et inconvénients

Sendmail est très critiqué pour sa lenteur, sa complexité et sa maintenance difficile en comparaison avec d'autres MTA tels que Qmail et Postfix. Toutefois, il reste le MTA le plus populaire sur Internet, ce qui est certainement dû à son implémentation par défaut dans les différentes variantes d'Unix. Une étude de novembre 2001 estime que 42 % des serveurs mails accessibles publiquement utilisent sendmail.
Il a été implanté dans Mac OS X avec les versions 10.0 à 10.3. **[WSM-22].**

2.4 Configuration

La configuration de SendMail a toujours été une étape difficile dans la vie d'un administrateur de messagerie. Les difficultés ont sans doute des origines multiples, mais ont généralement en commun la complexité et la diversité des situations que doit gérer SendMail.

Une nouvelle évolution s'est dessinée ces dernières années : il n'y a plus guère de configurations de SendMail écrites à la main. La plupart des configurations sont obtenues à partir de définitions de plus ou moins haut niveau, et le fichier `sendmail.cf` est généré automatiquement. **[SAM-15].**

2.5 Le rôle de Send Mail

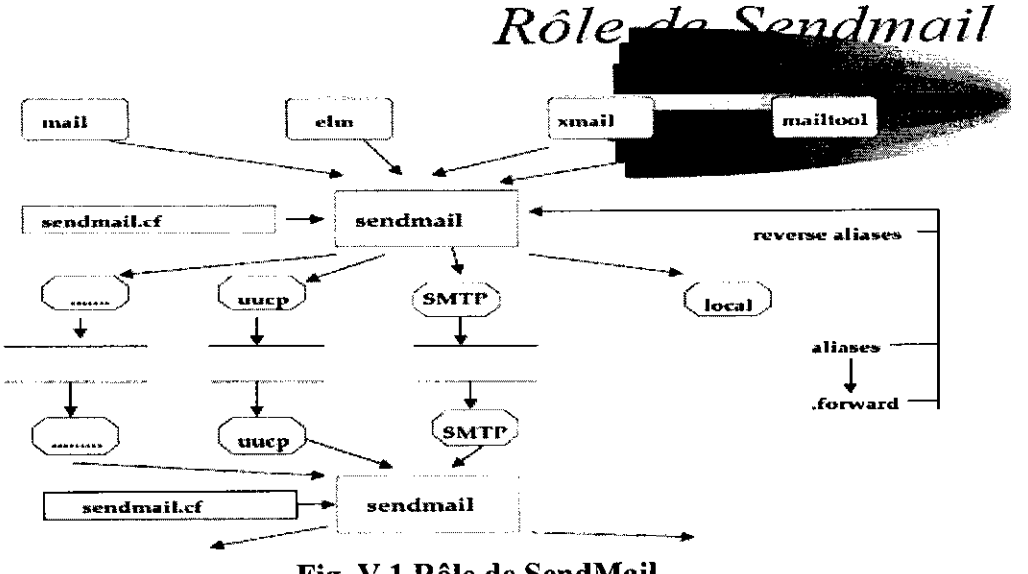


Fig. V.1 Rôle de SendMail.

3. Spam assassin

3.1 Introduction

Spamassassin est un outil anti-spam pouvant être utilisé en collaboration avec SendMail. Il attribue des "scores" à chaque message électronique reçu selon différents tests destinés à déterminer s'il s'agit ou non d'un spam. Pour cela, il effectue de nombreuses vérifications comme :

- la validité des adresses de l'expéditeur et du destinataire,
- la date des messages,
- la présence dans le corps du message d'un mot répertorié sur une liste de mots interdits,
- l'appartenance ou non d'un des serveurs expéditeurs à une liste noire,
- etc....

Chaque test positif augmente le score total d'un message. Quand le score final obtenu dépasse le seuil défini dans la configuration, le message est considéré comme étant un spam.

La base de données de SpamAssassin peut être mise à jour régulièrement grâce à la possibilité d'intégrer les outils Razor, Pyzor et DCC. SpamAssassin inclut également un algorithme de Bayes qui lui permet de reconnaître les nouveaux messages non sollicités à partir d'anciens spam. Enfin, Spamassassin peut être soit utilisé à la demande (appel de l'application chaque fois qu'un message est reçu), soit exécuté en tant que démon. Nous utiliserons, dans le cas de l'interaction avec SendMail, la configuration en mode démon. **[MSI-24].**

3.2 Architecture logique de SpamAssassin

L'architecture logique pour la transmission de serveur de messages se compose :

- d'un premier serveur de mail associé à un anti-spam situé sur la DMZ,
- d'un second serveur de mail situé sur le réseau local.

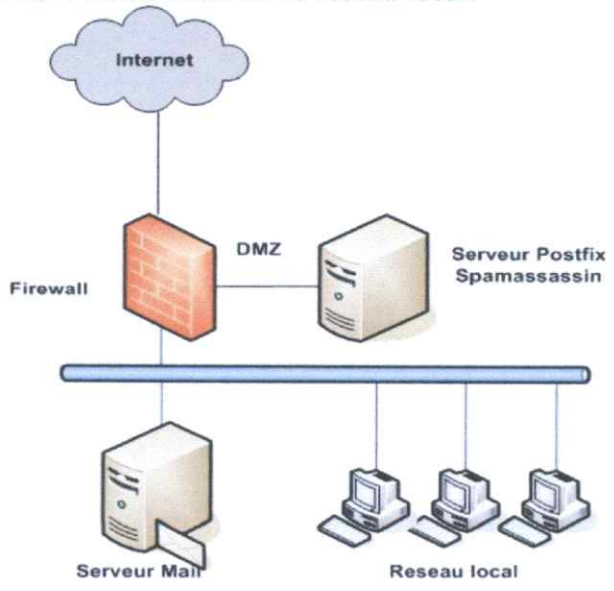


Fig. V.2 Architecture logique de Spamassassin.

3.3 Installation

Vérifiez que vous êtes logués en tant que administrateur (*root*) sur le serveur pour effectuer l'installation de SpamAssassin. Vous devez exécuter les commandes suivantes dans le répertoire /usr/ports/mail/p5-Mail-Spamassassin :

- Sinon, faire séparément :
  - #make
  - #make install
  - #make clean

L'installation a ajouté l'application (*spamc)* et le démon (*spamd)* dans le répertoire /usr/local/bin. Vous devez créer un utilisateur et un groupe « spamassassin » pour permettre l'interaction avec SendMail. **[SAB-23].**

Pour lancer le démon *spamd* au démarrage, vous devez ajouter la ligne suivante dans le fichier /etc/rc.conf : spamd_enable=« YES »

Pour lancer manuellement le démon en mode « debug », tapez la commande suivante :

- #/usr/local/bin/spamd –D –c –d –r /var/run/spamd.pid

3.4 Configuration

Le fichier de configuration local.cf de SpamAssassin se trouvent dans le répertoire /usr/local/etc/mail/spamassassin. Et vous pouvez l'utiliser pour régler et même définir différentes configurations sur Spamassassin.

3.4.1 Configuration de base

Pour faire une configuration basique, vous devez, avant tout, connaître le contexte dans lequel SpamAssassin doit fonctionner. La principale question est de savoir si la tolérance aux spam est faible, moyenne ou élevée. Nous vous conseillons de paramétrer une tolérance moyenne (required_hits 7.0) dans un premier temps et de modifier ce choix si nécessaire.

Voici, ci-dessous, les paramètres que vous devez appliquer dans le fichier local.cf :

- # combien de coups avant de considérer le message comme
- spam required_hits 7.0
- # encapsuler le spam dans un fichier attaché (O non,1 oui)
- report_safe 0
- # langues acceptées (fr : France)
- Ok_languages fr

Pour faire une configuration évolutive, il vous suffit d'intégrer les outils Razor, Pyzor, DCC et l'algorithme de Bayes. **[SAB-23].**

3.4.2 Configuration de Razor (intégré avec SpamAssassin)

Vérifiez que vous êtes logués en tant que root sur le serveur pour effectuer la configuration de Razor. Exécutez les commandes ci-après pour mettre à jour la configuration de Razor :

- #razor-client
- #razor-admin -home=/usr/local/etc/mail/spamassassin/.razor -create
- #razor-admin -home=/usr/local/etc/mail/spamassassin/.razor -discover
- #razor-admin -home=/usr/local/etc/mail/spamassassin/.razor -register

Vérifiez dans le fichier de log /usr/local/etc/mail/spamassassin/.razor/razor-agent.log que l'exécution de chacune des commandes s'est correctement réalisée.

Dans le fichier de configuration de SpamAssassin local.cf, ajoutez les lignes suivantes :

- # utiliser Razor ? oui=1 non=0
- use_razor2 1
- # chemin d'acces au fichier de configuration de razor
- razor_config /usr/local/etc/mail/spamassassin/.razor/razor-agent.conf

3.4.3 Configuration de Pyzor

Vérifiez que vous êtes logués en tant que root sur le serveur pour effectuer l'installation de Pyzor. Vous devez exécuter les commandes suivantes dans le répertoire /usr/ports/mail/pyzor

- Sinon, faire séparément :

  - #make
  - #make install
  - #make clean

Dans le fichier de configuration de SpamAssassin local.cf, ajoutez les lignes suivantes :

- # utiliser Pyzor ? oui=1 non=0
- use_pyzor 1
- # chemin d'acces au client pyzor
- pyzor_path /usr/local/bin/pyzor
- # modification de l'entête du mail par Pyzor? oui=1 non=0
- pyzor_add_header 1

[SAB-23]

3.4.4 Configuration de DCC

Vérifiez que vous êtes logués en tant que root sur le serveur pour effectuer l'installation de DCC.

Vous devez exécuter les commandes suivantes dans le répertoire /usr/ports/mail/dcc-dccd :

- Sinon, faire séparément :

```
#make
#make install
#make clean
```

L'installation est effectuée. Dans le fichier de configuration de SpamAssassin local.cf, ajoutez les lignes suivantes :

```
# utiliser DCC ? oui=1 non=0
use_dcc 1
# chemin d'acces au client DCC
dcc_path /usr/local/bin/dccproc
# modification de l'entête du mail par DCC? oui=1 non=0
dcc_add_header 1
```

3.4.5 Configuration des filtres Bayesiens

Dans le fichier de configuration de SpamAssassin local.cf, ajoutez les lignes suivantes :

```
# utiliser Bayes ? oui=1 non=0
use_bayes 1
# chemin d'acces à Bayes
bayes_path /usr/local/bin/
# auto apprentissage
bayes_auto_learn 1
#
bayes_min_ham_num 100
```

Vous devez vous procurer un fichier contenant des spam afin d'initialiser Bayes. Pour cela, procurez vous des mails de spam et générez un fichier contenant tout ces mails. Il est conseillé d'utiliser le programme « MailTest.exe » et d'envoyer la liste des mails à l'adresse IP du serveur SendMail à l'utilisateur root, le fichier concaténé de spam sera alors le fichier /var/mail/root. **[SAS-25].**

Une fois le fichier obtenu, lancez la commande :

```
#sa-learn --spam --showdots --mbox fichier_spam
```

4. Serveur FTP

Le File Transfer Protocol (*protocole de transfert de fichiers*), ou FTP, est un protocole de communication dédié à l'échange informatique de fichiers sur un réseau TCP/IP. Il permet, depuis un ordinateur, de copier des fichiers depuis ou vers un autre ordinateur du réseau, d'administrer un site web, ou encore de supprimer ou modifier des fichiers sur cet ordinateur.

La variante sécurisée de FTP avec les protocoles SSL ou TLS s'appelle FTPS.

FTP obéit à un modèle client-serveur, c'est-à-dire qu'une des deux parties, le *client*, envoie des requêtes auxquelles réagit l'autre, appelé *serveur*. En pratique, le serveur est un ordinateur sur lequel fonctionne un logiciel lui-même appelé serveur FTP, qui rend publique une arborescence de fichiers similaire à un système de fichiers Unix. Pour accéder à un serveur FTP, on utilise un logiciel client FTP (possédant une interface graphique ou en ligne de commande).

Le protocole, qui appartient à la couche session du modèle OSI et à la couche application du modèle ARPA, utilise une connexion TCP. Il peut être utilisé de deux façons différentes :

1. Mode actif

C'est le client FTP qui détermine le port de connexion à utiliser pour permettre le transfert des données. Ainsi, pour que l'échange des données puisse se faire, le serveur FTP initialisera la connexion de son port de données (port 20) vers le port spécifié par le client. Le client devra alors configurer son pare-feu pour autoriser les nouvelles connexions entrantes afin que l'échange des données soit fait. Ce mode est donc moins sécurisé que le FTP passif pour le client. De plus il peut s'avérer problématique pour les utilisateurs essayant d'accéder à des serveurs FTP lorsqu'ils sont derrière une passerelle NAT. Vu la façon dont fonctionne le NAT, le serveur FTP initie la connexion de données en se connectant à l'adresse externe de la passerelle NAT sur le port choisi. Si la passerelle NAT n'ayant pas de correspondance pour le paquet reçu dans sa table d'état le paquet sera ignoré et ne sera pas délivré au client.

2. Mode passif

Le serveur FTP détermine lui-même le port de connexion à utiliser pour permettre le transfert des données (*data connexion*) et le communique au client. Dans le cas de l'existence d'un pare-feu devant le serveur FTP celui-ci devra être configuré pour autoriser la connexion de données. L'avantage de ce mode, est que le serveur FTP n'initie aucune connexion. Dans le cas des clients FTP sur un réseau local, ce mode est beaucoup plus sécurisé que le FTP en mode actif, car le pare-feu ne devra laisser passer que les flux sortant vers internet pour permettre aux clients d'échanger des données avec le serveur. Il fonctionne donc sans problèmes avec une passerelle NAT.

Deux ports sont standardisés (*well known ports*) pour les connexions FTP : le port 21 pour les commandes et le port 20 pour les données.

Ce protocole peut fonctionner avec IPv4 et IPv6.

Pour accéder à un serveur FTP, on utilise un client ftp, en ligne de commande ou avec une interface graphique.

Les utilisateurs de GNU-Linux ou d'un Unix peuvent consulter une documentation (la plupart du temps installée par défaut), en tapant « man ftp »

La plupart des navigateurs récents autorisent les connexions FTP en utilisant une URL de type :

ftp://nom_d'utilisateur:mot_de_passe@nom_ou_adresse_du_serveur:port_ftp

En informatique anglophone on utilise login pour utilisateur et passwd ou password pour mot de passe.

Par mesure de sécurité, il est conseillé de ne pas préciser le mot de passe, le serveur le demandera. Cela évite de le laisser visible dans l'historique du navigateur, mais ne change rien au fait qu'il soit transmis en clair à travers le réseau.

Il existe de nombreux clients en ligne de commande, comme par exemple ftp, « Wget », « Curl » en téléchargement ou « Wput » pour déposer des fichiers.

La plupart des logiciels possèdent désormais une interface graphique permettant de mettre en place un serveur FTP ou de se connecter à celui-ci pour y copier des données (client FTP).

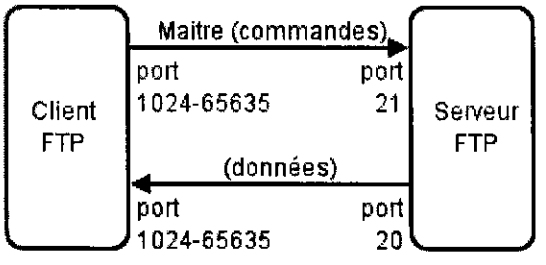


Fig. V.3 Connexion Data en mode Actif.

Le protocole utilise deux types de connexions TCP :

- Une connexion de *contrôle* initialisée par le client, vers le serveur (port 21 en général), pour transmettre les commandes de fichiers (transfert, suppression de fichiers, renommage, liste des fichiers…).
- Une connexion de *données* initialisée par le client ou le serveur pour transférer les données requises (contenu des fichiers, liste de fichiers), cette connexion utilise le protocole Telnet. Le client envoie une commande sous la forme d'une ligne de texte terminée par un retour à la ligne (CR suivi de LF, soit \r\n, ou 0D0A en hexadécimal).

Par exemple, la commande suivante demande le téléchargement du fichier "fichier.txt" :

```
RETR fichier.txt
```

N.B

Les commandes telles que GET ou PUT ne sont pas reconnues dans le protocole FTP, mais souvent utilisées par les logiciels de client FTP.

A la suite de l'envoi de la commande, le client reçoit une ou plusieurs réponses du serveur. Chaque réponse est précédée d'un code décimal permettant au client FTP de traiter la réponse qui peut comporter une ou plusieurs lignes de texte.

Pour l'exemple précédent, si le serveur trouve le fichier demandé, il envoie au client :

150 File status okay ;about to open data connection

Selon ce que le client et le serveur ont convenu, l'un des deux écoute sur le port TCP convenu, et l'autre s'y connecte pour établir la connexion de données. Puis le serveur envoie au client le contenu du fichier demandé, ferme la connexion de données, et envoie la réponse suivante sur la connexion de contrôle :

226 Closing data connection

La connexion de données est établie pour la durée de transmission de données (contenu de fichiers, ou liste de fichiers). En général, elle est établie pour le transfert de données d'une seule commande, à moins qu'un autre mode de transmission soit sélectionné et supporté par le serveur.

La commande « PASV » indique au serveur qu'il doit attendre passivement la connexion en écoutant un port TCP (en général, le port 20). Le port écouté par le serveur est indiqué dans la réponse :

277 Entering Passive Mode (h1,h2,h3,h4,p1,p2)

Où h1 à h4 sont 4 nombres entiers entre 0 et 255 représentant l'adresse IP du serveur, et p1 et p2 représentent le port TCP où le serveur attend la connexion, sous la forme de deux entiers entre 0 et 255 (port_TCP = p1 * 256 + p2).

Dans le cas contraire où le client attend la connexion sur un port TCP, il indique sous la même forme le port écouté en envoyant la commande PORT :

PORT h1,h2,h3,h4,p1,p2

Si tout se passe bien, le serveur répond :

200 Command Okay

Lors du transfert de fichier sur la connexion de données, 2 modes peuvent être utilisés:

- Le mode Binaire: le fichier est transmis tel quel.

- Le mode ASCII: uniquement destiné aux fichiers texte. Le fichier est examiné et des transformations apportées pour conserver un format correct. Par exemple, la fin de

  Ligne est représentée par le caractère <LF> sur un système UNIX, et par la paire <CR><LF> sous Windows. Une machine Windows recevant un fichier texte par FTP récupère donc au final un fichier avec des <CR><LF> en mode ASCII et des <CR> en mode binaire. Ce mode a donc ses avantages, mais peut être source de corruption de fichiers (non texte) pendant le transfert si on utilise un client ancien / en ligne de commande, incapable de s'adapter au type de fichier. Il faut alors basculer de mode (en utilisant généralement la commande BIN) avant le transfert, afin de le conserver intact.

La spécification du protocole FTP (RFC 959) précise qu'il est possible d'effectuer un transfert de fichiers directement entre deux serveurs FTP. Cette fonctionnalité est peu connue, car non disponible dans les logiciels de client FTP.

Pour cela, le client établit une connexion de contrôle par serveur. Au moment d'établir la connexion de données, le client demande à un serveur d'attendre la connexion (commande PASV), et capture l'adresse IP et le port TCP écouté. Ces deux paramètres sont envoyés à l'autre serveur en utilisant la commande PORT. A partir de là, la connexion de données est établie entre les deux serveurs.

Le client est informé de la fin du transfert en recevant une réponse positive de chaque serveur, on appelle couramment ce protocole le FXP.

5. Linux Redhat

Linus B.Torvalds est à l'origine de ce système d'exploitation entièrement libre. Au début des années 90, il voulait mettre au point son propre système d'exploitation pour son projet de fin d'étude. Linus Torvalds avait pour intention de développer une version d'UNIX pouvant être utilisé sur une architecture de type 80386. Le premier clone d'UNIX fonctionnant sur PC a été Minix, écrit par Andrew Tanenbaum, un système d'exploitation minimal pouvant être utilisé sur PC. Linus Torvalds décida donc d'étendre les possibilités de Minix, en créant ce qui allait devenir Linux. Amusées par cette initiative, de nombreuses personnes ont contribué à aider Linus Torvalds à réaliser ce système, si bien qu'en 1991 une première version du système a vu le jour. C'est en mars 1992 qu'a été diffusée la première version ne comportant quasiment aucun bug, avec le nombre croissant de développeurs travaillant sur ce système, celui-ci a rapidement pu intégrer des redéveloppements libres des outils présents sous les systèmes UNIX commerciaux. De nouveaux outils pour Linux apparaissent désormais à une vitesse vertigineuse.

L'originalité de ce système réside dans le fait que Linux n'a pas été développé dans un but commercial.

En effet aucune ligne de code n'a été copiée des systèmes UNIX originaux (en effet Linux s'inspire de nombreuses versions d'UNIX commerciales: BSD UNIX, System V.). Ainsi, tout le monde, depuis sa création, est libre de l'utiliser mais aussi de l'améliorer.

Il existe plusieurs façons de prononcer Linux :

- Dans les pays anglo-saxons deux prononciations existent :
  - Lineuxe.
  - Laïnuxe.
- En France on prononce tout simplement :
  -Linux

Bien que Linux ait été initialement conçu pour fonctionner sur plateforme PC, il a désormais été porté (c'est-à-dire adapté) vers de nombreuses autres plates formes, telles que Macintosh, stations SPARC, stations DEC Alpha, et même des plates formes telles que des assistants personnels PDA, voire des consoles de jeu vidéo !

5.1 Notions de noyau et de distribution

Linux est architecturé autour d'un **noyau** (***kernel***) chargé de prendre en charge le matériel. On appelle **distribution** l'assemblage d'un ensemble de logiciels autour d'un noyau Linux afin de fournir un système clé en main. Le noyau d'une distribution peut-être mis à jour afin de permettre la prise en compte de matériels récents, toutefois cette manipulation consistant à recompiler le noyau est délicate car elle nécessite un certain niveau de connaissance du système et du matériel. La recompilation du noyau est à réserver aux spécialistes ou bien aux utilisateurs prêts à rendre inutilisable leur système dans le but d'apprendre.

5.2 Caractéristique du système :

Linux est un système d'exploitation proche des systèmes UNIX pouvant être exécuté sur différentes plates-formes matérielles : x86 (c'est-à-dire des plates-formes à base de processeurs Intel, AMD, etc.), Sparc, PowerPC, Alpha, ARM, etc.
Linux est ainsi un système multi plate-forme. Il est également multi-utilisateurs (plusieurs personnes peuvent en même temps travailler sur le même ordinateur), mais aussi multi-tâches (plusieurs applications peuvent être lancées en même temps sans qu'aucune n'affecte les autres) à noter qu'il est aussi multiprocesseur.
Linux est considéré comme un système fiable, robuste et puissant. Il est d'ailleurs capable de fonctionner avec très peu de ressources sur des ordinateurs bas de gamme très peu puissants.
Le système Linux possède notamment les avantages suivants :

- Le support des standards de l'internet, c'est-à-dire des protocoles TCP/IP, la famille de protocoles utilisée sur Internet. Linux est donc un moyen gratuit de créer un réseau local, de se connecter à Internet et de mettre en place un serveur.
- Une sécurité accrue dû à la transparence de son code source et de la réactivité de la communauté lors des annonces de vulnérabilités.

- Un cloisonnement des espaces mémoire et de l'espace disque couplé à une gestion pointue des droits permettant de gérer un grand nombre d'utilisateurs avec un niveau de risque minimal.
- Un noyau entièrement configurable en fonction du matériel de la machine sur laquelle le système est installé afin de maximiser les performances.
- Open source (on dit libre en français), et gratuit (aucune licence à payer).
- Compatibilité multi architecturales.
- Une distribution Linux récente peut fonctionner sur des ordinateurs à faible performances (à partir des Architecture Intel 386).
- Peut s'installer et démarrer sur une clé USB (via votre BIOS).
- Peut prendre en charge plusieurs versions de noyau dans un même système.
- Est beaucoup moins gourmand en ressources systèmes.
- Accepte beaucoup plus de requêtes sollicitées à la seconde.
- Linux est un vrai système multitâche.
- Gestions de plusieurs types de partitions Linux et Windows (fat16, fat32, ntfs, ext2fs, ext3fs, ext4fs (?), reiserfs, xfs,...).
- Pas besoin de défragmenter les partitions de type Linux (ext2fs, ext3fs, reiserfs, xfs,...).
- Linux peut avoir plusieurs interfaces graphiques et de bureaux virtuels.
- Est parfaitement fonctionnelle quand Linux est installé en mode terminal (c'est-à-dire sans Interface graphique).
- L'interpréteur de commande n'est pas intégré de manière fixe au noyau.
- Les commandes dans un terminal peuvent interagir avec les programmes graphiques.
- Un crash d'un environnement graphique (comme KDE ou Gnome) n'entraîne pas Un crash du système. L'interface graphique est totalement indépendante du système.
- Possibilité de télécharger et de gérer des milliers de logiciels via les outils de mise à jour du Système (qui est personnalisable) comme l'utilitaire Synaptic pour les paquetages Debian (.deb).
- L'installation des paquetages dans Linux est plus uniformisée (voir organisée) que ceux dans Windows (.exe, .msi), ce qui contribue aussi à la stabilité de Linux et des logiciels installés.
- Les bogues critiques sont corrigés rapidement.
- Une sécurité plus sévère et efficace sur l'accessibilité des programmes.
- Dépendance partielle du BIOS de l'ordinateur car Linux possède son propre BIOS (Noyau) détections matériels.
- A une plus grande tolérance aux pannes que Windows.
- Gestion et administration de tout le système à distance.
- Existe en version Live CD parfaitement fonctionnelle.
- Le copier/coller rapide avec sélection d'un texte.

5.3 Inconvénients :

- Est plus complexe que Windows.
- Le nombre impressionnant de distributions Linux peut dérouter le débutant Linuxien.
- Pas assez de publicité pour mieux faire connaître Linux dans le monde.
- Un manque de pilotes ou de drivers propriétaires pour Linux.
- Pas suffisamment de logiciels propriétaires professionnelle.

CHAPITRE VI.

Etude des HoneyPot.

Prefer the standard to the offbeat.

– Strunk & White.

1. Définition

Les HoneyPot (*pots de miel*) sont des systèmes vulnérables déployés sur Internet et surveillés par des administrateurs et des experts de sécurité. Les HoneyPot sont utilisés pour faire de la collection d'informations sous forme de fichiers journaliers (*log files*).
Cette information est utilisée pour connaitre les nouvelles techniques, outils et motivations des hackers pour mieux parer contre les attaques sur les systèmes en production. Les HoneyPot déployés pour ce type d'utilisation sont appelés des HoneyPot de recherche.

Un HoneyPot peut protéger une organisation par trois moyens :

- La prévention des attaques : laissé un hacker jouer sur l'honeypot au lieu des systèmes de production.
- La détection d'activités malicieuses : utilisation des systèmes de détection d'intrusions (IDS).
- La réponse aux attaques : routage de tout trafic malencontreux vers des honeypots.

