République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université d'Ibn Khaldoun – Tiaret

Faculté des Mathématiques et de l'informatique

Département d'Informatique

Thème

Détection des attaques D-DOS

Pour l'obtention du diplôme de MASTER

Option :

Réseaux & Télécommunications

Réalisé par : Mr. GHALEM Mustapha

Encadré par : Mr. MOSTEFAOUI Sid Ahmed Mokhtar

Année universitaire : 2013 - 2014

Remerciements

*Je remercie **ALLAH** de m'avoir donné le courage, la santé et la motivation de finir ce mémoire de fin d'études dans les meilleures conditions.*

*Mes sincères remerciements s'adressent à mon encadreur **Mr. MOSTFAOUI** Sid Ahmed, pour l'aide qu'il m'a apporté, ses encouragements et ses précieux conseils ainsi pour l'inspiration, et le temps qu'il a bien voulu me consacrer pour réaliser ce travail.*

Mes vifs remerciements vont également aux membres du jury pour l'honneur qu'ils m'ont fait en s'intéressant à ce travail en acceptant de l'examiner.

*Je remercie aussi tous **mes enseignants**, et mes camarades de la promotion pour ce qu'ils ont fait pour moi.*

Sommaire

Tables des illustrations

Liste des figures

Liste des formules

Liste des tableaux

Introduction

Du fait des nombreux services qu‘offrent les réseaux informatiques de nos jours, ainsi que le déploiement des réseaux à grande échelle, des menaces de diverse sorte ne cessent d‘apparaitre, et avec le développement des outils de piratage, même des amateurs peuvent causer des dommages et des pertes de données. C’est pour cela que la sécurité informatique devient un enjeu capital.

L‘un des outils clés dans une stratégie de sécurité moderne est le système de détection d‘intrusions (IDS – Intrusions Detection System). Un IDS permet de détecter des actions malicieuses ciblant le système informatique, et d‘en alerter le responsable de sécurité. En effet, les IDSs génèrent des alertes qui rapportent des attaques élémentaires, tandis qu’il existe des attaques complexes (séquence ordonnée d’attaques élémentaires) dont le responsable de sécurité est sensé de détecter, cette tache devient lourde avec le grand volume d’alertes à corréler manuellement. En plus, la prédiction d’attaques est très difficile vue le nombre important de scénarios qui résultent de la corrélation.

A cette fin, l’objectif de notre travail consiste à mettre en œuvre un IDS qui sera en mesure de faire la prédiction des scénarios d’attaque, tout en utilisons les réseaux bayésiens. Pour ce faire, nous avons décomposé notre travail en quatre (04) chapitres, les deux premiers seront consacrés à l’état de l’art :

- ✓ Une introduction à la sécurité des réseaux avec une classification des menaces excitantes seront données dans le premier chapitre.
- ✓ Nous verrons la détection des intrusions et une taxonomie des systèmes de détection d’intrusions dans le second chapitre.
- ✓ Le troisième chapitre fera sujet de la corrélation d’alertes avec les réseaux bayésiens.
- ✓ Le dernier sera consacré à la mise en œuvre de notre IDS et l’application des nouvelles règles obtenu de la traduction du modèle bayésien.

Chapitre I :

La sécurité et les menaces dans les réseaux

Chapitre I : La sécurité et les menaces dans les réseaux

I.1-Introduction

L'information dans le milieu professionnel est d'une importance majeur, ce qui rend sa protection une priorité. Pour cela, il faut mettre une politique de sécurité définissant des règles et des mécanismes préventifs.

Dans ce chapitre, nous commençons par des notions de base en sécurité informatique, et nous verrons la sécurité réseau un peu plus en détails. Par la suite, nous classifions les attaques orientées réseau reposantes sur les faiblesses de sécurité et enfin nous citons une série d'outils et de techniques étant être des contre mesure permettant de sécuriser son réseau.

I.2- La sécurité informatique

I.2.1- Définition

La définition de la « sécurité informatique » n'est pas triviale **[1]**. Il est difficile de donner une définition précise pour décrire ce qu'est vraiment la sécurité.

Le dictionnaire LAROUSSE définie la sécurité comme suit : «Situation dans laquelle quelqu'un, quelque chose n'est exposé à aucun danger, à aucun risque, en particulier d'agression physique, d'accidents, de vol, de détérioration : Cette installation présente une sécurité totale» **[2]**. Dans le monde de l'informatique, l'organisation internationale de standardisation ISO a défini la sécurité informatique par la préservation des trois propriétés suivantes (confidentialité, intégrité, disponibilité) **[1]**.

Confidentialité : consiste à s'assurer que l'information est accessible uniquement aux utilisateurs autorisés **[3]**. Toute interception ne doit pas être en mesure d'aboutir. Les données doivent être cryptées et seuls les acteurs de la transaction possèdent la clé de compréhension.

Intégrité des données : consiste à s'assurer de l'exactitude et la complétude de l'information **[3]**, c'est à dire, il faut garantir à chaque instant que les données qui circulent sont bien celles que l'on croit, qu'il n'y a pas eu d'altération (volontaire ou non) au cours de la communication **[4]**.

La disponibilité : consiste à s'assurer que l'accès à l'information est continuellement disponible aux utilisateurs autorisés **[3]**. La disponibilité d'un équipement se mesure en divisant la durée durant laquelle cet équipement est opérationnel par la durée durant laquelle il aurait dû être opérationnel **[4]**.

D'autres concepts sont souvent considérés comme faisant partie de la taxonomie de la sécurité informatique telle que la "Non-répudiation" et la "Vie privée".

I.2.2- Termes connexes

Sécurité informatique dans l'entreprise : de façon générale la sécurité informatique dans l'entreprise peut être définie par l'ensemble des moyens matériels, logiciels et humains mis en œuvre pour minimiser les vulnérabilités d'un système d'information, et le protéger contre les menaces accidentelles ou intentionnelles, provenant de l'intérieur ou de l'extérieur de l'entreprise **[1]**.

Vulnérabilité : La vulnérabilité est souvent définie comme une faille dans le système permettant de violer la politique de sécurité **[1]**.

Politique de sécurité : une politique de sécurité représente l'ensemble de règles qui fixent les actions autorisées et interdites dans le domaine de la sécurité **[1]**. Une politique doit préserver les propriétés de la sécurité, c'est à dire la confidentialité, l'intégrité et la disponibilité.

I.3- La sécurité réseaux

La sécurité informatique est un vaste domaine qui lui-même englobe plusieurs sous-domaines tels que la sécurité des logiciels, personnels et les réseaux.

I.3.1-Les menaces réseaux

Les différentes catégories de menaces qui pèsent sur un réseau peuvent être classées comme illustré à la figure suivante **[7]**.

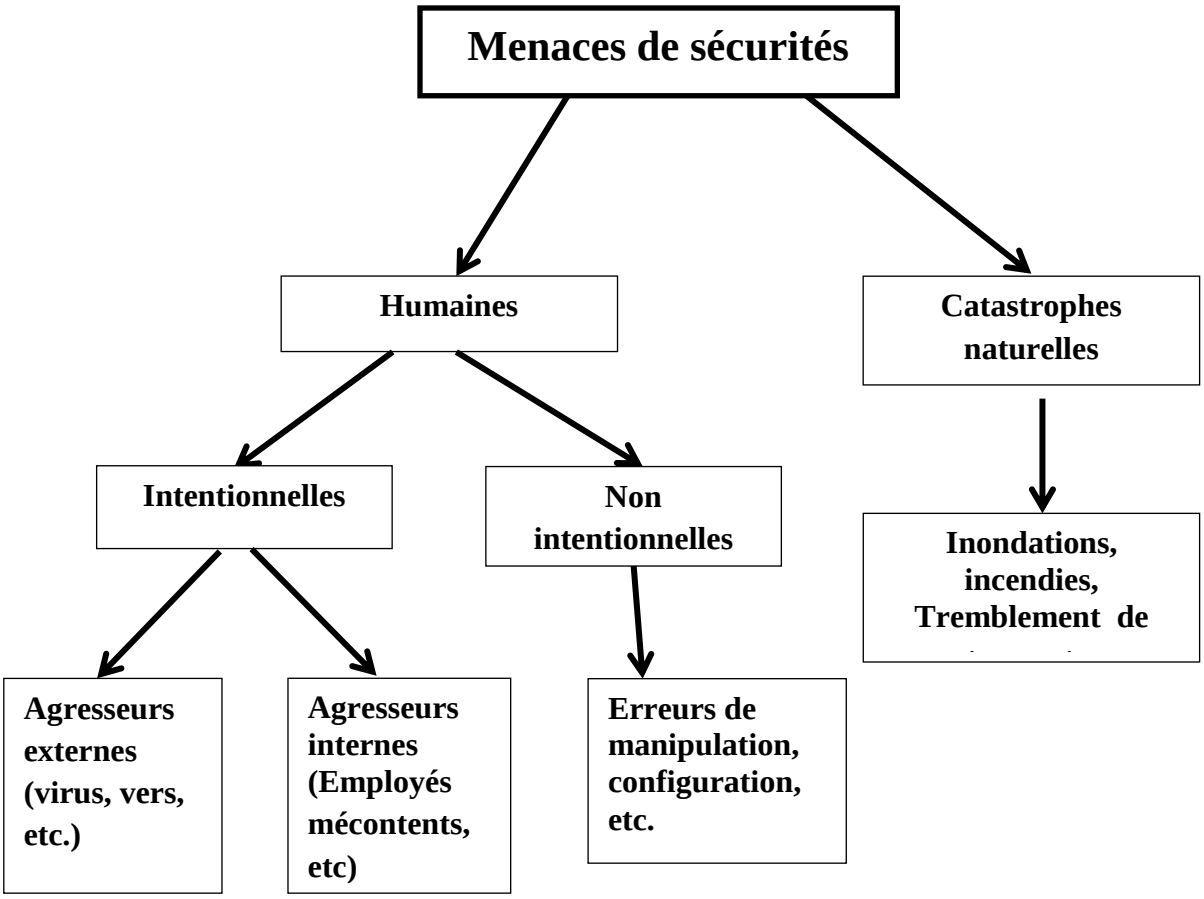

Figure I.1 : Typologie des menaces.

Les menaces intentionnelles mettent généralement en œuvre des outils et des techniques d'attaques très variés.

Définition attaque : C'est une action malveillante qui tente d'exploiter une faiblesse dans le système et de violer un ou plusieurs besoins de sécurité **[8]**.

Ils existent différentes attaques susceptibles d'affecter un réseau et les systèmes qui le composent. Avec la généralisation d'Internet et des moyens de communication modernes, une nouvelle forme d'insécurité s'est répandue, qui s'appuie sur l'utilisation de codes informatiques pour perturber ou pénétrer les réseaux et les ordinateurs.

Les attaques touchent généralement les trois composantes suivantes d'un système : la couche réseau (IP, TCP, etc.), s'en charge de connecter le système au réseau, le système d'exploitation (Windows, Mac, etc.), s'en charge d'offrir un noyau de fonction au système, et

la couche applicative (Navigateur web, application de courrier électronique, etc.), s'en charge d'offrir des services spécifiques **[6] [7]**.

I.3.2- Anatomie d'une attaque

Une attaque est souvent décrite à l'aide des 5 "p" **[9]**:

- ✓ **Probe(Analyser) :** c'est la collecte d'informations sur le système cible, elle peut s'effectuer de plusieurs manières. Comme par exemple un scan des ports grâce au programme Nmap pour déterminer la version des logiciels utilisés, et des outils comme firewalk, hping ou SNMP Walk permettent quant à eux de découvrir la nature d'un réseau.
- ✓ **Penetrate (Pénétrer) :** utilisation des informations récoltées pour pénétrer un réseau. Des technique comme le brute force ou les attaques par dictionnaires peuvent être utilisées pour outrepasser les protections par mot de passe.
- ✓ **Persist (Peréniser) :** création d'un compte avec les droits de super utilisateur pour pouvoir se réinfilter ultérieurement .Une autre technique consiste à installer une application de contrôle à distance capable de résister à un reboot
- ✓ **Propagate(Propager) :** cette étape consiste à observer ce qui est accessible et disponible sur le réseau local.
- ✓ **Paralyse (Paralyser) :** cette étape peut consister en plusieurs actions. Le pirate peut utiliser le serveur pour mener une attaque sur une autre machine, détruire des données ou encore endommager le système d'exploitation dans le but de planter le serveur.

I.3.3-Les attaques réseau

Les attaques réseau sont aujourd'hui si nombreuses qu'il serait difficile de les décrire toutes. Cependant il est possible de dresser une typologie des faiblesses de sécurité afin de mieux appréhender ces attaques, qui ont pour point commun d'exploiter des faiblesses de sécurité **[5]**.

Les attaques réseau s'appuient sur divers types de faiblesses, que l'on peut classifier par catégorie :

- ✓ **Faiblesses des protocoles,** aucun protocoles réseau n'a été conçu pour tenir compte des problèmes de sécurité, la plupart des protocoles utilisés dans un réseau, tels SNMP (Simple Network Management Protocol) pour la supervision ou BGP (Border Gateway Protocol) pour le routage, n'implémentent pas de véritable couche de sécurité et s'exposent à diverses attaques, comme les attaques par fragmentation, déni de service, etc.
- ✓ **Faiblesses d'authentification,** les protocoles réseau n'ont prévu aucun mécanisme d'authentification véritable et subissent des attaques qui s'appuient sur ces faiblesses d'authentification, comme les attaques de type spoofing, man-in-the-middle, etc.
- ✓ **Faiblesses d'implémentation ou bogues des programmes** (système d'exploitation, application de routage, etc.) exposent à d'autres attaques, qui sont de loin les plus importantes en nombre. La raison à cela est que le développement des logiciels et des piles réseau se fait de plus en plus rapidement et sans règles strictes. Parmi les innombrables attaques qui utilisent de mauvaises implémentations ou des erreurs de programmation, citons les attaques de type SYN flooding et ping-of-death.
- ✓ **Faiblesses de configuration des équipements réseau** peuvent provenir d'une mauvaise configuration d'un pare-feu, laissant passer du trafic non autorisé par la politique de sécurité, ou d'un équipement réseau, permettant à un attaquant d'y accéder, etc.

Les attaques peuvent aussi être regroupées en trois familles différentes :

- ✓ ***Les attaques directes***

Les attaques réseau peuvent être lancées directement, le pirate attaquant sa victime et exposant ainsi son identité **[5] [10]**, comme l'illustre la Figure I.2.

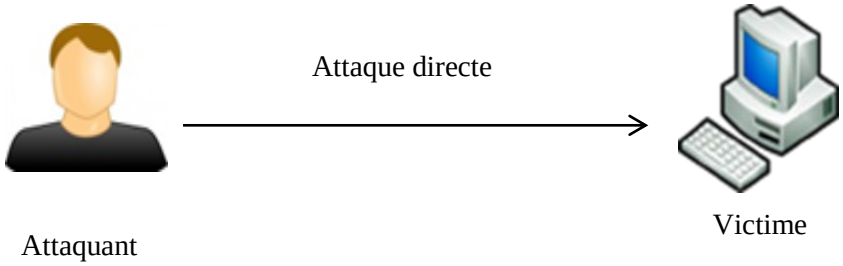

Figure I.2 : Attaque directe.

✓ ***Les attaques indirectes par rebond***

Les attaques réseau peuvent aussi être lancées indirectement par l'intermédiaire d'un système rebond afin de masquer l'identité (adresse IP) du pirate et d'utiliser les ressources du système intermédiaire **[10]**. Les paquets d'attaque sont dans ce cas envoyés au système intermédiaire, lequel répercute l'attaque vers le système cible **[5]**, comme l'illustre la Figure I.3

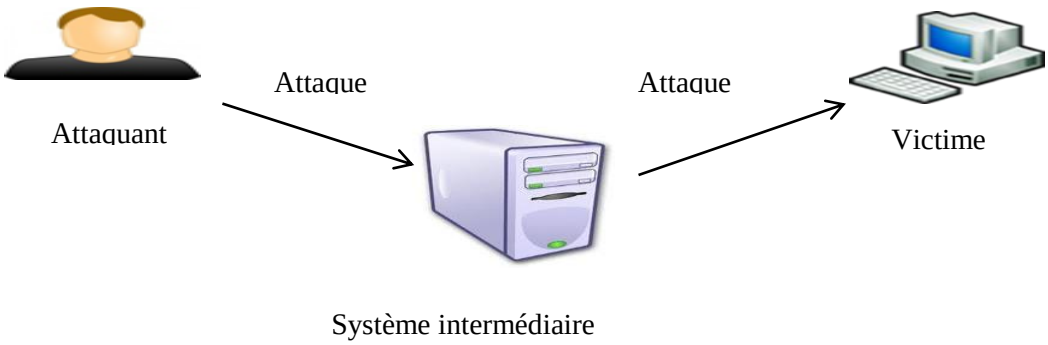

Figure I.3 : Attaque indirecte par rebond.

