Détection et prévention d'intrusion : présentation et limites

Nathalie Dagorn

Rapport de recherche

Détection et prévention d'intrusion : présentation et limites

Nathalie Dagorn[1]

[1] Université de Nancy1
Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA)
Campus Scientifique BP 239, F-54506 Vandœuvre-lès-Nancy Cedex, France
nathalie.dagorn@loria.fr
http://www.loria.fr/

Résumé. Ce rapport de recherche a pour objectif de fournir (i) une présentation générale des techniques et familles de détection et de prévention d'intrusion, ainsi que (ii) une description pertinente des limites technologiques de chacune des solutions présentées. Les résultats de cette recherche (limites et fonctionnalités des outils évoqués) sont issus à la fois d'une analyse scrupuleuse de la littérature spécifique récente et de retours d'expérience d'administrateurs système.

Mots-clés. Détection d'intrusion, détection d'anomalie, détection d'abus, prévention d'intrusion, limites.

Introduction

Aucun système d'information n'est sûr à 100% ! Parmi les préceptes connus sur la sécurité informatique se trouve celui énonçant que, pour une entreprise connectée à l'Internet, le problème aujourd'hui n'est plus de savoir *si* elle va se faire attaquer, mais *quand* cela va arriver ; une solution possible est alors d'essayer de repousser les risques dans le temps par la mise en œuvre de divers moyens destinés à augmenter le niveau de sécurité. Pour contrer les menaces d'intrusion, les entreprises se tournent de plus en plus vers les solutions de détection d'intrusion, dont les possibilités faramineuses sont vantées par les sociétés éditrices de ces logiciels. Mais le décalage entre le discours commercial et les possibilités techniques réelles de ces produits peut être important, et les conséquences fâcheuses lorsqu'il s'agit de sécurité de l'information !

D'acceptation courante, l'intrusion est définie comme l'action de s'introduire sans autorisation dans un lieu dont l'intrus n'est pas propriétaire[1]. Cette définition s'applique également en informatique lorsqu'un tiers pénètre dans un système et accède illégitimement à ses données. Une intrusion peut résulter notamment de l'action d'un pirate désireux de nuire aux biens d'une organisation, d'un ver cherchant à assurer sa propagation, d'une attaque automatisée, etc. Les attaques d'intrusion peuvent être répertoriées selon trois types :

- les attaques réseau ;
- les attaques par portes dérobées (*backdoors*) et canaux de communication cachés (*cover channels*) ;
- les attaques sur les services, qui gagnent une importance croissante (de nos jours, à peu près 60% des intrusions se font via le Web [15]).

La détection d'intrusion est étudiée depuis plus de vingt-cinq ans[2]. Les systèmes dits "passifs" de détection d'intrusion (IDS pour *Intrusion Detection Systems*) ont été déployés de plus en plus largement, suivis aujourd'hui par des systèmes dits "actifs" de prévention d'intrusion (IPS pour *Intrusion Prevention Systems*). La recherche en détection (et prévention) d'intrusion est toujours très active, notamment en raison des évolutions rapides et incessantes dans les technologies de l'information et de la communication.

Ce rapport de recherche présente une vue d'ensemble des techniques et familles de détection et de prévention d'intrusion, ainsi que leurs principales limites. Il n'a pas la prétention d'être exhaustif ni de répondre à toutes les questions concernant la détection/prévention d'intrusion, mais simplement de donner un aperçu du domaine, de façon à dégager les principales limites technologiques des solutions actuelles. Les résultats de cette recherche (limites et fonctionnalités des outils évoqués) sont issus à la fois d'une analyse scrupuleuse de la littérature spécifique récente et de retours d'expérience d'administrateurs système.