Lorsqu'un HoneyPot protège une organisation c'est un honeypot de production. Comparé à un honeypot de recherche, un honeypot de production génère moins de trafic. **[HP-28].**

Le tableau suivant donne un récapullatif de ces deux types de HoneyPot :

Niveau d'interaction	Type d'honeypot	Utilité de l'honeypot
Faible	Production	Protection d'une organisation
Forte	Recherche	Collecte d'informations

Table VI.1 Caractéristiques des deux types d'HoneyPot.

Théoriquement, un HoneyPot n'a pas normalement de trafic parce qu'il n'a pas d'activité légitime. Cela veut dire que, chaque interaction avec un honeypot signifie un malicieux ou une mauvaise activité. Une connexion quelconque à un HoneyPot est considérée comme une attaque ou compromise. Grâce à sa simplicité, il a énormément d'avantages et bien évidemment des inconvénients.

2. Avantages et Inconvénients

2.1 Avantages

- Le HoneyPot est une conception très simple qui donne des petits ensembles de données mais de haute valeur : le HoneyPot collecte un petit nombre d'information. Au lieu de collecté des Go par jour, il peut collecter environ un Mo de donnée par jour. Au lieu de générer 10 milles alertes par jour, il ne peut générer que 10. Mais il faut souligner que le honeypot ne capture que des mauvaises activités. Chaque interaction avec un honeypot signifie une activité dangereuse. Alors, l'HoneyPot réduit le taux de collecte, toute en gardant l'information très valable, qui ne concerne que l'attaquant. Cela signifie qu'il est plus facile (et moins chère) à analyser les données collectées pour dériver la valeur à partir de ces données.
- Ressource minimale : Un HoneyPot n'a besoin que de peu de ressources pour capturer les mauvaises activités. Un ancien ordinateur Pentium avec 128Mo de mémoire peut facilement contrôler la classe B du réseau.
- Chiffrement ou IPv6 : Différence à presque toutes les technologies de sécurité (*comme IDS système*), l'honeypot travaille bien dans l'environnement Ipv6 ou avec des données chiffrées. N'importe quelles données (instruction ou commande) qu'un attaquant envoie, sera détectée, capturée et analysée.
- Simplicité : En fin, l'honeypot est simple, il n'y a pas d'algorithme à développer et il est facile à configurer.

2.2 Inconvénients

Comme toutes les technologies, l'honeypot a aussi des faiblesses. Parce qu'il ne remplace aucune technologie courante, mais il travaille avec les technologies existantes.

- Vue limitée : l'HoneyPot ne peut détecter et capturer que des activités directement interactives avec lui.
- Risque : Toutes les technologies de sécurité ont des risques. Le pare-feu a le risque d'être pénétré, le chiffrement à risque d'être démoli. L'IDS a le risque de se tromper. L'honeypot a

aussi des risques. Il a un risque d'être pris la relève par un attaquant et d'être utilisé à faire du mal aux autres systèmes. Les risques sont variés selon chaque type d'honeypot.

3. Les types d'HoneyPot

Il y a deux catégories d'HoneyPot: *low-interaction* et *high-interaction* honeypots. Ces deux catégories nous donnent des connaissances sur quel type d'honeypot on travaille et avec qui on a une interaction, sa faiblesse et sa force. L'interaction définie le niveau d'activité qu'un honeypot permet aux attaquants. Low-interaction HoneyPot a un nombre d'interaction limité, il travaille grâce à une simulation des services et des systèmes d'exploitation. Les activités des attaquants sont limitées au niveau de la simulation par l'honeypot. Par exemple, Un service simule SMTP entendu sur le porte 25 peut simuler l'ouverture d'une session SMTP ou il est également supporter quelques autres commands SMTP. L'avantage d'une low-interaction est la simplicité. L'honeypot a tendance à déployer et maintenir facilement avec un risque minimal. Normalement, il a besoin d'installer des logiciels, de sélectionner des services ou de système d'exploitation que l'on veut simuler et surveiller. Ainsi l'attaquant n'accède jamais au système d'exploitation principal pour lui faire du mal.

L'inconvénient principal de low-interaction HoneyPot est que les informations qu'il a collectées sont limitées. Donc, il est plus facile pour un attaquant de détecter une low-interaction HoneyPot. Bien qu'un low-interaction HoneyPot soit bien configuré, un attaquant qualifié peut détecter sa présence. Les exemples d'honeypot sont: *Spector, Honeyd et KFSensor.*

High-interaction HoneyPot est différent. Il est a des solutions complexes parce qu'il a besoin d'applications réelles et d'un système d'exploitation réel. Aucun n'est simulé, on donne aux attaquants des choses réels. Si on veut un HoneyPot Linux qui lance le serveur SMTP, on construit un système Linux réel qui lance un serveur SMTP (Sendmail par exemple). Il y a deux avantages majeurs de cette solution :

- on peut capturer extensivement des informations. En donnant aux attaquants un système réel à communiquer avec, on peut suivre tous ses comportements.
- une High-interaction HoneyPot ne fait pas attention à comment un attaquant se comportera. Au lieu de cela, elle propose un environnement ouvert qui capture toutes les activités.

En générale, une High-interaction HoneyPot peut faire ce qu'un low-interaction honeypot fait et beaucoup plus. Mais, il est plus complexe à déployer et maintenir. Voici quelque HoneyPot High-interaction : *Symantec Decoy Server, Honeynet.*
Pour mieux comprendre le mode de fonctionnement des deux types de HoneyPot cités ci-dessus, on va commencer par : low-interaction HoneyPot ou bien le « *Honeyd* ». **[HHN-30].**

3.1 Low-interaction HoneyPot « Honeyd »

Honeyd est un HoneyPot low-interaction développé par *Niels Provos*. C'est un Open Source conçu pour le système Unix. Il surveille les espaces IP qui ne sont pas encore utilisés.
Quand il trouve une connexion il l'accepte et commence à communiquer avec l'attaquant en se considérant comme victime.

Par défaut, Honeyd détecte et collecte n'importe quelle connexion vers les ports TCP et UDP. En plus, on peut configurer les services simulés pour contrôler les ports spécifiques comme un serveur FTP simulé qui contrôle le port 21ou un serveur mail comme un serveur SMTP simulé qui contrôle le port 25.
Quand un attaquant connecte au service simulée, honeyd non seulement détecte et collecte les activités, mais il capture aussi les interactions de l'attaquant avec le service simulée. Dans le cas de serveur SMTP, on peut potentiellement capturer le compte que l'attaquant a utilisé, les commandes qui a introduit, ainsi que sont identité.
Quelque honeypots, comme Honeyd, peuvent non seulement simuler des services, mais simuler des systèmes d'exploitation actuels. D'autre part, Honeyd peut apparaître comme routeur Cisco, web server Win XP ou serveur DNS sous Linux. Il y a certains avantages quand on simule des systèmes d'exploitation différents.

D'abord, l'HoneyPot peut être mieux mélangé dans le réseau existant s'il a la même apparence et comportement du produit système. En suite, on peut choisir des attaquants spécifiques en leur proposant des systèmes et des services pour mieux connaitre leurs objectifs ou on proposant ceux que l'on veut étudier. La clé de toutes ces simulations est l'existence d'un élément relié au système d'exploitation simulé, c'est le service simulé.
Quand un attaquant connecte au service simulé, ce service a un comportement et apparaît comme un sous système d'exploitation spécifique. Par exemple, si on a un service simulé d'un serveur web et on veut que notre HoneyPot apparaît comme Window2000 serveur, on peut simuler son comportement à la façon qu'un IIS web server se comporte. Pour Linux, on peut le simuler à la façon d'un web server Apache ; et de même pour simuler un serveur Mail sous linux en utilisant un serveur de messagerie comme Sendmail, La plupart des honeypots simulent le système d'exploitation selon cette manière. **[HPW-27].**

3.2 High-interaction HoneyPot « Honeynet »

Honeynet est un exemple de HoneyPot High-interaction. Il n'est pas un produit, ni une solution logiciel que l'on installe sur un ordinateur. Au lieu de cela, honeynet est une architecture, une entité du réseau des ordinateurs construits pour être attaqué. L'idée est de proposer une architecture qui crée un réseau contrôlé à haut niveau où toutes les activités sont contrôlées et capturées.
Dans le réseau on met des ordinateurs intentionnellement vulnérables, ce sont des ordinateurs réels qui lancent des applications réelles. L'attaquant trouve une fuite d'attaque et entre dans les systèmes proposés. Quand il fait cela, il ne prend pas conscience qu'il est dans un Honeynet.
Toutes ses activités, de la session SSH encryptée vers email et téléchargement des fichiers, sont capturées sans aucune connaissance de l'attaquant.
Cela se fait par insertion des modules dans le noyau du système victime qui capture toutes les actions des attaquants. En même temps, l'honeynet contrôle les activités des attaquants. Il le fait en utilisant une passerelle Honeywall. Cette passerelle permet le trafic d'entrer vers le système victime, mais contrôle le trafic sortie en utilisant la technologie de prédiction des instructions. Cela donnera aux attaquants une flexibilité dans l'interaction avec le système victime. L'exemple du déploiement de honeynet est dans la figure suivante:

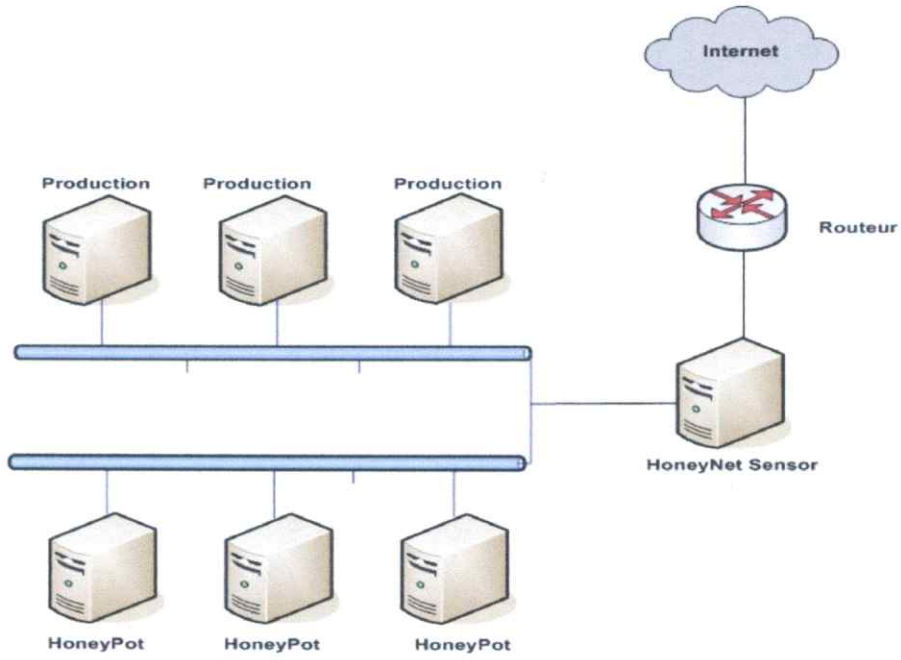


Fig.VI.1 Le schéma de HoneyNet.

3.2.1 Contrôle de données

Le HoneyNet doit contenir toute activité. S'il est compromis le honeypot ne doit pas compromettre d'autres systèmes ne faisant pas partie du réseau d'honeypot. Si le honeypot est placé derrière un pare-feu ou routeur, les connexions en sortie peuvent effectivement être contrôlées.
Le honeypot doit être souple. Pour qu'il n'y ait pas une suspicion de la part du hacker sur l'honeypot, les droits d'utilisation sur le honeypot ne doivent pas être limités uniquement à la lecture. Il faut aussi permettre l'exécution ou l'écriture des fichiers contrôlée. **[HP-28].**

3.2.2 Capture des données

Toutes les activités sur l'honeynet doivent être enregistrées. Il est important de pouvoir capturer le plus d'informations possibles sur le hacker sans qu'il le sache.
Par contre, les données enregistrées ne devraient pas être stockées uniquement sur l'honeypot. Il faut pouvoir enregistrer les données depuis plusieurs sources (fichiers de log du système, log des IDS, analyseurs réseau). Il est important aussi pour la capture et le contrôle de données, d'assurer l'intégrité de données. **[HP-28].**

3.2.3 Collection des données

Dans un environnement distribué qu'il soit un réseau physiquement distribué à l'échelle mondiale ou un réseau logiquement distribué, les données doivent être stockées dans un lieu central. La corrélation de ces données par rapport au temps permettra d'étudier les motifs d'attaque d'un seul ou d'un groupe de hackers. **[HP-28].**

3.2.4 Génération de HoneyNet

1. La 1ère génération des honeynets

Sur les HoneyNet de 1ère génération, le contrôle et la capture des données ont été implémentés via plusieurs dispositifs (routeur, pare-feu, IDS) permettant l'enregistrement et le contrôle de données en Multi-couches. **[HP-28].**

La figure suivante illustre la 1ère génération des HoneyNet :

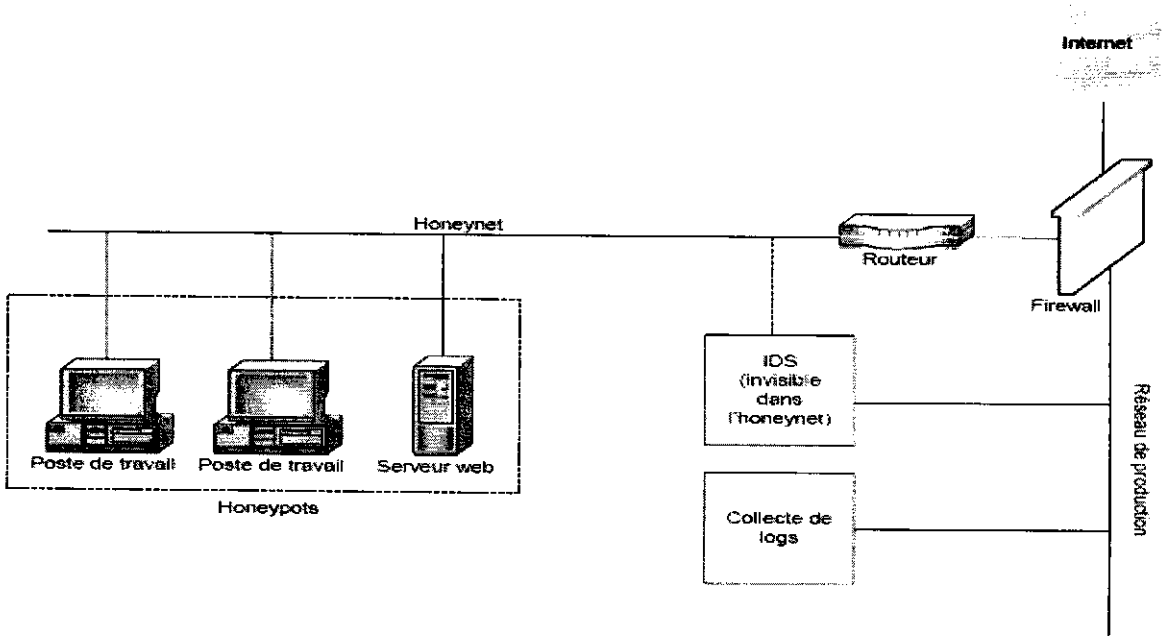


Fig.VI.2 1ère génération de HoneyNet.

L'avantage de multicouche est que si un des dispositifs tombe en panne, la surveillance et la capture de données sont encore garanties. Par contre, ce type d'honeynet est difficile à administrer dû à la complexité de l'architecture réseau.
Voici comment le contrôle et la capture des données sont implémentés sur un honeynet de première génération.

a. Le contrôle de données

- Le firewall offre la première couche de protection. Le firewall ne contrôle pas le nombre de connexions vers l'extérieur mais limite le nombre de connexions en entrée.
- Le routeur limite toute connexion vers l'Internet. Il permet aussi de cacher le firewall du réseau intérieur. C'est la deuxième couche de protection.

b. La capture de données

- Le firewall garde un journal de toutes les connexions de l'Internet et de honeynet (1ère couche de capture).
- Le système de détection d'intrusion garde aussi un log de toutes les activités du réseau d'honeynet. Il est accessible depuis le réseau de production mais non pas depuis le honeynet (2ème couche de capture).
- Sur les HoneyPot eux-mêmes, les journaux du système (système logs) peuvent être exploitables (3éme couche de capture).

2. La 2ème génération de HoneyNet

Sur un HoneyNet de deuxième génération, le contrôle et la capture de données sont implémentés sur un seul dispositif. Dans l'honeynet de première génération, l'élément de contrôle est principalement le pare-feu, mais celui-ci est remplacé par un pont (bridge) dans un honeynet de deuxième génération. Le bridge est difficilement détectable car il n'a pas d'adresse IP et il ne décrémente pas le champ TTL. **[HHN-30].**

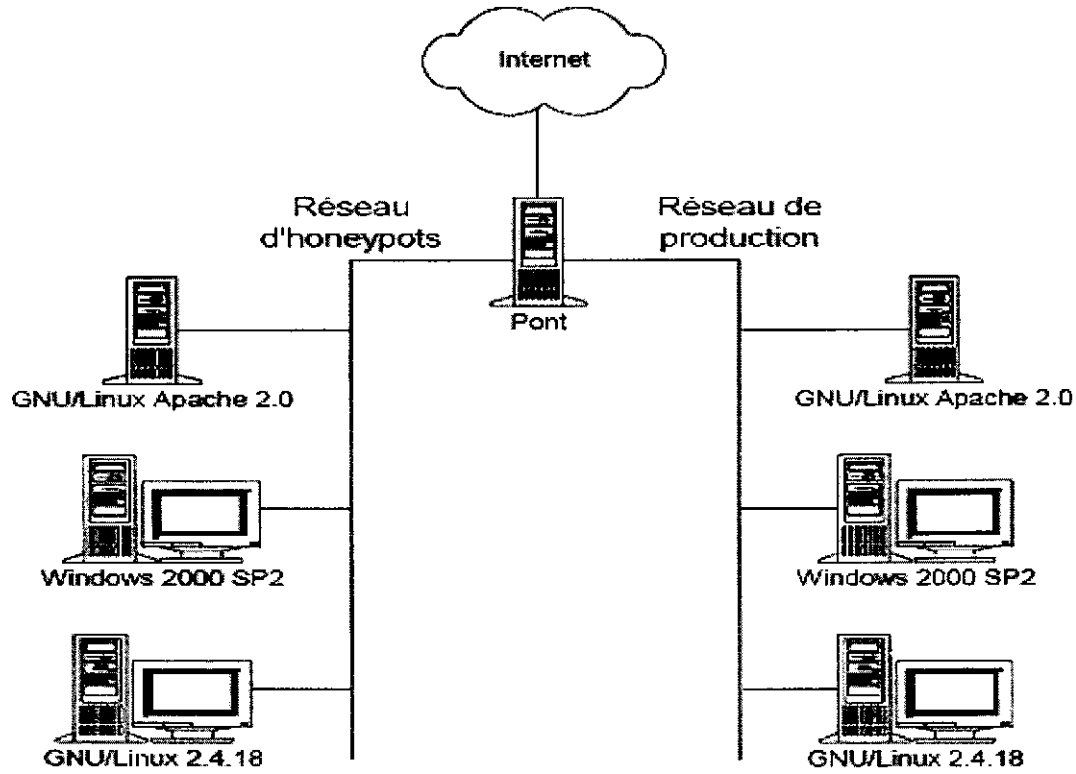


Fig.VI.3 2ème génération de HoneyNet.

Le principe de HoneyPot de 2ème génération est la détection du trafic malveillant destiné au réseau de production, puis le routage de celui-ci vers le réseau de honeypot.

3.3 HoneyPot virtuel

Un HoneyPot virtuel est une simulation d'architecture d'une machine sur une machine servant comme hôte. Cette possibilité permet de simuler un réseau complexe d'honeypots sur une seule machine. Les honeypots virtuels peuvent être classés en deux catégories:

- Dans un HoneyPot purement virtuel, les dispositifs de contrôle et capture de données ainsi que les honeypots sont déployés sur une seule machine.

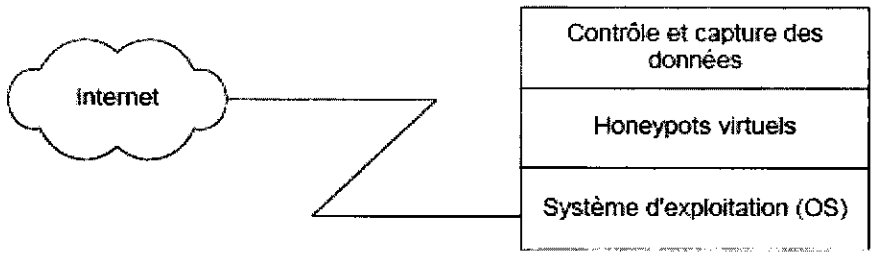


Fig.VI.4 HoneyPot Virtuel.

- Dans un HoneyPot virtuel hybride, les dispositifs de contrôle et de capture de données sont déployés sur une machine séparée.

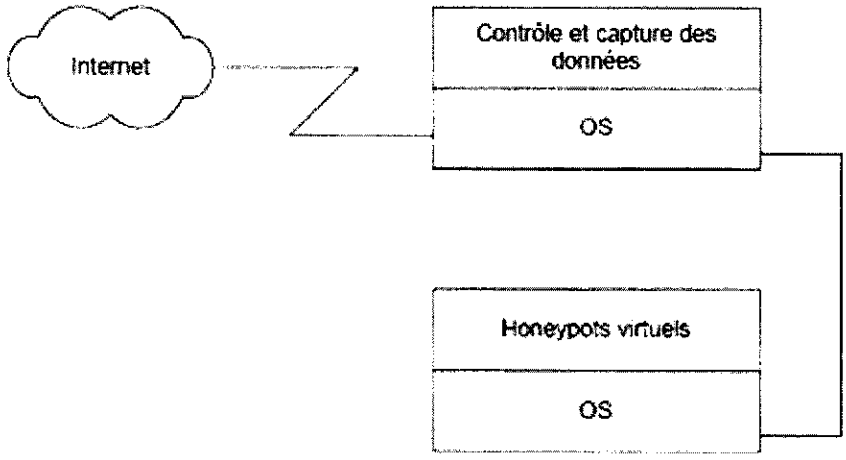


Fig.VI.5 Architecture du HoneyPot Virtuel hybride.

Un HoneyPot purement virtuel permet une administration facile sur une seule machine. Toutefois si un hacker parvient à prendre contrôle de l'hôte, tout l'honeynet est compromis, ce qui n'est pas le cas dans un honeynet virtuel hybride.

4. Les HoneyPot virtuels avec Honeyd

4.1 Introduction

Les HoneyPot virtuels permettent une mise en oeuvre facile des honeypots due à la simplicité de leur installation et leur administration qui peut se faire sur une seule machine hôte. La simulation des honeypot virtuels avec *honeyd* permet de créer une topologie de serveurs et machines vulnérables et attirantes aux hackers. De plus, des possibilités de routage offertes permettent la simulation par exemple de la décrémentation de TTL des paquets, la latence et même des paquets perdus.

4.2 Fonctionnement

4.2.1 Fichier de configuration

Honeyd est un daemon exécuté sur une machine qui répond aux paquets destinés aux honeypots virtuels qu'il simule.

4.2.2 HoneyPot client

Récemment, Christian Seifert (*Honeynet New-Zealand*) a publié un outil très intéressant. HoneyC de son nom, permet non pas d'attendre passivement les requêtes des botnets mais HoneyC via le moteur de recherche Yahoo ! Envoi des requêtes http afin de détecter des URLs hébergeant des malwares (notion de client honeypot). On note l'initiative *stopbadware.org* dont Google fait parti qui a recensé, à l'écriture de cet article, 405 URL dangereuses. Si vous ouvrez une de ces URLs après une requête de recherche sur Google, alors vous serez redirigé vers un message d'avertissement indiquant que l'URL contient un malware.

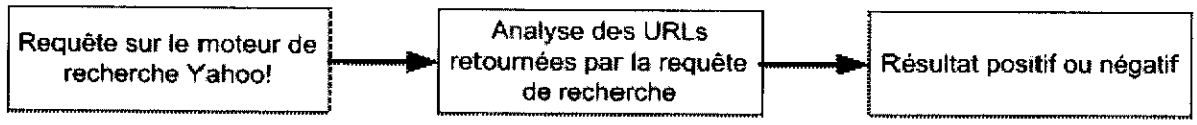


Fig.VI.6 Architecture de mode de fonctionnement du HoneyC.

4.2.3 Valeurs de HoneyPot

HoneyPot aide à prévenir par certaines façons. Premièrement, il contre les attaques automatiques. Ces attaques basent sur les outils qui scannent les entités du réseau de façon aléatoire pour chercher les vulnérabilités du système. Si une des vulnérabilités est trouvée, les outils automatiques attaqueront le système et obtiendront le droit d'accès au système.

Deuxièmement, l'HoneyPot peut aider à contrer ces attaques par un retardement de scans, même s'il faut les arrêtés. Les HoneyPot retardent les actions des hackers. Ils les font en utilisant une variété de valeurs, comme Window taille zéro... les HoneyPot peuvent protéger votre organisation des attaquants humains. L'idée est de refuser un attaquant, lui faire perdre son temps

à communiquer avec les honetpots. Pendant ce temps là, votre organisation détecte les activités de l'attaquant et a assez de temps pour répondre ou arrêter cet attaquant. Si l'attaquant sait que votre organisation utilise l'honeypot, mais il ne sait pas quels systèmes sont les honeypots et quels systèmes sont les machines légitimes. Il a des risques d'être pris dans le système d'honeypot et il décide donc de ne pas attaquer votre organisation. L'honeypot dissuade l'attaquant. Un exemple est Déception *Toolkit,* qui est un *low-interaction HoneyPot.* **[HPF-29].**

5. Conclusion

Malgré l'existence des hackers chevronnés qui réussissent souvent à franchir les systèmes hautement sécurisés, ils peuvent êtres pris au piège par les HoneyPot et suivi dans toutes leurs activités tout cela sans se rendre compte, c'est l'objectif de la mise en place d'un tel système.

CHAPITRE VII.

Format de message.

"The time has come," the Walrus said,
"to talk of many things".

– L.Carroll.

1. Introduction

Pour pouvoir arriver à bien comprendre les systèmes les plus compliqués qui existent il faut bien comprendre les éléments les plus atomiques constituant ces gros systèmes et déterminer avec exactitude les objets manipulés, ce qui rend l'interaction avec de tels systèmes plus flexible.

2. Composants génériques

Les messages électroniques tout comme les messages postaux sont composés d'une **enveloppe** et d'un **contenu.**

L'enveloppe contient les informations nécessaires pour effectuer la transmission et la distribution. Tandis que le contenu contient l'objet à livrer au destinataire.

