الجمـــــهوريـــــة الجـــزائريـــة الديـــمقراطيـــــــــة الشـــعبيـــــــة
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعـــلـيـــم الـعـالـــي و الـبـحـث الـعـلـــمي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
جـامـعـــة عـمّـــار ثـلـيـــجـــي بالأغـــــواط
UNIVERSITE AMAR TELIDJI LAGHOUAT

كـلـيـــــة الـعـلـــــوم
FACULTE DES SCIENCES

DEPARTEMENT DE MATHEMATIQUES ET INFORMATIQUE

Mémoire de MASTER

Domaine: Mathématiques et Informatique
Filière: Informatique
Option: Réseaux, Systèmes et Applications Réparties

Par :
M'hammed ACHOUR
Moussa GHERASLIA

THEME

L'implémentation d'un système de détection de fuite de données

Soutenu publiquement devant le jury composé de:

Mr. Tahar ALLAOUI	*M.A.(A)*	*Président*
Mr. Noureddine CHAIB	*M.A.(A)*	*Examinateur*
Mr. Lakhdar OULADDJEDID	*M.A.(A)*	*Examinateur*
Mr. Mohamed Lahcen BENSAAD	*M.C.(B)*	*Encadreur*

Année Universitaire 2014/2015

Remerciements

Nos remerciements vont tout d'abord à *ALLAH* pour tout ce qu'IL nous a donné.

Nos remerciements vont aussi à tous ceux qui nous ont aidé et soutenu dans notre travail, en particulier notre superviseur, Dr. Bensaad Lahcen, pour toute sa gentillesse, pour ses précieux conseils et pour sa patience, et vont aussi à CISSE Abdoul Aziz M. qui a nous aidé lors de la rédaction.

Et toutes nos reconnaissances sont adressées à nos parents, nos familles, nos amis pour leurs sacrifices et leur soutien, durant ces années d'études.

ملخص

اليوم، أصبح الدخول الى الأنترنت ضرورة بالنسبة للمؤسسات مما جعل مراقبة المعلومات الخارجة من المؤسسة وكذا حماية المعلومات المتبادلة تمثل تحديا كبيرا. من أصناف الأنظمة التي طورت لحماية المعلومات الحساسة نجد أنظمة منع تسرب المعلومات.

يتمثل عملنا في تطوير تقنية لكشف تسرب المعلومات في حدود المؤسسة أو في الضرورة خارج المؤسسة. يمكن اعتبار طريقتنا، التي تعتمد على مقارنة المحتوى، امتدادا لتقنيات منع تسرب المعلومات الحالية وبالتحديد في الموجهات سيئة السلوك (متحكم فيها عن طريق مهاجم). رغم أنها لا تعتمد على مبدأ التحقيق العميق في مقابل قاعدة معطيات المعلومات الحساسة.

الكلمات المفتاحية: المعلومات الحساسة، أنظمة منع تسرب المعلومات، مقارنة على أساس المحتوى، تسرب المعلومات، الموجهات سيئة السلوك، التحقيق العميق.

Résumé

Aujourd'hui, l'accès à Internet devient une nécessité pour les entreprises, ce qui fait du contrôle et de la protection des informations qui sortent de l'entreprise un énorme défi. Une des catégories des systèmes développés pour protéger les informations sensibles contre les fuites sont les systèmes DLP.

Notre travail présente le développement d'une technique de comparaison à base de contenu qui vise à détecter la fuite de données à l'extrémité et si nécessaire hors de l'entreprise. On peut considérer cette technique comme une extension des techniques DLP actuelles en particulier dans les routeurs malicieux (controlés par un attaquant). malgré qu'elle ne se base pas sur l'inspection approfondie par rapport à la base de données contenant les informations sensibles.

Les mots clés: les informations sensibles, les systèmes DLP, comparaison à base de contenu, la fuite de données, les routeurs malicieux, une inspection approfondie.

Abstract

Today, the access to Internet became a necessity for the enterprises which make the control of information outing the enterprise and the information exchanged protection as a big challenge. One of the systems categories which are developed for protecting the sensitive data against leaks are the DLP systems.

Our work present a development of a content-based comparison technique aims to detect the data leak at the enterprise extremity and if necessary out of the enterprise. We considered this technique like an extension of the actual DLP systems particularly in malicious routers (controlled by an attacker), in spite that it don’t base on deep inspection compared with a sensitive data base.

Key words: sensitive data, DLP systems, content-based comparison, data leak, malicious routers, deep inspection.

Sommaire

Table des figures

Chapitre 1

Introduction

1.1 Problématique

L'évolution d'Internet et l'apparition de nouveaux applications font de la sécurité dans cette énorme réseau un grand défi, incluant la protection des informations sensibles qui la traversent. Les systèmes de prévention de fuite de données ou les systèmes DLP (Data Leak Prevention) peut protéger l'entreprise contre les fuites internes. Cependant, dès que l'information sort de l'entreprise la protection de cette information ne concerne pas les systèmes DLP. Par conséquent, L'apparition des attaques sur l'infrastructure réseau, qui a touché les nouveaux routeurs programmables, a créé une menace externe pour la protection des informations échangées et les routeurs malicieux sont devenus un chemin d'acheminement pour celles-ci. Comment peut-on détecter la fuite si elle arrive dans un routeur situé dans le chemin d'information ?

1.2 Motivation

Les systèmes DLP corrigent le problème de la fuite de données à l'intérieur de l'entreprise. Mais l'apparition de nouvelles attaques sur l'infrastructure réseau a créé le problème de fuite de données à l'extérieur de l'entreprise que

les systèmes DLP actuels ne peuvent détecter, puisqu'elles sont construites pour fonctionner seulement à l'intérieur de l'entreprise.

L'existence des techniques de détection des routeurs malicieux, qui ne sont pas construites spécialement pour ce genre d'attaques, nous permet d'utiliser les approches suivies pour développer une technique de détection de fuite de données dans les routeurs à l'extrémité et hors de l'entreprise, ce qu'on peut considérer comme une extension des systèmes DLP à l'extérieur.

1.3 Les objectifs

Aujourd'hui, les informations sensibles des entreprises sont devenues une cible intéressante et font donc l'objet de plusieurs attaques. Ce phénomène rend la sécurité de celles-ci (les informations) un souci pour les entreprises. Ces dernières cherchent des techniques pour protéger ces informations contre les menaces intérieures et extérieures. Nous voulons avancer d'un pas vers l'extension des systèmes DLP pour les faire fonctionner à l'extérieur de l'entreprise. Pour faire cela, nous utilisons une technique basée sur la comparaison de contenus pour détecter les fuites de données dans les routeurs de réseau.

1.4 L'organisation du mémoire

On a divisé notre mémoire en quatre chapitres :

— Le premier chapitre est une introduction générale, où nous avons cité la problématique, les motivations et l'objectif que nous voulons atteindre par notre travail.
— Le deuxième chapitre présente une brève introduction et quelques définitions sur les systèmes DLP et les attaques des infrastructures.

Ensuite, on va parler des techniques centralisées et distribuées développées pour détecter les routeurs malicieux. Finalement, on va citer quelques travaux précédents et quelques discussions.

— Nous allons présenter dans le troisième chapitre la méthode suivie, les phases principales de celle-ci et quelques discussions concernant cette méthode.

— Dans le quatrième chapitre nous allons présenter notre application qu'on a développé pour implémenter l'algorithme proposé dans le chapitre précédent ainsi que leur manuel d'utilisation.

Chapitre 2

Notions généraux autour de la fuite de données.

Dans ce chapitre, on va aborder quelques notions sur les systèmes DLP existants qui détectent les fuites des données dans les systèmes finaux implémentés dans les stations de travail ou dans les passerelles des entreprises. Avec l'intégration de la pragrammabilité dans les dispositifs réseaux, de nouvelles formes d'attaques sur l'infrastructure réseau sont apparues incluant les fuites de données dans ces dispositifs. On vas présenter une vue brève sur les attaques infrastructurels dans la section suivante. Étant donné qu'il y a des techniques destinées à la détection et l'isolation des routeurs malicieux, à l'origine de fuites de données, on va présenter quelques notions générales sur ces techniques. Enfin, on va citer quelques travaux récents concernant ce domaine et quelques discussions.

2.1 Les réseaux

Un réseau est un ensemble d'équipements reliées entre eux, qui permettent d'échanger les informations numériques entre les terminaux (ordinateur, serveur, périphérique) via un support pour véhiculer les données (les câbles,

l'atmosphère, les fibres optiques). La liaison entre les différents réseaux et l'orientation des informations sont assurés par des dispositifs réseaux (concentrateur, commutateur, routeur, passerelle). Internet est l'exemple le plus concret du succès des réseaux dans le monde à laquelle la plupart des réseaux locaux ont accès [5], voir la figure 2.1 (pour voir plus d'informations consernant les réseaux consulter [6]) .