1 La détection d'intrusion

En sécurité informatique, la détection d'intrusion est l'acte de détecter les actions qui essaient de compromettre la confidentialité, l'intégrité ou la disponibilité d'une ressource. La détection d'intrusion peut être effectuée manuellement ou automatiquement. Dans le processus de détection d'intrusion manuelle, un analyste humain procède à l'examen de fichiers de logs à la recherche de tout signe suspect pouvant indiquer une intrusion. Un

[1] INTRUS, -USE adj. et n. XIVe siècle. Emprunté du latin ecclésiastique *intrusus*, "qui s'est introduit irrégulièrement ; illégitime". Personne qui s'introduit quelque part sans avoir qualité pour y être admise ou sans y être invitée. (Dictionnaire de l'Académie Française, 9e édition)
INTRUSION n. f. XIVe siècle. Emprunté du latin médiéval *intrusio*, "installation non canonique ; occupation par la force". Le fait de s'introduire, contre le droit ou la forme, dans un lieu, dans une société ou une compagnie, dans une charge. (Dictionnaire de l'Académie Française, 9e édition)

[2] Voir notamment le rapport d'Anderson [1], le modèle de Denning [4], etc.

système qui effectue une détection d'intrusion automatisée est appelé système de détection d'intrusion (IDS). Lorsqu'une intrusion est découverte par un IDS, les actions typiques qu'il peut entreprendre sont par exemple d'enregistrer l'information pertinente dans un fichier ou une base de données, de générer une alerte par e-mail ou un message sur un pager ou un téléphone mobile. Déterminer quelle est réellement l'intrusion détectée et entreprendre certaines actions pour y mettre fin ou l'empêcher de se reproduire, ne font généralement pas partie du domaine de la détection d'intrusion. Cependant, quelques formes de réaction automatique peuvent être implémentées par l'interaction de l'IDS et de systèmes de contrôle d'accès tels que les pare-feu.

1.1 Les systèmes de détection d'intrusion (IDS)

Un système de détection d'intrusion est un mécanisme destiné à repérer des activités anormales ou suspectes sur la cible analysée (réseau, hôte).

Les techniques de détection d'intrusion. Deux techniques de détection d'intrusion sont généralement mises en œuvre par les IDS courants.

La détection d'abus (misuse detection). Dans la détection d'abus (aussi appelée détection de mauvaise utilisation), l'IDS analyse l'information recueillie et la compare (*pattern matching*, approche par scénarii) avec une base de données de signatures (motifs définis, caractéristiques explicites) d'attaques connues (i.e., qui ont déjà été documentées), et toute activité correspondante est considérée comme une attaque (avec différents niveaux de sévérité). Un système de détection d'abus est, par exemple, STAT (*State Transition Analysis Toolkit*) [6].

La détection d'anomalie (anomaly detection). La détection d'anomalie de comportement est une technique assez ancienne (elle est utilisée également pour détecter des comportements suspects en téléphonie, comme le *phreaking*). L'idée principale est de modéliser durant une période d'apprentissage le comportement "normal" d'un système/programme/utilisateur en définissant une ligne de conduite (dite *baseline* ou profil[3]), et de considérer ensuite (en phase de détection) comme suspect tout comportement inhabituel (les déviations significatives par rapport au modèle de comportement "normal"). Les modèles de détection d'anomalie incluent fréquemment des modèles statistiques, tels que ceux utilisés dans NIDES (NIDES/STAT) [7] ou Haystack

[3] Par exemple, profil de connexion : fréquence de login (combien de fois par jour/semaine l'utilisateur se connecte-t-il ?), lieu de login (statistiques sur la connexion distante/locale), etc. Un profil est donné par une métrique et un modèle statistique. Une métrique est une variable aléatoire X modélisant le comportement quantitatif sur une période de temps. Un modèle est utilisé pour détecter si les nouvelles valeurs de X concordent avec les valeurs de X déjà observées (et supposées légitimes).