Cependant, quelques systèmes de messagerie peuvent utiliser une partie de l'information qui est situé dans le contenu pour créer une enveloppe. Ce standard a pour intention de faciliter l'acquisition de telles informations par programmes. Ainsi que d'autres systèmes de messagerie peuvent enregistrer les messages dans des formats qui diffèrent de celui définit par ce standard.

Les messages sont des **lignes de texte**. Aucune spécification particulière n'est prévue pour coder les dessins, les textes structurés ou tableaux. On n'a pas pris en considération les questions de compression ou d'efficacité de stockage ou de transmission et le standard tend à être libéral au niveau du nombre de bit consommé.

Une structure plus robuste permet le codage de l'information avec des fontes multiples, multicolores et multidimensionnelles. Une structure moins robuste, présente dans la plupart des machines restreindrait, avec plus de sévérité, le potentiel à ajouter des champs et de décider d'inclure des champs spécifiques. Contrairement à la communication basée sur le papier, il est intéressant de noter que le destinataire d'un message peut exercer une quantité extraordinaire de contrôle sur l'apparence du message.

3.3.1 Description générale

Un message comprend des champs d'en-tête, et facultativement un corps.
Le corps est une simple séquence de lignes contenant des caractères ASCII. Il est séparé des en-têtes par une ligne nulle (c'est à dire une ligne que rienne précède appelée un retour chariot à la ligne ou bien CRLF).

3.2 Les champs d'en-tête

Chaque champ d'en-tête peut être vu comme une simple ligne logique de caractères ASCII, comprenant un **nom** de champs et un **corps** de champs.

Par commodité, la portion du corps de champ de cette entité conceptuelle peut être découpée en une représentation multi-ligne. On appelle cela le "**plissage**" (*folding*). La règle générale est que, partout où il peut y avoir un retour à la ligne avec des espaces blancs (*linear-white-space (LWSP)*) -pas simplement un caractère LWSP- un retour chariot à la ligne (*CRLF*) immédiatement suivie par **au moins** un caractère LWSP peut être inséré à sa place.

Ainsi la simple ligne :

```
To: "Joe & J. Harvey" <user @Org>, JJV @ BBN
```

Fig. VII.1 Champ d'en-tête « TO » model 1.

peut être représenté par:

```
To: "Joe & J. Harvey" <user @ Org>,
                    JJV@BBN
```

Fig. VII.2 Champ d'en-tête « TO » model 2.

et par

```
To: "Joe & J. Harvey"
                <user @ Org>, JJV
@BBN
```

Fig. VII.3 Champ d'en-tête « TO » model 3.

et par

```
To:"Joe &
J. Harvey" <user @Org>, JJV @ BBN    .
```

Fig. VII.4 Champ d'en-tête « TO » model 4.

Le processus qui consiste à modifier cette représentation multi ligne de champs d'en-tête en une représentation mono ligne est appelé "**déplissage**". Le déplissage s'accomplit en considérant le CRLF immédiatement suivi par le caractère LWSP, comme un simple caractère LWSP.

Remarque: Alors que le standard autorise le plissage n'importe où une ligne d'espace blanc est permise, il est recommandé pour les champs structurés, tel que ceux contenant les adresses, de limiter le plissage aux coupures syntaxiques de haut niveau. Pour les champs d'adresses, il est recommandé que de tels plissages se passent entre les adresses, après les virgules (comma) de séparation.

Une fois que le champ a été déplissé, la structure peut être vue comme étant composée d'un nom de champ suivie de deux points (:) suivie par le corps du champ et terminée par un retour chariot à la ligne (CRLF). Le nom du champ doit se composer de caractère ASCII imprimable c'est à dire des caractères ayant une valeur ASCII comprise entre 33 et 126, des décimaux, exception faites aux deux points.

3.3 Définition des champs d'en-tête

Ces règles indiquent une méta-syntaxe de champ, sans chercher dans la syntaxe interne ou les types particuliers. Leur objectif est de permettre la détection des champs ; Ainsi, elles donnent, à travers une analyse syntaxique de haut niveau, une image de chaque champ sur une seule ligne. Le tableau suivant illustre ces différents champs :

champ	nom de champ":" [corps du champ] CRLF.
nom de champ	n'importe quel CHAR, à l'exclusion des CTL, SPACE, et ":".
corps de champ	contenu du corps de champ [CRLF caractère-LWSP corps de champ].
contenu du corps de champ	Les caractères ASCII créent le corps de champ, comme ce qui est défini dans les sections suivantes, qui consiste à des combinaisons d'atome, des chaînes entre guillemets, et marqueurs spéciaux, ou autre objets comprenant du texte.

Table VII.1 Les Champs d'en-tête.

4. Analyse Syntaxique

4.1 Champs spécifiques

La syntaxe indique que, quand ils sont présents, certains champs, doivent être dans un ordre particulier.

Cependant il **n'est pas** nécessaire que les champs d'en-tête soient dans un ordre particulier, excepté que le corps du message doit être **après** les en-têtes. Il est recommandé, que s'ils sont présents, les en-têtes sont envoyés dans l'ordre suivant :

"Return-Path", "Received", "Date", "From", "Subject", "Sender", "To", "cc", etc.

Le tableau suivant illustre les différents champs syntaxiquement spécifiés:

Champs	Définition	Commentaires
message	champs *(CRLF *texte)	Tout ce qui est après la ligne nulle est le corps du message.
champs	Dates Source Destination Champs optionnel	Instants de création. Auteur id & une. Adresse nécessite. D'autres champs.
source	[trace] Origine [resent]	Réseau traversé. Courrier originel. Renvoyer, fais suivre.
trace	return received	Chemin vers l'expéditeur. Marque le destinataire.
return	"Return-path" ":" route-addr	Adresse de retour à l'envoyeur.
received	"Received" ":" ["from" domaine] ["by" domaine] ["via" atome] ("with" atome) ["id" msg-id] ["for" addr-spec] ";" date-time	Un par relais Hôte expéditeur Hôte receveur Chemin physique Protocole de courrier/lien identificateur de message Forme initiale Temps de réception (time received)
originator	Authentic ["Reply-To" ":" 1#address]	Adresse authentifiée.
authentic	"From" ":" mailbox ("Sender" ":" mailbox "From" ":" 1#mailbox)	Un simple auteur. Soumissionnaire réel. Auteurs multiples ou non expéditeur.
resent	resent-authentic ["Resent-Reply-To" ":" 1#address])	

resent-authentic	"Resent-From" ":" mailbox ("Resent-Sender" ":" mailbox "Resent-From" ":" 1#mailbox)	
dates	origin-date [resent-date]	Date d'origine. Date de renvoie.
orig-date	"Date" ":" date-temps	
resent-date	"Resent-Date" ":" date-temps	
destination	"To" ":" 1#address "Resent-To" ":" 1#address "cc" ":" 1#address "Resent-cc" ":" 1#address "bcc" ":" #address "Resent-bcc" ":" #address	Destination primaire. Destination secondaire. Copie invisible (Blind carbon).
optional-field	"Message-ID" ":" msg-id "Resent-Message-ID" ":" msg-id "In-Reply-To" ":" *(phrase / msg-id) "References" ":" (phrase / msg-id) "Keywords" ":" #phrase "Subject" ":" text "Comments" ":" text "Encrypted" ":" 1#2word extension-field user-defined-field	Destinés à êtres défini et peuvent êtres pré-vidés.
msg-id	"<" addr-spec ">"	Id unique de message.
Extension Field	N'importe quel champ qui est défini dans un document publié comme une extension formelle à cette spécification.	
user-defined-field	Des champs spécifiés par l'utilisateur et les noms pour ces champs doivent être unique et peuvent être pré-vide par des extensions publiques.	

Table VII.2 Les Marqueurs lexicaux.

4.2 Éclaircissements

4.2.1 Faire suivre « *Forwarding* »

Certains systèmes permettent aux destinataires des courriers de répondre aux messages (*forwarding*), en s'appropriant l'en-tête d'origine, et en y ajoutant quelques champs nouveaux.

Généralement les champs ainsi ajoutés contiennent l'en-tête du message original inséré au début du corps du nouveau message.

4.2.2 Reçu « *received* »

Une copie de ce champ est ajoutée par le service de transport qui relaye le message. Cette information peut être assez utile pour suivre les problèmes de transport.

Les noms d'hôtes d'expédition et de réception et de date-temps de réception peuvent être spécifiés. Le paramètre "via" peut être utilisé, pour indiquer par quel mécanisme le message est envoyé tel que Arpanet ou Phonenet, et le paramètre "with" peut être utilisé pour indiquer quel niveau de protocole courrier ou connexion doit être utilisé, tel que le protocole de courrier SMTP ou le protocole de transport X.25.

Remarque: Plusieurs paramètres "with" peuvent être inclus pour désigner exactement le jeu de protocole utilisé.

Certains services transports mettent en file d'attente les courriers ; l'identificateur de message interne qui est assigné au message peut être noté en utilisant le paramètre "id". Quand le serveur hôte d'expédition utilise la spécification d'une adresse de destination que l'hôte receveur réinterprète, à travers l'extension ou la transformation, le serveur receveur pourrait enregistrer la spécification originelle, en utilisant le paramètre "for". Par exemple, quand une copie de courrier est envoyée à un membre d'une liste de télédistribution, ce paramètre peut être utilisé pour enregistrer les adresses originelles qui sont utilisé pour définir la liste.

4.2.3 Le champs « *From* »

Ce champ contient l'identité de la ou des personnes qui ont voulu que ce message soit envoyé. Le processus de création de message devrait mettre par défaut une seule adresse de machine authentifiée, indiquant l'**agent** (personne, système ou processus) qui a voulu envoyer le message. Si cela n'est pas fait, le champ expéditeur (sender) doit être présent. Si le champ "from" est mis par défaut, le champ "expéditeur" ("sender") est optionnel et est redondant avec le champ "From". Ainsi le champ «sender » a pour but d'être utilisé quand l'expéditeur n'est pas l'auteur du message ou pour indiquer parmi un groupe d'auteur expéditeur qui a réellement expédié le message.
Dans tous les cas, les adresses dans le champ "From" doivent être utilisables par la machine et ne doivent pas contenir des listes nommées (groupes).

4.2.4 Le champs « *replay to* »

Ce champ fournit un mécanisme global pour indiquer n'importe quelle boite aux lettres vers laquelle la réponse doit être envoyée. Trois utilisations classiques pour cette caractéristique peuvent être distinguées. Dans le premier cas, le(s) auteur(s) peuvent ne pas avoir de boite à lettre régulièrement basée sur un ordinateur, et qui veulent indiquer une adresse de machine alternative. Dans le second cas, un auteur veut que des personnes supplémentaires soient averties, ou soient responsables. Une utilisation quelque peu différente peut en être faite pour aider les groupes de téléconférence en mode texte équipés avec des services de télédistribution automatique : incluant l'adresse de ce service dans le champ "Reply-To" pour tous les messages soumis à la téléconférence ; Alors les participants peuvent "répondre" aux soumissions faites pendant la conférence pour garantir la distribution correcte de n'importe quelle soumission de leur part.

Remarque: Le champ "Return-Path" est ajouté par le service de transport de courrier, au moment de la distribution finale. Cela à pour but d'identifier le chemin de retour vers l'originaire du message. Le champ "Reply-To" est ajouté par l'originaire du message et s'attend à des réponses directes.

4.2.5 Le champ « *BCC (blind carbon copy)* »

Alors qu'en français cci (*copie conforme invisible*) ce champ contient l'identité des destinataires supplémentaires du message de telle sorte que si vous ajoutez le nom d'un destinataire au champ **Cci** dans un message électronique, une copie du message est envoyée au destinataire, mais le nom de celui-ci n'est pas visible pour les autres destinataires du message.

4.2.6 Identificateur de message « *Message-ID* »

Ce champ contient un identificateur unique (l'élément d'adresse de la partie locale) qui référence la version de ce message. L'unicité de l'identificateur du message est garantie par le système d'hôte qui le génère. Cet identificateur à pour but d'être lisible à la machine et non d'être nécessairement compréhensible pour les humains.
A un identificateur de message correspond uniquement une seule instance d'un message précis ; Les révisions successives du message devront chacune recevoir de nouveaux identificateurs de message.

5. Le cryptage

Parfois, la cryptographie est utilisée pour augmenter la confidentialité du contenu du message. Si le corps du message a été crypté, pour garder son contenu privé, le champ "Encrypted" peut être utilisé pour noter le fait que le message soit crypté, et pour indiquer la nature du cryptage. Le premier paramètre de type <word> indique le logiciel utilisé pour crypter le corps, et le second, optionnel de type <word> a pour intention d'aider le destinataire dans la sélection de la bonne clé de décryptage.

Remarque: Malheureusement, les en-têtes doivent contenir aussi bien l'enveloppe, que le contenu et l'information. Par conséquent, il est nécessaire qu'elles restent non cryptées, afin que le service de transport de courrier puisse y accéder. Puisque le contenu des champs de noms, d'adresses, et "Sujet" peut contenir de l'information sensible, cela limite la confidentialité totale du message.

6.6.1 Syntaxe — Spécifications de l'adresse

Address	Mailbox Group	Une adresse. Liste nommée.
Group	Phrase ":" [#mailbox] ";"	
mailbox	addresse-spec phrase route-addresse	Adresse simple. Nom et spécification d'adresse.
route-addr	"<" [route] addr-spec ">"	

route	1#("@" domain) ":"	Chemin relatif (path-relative).
addr-spec	local-part "@" domain	Adresse globale.
local-part	word *("." word)	Non interprétée. préservation de la case
sub-domain	domain-ref / domain-literal	
domain	sub-domain *("." sub-domain)	
sub-domain	domain-ref / domain-literal	
domain-ref	atom	Référence symbolique.

Table VII.3 spécification syntaxique de l'adresse.

6.2 Sémantique

Une boite aux lettres reçoit des courriers. C'est une entité conceptuelle qui n'est pas forcement en rapport avec le stockage de fichier. Par exemple, quelques sites peuvent choisir d'imprimer le courrier sur une imprimante en ligne et remettre la sortie au bureau du destinataire.
Une spécification de boite aux lettres comprend une référence à un nom de personne, de système ou de processus, une chaîne dépendant d'un domaine, et une référence à un nom de domaine. La référence au nom est optionnelle et on l'utilise généralement pour indiquer le nom humain de la personne destinataire. La référence au nom de domaine désigne une série de sous-domaines. La chaîne dépendant du domaine n'est pas interprétée, exception faite du sous domaine final ; le reste du service de courrier, la transmet simplement comme une chaîne littérale.

6.3 Les domaines

Un nom de domaine est un ensemble de noms répertoriés. Une définition de nom de domaine décide de la définition de nom de domaine subordonné ou d'une chaîne terminale dépendante d'un domaine. Par conséquent, la définition d'un domaine est extensible, permettant n'importe quels nombres de niveaux d'enregistrement.

Les noms de domaine modélisent le schéma hiérarchique, logique, et global d'adressage. Le modèle est logique, du fait que la définition d'adressage est relative au nom d'enregistrement et n'est pas obligatoirement liée au chemin de transmission. La hiérarchie du modèle est un graphe orienté, appelé arbre entrant (*in-tree*), de telle manière qu'il n'y ait qu'un seul chemin de la racine de cet arbre vers n'importe élément de la hiérarchie. Si plus d'un chemin existe réellement, on considère que l'on a affaire à des adresses différentes.
Le nœud racine est commun à toutes les adresses ; Par conséquent, il n'est pas référencé. Ces enfants constituent les noms de domaine de haut niveau. Habituellement, un service a accès à sa définition complète de nom de domaine et à tous les domaines de nom de niveau supérieur.
Le "sommet" de la hiérarchie d'adressage de domaine - un des fils de la racine - est indiqué par le champ situé le plus à gauche dans la définition d'un domaine. Ses enfants sont situés à gauche, leur fils à gauche, et ainsi de suite.
Certains groupes fournissent des services officiels d'enregistrements ; constituant des domaines de noms qui sont indépendant logiquement de machines spécifiques. En plus, les machines et les réseaux composent implicitement des domaines de noms, puisque leurs membres sont habituellement enregistrés dans des tables de noms. Dans le cas d'un

enregistrement officiel, une organisation implémente une base de données (distribuée) qui fournit un service de correspondance adresse-route pour les adresses qui sont de la forme :

person@registry.organization

Notez que "organization" est une entité logique, indépendante de tout réseau de communication.
Un mécanisme pour accéder à l'organization est disponible pour tout le monde. Ce mécanisme, en boucle, cherche une instance de l'enregistrement ; sa localisation n'est pas spécifiée sous forme d'adresse. On suppose que le système qui opère sous le nom "Organization" sait comment trouver un registre subordonné. Le registre utilisera alors la chaîne "person" pour déterminer ou envoyer la spécification du courrier. Ce dernier permet une spécification simple et directe d'adresse relative à un attachement, comme:
user@host.network

Une fois que le réseau est atteint, on s'attend à ce que le message ira directement vers l'hôte et que l'hôte résoudra le nom de l'utilisateur, le message est placé dans la boite aux lettres de l'utilisateur.

6.4 Spécifications d'abréviation de domaine

Puisqu'un nombre quelconque de niveaux est possible dans la hiérarchie d'un domaine, la spécification d'une adresse pleinement qualifiée, peut devenir un inconvénient. Ce standard permet des définitions de domaine mises sous forme abrégées pour le cas particulier :

Pour les adresses de l'expéditeur, on appelle Niveau N le sous domaine le plus à gauche. Dans une adresse d'en-tête, si tous les sous domaines situés au-dessus (c.à.d. à droite) de Niveau N sont les même que celle de l'expéditeur, alors ils n'ont pas à apparaître dans la spécification. Sinon, l'adresse doit être pleinement qualifiée (*fully qualified*).
Cette caractéristique est sujet d'approbation par les sous domaines locaux. Les sous domaines individuellement peuvent avoir besoin de leur système membre, qui donne l'origine du courrier, pour uniquement fournir une définition de domaine complète.
Quand c'est autorisé, les abréviations peuvent être présentes seulement quand le message reste à l'intérieur du sous domaine.
L'utilisation de ce mécanisme nécessite que le sous-domaine de l'expéditeur réserve les noms de tous les domaines de niveau supérieur, de cette façon ses définitions complètes peuvent être différenciées des définitions mise sous forme d'abréviations.

Par exemple, si l'adresse d'un expéditeur est :
expediteur@registry-A.registry-1.organization-X
et l'adresse d'un destinataire est :
destinataire@registry-B.registry-1.organization-X
et qu'un autre soit :
destinataire@registry-C.registry-2.organization-X

Alors ".registry-1.organization-X" n'a pas besoin d'être indiqué dans le message, mais "registry-C.registry-2" **doit** être indiqué. C'est à dire, les deux premières adresses peuvent être mises sous forme abrégée mais la troisième doit être entièrement définie.

Quand un message traverse une limite de domaine, toutes les adresses doivent être déclarées sous le format complet, en terminant par le nom de domaine de niveau supérieur sur le champ le plus à droite. C'est la responsabilité du service d'expédition de courrier de s'assurer que les adresses soient en conformité avec cette exigence. Dans le cas des adresses mises sous abréviation le serveur qui fait relais doit faire l'extension nécessaire. On doit noter que c'est souvent difficile pour de tel service de localiser toutes les occurrences des abréviations d'adresse. Par exemple, il sera impossible de trouver ces abréviations à l'intérieur du corps d'un message. Le champ de " chemin de retour " peut aider les destinataires dans la correction de ce type d'erreur.

Remarque: Quand on passe n'importe quelle portion d'une définition d'adresse à un processus qui n'interprète pas les données en accord avec ce standard (par exemple les serveurs de protocole courrier). Il ne doit pas y avoir de caractère LWSP précédant ou suivant le signe « at @» ou aucun point de délimitation ("."), tel que ce qui a été montré dans les exemples au-dessus, et seulement un espace entre les mots <word>: contiguës.

6.5 Les termes du domaine

Une référence de domaine doit être le nom officiel d'un registre, d'un réseau ou un hôte. C'est une référence symbolique, à l'intérieur d'un nom de sous domaine. A tout moment, il est nécessaire de passer outre les mécanismes du standard pour résoudre de telles références, en utilisant des informations plus primaires, tel que les adresses d'hôte de réseau plutôt que leur nom d'hôte associé.

Remarque: l'utilisation de littéral de domaine est **fortement découragée**. Cela est permit seulement comme un moyen de contourner temporairement les limitations système, tel que les tables de nom qui ne sont pas complètes.

La partie locale d'une spécification d'adresse dans une déclaration d'adresse (c.à.d. le nom d'hôte pour la boite aux lettres) est interprétée comme étant ce que le serveur receveur d'email autorise. Par exemple, certains systèmes ne comprennent pas les déclarations de boite aux lettres de la forme "P. D. Q. Bach", mais d'autres le font.
Cette norme traite les points (".") en tant que séparateurs lexicaux. En conséquence, leur présence dans les parties locales qui ne sont pas entre les chaînes mise entre guillemets, est détectée. Cependant de telles occurrences n'apportent aucune sémantique. Cela signifie, que si une partie locale avait des points dans cette partie, un analyseur d'adresse la diviserait en plusieurs marques syntaxiques, mais la séquence de ces marques serait traitée comme une seule unité ininterrompue. La séquence devra être réassemblée, quand l'adresse sera passée hors du système comme celui du service de protocole de mail.

Dans certains cas, la frontière entre la partie locale et le domaine peut être flexible. La partie locale peut être une simple chaîne de caractère, qui est utilisé pour déterminer la boite aux lettres du destinataire final. Tous les autres niveaux font par conséquent, partie du domaine.

6.6 Boite aux lettres multiples

Un individu peut avoir plusieurs boites aux lettres et aimerait recevoir du courrier dans la boite aux lettres qui serait la plus pratique à accéder pour l'expéditeur. Et par conséquent un

ensemble d'individu peut vouloir recevoir un courrier comme une seule unité (c.à.d. une liste de distribution). Le constructeur de <group> permet la spécification à la manière d'une liste. Les boites aux lettres des destinataires sont spécifiées à l'intérieur de la partie entourée de parenthèse (bracket). Une copie du message est envoyée à chacune des boites aux lettres listées. Le standard ne permet pas des spécifications récursives de groupe à l'intérieur de groupes.

Certains services de courrier peuvent fournir un mécanisme de distribution par groupe-liste, acceptant une simple référence de boite aux lettres, en développant une liste de distribution complète, et en relayant le courrier aux membres de la liste. L'utilisation d'adresse alternative <group>, qui contient à l'intérieur, une boite aux lettres, peut signifier que, soit la référence de la boite aux lettres sera développée en une liste, soit qu'il y a un groupe avec un seul membre.

6.7 Spécification explicite du chemin

Quelquefois, une personne à l'origine d'un message doit pouvoir indiquer quel chemin de transmission, le message doit suivre. On appelle cela le **routage à la source**. Le schéma normal d'adressage, utilisé dans la spécification d'adresse, est dissocie par prudence de ces informations. On spécifie la séquence de serveurs hôte et/ou les services de transmission qui sont traversés. A la fois, les références de domaines et les littéraux de domaine peuvent être utilisées.

Remarque: L'utilisation du routage à la source est découragée. A moins que l'expéditeur a un besoin spécifique pour des restrictions de route, le choix de la route de transmission devra être laissé au service de transport de courrier.

6.8 Adresses réservées

Il est souvent nécessaire d'envoyer un courrier vers un site, sans connaître aucunes adresses valides. Par exemple, il y aura peut être des dysfonctionnements du système de courrier quand un utilisateur veut connaître l'adresse correcte d'une personne sur ce site.

Des standards spécifient une simple adresse, adresse réservée de boite aux lettres (*partie locale*) qui doit être valide pour chaque site. Le courrier envoyé à cette adresse doit être routé vers une personne responsable pour le système du courrier. Le nom de la partie locale de l'adresse est : **Postmaster**, ainsi on doit avoir "Postmaster@domain" pour que ce soit valide.

7. Conclusion

La simplicité apparente de la manipulation du message est due à l'établissement de différents formats pour mieux adapter le plus efficace, cette approche a tendance de respecter les standards établis par les plus grandes organisations pour minimiser au maximum les incohérences de communications.

CHAPITRE VIII.

Conception de B2SAM.

En arrivant là; là se trouve l'embarras.
– William Shakespeare.

1. Introduction

1.1 UML

Dans ce chapitre, nous utilisons UML (*Unified Modeling Language)* pour décrire la conception de notre système HoneyPot et pour cela nous avons utilisé le modèle en V.

Le modèle en V demeure actuellement le cycle de vie le plus connu et certainement le plus utilisé. Le principe de ce modèle est qu'avec toute décomposition doit être décrite la recomposition, et que toute description d'un composant est accompagnée de tests qui permettront de s'assurer qu'il correspond à sa description. Ceci rend explicite la préparation des dernières phases (validation-vérification) par les premières (Construction du logiciel), et permet ainsi d'éviter un écueil bien connu de la spécification du logiciel :

« Énoncer une propriété qu'il est impossible de vérifier objectivement après la réalisation ». **[UML-31]**

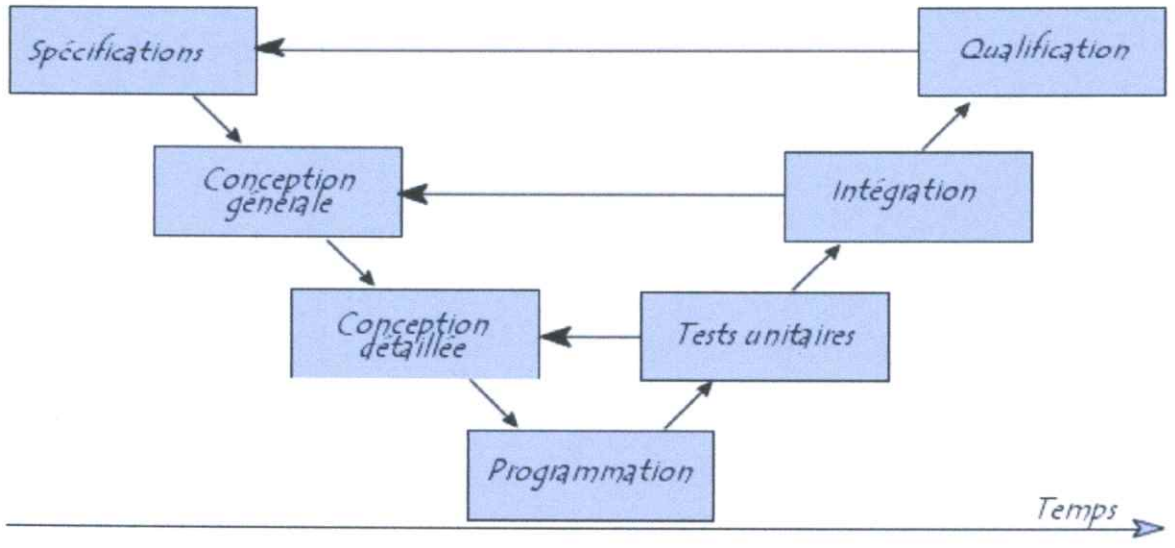


Fig. VIII.1 Model de cycle de vie d'un logiciel.