✓ ***Attaques indirectes par réponse***

Certaines attaques, dites indirectes par réponse, offrent au pirate les mêmes avantages que les attaques par rebond **[10]**. Au lieu d'envoyer l'attaque au système intermédiaire pour qu'il la répercute, l'attaquant lui envoie une requête, et c'est la réponse à cette requête qui est envoyée au système cible **[5]**, comme l'illustre la Figure I.4.

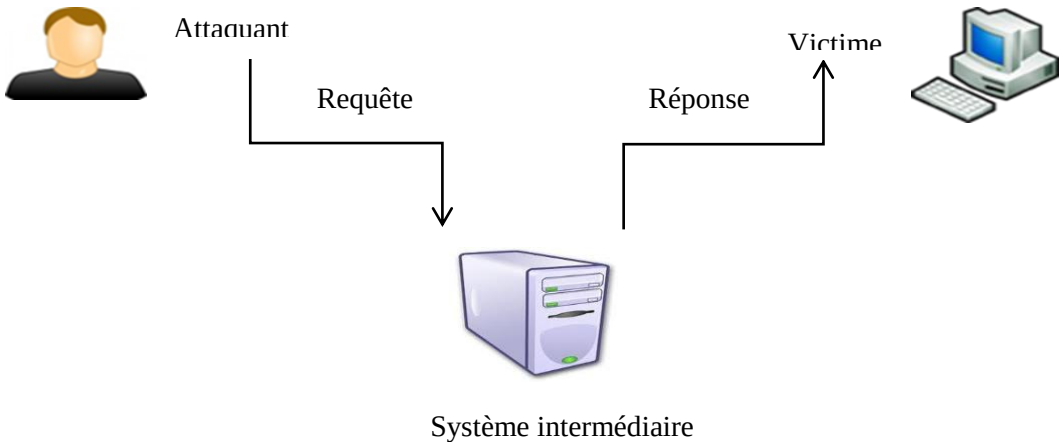

Figure I.4 : Attaque indirecte par réponse.

I.3.3.1-Les attaques sur les protocoles réseau

✓ ***Attaques permettant d'établir la cartographie du réseau***

Les attaques visant à établir la cartographie d'un réseau ont pour but de dresser les artères de communication des futurs systèmes cibles. Elles ont recours pour cela à des outils de diagnostic tel que « Traceroute », qui permet de visualiser le chemin suivi par un paquet IP d'un hôte à un autre **[5]**.

✓ ***Attaques permettant d'identifier les systèmes réseau (scanning)***

Certaines attaques visent à identifier un système dans le but de dresser les futurs moyens de pénétration de ce système **[5]**.

Il existe différentes techniques de balayage des systèmes comme illustré à la figure suivante :

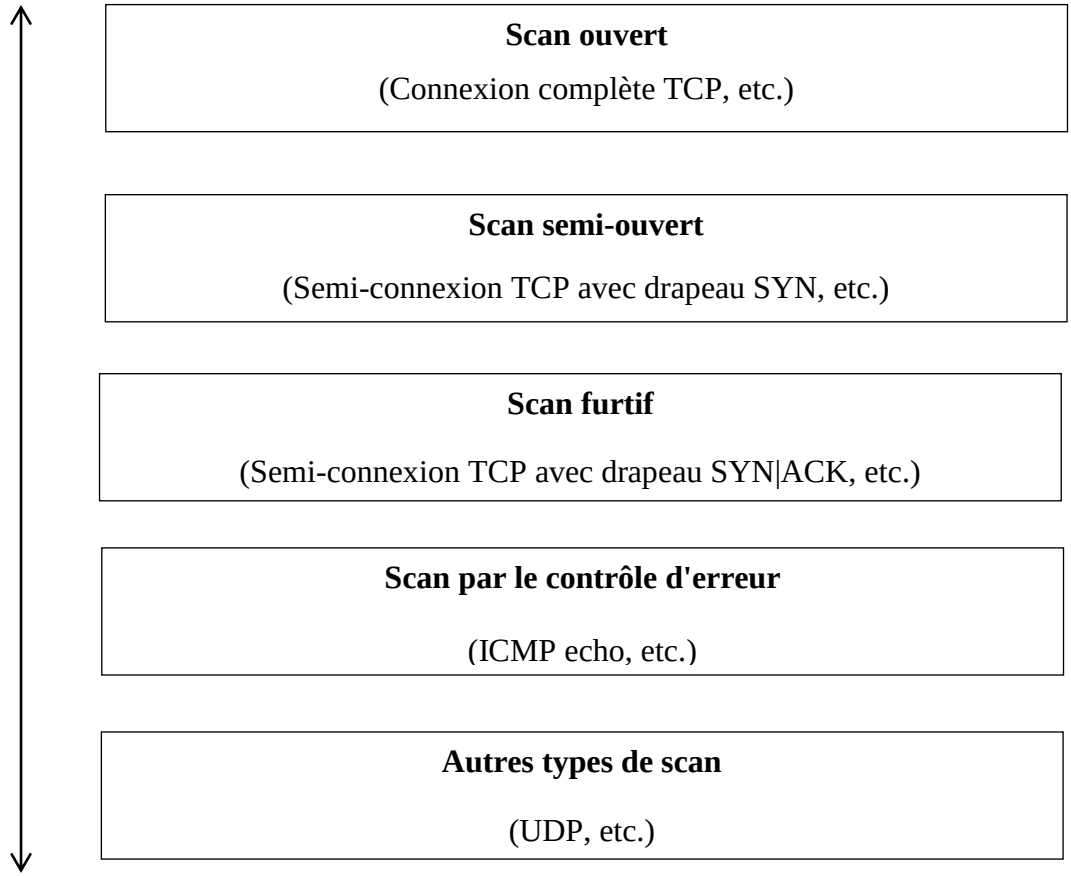

Figure I.5 : Les différents types de scan.

✓ ***Attaques permettant d'écouter le trafic réseau (sniffing)***

L'attaque par sniffing est généralement utilisée par les pirates pour capturer les mots de passe. Lorsqu'on se connecte à un réseau qui utilise le mode broadcast, toutes les données en transit arrivent à toutes les cartes réseau connectées à ce réseau **[11]**. En temps normal, seules les trames destinées à la machine sont lues, les autres étant ignorées.

✓ ***Attaques par déni de service et par inondation (DoS)***

Le déni de service est une attaque qui vise à rendre indisponible un service, un système ou un réseau. Ces attaques se basent généralement soit sur une faiblesse d'implémentation ou bogue, soit sur une faiblesse d'un protocole **[5]**.

Les premières attaques par déni de service sont apparues entre 1998 et l'an 2000 et visaient de grands sites Internet (Yahoo, Ebay, eTrade, etc.). Concernant le site Yahoo, ce site a été attaqué en février 2000 et a été "noyé" (flood) sous un gigabyte de données en quelques secondes pendant plus de 3 heures d'au moins50 points réseau différents **[7]**.

- ***L'inondation :*** généralement l'inondation est la méthode la plus classique pour empêcher un réseau d'assurer sa mission. Son principe de fonctionnement est simple, une ou plusieurs machines inondent le réseau avec des paquets réseau afin de saturer la bande passante de celui-ci. Une fois que toute la bande est occupée, les autres machines ne peuvent plus travailler, ce qui génère une situation de refus de service **[5] [11]**.
- ***L'attaque DDoS (Distributed Denial Of Services) :*** est un dérivé de la précédente sous une forme distribuée comme illustré à la figure suivante :

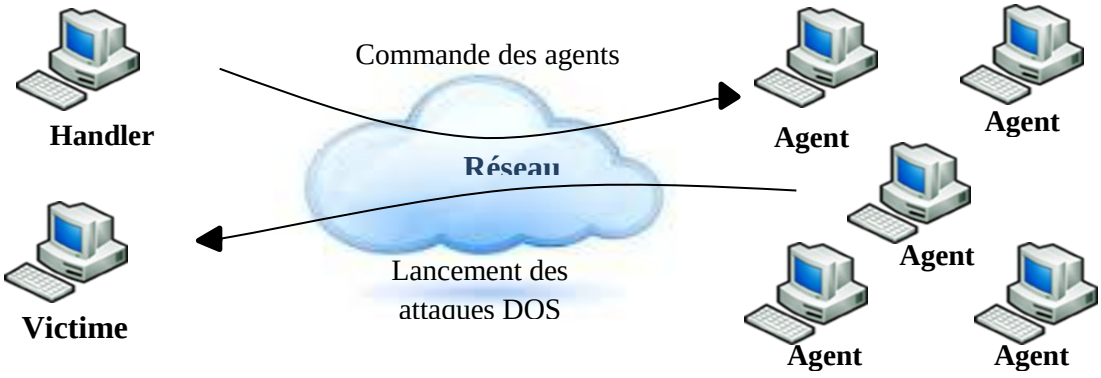

Figure I.6 : Attaque par déni de service distribué.

Parmi les nombreuses attaques DDoS, citons :

- ***TFN (Tribe Flood Network) :*** historiquement c'est la première attaque.
- ***Stacheldraht :*** qui chiffre les ordres de commandes échangés entre les handlers et les agents dans le champ données des paquets ICMP.

Ces attaques ont fait des émules, et d'autres attaques sont apparues, telles que :

- ***Trinoo :*** qui s'appuie sur UDP pour les communications des ordres entre handlers et agents.
- ***TFN2K :*** une version entièrement revue de TFN, qui introduit des phénomènes de génération aléatoire des ports utilisés pour les communications des ordres entre les handlers et les agents, ainsi qu'un phénomène aléatoire dans le lancement des attaques vers les systèmes cible.

I.3.3.2- Les attaques s'appuyant sur les faiblesses d'authentification

✓ ***L'attaque IP spoofing***

L'attaque IP spoofing consiste à se faire passer pour un autre système en falsifiant son adresse IP **[7]**. Le pirate commence par choisir le système qu'il veut attaquer. Après avoir obtenu le maximum de détails sur ce système cible, il détermine les systèmes ou adresses IP autorisés à se connecter au système cible.

✓ ***L'attaque man-in-the-middle***

L'attaque man-in-the-middle consiste à faire passer les échanges réseau entre deux systèmes par le biais d'un troisième, sous le contrôle du pirate **[5] [7]**. Ce dernier peut transformer à sa guise les données à la volée, tout en masquant parfaitement à chaque acteur de l'échange la réalité de son interlocuteur.

I.3.3.3- Les attaques s'appuyant sur les faiblesses d'implémentation

✓ ***L'attaque TCP SYN***

La technique d'inondation SYN, ou SYN flooding, n'est pas une inondation simple **[5]**. Elle s'appuie sur une demande de connexion qui n'aboutit pas et qui sature les ressources du système visé.

Pour gérer les états de connexion entre deux parties, le protocole TCP recourt aux drapeaux URG, ACK, PUSH, RST, SYN et FIN présents dans l'en-tête TCP **[11]**.

✓ ***Les faiblesses du code***

Les piles IP/TCP développées par différents constructeurs ou fournisseurs de services manifestent des différences de comportement malgré les définitions des RFC et contiennent de multiples faiblesses, qui peuvent être exploitées par des attaques bien ciblées **[11]**.

Comme il est théoriquement impossible de vérifier l'absence de bogues dans un programme conçu avec les langages de programmation modernes, il existe une forte probabilité que des bogues permettent à des pirates de gagner des privilèges.

Les principales attaques qui s'appuient sur les erreurs de programmation associées aux piles TCP/IP sont :

- L'attaque "ping de la mort"
- L'attaque "baiser de la mort"
- L'attaque "winnuke"
- L'attaque "land"
- L'attaque "teardrop"

✓ ***Attaques sur les bogues des systèmes d'exploitation***

Les systèmes d'exploitation et les produits ou services additionnels qui y sont greffés contiennent de multiples faiblesses susceptibles d'être exploitées par des attaques ciblées tel que le Buffer Overflow (débordement de tampon) **[5]**.

I.3.3.4- Les faiblesses de configuration

Les faiblesses de configuration des équipements réseau, pare-feu, etc., sont également souvent utilisées pour mener à bien des attaques. Ces dernières peuvent provenir de :

- ✓ L'exploitation d'erreurs de configuration du système.
- ✓ Des configurations des équipements réseau, qui doivent suivre des règles strictes afin d'éviter que le réseau ne joue, un rôle de rebond dans des attaques éventuelles, notamment l'attaque smurf.
- ✓ D'une politique d'accès ou de mots de passe trop laxiste.

- ✓ De comptes utilisateurs génériques, standardisés avec des mots de passe triviaux et associés à des droits d'accès permissifs. Dès lors, un pirate peut commencer son intrusion non pas par la recherche de failles exploitables, mais simplement par des tentatives itératives de pénétration. Celles-ci peuvent commencer par les comptes classiques comme oracle, admin, toor, sybase, solaris, linux, etc. et avec des mots de passe identiques aux noms des comptes.

I.4- Solutions en matière de sécurité réseaux

Actuellement, toute une série d'outils et de techniques permettent à un administrateur de sécuriser facilement son réseau et les machines qui le composent. Chacune de ces techniques se base sur des principes fondamentalement différents, mais celles-ci ont un but commun et qui se présente dans « permettre une connexion entre Internet (réseau non sécurisé) et le réseau de l'entreprise concernée, en assurant la sécurité des équipements et des informations disponibles sur ce réseau » **[12]**.

Nous allons citer et expliquer brièvement quelques outils de sécurité courants

I.4.1 -Les Firewalls

Le mot Firewall (Pare-feu) signifie qu'on instaure une série de protections en un point particulier entre deux entités connectées, en l'occurrence entre Internet et le réseau interne d'une entreprise. En pratique, le Firewall consiste en une architecture, plutôt qu'un matériel ou un logiciel précis. Cette architecture intègre alors une série de composants matériels et logiciels, qui tentent précisément d'assurer le niveau de sécurité requis.

L'architecture la plus utilisée actuellement est basée sur une « Zone démilitarisée », communément appelée DMZ (Demilitarized Zone). Elle consiste à placer un réseau intermédiaire entre l'accès Internet et le réseau interne (éventuellement plusieurs). Cette DMZ sera isolée, aussi bien vis-à-vis de l'Internet que du réseau local, par des systèmes de filtrage (filtres de paquets entrant et sortant). Ensuite, les éventuels serveurs nécessaires à l'entreprise devant continuer à être accessibles de l'extérieur seront connectés directement sur cette DMZ, de manière à les séparer du réseau interne. Par exemple, on pourra y trouver un serveur Web, un serveur DNS (Domain Name Service), un serveur de mails, un serveur FTP (File Transfer

Protocol), etc. Dans le cas où l'un de ces serveurs serait compromis, le filtrage entre la DMZ et le réseau interne doit être capable d'assurer une protection suffisante au réseau interne.

I.4.2- Les filtre de paquets

Un filtre de paquets, tout comme son nom l'indique, permet de filtrer les paquets circulant sur un réseau. Plus précisément, on peut même dire que le filtrage s'effectue sur les paquets traversant une interface réseau. Celui-ci fonctionne en analysant le contenu de ces paquets, principalement en observant les valeurs de certains champs des en-têtes des protocoles IP (Internet Protocol), ICMP (Internet Control Message Protocol), UDP (User Datagramme Protocol) et TCP (Transmission Control Protocol). Cela permet par exemple d'interdire des paquets provenant d'une source précise, étant destinés à une destination précise, des paquets réceptionnés sur une interface précise, des paquets avec des ports sources ou cibles précis, d'intégrer des contraintes d'heures éventuelles d'après l'horaire d'une entreprise, etc.

Au niveau de la configuration, on fait établir une série de règles de filtrage qui reflète la politique de sécurité de l'entreprise. Les paquets ne satisfaisant pas aux règles de filtrage seront alors bloqués, et peuvent entraîner éventuellement la génération d'un message d'erreur (via un protocole comme ICMP).