[14], par exemple. La détection d'anomalie peut invoquer les fonctionnalités suivantes [11] :

- Détection de seuil (*threshold detection*) : des compteurs délimitent une zone ou sont utilisés pour compter combien de fois quelque-chose est exécuté, ouvert, démarré... Cette analyse statique peut être améliorée par la détection "heuristique" de seuil[4].
- Détection basée sur des règles définies (*rule-based detection*) : si l'usage dévie de ces règles, une alarme est déclenchée.
- Mesure statique (*static measure*) : le comportement de l'utilisateur et du système se conforme à une signature prédéfinie. Un programme notant l'activité "normale" de l'utilisateur pour la définition de signatures est souvent inclus.
- Détection d'anomalie de protocole (*protocol anomaly detection*) : elle représente un sous-groupe de la détection d'anomalie. C'est une technique relativement récente fonctionnant, à la base, comme la détection d'anomalie. Chaque protocole a une signature prédéfinie (voir les RFC correspondantes) ; le but de la détection d'anomalie de protocole est de trouver si le comportement du protocole est conforme à celui prédéfini ou non. Beaucoup d'attaques sont basées sur l'abus de protocole ; ce sous-groupe est donc assez important relativement aux IDS.

Dans la pratique, la détection d'abus et la détection d'anomalie sont souvent utilisées de façon complémentaire par les IDS (par exemple, EMERALD [12], JiNao [2, 16])[5].

Le diagramme en Fig. 1 illustre le fonctionnement d'un IDS.

[4] Le compteur est dynamique, et non pas initialement fixé à une valeur statique.

[5] Des variantes plus ou moins proches de ces techniques existent, par exemple, la détection basée sur une politique (*policy-based detection*). Dans ce cas, l'IDS effectue la détection d'intrusion en se basant sur une politique de sécurité prédéfinie (consiste en une spécification de politique de sécurité logique et un algorithme de validation de trace d'exécution) [17]. Une telle approche a le potentiel d'apporter des améliorations sensibles par rapport à la détection statistique d'anomalie et à la détection de scénario en termes de fiabilité, d'exactitude et de maintenance requise. Idéalement, la maintenance est nécessaire uniquement pour changer la politique spécifiée, sans besoin de mise à jour ou de phase d'adaptation. Tout comme la détection d'abus, cette technique se révèle efficace pour détecter les attaques connues, le taux de fausses alarmes est faible, et le processus est normalement moins consommateur de ressources informatiques. Malheureusement cette technique est inefficace pour détecter des attaques nouvelles ou inhabituelles. Ecrire les règles de la politique de détection peut se révéler très pénible ; en outre, si ces règles venaient à être connues de l'attaquant, elles pourraient être contournées. La détection basée sur une politique n'est pas détaillée dans ce rapport introductif.

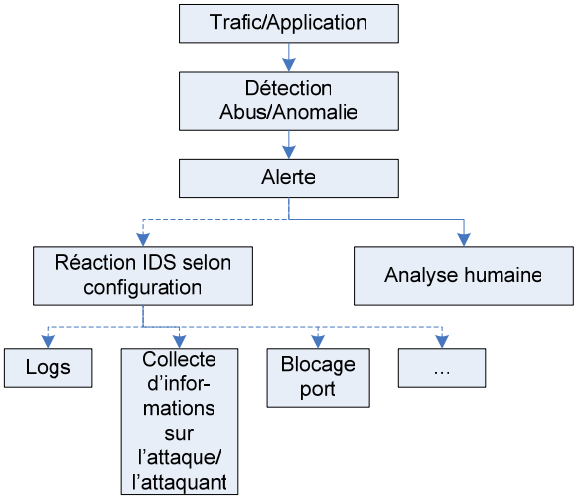

Fig. 1. Fonctionnement d'un IDS

Les familles d'IDS et leurs variantes (localisation). Selon l'endroit qu'ils surveillent et ce qu'ils contrôlent (les "sources d'information"), deux familles principales d'IDS sont usuellement distinguées [9, 10] :