La notation UML que nous avons utilisée est née de la fusion des trois méthodes qui ont les plus influencées de la modélisation objet au milieu des années 90 : OMT, Booch et OOSE. De ce fait, elle permet de couvrir le cycle de vie d'un logiciel depuis l'analyse du besoin jusqu'au codage. Cette notation est d'une très grande richesse. Elle permet de couvrir à peu près toutes les phases du développement.

Parmi les avantages d'UML:

- Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions.
- L'aspect formel et normalisé de sa notation, limite les ambiguïtés et les incompréhensions.
- Son indépendance par rapport aux langages de programmation, aux domaines d'application et aux processus, en font un langage universel.
  [UMF-32]

1.2 UP (*Unified process*)

Le processus unifié a été élaboré par Jacobson il a particulièrement été bien adapté pour la mise en œuvre d'un processus de développement basé sur la notation UML.
Une réalisation conforme à UP, pour transformer les besoins des utilisateurs en logiciel, doit nécessairement présenter les caractéristiques suivantes :

- UP est à base de composants.
- UP utilise UML (*ensemble d'outils et de diagramme*).
- UP est piloté par les cas d'utilisation.
- UP est centré sur l'architecture.
- UP est itératif et incrémental.

1.2.1 Phase de Réalisation d'UP

- **Phase de Création**

Traduit une idée en vision de produit fini et présente une étude de rentabilité pour ce produit.
Elle répond aux questions suivantes :

- que va faire le système ?
- quelle va être l'architecture générale (cible) de ce système.
- quels vont être : les délais, les coûts, les ressources, les moyens à déployer?

- **Phase d'élaboration**

Permet de préciser la plupart des cas d'utilisation et de concevoir l'architecture du Système.
Les taches à effectuer dans la phase élaboration sont les suivantes :

- Créer une architecture de référence.
- Identifier les risques, ceux qui sont de nature à bouleverser le plan, le coût et le calendrier
- Définir les niveaux de qualité à atteindre
- Élaborer une offre abordant les questions de calendrier, de personnel et de budget

- **Phase de construction**

L'architecture de référence se métamorphose en produit complet, elle est maintenue Stable.

- **Phase de transition**

Un groupe d'utilisateurs essaye le produit et détecte les anomalies et défauts.

2. Présentation de l'application

Apres une étude approfondie sur les Spams et les technique de spamming ainsi que les HoneyPot et même les différentes techniques d'attaques sur les réseaux et systèmes d'exploitations on a essayé de mettre au point un système anti-spamming utilisant des ressources distantes, ainsi que différentes techniques de détection de spam pour préserver au mieux l'aspect objectif du mailing; et pour cela on a proposé l'architecture suivante (*Fig VIII.2*) toute en utilisant des logiciels open source définit au préalable.

Ainsi l'architecture de notre système est composée d'éléments suivants :

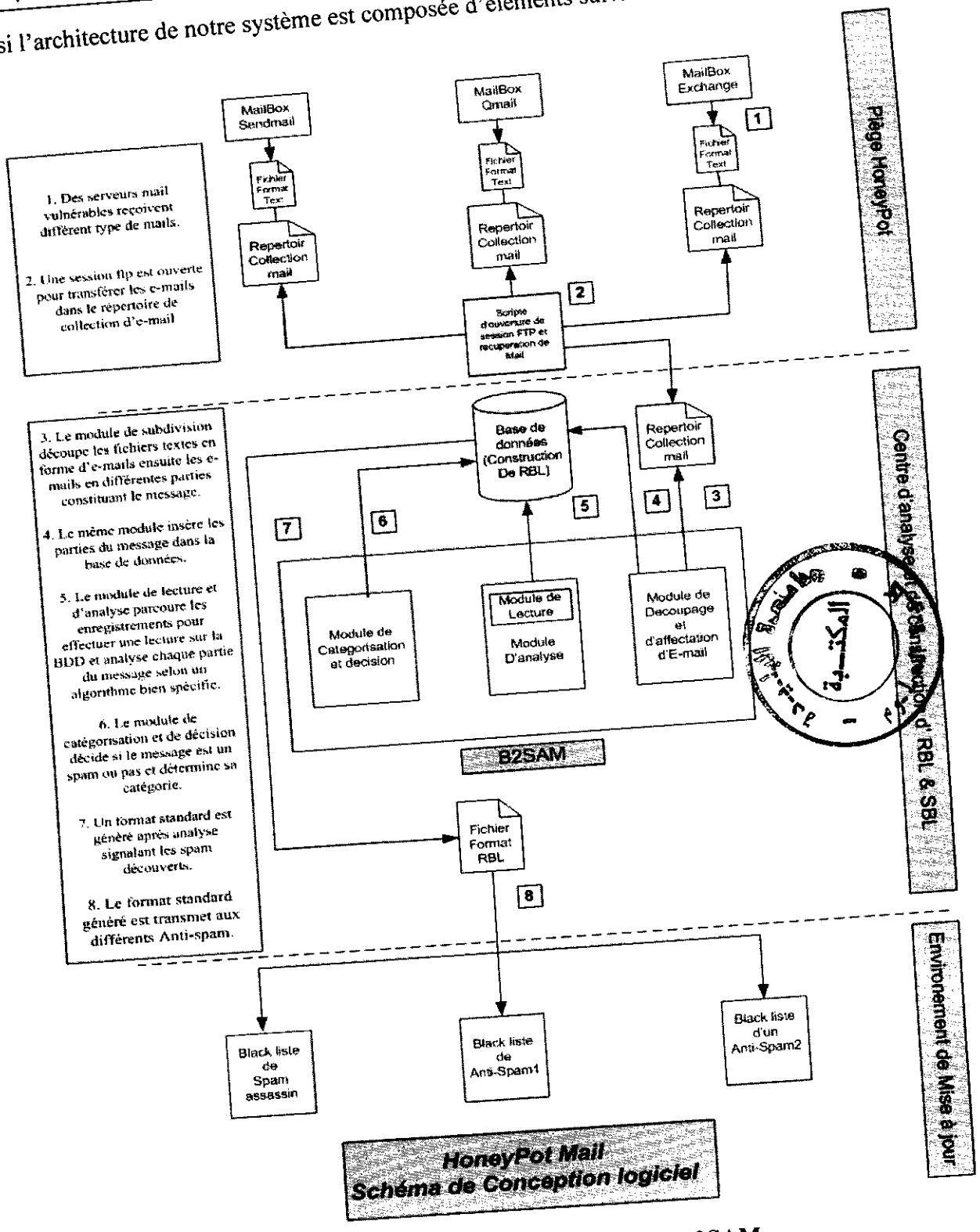


Fig. VIII.2 Architecture du système B2SAM.

2.1 La 1ère partie : « piège HoneyPot »

- **Serveurs mails virtuels volontairement vulnérables**

Dans le but de collecter le plus maximum possible d'e-mails et d'attirer les spammeurs qui cherche à bousier les serveurs mails public ou bien utiliser le mailing à des fins négatives on a utilisé plusieurs serveurs mail volontairement vulnérables et à une grande tendance d'importance, aussi on a utilisé plusieurs systèmes d'exploitation avec divers Services mails tel que LINUX RED HAT avec « Sendmail & Qmail »et WINDOWS SERVEUR 2003 avec « Exchange ».
Pour notre architecture proposée on a adopté « SendMail » sous linux Redhat Entreprise Edition(V4) pour les raison bien expliqués dans le chapitre « outils ».
Et ainsi pour définir le point de vulnérabilité on a utilisé différentes méthodes pour que nos domaines soient volontairement connus par les internautes et plus précisément les spammeurs en s'enregistrant aux forums de hacking de publicité de commerce...Etc.

- **Module de récupération d'e-mails**

Ce module fait partie de notre système d'analyse « B2SAM » qui sera détaillé ultérieurement, en utilisant un script d'ouverture de session ftp qui est programmé comme étant une tache planifiée qui s'exécute chaque semaine, il essaye de se connecter à notre système d'analyse qui contient déjà un server ftp installé avec le pack Microsoft appelé (Internet Information Services-File Transfer Protocol Service), puis après avoir ouvert la session ftp il récupère les E-mail à partir des boites aux lettres de différents serveurs mails qu'on a mis au point sous format texte, et les envois directement au répertoire qui se trouve sur le B2SAM central, cela rend les e-mails locaux pour le module d'analyse de B2SAM.

Note : Les e-mails récupérés sont sous format texte, mais il est à noter que **tous** les e-mails reçus pour un utilisateur donné sont regroupés dans un seul fichier texte pour éviter les conflits de stockage et de transfère ainsi que de consultation et de manipulation; cette technique est utilisée par défaut dans tous les systèmes d'exploitation basés sur UNIX et elle est ainsi préserver par « SendMail ».

Le fait d'automatiser cette procédure de collecte de mail, l'ouverture de session ftp et de transfère de données rend notre système plus autonome (car l'analyse est séparée de la collecte d'e-mails) d'une part, d'autre part si on décide un jour de mettre un nouveau serveur de collecte ou d'ajouter un nouveau centre de collecte on pas besoin de configuration manuel, il suffit juste d'exécuter le script une seul fois -même si on n'est pas des spécialistes- et le script fonctionnera sans aucune intervention de l'utilisateur. Ce procéder définit un point d'extensibilité très important tant au niveau de collecte d'informations qu'au niveau de compatibilité.

2.2 La 2ème partie : « B2SAM CENTRAL »

C'est le noyau de notre système ; et il a été conçu pour répondre au grand besoin de sécurisé les serveurs mail et bien respecter la notion du mailing dans un sens constructif.

Il est constitué des modules suivants :

1. Module de lecture du fichier texte, de décomposition et d'écriture sur la base de données

Apres la collecte d'e-mails et le transfert de ces derniers par une session ftp, ils deviennent locaux à notre système d'analyse, et ils se situeront exactement dans le répertoire dédié pour le stockage appelé « Rep_Centr » ; c'est la qu'intervient ce module qui doit comme première étape charger les fichiers textes ainsi transférés à partir du répertoire de stockage pour les rendre prét pour l'analyse, ensuite dans la deuxième phase il doit les décomposer localement en différents mails séparés, puis il procède a la décomposition dans la troisième étape de ces mails en différentes parties constituant le mail et les insère au fur et à mesure dans les différentes champs de la table d'analyse temporaire de la base de données, cette table qui est appelée « Mail_decomp » sera utilisée lors de l'analyse par les différents modules d'analyse qui utilisent des champs déjà existant de cette table pour remplir les autres champs de l'analyse.

Ces champs obtenus sont les composants principaux du message à qui sont appliquer les critères d'analyse et de catégorisation, et qui ont été bien détaillés dans le chapitre traitant le format du message.

Les parties des messages qui vont êtres utilisées pour l'analyse et la catégorisation sont les suivants :

1. L'adresse IP de l'expéditeur.
2. Le domaine à qui appartient l'expéditeur.
3. Le sujet du message.
4. La date de l'envoi du message.
5. Le corps du message.

Apres l'extraction de ces parties le module écrit les informations estradiers un par un dans la base de données temporaire afin de les analyser avec le module d'analyse.
On a utilisé seulement ces parties parmi tout le jeu de composants que contient le message parce que sont les parties les plus significatives dans le message et à partir de ces composants que la plus grande partie des formats standards de mise à jour pour les anti-spams sont générés, car ils présentent l'identité du message, l'identité de l'expéditeur et même le domaines à qui appartient l'expéditeur.

2. Module d'analyse

Afin de détecter qu'un e-mail est un spam ou pas, ce module utilise plusieurs méthodes d'analyse combinées décrites ci-dessous pour pouvoir porter un jugement correcte sur l'e-mail en question.

2.1 Analyse de l'adresse IP de l'expéditeur

Le module d'analyse fait un test comparatif de l'adresse IP de l'expéditeur de chaque message contre une liste d'adresses IP définit au préalable et stockée dans la base de donnée locale de l'analyse plus précisément dans la table « IP_suspect » qui contient plus de 11000 adresses IP signalées comme suspectes par les fournisseurs d'internet les plus connus au monde. Cette liste d'adresse est tirée de la dernière version (Mai 2007) des adresses IP suspectes reconnues mondialement et utilisées par le projet Spamhaus sur le site (*www.spamhaus.prg*), et sera mise à jour automatiquement après la détection de nouveau spam par d'autre modules tel que analyse sur le sujet, les mots suspects etc...., et pour préserver l'efficacité et l'informité par rapport aux standards universels on a utilisé le format SBL utilisées par les grands centres d'observation et de détection de spam à travers le monde (des statistiques furent établît par le projet « Spamhaus » reportent que plus de 85% des format standards de mise à jour sont sous format SBL, 15% sous format RBL et le reste sous plusieurs formats définit par d'autres organismes). **[ANNEXE Spamhaus]**.

Ce test comparatif à la base de fonctionnement de ce module signale si l'adresse du message fait partie de la liste des adresses IP suspectes ou pas pour attribuer un score précis au message en cours.

2.2 Analyse du domaine de l'expéditeur

En utilisant le même principe de l'analyse de l'adresse IP mais cette fois ci en utilisant une table prés définit appelée « Domaine_susp » qui contient les adresse IP des domaines suspects et les domaines correspondants en jouant le rôle d'un serveur DNS, cette liste qui a été extraite du projet Spamhaus, est considérée comme la version la plus ressente (on à utilisé une liste de plus de 4000 domaines-IP suspects), cette dernière sera met à jour automatiquement après la détection de spam par d'autre module d'analyse tout comme le premier module d'analyse d'adresse IP, le module d'analyse utilise cette liste pour la détection des domaines blacklistés en les comparant avec listes qui contiennent bien évidemment les domaines considérés comme étant potentiellement source de spam ou tout simplement blacklistés.

A l'essor de ce module un score est généré selon l'appartenance ou non du message à un domaine suspect.

2.3 Analyse du sujet

Dans cette étape le module d'analyse vérifie le sujet du message s'il respecte les spécifications standards dictées par les organismes de définition du format standard du mail et ces différentes parties notamment les RFC, et en particulier la RFC822; donc on adopté les spécifications suivantes :

- Le sujet d'un message ne doit pas être vide.
- Ne doit pas commencer par deux chiffres.
- Ne doit pas contenir que des chiffres.
- Le sujet ne doit pas commencer par au moins deux tirets (underscore).
- Le sujet ne doit pas commencer par au moins trois points.
- Ne doit pas contenir un mot suspect ou une phrase suspecte.

2.4 Analyse de la date d'envoie du message

Le module d'analyse parcourt séquentiellement tous les enregistrements puis, s'il trouve des messages envoyés par le même émetteur il calcule le laps de temps séparant les dates d'envoi des messages ; bien entendus cette analyse est basée sur un seuil c.-à-d. si la différence entre les deux date est inferieur ou égale au seuil, l'envoi sera considéré comme une attaque et l'émetteur est hautement qualifie d'être un spammeur.

Exemple : 1- expéditeur : David date : 06/15/2007 04:30 :22 PM.
2- expéditeur : David date : 06/15/2007 04:31 :11 PM.
3- expéditeur : David date : 06/15/2007 04:32 :10 PM.
4- expéditeur : David date : 06/15/2007 04:32 :50 PM.

Si par exemple le seuil est fixer à 3 minute, et le nombre de message est fixer à 4, donc après une comparaison sur l'adresse et sur l'expéditeur ainsi que sur la date et le temps on considère David comme spammeur ; et son adresse est ajoutée à la liste d'adresses suspectes sur la base de données, son domaine est ajouté à la liste des domaines suspects sur la base de données, et même le sujet de son message est ajouté à la liste des mots suspects sur la base de données. Mais ces mises à jour ne sont faites bien sure que si le score obtenu par cet e-mail atteint le score qualifiant l'e-mail comme étant un éventuel spam.

2.5 Analyse du corps

C'est la partie la plus complexe et la plus délicate car malgré toutes les techniques d'analyse même les plus avancées utilisant les algorithmes mathématiques les plus compliqués comme celle des filtres Bayaisien on arrive pas à des ajustements rigoureux de détection de spam, car il ne s'agit pas seulement d'un jeu de mots déterminé auparavant pour en arriver à classer un tel message, ce qui explique bien la délicatesse de cette tache, par exemple une société pharmaceutique, lorsqu'elle manipule le courrier électronique doit obligatoirement utiliser des mots qui sont bien considérés par la plupart des anti-spam comme des mots suspects comme par exemple les noms des produits (viagra, gélule,..Etc.) Et par la suite la plus grande partie des e-mails de cette société seront considérés comme des Spams. **[AOS-33]**

Cela dit, la méthode qu'on a adoptée pour ce module est la suivante :

Le module d'analyse parcourt tous l'e-mail (le corps de l'e-mail) un par un pour extraire les mots suspicieux en utilisant une liste de mots suspects déjà existante cette liste est construite en utilisant plusieurs méthodes de collecte de mots suspects :

- la collecte des mots suspects s'est faite en utilisant des listes définies par la plupart des anti-Spam en essayant de garder une vue logique et cohérente par rapport à toutes les listes qu'on a utilisé.
- Une analyse manuel de plusieurs Spam reçus sur des boites aux lettres utilisées spécialement pour collecté des spam et pour voir les différents formats et types (adresse1@hotmail.com, adresse2@usthb.dz.com, adresse3@gmail.com...etc. plus de 2000 spam reçu).
- Après une étude prolongée sur les techniques utilisées par les spammeurs pour échapper aux différents filtres anti-spams, on a remplacé quelques lettres par des caractères que les spammeurs utilisent par exemple : « a par @ », « i par | », « g par 6 », « s par 5 », « l'ajout des espace entre les lettre dans quelques mots ».
- Faire toutes les combinaisons possibles de ces remplacements pour avoir une liste exhaustive de tout ce qui pourra être utilisé pour dévier aux filtres anti-spams.

Cette analyse du corps basée sur les mots suspects est la première étape, dans la seconde phase on a utilisé une technique presque jamais utilisée jusqu'à présent cette technique consiste à l'analyse basée sur des **phrases** ; c.-à-d. que le module d'analyse du corps parcourt tout le contenu de chaque message en recherchant l'occurrence d'une phrase suspecte définie au préalable dans une table appelée « Phrase_susp », l'analyse basée sur des phrases est plus significative et plus efficace car c'est une comparaison basée sur le contexte et sur une certaine sémantique et non seulement sur de simples mots.

Le seul souci qui reste c'est la construction d'une table des phrases suspectes, ce qui n'est pas toujours évident, car de simples phrases utilisées dans un contexte peuvent ne pas avoir le même

sens dans un autre contexte ce qui rend la tache plus délicate et plus sensible aux variations du contexte. C'est pour cette raison qu'on a utilisé les deux méthodes (analyse par mot et par phrase) en même temps et combinées entre eux de telle sorte quelles soient les plus efficaces possible, et toute en jouant sur l'attribution des scores pour mieux garder la cohérence entre les deux méthodes car les phrases suspectes peuvent contenir des mots suspects qui vont êtres détectés seulement par l'analyse basée sur les mots et par la suite l'analyse basée sur les phrases se révèle inutile tandis que l'analyse basée sur les phrases tient compte de la **combinaison** des mots et donc du contexte ce qui rend l'analyse basée sur seulement des mots peut promotrice. Ce qui explique bien la nécessité de combiner les deux méthodes soigneusement pour en profiter des avantages de chacune d'entre elles.

Note

- toutes les tables qu'on a utilisé (table des adresses IP suspect; table des domaines suspects et ainsi que les deux tables des mots et des phrases suspectes) sont susceptibles de variations et de mise à jour soit manuellement par l'administrateur utilisant notre solution ou bien automatiquement par de nouvelles tables récupérées auprès des organisations professionnelles et des sites de recherche travaillant sur des thèmes similaires. Ceci est un point d'**extensibilité** très important car la séparation entre les modules d'analyse et les tables utilisées dans l'analyse permet d'utiliser et d'ajouter de nouvelles tables (table des adresses IP suspect ou bien table des domaines suspects...) sans se soucier du fonctionnement interne des modules ce qui rend une telle interaction plus flexible.

- Lors de la collecte de mots suspects et les phrases suspectes on a tenu compte du rapport publie par la CNIL (Commission Nationale de l'Informatique et des Libertés) :

Typologie des spams en fonction de la langue utilisée	
Spams de langue anglaise	84,8 %
Spams de langues asiatiques	8 %
Spams de langue française	7 %

Table VIII.1 pourcentage des Spams en fonction de la langue utilisée.

Ainsi on s'appuyant sur ces pourcentages notre collecte de mots suspects ainsi que les phrases suspectes est basée sur la langue anglaise beaucoup plus que d'autre langue, et on a ainsi utilisé la langue française et l'allemand pour avoir le pourcentage le plus élevé des mots et phrases et rendre l'analyse plus rigoureuse.

3. Attribution du score

Un score est donné après chaque phase d'analyse et plus exactement par chaque module d'analyse portant sur une partie bien définie, il est ajouté au score de la phase précédente pour

avoir le score final calculé par l'addition de différents scores issus de chaque module d'analyse du message.

En réglant le score donné par chaque phase d'analyse en peut assez bien combler les lacunes que chaque phase peut générer, pour y remédier on a utilisé des statistiques proposées par le projet APACHE qui nous donne les rapports les plus appropries du scorage de chaque module, voici les formules qu'on a utilisé pour déterminer les score avec exactitude :

- L'adresse IP représente 11.1% de Spams reçu.
- Le nom du domaine représente 22.19% de Spams reçu.
- Le sujet ne représente que 6.66%.
- La date représente 15.55%.
- Les mots suspects représentent 11.01%.
- Les phrases suspectes représentent 33.33%.

Note : L'administrateur de notre solution peut changer la configuration par défaut et régler les scores comme bon lui semble selon les besoins ou bien l'orientation du domaine d'études désiré.
En basant sur les statistiques données ci-dessus on a établir la stratégie de scorage suivante qui est la stratégie par défaut :

- Module d'analyse de l'adresse IP : 5 points.
- Module d'analyse du domaine : 10 points.
- Module d'analyse du sujet : 3 points
- Module d'analyse de la date : 7 points
- Module d'analyse du corps basée sur les mots : 5 points.
- Module d'analyse du corps basée sur les phrases : 15 points.

Apres avoir calculé les pourcentages appropries attribuer à chaque module et les points que chaque module peut donner on a décidé de fixer le seuil de signalisation de spam à **10** points (d'être dans le milieu des scores (max+min)/2).
Et par conséquent on a critères suivantes:

- Un message ayant un score inferieur à 10 est considéré comme non spam.
- Un message ayant un score supérieur ou égal à 10 est considéré comme spam.

Un message qui provient d'un domaine suspect ou bien contient une phrase suspecte doit être considéré comme étant un spam, et bien effectivement il est car le score est supérieur ou égale à 10. Tandis qu'un message qui contient un mot suspect ou bien le sujet est suspect ne doit pas être qualifier directement spam, il faut vérifier les autres champs pour décider; d'où provient la délicatesse d'attribution du score.

4. Module de catégorisation

Comme son nom l'indique il répertorier les Spams détectés en différentes catégories en remplissant la case « catégorie de spam » dans la base de données temporaire. Cette catégorisation nous permet d'avoir des idées claires sur les comportements des spammeurs ainsi que les différents domaines qu'ils utilisent pour piéger les internautes, elle est donnée sous forme d'un commentaire, par exemple : publicité, loterie, santé, voyance ...etc.

La catégorisation permet de détecter les nouveaux centres d'intérêts des spammeurs pour mieux les anticiper et mieux régler la configuration des anti-Spams pour qu'ils puissent êtres à jour en terme de nouvelles techniques de spamming ; et on peut aussi lui en servir aussi comme un argument de statistique.

5. Base de données temporaire

Cette base de données stocke les différentes parties du message après décomposition, ainsi que le score final et la catégorie du message; donc elle doit contenir toutes les informations nécessaires pour l'analyse du message aussi bien que les informations pour les formats standards générés(les outputs) après analyse, elle contient les colonnes suivantes :

- Le numéro séquentiel d'entrée dans la table.
- L'adresse IP expéditeur.
- L'adresse du domaine de l'expéditeur.
- Le sujet du message.
- Le corps du message.
- La date du message.
- La catégorie du message.
- Le score du message.
- Type du message (SPAM, Non spam).

6. Module de mise-à-jour

Ce module en récupérant les informations nécessaires à partir de la base de données temporaire pour générer un format standard c-à-d accepté par la plus part des anti-Spams, se connecte aux base de données de ces derniers pour les mettre à jour ou bien tout simplement dépose la liste de la mise à jour dans les base de données distribuées afin que les anti-Spams puissent l'avoir lorsqu'ils font les mise-à-jour.

3. Les cas d'utilisations

3.1 Définition des Acteurs

ACTEUR	Catégorie	Définition
Expéditeur	(externe)	➢ Envoi des E-mail
Destinataire	(interne)	➢ Réception des E-mail
Serveur Mail	(interne)	➢ Serveur mail vulnérable
Rep_Dist	(interne)	➢ Répertoire ou les E-mail sont stockés par le serveur mail (fichier texte)
Sys_Exp_loc	(interne)	➢ Le system ou toutes les analyses sont faite
Sys_Exp_dist	(interne)	➢ Le system ou il y a u serveur mail vulnérable
Rep_centr	(interne)	➢ Liste des domaines suspects (une table dans une base de données)
Domaine_susp	(interne)	➢ Liste des Domaines suspects (une table dans une base de données)
Mot_susp	(interne)	➢ Liste des mots suspects (une table dans une base de données)
IP_susp	(interne)	➢ Liste des IP suspectes (une table dans une base de données)
Anti-spam	(externe)	➢ Spamme Assassin (l'anti-spam ou on va tester notre RBL)

Table VIII.2 Définition des acteurs.

3.2 Les diagrammes de cas d'utilisation

3.2.1 Envoi d'E-mail

ACTEUR	CAS D'UTILISATION
Expéditeur	➢ Envoi un E-mail : - L'E-mail peut être un spam - L'E-mail peut être un Non spam
Destinataire	➢ Reçoit un E-mail : - L'E-mail peut être un spam - L'E-mail peut être un Non spam

Table VIII.3 Rôle des acteurs d'envoi d'E-mail.