I.4.3- Les scanners et les outils relatifs à la sécurité

Etant donné que les hackers (pirates informatique) trouvent de plus en plus les outils nécessaires à la réalisation de leurs attaques, les entreprises travaillant dans le domaine de la sécurité ont petit à petit commencé à proposer leurs propres outils de vérification de vulnérabilités. C'est ainsi qu'on commence à avoir apparaître toute une série de scanners, qui offrent de nombreuses possibilités. Il est primordial à l'heure actuelle d'effectuer de nombreux tests de sécurité réguliers, car ces tests permettent de mettre en avance des modifications dans l'architecture et dans la configuration du réseau et des machines qui le composent. Ces outils sont décomposés en toute une série de catégories, dont notamment :

- ✓ Les scanners de vulnérabilités.
- ✓ Les scanners orientés réseaux.
- ✓ Les scanners orientés hôtes (machines).

- ✓ Les sniffers.
- ✓ Les vérificateurs de mots de passe.

I.4.4- Les systèmes de détection d'intrusions

Un IDS a pour fonction d'analyser en temps réel ou différé les évènements en provenance des différents systèmes à travers le réseau, de détecter et de prévenir les attaques. Les IDS ont donc un rôle d'alarme. Les buts sont nombreux :

- ✓ Collecter des informations sur les intrusions.
- ✓ Gestion centralisée des alertes.
- ✓ Effectuer un premier diagnostic sur la nature de l'attaque permettant une réponse rapide et efficace.
- ✓ Réagir activement à l'attaque pour la ralentir ou la stopper.

I.5-Conclusion

Les menaces sur le réseau résident dans le fait d'avoir des vulnérabilités et des faiblesses existantes dans les protocoles, les infrastructures, ainsi que les systèmes d'exploitation des machines qui forment ces réseaux.

Afin d'assurer la sécurité des réseaux et de lutter contre ces menaces, plusieurs solutions ont était suggérées, parmi ces solution on trouve les Firewall (pare-feu), la technique de filtrage de paquets, les scanners ainsi que les systèmes de détection d'intrusion, ces derniers feront sujet du prochaine chapitre.

Chapitre II :

La détection d'intrusions

Chapitre II : La détection d'intrusions

II.1- Introduction

L'enjeu de la sécurité réseau et de diminuer les risques de se faire attaquer, ainsi que de détecter ces attaques dans le cas où elles se produisent. De ce fait, des outils de diagnostic et de détection doivent être déployés pour implémenter la politique de sécurité. C'est la détection d'intrusion qui fera sujet de ce chapitre, où nous verrons une classification des détecteurs d'intrusion, tout en abordons le point de la corrélation d'alertes.

II.2-Détection d'intrusion

J.P. Anderson a introduit le concept de la détection d'intrusions en 1980, il a été le premier à montrer l'importance de l'audit de sécurité **[13]**. Ce concept consisté à surveiller les événements qui se produisent dans un ordinateur ou dans un réseau informatique et de les analyser pour repérer les intrusions. Ces événements sont souvent définis comme des tentatives de violation de la politique de sécurité.

La détection d'intrusion peut être effectuée de deux (02) façons « manuelle ou automatique (à l'aide d'un système) », dans la détection manuelle, un humain analyse les fichiers de logs à la recherche de tout signe suspect pouvant indiquer une intrusion. Tandis que, dans la détection automatique, un système de détection d'intrusion noté « SDI » et mis en œuvre pour ce but.

II.2.1- Nomenclature

- ✓ **Système :** dénote un système d'information contrôlé par un système de détection d'intrusions.
- ✓ **Alarme :** c'est la réponse générée par le système de détection d'intrusions lors de la détection d'une intrusion. Cependant les erreurs de détection peuvent être classées selon deux types :
  - ➢ ***Le positif faux :*** signifie qu'un système de détection d'intrusions signale une intrusion là où aucune intrusion réelle n'a été commise.
  - ➢ ***Le négatif faux :*** A l'inverse de «positif faux », «négatif faux » signifie que le système de détection d'intrusions n'a pas détecté une intrusion ayant réussi.

II.2.2- Description du système de détection d'intrusions

Un système de détection d'intrusions est simplifié par un détecteur qui analyse les informations en provenance du système surveillé comme l'illustre la figure suivante :

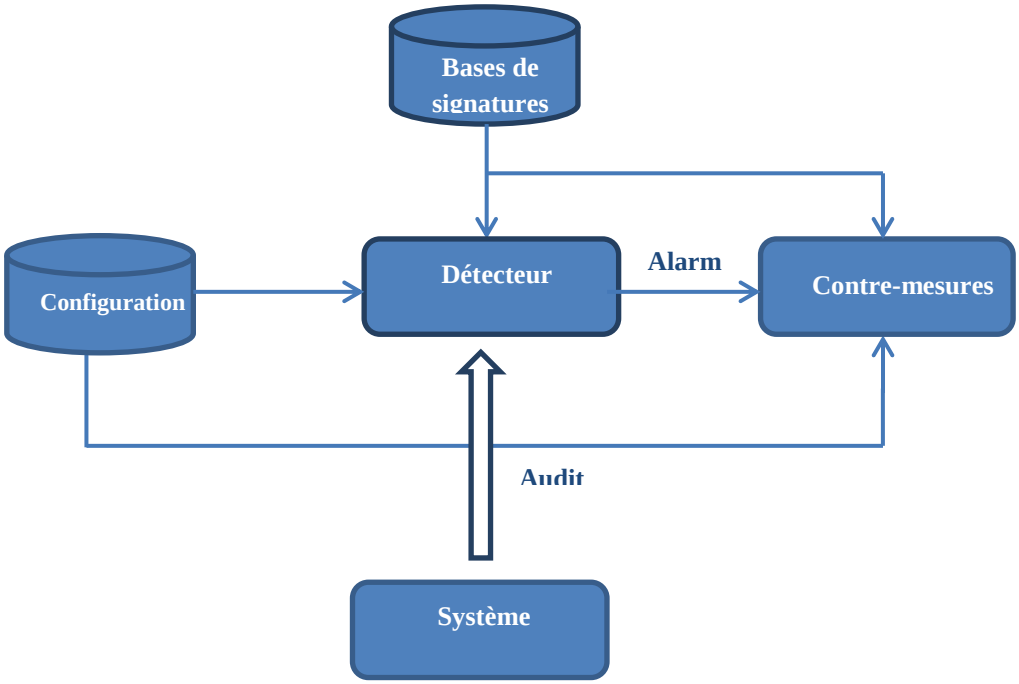

Figure II.1 : Modèle simplifié d'un système de détection d'intrusions.

Le détecteur analyse trois types d'informations : les informations de long terme relatives aux techniques utilisées dans la détection (Base de données des signatures), les informations de configuration qui déterminent l'état courant du système, et les informations d'audit qui décrivent les évènements survenant dans le système **[14]**.

II.2.3- Efficacité du système de détection d'intrusions

Selon Philip dans **[15]**, il définit trois critères pour évaluer l'efficacité des systèmes de détection d'intrusion :

- ✓ **L'exactitude (accuracy) :** on parle de l'exactitude quand le système de détection d'intrusion déclare comme malicieux une activité légale. Ce critère correspond au faux positif.

- ✓ **La performance (performance) :** la performance du système de détection d'intrusion est le taux de traitement des évènements. Si ce taux est faible, la détection en temps réel est donc impossible.
- ✓ **La complétude (completeness) :** on parle de la complétude quand le système de détection d'intrusion rate la détection d'une attaque. Ce critère est le plus difficile, parce qu'il est impossible d'avoir une connaissance globale sur les attaques.
  On note aussi que Debar dans **[14]** a rajouté également les deux critères suivants :

- ✓ **La tolérance aux fautes (Fault tolerance) :** ceci est important, le système de détection d'intrusion doit lui-même résisté aux attaques, particulièrement au déni de service, parce que plusieurs systèmes de détection d'intrusion s'exécutent sur des matériels ou logiciels connus comme vulnérables aux attaques.
- ✓ **La réaction à temps (Timeliness) :** le système de détection d'intrusion doit s'exécuter et propager les résultats de l'analyse le plus tôt possible, pour permettre à l'officier de sécurité de réagir avant que de graves dommages n'aient lieu. Ceci implique plus qu'un calcul de performance, parce qu'il ne s'agit pas seulement de temps de traitement des évènements, mais aussi de temps nécessaire pour la propagation et la réaction à cet évènement.

II.3- Taxonomie des systèmes de détection d'intrusions

Depuis les travaux d'Anderson **[13]** et de Ning **[16]**, le domaine de la détection d'intrusions est en plein développement. On trouve à l'heure actuelle plusieurs systèmes de détection d'intrusions opérationnels, que ce soit des produits commerciaux ou du domaine public.

Les différents systèmes de détection d'intrusions disponibles peuvent être classés **[14]** selon plusieurs critères, ils sont décrits par la figure suivante :

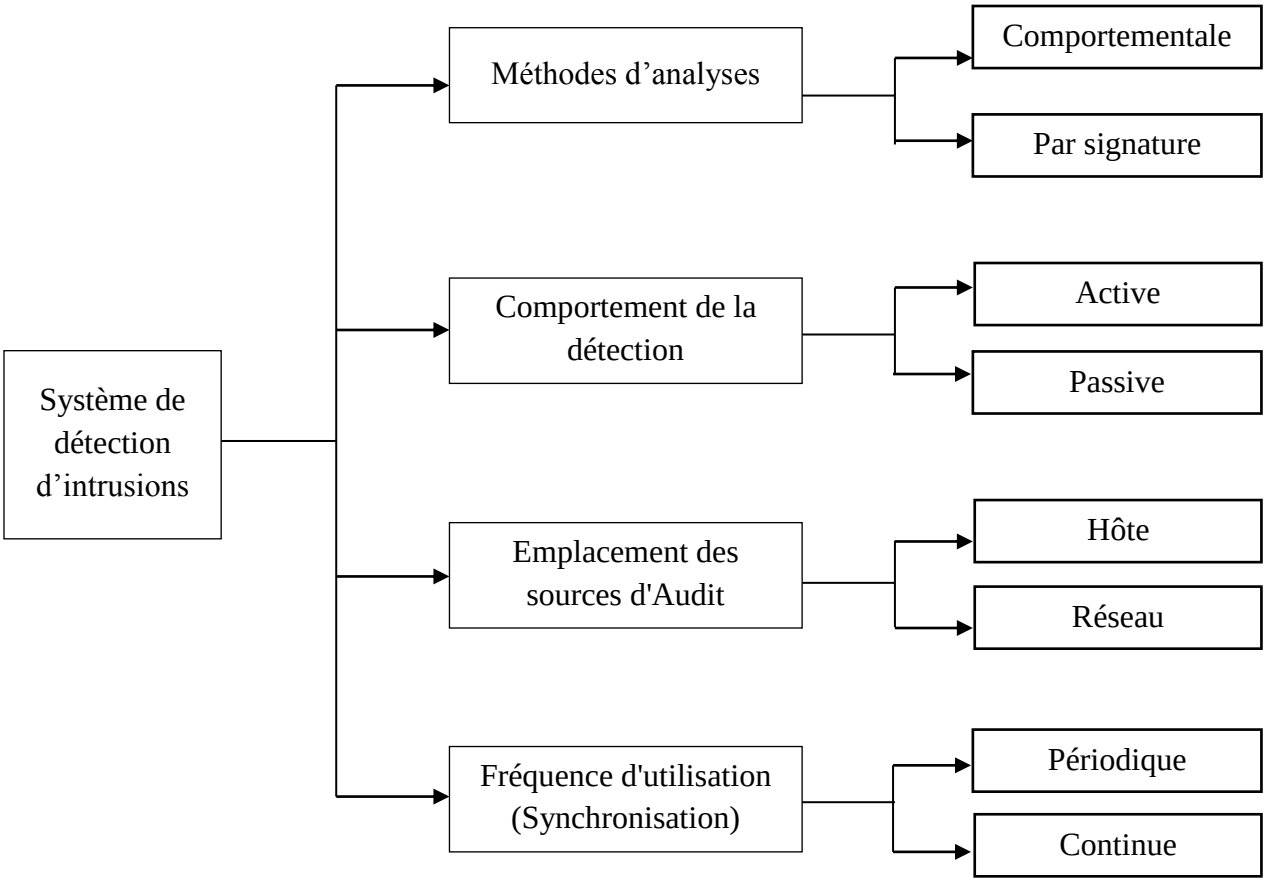

Figure II.2 : Taxonomie des systèmes de détection d'intrusion.

II.3.1- Le comportement de la détection

Le comportement d'un IDS après la détection d'une intrusion est l'ensemble des actions prises par le système lorsqu'il détecte une attaque. Ces réponses peuvent être *passives* ou bien *actives.*

II.3.1.1-passive

La plupart des systèmes de détection d'intrusions n'apportent qu'une réponse passive à l'intrusion dont les principales sont :

- ✓ **Alarme :** Les alarmes sont produites par les SDIs pour informer les administrateurs réseau lorsque des attaques sont détectées. La forme la plus commune est d'afficher un message d'alerte contenant des informations détaillées de l'intrusion détectée sur la console du responsable de la sécurité.

- ✓ **SNMP Trap :** Certains *IDSs* sont conçus pour produire des alertes et envoyer les rapports au système de gestion de réseau (network management system). Ils utilisent le protocole *SNMP* (*Sample Network Management Protocol*), qui est un protocole dédié à la gestion du réseau **[1]**.
- ✓ **L'archivage (logging) :** permet aux analystes de faire des analyses approfondies, et de faire des corrélations avec l'historique dont ils disposent concernant les évènements qui se sont produits auparavant.

II.3.1.2- active

D'autres systèmes de détection d'intrusions peuvent, en plus de la notification à l'opérateur, prendre automatiquement des mesures pour stopper l'attaque en cours. Il y a trois catégories de réponses actives :

- ✓ **Rassembler des informations additionnelles :** il est très important de rassembler des informations additionnelles sur une attaque afin de l'identifier avec précision, faire des corrélations, ou bien communiquer avec d'autres types d'*IDSs* installés sur le réseau **[1]**.
- ✓ **Changer l'environnement :** stopper l'attaque en progression et puis bloquer l'accès de l'attaquant. Typiquement, les *IDSs* n'ont pas les capacités de bloquer l'accès d'une personne spécifique, mais ils peuvent uniquement rompre des connexions ou bloquer certains paquets spécifiques en s'appuyant sur les mécanismes des protocoles Internet. Cela est dû à la capacité du hacker expert de construire des paquets falsifiés (forging packets) **[1]**. Parmi ces actions la configuration des routeurs et des Firewalls pour bloquer les paquets provenant de l'adresse *IP* de l'attaquant ou bien de bloquer les paquets selon le numéro de port, le protocole, ou le service utilisé par l'attaquant.
- ✓ **Agir contre l'intrus :** cette forme implique le lancement des contre-attaques ou d'essayer d'obtenir activement des informations sur l'hôte ou l'emplacement de l'attaquant **[1]**. Cette mesure est potentiellement dangereuse car, en trompant l'IDS elle peut mener à des dénis de service provoqués par l'IDS, il est conseillé de laisser l'être humain prendre la décision finale.

II.3.2-L'emplacement des sources d'audits

C'est la manière la plus connue pour classifier les *IDSs*. Certains *IDSs* analysent des paquets capturés à partir du réseau, d'autres analysent des informations produites par le système d'exploitation ou par des applications pour la recherche des signes d'intrusions.

II.3.2.1-NIDS (Network-Based IDS)

Ces outils analysent le trafic réseau comportant généralement une sonde qui "écoute" sur le segment de réseau à surveiller et un moteur qui réalise l'analyse du trafic afin de détecter les signatures d'attaques. La plupart des *NIDS* sont aussi dits *IDS on-line* car ils analysent le flux en temps réel.

De tels *IDSs* doivent être de plus en plus performants afin d'analyser les volumes de données de plus en plus importants pouvant transiter sur les réseaux **[1]**.

II.3.2.2-HIDS (Hôte-Based IDS)

Les *IDSs* de ce type analysent le fonctionnement et l'état des machines sur lesquelles ils sont installés afin de détecter les attaques. Pour cela ils analysent les journaux système (logs), le contrôle d'accès aux appels systèmes, la vérification d'intégrité des systèmes de fichiers, etc. Ils sont très dépendants du système sur lequel ils sont installés. Ces *IDSs* peuvent s'appuyer sur des fonctionnalités d'audit propres ou non, au système d'exploitation, pour en vérifier l'intégrité, et générer des alertes.