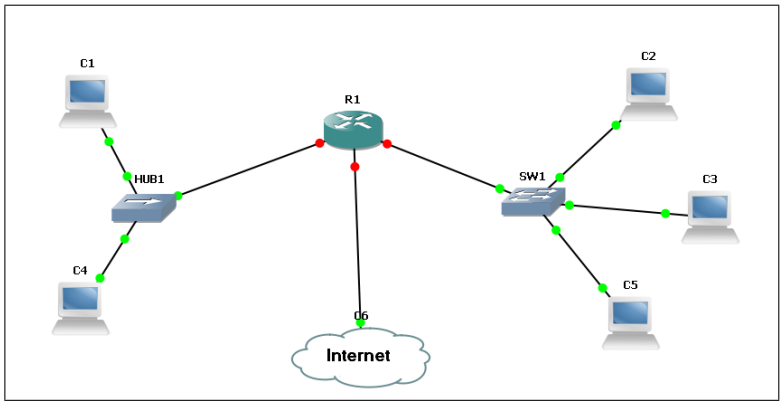

FIGURE 2.1 – Exemple d'un réseau informatique (réalisé par GNS3)

Dès que les attaques informatiques sont apparues, la sécurité a de plus en plus commencé à préoccuper les développeurs. Plusieurs solutions étaient proposées pour sécuriser les réseaux et les systèmes finaux comme les scanneurs, les pare-feux [7], les IDS (Intrusion Detection System) [8] et les systèmes DLP (Data Leak Prevention). Ces derniers sont développés surtout pour la détection des fuites des données, ils sont relativement nouveaux et sont considérés comme étant plus efficaces que les solutions précédentes. En effet, les IDS et les pare-feux peuvent ne pas fonctionner proprement si la fuite arrive spontanément à cause d'un utilisateur autorisé [9].

2.2 Les fuites des données

Une fuite de données est un transfert non autorisé des informations classifiées depuis un ordinateur ou un centre de données vers le monde externe. La fuite de données peut être accomplie simplement par la mémorisation de ce qui a été vu, par des amovibles magnétiques physiques, disques et rapports ou par des outils subtils (voir stéganographie) [10].

Pour les entreprises qui détiennent des informations stratégiques (données sur les clients, propriété intellectuelle, secrets professionnels, informations propriétaires, etc), le risque de fuites est plus élevé qu'il ne l'a jamais été [11].

Il est essentiel de comprendre pourquoi elle se produit. Les recherches de sociétés tierces sur les causes des fuites de données, collectées par l'Equipe Verizon chargée de la gestion des risques économiques et la Open Security Foundation, mettent en lumière trois types de causes : employés internes bien intentionnés, employés malveillants et attaques ciblées. Dans bon nombre de cas, les fuites sont causées par une combinaison de ces facteurs. Par exemple, les attaques ciblées sont souvent rendues possibles par des employés bien intentionnés qui oublient d'appliquer les politiques de sécurité, ouvrant ainsi la porte aux éventuelles fuites [11]. C'est pour protéger le système contre ce type de risque, que les systèmes DLP ont été développés.

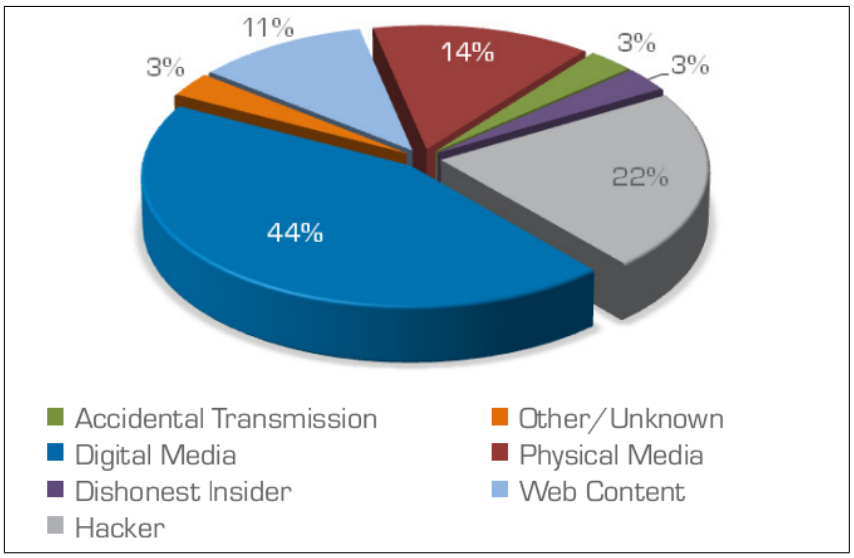

FIGURE 2.2 – La violation de données par type [1].

2.3 Les systèmes DLP (Data Leak Prevention)

La prévention de fuites de données est la catégorie de solutions qui aide une organisation à appliquer des contrôles pour la prévention des fuites d'informations sensibles indésirables accidentelles ou malicieuses vers une entité non autorisé dans ou hors de l'organisation [12].

Les systèmes de prévention des fuites de données ou les systèmes DLP sont spécialement développés pour l'identification, la surveillance, la détection et la prévention des fuites de données confidentielles [9] incluant des technologies qui facilitent l'atteinte de trois objectifs [13] :

— Localiser et cataloguer l'information sensible stockée dans l'entreprise.
— Surveiller et contrôler le mouvement des données sensibles à travers les réseaux de l'entreprise.
— Surveiller et contrôler le mouvement des données sensibles dans les systèmes finaux des utilisateurs.

Les données contrôlées par les systèmes DLP sont alors classifiées en trois classes (voir [14]) :

1. Les données en repos (data at rest) : sont tous les données stockées dans les stations de travail et les centres de données contenants à la fois les données sensibles (les numéros de cartes de crédits, clés de chiffrement, ...) et les données moins sensibles.
2. Les données en mouvement (data in motion) : ce sont les données sortantes de l'entreprise qui peuvent contenir des données sensibles injectées dans les emails ou n'importe quel type de transaction [15].
3. Les données utilisées (data in use) : ce sont les données manipulées par l'utilisateur, qui peut les copier dans une clé USB ou simplement les mémoriser dans la tête.

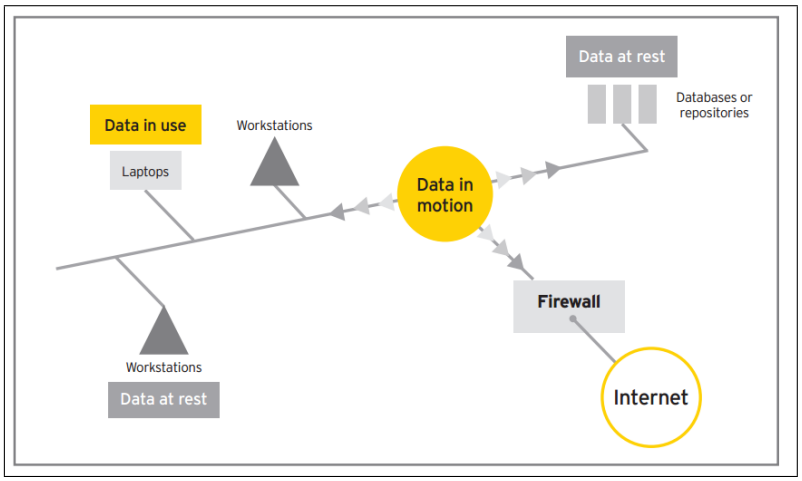

FIGURE 2.3 – La classification de données contrôlées par DLP [2].

Généralement, le système DLP scanne le stockage, intercepte le trafic et surveille les actions des utilisateurs. Tous ces trois sont réalisées dans une première étape à découvert ; dans une deuxième étape le système identifie les information confidentielles parmi les autres (c'est possible grâce à

plusieurs techniques. Chacune a ses avantages et inconvénients concernant la représentation des données sensibles. ex : les expression régulières, keywords, hachage, ...) et la dernière étape est l'application des politiques de sécurité de l'entreprise (crypter, bloquer, ...).

2.4 La vulnérabilité de nouveaux routeurs

L'augmentation des performances des processeurs multi-coeurs et le coût élevé de développement des routeurs ASIC (application-specific integrated circuits, voir [16]) et l'inflexibilité des circuits ASIC et les nouveaux applications réseaux (ex : garantir la QoS, nouveaux protocoles, ...) tous ces facteurs et d'autres nous amènent au développement de nouveaux types de routeurs basés sur les processeurs des réseaux (Network Processors [17]). La programmabilité de ces routeurs les rend plus flexibles et plus performants que les routeurs ASIC. Par conséquent, les mêmes vulnérabilités des systèmes finaux (end-systems) sont héritées dans les systèmes implémentés dans les routeurs programmables. Cependant, on ne peut pas utiliser les mêmes solutions de sécurité des systèmes finaux (scanneurs, HIDS, ...) pour les derniers à cause de la faible capacité de ces processeurs(voir [18]).