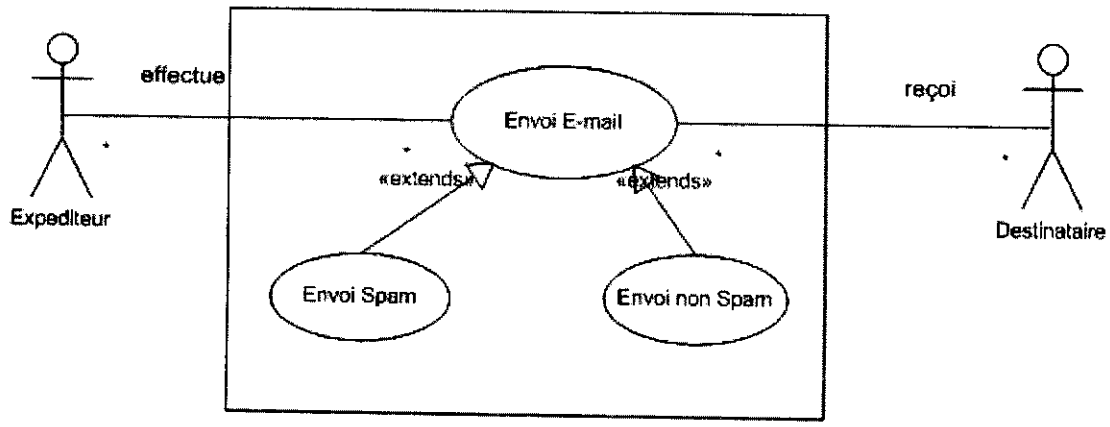


Fig. VIII.3 Cas d'utilisation d'envoi d'E-mail.

3.2.2 Stockage des E-mail

ACTEUR	CAS D'UTILISATION
Serveur Mail	➢ Apres la réception des E-mail il les stocks dans un répertoire local sous format de fichier Texte.
Répertoire Distant	➢ les E-mail sont stockés dans ce répertoire en format Texte.

Table VIII.4 Rôle des acteurs de stockage des E-mail.

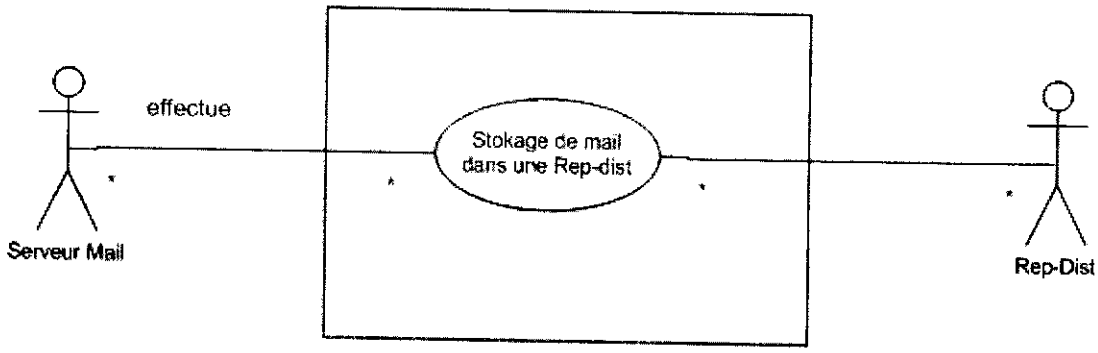


Fig. VIII.4 Cas d'utilisation de stockage des E-mail dans un Répertoire.

3.2.3 Ouverture d'une session ftp

ACTEUR	CAS D'UTILISATION
Système d'exploitation local	➢ Une session FTP est ouverte sur le serveur.
Système D'exploitation distant	➢ Ouvre une session FTP sur le système d'exploitation local. ➢ Procède à un dépôt de fichier sur le Répertoire central.
Répertoire Central	➢ Les fichiers sont stockés dans ce répertoire.

Table VIII.5 Rôle des acteurs d'ouverture d'une session FTP.

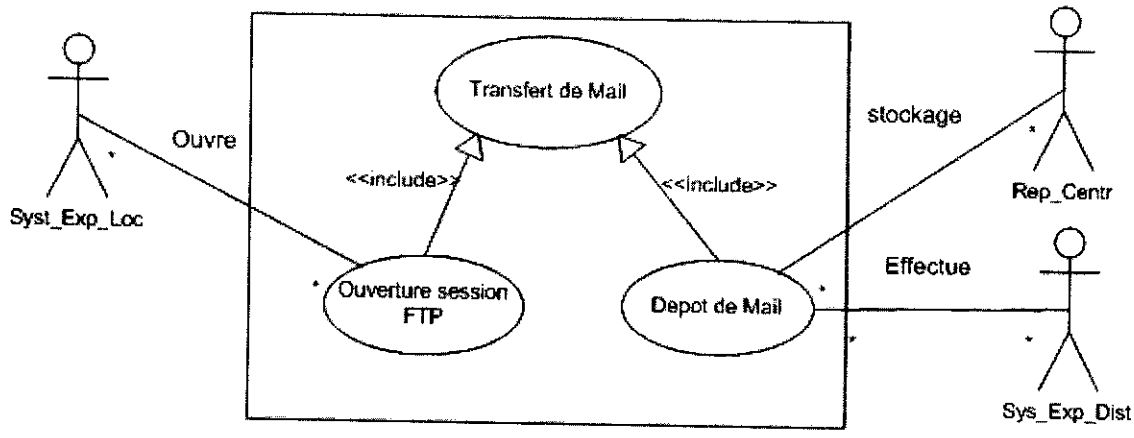


Fig. VIII.5 Cas d'utilisation d'ouverture de session FTP et Transfert d'E-mail.

3.2.4 Affectation dans la base de données

ACTEUR	CAS D'UTILISATION
Répertoire Central	➢ Une lecture est effectue sur le Répertoire central. ➢ Les fichiers inclues dans le répertoire sont découpés par le module de découpage. ➢ Apres chaque extraction d'une partie de L'E-mail il est affecté dans une table de la base de données

Table VIII.6 Rôle des acteurs d'affectation dans la base de données.

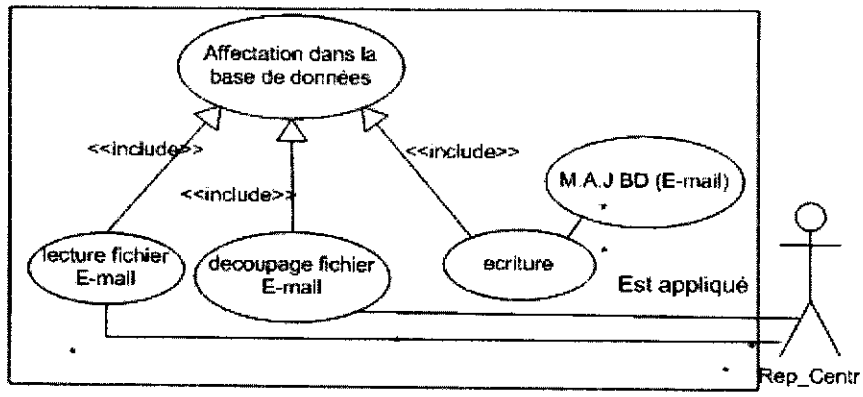


Fig. VIII.6 Cas d'utilisation d'affectation dans la base de données.

3.2.5 Analyse des E-mail

ACTEUR	CAS D'UTILISATION
Mot_susp	➢ Quand l'analyse et lancer le module « analyse Mot » consulte la base de données plus précisément la table des mots suspects.
Domaine_susp	➢ Quand l'analyse et lancer le module « analyse domaine » consulte la base de données plus précisément la table des domaines suspects.
IP_susp	➢ Quand l'analyse et lancer le module « analyse IP » consulte la base de données plus précisément la table des adresses IP suspects.

Table VIII.7 Rôle des acteurs d'analyse des E-mail.

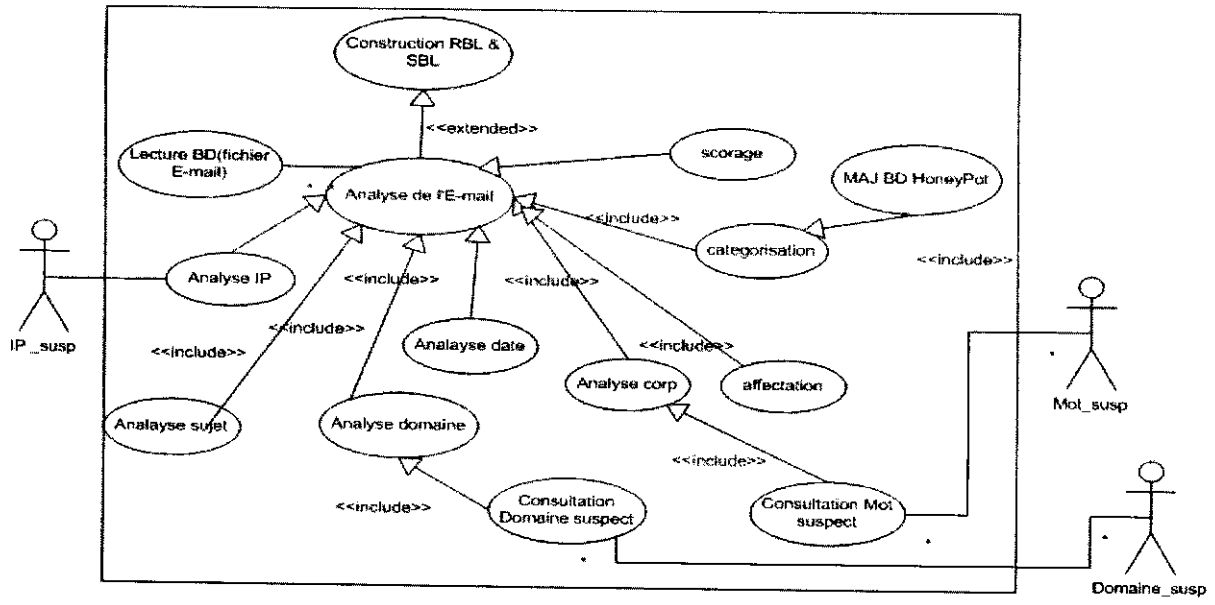


Fig. VIII.7 Cas d'utilisation d'analyse des E-mail.

3.2.6 Mise à jour des Anti-spam :

ACTEUR	CAS D'UTILISATION
Système d'exploitation distant	➢ Ouvre une session sur la base de données du répertoire central
Anti-spam	➢ Récupère le fichier en format RBL ou SBL on consultant la base de données.

Table VIII.8 Rôle des acteurs de mise à jour de l'Anti-spam.

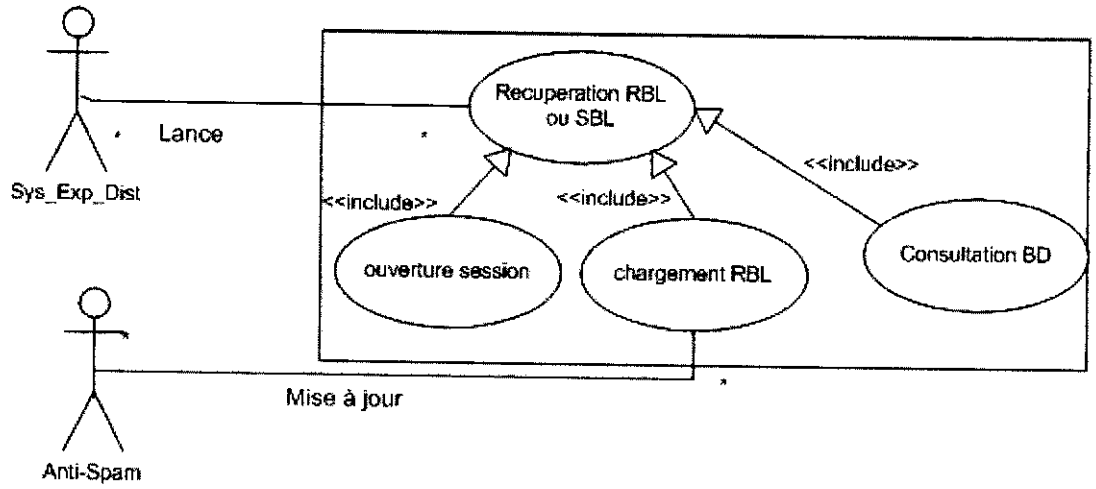


Fig. VIII.8 cas d'utilisation de mise à jour des anti-spam.

3.3 Diagramme de cas d'utilisation global

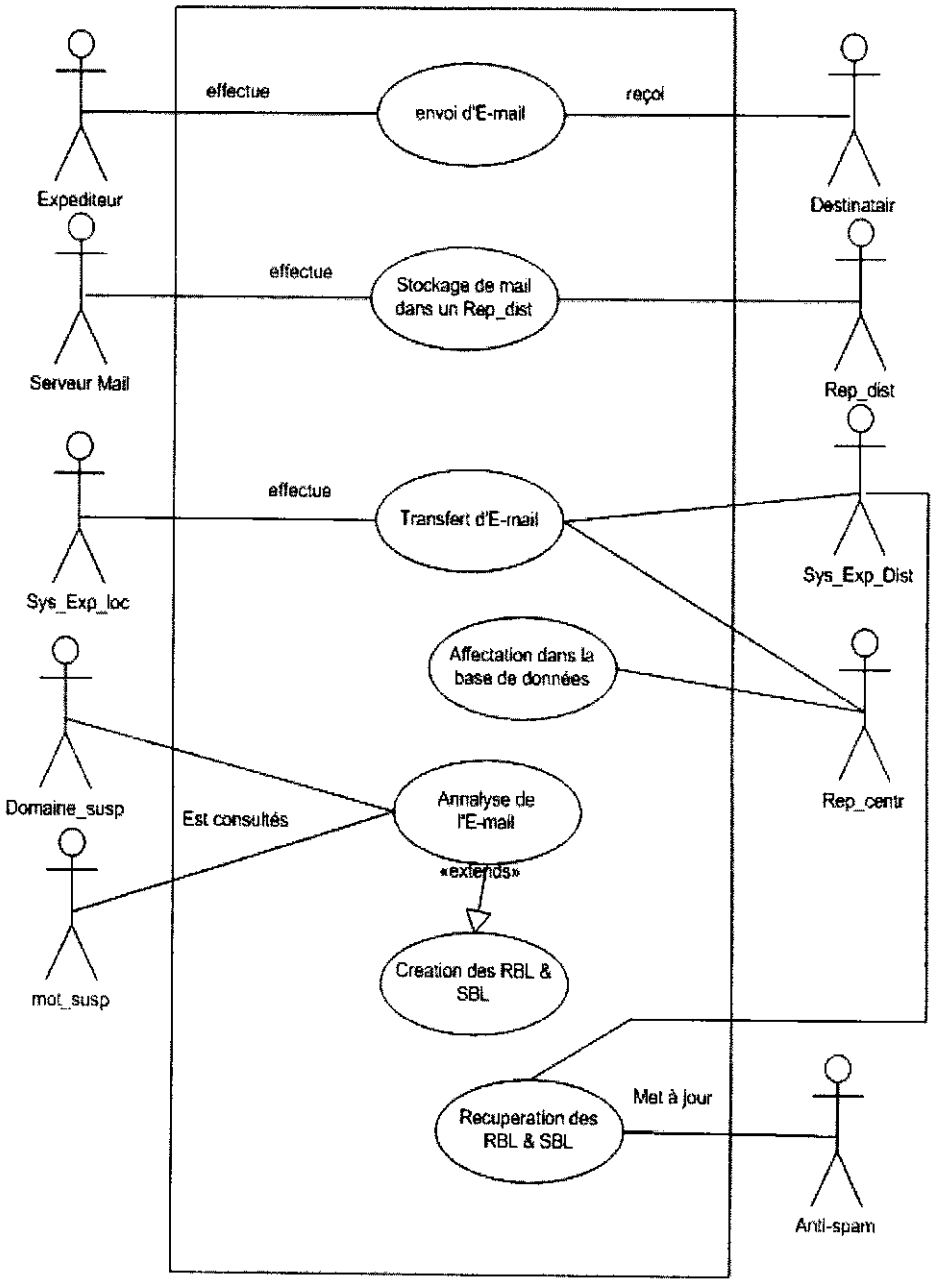


Fig. VIII.9 Diagramme de cas d'utilisation global.

3.4 Digramme de transition

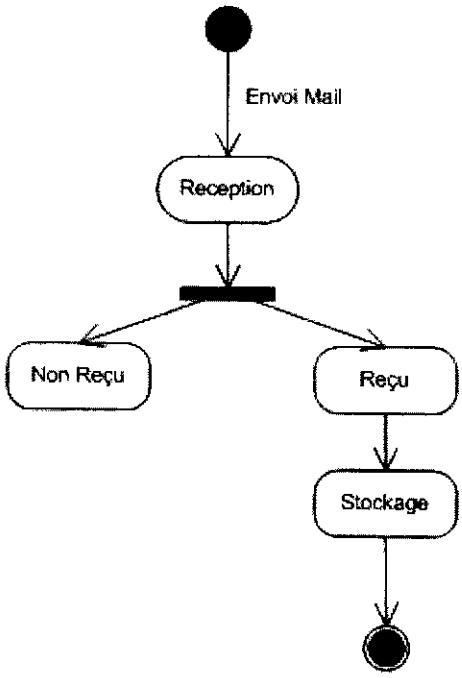


Fig. VIII.10 Diagramme de transition d'envoi d'E-mail.

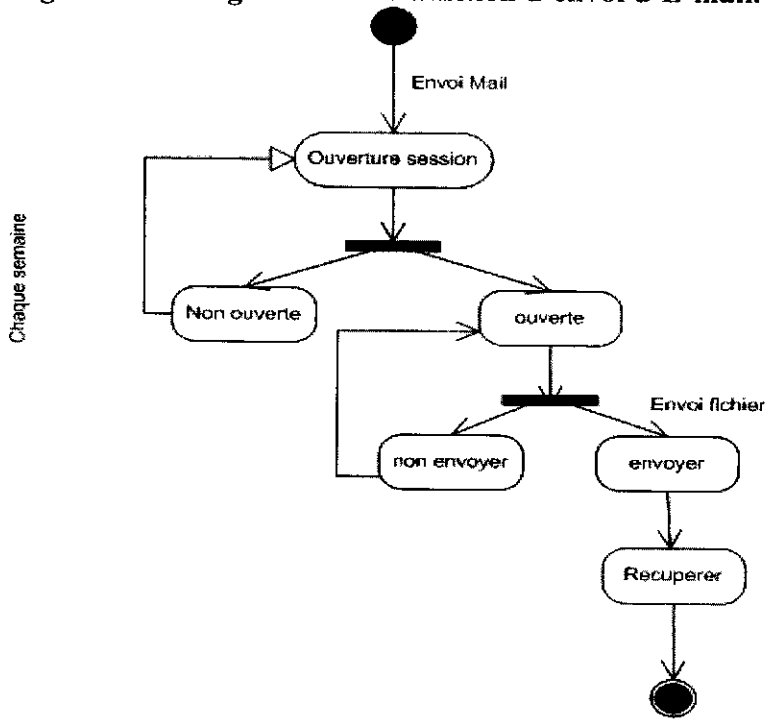


Fig. VIII.11 Diagramme de transition d'ouverture de session.

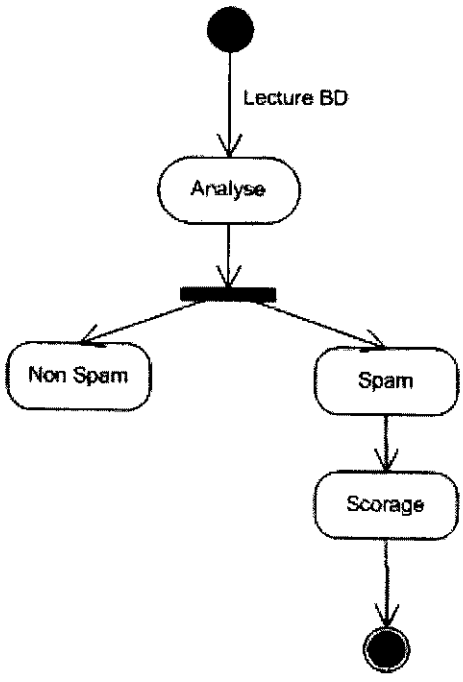


Fig. VIII.12 Digramme de transition d'analyse d'E-mail.

3.5 Diagramme de séquence et de communication

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur. Il indique les objets que l'acteur va manipuler et les opérations qui font passer d'un objet à l'autre. On peut représenter les mêmes opérations par un diagramme de communication graphe dont les nœuds sont des objets et les arcs (numérotés selon la chronologie) les échanges entre objets.

En fait, diagramme de séquence et diagramme de communication sont deux vues différentes mais logiquement équivalentes (on peut construire l'une à partir de l'autre) d'une même chronologie. Ce sont des diagrammes d'interaction.

3.5.1 L'envoi d'un mail

1. un expéditeur envoie un ou plusieurs mails à un ou plusieurs serveurs.
2. le serveur mail stocke les emails dans le rep_temp_dest.

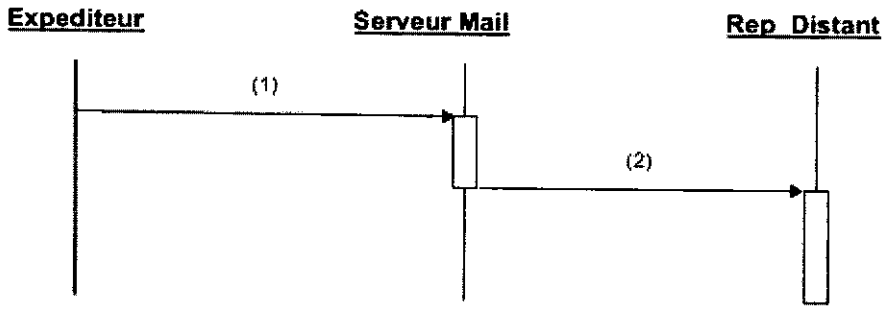


Fig. VIII.13 Diagramme de séquence d'envoi d'E-mail.

3.5.2 Ouverture d'une session ftp

1. le module d'ouverture de session ftp ouvre une session ftp sur le système distant.
2. une confirmation d'ouverture est envoyée au module d'ouverture de session.
3. le module transfère le contenu du rep_dist au rep_temp_central.

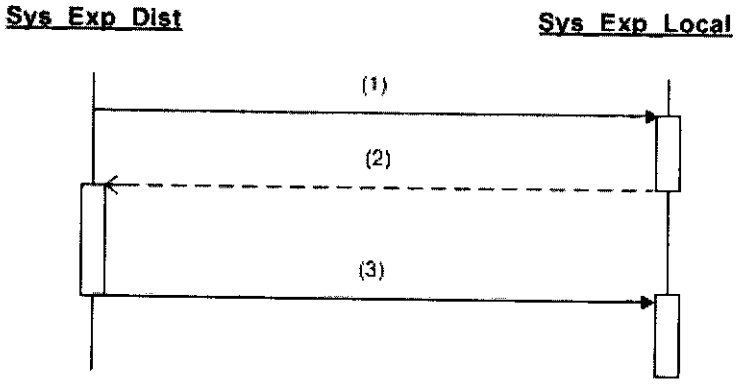


Fig. VIII.14 Diagramme de séquence d'ouverture d'une session FTP.

3.5.3 Module de subdivision

1. Le module de Découpage lit et décompose le fichier à partir Fichier_mail.
2. Le fichier est Décomposé en différentes mails.
3. Le module de Découpage décompose le mail en différentes champs.
4. Le mail est décomposé en différents champs.
5. Récupère chaque champ.
6. Le champ est récupéré.
7. Insère les différents champs dans mail_décomp (BD).
8. Tous les champs d'E-mail sont stockés dans la base de données .

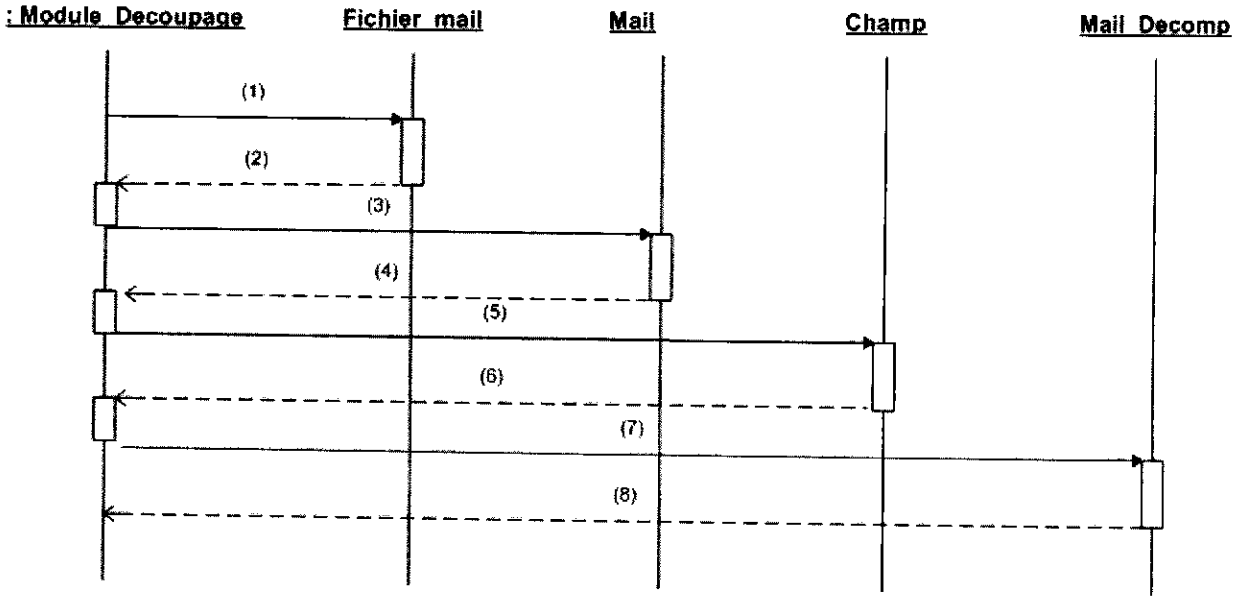


Fig. VIII.15 Diagramme de séquence de subdivision.

3.5.4 Module d'analyse et de catégorisation

1. Le 1er algorithme vérifie le **[sujet]** ; puis il donne un score d'évaluation selon qu'il est suspect ou pas, il ajoute le score ainsi obtenu au score précédemment donné au mail.
2. La fin du 1er algorithme déclenche le début du 2eme algorithme.

3. Le 2eme algorithme vérifie le champ **[adresse IP]** ; puis il donne un score d'évaluation selon qu'elle est black-listée ou pas, il ajoute le score ainsi obtenu au score précédemment donné au mail.
4. La fin du 2eme algorithme déclenche le début du 3eme algorithme.

5. Le 3eme algorithme vérifie le champ **[domain-Sender]** ; puis il donne un score d'évaluation selon qu'il est black-listé ou pas, ou bien s'il est trusted ou pas, il ajoute le score ainsi obtenu au score précédemment donné au mail.
6. La fin du 3eme algorithme déclenche le début du 4eme algorithme.

7. Le 4eme algorithme vérifie la **[date]** ; puis il donne un score d'évaluation selon le temps séparant deux réceptions du message du même Sender, il ajoute le score ainsi obtenu au score précédemment donné au mail.
8. La fin du 4eme algorithme déclenche le début du 5eme algorithme.