II.3.2.3- IDS hybrides

Les IDS hybrides rassemblent les caractéristiques de plusieurs *IDS* différents. En pratique, on ne retrouve que la combinaison de *NIDS* et *HIDS* **[1]**. Ils permettent, en un seul outil, de surveiller le réseau et l'hôte. Les sondes sont placées dans des points stratégiques, et agissent comme *NIDS* et/ou *HIDS* suivant leurs emplacements. Toutes ces sondes remontent alors les alertes à une machine qui va centraliser, agréger, et lier les informations d'origines multiples.

II.3.3- La fréquence d'utilisation (la synchronisation)

II.3.3.1- En temps différé (Périodique)

Ce type de système de détection d'intrusions, analyse périodiquement les différentes sources de données, à la recherche d'une éventuelle intrusion ou une anomalie passée. Cette approche est employée surtout, dans les *Host-IDSs*, qui analysent les logs du système d'exploitation dans des intervalles de temps réguliers **[1]**.

II.3.3.2- En temps réel (Continu)

Les *IDSs* en temps réel, traitent des flux continus d'informations à partir des différentes sources d'informations. C'est la technique dominante dans les *IDSs* réseau, qui recueillent l'information du trafic réseau **[1]**.

II.3.4- La méthode d'analyse

Deux grandes approches ont été proposées dans la littérature :

- ✓ La détection d'intrusions par signature ;
- ✓ La détection d'intrusions comportementale.

Ces deux approches s'opposent dans leur principe de détection : l'approche par signature se fonde sur la recherche de traces d'attaques ou d'intrusions **[14]**, alors que l'approche comportementale recherche les déviations du comportement de l'entité observée par rapport à un modèle du comportement normal de cette entité. Cette dernière a été la première approche proposée par Anderson en 1980 **[13]**.

II.3.4.1- La détection par scénario

Les systèmes de détection d'intrusions par signature fondent leur détection sur la reconnaissance, dans le flux d'évènements générés par une ou plusieurs sondes de signatures d'attaques qui sont contenues dans une base de signatures.

Une signature est un motif, dans le flux d'événements, de scénarios d'attaques définis au préalable. Un IDS par signature se compose :

- ✓ d'une ou plusieurs sondes, générant un flux d'événements, qui peuvent être de type réseau ou hôte.
- ✓ d'une base de signatures.
- ✓ d'un système de reconnaissance de motifs dans le flux d'événements.

a) **La base de signatures :** le taux de couverture de l'IDS, dépend essentiellement de la qualité de la base de données, puisque seules les attaques, dont la signature est présente dans la base, sont susceptibles d'être détectées. Les signatures sont décrites à l'aide de langages de description d'attaques **[17]**, elles sont la plupart du temps, définies par un opérateur, bien que, des travaux récents permettent la génération automatique des signatures **[18] [19]** la base de signatures doit également être maintenue (intégration des nouvelles signatures découvertes) car sans maintenance, l'IDS ne peut pas détecter les nouvelles attaques.

b) **Le système de reconnaissance de motifs :** le système de reconnaissance de motifs est chargé d'identifier les motifs présents dans la base de signature, dans le flux d'événements. Différents systèmes de reconnaissance de motifs ont été définis dans la littérature, cela va de systèmes simples à base de règles, ou de correspondances de chaînes de caractères comme dans **[20]** (string matching) à des systèmes bien plus complexes à base de systèmes experts comme **[21]** ou de modélisation d'états comme**[17]**.

Les deux principaux avantages de solutions suivant l'approche par signature sont la pertinence des alertes et une certaine facilité de mise en place de l'IDS. Cependant, cette approche présente certaines limites, dont la plus gênante, est de ne pouvoir détecter que les attaques connues.

II.3.4.2-L'analyse comportementale (d'anomalie)

Comme première approche proposée et développée, Anderson **[13]** propose de détecter des violations de la politique de sécurité du système, en observant le comportement des utilisateurs et en le comparant à un modèle du comportement considéré comme normal, appelé profil.

D'une manière générale, l'approche comportementale comporte deux phases :

- ✓ Phase d'apprentissage où le profil est constitué en observant le comportement de l'entité surveillée.
- ✓ Phase de détection pendant laquelle l'IDS observe le comportement de l'entité, mesure la similarité entre ce dernier et le profil puis émet une alerte si la déviation est trop importante.

L'idée principale de cette approche est de considérer toute anomalie dans le comportement comme une intrusion.

Le modèle de comportement normal, est dit correct, s'il ne modélise que le comportement légitime, du point de vue de la politique de sécurité, de l'entité surveillée. Toutes les intrusions sont alors détectées par l'IDS : il n'y a pas de faux négatif.

Le modèle de comportement normal, est dit complet, s'il modélise entièrement le comportement légitime, du point de vue de la politique de sécurité, de l'entité surveillée. Dans ce cas, toutes les alertes correspondent à des intrusions : il n'ya pas de faux positifs.

Plusieurs méthodes de modélisation, ont été proposées, pour établir le profil de l'entité surveillée : modèles statistiques **[22] [23]**, systèmes experts **[26]**, réseaux de neurones **[24]**, approche immunologique **[25]**.

- ✓ **Modèles statistiques :** Dans cette approche, le profil est établi, en observant, la valeur de certains paramètres du système considéré comme des variables aléatoires. Pour chaque paramètre du système, un modèle statistique est utilisé pour établir la distribution de la variable aléatoire correspondante. Une fois le modèle établi, un vecteur distance est calculé entre le flux d'événements observés et le profil. Si la distance dépasse un certain seuil, une alerte est émise.

Les premiers IDS utilisant un modèle statistique comme, par exemple, celui proposé par Anderson **[13]** ou IDES **[22]** (Intrusion Detection Expert System) ciblent la détection de comportements anormaux d'utilisateurs, en étudiant des paramètres, comme le temps processeur, la durée des sessions, le nombre de tentatives de login, etc.

NIDES **[23]** (Next Intrusion Detection Expert System) propose des améliorations par rapport à IDES en combinant les approches comportementales et par signature, et étend cette approche à la modélisation statistique d'applications.

- ✓ **Systèmes experts :** Dans cette approche, le profil est établi, en observant le flux d'événements pour en déduire un certain nombre de règles, qui décrivent le comportement normal du système. Pendant la phase de détection, le système applique les règles au flux d'événements et, vérifie si ce flux d'événements respecte ou non les règles apprises.

Ce modèle est utilisé dans le projet W&S **[26]** (Wisdom and Sense) pour détecter des comportements inhabituels chez les utilisateurs d'un système d'information.

- ✓ **Réseaux de neurones :** Un réseau de neurones, peut être utilisé pour modéliser le comportement du système, et apprendre à classifier les comportements normaux et anormaux. Ce modèle, permet de prendre naturellement en compte, des séries temporelles, soit avec des réseaux de neurones récurrents ou des techniques de fenêtre glissantes en entrée du réseau de neurones.

C'est la méthode employée dans **[24]** pour modéliser le comportement des utilisateurs d'un système informatique. Le réseau de neurones, est utilisé en conjonction avec un système expert pour décider de la présence d'intrusions ou non.

- ✓ **Approche Immunologique :** Cette méthode a été proposée par Forrest et al. **[27]** et vise à détecter les comportements anormaux d'applications, en observant les séquences d'appels système, qu'effectue l'application surveillée.

L'avantage majeur de l'approche comportementale, par rapport à l'approche par signature, est de ne pas chercher à caractériser les intrusions, mais le comportement attendu du système, et donc, de pouvoir détecter des intrusions inconnues.

De manière générale, les IDS fondés sur cette approche, sont fiables, car une intrusion génère souvent une anomalie dans le comportement observé. Cela reste cependant une question ouverte, comme le font remarquer Myers **[28]** et Anderson **[13]** et comme le montrent les développements récents dans le domaine des mimicry attacks **[29] [30]**.

De plus, la phase d'apprentissage présente quelques problèmes : il faut s'assurer que la base d'apprentissage soit exempte d'intrusions. Dans le cas contraire, l'IDS risquerait d'apprendre des comportements intrusifs, et ne serait donc pas capable, de les détecter ensuite.

II.4- Corrélation d'alertes en détection d'intrusion

Les IDSs sont bien connus pour générer de grandes quantités d'alertes dont la plupart sont fausses et redondantes. Ce problème est dû à plusieurs raisons telles que des paramètres inappropriés pour les IDSs, etc. **[31]**. Afin de faire face à de telles quantités d'alertes, des approches de corrélation d'alerte ont été proposées **[32]**.

II.4.1- La corrélation d'alertes

La corrélation d'alertes d'après **[32]** consiste à analyser les alertes générées par un ou plusieurs IDSs et éventuellement par d'autres outils de sécurité afin de fournir une vue *synthétique* et de *haut niveau* des événements malveillants détectés.

Les données en entrée pour les outils de corrélation d'alertes sont collectées auprès de diverses sources telles que les IDSs, les pare-feu, logs de serveurs web, etc.

On peut résumer les principaux objectifs de la corrélation d'alertes dans les points suivants :

a) ***Réduction d'alerte et d'élimination des alertes redondante :*** L'objectif ici est d'éliminer la redondance des alertes par agrégation ou fusion d'alertes similaires **[32]**. En effet, les IDSs génèrent souvent de grandes quantités d'alertes redondantes en raison de la multiplicité des IDSs et la répétitivité de certains événements malveillants tels que les scans, les attaques DoS, DDoS, etc.

b) ***Détection de plans d'attaques :*** La plupart des IDSs ne rapportent que des événements élémentaires alors que plusieurs attaques malveillantes s'étalent sur plusieurs étapes où chacune peut être révélée par une alerte. Détecter les plans attaques requiert l'analyse des relations entre plusieurs alertes.

c) ***Filtrage et priorisation d'alertes :*** Parmi les énormes quantités d'alertes générées, des administrateurs doivent sélectionner un sous-ensemble d'alertes en fonction de leur dangerosité et du contexte de chaque système d'information. Le filtrage et priorisation d'alertes ont pour but de présenter aux administrateurs seulement les alertes qu'ils souhaitent analyser en priorité **[33]**.

II.4.2- Les fonctions principales dans la corrélation

D'après **[34] [35]** on distingue plusieurs fonctions principales dans le processus de corrélation :

- ✓ ***Elimination de la redondance:*** une première fonction d'un système d'agrégation/corrélation d'alertes est de déterminer si deux alertes ont été générées suite à l'observation d'un même événement. Ainsi, l'élimination de la redondance réduit le nombre d'alertes à traiter.
- ✓ ***Agrégation d'alertes :*** certaines attaques produisent plus d'un seul événement élémentaire. Ainsi, le regroupement des événements élémentaires réduit le flux d'alertes, ce qui simplifie l'analyse pour les opérateurs.
- ✓ ***La fusion d'alertes :*** après le regroupement des alertes en clusters, une fonction plus avancée de l'agrégation/corrélation consiste à produire une alerte globale pour chaque groupe. Ces dernières résument les activités malicieuses rapportées par ces groupes d'alertes.
- ✓ ***Reconnaissance des scénarios d'attaque (corrélation de contexte)*****:** cette fonction est encore plus poussée et nécessite des mécanismes plus complexes pour certaines attaques qui sont réalisées en plusieurs étapes. Ainsi, les attaques sont mieux comprises sous forme de scénarios qu'individuellement.

Généralement les approches de corrélation d'alerte sont souvent regroupées en approches fondées basée sur la similarité entre alertes **[32]**, sur le scénario d'attaque prédéfinis **[36]** et approches statistiques **[37]**.

II.5- Conclusion

De nos jours, les systèmes de détections d'intrusions possèdent une grande importance vue leur participation dans les mécanismes de la sécurité des systèmes d'information. Ils sont de plus en plus utilisés dans le monde professionnel afin d'assurer la surveillance des systèmes d'informations.

Les IDSs se caractérisent par la méthode de détection (Signature ou Anomalie), les sources d'information (Hôte ou Réseau), leur comportement (Actif ou Passif), ainsi que la synchronisation entre les sources et le détecteur (Périodique ou Temps réel).

Comme tout autre système, les IDSs possèdent des problèmes qui sont liés à certains facteurs tels que le facteur humain, mais le problème majeur réside dans le taux trop élevé de fausses alertes. Pour faire face à ça on fera recours à la corrélation d’alertes. Il y a plusieurs méthodes et techniques de corrélation :

- ✓ Par des attributs similaires.
- ✓ Par des scénarios d'attaques prédéfinis.
- ✓ Par les prérequis et conséquences des attaques.
- ✓ Par l'analyse statistique de causalité.

Chapitre III :

Réseaux Bayésiens & la détection d'intrusions

Chapitre III : Réseaux Bayésiens & la détection d'intrusions

III.1- Introduction

Durant ces dernières années, de nouveaux formalismes et techniques sont utilisés pour améliorer la détection et faire face à certains problèmes tels que la complexité du problème de détection due à la nature des données à analyser. En effet, les données qu'un administrateur réseau doit analyser sont souvent hétérogènes, d'une taille très importante et contiennent des informations incertaines, incomplètes, imprécises ou manquantes.

Parmi ces nouveaux formalismes, on trouve les modèles graphiques qui par leurs aspects algorithmiques (apprentissage automatique, inférence, etc.) et modulaire ont montré leur efficacité dans le domaine de la détection d'intrusions. Les réseaux bayésiens, les réseaux de neurones sont parmi les modèles graphiques les plus utilisés en détection d'intrusions.

III.2- Réseaux bayésiens

Les réseaux bayésiens **[38] [40]**, également appelés réseaux probabilistes, sont des outils de modélisation de connaissances incertaines et complexes. Ils permettent aussi la représentation des relations d'influences entre ces connaissances. Ils ont été utilisés dans de nombreux domaines tels que dans le diagnostic médical **[41]**, corrélation d'alertes **[42]**, filtrage de spams **[43]**, détection d'intrusions **[42] [44] [39]**, etc.

Ces modèles de raisonnement probabiliste se caractérisent par deux aspects :

- ✓ un aspect graphique ou qualitatif : permettant de représenter d'une manière très simple la connaissance sous forme d'un graphe orienté sans cycles,
- ✓ un aspect probabiliste ou quantitatif : offrant un moyen de quantifier l'incertitude des relations d'influences entre les variables du domaine étudié.

Plus précisément, un réseau bayésien est défini comme un graphe orienté acyclique (DAG) permettant de représenter les dépendances directes ou conditionnelles entre les variables du domaine étudié. Il est muni d'un ensemble de tables de probabilités conditionnelles (CPT) pour quantifier l'incertitude relative aux relations d'influences.

Formellement, étant donné un ensemble de variables aléatoires X = {X1,X2,…, Xn}, β =< G,Θ > est un réseau bayésien où G = (X,E) est un graphe orienté acyclique représentant la structure graphique de β. X est l'ensemble des nœuds où chacun représente une variable aléatoire Xi. A chaque Xi, on associe une table de probabilités locales Θi qui représente les probabilités des valeurs de Xi sachant toutes les valeurs possibles de leurs parents. E est l'ensemble des arcs représentant les relations de dépendance directe entre les différents nœuds du graphe G.

Les réseaux bayésiens permettent de représenter d'une manière compacte une distribution de probabilités jointe associée à l'ensemble des variables en utilisant la notion d'indépendance. Une distribution de probabilité jointe sur n variables binaires est composée de 2^2 entrées. La distribution de probabilités jointe est décomposée sous forme d'un produit des distributions de probabilités locales selon la règle de chaînage.

$$\mathbf{P(X1, X2, \dots, Xn)} = \prod_{\mathbf{i=1}}^{\mathbf{n}} (\boldsymbol{P\,Xi \mid Pa(Xi)})$$

Formule III. 1

Où Pa(X_i) représente l'ensemble des parents de X_i.