IDS réseau (NIDS). Le rôle essentiel d'un IDS réseau, appelé NIDS (*Network-based Intrusion Detection System*), est l'analyse et l'interprétation des paquets circulant sur ce réseau. Afin de repérer les paquets à contenu malicieux "caractéristique", comme par exemple `/etc/passwd`, des signatures sont créées. Des détecteurs (souvent de simples hôtes) sont utilisés pour analyser le trafic et si nécessaire envoyer une alerte[6]. Un IDS réseau travaille sur les trames réseau à tous les niveaux (couches réseau, transport, application). De plus en plus, en disséquant les paquets et en "comprenant" les protocoles, il est capable de détecter des paquets malveillants conçus pour outrepasser un pare-feu aux règles de filtrage trop laxistes, et de chercher des signes d'attaque à différents endroits sur le réseau. Quelques exemples de NIDS : NetRanger, NFR, Snort, DTK, ISS RealSecure[7].

Les IDS réseau ont des atouts, par exemple, les détecteurs peuvent être bien sécurisés puisqu'ils se "contentent" d'observer le trafic, les *scans* sont détectés plus facilement grâce aux signatures, etc. Cependant, les problèmes majeurs liés aux NIDS sont de conserver toujours une bande passante suffisante pour l'écoute de l'ensemble des paquets, et de bien positionner l'IDS pour qu'il soit efficace.

---

[6] Dans ce contexte, le mode furtif (*stealth mode*) est souvent choisi, c'est-à-dire que les capteurs agissent de manière invisible, les rendant plus difficiles à localiser et à atteindre pour un attaquant.

[7] Resp. http://www.cisco.com; http://www.nfr.net; http://www.snort.org; http://all.net/dtk/dtk.html; http://www.iss.net/products_services/enterprise_protection/rsnetwork/sensor.php

IDS hôte (HIDS). Les systèmes de détection d'intrusion basés sur l'hôte (poste de travail, serveur, etc.), ou HIDS (*Host-based IDS*), analysent exclusivement l'information concernant cet hôte. Comme ils n'ont pas à contrôler le trafic du réseau mais "seulement" les activités d'un hôte, ils se montrent habituellement plus précis sur les variétés d'attaques. Ces IDS utilisent deux types de sources pour fournir une information sur l'activité : les *logs* et les traces d'audit du système d'exploitation. Chacun a ses avantages : les traces d'audit sont plus précises, détaillées et fournissent une meilleure information ; les *logs*, qui ne fournissent que l'information essentielle, sont plus petits et peuvent être mieux analysés en raison de leur taille. Il n'existe pas de solution unique HIDS couvrant l'ensemble des besoins, mais les solutions existantes couvrent chacune un champ d'activité spécifique, comme l'analyse de *logs* système et applicatifs, la vérification de l'intégrité des systèmes de fichiers, l'analyse du trafic réseau en direction/provenance de l'hôte, le contrôle d'accès aux appels système, l'activité sur les ports réseau, etc. Par exemple, le démon syslog peut être considéré partiellement comme un système HIDS, car il permet de consigner certaines activités, et à l'aide d'un analyseur comme Swatch[8], de détecter certaines tentatives d'intrusion (comme *bad login*) ; Tripwire[9], centrant son activité sur l'intégrité du système de fichiers, peut aussi être vu comme un HIDS ; Security Manager[10], etc.

Les systèmes de détection d'intrusion basés sur l'hôte ont certains avantages : l'impact d'une attaque peut être constaté et permet une meilleure réaction, des attaques dans un trafic chiffré peuvent être détectées (impossible avec un IDS réseau), les activités sur l'hôte peuvent être observées avec précision, etc. Ils présentent néanmoins des inconvénients, parmi lesquels : les *scans* sont détectés avec moins de facilité ; ils sont plus vulnérables aux attaques de type DoS ; l'analyse des traces d'audit du système est très contraignante en raison de la taille de ces dernières ; ils consomment beaucoup de ressources CPU, etc.