Jusqu'à récemment, l'infrastructure du réseau elle-même n'était pas une préoccupation majeure dans la sécurité des réseaux car elle ne présente pas une cible pratique d'attaque. Cependant, la technologie utilisée pour l'implémentation des routeurs des réseaux est changée dans les années récentes et des nouvelles vulnérabilités sont apparues [19]. Comme les routeurs ont un rôle vital dans l'infrastructure réseau, les routeurs programmables représentaient une cible attirante pour les attaquants.

Une fois l'attaquant arrive à compromettre le routeur, il peut faire des attaques destructives comme le dénie de services, destruction sélective des paquets, modification des paquets, etc.

2.5 Les techniques de détection des routeurs malicieux

Les méfaits élémentaires d'un routeur compromis sont limités aux actions suivantes [3] :

1. La destruction des paquets : un routeur compromis peut sélectivement détruire les paquets appartenant au flux qui le traverse.
2. La génération des paquets : le routeur peut injecter des paquets générés localement dans le flux de trafic qui le traverse.
3. La modification des paquets : le routeur peut modifier les paquets (contenus, entêtes).
4. La réordonnancement des paquets : ce qui pose des problèmes de performance.
5. Le retardement des paquets : un routeur compromis peut affecter la gigue du trafic multimédia.

Dès que le routeur est compromis, il peut faire les actions précédentes et la fuite de données devient possible dans celui-ci.

Plusieurs techniques pour détecter et isoler les routeurs malicieux ont été développées. Ces techniques, dont le but principal est de trouver un chemin qui ne contient aucun routeur compromis entre la source et la destination, exploitent un des deux approches suivantes :

1. La détection centralisée : chaque routeur est associé à un détecteur qui surveille le trafic entrant et sortant. Le détecteur capture les flux entrants

et sortants du routeur surveillé et prend sa décision localement en basant sur un algorithme de comparaison bien défini, s'il y a une divergence, une alerte sera lancée [20].

2. La détection distribuée : à la place du détecteur centralisé, la plupart des systèmes approximent ce service via un protocole distribué qui consiste à la participation des bons routeurs, les routeurs ont une prédiction de comment normalement le trafic va être acheminé. Ainsi, dès que le trafic traverse les bons routeurs voisins, on aura un nombre suffisants d'observations pour détecter un comportement négatif [20].

Ces techniques sont basées sur la validation de trafic. En effet, la description la plus précise du trafic est lui-même : le contenu exact des paquets [19]. La validation de trafic peut se faire de trois façons :

Par routeur : dans cette architecture on valide l'ensemble de tous les flux entrants depuis toutes les interfaces de routeur avec les flux sortants (la figure 2.4).

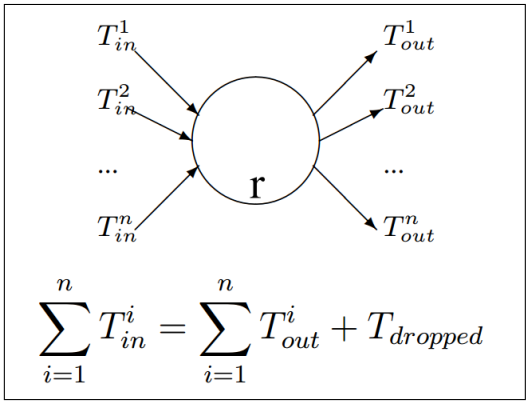

FIGURE 2.4 – La conservation de trafic par routeur [3].

Par interface : pour chaque interface, on valide le trafic entrant ou sortant

avec tous les trafics sortants ou entrants respectivement des autres interfaces (la figure 2.5).

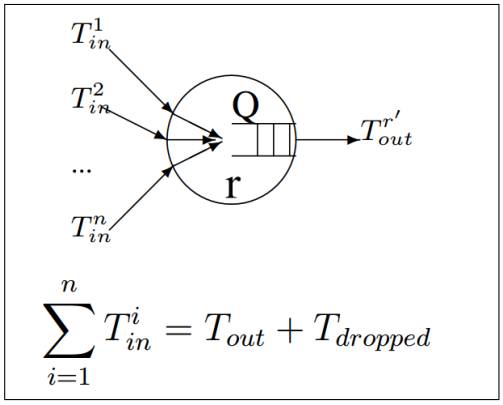

FIGURE 2.5 – La conservation de trafic par interface [3].

Par segment : les extrémités d'un segment manipulent ce type de validation, la plupart des techniques de détection des routeurs malicieux sont basés sur cette architecture, les routeurs testés sont les routeurs intermédiaires (la figure 2.6).

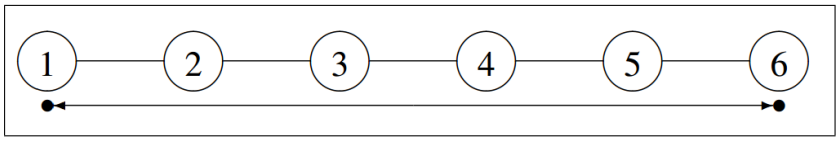

FIGURE 2.6 – La conservation de trafic par segment [3].

Plutôt que de faire une validation complète du trafic qui n'est pas pratique surtout avec l'approche distribué, on valide une des propriétés de trafic suivants :

La conservation de flux : valide le volume de trafic, détecte la destruction des paquets.

La conservation de contenu : valide le contenu de trafic, détecte la modification des paquets.

La conservation d'ordre : valide l'ordre de paquets, détecte la réordonnancement des paquets.

La conservation de temps : valide le délai des paquets, détecte le retardement des paquets.

2.6 L'état de l'art

Il est très difficile de détecter et minimiser l'impact d'un routeur particulier ou un groupe de routeurs qui sont compromis. Dans les réseaux actuels, où le trafic est véhiculé à travers plusieurs routeurs qui appartiennent aux différents vendeurs administratifs, détecter un routeur malicieux et minimiser son impact dans différents protocoles réseaux est un défi [4]. On va citer deux techniques distribuées et deux autres techniques centralisées concernant ce problème.

WATCHERS : (Watching for Anomalies in Transit Conservation : a Heuristic for Ensuring Router Security) [21] est une approche distribuée de surveillance réseaux. Un routeur malicieux est défini comme celui qui détruit les paquets, ou ce qui ne respecte pas le protocole, WATCHERS exploite la conservation de flux avec une architecture de validation par routeur pour détecter les routeurs malicieux.

SecTrace : (Secure Traceroute to Detect Faulty or Malicious Routing) [22] était développé pour être un outil pratique pour tracer de manière sécurisée le chemin du trafic depuis la source vers la destination. SecTrace valide la propriété de conservation de contenu de manière distribuée, si le routeur source détecte une divergence dans le trafic, un problème est détecté et une alerte sera lancé. Il ya d'autres travaux distribués, citons [23], [24].

La comparaison avec un routeur réplique : Des autres techniques qui se basent sur l'approche centralisé peuvent détecter les routeurs malicieux.

Le détecteur centralisé via réplication active [20] vérifie le comportement du routeur par la livraison d'une copie du trafic vers un routeur identique du routeur contrôlé et vérifie si les deux flux sortants du routeur et de son réplique sont identiques.

La détection à base de contrôleur : R. Patil, M. P. Tahiliani ont proposé une technique qui détecte les modifications malicieuses des paquets à base de contrôleur [4]. Le routeur à surveillé est connecté avec ses voisins par un commutateur (switch) administrable (voir la figure 2.7). Le noeud contrôleur n'appartient pas au chemin de trafic. Tout le trafic entrant et sortant du routeur peut être capturé par le contrôleur. C'est possible grâce à un port miroir dans le commutateur. Dans un temps particulier "t", le contrôleur surveille le comportement d'acheminement du routeur. Puis, le contrôleur fait une comparaison à base de hache dans les paquets reçus pour détecter s'il y a une modification [4].

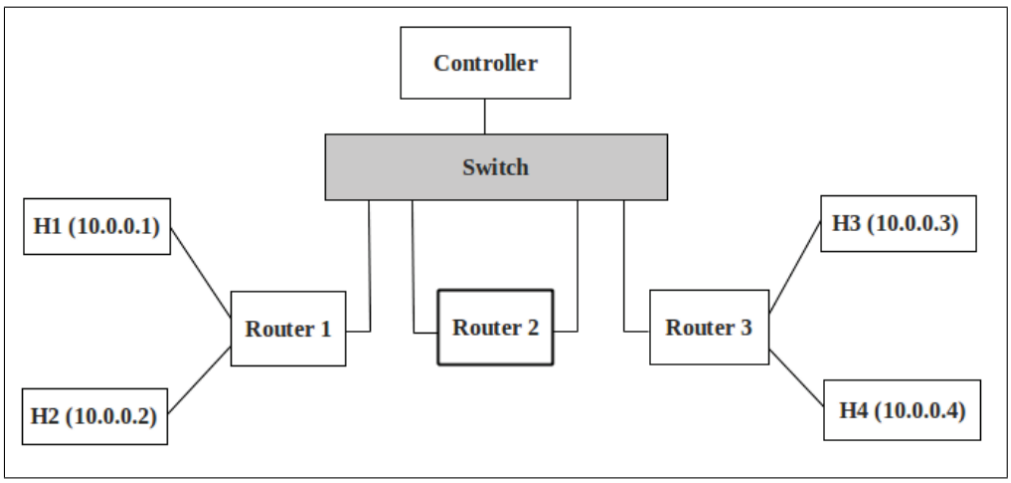

FIGURE 2.7 – La détection de la modification malicieuse des paquets [4].