La probabilité conditionnelle d'une valeur d'une variable X_i sachant la valeur d'une autre variable X_j peut être calculée par la loi de Bayes de la manière suivante :

$$\boldsymbol{P(\,Xi \mid Xj\,)} = \frac{\boldsymbol{P(\,Xj \mid Xi\,)\,.\,P\,(Xi)}}{\boldsymbol{P(Xj)}}$$

Formule III. 2

Les distributions de probabilités locales doivent satisfaire les conditions de normalisation :

– Si X_i est un nœud sans parent du réseau bayésien alors la distribution locale associée à X_i doit satisfaire la condition suivante :

$$\forall\, \boldsymbol{x_i} \in \boldsymbol{Dx_i}, \sum_{\boldsymbol{xi}} \boldsymbol{P(X = x_i) = 1}$$

Formule III. 3

– Si X a des parents dans le réseau bayésien, alors la distribution conditionnelle associée à X doit satisfaire :

$$\sum_{\boldsymbol{x_i}} \boldsymbol{P(X = x_i | Pa(X) = y) = 1}$$

Formule III. 4

III.2.1- Apprentissage des réseaux bayésiens

L'apprentissage d'un réseau bayésien consiste à définir la structure graphique et associer des tables de probabilités conditionnelles à chaque variable du problème à modéliser. Il s'agit donc d'apprentissage de structures et de paramètres.

III.2.1.1- Apprentissage de structures

L'apprentissage de structure d'un réseau bayésien consiste à identifier les nœuds et les relations possibles entre ces nœuds à partir des données d'apprentissage. La recherche de structure de réseaux bayésiens est un problème difficile **[45]**, principalement à cause du fait que l'espace de recherche est exponentiel en fonction du nombre de variables décrivant le domaine. C'est pourquoi, de nombreux algorithmes d'apprentissage automatique ont été proposés : algorithme de recherche de causalité **[46]**, algorithme de poids minimum **[47]**, l'algorithme K2 **[48]**, consistent k-graphs **[49]**, etc. La recherche de la meilleure structure d'un réseau bayésien peut se faire par exemple en parcourant tous les graphes possibles, de leur associer une valeur quantitative (score), puis de choisir le graphe ayant le score le plus élevé **[48]**.

III.2.1.2- Apprentissage de paramètres

L'apprentissage de paramètres d'un réseau bayésien consiste à associer une table de probabilités locales à chaque nœud de la structure du réseau préalablement élaborée ou

apprise. Les paramètres peuvent être donnés directement par l'expert (ses connaissances subjectives) ou calculés à partir de données d'apprentissage. En présence d'un ensemble de données d'apprentissage et de la structure du réseau, il est simple de calculer les probabilités conditionnelles. Le calcul peut se faire de deux manières différentes selon la nature des données (données complètes ou incomplètes) :

- ✓ Concernant le cas où toutes les données sont complètes (données observées), les probabilités sont calculées sur la base des fréquences qui représentent le nombre d'apparitions de chacune des valeurs que le nœud peut prendre. Il s'agit dans ce cas d'un apprentissage statistique basé sur le maximum de vraisemblance. D'autres méthodes peuvent être également utilisées. Nous citons par exemple des méthodes qui se basent sur des estimations bayésiennes (maximum a posteriori et espérance a posteriori).
- ✓ Lorsque les données sont incomplètes, c'est-à-dire que les variables sont complètement manquantes ou ne sont observées que partiellement, plusieurs traitements sont possibles selon la nature des données. Nous citons par exemple la méthode basée sur l'analyse des exemples disponibles qui consiste à calculer seulement les probabilités des variables Xi et les probabilités de leur parent Pa (Xi). Ainsi, pour estimer P(Xi|Pa(Xi)), il suffit d'utiliser tous les exemples où Xi et Pa(Xi) sont complètement observées.

III.2.2- Inférence dans les réseaux bayésiens

L'inférence dans un réseau bayésien concerne le calcul de la probabilité de n'importe quelle variable ou sous ensemble de variables à partir des autres variables observées. Il s'agit donc de déterminer les probabilités conditionnelles d'événements reliés par des relations d'influences. Les algorithmes d'inférence dans les réseaux bayésiens se répartissent en deux groupes : algorithmes d'inférence exacte et algorithmes d'inférence approchée. Les algorithmes d'inférence exacte exploitent les indépendances conditionnelles contenues dans le réseau pour calculer les probabilités a posteriori exactes. Concernant la deuxième catégorie d'algorithmes, les méthodes utilisées donnent des estimations approchées des probabilités a posteriori **[50]**.

III.2.3- Classification dans les réseaux bayésiens

La classification est considérée comme un cas particulier d'inférence : une seule variable, dite variable de classe que nous symbolisons par *C* et les autres variables notées ***A****i*, constituent les attributs. A partir des valeurs des attributs, la classification rend comme résultat la classe ayant la plus grande probabilité a posteriori P(*ci*=A).

Comme exemples de classifieurs bayésiens, on trouve : classifieur naïf de Bayes (Naive Bayes) **[51]**, classifieur bayésien naïf augmenté TAN (Tree Augmented Naive Bayes) **[52]**, classifieur BAN (Bayesian Network Augmented Naive Bayes) **[51]**, les BMN (Multi-Nets Bayesians) **[53]**.

III.3- Détection d'intrusions, techniques d'apprentissage automatique et classification

La détection d'intrusions peut être vue ou traitée comme un problème de classification **[54]**. Il s'agit d'analyser les évènements ayant lieu au sein du système d'informations dans le but de les classifier ou les identifier selon la nature des activités qu'ils comportent (normales ou anormales). Cette approche est très prometteuse car les modèles de classification peuvent être construits directement à partir de données représentant des activités normales et anormales. Une fois le modèle construit, il sera utilisé pour classer des nouveaux événements.

Les techniques de classification permettent de déterminer l'appartenance d'un objet à une catégorie prédéfinie. Ces techniques consistent à élaborer sur les données d'apprentissage, un modèle de classification pouvant être généralisé sur l'ensemble des données du problème.

En détection d'intrusions, un objet correspond à un événement qui peut être une connexion ou un paquet caractérisé par un vecteur d'attributs (protocole, durée, flag, service, etc.). La classe de l'événement est soit normale, soit anormale (nom de l'attaque).

Les réseaux bayésiens sont largement utilisés pour ce type de problèmes. Parmi les travaux, nous citons par exemple les travaux de Lee **[54]** où des techniques de fouilles de données (telles que les règles d'associations) sont utilisées sur la base de données KDD099 **[55]** dans le but de construire des modèles permettant de distinguer entre activités normales et suspectes. Dans **[39]**, les auteurs ont expérimenté les réseaux bayésiens naïfs **[51]** et les arbres

de décisions **[56]** sur la base de données KDD099, et ils ont montré que ces deux algorithmes donnent des résultats satisfaisants.

Dans **[57]**, un schéma d'hybridation exploitant les performances de chaque algorithme (réseau bayésien naïf et arbre de décisions) a été proposé ayant pour résultat une amélioration significative dans les taux de classifications globales et les taux de détection des attaques R2L et U2R sur la base KDD099. Une nouvelle approche basée sur l'apprentissage supervisé afin de détecter des nouvelles attaques a été proposée dans **[58]**. Plusieurs d'autres travaux existent dans cette thématique de recherche, nous citons par exemple les travaux suivants **[59] [60]**.

III.4- Application des réseaux bayésiens pour la corrélation d'alertes

Peu de travaux ont appliqué les RB à la corrélation d'alertes **[61]**, une approche basée sur un réseau Bayésien est utilisée dans **[37]** pour la fusion d'alertes.

Récemment, des classificateurs Bayésiens étaient utilisés comme dans **[62] [63]** où les auteurs utilisent principalement des modèles naïfs et TAN pour la détection des plans d'attaques.

Salem Benfarhat et Tayeb Kenaza dans **[1]** ont proposé une approche plus facile à mettre en œuvre et qui ne nécessite pas une grande contribution des connaissances d'experts, en effet dans cette dernière, les scénarios d'attaque sont obtenus automatiquement (ils n'avaient pas besoin de déterminer a priori l'ensemble des actions impliquées dans les scénarios).

III.5- La modélisation Bayésienne pour la détection d'intrusions

III.5.1- Scénario d'attaque

Un scénario d'attaque est défini comme étant un ensemble S = {A1, A2, ..., An, O}, dont :

Les Ai représentent des instances d'actions et O, peut être une phase ou l'objectif d'intrusion. Ai à une influence positive sur O et plus précisément, une action A à une influence positive sur un objectif d'intrusion O si P(O|A) >P(O).

L'objectif de notre travail est de détecter les scénarios d'attaque le plutôt possible et de prévoir ceux qui sont les plus plausibles. On s'intéresse pas à déterminer l'ordre exact dans lequel un ensemble d'actions a été exécuté de manière à atteindre différentes phases mais de déterminer l'ordre de ces phases qui mènent à des objectifs d'intrusions.

III.5.2- Modélisation du D-DoS et évaluation sur les données DARPA 2000

L'évaluation faite par DARPA en 2000 comporte deux scénarios d'attaque de type Déni de Service Distribué (DDoS). Notre modèle sera illustrée sur le premier scénario, appelé LLDDoS 1.0 (Lincoln Laboratory Scenario (DDoS) 1.0). Ce scénario est conçu pour être mené par des attaquants débutants en utilisant des scripts prédéfinis. Il est divisé en 5 phases et exécuté sur 3 réseaux : un sous-réseau local privé, une DMZ (Zone démilitarisée) et le réseau public (Internet).

L'objectif de l'attaque est de permettre aux attaquants débutants de pénétrer dans plusieurs sites à travers Internet, installer les outils (programmes) nécessaires pour le DDoS et lancer le DDoS contre des sites gouvernementaux. Cette attaque exploite la vulnérabilité de l'outil Remote-To-Root Sadmind (outil d'administration à distance sur des stations de travail Solaris) pour obtenir un accès root sur les trois hôtes Solaris du site de l'Air Force Base (AFB). L'attaque se déroule en 5 phases :

1. Balayage IP du site AFB à partir d'un site distant afin de connaitre les adresses IP en service.

2. Scan de ports sur les adresses relevés depuis la phase 01 à la recherche des adresses IP des ordinateurs Solaris exécutant le service Sadmind.

3. Compromission des ordinateurs via une vulnérabilité dans Sadmind.

4. Installation du « trojan mstream » DDoS sur les trois ordinateurs du site AFB.

5. Lancement du DDoS.

Les données DARPA'2000 contiennent un trafic réseau brute capturé par un analyseur de trafic réseau (SNORT) pendant l'exécution du scénario d'attaque. Il nous faut maintenant faire le prétraitement afin de déterminer les actions de ce scénario.

III.5.3- Prétraitement des observations et sélection des attributs

Après l'analyse des données DARPA'2000 avec l'outil WIRESHARK, nous avons constaté que les actions générées par le hacker sont comme illustré sur le Tableau III.1 :

Action	*Description*
A1 : ping_ICMP	Message ICMP-Request demandant si une machine est en marche
A2 : rpc_sadmind_ping	Requête RPC (Remote Procedure Call) demandant le port d'exécution de l'outil Sadmind
A3 : ping_sadmind	Requête vérifiant l'existance de Sadmind
A4 : sadmind_root_query	Requête au serveur Sadmind avec des privilèges administrateur
A5 : sadmind_bof	Tentative d'attaque par débordement de tampon (buffer overflow) sur Sadmind
A6 : ICMP_reply	Message ICMP-Reply (réponse au ping) confirmant que la machine est en marche
A7 : telnet_info	Ouverture d'une session Telnet avec succès
A8 : telnet_login_incorrect	Tentative d'ouverture d'une session Telnet avec échec
A9 : telnet_bad_login	La troisième tentative d'ouverture d'une session Telnet avec echec
A10 : rsh_root	Ouverture d'une session RSH (remote shell) avec succès
A11: icmp_port_unreachable	Réponse négative au ping

Tableau III.1 : Les actions des données DARPA'2000

III.5.3.1- Le modèle bayésien proposé

A partir de ces données et notre connaissance préalable sur le corpus on a pu proposer le modèle bayésien du premier scénario DARPA 2000 comme l'illustre la ***Figure III.1.***

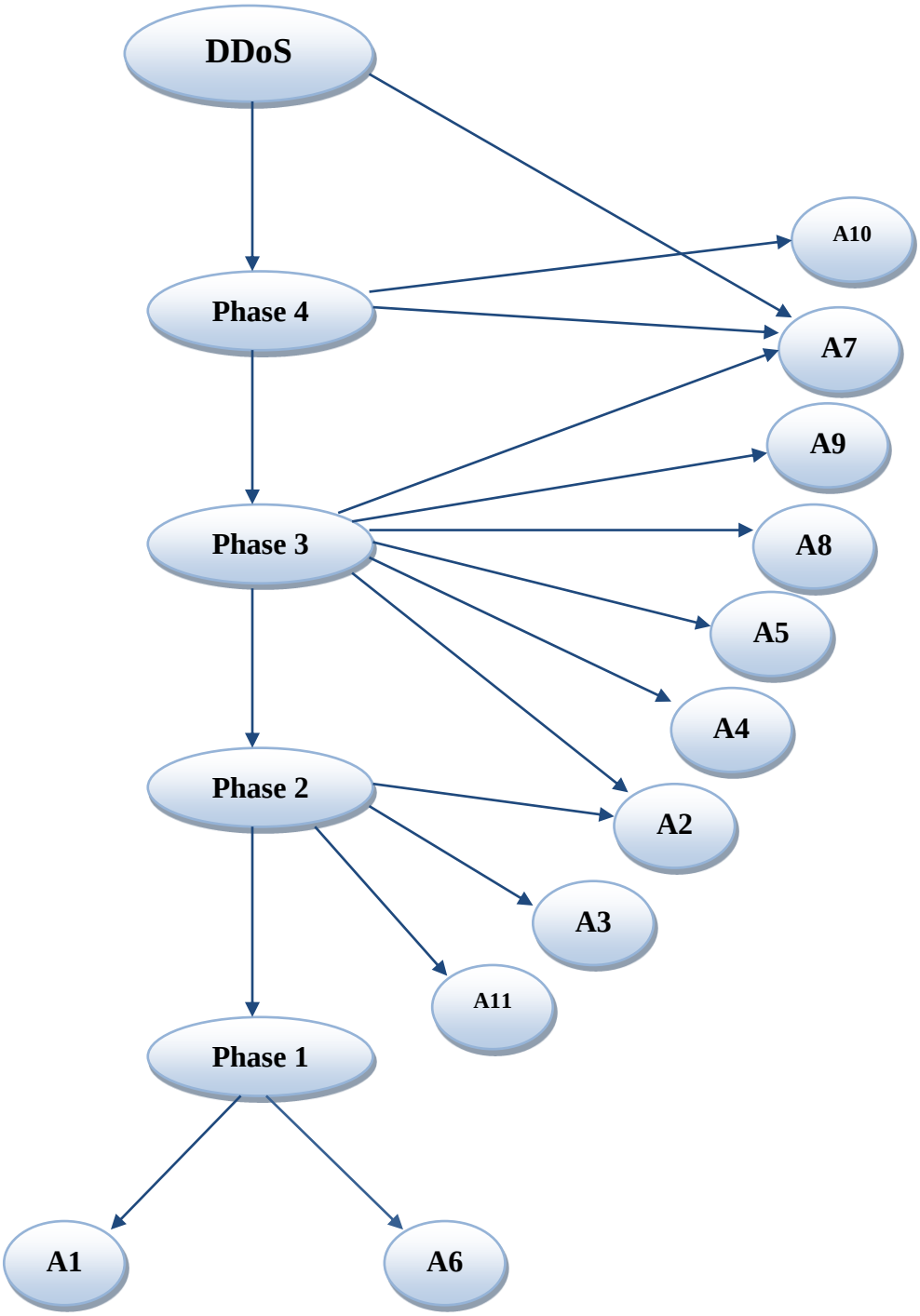

Figure III.1 : Le modèle bayésien proposé.

III.5.4- Apprentissage et prédiction des objectifs d'intrusion

Les distributions de probabilités conditionnelles de chaque variable étant donnée la classe (action sachant phase, phase sachant phase suivante) sont d'abord appris à partir des données obtenues lors de la première étape.