Une version d'*IDS hybride* est possible et désormais supportée par différentes offres commerciales. Même si la distinction entre HIDS et NIDS est encore courante, certains HIDS possèdent maintenant les fonctionnalités de base des NIDS. Des IDS bien connus comme ISS RealSecure se nomment aujourd'hui "IDS hôte et réseau". Dans un futur proche, la différence entre les deux familles devrait s'estomper de plus en plus (ces systèmes devraient évoluer ensemble). De ces deux familles principales, de nombreuses variantes sont issues :

IDS de nœud réseau (NNIDS). Les systèmes de détection d'intrusion de nœud réseau (NNIDS pour *Network Node IDS*) fonctionnent comme les NIDS classiques, c'est-à-dire qu'ils analysent les paquets du trafic réseau, hormis que (i) cela ne concerne que les

[8] http://swatch.sourceforge.net/
[9] http://www.tripwire.com/
[10] http://www.netiq.com/products/sm/default.asp

paquets destinés à un nœud du réseau, et (ii) à la différence des NIDS classiques, les NNIDS ne fonctionnent pas en mode "*promiscuous*"[11], mais d'un autre côté, puisque tous les paquets ne sont pas vérifiés, les performances de l'analyse sont améliorées.

IDS basé sur une application (ABIDS). Les IDS basés sur les applications (ABIDS pour *Application-Based IDS*) sont un sous-groupe des IDS hôtes, parfois mentionnés séparément. Ils contrôlent l'interaction entre un utilisateur et un programme en ajoutant des fichiers de *logs* afin de fournir de plus amples informations sur les activités. Opérant entre utilisateur et programme, il est facile pour l'ABIDS de filtrer tout comportement notable.

Ses principaux avantages sont de travailler en clair (contrairement aux NIDS, par exemple) d'où une analyse plus facile, et la possibilité de détecter et d'empêcher des commandes particulières dont l'utilisateur pourrait se servir avec le programme. Deux inconvénients majeurs sont identifiés : le peu de chances de détecter, par exemple, un cheval de Troie (puisque l'ABIDS n'agit pas dans l'espace du noyau) ; en outre, les fichiers de *logs* générés par ce type d'IDS sont des cibles faciles pour les attaquants (ils ne sont pas aussi sûrs que les traces d'audit du système, par exemple).

IDS basé sur la pile (SBIDS). Les systèmes de détection d'intrusion basés sur une pile (SBIDS pour *Stack-Based IDS*), travaillant étroitement avec la pile TCP/IP, octroient la consultation des paquets lorsqu'ils montent à travers les couches OSI et permettent ainsi à l'IDS de retirer les paquets de la pile avant que le système d'exploitation ou l'application n'ait eu la possibilité d'élaborer la charge virale. L'IDS basé sur une pile peut être efficace contre certaines formes de chiffrement en retraçant les paquets après qu'ils aient été déchiffrés par la pile TCP/IP.

Pot de miel (honeypot). Possédant de nombreuses similitudes avec les IDS, les "pots de miel" (traduction littérale de l'expression anglaise *honeypot*) sont quelquefois considérés comme faisant partie des IDS. Un pot de miel est un outil informatique (système, serveur, programme, etc.), volontairement exposé et vulnérable à une ou plusieurs failles connues, destiné à attirer et à piéger les pirates, tout en permettant d'observer leur comportement en pleine action et d'enregistrer leurs méthodes d'attaque pour mieux les étudier, les comprendre et les anticiper. La tâche principale d'un pot de miel consiste à analyser le trafic, c'est-à-dire à informer du démarrage de certains processus, de la modification de fichiers... permettant ainsi de créer un profil élaboré des attaquants potentiels. Tout l'art du pot de miel consiste à remonter jusqu'à l'origine de l'attaque, sans que le pirate s'en doute et que jamais il ne puisse soupçonner le fait que le site visité ne soit qu'un leurre (montage). Le réseau interne n'est pas exposé puisque le pot de miel est généralement placé dans la DMZ (zone démilitarisée). Il existe trois variantes de pots de miel :

[11] Le mode "*promiscuous*" permet de capturer l'ensemble des trames circulant sur le réseau.