2.7 Discussion

Généralement, toutes les techniques distribuées pour détecter des routeurs malicieux exige les hypothèses suivantes pour fonctionner :

1. Un chemin correct entre les bons routeurs : les techniques distribuées imposent au moins un chemin correct entre tout pair de bons routeurs.
2. Des bons routeurs finaux : le routeur source et la destination sont considérés comme des bons routeurs. Ainsi, le problème des passerelles malicieux n'est pas résolu par ces techniques.
3. Les outils cryptographiques et la distribution des clés : certains protocoles exigent la cryptographie dans son mode de fonctionnement ce qui exige un algorithme pour la distribution des clés.

En plus de ces hypothèses chaque protocole impose ses hypothèses propres (ex : WATCHERS impose des contraintes sur les nombres des bons routeurs).

Le détecteur centralisé via réplication active est considéré comme une approche non pratique à cause de ses limitations :

Implémentation complexe : la synchronisation entre le routeur et son réplique est un défi. En effet, un petit décalage entre les temps de mise à jour des tables de routages influe sur les trafics sortants des routeurs.

Exigences des ressources : cette technique est très chère en terme de ressources. En effet, l'implémentation d'un routeur réplique pour chaque routeur est très couteuse.

L'approche proposé dans [4] détecte la modification des entêtes des paquets en exploitant la conservation de contenu. Mais, Les hypothèses proposées ne sont pas pratique comme l'élimination des autres fonctionnalités du routeur,

ce qui n'est pas toujours le cas dans un environnement réel, et la supposition que le routeur ne détruit pas les paquets.

Le but principal des techniques précédentes est de trouver un chemin correct entre la source et la destination pour acheminer les informations, ce qui n'est pas vraiment le cas dans notre situation : savoir si un routeur sous nos mains fuit les données ou non. Cependant, ces techniques ont essayé de résoudre les problèmes ayant un effet direct sur la fuite de données en exploitant des approches qui peuvent être utile dans notre situation.

Chapitre 3

Algorithme implementé

Le contrôle du flux entrant et sortant du routeur est une méthode très efficace pour la détection du mauvais comportement des routeurs malicieux, car le comportement usuel et prédictible. Nous allons suivre la même approche pour la détection des fuites de données dans les routeurs par l'analyse du trafic trouvé dans le trafic sortant qu'on ne trouve pas dans le trafic entrant.

3.1 Terminologie

Flux entrant du routeur : c'est le flux qui entre dans le routeur depuis toutes ses interfaces.

Flux sortant du routeur : c'est le flux qui sort du routeur à partir de toutes ses interfaces.

Le flux additif : c'est la différence entre le flux sortant du routeur et le flux entrant.

Seuil : c'est un nombre entre 0 et 100 définie par l'utilisateur pour déterminer les flux qui ont un sens.

Flux a un sens : c'est un flux qui appartient au flux additif envoyé vers une adresse, destination précise. Ce flux contient une partie d'un flux

associé à une adresse ou pair d'adresses présent dans le flux entrant du routeur.Cette partie représente un pourcentage supérieure ou égale au seuil choisi par l'utilisateur.

Flux entrant d'une adresse "adr" : ce sont tous les paquets dont l'adresse destination est "adr".

Flux sortant d'une adresse "adr" : ce sont tous les paquets dont l'adresse source est "adr".

Flux source-destination : c'est le flux échangé entre une paire d'adresses.

Flux entrant/sortant d'une adresse "adr" : ce sont tous les paquets dont l'adresse destination ou source est "adr".

3.2 Les hypothèses

1. L'existence d'une technique pour capturer le flux entrant et le flux sortant de toutes les interfaces du routeur contrôlé (on peut utiliser la technique mentionnée dans [4]).
2. L'élimination des paquets de contrôles (mise à jour de table de routage, paquet de découvert, ...) avant d'appliquer la méthode.

3.3 Les phases principales

3.3.1 La phase de capture :

Pour une période "t", on ajoute chaque paquet qui entre dans le routeur au flux entrant et chaque paquet sortant du routeur au flux sortant. Après l'expiration de "t" on construit le flux additif $Flux_a$ comme suit :

$$Flux_a = Flux_s - Flux_e$$

Tel que :

$Flux_s$: Flux sortant du routeur.

$Flux_e$: Flux entrant du routeur.

3.3.2 La phase d'analyse :

Dans cette phase, on analyse pour chaque adresse destination "Y" dans le flux additif si le trafic entrant de cette adresse à un sens. Pour atteindre ce but il faut calculer les valeurs $P_{X,Y}$ (pourcentages) pour chacune de celles-ci tel que "X" est une adresse dans le flux entrant.

On distingue quatre type d'analyses, un pour chaque type de flux ; analyse entrant, analyse sortant, analyse entrant/sortant et analyse source-destination.

L'analyse entrant : pour chaque adresse destination "X" dans le flux entrant et pour chaque adresse destination "Y" ($Y \neq X$) dans le flux additif on calcule :

$$P_{X,Y} = 1 - \frac{|DiffData(FluxE_e(X),FluxE_a(Y))|}{|FluxE_e(X)|}$$

On prend le flux entrant de "Y" comme un flux qui a un sens si : $P_{X,Y} \geqslant \frac{seuil}{100}$.

La valeur $P_{X,Y}$ représente le complément de pourcentage de flux $FluxE_e(X)$ qui n'existe pas dans $FluxE_a(Y)$. Ainsi, $P_{X,Y}$ représente le pourcentage de $FluxE_e(X)$ existé dans $FluxE_a(Y)$.

Tel que :

$FluxE_e(X)$: flux avec une adresse destination X qui se trouve dans le flux entrant du routeur.

$FluxE_a(X)$: flux avec une adresse destination X qui se trouve dans le flux additif.

DiffData() : différence à base de contenu (si p et q deux paquets, on dit que p égale à q si p.Data=q.Data).

L'analyse sortant : pour chaque adresse source "X" dans le flux entrant et pour chaque adresse destination "Y" ($Y \neq X$) dans le flux additif on calcule :

$$P_{X,Y} = 1 - \frac{|DiffData(FluxS_e(X),FluxE_a(Y))|}{|FluxS_e(X)|}$$

On prend le flux entrant de "Y" comme un flux a un sens si : $P_{X,Y} \geqslant \frac{seuil}{100}$

Tel que :

$FluxS_e(X)$: flux avec une adresse source X qui se trouve dans le flux entrant du routeur.

L'analyse entrant/sortant : pour chaque adresse "X" dans le flux entrant et pour chaque adresse destination "Y" ($Y \neq X$) dans le flux additif on calcule :

$$P_{X,Y} = 1 - \frac{|DiffData(FluxES_e(X),FluxE_a(Y))|}{|FluxES_e(X)|}$$

On prend le flux entrant de "Y" comme un flux a un sens si : $P_{X,Y} \geqslant \frac{seuil}{100}$

Tel que :

$FluxES_e(X)$: flux avec une adresse source ou destination X qui se trouve dans le flux entrant du routeur.

L'analyse source-destination : pour chaque paire source-destination d'adresses "XZ" dans le flux entrant et pour chaque adresse destination "Y" dans le flux sortant tel que $Y \neq X$ et $Y \neq Z$ on calcule :

$$P_{X,Y} = 1 - \frac{|DiffData(Flux_e(XZ),FluxE_a(Y))|}{|Flux_e(XZ)|}$$

On prend le flux entrant de "Y" comme un flux a un sens si : $P_{X,Y} \geqslant \frac{seuil}{100}$

Tel que :

$Flux_e(XY)$: flux avec une pair d'adresse source-destination XY ou YX qui se trouve dans le flux entrant du routeur.

3.3.3 La phase de décision :

Après avoir appliqué les deux phases précédentes sur plusieurs échantillons on peut décider si le routeur fuit les données ou non. On peut laisser la décision concernant l'arrivée d'une fuite de données à l'utilisateur, puisque c'est lui qui a choisi le seuil au début pour tous les analyses.

Quel que soit l'approche suivie par l'utilisateur pour la décision, si la période de capture est grande et si le seuil choisi est important et avec plusieurs analyses et l'apparition d'une adresse précise dans la plupart des résultats, alors la probabilité que cette adresse soit une adresse d'un attaquant est très grande.