1. *Détermination de la prédiction*

Pour déterminer la prédiction, il faut calculer les probabilités suivantes :

- ✓ $\mathrm{P}\ (\mathrm{P1}|\mathrm{A1}, \mathrm{A6})$
- ✓ $\mathrm{P}(\mathrm{P2}|\mathrm{P1}, \mathrm{A2}, \mathrm{A3}, \mathrm{A11})$
- ✓ $\mathrm{P}(\mathrm{P3}|\mathrm{P2}\ , \mathrm{A2}, \mathrm{A4}, \mathrm{A5}, \mathrm{A1}, \mathrm{A8}, \mathrm{A9})$
- ✓ $\mathrm{P}(\mathrm{P4}|\mathrm{P3}\ , \mathrm{A7}, \mathrm{A10})$
- ✓ $\mathrm{P}(\mathrm{DDOS}|\mathrm{P4}, \mathrm{A7})$

a. Phase une

$$\mathrm{P}(\mathrm{P1}, \mathrm{P2}, \mathrm{A1}, \mathrm{A6}) = \mathrm{P}(\mathrm{A1})\ \mathrm{P}(\mathrm{A6}|\mathrm{A1})\ \mathrm{P}(\mathrm{P1}|\mathrm{A1}, \mathrm{A6})\ \mathrm{P}(\mathrm{P2}|\mathrm{P1})$$

$$= \mathrm{P}(\mathrm{P1})\ \mathrm{P}(\mathrm{A6})\ \mathrm{P}(\mathrm{P1}|\mathrm{A1}, \mathrm{A6})\mathrm{P}(\mathrm{P2}|\mathrm{P1})$$

$$\mathrm{P}(\mathrm{P2}, \mathrm{P1}\ , \mathrm{A1}, \mathrm{A6}) = \mathrm{P}(\mathrm{P1})\mathrm{P}(\mathrm{A1}|\mathrm{P1})\mathrm{P}(\mathrm{A6}|\mathrm{P1})\ \mathrm{P}(\mathrm{P2}|\mathrm{P1})$$

$$\mathrm{P}(\mathrm{P1}|\mathrm{A1}, \mathrm{A6}) = \frac{\mathrm{P}(\mathrm{P1})\mathrm{P}(\mathrm{A1}|\mathrm{P1})\mathrm{P}(\mathrm{A6}|\mathrm{P1})\mathrm{P}(\mathrm{P2}|\mathrm{P1})}{\mathrm{P}(\mathrm{A1})\mathrm{P}(\mathrm{A6})\mathrm{P}(\mathrm{P2}|\mathrm{P1})}$$

$$\mathrm{P}(\mathrm{P1}|\mathrm{A1}, \mathrm{A6}) \cong \frac{\mathrm{P}(\mathrm{P1})\mathrm{P}(\mathrm{A1}|\mathrm{P1})\mathrm{P}(\mathrm{A6}|\mathrm{P1})}{\mathrm{P}(\mathrm{A1})\mathrm{P}(\mathrm{A6})}$$

On remarque que si on fixe les valeurs A1 et A6, la valeur de P1 que ce soit pour zéro ou pour un, le dénominateur reste invariant, de ce fait la comparaison se fera au niveau du numérateur :

$$\mathbf{P(P1|A1, A6)} \cong \mathbf{P(P1)P(A1|P1)P(A6|P1)}$$

Formule III. 5

b. Phase deux :

$$\mathrm{P(P2, P1, A2, A3, A11)} = \mathrm{P(A2)\ P(A3|A2)P(A11|A3, A2)P(P1|A11, A3, A2)\ P(P2|P1, A3, A11, A2)}$$

$$= \mathrm{P(P2|P1, A3, A11, A2)\ P(P1)\ P(A2)\ P(A3)\ P(A11)}$$

$$= \mathrm{P(P2|P1, A3, A11, A2)\ P(P1)\ P(A2)\ P(A3)\ P(A11)}$$

$$\mathrm{P(P2, P1, A2, A3, A11)} = \mathrm{P(P2)\ P(P1|P2)\ P(A11|P2)\ P(A2|P2)\ P(A3|P2)}$$

$$\mathrm{P(P2|P1, A2, A3, A11)} = \frac{\mathrm{P(P2)P(P1|P2)P(A11|P2)\ P(A2|P2)\ P(A3|P2)}}{\mathrm{P(P1)P(A2)\ P(A3)\ P(A11)}}$$

Même principe que la fonction précédente, On obtient :

$$\mathbf{P(P2|P1, A2, A3, A11)} \cong \mathbf{P(P2)P(P1|P2)P(A11|P2)P(A3|P2)P(A2|P2)}$$

Formule III. 6

En suivant les mêmes étapes, on a obtenu :

c. Phase trois

$$\mathbf{P(P3|P2, A2, A4, A5, A7, A8, A9)}$$
$$\cong \mathbf{P(P3)P(P2|P3)P(A2|P3)P(A4|P3)P(A5|P3)P(A7|P3)P(A8|P3)P(A9|P3)}$$

Formule III. 7

d. Phase quatre

$$\mathbf{P(P4|P3, A7, A10) \cong P(P4)P(P3|P4)P(A7|P4)P(A10|P4)}$$

Formule III. 8

e. Phase cinq

$$\mathbf{P(DDoS|P4, A7) \cong P(DDoS)P(P4|DDoS)P(A7|DDoS)}$$

Formule III. 9

De a, b, c, d, e, on déduit que :

$$\mathbf{P(Pn|Pn-1, A1..Ak) \cong P(Pn)\ P(Pn-1|Pn) \prod_{i=1}^{k} P(Ai|Pn)}$$

Formule III. 10

Il reste de déterminer les probabilités a priori P(Pn).

2. *Calcul des probabilités a priori :*

$$\mathbf{P(P1, P2) = P\ (P2)\ P\ (\ P1|P2)}$$

Par inférence :

$$\mathbf{P(P1) = \sum_{p2=0}^{1} P(P1, P2) = \sum_{p2=0}^{1} P(P2)\ P(P1|P2)}$$

Dont :

$$\mathbf{P(P2) = \sum_{p3=0}^{1} P\ (P2, P3)\ = \sum_{p3=0}^{1} P(P3)\ P\ (P2|P3)}$$

$$\mathbf{P(P3) = \sum_{p4=0}^{1} P(P3, P4)\ = \sum_{p4=0}^{1} P(P4)\ P(P3|P4)}$$

$$\mathbf{P(P4)} = \sum_{p5=0}^{1} \mathbf{P(P4, DDoS)} \;\; = \sum_{p5=0}^{1} \mathbf{P(DDoS)}\, \mathbf{P(P4|DDoS)}$$

Puisque D-DoS est un (01) a priori :

$$\mathbf{P(DDoS = 0) = P(DDoS = 1) = \; 0,5}$$

De ce fait on pourra calculer les autres probabilités sans difficulté vue que les variables sont calculées précédemment par l'apprentissage statique.

III.6- Conclusion

Les modèles graphiques peuvent traiter des données incomplètes et surtout incertaines, ce qui est souvent le cas en détection d'intrusions. En outre, les modèles graphiques permettent de modéliser plusieurs types de dépendances entre les variables du système. En effet, selon le modèle utilisé, ils permettent de modéliser des dépendances simples, multiples, causales, temporelles, etc.

Dans ce chapitre nous nous sommes intéressés particulièrement aux réseaux bayésiens qui sont considérés parmi les modèles graphiques les plus utilisés, nous avons illustré notre modèle sur le premier scénario de DARPA 2000. Après le prétraitement des données nous avons pu tirer les variables d'intérêt du modèle Bayésien proposé, ces dernières étaient les différentes actions ainsi que les 5 phases de l'attaque D-DoS, on a également appris notre modèle afin de prédire les objectifs d'intrusions.

Chapitre IV :

Implémentation avec L'IDS SNORT

Chapitre IV : Implémentation avec L'IDS SNORT

IV.1- Introduction

Ce chapitre sera consacré à la mise en œuvre de notre IDS et l'application des nouvelles règles obtenues de la traduction du modèle bayésien. On va décrire le langage d'écriture de règles propre à Snort, puis nous verrons aussi comment configurer Snort, et comment l'utiliser.

IV.2- Outil de travail

Dans notre travail on a fait recours aux outils suivants, voici une brève description :

IV.2.1- VMware Workstaion

C'est la version station de travail du logiciel VMware. Il permet la création d'une ou plusieurs machines virtuelles au sein d'un même système d'exploitation (généralement Windows ou Linux), ceux-ci pouvant être reliés au réseau local avec une adresse IP différente, tout en étant sur la même machine physique (machine existant réellement). Il est possible de faire fonctionner plusieurs machines virtuelles en même temps, la limite correspondant aux performances de l'ordinateur hôte.

IV.2.2- Linux (distribution Ubuntu-13.10 Desktop)

Ubuntu est un système d'exploitation libre commandité par la société Canonical et une marque déposée par cette même société (Ubuntu).

Fondé sur la distribution Linux Debian, ce système d'exploitation est constitué de logiciels libres et il est disponible gratuitement, y compris pour les entreprises, selon un principe lié à la philosophie affichée du projet.

IV.2.3- SNORT

Snort est un système de détection d'intrusion (IDS) libre publié sous licence GNU GPL. À l'origine écrit par Martin Roesch, il appartient actuellement à Sourcefire. Des versions commerciales intégrant du matériel et des services de supports sont vendus par Sourcefire.

Snort est un des plus actifs NIDS Open Source et possède une communauté importante contribuant à son succès.

Snort est capable d'effectuer aussi en temps réel des analyses de trafic et de logger les paquets sur un réseau IP. Il peut effectuer des analyses de protocole, recherche/correspondance de contenu et peut être utilisé pour détecter une grande variété d'attaques comme des dépassements de tampons, scans, et bien plus.

Pour effectuer ces analyses, Snort se fonde sur des règles. Celles-ci sont écrites par Sourcefire ou bien fournies par la communauté de Snort. Il est fourni avec certaines règles de base mais cependant, comme tout logiciel, Snort n'est pas infaillible et demande donc une mise à jour régulière.

IV.3- Plan de travail

On va installer la version Linux de Snort sous une distribution Ubuntu, pour ce faire on a procédé comme suit :

- ✓ A l'aide de VMware on a créé une machine virtuelle puis installé le système d'exploitation Linux Ubuntu-13.10 Desktop 32 bits.
- ✓ Installation de Snort et quelques outils d'administration (consulter l'annexe pour plus de détail).
- ✓ Configuration de Snort.

IV.4- L'écriture de règles Snort

Comme on a déjà cité que Snort analyse le trafic à la base de règles de filtrage, nous allons expliquer comment écrire ces règles.

IV.4.1- Les base

Snort utilise un langage simple et léger de description de règles qui est flexible et assez puissant. Il y a quelques indications simples à se souvenir en développant des règles Snort, la première est que les règles Snort doivent être complètement contenues sur une seule ligne, l'analyseur de règles de Snort ne sait pas comment traiter des règles sur plusieurs lignes.

Les règles Snort sont divisées en deux sections logiques, l'entête de la règle et les options de la règle. L'entête de règle contient comme informations l'action de la règle, le protocole, les adresses IP source et destination et les masques réseau, et les ports source et destination. La section options de la règle contient les messages d'alerte et les informations sur les parties du paquet qui doivent être inspectées pour déterminer si l'action de la règle doit être acceptée. Voici une règle d'exemple :

```
Alert tcp any any -> 192.168.1.0/24 111 (content: "|00 01 86 a5|"; msg: "mountd access";)
```

Le texte jusqu'à la première parenthèse est l'entête de règle et la section entourée par les parenthèses est les options de la règle. Les mots avant les deux points dans la section des options de la règle sont appelés les mots clés des options. Notez que la section des options de la règle n'est spécifiquement requise par aucune règle, elles sont juste utilisées pour le bien de rendre plus strictes les définitions des paquets à collecter ou à alarmer (ou à ignorer, d'ailleurs). Tous les éléments dans cette composition de règle doivent être vrais pour que l'action de règle indiquée soit acceptée. En les prenants ensembles, les éléments peuvent être considérés comme formant la déclaration d'un ET logique. En même temps, les diverses règles dans un fichier bibliothèque de règles Snort peuvent être considérées comme formant une large déclaration d'un OU logique.

IV.4.1.1- Les inclusions

Le mot clé « include » permet à d'autres fichiers de règles d'être inclus dans le fichier de règles indiqué sur la ligne de commande de Snort. Format :

```
Include : <répertoire/nom du fichier include>
```

Notez qu'il n'y a pas de point-virgule à la fin de la ligne. Les fichiers inclus substitueront toute valeur de variable prédéfinie dans le fichier de règles de Snort.

IV.4.1.2- Les variables

Des variables peuvent être définies dans Snort. Ce sont de simples substitutions des variables fixées avec le mot clé var. Pour les ports et les plages d'adresses IP vous prouver utilisés portvar et ipvar. Format :

```
var: <nom> <valeur>
```

Exemple de définition et d'utilisation de variable

```
var MY_NET [192.168.1.0/24,10.1.1.0/24]

alert tcp any any -> $MY_NET any (flags: S; msg: "SYN packet";)
```

Les noms de variables de règles peuvent être modifiés de plusieurs façons. Vous pouvez définir des méta-variables en utilisant l'opérateur "$". Celles-ci peuvent être utilisées avec les opérateurs de modification de variables, "?" et "-".

- ✓ $var - défini la méta-variable
- ✓ $(var) - remplace avec le contenu du variable "var"
- ✓ $(var:-défaut) - remplace avec le contenu du variable "var" ou avec "défaut" si "var" est indéfini.
- ✓ $(var:?message) - remplace avec le contenu de la variable "var" ou affiche le message d'erreur "message" et quitte

Exemple d'utilisation avancée de variable

```
var MY_NET $(MY_NET:-192.168.1.0/24)

log tcp any any -> $(MY_NET:?MY_NET is undefined!) 23
```

IV.4.2- Les entêtes de règle

IV.4.2.1- L'action de règle

L'entête de règle contient l'information qui définit le "qui, où, et quoi" d'un paquet, ainsi que quoi faire dans l'événement où le paquet avec tous les attributs indiqués dans la règle devrait se présenter. Le premier élément dans une règle est l'action de règle. L'action de règle dit à Snort quoi faire quand il trouve un paquet qui correspond aux critères de la règle. Il y a huit (08) actions accessibles par défaut dans Snort, alert, log, pass, activate, et dynamic, drop, reject, sdrop.

- ✓ alert - génère une alerte en utilisant la méthode d'alerte sélectionnée, et alors journalise le paquet
- ✓ log - journalise le paquet
- ✓ pass - ignore le paquet
- ✓ activate - alerte et alors active une autre règle dynamic
- ✓ dynamic - reste passive jusqu'à être activée par une règle activate, alors agit comme une règle log
- ✓ drop – bloqué et journalisé le paquet.
- ✓ reject – bloque, journalise, puis envoie une « TCP reset » si le protocole est TCP ou un « ICMP destination unreachabale » si le protocole est UDP.
- ✓ sdrop – bloqué le paquet et ne le journalise pas.

Vous pouvez aussi définir vos propres types de règles et associer un ou plusieurs plugins de sortie avec eux. Vous pouvez alors utiliser le type de règle comme action dans les règles Snort.

Cet exemple créera un type qui journalisera juste vers tcpdump :

```
ruletype suspicious
{
        type log
        output log_tcpdump: suspicious.log
}
```

Cet exemple créera un type de règle qui journalisera vers syslog et une base de données mysql :

```
ruletype redalert
{
        type alert
        output alert_syslog: LOG_AUTH LOG_ALERT
        output database: log, mysql, user=snort dbname=snort host=localhost
}
```

IV.4.2.2- Les protocoles

Le champ suivant dans une règle est le protocole. Il y a quatre (04) protocoles que Snort analyse actuellement pour des comportements suspicieux.

- ✓ TCP.
- ✓ UDP.
- ✓ ICMP.
- ✓ IP.

Dans le futur il pourra y en avoir plus, tels que ARP, IGRP, GRE, OSPF, RIP, IPX, etc.

IV.4.2.3- Les adresses IP

La section suivante de l'entête de règle s'occupe comme information de l'adresse IP et du port pour une règle donnée. Le mot clé "any" peut être utilisé pour définir n'importe quelle adresse. Snort n'a pas de mécanisme pour fournir de la résolution de nom pour le champ de l'adresse IP dans le fichier de règles. Les adresses sont formées par une simple adresse IP numérique et un bloc CIDR. Le bloc CIDR indique le masque réseau qui doit être appliqué à l'adresse de la règle et à tout paquet qui est testé par rapport à la règle. Un masque de bloc CIDR de /24 indique un réseau de classe C, /16 un réseau de classe B, et /32 indique l'adresse spécifique d'une machine. Par exemple, la combinaison d'adresse/CIDR 192.168.1.0/24 devrait signifier le bloc d'adresses de 192.168.1.1 à 192.168.1.255. Toute règle qui utilise cette désignation pour, disons, l'adresse destination devrait correspondre à toute adresse dans

cet intervalle. Les désignations CIDR nous donne une façon rapide de désigner de larges espaces d'adresses avec juste quelques caractères.