9. Le 5eme algorithme **analyse** le **[corps du message]** ; puis il donne un score d'évaluation selon les mots utilisés ou bien les liens insérés, il ajoute le score ainsi obtenu au score précédemment donné au mail.
10. La fin du 5eme algorithme déclenche le début de l'algorithme de catégorisation.

11. L'algorithme de **catégorisation** selon des critères bien évalués détermine la catégorie du mail et l'écrit dans la case dédiée.
12. La fin de l'algorithme de catégorisation déclenche le début de l'algorithme de décision.

13. L'algorithme de **décision** selon le score et la catégorisation du message décide s'il est spam ou non en utilisant un mécanisme de décision bien adopté.

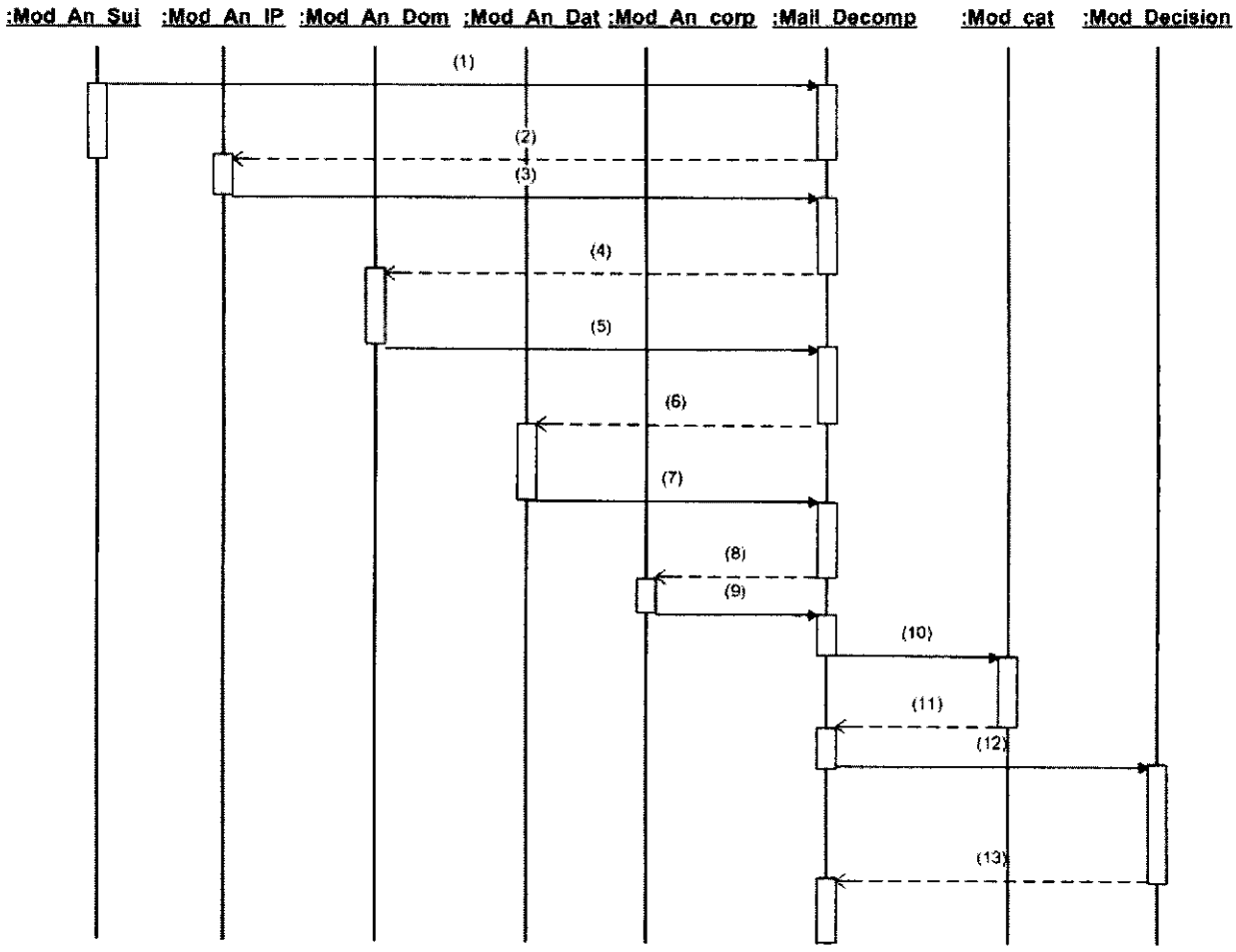


Fig. VIII.16 Diagramme de séquence D'analyse d'E-mail.

3.5.5 Manipulation de RBL

1. à partir de la table mail_décomp un module de génération de format standard génère un format standard (RBL ou SBL).
2. Un module de mise-à-jour communique la table RBL ainsi générée aux Anti-spam pour les mettre à jour.

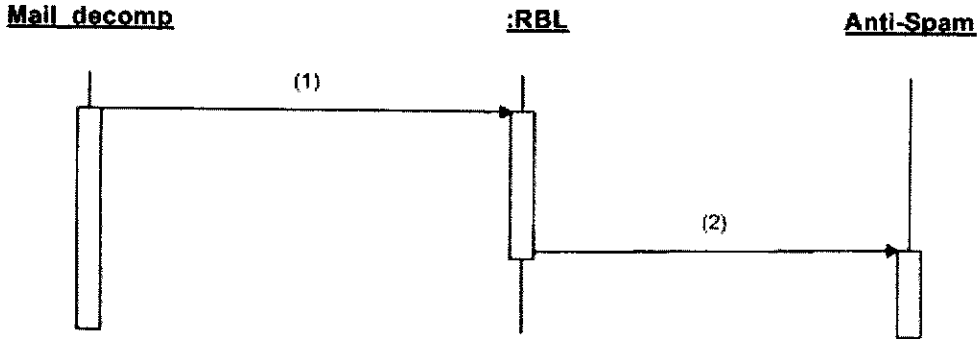


Fig. VIII.17 Diagramme de séquence d'envoi d'E-mail.

4. Diagramme de classe

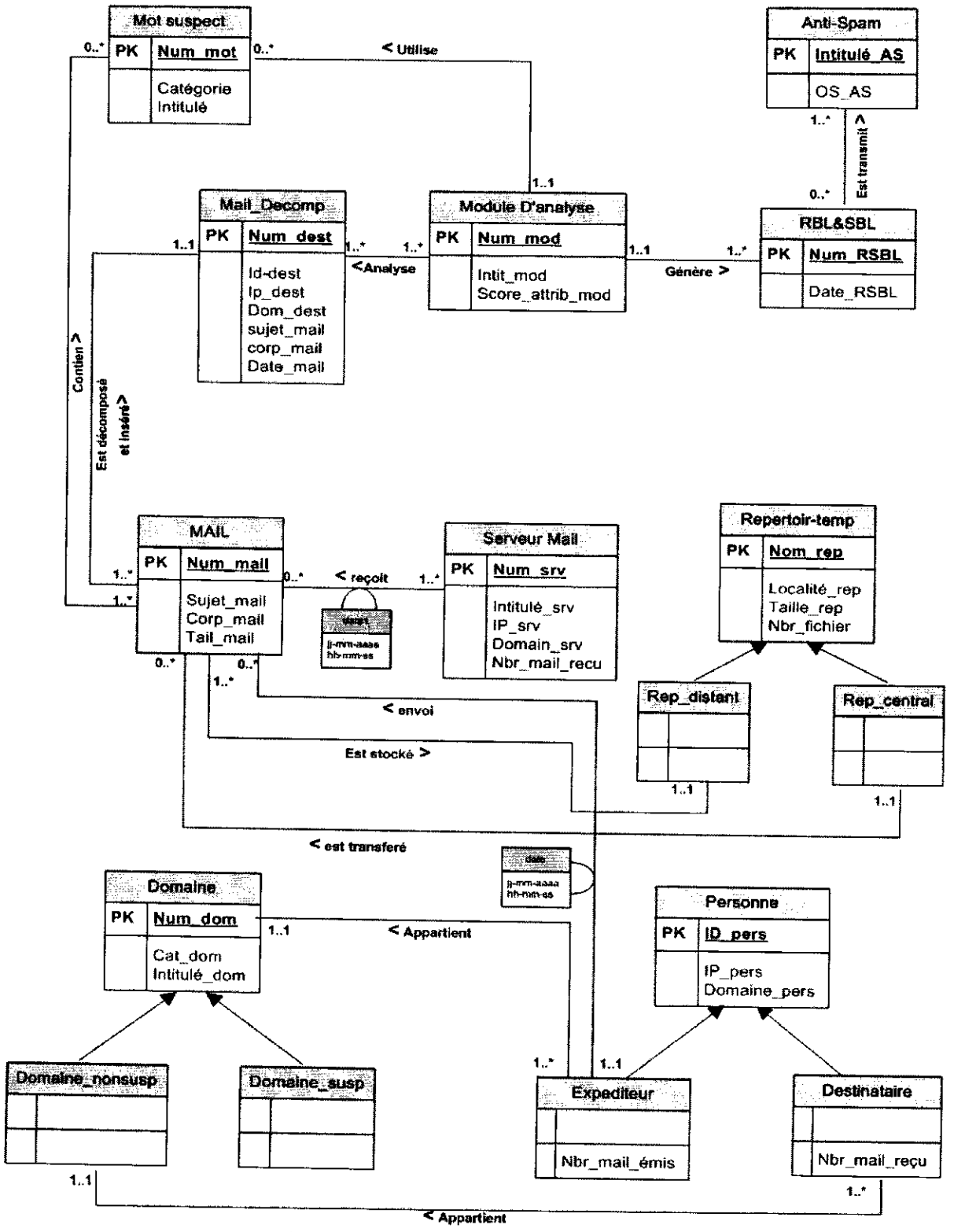


Fig. VIII.18 Diagramme de Classe.

5. Diagramme de déploiement

5.1 Définition

Les diagrammes de déploiement montrent la disposition physique des matériels qui composent le système et la répartition des composants sur ces matériels.
Les ressources matérielles sont représentées sous forme de nœuds. Les nœuds sont connectés Entre eux, à l'aide d'un support de communication. La nature des lignes de communication et leurs caractéristiques peuvent être précisées. Les diagrammes de déploiement peuvent montrer des instances de nœuds (un matériel précis), ou des classes de nœuds. **[UMD-34].**

5.2 Le diagramme de déploiement

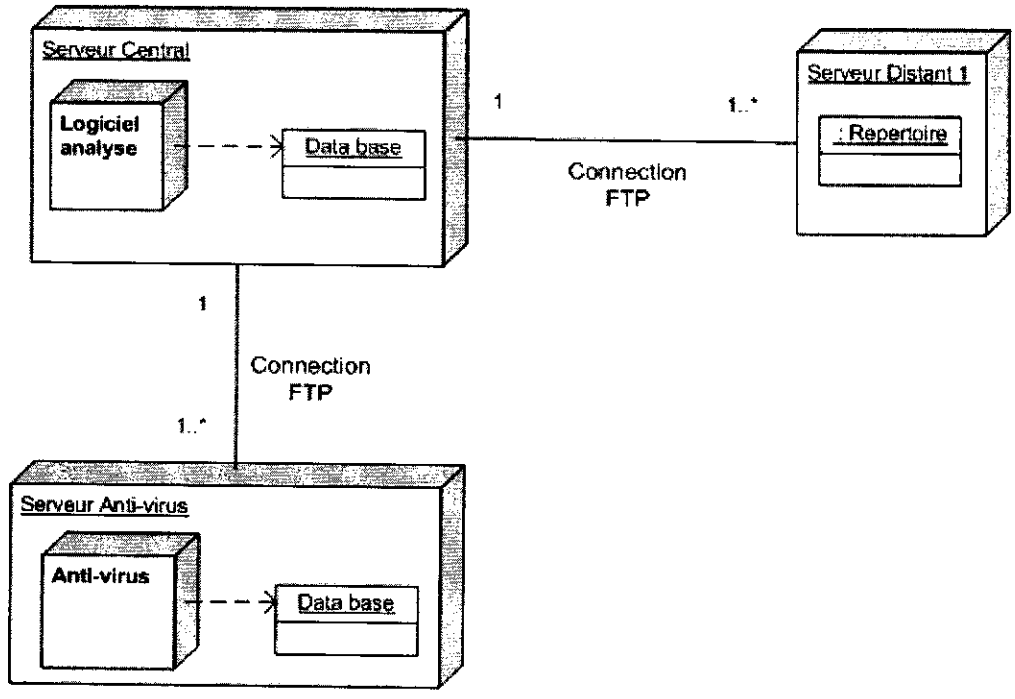


Fig. VIII.19 Diagramme de déploiement.

6. Conclusion

Dans ce chapitre, nous avons présenté une conception de l'application et une conception du système B2SAM.
De plus, par l'intermédiaire de plusieurs diagrammes, nous avons essayé de bien expliquer la solution proposée tout en illustrant l'ensemble de ses composants et leur manière d'agir, après la modélisation de la solution proposée, on passe maintenant à la réalisation, ce qui est proposé dans le chapitre suivant.

CHAPITRE IX.

Implémentation

Et mise en œuvre.

La ou il y a une volonté il y a un chemin.
– Belkas Mahmoud.

1. Introduction

Dans ce chapitre nous présentons une proposition d'une mise en œuvre de notre architecture déjà expliquée dans le chapitre décrivant la conception du « B2SAM », ainsi que l'explication illustrée de tous les modules d'analyse et de leurs interactions, en exposant tous les aspects importants de l'implémentation. Et on terminera par exposé les résultats expérimentaux de tests de notre système avec différents types de situations.

2. Choix du langage de programmation

L'efficacité des applications les plus avancées repose sur l'utilisation des langages les plus robustes pour garantir une meilleure interaction avec les différentes plateformes, le langage C++ est l'un des langages les plus appropriés pour le développement des applications même les plus complexes aux niveaux conceptuels.

Ainsi parmi les avantages de C++ qui on favorisé notre choix les suivants :

- C++ est un langage orienté objet qui préserve l'aspect de la programmation modulaire.
- Est le langage le plus utilisé dans le développement des Systèmes d'exploitation.
- La mobilité de l'exécutable vers différentes machines.
- Flexibilité de manipulation.
- Meilleur interaction avec le système d'exploitation et meilleur gestion des API, ainsi que l'efficacité de gestion des exceptions.

Pour IDE « Integrated Development Environment - environnement de développement intégré » on a choisi le « Borland Builder C++ Version 6 Entreprise Edition » grâce à sa meilleure gestion d'interface graphique la plus approprié pour l'utilisateur, et qui rend l'interaction avec un tel environnement plus flexible et plus efficace, ainsi que l'utilisation du débuggeur avancé qui est totalement compatible avec le système d'exploitation même au niveau le plus basic, aussi l'intégration de la technologie Microsoft ADO (*Active Data Object*) pour la gestion des base de données, et le regroupement d' un jeux de composants VCL intéressent.

3. Implémentation

Dans cette section nous allons exposer les modules utilisés par notre système dans le cadre de développement logiciel, et qui ont été spécifiés dans l'architecture proposées du système, présentées dans le chapitre « conception du B2SAM ».

3.1 Script d'ouverture d'une session ftp

Le script utilisé ici est un script écrit pour un système d'exploitation UNIX et ces dérivés en utilisant le Shell appelé « BASH » sous Linux RedHat pour ouvrir une session ftp. L'ouverture de cette session ftp est programmée chaque semaine pour récupérer les fichiers d'e-mails des systèmes distants on les déposant dans un répertoire spécifique et dédies à cette effet, ce répertoire est par défaut installé dans le « C:\ », et le fichier des e-mails est appelé « ANALYSE.txt ».

L'image suivante montre les instructions Shell utilisées pour programmer la tache d'ouverture de la session ftp sous Linux :

```
EDITOR = vi
export EDITOR
crontab -e
  10 21 * * 0 /etc/init.d/ftpagent
```

Fig. IX.1 Commandes d'automatisation de la tache.

Et voici le script approprie :

```
#!/bin/sh
#shell script to open an ftp session and get mails
#
 ftp
 open "name of local_domain"
 anonymous
 user@domain.com
 if [ftp session fails];then  loop
 else
     lcd /var/spool/mail
     binary
     put root
 fi
 bye
```

Fig. IX.2 Script d'ouverture d'une session ftp.

3.2 Module de chargement des e-mails

Ce module ouvre le répertoire par défaut (celui spécifié dans le script) et charge le fichier appropries dans le système pour l'analyser; ainsi si le répertoire par défaut n'existe pas seul l'administrateur pourra changer de répertoire pour charger les fichiers de l'analyse, c'est pour cette raison on a distingué entre l'utilisateur normale du système et l'administrateur qui possède le mot de passe approprie pour effectuer toutes les modifications nécessaires.

Le code suivant montre les méthodes utilisées pour la vérification de mot de passe ainsi que le code source du module de vérification de chemin et du chargement d'e-mails :

```
void __fastcall Password_check (TObject *Sender)
{
   // Vérifier si le mot de passe entré est valide :
 if (MaskEdit1->Text == Password)
  {
   pass = true;
   ...
  }
else
  {
  MessageDlg ("\n Mot de passe invalide, Veuillez consulter votre
   administrateur.", mtWarning, TMsgDlgButtons () << mbOK, 0);
  ...
  ...
  }
}
```

Fig. IX.3 Méthode de vérification de mot de passe.

```
String __fastcall path_ctrl (String path)
{
if (!FileExists(path))
  {
   if (MessageBox(Handle,"Le dossier source n'existe pas. ",
      "Erreur de chemin", MB_YESNO) == IDYES)
     {
      Password_Form->ShowModal();
      if (Password_Form->pass == true)
        {
         .
         .
         .
        }
     }
  }
  return path;
}
```

Fig. IX.4 Code de vérification de chemin.

```
void __fastcall chargement_Module (TObject *Sender)
{
String source_folder = "C:\\ANALYSE\\Analyse.txt";
try
   {
    source_folder = path_ctrl (source_folder);
    TStringList *Liste = new TStringList;
    Liste->LoadFromFile (source_folder);
    .
    .
    .
    delete Liste;
   }
catch (...) {}
}
```

Fig. IX.5 Code de chargement d'e-mails.

La figure suivante présente le code source du découpage des e-mails en utilisant un **automate** pour la gestion du procédé de découpage :

```
void __fastcall decoupage_Module (TObject *Sender)
{
// Module de découpage des e-mails chargés.

if (!Text_source->Text.IsEmpty())
  {
  for (int i=0; i <= Text_source->Lines->Count-1; i++)
     {
      .
      .   // Insertion des fragments des messages dans la table
      .   // d'analyse.
     }
  }
```

Fig. IX.6 Code de découpage des e-mails.

3.3 Module d'analyse entre deux tables de base de données :

```
void __fastcall Analyser (TADOTable *Table_work, TADOTable
   *Table_model, String champ_work, String champ_model, float
   score)
{
Table_work->DisableControls();
Table_model->DisableControls();
try
{
  Table_work->First();
  while (!Table_work->Eof)
  {
    Table_model->First();
    while (!Table_model->Eof)
    {
     .
     .
     .
    }
     Table_model->Next();
    }
    Table_work->Next();
  }
}
__finally
{
  Table_work->EnableControls();
  Table_model->EnableControls();
}
}
```

Fig. IX.7 Code de l'analyse entre deux tables.

3.4 Module d'analyse d'adresse IP

```
Analyser (Mail_decomp_table, IP_table, "IP_Field", "Intitule",
   score_IP);
```

Fig. IX.8 Code de l'analyse de l'adresse IP.

3.5 Module d'analyse du domaine de l'expéditeur

```
Analyser (Mail_decomp_table, Domain_table, " Domain_Field ",
   "Intitule", score_ Domain);
```

Fig. IX.9 Code de l'analyse du domaine.

3.6 Module d'analyse du sujet de message

```
void __fastcall Subject_analyser (TObject *Sender)
{
Mail_decomp_table->DisableControls();
String subject_buffer = "";
try
{
  Mail_decomp_table->First();
  while (!Mail_decomp_table->Eof)
  {
    if(   subject_buffer.IsEmpty() ||
          .
          .
          .
          !subject_buffer.SubString(1,2).AnsiCompare("__")
      )
      {
       ... // l'ajout du score.
      }

    Mail_decomp_table->Next();
  }
}
__finally
 {
  Mail_decomp_table->EnableControls();
 }
}
```

Fig. IX.10 Code de l'analyse du sujet.

3.7 Module d'analyse du corps des E-mail

```
void __fastcall Body_analyser (TObject *Sender)
{
Analyser (Mail_decomp_table, Sword_table, "Corp_mail",
   "Intitule", score_mot);      // analyse basée sur les mots.
Analyser (Mail_decomp_table, Phrase_table, "Corp_mail",
   "intitule", score_phrase);  // analyse basée sur les phrases.
}
```

Fig. IX.11 Code de l'analyse du corps du message.

4. Réalisation de l'application

L'application réalisée est une interface graphique manipulant notre système de traitement des e-mails jusqu'à la génération du format standard de mise à jour.
Ainsi la figure suivante présente l'interface principale de notre logiciel :

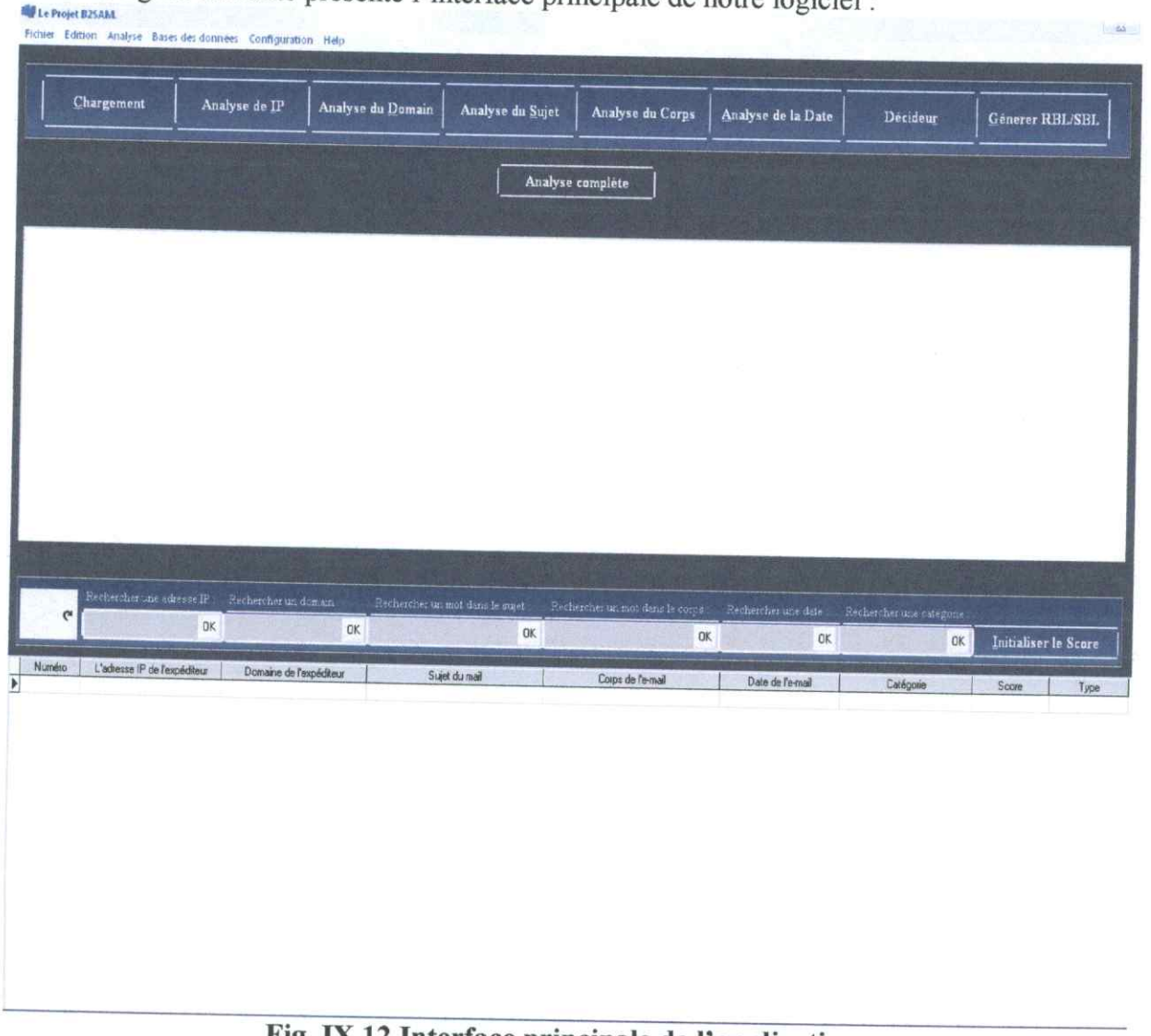


Fig. IX.12 Interface principale de l'application.

4.1 Modification de la configuration

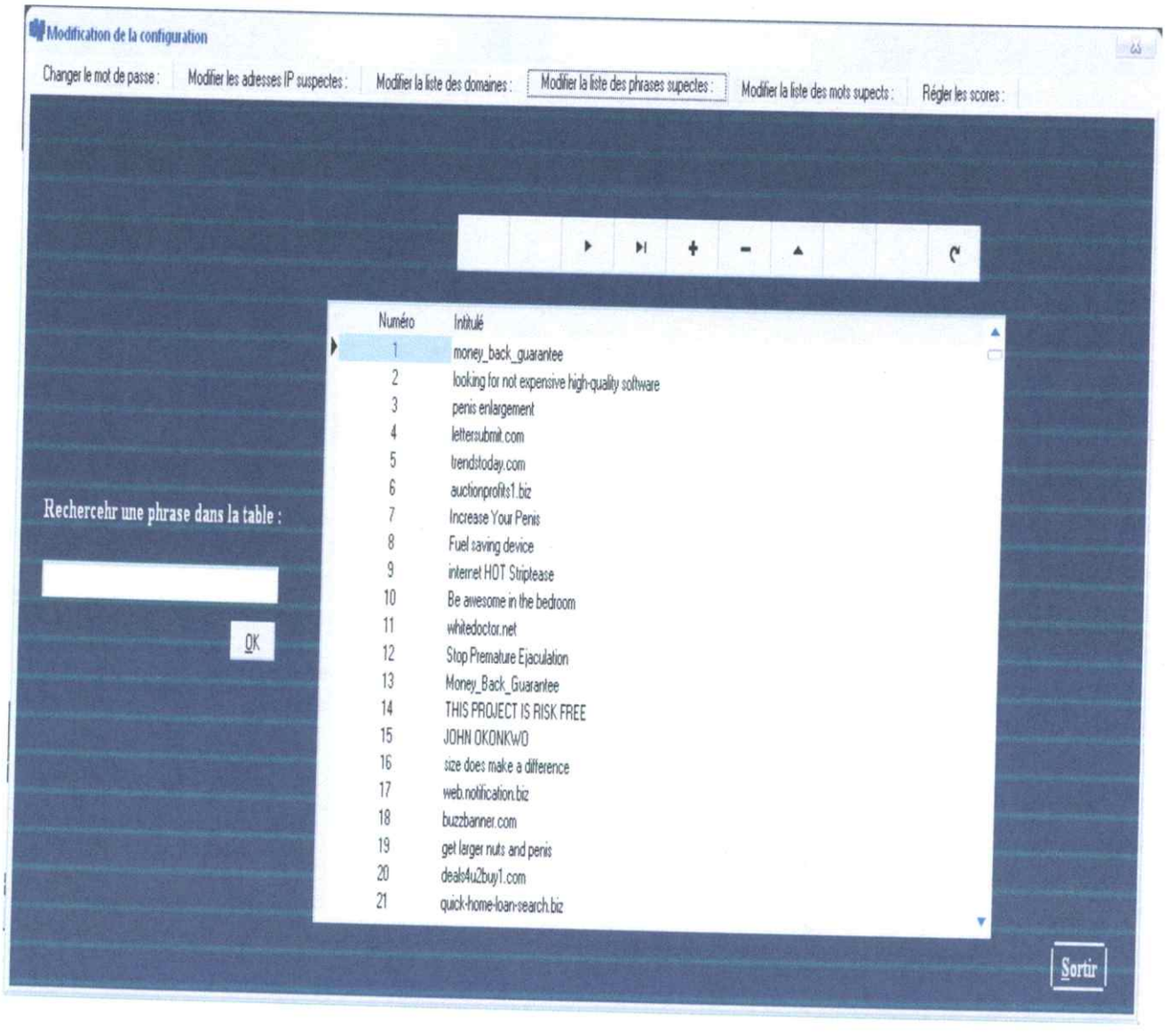


Fig. IX.13 Fiche de modification de la configuration.