3.4 Algorithme

3.4.1 Les fonctions et les procédures :

1. listAdrSrc(Flux X) : retourne la liste des adresses sources présentes dans le flux X.
2. listAdrDest(Flux X) : retourne la liste des adresses destinations présentes dans le flux X.
3. listAdrSrcDest(Flux X) : retourne la liste des paires d'adresses source-destination présentes dans X tel que $xy = yx$.
4. listAdr(Flux X) : retourne la liste des adresses présentes dans le flux X.
5. Existe(Flux X, Data d) : retourne vrai s'il existe une paquet dans X dont le contenu égale à d et faux sinon.
6. supprimerPaquet(Flux X, Paquet p) : supprime le paquet p qu'existe dans le flux X.
7. supprimerData(Flux X, Data d) : supprime le premier paquet p qu'existe dans le flux X où : p.Data=d.

8. flux(Adresse X, Chaine S, flux f) :retourne le flux de type S associe a l'adresse/paire d'adresses X qui (flux retourné) appartient à f.
9. taille(Flux f) : retourne le nombre des paquets présents dans le flux f.

3.4.2 Initialisation :

```
analyse entrant:                liste:= listAdrDest(Flux X)
                                S:="entrant"
analyse sortant:                liste:= listAdrSrc(Flux X)
                                S:="sortant"
analyse entrant/sortant:        liste:= listAdr(Flux X)
                                S:="entrant/sortant"
analyse source-destination:     liste:= listAdrSrcDest(Flux X)
                                S:="source-destination"
```

3.4.3 Pseudo code :

```
F, f: flux;
liste: liste d'adresses;
i, j, n: entier;
Pourcentage[][]: matrice de reels;
j:=1
Pour X: Adresse dans liste
    F:=flux(X,S,Flux_entrant_routeur);
    i:=1;
    Pour Y: Adresse dans listeAdrDest(Flux_additif)
        f:= flux(Y, "entrant", Flux_additif);
        n:=0;
        Pour p: Paquet dans F
            Si Existe(f,p.payload) alors
                n:=n+1;
                SupprimerPaquet(F,p);
                SupprimerData(f,p.payload);
            FinSi
        FinPour
        F:=flux(X,S,Flux_entrant_routeur);
        Pourcentage[i][j]:=n/taille(F);
        i:=i+1;
    FinPour
    j:=j+1;
FinPour
```

3.5 Discussions

3.5.1 Le seuil :

Normalement, nous travaillons sur l'idée que l'attaquant veut s'approprier totalement un flux, mais on a donné le choix à l'utilisateur pour prendre le seuil dont seulement le dépassement est détecté, à cause des raisons suivantes :

1. La différence entre le flux entrant et sortant du routeur même si le routeur ne fuit pas les données. En effet, les paquets présents dans le routeur avant le lancement de capture sont ajoutés au flux sortant et les paquets restés dans le routeur après l'expiration de temps de capture ne sont pas considérés dans le flux sortant.
2. Si le routeur est malicieux, il peut changer le flux qui aura fui. Par conséquent, s'il n'y a pas de seuil la fuite ne sera pas détectée.
3. On n'est pas sure que l'attaquant va totalement prendre un flux même si c'est vrai dans la majorité des cas.

Si on prend un grand seuil, l'effet des deux premiers éléments précédents sera diminué. D'autre part, si on prend un petit seuil les attaques mentionnées dans le troisième élément seront détectés.

3.5.2 La période de capture :

La période de capture a un effet très important sur les résultats obtenues et peut rendre le seuil sans intérêt. En effet, le choix d'une courte période augmente l'influence de la différence entre le flux entrant et le flux sortant capturés sur les résultats (si t_s est le temps consommé pour que le routeur serve toutes les files d'attentes si elles sont pleines, pour pallier ce problème on peut prendre une période de capture t de tel sorte que le rapport t_s/t sera

négligeable, si t_s est très grand ou les files d'attente ont une taille importante, on peut pendre la taille des files d'attente qui a un taux d'apparence élevé pour déterminer le temps de service t_s). En plus, le flux entrant capturé dans cette période sera de petit volume ce qui rend la similitude de contenue entre les flux appartenant à celui-ci plus grand.

3.5.3 Les faux :

L'analyse dans la méthode suivie est basée sur le contenu par paquet (l'ordre et la continuité ne sont pas prises en compte) ce qui rend les résultats incertains, par exemple si on a obtenu que l'adresse "X" a pris 70% du flux entrant de l'adresse "Y", ce ne peut être pas vrai car ce pourcentage peut être assemblée par plusieurs autre flux ayant des intersections avec le flux de "Y" que "X" a les prises. Dans ce cas, on est sure que "X" prend 70% du flux entrant de "Y", mais on est pas sure que "X" prend ce pourcentage totalement à partir de "Y". Malgré tout, ce qui est dit précédemment n'est pas un problème surtout avec la répétition de la phase d'analyse sur plusieurs échantillons avant la décision.

3.5.4 Les routeur malicieux :

Le routeur malicieux peut détruire, modifier et générer les paquets, ce qui apparait parfois comme une fuite de données et peut aussi gêner la phase d'analyse, mais à l'aide de plusieurs essais on peut différencier entre les deux cas.

3.5.5 La comparaison :

Pendant notre recherche, on a pas trouvé une méthode développée spécialement pour la détection des fuites dans les routeurs. Ainsi, on ne peut pas faire une comparaison précise basée sur les tests pratiques. Mais, il y a

plusieurs techniques qui détectent les mauvais comportements des routeurs, qu'on peut considérer la fuite de données comme étant une de ceux-ci. On remarque que ces techniques incisent quelques conditions qui ne sont pas toujours disponibles, citons par exemple l'hypothèse que les routeurs finaux (passerelles) doivent être en bonne état. Cependant, la plupart des attaques des fuites des données ciblent ce type de routeurs en particulier, ce qui rend le risque de fuite de données toujours persistante, malgré l'utilisation de ces techniques. En effet, ces techniques ne sont pas satisfaisantes car ils ne font que l'isolation de réseaux qui se trouvent derrière les passerelles malicieux.

D'autre part, la combinaison de notre méthode pour détecter les fuites des données sur les passerelles, et les techniques précédentes sur les routeurs situés dans le cœur de réseau, peut être une forte défense contre la fuite de données dans le réseau globalement. Ce n'est pas dire que notre méthode ne fonctionne que dans les passerelles. En effet, on peut l'exploiter dans tous les routeurs au long du réseau.

Chapitre 4

La réalisation de l'algorithme

Dans ce chapitre, nous allons tout d'abord décrire les outils utilisés, à savoir le langage Java et l'analyseur Wireshark. On va décrire brièvement l'implémentation qu'on a proposée pour l'algorithme décrit dans le chapitre précédent, les classes utilisées dans notre application et leurs interfaces.

4.1 Les outils

4.1.1 Le langage de programmation :

On a choisis pour notre implémentation le langage Java, ce dernier est un langage de programmation informatique orienté objet. La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitations tels que UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. Pour cela, divers plateformes et frameworks associés visent à guider, sinon garantir, cette portabilité des applications développées sur Java [25] [26].

Nous avons utilisé comme environnement de développement l'éditeur NetBeans version 7.1 sous la distribution Ubuntu de linux version 14.04.

4.1.2 L'outil de capture :

Pour simuler la capture des paquets entrants et sortants du routeur, on a utilisé le sniffer Wireshark, ce dernier est un analyseur de paquets des réseaux. Un analyseur de paquet de réseaux essaye de capturer les paquets des réseaux et essaye de montrer les données des paquets de façon détaillé autant que possible [27]. La figure 4.1 présente un flux capturé par Wireshark.

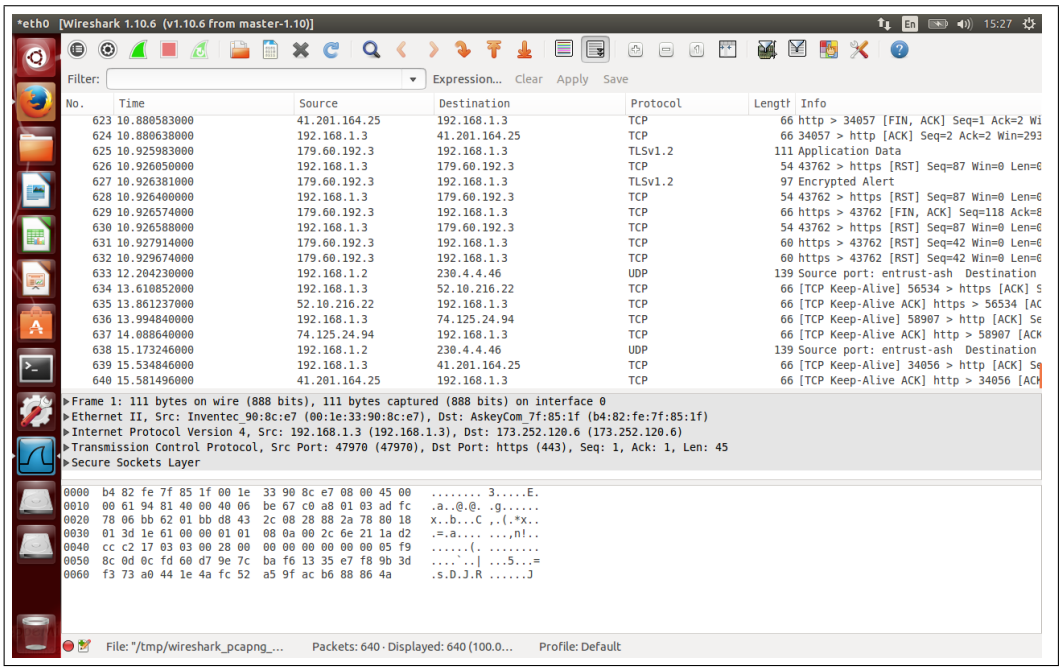

FIGURE 4.1 – Exemple de flux capturé par WireShark

4.2 Les interfaces

On a mentionné dans le chapitre précédent que l'algorithme suivie pour la détection des fuites a trois phases ; la phase de capture, la phase d'analyse et la phase de décision.