Il y a un opérateur qui peut être appliqué aux adresses IP, l'opérateur de négation. Cet opérateur dit à Snort de correspondre à toute adresse IP sauf celles indiquées par la liste d'adresses IP. L'opérateur de négation est indiqué par un "!". Par exemple, une modification facile à l'exemple initial est de le faire alerter sur tout trafic qui provient de l'extérieur du réseau local avec l'opérateur de négation comme montré dans l'exemple :

```
alert tcp !192.168.1.0/24 any -> 192.168.1.0/24 111 (content: "|00 01 86 a5|"; msg: "external mountd access";)
```

Ces adresses IP de la règle indiquent "tout paquet tcp avec une adresse source ne provenant pas du réseau interne et à destination du réseau interne".

Vous pouvez aussi spécifier des listes d'adresses IP. Une liste IP spécifiée en entourant une liste d'adresses IP et de blocs CIDR séparés par des virgules entre crochets. Pour le moment, une liste IP ne peut pas inclure d'espaces entre les adresses.

```
alert tcp ![192.168.1.0/24,10.1.1.0/24] any -> [192.168.1.0/24,10.1.1.0/24] 111 (content: "|00 01 86 a5|"; msg: "external mountd access";)
```

IV. 4.2.4- Les numéros de ports

Les numéros de ports peuvent être spécifiés de nombre de façons, incluant "any" (tous les ports), des définitions de ports statiques, des intervalles et des négations. Les ports "any" sont les valeurs génériques, signifiant littéralement tous les ports. Les ports statiques sont indiqués par un seul numéro de port, tel que 111 pour portmapper, 23 pour telnet, ou 80 pour http, etc. Les intervalles de ports sont indiqués avec l'opérateur d'intervalle ":".

```
log udp any any -> 192.168.1.0/24 1:1024
```

Journalise le trafic udp provenant de tout port et à destination de ports dans l'intervalle de 1 à 1024.

```
log tcp any any -> 192.168.1.0/24 :6000
```

Journalise le trafic tcp depuis tout port et allant vers les ports inférieurs ou égaux à 6000.

```
log tcp any :1024 -> 192.168.1.0/24 500:
```

Journalise le trafic tcp depuis les ports privilégiés inférieurs ou égaux à 1024 allant vers les ports supérieurs ou égaux à 500.

La négation de port est indiquée en utilisant l'opérateur de négation "!". L'opérateur de négation peut être appliqué à tous les autres types de règles (excepté "any", qui se traduirait en aucun, combien ce serait Zen...). Par exemple, si pour quelque raison tordue vous voulez tout journaliser sauf les ports X Windows, vous pourriez faire quelque chose comme la règle suivante :

```
log tcp any any -> 192.168.1.0/24 !6000:6010
```

IV. 4.2.5- L'opérateur de direction

L'opérateur de direction "->" indique l'orientation, ou la "direction", du trafic auquel la règle s'applique. L'adresse IP et les numéros de ports du côté gauche de l'opérateur de direction est considéré comme le trafic provenant du système source, et les informations d'adresse et de port du côté droit de l'opérateur est le système destination. Il y a aussi un opérateur bidirectionnel, qui est indiqué par le symbole "<>". Ceci dit à Snort de considérer les paires adresse/port ou bien en source ou bien en destination de l'orientation. C'est utile pour enregistrer/analyser les deux côtés d'une conversation, tel que des sessions Telnet ou POP3. Un exemple de l'opérateur bidirectionnel étant utilisé pour enregistrer les deux côtés d'une session Telnet :

```
log !192.168.1.0/24 any <> 192.168.1.0/24 23
```

IV.4.2.6- Les règles activate/dynamic

La paire de règle activate/dynamic donne à Snort une capacité puissante. Vous pouvez maintenant avoir une règle qui en active une autre pour un nombre fixé de paquets quand son action est accomplie. Ceci est très utile si vous voulez configurer Snort pour effectuer l'enregistrement de ce qui suit quand une règle spécifique "se désactive". Les règles d'activation se comportent juste comme les règles alert, excepté qu'elles ont un champ d'option *obligatoire* : "activates". Les règles dynamiques se comportent juste comme les règles log, mais elles ont un champ d'option différent : "activated_by". Les règles dynamiques ont également un second champ obligatoire, "count". Quand la règle "activate" se désactive, elle active la règle dynamique qui lui est liée (indiquée par les numéros d'option activates/activated_by) pour "count" paquets (50 dans ce cas).

activate tcp !$HOME_NET any -> $HOME_NET 143 (flags: PA; content: "|E8C0FFFFFF|\bin|; activates: 1; msg: "IMAP buffer overflow!";)

dynamic tcp !$HOME_NET any -> $HOME_NET 143 (activated_by: 1; count: 50;)

Ces règles disent à Snort d'alerter quand il détecte un débordement de tampon dans IMAP et collecte les 50 prochains paquets destinés au port 143 provenant de l'extérieur de $HOME_NET et destinés à $HOME_NET. Si le débordement de tampon arrive et a réussi, il y a de très bonnes possibilités que des données utiles seront contenues dans les prochains 50 (ou quel que soit) paquets allant au même port de service sur le réseau, ainsi il y a avantage à collecter ces paquets pour une analyse ultérieure.

IV.4.3- Les options de règle

Les options de règle forment le cœur du moteur de détection d'intrusion de Snort, combinant facilité d'utilisation, puissance et flexibilité. Toutes les options de règle de Snort sont séparées les unes des autres par le caractère point-virgule ";" et les arguments d'une option sont séparés par la virgule ",". Les mots clés des options de règle sont séparés de leurs arguments avec un caractère deux points ":". Parmi les mots clé d'option de règle disponibles dans Snort on trouve :

- ✓ msg - affiche un message dans les alertes et journalise les paquets.
- ✓ logto - journalise le paquet dans un fichier nommé par l'utilisateur au lieu de la sortie standard.
- ✓ ttl - teste la valeur du champ TTL de l'entête IP.
- ✓ tos - teste la valeur du champ TOS de l'entête IP.
- ✓ id - teste le champ ID de fragment de l'entête IP pour une valeur spécifiée.
- ✓ ipoption - regarde les champs des options IP pour des codes spécifiques.
- ✓ fragbits - teste les bits de fragmentation de l'entête IP.
- ✓ dsize - teste la taille de la charge du paquet contre une valeur.
- ✓ flags - teste les drapeaux TCP pour certaines valeurs.
- ✓ seq - teste le champ TCP de numéro de séquence pour une valeur spécifique.
- ✓ ack - teste le champ TCP d'acquittement pour une valeur spécifiée.
- ✓ itype - teste le champ type ICMP contre une valeur spécifiée.
- ✓ icode - teste le champ code ICMP contre une valeur spécifiée.
- ✓ icmp_id - teste la champ ICMP ECHO ID contre une valeur spécifiée.
- ✓ icmp_seq - teste le numéro de séquence ECHO ICMP contre une valeur spécifique.
- ✓ content - recherche un motif dans la charge d'un paquet.
- ✓ content-list - recherche un ensemble de motifs dans la charge d'un paquet.
- ✓ offset - modifie l'option "content", fixe le décalage du début de la tentative de correspondance de motif.
- ✓ depth - modifie l'option "content", fixe la profondeur maximale de recherche pour la tentative de correspondance de motif.
- ✓ nocase - correspond à la procédure de chaîne de contenu sans sensibilité aux différences majuscules/minuscules.
- ✓ session - affiche l'information de la couche applicative pour la session donnée.
- ✓ rpc - regarde les services RPC pour des appels à des applications/procédures spécifiques.
- ✓ resp - réponse active (ferme les connexions, etc).
- ✓ react - réponse active (bloque les sites web).

  Pour plus de détail consulter le manuel de Snort (section 3.4).

IV.5- Configuration de Snort

Snort possède un fichier de configuration « *snort.conf* » qui se trouve dans le répertoire « Répertoire D'installation/*etc/* ». Ce fichier contient une simple configuration, il suffit de modifier ce fichier pour le rendre adéquat à nos besoins. Les modifications possible peuvent se résumé comme suit :

- ✓ Définir les variables du réseau tel que l'adresse IP du réseau à protéger, les adresses IP des serveurs existant sur notre réseau (DNS, WEB, FTP, etc.) ainsi que les ports.
- ✓ Configurer le décodeur.
- ✓ Configurer le moteur de détection de base.
- ✓ Configurer les bibliothèques chargées dynamiquement.
- ✓ Configurez les préprocesseurs.
- ✓ Configurer les plugins de sortie (dans notre cas on a utilisé Unified2).
- ✓ Personnalisez le jeu de règles.
- ✓ Personnaliser préprocesseur et règle de décodeur ensemble.
- ✓ Personnaliser partagé ensemble de règles d'objet.

IV.5.1- Editer les fichiers de configuration

Sur un terminal, tapez :

sudo gedit Chemin_ver_le_fichier/snort.conf

Sur le fichier Snort.conf on a changé les lignes suivantes

Etape #1: ***var WHITE_LIST_PATH ../rules***

var BLACK_LIST_PATH ../rules

Etape #4: ***dynamicpreprocessor directory /usr/local/lib/snort_dynamicpreprocessor/***

dynamicengine /usr/local/lib/snort_dynamicengine/libsf_engine.so

dynamicdetection directory /usr/local/lib/snort_dynamicrules

Par celles-ci :

Etape #1: ***var WHITE_LIST_PATH /usr/local/snort/rules***

var BLACK_LIST_PATH /usr/local/snort/rules

Etape #4: ***dynamicpreprocessor directory /usr/local/snort/lib/snort_dynamicpreprocessor/***

dynamicengine /usr/local/snort/lib/snort_dynamicengine/libsf_engine.so

dynamicdetection directory /usr/local/snort/lib/snort_dynamicrules

Et dans la partie des plugins de sortie on à ajouter la ligne suivante

Etape #6: ***output unified2: filename snort.u2, limit 128***

Barnyard2 améliore l'efficacité de Snort en réduisant la charge sur le moteur de détection principale. Il lit les fichiers de sortie de journalisation unifiée de Snort et les inscrits dans une base de données. Après l'installation de cet outil, et sur son fichier de configuration « Barnyard2.conf » on va changer les lignes suivantes, sur un terminal tapez :

sudo gedit Chemin_ver_le_fichier/barnyard2.conf

```
config reference_file: /etc/snort/reference.config
config classification_file: /etc/snort/classification.config
config gen_file: /etc/snort/gen-msg.map
config sid_file: /etc/snort/sid-msg.map

#config hostname: thor
#config interface: eth0

#output database: log, mysql, user=root password=test dbname=db host=localhost
```

Par :

```
config reference_file: /usr/local/snort/etc/reference.config
config classification_file: /usr/local/snort/etc/classification.config
config gen_file: /usr/local/snort/etc/gen-msg.map
config sid_file: /usr/local/snort/etc/sid-msg.map

config hostname: localhost
config interface: eth1  <Interface qu'on veut surveiller>

output database: log, mysql, user=snort  password=***** dbname=snort host=localhost
```

IV.5.2- Modes de lancement de Snort

IV.5.2.1- Moniteur réseau

Vous souhaitez utiliser Snort pour capturer et afficher les paquets en temps réel pour surveiller le trafic réseau

```
sudo    /usr/local/snort/bin/snort      – <Options>
```

- ✓ Utilisez l'option-v pour voir les informations d'en-tête de paquet TCP et IP
- ✓ Utiliser l'option-d pour voir les en-têtes de la couche application.
- ✓ Utilisez l'option-e pour voir les en-têtes de couche de liaison de données.

Ainsi que vous pouvez combiner entre les trois options en ligne de commande

IV.5.2.2- Journalisation des paquets

Vous souhaitez utiliser Snort pour journaliser votre trafic réseau à des fichiers en temps réel.

```
sudo /usr/local/snort/bin/snort – <Options> -l <Répertoire_de_journalisatin>
```

IV.5.2.3- IDS

Maintenant il nous reste qu'à essayer de lancer Snort on mode IDS, sur un terminale tapez :

```
sudo /usr/local/snort/bin/snort –c <Fichier de configuration>
```

L'IDS Snort est démarré lorsque vous apercevrais le message "Commencing packet processing". Il existe d'autre arguments que vous pouvez utiliser tel que :

- ✓ -i <interface réseau 1 : interface réseau 2 : interface réseau i> pour surveiller plusieurs interfaces.
- ✓ -D lance Snort en processus de fond (mode Daemon).
- ✓ -p Vous souhaitez capturer et enregistrer les paquets sans mettre l'interface en mode promiscuité

IV.5.3- Lecture des fichiers de capture

Les fichiers de sortie de journalisation peuvent être visualisés avec Barnyard2, sur un terminal tapez :

```
sudo barnyard2 -c <Barnyard2.conf> -o <le(s) fichier(s) de log>
```

IV.5.4- Débogage des règles

Pour tester le fonctionnement de nos règles on utilise la commande suivante :

```
sudo snort -i <interface réseau> -n <nombre de paquets> -c <nom du fichier des règles>
```

IV.5.5- Lancement au démarrage du système

Vous aurez probablement envie de lancer Snort au moment du démarrage et en cours d'exécution en arrière-plan. Pour définir Snort à démarrer automatiquement sur votre machine éditer le fichier « rc.local » avec la commande suivante:

```
sudo    gedit    /etc/rc.local
```

Ensuite, collez le contenu suivant dans le fichier (avant la ligne "exit 0") :

```
ifconfig eth1 up
/usr/local/snort/bin/snort -D -u snort -g snort -c /usr/local/snort/etc/snort.conf -i eth1
/usr/local/bin/barnyard2 -c /usr/local/snort/etc/barnyard2.conf \
        -G /usr/local/snort/etc/gen-msg.map \
        -S /usr/local/snort/etc/sid-msg.map \
        -d /var/log/snort \
        -f snort.u2 \
        -w /var/log/snort/barnyard2.waldo \
        -D
```

Enregistrez le fichier et quittez. Puis redémarrer ou utiliser la commande suivante pour démarrer Snort :

```
sudo    /etc/init.d/rc.local      start
```

IV.6- Conclusion

Dans ce chapitre on a tenté d'interpréter notre modèle bayésien sous forme de règles de Snort, ce qui s'est avéré impossible, car le langage de règles de Snort nous a limiter. On a pu aussi vous montrer comment configurer Snort afin de l'utiliser autant qu'un IDS.

Conclusion

Dans notre travail, on a pu déduire que la sécurisation des réseaux tend vers la diminution des vulnérabilités et les failles, pour ce faire l'application d'une politique de sécurité devient une priorité. Parmi les outils utilisés pour mettre en œuvre cette politique on trouve les Firewall (pare-feu), les scanners, ainsi que les systèmes de détection d'intrusions. Ces derniers sont faciles à mettre en œuvre mais possèdent un problème majeur qui figure dans la corrélation des alertes.

On a abordé le problème de corrélation d'alertes, dans le but de trouver un modèle de détection d'attaques complexes qui se base sur la théorie des réseaux bayésiens. Par la suite nous avons essayé d'implémenter ce modèle sous forme de règles sur le système de détection d'intrusions SNORT, mais l'incompatibilité de notre modèle avec le langage de Snort était incontournable.

Comme perspective nous voulons suggérons d'essayé l'implémentation de notre modèle bayésien comme un préprocesseur sur l'IDS SNORT.

Bibliographie

[1] : S. Benfarhat, T. Kenaza et A. Mokhtari .*Modèles graphiques probabilistes pour la corrélation d'alertes en détection d'intrusions*. PhD thesis, 2011.

[3] : ISO-7498-2. *Information processing systems - open systems interconnection- basic reference model - part 2: Security architecture*. Technical report, International Organization for Standardization, Geneva, Switzerland, 1989.

[4] : F. Ebel, S. Baudru, R. Crocfer, D. Puche, J. Hennecart, S. Lasson, et M. Agé. *Sécurité informatique - Ethical Hacking - Apprendre l'attaque pour mieux se défendre*. Editions ENI, 2006.

[5] : C. Llorens, L. Levier, D. Valois. *Tableaux de bord de la sécurité réseau*. Eyrolles 2ème édition, 2006.

[6] : V. Remazeilles. *La sécurité des réseaux avec CISCO*. Editions ENI, Février 2009.