4.2 Génération du format standard

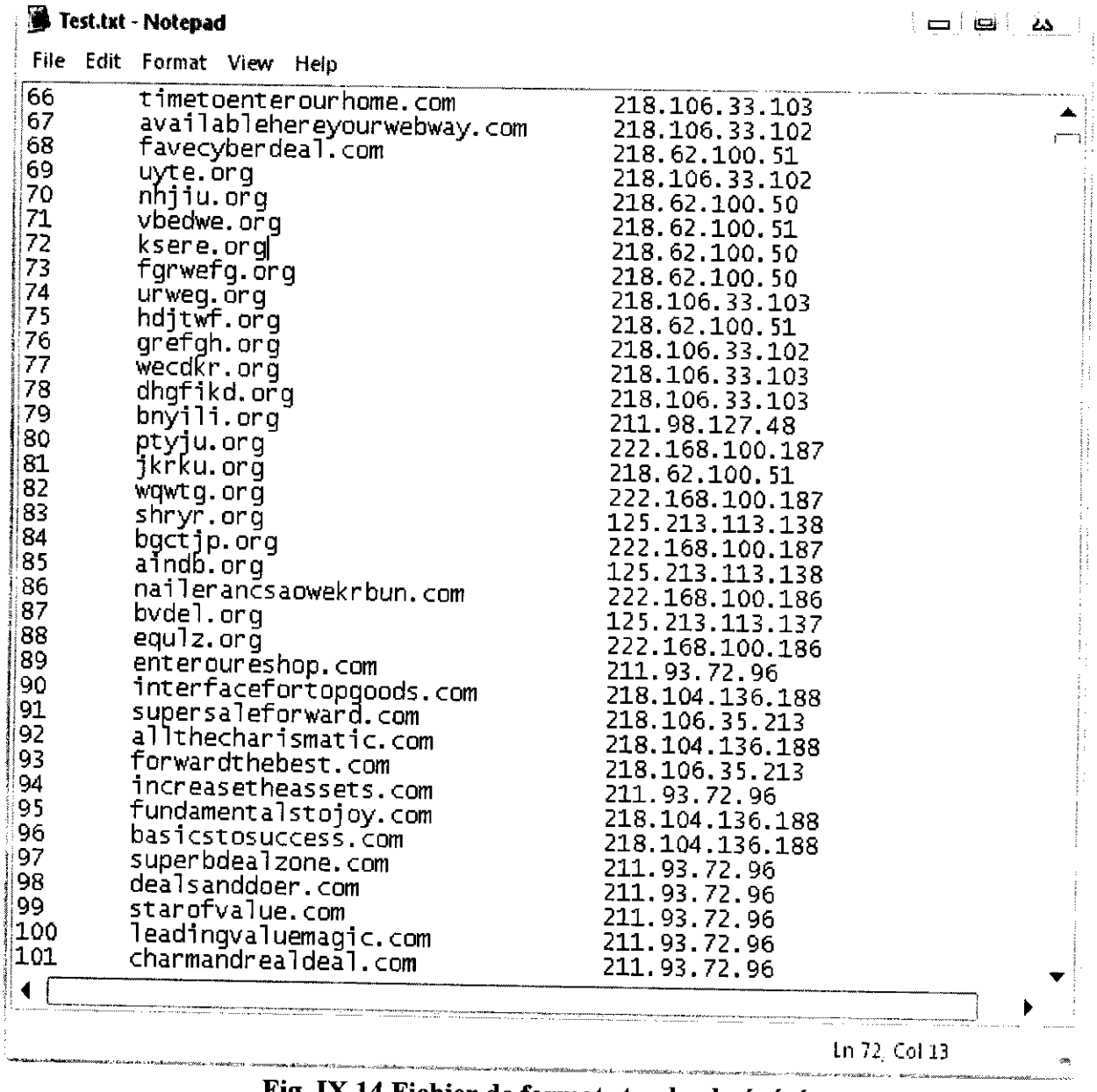


Fig. IX.14 Fichier de format standard généré.

5. Test effectué

Apres l'implémentation et la réalisation du système d'analyse B2SAM, et dans le but de tester son efficacité et son mise en œuvre on a mis au point une architecture réseau bien spécifique pour bien visualiser les étapes d'acheminement des E-mail à partir de centre de collecte d'E-mail jusqu'à la création du format standard RBL et mise à jour des anti-spams en passant par la moulinette de l'analyse, la figure suivante présente l'architecture de test globale du système HoneyPot Mail, avec le serveur mail « SendMail » et le system d'exploitation Linux RedHat qui on été utilisés dans le but de la collecte des E-mail, l'ouverture d'une session FTP et l'envoi du fichier des E-mail en format texte au système local de l'analyse dans un répertoire bien spécifié ainsi que la mise à jour de l'anti-spam « *Spamassassin* » sous Linux et avec un serveur mail « SendMail ».

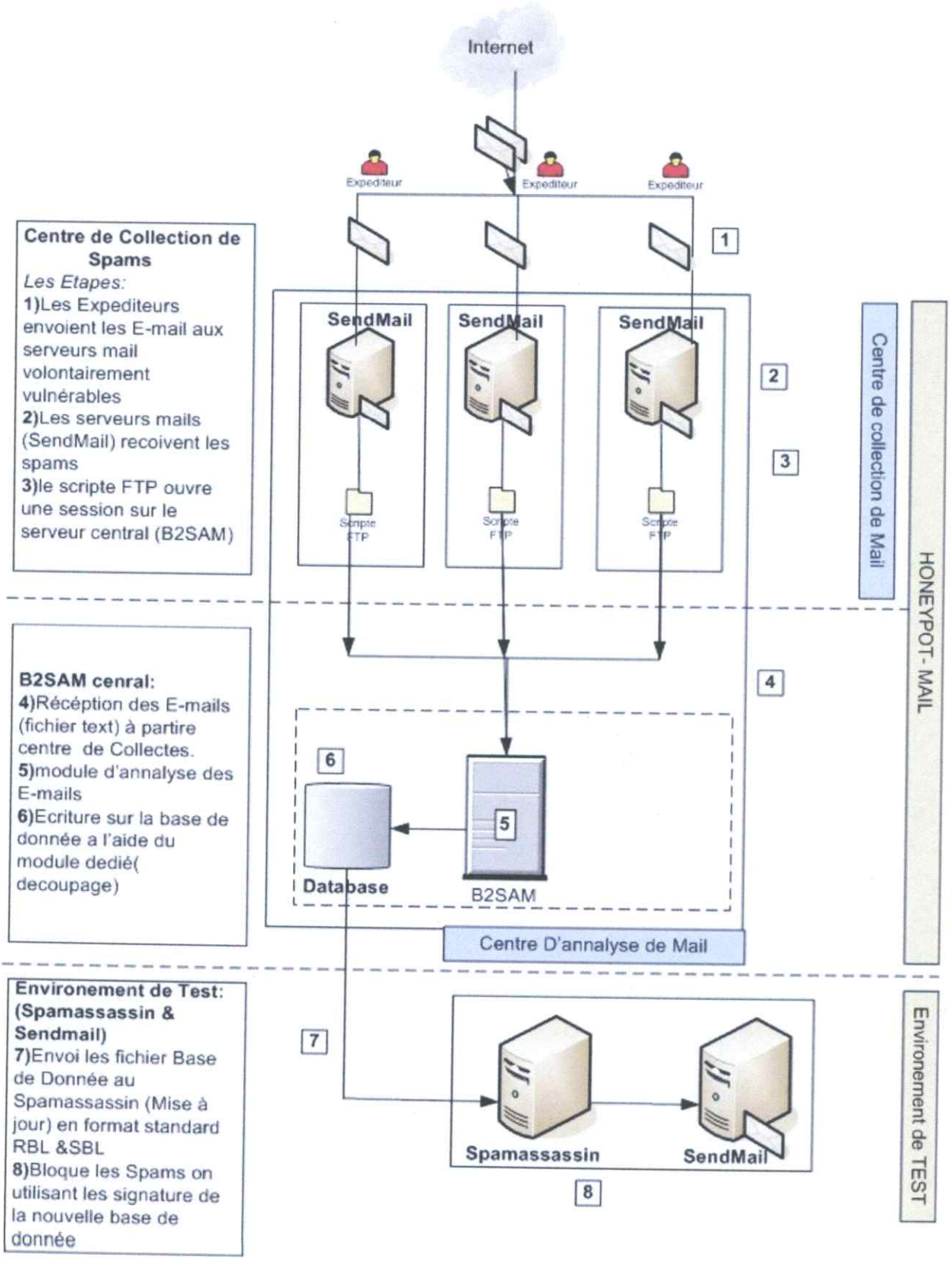


Fig. IX.15 Environnement de test du HoneyPot Mail.

L'utilisation des serveurs mail demande l'acquisition de **domaines** pour qu'ils puissent êtres publiés et par conséquence attaqués ; ce qui n'est pas dans le cadre de cette dissertation, donc on a simulé la mise en place de ces serveurs mail dans le but de tester l'efficacité de notre solution HoneyPot.

La figure suivante représente le test effectué en recevant des e-mails dans un domaine local entre deux machines de test possédant deux noms de domaines différents, en simulant le domaine (suspect) de l'émetteur et le domaine local du destinataire :

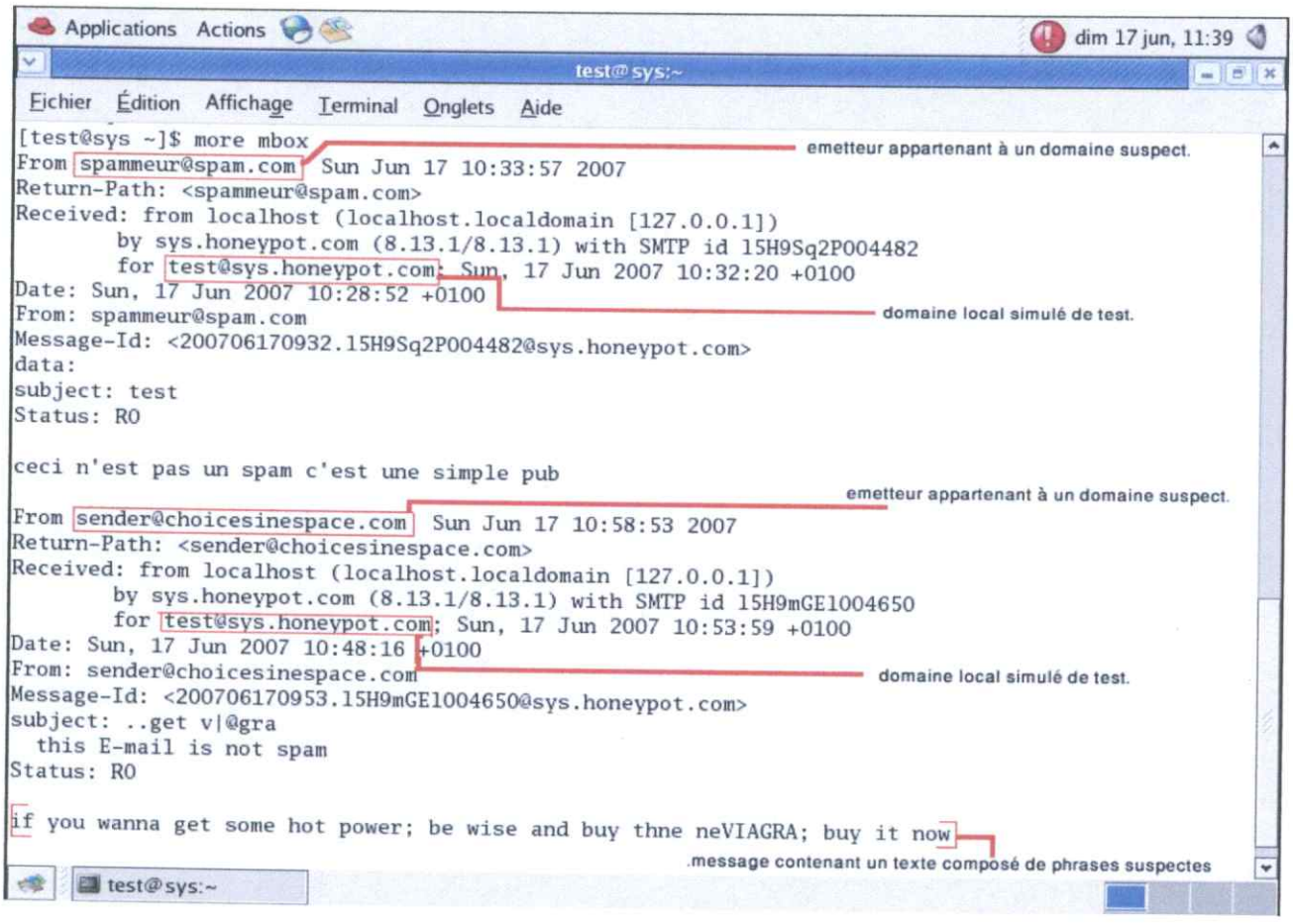


Fig. IX.16 Aperçus des messages reçus de différents domaines.

A cause de l'autonomie du fonctionnement de notre système c-à-d que la collecte des e-mails est indépendante de l'analyse, le déroulement de ce test commence par le lancement de notre application avec la configuration par défaut contenant le mot de passe « admin » et chemin de recherche de fichiers « C:\ANALYSE\ANALYSE.txt », on cliquant sur le bouton « chargement » le fichier par défaut est chargé dans l'éditeur de texte pour visualiser tous les e-mails qui ont été récupérés auprès les serveurs de collecte, puis l'analyse sera déclenchée par le bouton « analyse complète » et qui se déroule sur 6 étapes :

1. Découpage des e-mails et insertion dans la table de l'analyse.

2. Les analyses consécutives de l'adresse IP, du domaine, de sujet, et de la date et le corps du message sont exécutées une après l'autre et chaque analyse donne un score spécifique à elle.
3. Le module de catégorisation détermine la catégorie de chaque message pour mis à jour le champ « catégorie ».
4. Le module de décision en analysant les scores obtenus décide si le message est spam ou pas et écrit dans la case « type » la décision élue.
5. Le module de génération de format standard parcoure la table de l'analyse puis génère un format standard en incluant les messages qui ont été signalés comme spam ; ce format standard est sous forme texte et sauvegardé dans le fichier spécifique « C:\RBL/SBL\fichier1 ».
6. Le format standard ainsi généré est contacté aux anti-spams pour les mettre à jour et spécialement « *spam assassin* » dans la base de données locale de ce dernier.

La figure suivante montre la table d'analyse complète avec tous les enregistrements remplis :

Rechercher une adresse IP : GO | Rechercher un domain : GO | Rechercher un mot dans le sujet : GO | Rechercher un mot dans le corps : GO | Rechercher une date : GO | Rechercher une catégorie : GO | Initialiser le Score

Numéro	L'adresse IP de l'expéditeur	Domaine de l'expéditeur	Sujet du mail	Corps de l'e-mail	Date de l'e-mail	Catégorie	Score	Type
1	41.201.160.5	77-mail.net	_take care	take care of your self.....	5/12/2007 4:22:50 PM	Escroquerie	18	SPAM
2	64.233.171.83	gmail.com	urgent	bonjour je vous pris.....	5/12/2007 6:11:02 AM		0	Non spam
3	41.252.156.30	orlawyers.com	CAN YOU IMAGINE THAT YOU ARE HE.	LegalaRxMedication drug-shop propose a	5/13/2007 11:03:11 AM	Sante	23	SPAM
4	216.130.199.25	ns1.aisleymore.com	summer is almost here,be ready	take adventage of the chance--	5/13/2007 11:06:10 AM	Tourisme	28	SPAM
5	222.60.80.191	coldustesalandit.com	get out of the obese crowd	do not waste opportunity! anarim, The...	5/13/2007 11:56:20 AM	pornographie/escroquerie	25	SPAM
6	218.62.100.51	supersaleessential.com	stop the painful craving fot\r more food	do your counte up all the cases when you	5/16/2007 2:56:20 AM	pornographie/escroquerie	30	SPAM
7	218.62.100.51	favecybermarket.com	stop the painful craving fot\r more food	do your counte up all the cases when you	5/16/2007 2:57:10 AM	pornographie/escroquerie	30	SPAM
8	218.62.100.51	favecybermarket.com	stop the painful craving fot\r more food	do your counte up all the cases when you	5/16/2007 2:57:22 AM	pornographie/escroquerie	30	SPAM
9	218.62.100.51	favecybermarket.com	stop the painful craving fot\r more food	do your counte up all the cases when you	5/16/2007 2:58:54 AM	pornographie/escroquerie	30	SPAM
10	218.62.100.51	favecybermarket.com	stop the painful craving fot\r more food	do your counte up all the cases when you	5/16/2007 2:59:03 AM	pornographie/escroquerie	30	SPAM
11	64.4.33.7	hotmail.com	invitation	la societe TechRepublic organise.....	5/18/2007 4:32:20 PM		5	Non spam

Fig. IX.17 Table d'analyse dans un environnement de test.

Si l'utilisateur désire changer la configuration ou bien modifier la table des mots suspects et ou bien la table des phrases suspectes il invité à fournir le mot de passe administrateur comme illustre la figure suivante :

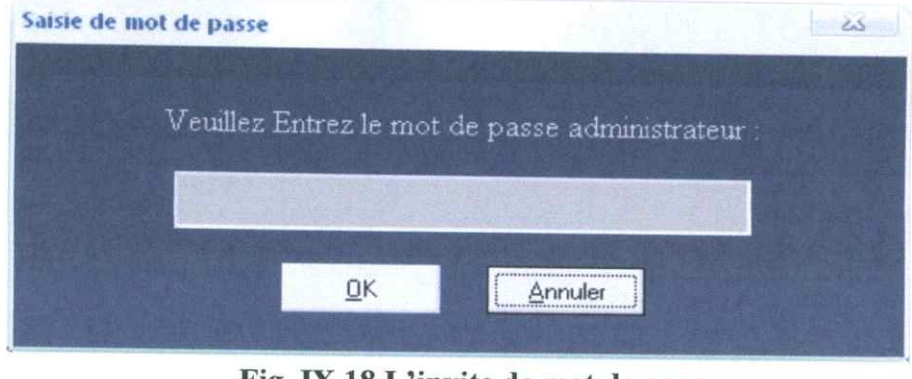


Fig. IX.18 L'invite de mot de passe.

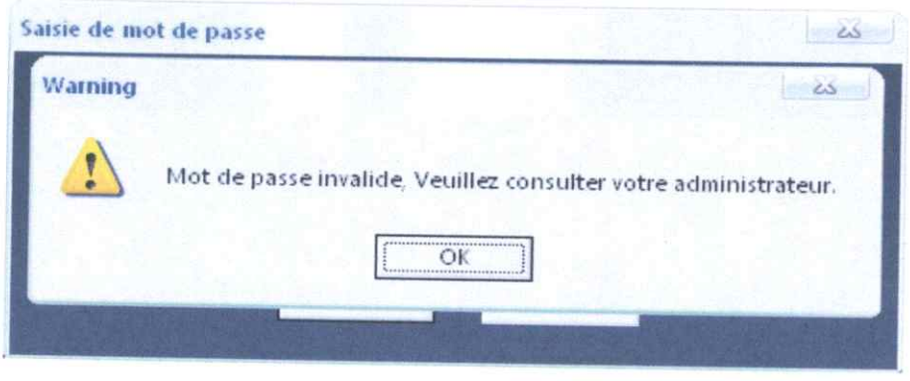


Fig. IX.19 Mot de passe invalide.

Si le mot de passe est valide il peut personnaliser la configuration comme il est montré dans la **Fig. IX.13.**

6. Conclusion

Dans ce Chapitre, nous avons présenté. Une proposition d'une mise en œuvre possible de notre architecture décrit auparavant, à base d'un mélange de techniques et de méthodes pour récupérer les emails et faire les analyses correspondantes pour enfin générer un format standard de mis à jour.

Le coté conversationnel a été possible grâce à l'utilisation de techniques de programmation les plus avancées et les API de manipulation des systèmes pour formaliser les interactions nécessaires.

Notre système a donné des résultats satisfaisants, mais des améliorations restent encore à faire, notamment en ce qui peut survenir pour les techniques d'analyse surtout avec le développement de la technique d'analyse des images.

Conclusion générale.

Le gain de notre étude
c'est en être devenu
meilleur et plus sage.
-Michel Eyquem.

L'objectif primordial du présent travail consistait à mettre en œuvre une solution distribuée dans le but de piéger les spammeurs et connaitre leurs méthodes de procéder pour mieux protéger les systèmes de messagerie en offrant des formats standards de mise à jour pour les anti-spams.

La motivation qui sous-tend ce travail réside principalement dans le fait que les pertes liées aux Spams sont considérables comme la limitation de la bande passante ou la perte du temps et de l'espace quelle soit du coté serveur de messagerie ou du coté utilisateur de mailing.

Au cours de notre projet nous avons développé un Système dans le but est de collecter le maximum de Spams en s'appuyant sur des serveurs mail volontairement vulnérables ce qui est bien le « HoneyPot mail » pour dévoiler le profile du spammeur et mieux l'anticiper, en masquant tous les obstacles liés aux hétérogénéités des systèmes.

Perspectives :

En perspective, nous proposons quelques extensions du système, qui se résument dans ce qui suit :

- L'ajout de module d'analyse basé sur l'analyse des images et de reconnaissance des formes.
- L'adoption de nouveaux formats de mise à jour basé sur les formats de base de données et non seulement les formats texte.

Annexe.

1. Le modèle OSI

1.1 Historique

À ses débuts, le développement des réseaux était désorganisé à de nombreux égards. Le début des années 1980 a été marqué par une croissance exceptionnelle du nombre et de la taille des réseaux. Pour profiter des avantages de l'utilisation de la technologie réseau, les sociétés se mirent à installer et à étendre des réseaux, à un rythme presque aussi rapide que l'introduction des nouvelles technologies.

Vers le milieu des années 80, ces sociétés commencèrent à rencontrer des problèmes, justement en raison de cette extension rapide. Tout comme il est difficile pour les gens qui ne parlent pas la même langue de communiquer, il devenait difficile pour des réseaux qui utilisent des spécifications et des implémentations différentes d'échanger des informations. Les sociétés qui développaient des solutions de réseau privé ou propriétaire furent confrontées au même problème. Le terme « propriétaire » signifie ici qu'une entreprise ou un petit groupe d'entreprises contrôle entièrement l'utilisation de la technologie. Les technologies réseau qui suivaient strictement des règles propriétaires ne pouvaient pas communiquer avec des technologies qui respectaient des règles propriétaires différentes.

Pour résoudre le problème de l'incompatibilité des réseaux, l'OSI (*International Organization for Standardization*) examina les modèles réseau tels que DECnet (*Digital Equipment Corporation net*), SNA (*Systems Network Architecture*) et TCP/IP afin de trouver un ensemble de règles applicables de façon générale à tous les réseaux. Sur la base de ces recherches, l'ISO a mis au point un modèle de réseau pour aider les fournisseurs à créer des réseaux compatibles avec d'autres réseaux.

Le modèle de référence OSI (*Open System Interconnection*) publié en 1984 fut le modèle descriptif de réseau créé par l'ISO. Ce modèle propose aux fournisseurs un ensemble de normes assurant une compatibilité et une interopérabilité accrues entre divers types de technologies réseau produites par de nombreuses entreprises à travers le monde.

Le modèle de référence OSI s'est imposé comme le principal modèle pour les communications réseau. Bien qu'il existe d'autres modèles, la plupart des constructeurs de réseau se fondent sur le modèle de référence OSI pour la conception de leurs produits. Cela est particulièrement vrai lorsqu'ils souhaitent éduquer les utilisateurs à l'utilisation de leurs produits. Ce modèle est considéré comme le meilleur outil disponible pour décrire l'envoi et la réception de données sur un réseau.

Les avantages du modèle OSI :

- Il réduit la complexité.
- Il uniformise les interfaces.
- Il facilite la conception modulaire.
- Il assure l'interopérabilité de la technologie.
- Il accélère l'évolution.
- Il simplifie l'enseignement et l'acquisition des connaissances.

1.2 Les couches du modèle OSI

Le modèle de référence OSI constitue un cadre qui aide à comprendre comment les informations circulent dans un réseau. Ce modèle explique comment les paquets transitent à travers les différentes couches vers un autre équipement du réseau, même si l'émetteur et le destinataire utilisent des types de média réseau différents.

Le modèle de référence OSI comporte sept couches numérotées, chacune illustrant une fonction réseau bien précise. Le découpage du réseau en sept couches présente les avantages suivants:

- Il permet de diviser les communications sur le réseau en éléments plus petits et plus gérables.
- Il uniformise les éléments du réseau afin de permettre le développement et le soutien multi constructeur.
- Il permet à différents types de matériel et de logiciel réseau de communiquer entre eux.
- Les modifications apportées à une couche n'affectent pas les autres couches.
- Il divise les communications sur le réseau en éléments plus petits, ce qui permet de les comprendre plus facilement.