4.2.1 La phase de capture :

pour implémenter la première phase on a utilisé le l'outil de capture Wireshark, après la capture il suffit d'appuyer sur **Ctrl+S** pour obtenir le fichier texte contenant le flux capturé, le fichier obtenu est vu comme dans la figure 4.2.

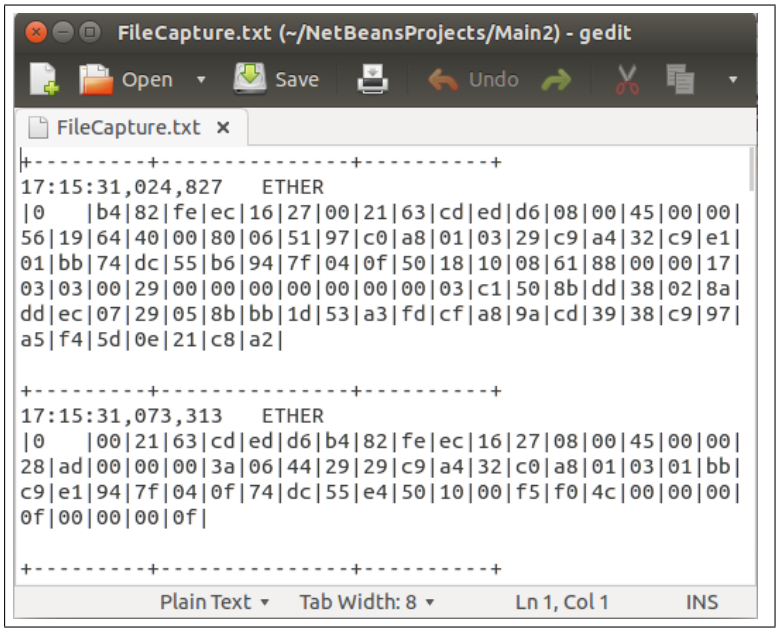

FIGURE 4.2 – Exemple de fichier capturé par WireShark

4.2.2 La phase d'analyse :

pour cette phase, on a développé une application de quatre fenêtres ; la première fenêtre est la fenêtre qui présente les actions principales : générer le flux sortant, scanner les deux flux entrant et sortant. La deuxième fenêtre a comme but de créer le flux entrant depuis le fichier de flux capturé et génère le flux sortant à partir du flux entrant. En effet, on n'a pas actuellement un flux sortant indépendant du flux entrant et qui représente une fuite réelle. Pour simuler la fuite on a développé un générateur de flux sortant a partir du flux

entrant. La troisième fenêtre représente les flux choisis par le générateur pour être attaqués et les adresses des attaquants pour tous les types d'analyse. Avec la quatrième fenêtre, on peut commencer l'analyse et voir les résultats pour tous les types d'analyses.

On va expliquer les composants des fenêtres précédentes et leur ordre d'apparition. La première fenêtre qui va apparaitre est présentée dans la figure 4.3.

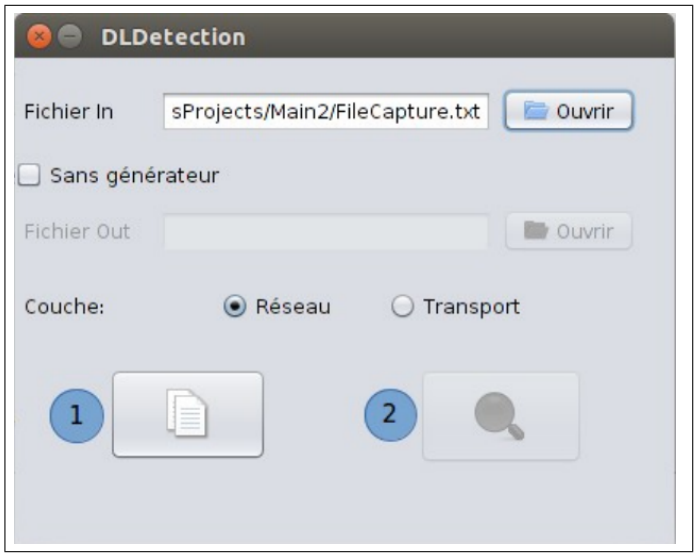

FIGURE 4.3 – La fenêtre principale si on a choisi d'utiliser le générateur

Après la sélection du chemin du fichier entrant obtenu par Wireshark, si on a pas le fichier sortant pour faire l'analyse directement, la case à cocher **Sans générateur** reste décoché.

Il ne reste que d'appuyer sur le bouton 1 (le bouton 2 est désactivé) pour lancer le générateur, la fenêtre figurée par 4.4 s'affiche.

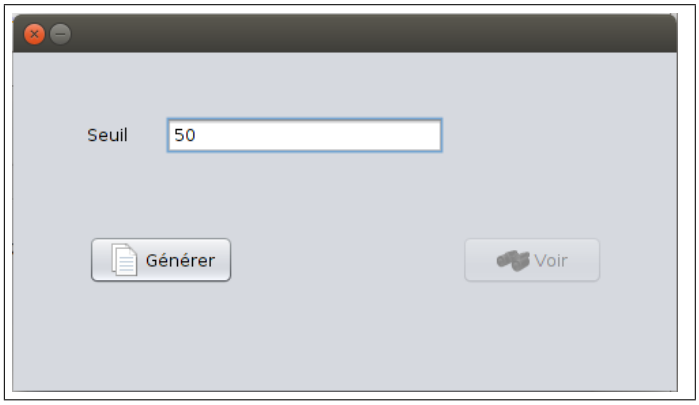

FIGURE 4.4 – La fenêtre de générateur

A partir de cette fenêtre, le générateur prend la valeur du seuil pour générer le flux sortant. Le bouton générer lance le générateur, après quelques secondes la génération sera terminée et le bouton **Voir** sera activé (la figure 4.5).

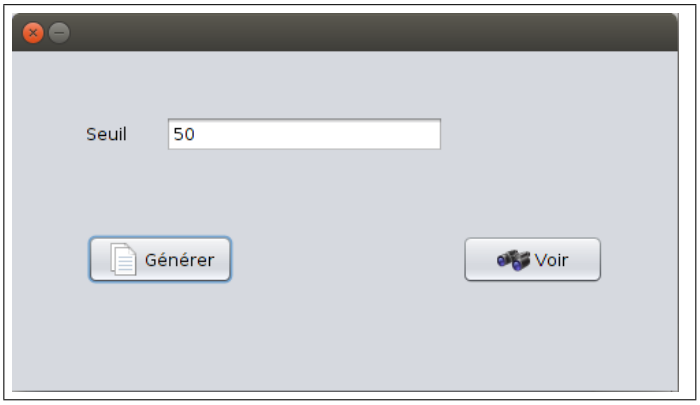

FIGURE 4.5 – La fenêtre de générateur après la génération

Pour chaque type de flux, le bouton **Voir** affiche : le flux choisi par

le générateur, la destination de fuite, la taille du flux et le pourcentage atteint. Ces informations sont présentés dans la figure 4.6.

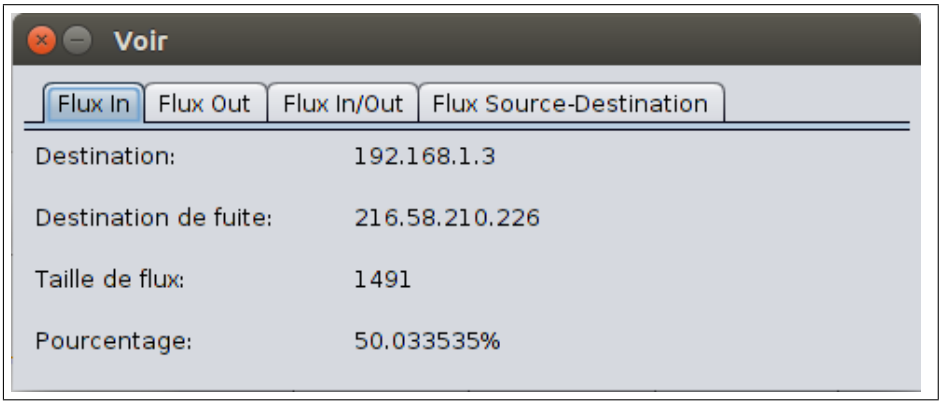

FIGURE 4.6 – Les adresses choisies par le générateur

Maintenant, on peut revenir à la fenêtre principale pour accomplir la phase d'analyse, le bouton 2 est activé (la figure 4.7).