[7] : A. Serhrouchni et C. Llorens. *Mesure de la sécurité "logique" d'un réseau d'un opérateur de télécommunications*. PhD thesis, Novembre 2005.

[8] : R. Ben younes. *Étude et mise en œuvre d'une méthode de détection d'intrusions dans les réseaux sans- fil 802.11 basée sur la vérification formelle de modèles*. PhD thesis, Décembre 2007.

[10] : *Le grand livre de la sécurité informatique*, février 2004. URL : http://www.securiteinfo.com.

[11] : J. Postel. *RFC 793- Transmission Control Protocol*- Internet Engineering Task Force- www.ietf.org. Standard, 1981.

[12] : R. Bace et P. Mell. *Intrusion Detection Systems* .NIST Special Publication on Intrusion Detection Systems.

[13] : J.P. Anderson. *Computer security threat monitoring and surveillance*. Technical report, James P. Anderson Company, Fort Washington, USA, Avril 1980.

[14] : H. Debar, M. Dacier, et A. Wespi. *Towards a taxonomy of intrusion detection systems.* Computer Networks, Elseiver, 1999.

[15] : A. Phillip, Porras et A. Valdes. Live *traffic analysis of tcp/ip gateways*. Proc. ISOC Symposium on Network and Distributed System Security (NDSS98).San Diego, Mars 1998

[16] : D.E. Denning. *An intrusion detection model*. IEEE Trans. Softw, 1987.

[17] : S.T. Eckmann, G. Vigna, et R. A. Kemmerer. Statl : *an attack language for state-based intrusion detection*. Journal of Computer Security, 2002.

[18] : S.W. Boyd et A.D. Keromytis. P*reventing SQL injection attacks*. In Proceedings of the second International Conference on Applied Cryptography and Network Security (ACNS 2004).Yellow Mountain, Chine, Juin 2004.

[19] : J. Newsome and D. Song. *Dynamic taint analysis for automatic detection, analysis, and signature generation of exploits on commodity software*. In Proceedings of the 12th Annual Network and Distributed System Security Symposium (NDSS 2005). San Diego, Févirer 2005.

[20] : B. Mukherjee, L.T. Heberlein et K.N. Levitt. *Network intrusion detection*. IEEE Network, Mai-Juin 1994.

[21] : P.A. Porras et P.G. Neumann. *Event monitoring enabling responses to anomalous live disturbances.* In Proc. of the 20th National Information Systems Security Conferenc. Baltimore, Octobre 1997.

[22] : T.F. Lunt, A. Tamaru, F. Gilham, R. Jagannathan, C. Jalali, H.S. Javitz, A. Valdes, et P.G. Neumann. *A real-time intrusion-detection expert system.* Technical report, SRI International, Juin 1990.

[23] : D. Anderson, T.F. Lunt, H. S. Javitz, A. Tamaru et A.Valdes. *Detecting unusual program behavior using the statistical components of nides*. Technical Report SRI-CSL-95-06, SRI International, Computer Science Laboratory. Menlo Park, Mai 1995.

[24] : H.Debar, M. Becker et D. Siboni. *A neural network component for an intrusion detection system*. In Proceedings of the IEEE Symposium of Research in Computer Security and Privacy. Oakland, Mai 1992.

[25] : P. D'haeseleer, S. Forrest et P. Helman. *An immunological approach to change detection : Algorithms, analysis and implications*. In Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy. Oakland, Mai 1996.

[26] : H.S. Vaccaro et Gunar E. Liepins. *Detection of anomalous computer session activity*. In Proceedings of the IEEE Symposium on Security and Privacy. Oakland, Mai 1989.

[27] : S. Forrest, S.A. Hofmeyr, A. Somayaji et T. A. Longstaff. *A sense of self for unix processes*. In Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy, IEEE Computer Society, IEEE Computer Society Press, May 1996.

[28] : P. Myers. *The neglected aspect of computer security*. Master's thesis, Naval Postgraduate School, Juin 1980.

[29] : K.M.C. Tan, K.S. Killourhy et R.A. Maxion. *Undermining an anomaly-based intrusion detection system using common exploits*. In Proceedings of the 5th International Symposium on Recent Advances in Intrusion Detection (RAID'2002), 2002.

[30] : D. Wagner et P. Soto. *Mimicry attacks on host-based intrusion detection systems*. In Proceedings of the 9th ACM conference on Computer and Communications Security (CCS'02).Washington, USA, Novembre 2002.

[31] : G.C. Tjhai, M. Papadaki, S. Furnell, N. L. Clarke. *Investigating the problem of IDS false alarms : An experimental study using Snort* ,23rd International Information Security Conference SEC 2008, 2008.

[32]: H. Debar, A. Wespi. *Aggregation and Correlation of Intrusion Detection Alerts, Recent Advances in Intrusion Detection*, Springer. London, 2001.

[33]: S. Benferhat, K. Sedki. *Alert Correlation based on a Logical Handling of Administrator Preferences and Knowledge* , International Conference on Security and Cryptography (SECRYPT'08). Porto, Portugal, juillet 2008.

[34] : J.R. Quinlan .C4.5 : *programs for machine learning*, Morgan Kaufmann Publishers Inc..San Francisco, USA, 1993.

[35] : A.Valdes , K.Skinner. *Probabilistic Alert Correlation, Recent Advances in Intrusion Detection*, Springer-Verlag. London, 2001.

[36] : J.R. Quinlan .C4.5 : *programs for machine learning*, Morgan Kaufmann Publishers Inc..San Francisco, USA, 1993.

[37] : A.Valdes , K.Skinner. *Probabilistic Alert Correlation, Recent Advances in Intrusion Detection,* Springer-Verlag. London, 2001.

[38] : F. V.Jensen. *Introduction to Bayesian networks*, UCL Press, London, 1996.

[39] : N. Ben Amor, S. Benferhat, et Z. Elouedi. *Réseaux Bayésiens Naifs et Arbres de Décision dans les Systèmes de Détection d'Intrusions*. Dans quatorzième Congrés Francophone AFRIF AFIA sur la Reconnaissance des Formes et l'Intelligence Artificielle (RFIA'04). Toulouse France, Janvier 2004

[40] : P. Naïm, P.H. Wuillemin, P. Leray, O. Pourret, et A. Becker. *Réseaux bayésiens*.

3 édition, 2007.

[41] : O. François et P. Leray. *Etude Comparative d'Algorithmes d'Apprentissage de Structure dans les Réseaux Bayésiens*. Journal électronique d'intelligence artificielle (JEDAI), 2004 .

[42] : S. Benferhat, T. Kenaza, et A. Mokhtari. *Tree-Augmented Naïve Bayes for Alert Correlation*. Dans 3rd conference on Advances in Computer Security and Forensics (ACSF'08), Juillet 2008.

[43] : M. Sahami, S. Dumais, D. Heckerman et E. Horvitz. *A Bayesian Approach to Filtering Junk E-mail*. Dans AAAI Workshop on Learning for Text Categorization, Juillet 1998.

[44] : C. Kruegel, D. Mutz, W. Robertson, et F. Valeur. *Bayesian event classification for intrusion detection. Dans 19th Annual Computer Security Applications Conference*. LasVegas, December 2003.

[45] : D. M. Chickering. *Learning Bayesian networks is NPcomplete* . Dans Learning from Data : Artificial Intelligence and Statistics V. Springer-Verlag, 1996.

[46] : P. Spirtes, C. Glymour et R Scheines. *Causation, Prediction, and Search, Second Edition (Adaptive Computation and Machine Learning).* The MIT Press, Janvier 2001.

[47] : C. Chow et C. Liu. *Approximating discrete probability distributions with dependence trees*. Information Theory, IEEE Transactions on, 1968.

[48] : G. F. Cooper et E. Herskovits. *A Bayesian Method for the Induction of Probabilistic Networks from Data*. Mach. Learn., 1992.

[49] : A. M. Carvalho et A. L. Oliveira. *Learning Bayesian Networks Consistent with the Optimal Branching*. Dans Proceedings of the Sixth International Conference on Machine Learning and Applications (ICMLA). Los Alamitos, USA, 2007.

[50] : R.D. Shachter et M.A. Peot. *Simulation Approaches to General Probabilistic Inference on Belief Networks* . Dans Proceedings of the Conference on Uncertainty in Artificial Intelligence. Amsterdam, The Netherlands, 1990.

[51] : N. Friedman, D. Geiger, et M. Goldszmidt. Bayesian Network Classifiers. *Machine Learning*, 1997.

[52] : E. J. Keogh et M.J. Pazzani. *Learning the Structure of Augmented Bayesian Classifiers*. International Journal on Artificial Intelligence Tools, 2002.

[53] : J. Cheng et R. Greiner. *Learning Bayesian Belief Network Classifiers : Algorithms and System*. Dans Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence : Advances in Artificial Intelligence, 2001.

[54] : W. Lee. *A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems*. PhD thesis, Computer, Juin 1999.

[55] : KDD. http :// kdd.ccs.uci.edu/databases/kddcup99, 1999.

[56] : J.R Quinlan. C4.5 : *Programs for Machine Learning. Morgan Kaufmann*, 1993.

[57] : S. Benferhat et K. Tabia. *On the combination of naïve Bayes and decision trees for intrusion detection* . Dans CIMCA'05 : Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation. Washington, USA, 2005.

[58] : Y. Bouzida. *Application de l'analyse en composante principale pour la détection d'intrusion et détection de nouvelles attaques par apprentissage supervisé*. PhD thesis , 2006.

[59] : Salem Benferhat et Karim Tabia. *Classification features for detecting server-side and client-side web attacks* . Dans 23rd International Security Conference(SEC'08). Springer LNCS, 2008.

[60] : W. Chimphlee, A. H. Abdullah, M. N. Md. Sap, S. Srinoy et S. Chimphlee. *Anomaly-Based Intrusion Detection using Fuzzy Rough Clustering*. Dans ICHIT '06 : Proceedings of the 2006 International Conference on Hybrid Information Technology.Washington, USA, 2006.

[61] : C.W. Geib , R. P .Goldman, *Plan Recognition in Intrusion Detection Systems*, DISCEX, vol. 1, 2001.

[62] **:** S. Benferhat, T. Kenaza, A.Mokhtari , *False alert filtering and detection of high severe alerts using Naive Bayes*, Computer Security Conference(CSC'08). South Carolina, avril 2008

[63] : A. Faour, P. Leray, *A SOM and bayesian network architecture for alert filtering in network intrusion detection systems*, RTS - Conference on Real-Time and Embedded Systems, 2006.

Webographie

[2]: http://www.larousse.fr/dictionnaires/francais/s%C3%A9curit%C3%A9/71792, consulté le 20/02/2014.

[9]: http://www-igm.univ-mlv.fr/~dr/XPOSE2007/plebacco_ids/A_attaque.html, consulté le 04/03/2014.

Annexe

1- Installation de Snort

1.1- Prérequis

Avant de passer à l'installation de Snort on doit ajouter quelque packages au système d'exploitation, sur un terminal entrer les commandes suivantes :

```
sudo apt-get install nmap
sudo apt-get install nbtscan
sudo apt-get install apache2
sudo apt-get install php5
sudo apt-get install php5-mysql
sudo apt-get install php5-gd
sudo apt-get install libpcap0.8-dev
sudo apt-get install libpcre3-dev
sudo apt-get install g++
sudo apt-get install bison
sudo apt-get install flex
sudo apt-get install libpcap-ruby
sudo apt-get install make
sudo apt-get install autoconf
sudo apt-get install libtool
sudo apt-get install mysql-server (*)
sudo apt-get install libmysqlclient-dev
```

(*) : Vous serez invité à choisir un mot de passe sécurisé (doit être une combinaison de trois (03) symboles au minimum choisis entre : lettre majuscule, lettre minuscule, chiffre, caractère spéciale) pour l'utilisateur root lorsque vous installez ce package. Ne pas oublier ce mot de passe.

Pour s'assurer que le système d'exploitation a les derniers correctifs de sécurité installés, exécuter les commandes suivantes:

```
sudo   apt-get      update
sudo   apt-get      upgrade
```

Vous pourriez avoir à redémarrer si le noyau Linux ou les bibliothèques de base sont mises à jour.

1.2- Installer JpGraph

Cette étape n'est pas nécessaire - c'est seulement à fournir la bibliothèque graphique pour le graphique sur la page principale du Snort Report.

Télécharger une version depuis le site http://jpgraph.net/download/, notre version est jpgraph-1.27.1. Sur un terminal tapez :

```
sudo   mkdir  /var/www/jpgraph
sudo   tar zxvf      jpgraph-1.27.1.tar.gz
sudo   cp -r jpgraph-1.27.1/src      /var/www/jpgraph/
```

1.3- Installer Snort Report

Télécharger Snort Report à partir de ce lien www.symmetrixtech.com/ids/snortreport-1.3.4.tar.gz. Ouvrez une invite de commande dans le répertoire où vous avez téléchargé Snort Report et exécutez les commandes suivantes :

```
sudo   tar  zxvf      snortreport-1.3.4.tar.gz  -C  /var/www/
sudo   gedit   /var/www/snortreport-1.3.4/srconf.php
```

Changer la valeur du mot de passe sur la ligne suivante $pass = "YOURPASS"; par le mot de passe que vous avez choisi lors de l'installation de MySQL.

1.4- Installer l'API d'acquisition de données

Télécharger DAQ à partir du lien suivant http://fr.sourceforge.jp/projects/sfnet_snort/releases/ et depuis un terminal tapez :

```
sudo   tar zxvf  daq-2.0.1.tar.gz
cd     daq-2.0.1
sudo   ./configure
sudo   make
sudo   make install
```

1.5- Installer libdnet

Télécharger la version appropriée depuis https://code.google.com/p/libdnet/downloads/list, en suite pour installer libdnet-1.12 à partir d'un terminal, taper :

```
sudo   tar zxvf libdnet-1.12.tgz
cd     libdnet-1.12/
sudo   ./configure
sudo   make
sudo   make install
sudo   ln -s /usr/local/lib/libdnet.1.0.1 /usr/lib/libdnet.1
```

1.6- Installer Snort-2.9.3

Télécharger la dernière version de snort (pour notre cas 2.9.3) à partir du lien suivant http://www.snort.org/snort-downloads. Installer Snort depuis un terminal avec :

```
sudo   tar zxvf        snort-2.9.3.tar.gz
cd     snort-2.9.3
sudo   ./configure --prefix=/usr/local/snort --enable-sourcefire sudo make
sudo   make install
sudo   mkdir /var/log/snort
sudo   mkdir /var/snort
sudo   groupadd snort
sudo   useradd -g snort snort
sudo   chown snort:snort /var/log/snort
```

La prochaine étape est de télécharger la dernière base de règles Snort à partir du lien suivant http://www.snort.org/snort-rules/ (Vous aurez besoin de vous connecter sur le site de Sourcefire pour obtenir le fichier). Il y a deux sections dans cette page une pour « les membres de la VRT » et l'autre pour « les utilisateurs enregistrés », la seule différence est que les fichiers de règles pour les utilisateurs enregistrés ont 30 jours de retard que ceux pour les abonnés.

Après le téléchargement du fichier de règles, et à partir d'un terminal taper :

```
sudo tar zxvf snortrules-snapshot-2930.tar.gz -C /usr/local/snort
sudo touch /usr/local/snort/rules/white_list.rules
sudo touch /usr/local/snort/rules/black_list.rules
sudo ldconfig
```

1.7- Installer Barnyard2

A l'aide d'un terminal, on va télécharger et installer Firnsy Barnyard2, c'est l'outil qui va nous permettre de visualiser les fichiers de sortie unifiés de Snort.

wget https://github.com/firnsy/barnyard2/archive/v2-1.13.tar.gz

sudo mv v2-1.13.tar.gz barnyard2-2-1.13.tar.gz

sudo tar zxvf barnyard2-2-1.13.tar.gz

cd barnyard2-2-1.13

sudo autoreconf -fvi -I ./m4

sudo ./configure --with-mysql --with-mysql-libraries=/usr/lib/x86_64-linux-gnu

sudo make

sudo make install

sudo cp etc/barnyard2.conf /usr/local/snort/etc

sudo mkdir /var/log/barnyard2

sudo chmod 666 /var/log/barnyard2

sudo touch /var/log/snort/barnyard2.waldo

sudo chown snort.snort /var/log/snort/barnyard2.waldo