La figure suivante illustre les différentes couches du modèle OSI :

Fig. A.I Les couches du modèle OSI.

Les différentes couches du modèle sont :

1. Couche physique

Assure une transmission binaire à l'aide des fils,connecteur, tension et débit.

2. Couche liaison de données

Elle fait un contrôle de liaison directe pour l'acces au media on assurant un transfert fiable des données par le media, et une connectivité et selection du chemin entre les système hotes.

3. Couche Réseau

Donne une adresse réseau et cherche le chemin le plus fiable pour le transfert des données on assurant un transfert fiable des données par le media, et une connectivité et selection du chemin entre les système hotes ainsi que l'adressage logique et acheminement au mieux.

4. Couche transport

Est une connexion de bout en bout, elle concerne les problèmes de transport entre les hôtes, on assurant :

- une fiabilité de transport de données.
- Etablissement, maintien et terminaison des circuits virtuels.
- Une détection des pannes et reprise du contrôle du flux d'information.

5. Couche session :

Elle assure une communication entre les hôtes, elle établit gère et ferme les sessions entre les applications.

6. Couche Présentation :

Elle assure :

- La lisibilité des données pour le système de destination.
- Format des données.
- Structure des données.
- Négocie la syntaxe de transfert des données pour la couche application.

7. Couche application :

Elle fournit des services aux processus d'application tels que le courrier électronique, le transfert de fichiers et l'émulation de terminal. **[AMO-4].**

2. Le Projet SPAMHAUS

Le projet Spamhaus est une organisation internationale à but non lucratif dont la mission est de dépister les troupes du Spam de l'Internet, d'assurer la protection sûre d'anti-Spam à temps réel pour des réseaux Internet, pour travailler avec des agences d'application de loi pour identifier et pour poursuivre des Spammeurs dans le monde entier, et pour inciter des gouvernements pour la législation efficace d'anti-Spam.

Fondé en 1998, Spamhaus est basé à Genève, en Suisse et à Londres, il est couru par une équipe consacrée de 25 investigateurs et spécialistes situés dans 9 pays.
Spamhaus édite le registre des opérations connues de Spam (ROKSO) – contient plus de 200 base de donnée qui stock les plus mauvaises troupes connues de Spam dans le monde entier.

D'après des statistiques dévoilées le 9 juin 2007 les listes de bloc de Spamhaus protègent 645.112.000 boîtes aux lettres d'utilisateur.
Spamhaus édite un certain nombre de bases de données de Spam-blocage en temps réel, y compris la liste de bloc de Spamhaus (SBL), la liste de bloc d'exploits (XBL) et la liste de bloc de politique (PBL).

Il est constitué d'un réseau de 40 serveurs de DNS dans 17 pays, les blocklists (SBL) de Spamhaus sont employés par plusieurs services fournisseurs d'internet principaux, des sociétés, des universités, des gouvernements et des réseaux militaires. L'infrastructure internationale est placée et contrôlée par une organisation séparée de logistique qui fournit des services utilisant des technologies de Spamhaus à l'industrie de filtre de Spam, y compris un service de synchronisation de blocklists de Spam assuré à de grands réseaux Internet et à organismes commerciaux de filtre de Spam. **[SHP- 32]**

Glossaire.

A

- **ASCII** (American Standard Code for Information Inter change) :
  Est la norme de codage de caractères en informatique la plus connue et la plus largement compatible. C'est également la variante américaine du codage de caractères ISO/CEI 646. Elle contient les caractères nécessaires pour écrire en anglais. Elle a été inventée par l'américain Bob Bemer en 1961. Elle est à la base de nombreuses autres normes (Unicode, ISO 8859-1, Windows-1252), qui l'étendent.
- **ASCII HT** (ASCII Horizontal tabulation).
- **ASRG** (Groupe de Recherche Anti-Spam).
- **ARPANET**(Advanced Research Projects Agency Network):
  Est le premier réseau à transfert de paquets développé aux États-Unis par la DARPA, le projet fut lancé en 1967 et la première démonstration officielle date d'octobre 1972. Il est le prédécesseur de l'Internet.
- **AUTODIN** (Automatic Digital Network) :
  Est un service de communications de données dans le département des Etats-Unis de la défense.

B

- **B2SAM** (Belkas Belhaoues System Analyses Module).
- **BAL** (Boîte Aux Lettres)
- **BBS** (Bulletin Board System):
  Consiste en un serveur équipé d'un logiciel offrant les services d'échange de messages, de stockage et d'échanges de fichiers, de jeux via un ou plusieurs modems reliés à des lignes téléphoniques.
- **Bcc** (Blind Carbon Copy) :
  Utilisée dans les courriels signifie en anglais « copie cachée ».Littéralement, il signifie "copie carbone aveugle".
- **BIOS** (Basic Input Output System):
  Un ensemble de fonctions, contenu dans la mémoire morte (ROM) de la carte mère servant à faire des opérations basiques (écrire un caractère à l'écran, lire un secteur sur un disque, etc. ...).
- **Booch :**
  L'une des méthodes d'analyse et de conception orientées objet à l'origine d'UML.
  Cette méthode permet de faciliter l'implémentation de programmes dans des langages de programmation orientée objet, ainsi que de représenter les différentes phases du développement d'un projet. Pour les programmeurs, l'avantage est d'obtenir une modélisation d'une solution qui soit indépendante de son implémentation dans un langage particulier (par exemple Java, C++, PHP...).

- **BST** (Basic Security Theorem):
  Théorème de sécurité des systèmes informatiques.

C

- **Cc** (copie carbone) :
  Etait à l' origine une méthode utilisant du papier carbone pour produire simultanément une ou plusieurs copies d'un document.
- **Cci** (Copie carbone invisible).
- **CERT** (Computer Emergency Response Team):
  Sont des centres d'alerte et de réaction aux attaques informatiques, destinés aux entreprises et/ou aux administrations, mais dont les informations sont généralement accessibles à tous.
- **CLUSIF** (Club de la sécurité de l'information français) :
  Est une association française d'entreprises et de collectivités réunis en groupes de réflexion autour de différents domaines de la sécurité de l'information : gestion des risques, droit, intelligence économique…
- **CLUSIS** (Association suisse de la sécurité des systèmes d'information).
- **CNIL** (Commission nationale de l'informatique et des libertés) :
  Est une autorité administrative indépendante française chargée de veiller à la protection des données à caractère personnel et de la vie privée. Elle a été créée par la loi n° 78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés.
- **CRLF LWP**(Chariot Return Linefeed) :
  Un retour chariot à la ligne.
- **CSMA/CD** (abréviation de Carrier Sense Multiple Access /Collision Detection) :
  Un protocole qui gère le partage de l'accès physique au réseau Ethernet.
- **CTSS** (Compatible Time-Sharing System) :
  Est l'un des premiers systèmes d'exploitation à temps partagé.

D

- **DCSSI** (Direction centrale de la Sécurité des systèmes d'information) :
  L'organisme interministériel officiel définissant les normes de la sécurité des systèmes d'information en particulier les normes sur l'évaluation et la certification des systèmes d'information.
- **DECnet** (Digital Equipment Corporation net):
  Le DECnet est un groupe de produits de communications de données, incluant une suite de protocole, développée et soutenue par la Digital Equipment Corporation (Digital). La première version du DECnet, 1975.
- **DNS** (Domain Name System) :
  Un système permettant d'établir une correspondance entre une adresse IP et un nom de domaine et, plus généralement, de trouver une information à partir d'un nom de domaine.
- **DOD** (Department Of Defense) :
  Est le ministère du gouvernement des États-Unis chargé de commander aux forces militaires du pays, en temps de guerre et en temps de paix.

- **DSL** (Digital Subscribe line):
  Est un mode de codage des données.

E

- **EBIOS** (Expression des Besoins et Identification des Objectifs de Sécurité) :
  Est une méthode d'évaluation des risques en informatique, développée par la Direction Centrale de la Sécurité des Systèmes d'Information.
- **EXIM** :
  Est un serveur de messagerie électronique (ou *Mail Transfer Agent* en anglais) utilisé sur de nombreux systèmes de type UNIX. La première version a été écrite en 1995 par Philip Hazel pour le service informatique de l'Université de Cambridge.

F

- **FAI** (Fournisseur d'accès à Internet) :
  Est un organisme (généralement une entreprise) offrant une connexion au réseau informatique Internet. Le terme anglais désignant un FAI est *Internet Service Provider* (ISP).
- **FDDI** (Fiber Distributed Data Interface):
  Est un type de réseau informatique LAN ou MAN permettant d'interconnecter plusieurs LAN à une vitesse de 100 Mbit/s sur de la fibre optique (ce qui lui permet d'atteindre une distance maximale de 200 km).
- **FEROS** (Fiche d'Expression Rationnelle des Objectifs de Sécurité des Systèmes d'information) :
  Est une méthode d'évaluation des risques développée par la Direction Centrale de la Sécurité des Systèmes d'Information.
- **FTP** (File Transfer Protocol):
  Est un protocole de communication dédié à l'échange informatique de fichiers sur un réseau TCP/IP.
- **FTPS** (Secure File Transfert Protocol) **:**
  Il permet au visiteur de vérifier l'identité du serveur auquel il accède grâce à un certificat d'authentification. Il permet également de chiffrer la communication.
- **FXP** (File Exchange Protocol):
  Est une variante non officielle du protocole FTP. Il permet le transfert direct de fichiers entre deux serveurs FTP sans passer par le client.

H

- **HLU**(Harrison, Ruzzo, Ullman model) :
  Modèle de sécurité qui fournit des politiques pour des droits d'accès changeants et des droites pour la création et la suppression des sujets et des objets. On le considère généralement comme l'un des modèles plus complexes de sécurité.

- **HTML** (Hyper Text Mark-Up Language):
  Est le langage informatique qui sert à écrire les pages Web.

I

- **ID** (Identificateur)
- **IDS** (Intrusion Detection System) :
  Est un mécanisme destiné à repérer des activités anormales ou suspectes sur la cible analysée (un réseau ou un hôte). Il permet ainsi d'avoir une action de prévention sur les risques d'intrusion.
- **IEEE** (Institute of Electrical and Electronics Engineers):
  Est l'organisation professionnelle qui compte le plus de membres, et possède différentes branches dans plusieurs parties du monde. Elle est constituée d'ingénieurs électriciens, d'informaticiens, de professionnels du domaine des télécommunications, etc.
  L'organisation a pour but de promouvoir la connaissance dans le domaine de l'ingénierie électrique.
- **IMAP** (Internet Message Access Protocol) :
  Est un protocole utilisé par les serveurs de messagerie électronique, fonctionnant pour la réception.
  Ce protocole permet de laisser les e-mails sur le serveur dans le but de pouvoir les consulter de différents clients e-mails ou web mail.
- **IPCE** (Interprocessus Communication Environnement) :
  Regroupent un ensemble de mécanismes permettant à des processus concurrents (ou distants) de communiquer. Ces mécanismes peuvent être classés en trois catégories :
  - les outils permettant aux processus de s'échanger des données ;
  - les outils permettant de synchroniser les processus, notamment pour gérer les sections critiques ;
  - les outils offrant directement les caractéristiques des deux premiers (ie : permettant d'échanger des données **et** de synchroniser des processus).
- **IPv4** (Internet Protocol version 4) :
  Est la première version d'IP à avoir été largement déployée, et forme encore la base (en 2007) de l'Internet.
  Elle utilise une adresse IP sur 32 bits, ce qui est un facteur limitant à l'expansion d'Internet puisque "seulement" 4 294 967 296 adresses sont possibles. Cette limitation conduit à la transition d'IPv4 vers IPv6, actuellement en cours de déploiement, qui devrait progressivement le remplacer.
- **IPv6** (Internet Protocol version 6) :
  Est le successeur du protocole IPv4 ; mais IPv4 est encore utilisé presque exclusivement sur Internet.
- **IRTF** (Internet Research Task Force)
- **ISO** (International Standards Organization) :
  Est un organisme de normalisation international composé de représentants d'organisations nationales de normalisation de 158 pays. Cette organisation crée en 1947 a pour but de produire des normes internationales dans les domaines industriels et commerciaux appelées normes ISO.

L

- **LAN** (Local Area Network) :
  Ce terme désigne un réseau informatique d'échelle géographique restreinte.
- **LWSP** (Linear White Space).

M

- **MAN** (Metropolitan Area Network) :
  Désigne un réseau étendu d'ordinateurs habituellement utilisé dans les campus ou dans les villes. Le réseau utilise généralement des fibres optiques .
- **Marion** (Méthode d'analyse de risques informatiques optimisée par niveau) :
  Une méthode d'audit, proposée depuis 1983 par le CLUSIF, visant à évaluer le niveau de sécurité informatique d'une entreprise. L'objectif est double:

  - situer l'entreprise auditée par rapport à un niveau jugé correct, et par rapport au niveau atteint par les entreprises similaires
  - identifier les menaces et vulnérabilités à contrer

- **Melisa** (Méthode d'évaluation de la vulnérabilité résiduelle des systèmes d'information) :
  Une méthode d'audit, proposée depuis 1985 par le CLUSIF, visant à évaluer le niveau de sécurité informatique d'une entreprise.
- **Méhari** (Méthode harmonisée d'analyse des risques) :
  A été développée et est proposée par le CLUSIF, visant à la sécurisation informatique d'une entreprise ou d'un organisme.
- **MIME** (Multipurpose Internet Mail Extensions) **:**
  Est un standard internet qui étend le format de données des courriels pour supporter des textes en différents codage de caractères autres que l'ASCII, des contenus non textuels, des contenus multiples, et des informations d'en-tête en d'autres codages que l'ASCII. Les courriels étant généralement envoyés via le protocole SMTP au format MIME, ces courriels sont souvent appelés courriels *SMTP/MIME*.
- **MIT** (Massachusetts Institute of Technology) :
  Est une institution de recherche et une université américaine, spécialisée dans les domaines de la science et de la technologie.
- **MMF** (Make money fast)
- **MSSP** (Managed Security Services Provider):
  Une société extérieure ou une consultation qui fournissent la conception de sécurité, le développement et les services de gestion.
- **MTA** (Mail Transfert Agent) :
  Est un agent de programme machine ou de logiciel qui transfère des messages de courrier électronique à partir d'un ordinateur à l'autre
- **MUA** (Mail User Agent) :
  Un client de messagerie est un logiciel qui sert à lire et envoyer des courriers électroniques. Les protocoles de communication utilisés pour la réception sont POP ou

IMAP. Le protocole SMTP est utilisé par la plupart des serveurs de courriel pour envoyer les messages.

N

- **NAT (Network Address Translation)** :
C'est un mécanisme qui est implémenté sur un routeur, permet de faire correspondre les adresses IP internes non-unique et souvent non routables d'un domaine vers un ensemble d'adresses externes uniques et routables. Ce mécanisme permet de pallier la carence d'adresses IPv4 sur Internet, le protocole IPv6 dispose d'un espace d'adressage plus important.

O

- **OMT** (Object Modeling Technique) :
Ce langage est l'œuvre de James Rumbaugh, qui est aussi l'un des pères d'UML (avec Grady Booch, fondateur du langage Booch, et Ivar Jacobson, fondateur du langage OOSE).
- **OOSE** (Object Oriented Software Engineering):
Est un langage de modélisation objet, comme OMT et Booch, avec lesquels il a fusionné pour donner UML.
- **OpenBSD** (Berkeley Software Distribution):
Désigne en informatique une famille de systèmes d'exploitation Unix, développés à l'université de Berkeley par un groupe de programmeurs : Bill Joy, Marshall Kirk McKusick, etc.
- **OQLF** (Office québécois de la langue française) :
est une institution publique du Québec créée le 24 mars 1961, au même moment que le Ministère des affaires culturelles du Québec (aujourd'hui le Ministère de la culture et des communications du Québec).
- **OSI** (Open System Interconnection):
Le modèle d'interconnexion en réseau des systèmes ouverts de l'ISO est un modèle de communications entre ordinateurs. Il décrit les fonctionnalités nécessaires à la communication et l'organisation de ces fonctions.
- **OSSIR** (Observatoire de la Sécurité des Systèmes d'Information et des Réseaux) :
Est une association française existant depuis 1996 et qui regroupe les utilisateurs intéressés par la sécurité des systèmes d'information et des réseaux.

P

- **PC** (Personal Computer) :
Le terme PC est aujourd'hui aussi utilisé pour désigner un ordinateur personnel, par opposition aux ordinateurs de tailles différentes.
- **PDA** (Personal Digital Assistant) :
Comme les agendas électroniques et les Ordinateurs de poches.
- **PKI** (Public Key Infrastructure) :
Est un ensemble de composants physiques (des ordinateurs, des équipements cryptographiques ou des cartes à puces), de procédures humaines (vérifications, validation) et de logiciels (système et application) en vue de gérer le cycle de vie des certificats numériques ou certificats électroniques.

- **POP** (Post Office Protocol) :
  Un protocole qui permet de récupérer les courriers électroniques situés sur un serveur
  De messagerie électronique. Cette opération nécessite une connexion à un réseau TCP/IP. Le port utilisé est le 110 .

R

- **RBL** (Real-time block list):
  Est une liste avec un format standard, elle contient des IP et Domaine suspect ainsi que des ID d'utilisateur (expéditeurs).
- **RNIS** (Réseau Numérique à Intégration de Services) :
  Réseau entièrement numérique dont le débit est plus important que celui des lignes téléphoniques classiques.
- **RSSI** (Responsable de la sécurité des systèmes d'information).

S

- **SAGE** (schéma d'aménagement et de gestion des eaux) :
  Est une unité hydrographique ou un système aquifère les grandes orientations définies par le SDAGE.
- **SBL** (Spamhaus Block List) :
  Est une base de données en temps réel des adresses IP des sources vérifiées de Spam et des opérations de Spam.
- **SDC** (System Development Corporation):
  Basé en Santa Monica, la Californie, était discutablement la première compagnie de logiciel d'ordinateur du monde.
- **SMTP** (Simple Mail Transfer Protocol) :
  Est un protocole de communication utilisé pour transférer le courrier électronique vers les serveurs de messagerie électronique
- **SNA** (Systems Network Architecture) :
  Est une architecture réseau en couches définie par IBM en 1974. SNA est désormais largement remplacé par TCP/IP
- **SOA** (Service-Oriented Architecture) :
  Est une forme d'architecture de médiation qui est un modèle d'interaction applicative qui met en œuvre des services (composants logiciels) :
  2- avec une forte cohérence interne (par l'utilisation d'un format d'échange pivot, le plus souvent XML).
  3- et des couplages externes « lâches » (par l'utilisation d'une couche d'interface interopérable, le plus souvent un service web.
- **SOAP** (Simple Object Access Protocol):
  Est un protocole de RPC orienté objet bâti sur XML.Il permet la transmission de messages entre objets distants, ce qui veut dire qu'il autorise un objet à invoquer des méthodes d'objets physiquement situés sur un autre serveur. Le transfert se fait le plus souvent à l'aide du protocole HTTP, mais peut également se faire par un autre

protocole, comme SMTP.

- **SONET** (Synchronous Optical Network) :
  Est un modèle de norme de transmission optique. C'est un protocole de couche 1 du modèle OSI.
- **SPARC** (Scalable Processor Architecture):
  Est une architecture d'ensemble d'instruction de microprocesseur de RISC à l'origine conçue en 1985 par Sun Microsystems.

- **SPIT** (Spam over Internet Telephony):
  Est le problème qui a suscité beaucoup d'attention des acheteurs et de commerçants. Quelques pandits ont commencé à se rapporter à lui. Comme les systèmes finis d'IP, (l'E-mail et d'autres applications d'Internet), sont susceptibles de l'abus par les parties malveillantes qui lancent des communications non sollicitées et non désirées. Les téléprospecteurs, et d'autres trompeurs de système de téléphone sont susceptibles de viser des systèmes de VoIP de plus en plus, en particulier si VoIP tend à supplanter la téléphonie conventionnelle.
- **SSH** (Secure Shell) :
  Est à la fois un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite toutes les trames sont chiffrées. Il devient donc impossible d'utiliser un sniffer pour voir ce que fait l'utilisateur. Le protocole **SSH** a été conçu avec l'objectif de remplacer les différents programmes rlogin, telnet et rsh.
- **SSI** (sécurité des systèmes informatiques) :
  Est l'ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires et mis en place pour conserver, rétablir, et garantir la sécurité de l'information et du système d'information.
- **SSII** (Sociétés de Services en Ingénierie Informatique, spécialisées en sécurité) :
  Est une société de services spécialisée en informatique.
- **SSL** (Secure Socket Layer) :
  Est un protocole de sécurisation des échanges sur Internet, développé à l'origine par Netscape.

T

- **TCP/IP** (Transmission Control Protocol/Internet Protocol) :
  La suite des protocoles Internet est l'ensemble des protocoles qui constituent la pile de protocole utilisés utilisée par Internet.
- **TCSEC** (Trusted Computer System Evaluation Criteria):
  Sont un ensemble de critères énoncés par le Département de la Défense, et permettant d'évaluer la fiabilité de systèmes informatiques centralisés.
- **TFTP** (Trivial File Transfer Protocol):
  Est un protocole simplifié de transfert de fichiers.
- **TLL** (Time To Live) :
  Le Time-to-live d'un paquet d'information transitant sur un réseau représente le nombre de secondes pendant lesquelles il est autorisé à passer dans les divers routeurs ponctuant sa route.
- **TLS** (Transport Layer Security) :
  Nouvelle nomination de SSL.

U

- **UDP** (User Datagram Protocol) :
  Un des principaux protocoles de télécommunication utilisé par Internet. Il fait partie de la couche transport de la pile de protocole TCP/IP.
- **UML** (Unified Modeling Language) :
  Un langage graphique de modélisation des données et des traitements.
- **USA** (United States of America)
- **USB** (Universal Serial Bus) :
  Est un bus informatique *plug-and-play* servant à brancher des périphériques informatiques à un ordinateur pour communiquer en série.
- **UTF-8** (Unicode Transformation Format-8) :
  Est un format de codage de caractères défini pour les caractères Unicode (UCS). Chaque caractère est codé sur une suite d'un à quatre octets. il a été conçu pour être compatible avec certains logiciels originellement prévus pour traiter des caractères d'un seul octet.
- **UUCP** (Unix to Unix Copy Protocol):
  Est un ensemble de programmes qui permettent à deux machines d'échanger des fichiers et d'exécuter des commandes sur la machine distante en passant par une ligne téléphonique (modem), mais aussi sur une couche TCP/IP (souvent à travers Ssh), voire via un câble série direct (null modem). Le mode modem reste cependant le cas de figure le plus utilisé.

V

- **VPN** (Virtual Private Network):
  Est une extension des réseaux locaux qui procure une norme de sécurité en télécommunications.

W

- **WAN** (Wide Area Network) :
  Un réseau informatique couvrant une grande zone géographique, typiquement à l'échelle d'un pays, d'un continent, voire de la planète entière. Le plus grand WAN est le réseau Internet.

X

- **XML** (Extensible Markup Language) :
  Est un langage informatique de balisage générique. Le World Wide Web Consortium (W3C), organisme qui promeut la compatibilité des technologies du World Wide Web en éditant des recommandations à valeur de standards industriels, recommande XML pour exprimer des langages de balisages spécifiques.

BIBLIOGRAPHIES.

1. **[CCNA1-1]**: "CCNA1", CISCO, Cisco Academy Program, V3.1.1.
2. **[ADR-2]** : "Architecture des réseaux", Didier Donzes, Université de valencienne, 2003
3. **[TINA-3]** : "TCP/IP Network administration", Graig HUNT, O'Reilly, 2000.
4. **[AMO-4]** : "Architecture Du Modèle OSI", Guy Pujolle, Université de Versailles.2002.
5. **[WPS-5]** : "Sécurité informatique", Wikipedia, http://fr.wikipedia.org ,2005.
6. **[SIN -6]** : "Sécurité informatique", www.vulnerability.com.2005.
7. **[SRS-7]** : "Sécurité réseaux et system", Jean François Pillot, DUNOD, 2006.
8. **[SAH-8]** : "Stratégies anti-hackers", Ryan Russell, Eyrolles. 2eme edition.2005.
9. **[VCE-9]** : "Vocabulaire du courriel électronique", Commission générale de Terminologie et de néologie, 22 aout 2003.
10. **[RFC-10]** : "Standard Pour le format des messages texte internet ARPA", DavidH.Crocker, Dept. EEUD, 13 Aout 1982.
11. **[WPC-11]** : "le courriel électronique", Wikipedia, http://fr.wikipedia.org/wiki/Courrier,2000.
12. **[WPP-12]** : "Le protocole SMTP", Wikipedia, http://fr.wikipedia.org/wiki/SMTP, 2001.
13. **[RFCf-13]** : "SIMPLE MAIL TRANSFER PROTOCOL", J.Postel/ISI, August 1982.fr.
14. **[RFCa-14]** : "SIMPLE MAIL TRANSFER PROTOCOL", J.Postel/ISI, August 1982.en.
15. **[SAM-15]** : "Linux Sendmail administration", Graig Hunt, Linux Edition, February 2001
16. **[DSP-16]** : "Definition of spam", http://www.mail-abuse.org.
17. **[GR-17]** : "Gartner Research", Joyce Graff& Maurene Grey, http://www.itap.purdue.edu/presentations,2003.
18. **[SMD-18]** : "Spam Definition", http://www.arobase.org/spam/comprendre.htm.
19. **[WST-19]** : "Terminologies de Spam", http://fr.wikipedia.org/wiki/Spam#Terminologie.
20. **[SC-20]** : "Le spam en chiffres", internet monde, http://www.journaldunet.com/cc/03 /spam.shtml.
21. **[SMH-21]** : "SendMail Historique", Redhat, http://www.europe.redhat.com/ Documentation/sendmail.php3.
22. **[WSM-22]** : "SendMail Definition", http://fr.wikipedia.org/wiki/Sendmail.

23. **[SAB-23]** : "Spam Assassin", Alan Schwartz, O'Reilly, July 2004.
24. **[MSI-24]** : "Messagerie d'internet", Jean Paul Gauthier & Philipe leca, UREC, 1999.
25. **[SAS-25]** : "Apache Project", Rudy Montier, http://www.supinfo projects.com/ Spamassassin, 2005.
26. **[PMG-26]** : "Proventia M10 User Guide", Internet Security System, Mars 2006.
27. **[HPW-27]** : "Honey Pot For Linux", Roger Grimes, Apress, June 2005.
28. **[HP-28]** : "Honey Pot", M Dizon, EIG, December 2003.
29. **[HPF-29]** : "Honey Pot Farms", Franck Veysset, EUROSEC, 2004.
30. **[HHN-30]** : "Honey Pot & Honey Net", http://www.authsecu.com/honeypots-honeynet.
31. **[LRH-31]** : "Linux Redhat Entreprise Edition", http://www.commentcamarche.net/linuxR.
32. **[SHP- 32]** : "Spamhaus Project", http://www.spamhaus.org/organization/index.lasso.