FIGURE 4.7 – La fenêtre principale après la génération

Si vous appuyer sur le bouton 2, la fenêtre figurée par 4.8 s'affiche.

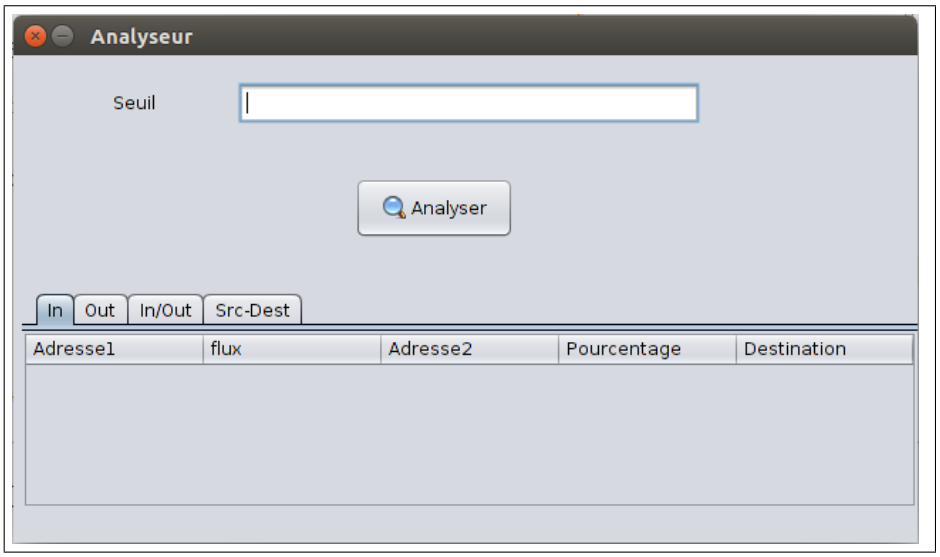

FIGURE 4.8 – La fenêtre d'analyse

Après avoir entré le seuil d'analyse et appuyé sur le bouton **Analyser**, la figure 4.9 représente la fenêtre obtenue.

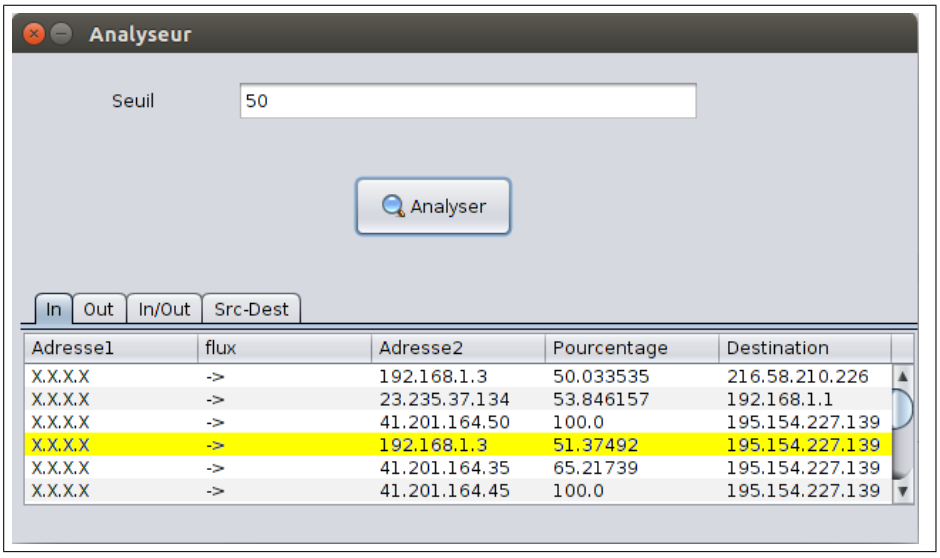

FIGURE 4.9 – La fenêtre après l'analyse

Dans la fenêtre au-dessus, les lignes de la table présentent que l'adresse **Destination** prend un **Pourcentage** du flux envoyé par **Addresse1** vers **Address2**. Dans la ligne jaune, l'adresse 195.154.227.139 prend 51.37492% du flux envoyé à l'adresse 192.168.1.3.

Vous pouvez désactiver le générateur et faire l'analyse à partir des deux fichiers entrant et sortant en cochant la case **sans générateur** (le bouton 1 est désactivé), voir la figure 4.10.

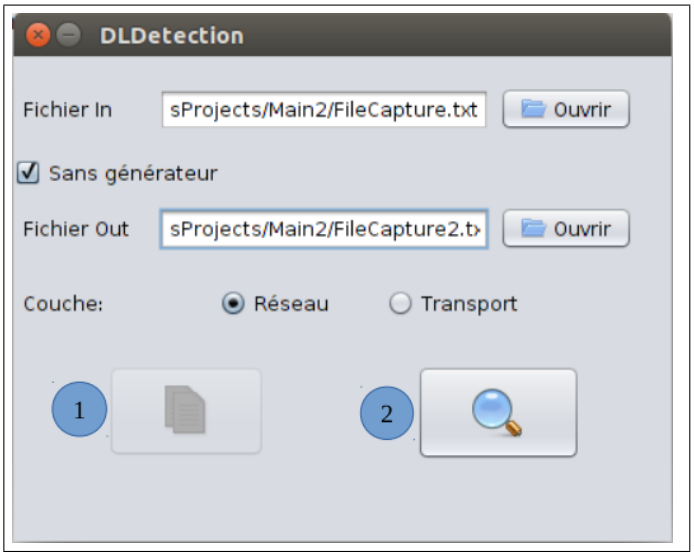

FIGURE 4.10 – La fenêtre principale si on a choisi le mode sans générateur

L'analyse se fait de la même manière présentée au-dessus.

4.3 L'explication des résultats

Nous avons développé le générateur pour deux causes ; le premier pour simuler les fuites, le second pour assurer que l'analyseur a bien fonctionné. Le générateur fonctionne comme suit : il choisit une adresse d'attaquant pour être la destination de fuite. Ensuite, il choisit un flux pour être attaqué par

la première adresse prise. Le générateur répète le même travail pour chaque type de flux de tel sorte que les adresses destination de fuite sont différentes entre elles. Ainsi, on obtient quatre adresses d'attaquant une adresse pour chaque type de flux. La figure 4.11 présente les résultats d'une analyse avec les informations de génération.

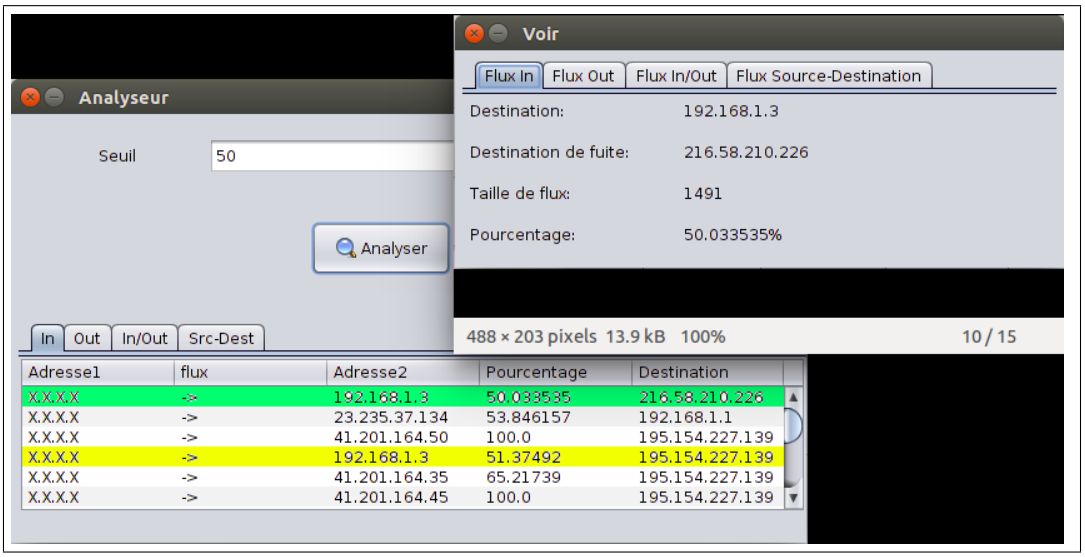

FIGURE 4.11 – Exemple de résultats d'analyse entrant

On remarque que le résultat de l'analyse nous a donné les mêmes adresses et valeurs choisies par le générateurs (la ligne vert). D'autre part, il y a d'autres résultats affichés que le générateur n'a pas choisi. Les autres flux existants dans l'affichage "Flux In" sont des parties des trois autres flux générés par le générateur.

Par exemple ; si le générateur choisit l'adresse A comme l'adresse destination de fuite pour le flux entant de l'adresse B ($\rightarrow$ B) et choisit l'adresse C comme l'adresse destination de fuite pour le flux sortant de l'adresse D (D $\rightarrow$), et s'il y a un flux (D $\rightarrow$ B). Donc il est possible qu'on trouve dans le résultat du flux entrant que C prend une part du flux entrant de B ($\rightarrow$ B).

4.4 Les classes

TABLE 4.1 – La liste des classes implémente dans notre programme

Classe	Description
Paquet.java	Contient l'adresse ip source, destination, et la donnée brute.
Count.java	Contient une adresse ip et le nombre de répétition de celle-ci dans une liste de Paquets.
Existe.java	Nécessaire pour la recherche.
Init.java	Prépare le flux entrant et génère le flux sortant à partir de celui-ci.
TrafficSrc.java	Prépare une liste des Counts pour les adresses sources présentes dans le flux entrant et fait l'analyse source.
TrafficDest.java	Prépare une liste des Counts pour les adresses destination présentes dans le flux entrant et fait l'analyse destination.
TrafficAll.java	Prépare une liste des Counts pour les adresses présentes dans le flux entrant et fait l'analyse entrant/sortant.
TrafficSrcDest.java	Prépare une liste des Counts pour les adresses sources présentes dans le flux entrant et fait l'analyse source-destination.
Traffic.java	La classe mère des quatre classes précédentes.
PrincipalWin.java	L'interface principale pour l'utilisateur.
Generate.java	Pour générer le flux sortant.
ShowGenerator.java	L'interface qui présente les adresses et les flux choisis par le générateur pour simuler la fuite.
ScanWin.java	L'interface d'analyse.

Conclusion et perspectives

Dans cet mémoire, nous nous sommes intéressés au problème de fuite de données dans les routeurs malicieux. Nous avons proposé une méthode permettant d'analyser le flux entrant et le flux sortant du routeur, afin de détecter s'il fuit les données.

Nous avons parlé brièvement sur les systèmes DLP, qui sont développés pour la détection des fuites de données dans les entreprises, et sur les techniques de détection des routeurs malicieux. Ensuite, on a présenté notre méthode qui se base sur la comparaison des contenus des paquets appartenant aux flux entrant et sortant.

Les systèmes DLP actuelles préviennent la fuite de données sensibles qui arrivent entre l'entreprise par l'inspection approfondie sur le flux sortant par rapport à une base de données sensibles préparée préalablement. Cette inspection se fait dans trois stations : les systèmes finaux, les serveurs de données et les extrémités de l'entreprise. Dans ce cas la cause de la fuite prévenue peut être d'origine humain ou provenir du logiciel. Cependant, notre technique vise à détecter les fuites par l'inspection sur le flux sortant par rapport au flux entrant des passerelles à l'extrémité de l'entreprise et des routeurs hors de l'entreprise. Dans ce cas, les logiciels implémentés dans les dispositifs précédents sont la cause de la fuite.

Ce qu'il faut dire, c'est que nous n'avons pas appliqué notre méthode dans un environnement réel ce qui rend notre travail inapproprié aux tests. La deuxième chose manquante est la comparaison avec d'autres techniques, puisque nous n'avons pas trouvé pendant notre recherche une technique ayant le même but que le nôtre, et qu'on peut adapter à leur environnement de fonctionnement pour faire une comparaison correcte et légale. Nous avons rencontré des difficultés lors de notre travail, puisqu'on n'a pas trouvé un travail récent ayant une relation directe avec le nôtre.

En fin, nous ne disons pas que notre travail est parfait, de ce fait, plusieurs extensions et perspectives futures pour l'algorithme et l'implémentation restent à faire, les plus importants sont :

1. **Pour l'algorithme :**
   - L'amélioration de l'algorithme pour détecter l'attaquant s'il a utilisé plusieurs adresses pour attaquer un flux visant à s'enfuir du seuil à travers le fait de diviser le flux.
   - Le développement d'un algorithme spécifié à la phase de décision, cet algorithme doit être basé sur une étude précise pour déterminer les seuils et le nombre d'analyses pour décider s'il y a réellement une attaque.
2. **Pour l'implémentation :**
   - La création d'un nouveau format pour les fichiers contenants les deux flux entrant et sortant.
   - La programmation des améliorations proposées pour l'algorithme.
   - La généralisation de l'implémentation pour tous les protocoles réseau et transport.

References

[1] Identity Finder. *Data Loss Prevention : Data-Rest vs. Data-in-Motion.* Identity Finder, LLC, 2009. viii, 7

[2] Risk Insights on governance and compliance. *Data loss prevention -Keeping your sensitive data out of the public domain-.* EYGM Limited, October 2011. viii, 8

[3] Alper Tugay Mızrak. *Detecting Malicious Routers.* PhD thesis, UNIVERSITY OF CALIFORNIA, SAN DIEGO, 2007. viii, 10, 11, 12

[4] R. Patil and M. P. Tahiliani. Detecting packet modification attack by misbehaving router. In *Networks and Soft Computing (ICNSC), 2014 First International Conference on*, pages 113–118. IEEE, 19-20 Aug. 2014. viii, 13, 14, 15, 18

[5] Dominique SERET Danièle DROMARD. *Architecture des réseaux.* Pearson Education France, 2009. 5

[6] James F. Kurose and Keith W. Ross. Computer networking : A top-down approach featuring the internet. URL http://www.behzadakbari.com/networking/ross/contents-1.htm. 5

[7] Dr. Ronald Krutz Dr. Eric Cole and James W. Conley. *Network Security Bible.* Wiley Publishing, Inc., 2005. 5

[8] Alexandre Viardin. *Un petit guide pour la sécurité*. MIRABELLUG, 2003. 5

[9] S. Alneyadi, E. Sithirasenan, and V. Muthukkumarasamy. Adaptable n-gram classification model for data leakage prevention. In *Signal Processing and Communication Systems (ICSPCS), 2013 7th International Conference on*, pages 1–8. IEEE, 16-18 Dec. 2013. 5, 7

[10] LLC. PCMag Digital Group Z. Davis. Data leak. URL http://www.pcmag.com/encyclopedia/term/61834/data-leakage. 6

[11] Symantec TM. *Fuites de données, mode d'emploi*. Symantec, 2010. 6

[12] Manasdeep. *DATA LEAKAGE PREVENTION -IMPLEMENTATION AND CHALLENGES-*. Network Intelligence (India), 2012. 7

[13] ISACA Team. *Data Leak Prevention*. ISACA, 2010. 7

[14] Prathaben Kanagasingham. *Data Loss Prevention*. SANS Institute, 2008. 8

[15] WatchGuard Technologies. *Data Loss Prevention : Keep Sensitive Data-In-Motion Safe*. WatchGuard Technologies, Inc., 2010. 8

[16] T. Wolf and R. Tessier. Design of a secure router system for next-generation networks. In *Third International Conference on Network and System Security*, 19-21 Oct. 2009. 9

[17] D. Chasaki and T. Wolf. Design of a secure packet processor. In *Architectures for Networking and Communications Systems (ANCS), 2010 ACM/IEEE Symposium on*, 25-26 Oct. 2010. 9

[18] T. Wolf and S. Parameswaran. Embedded systems security-an overview. *Design Automation for Embedded Systems*, 12 :173–183, 17 Jul. 2008. 9

[19] D. Chasaki, Q. Wu, and T. Wolf. Attacks on network infrastructure. In *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, pages 1–8. IEEE, 31 Jul. 2011. 9, 11

[20] S. Savage A. T. Mizrak and K. Marzullo. Detecting compromised routers via packet forwarding behavior. *IEEE Network*, 22 :34–39, 2008. 11, 14

[21] N. Puketza B. Mukherjee K. A. Bradley, S. Cheung and R. A. Olsson. Detecting disruptive routers : A distributed network monitoring approach. In *Proc. of the IEEE Symposium on Security and Privacy*, 3-6 May 1998. 13

[22] V. N. Padmanabhan and D. R. Simon. Secure traceroute to detect faulty or malicious routing. *ACM SIGCOMM Computer Communications Review*, 33(1) :77–82, 2003. 13

[23] S. Savage A. T. Mizrak and K. Marzullo. Detecting malicious packet losses. *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, 20(2) :191–206, 28 Feb. 2009. 13

[24] R. Wang I. Avramopoulos, H. Koboyashi and A. Krishnamurthy. Highly secure and efficient routing. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*. IEEE, 7-11 March 2004. 13

[25] Java (langage). URL http://fr.wikipedia.org/wiki/java_(langage). 27

[26] Apprenez à programmer en java. URL www.siteduzero.com. 27

[27] Ed Warnicke Ulf Lamping, Richard Sharpe. Wireshark user's guide. URL https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html. 28