- De prévention : les pots de miel ne sont pas les systèmes les plus appropriés pour éviter les attaques. Comme nous le verrons avec les systèmes “capitonnés”, un pot de miel mal programmé et mal configuré peut même faciliter les attaques.
- De détection: l'un des principaux avantages des pots de miel vient de ce qu'ils excellent à la détection d'attaques. Dans ce contexte, ils peuvent surtout analyser et interpréter les *logs*. Le placement du pot de miel joue un rôle décisif : les administrateurs ne peuvent tirer bénéfice des pots de miel que s'ils sont placés et configurés correctement. Souvent, ils sont placés entre des serveurs importants pour détecter des *scans* éventuels du réseau entier, ou à proximité d'un serveur essentiel pour détecter les accès illégaux à l'aide d'une redirection de port (si quelqu'un essaie d'accéder au serveur par certains ports, il est redirigé vers le pot de miel, et comme cet accès ne doit pas être autorisé, une alerte est générée).
- De réaction: selon sa configuration, un pot de miel peut réagir aux événements.

De même, les pots de miel peuvent être classés en deux sous-catégories : recherche et production. Les pots de miel de production ont pour objectif de diminuer les risques sur un réseau alors que ceux de recherche servent à obtenir des informations sur les attaquants.

Attention, les pots de miel ne sont pas totalement légaux et il convient d’être prudent dans leur utilisation. Un pot de miel peut être considéré comme une “invitation à l'attaque” et si l'hôte attaqué n’a pas pris de mesures réactives, cette “non-réaction” peut être interprétée comme une négligence. Quelques argumentations judiciaires contre les pots de miel semblent banales mais il est préférable de vérifier ce que dit la loi en vigueur dans le pays d’installation (en Europe, les pots de miel sont autorisés). Si quelqu'un attaque un autre réseau à partir du réseau du pot de miel et cause des dommages, l’entité qui a installé le pot de miel sera considérée comme responsable.

Systèmes “capitonnés” (padded cell systems). Les systèmes capitonnés fonctionnent habituellement avec l'un ou l'autre des systèmes abordés précédemment. Si l'IDS utilisé informe d'une attaque, l'attaquant est dirigé vers un hôte “capitonné”. Une fois dans cet hôte, il ne peut plus causer de dommages puisque l'environnement est simulé (c'est un leurre). Cet environnement doit être aussi réaliste que possible, autrement dit, l'attaquant doit penser que son attaque a été couronnée de succès. Il est alors possible d'enregistrer, de suivre et d'analyser toute activité de l'attaquant.

Les systèmes “capitonnés” présentent des inconvénients : d’une part leur usage peut être illégal (comme pour les pots de miel, ils sont tolérés en Europe) ; d’autre part, la mise en œuvre d'un tel système est assez difficile et réclame des compétences puisque l'ensemble de l'environnement doit être simulé correctement (si l'administrateur fait une petite erreur, ce système peut être à l'origine de nouveaux trous de sécurité).

1.2 Les limites des IDS

Historiquement, les entreprises et les gouvernements ont été réticents à révéler des informations concernant les attaques sur leurs systèmes de peur de perdre la confiance publique, ou de peur que d'autres attaquants exploitent la même vulnérabilité ou une vulnérabilité similaire. Aussi les données spécifiques et détaillées sur les attaques n'ont-elles pas été disponibles jusqu'à une époque très récente. Aujourd'hui encore les entreprises restent réticentes à dévoiler ce type de données, ce qui ne facilite pas le travail autour des IDS. La détection d'intrusion présente des limites et peut se montrer impuissante dans certains cas.

Limites générales des IDS. Ces limites s'appliquent aux techniques de détection d'abus comme à celles de détection d'anomalie.

Attaques sur les drapeaux TCP. Les IDS sont vulnérables à certaines attaques sur les drapeaux TCP *(TCP flags)* [13], comme par exemple :

- envoi d'un faux SYN ;
- insertion de données avec mauvais numéro de séquence ;
- FIN/RST *spoofing* avec mauvais numéro de séquence ;
- désynchronisation après connexion ;
- désynchronisation avant connexion [SYN (mauvaise somme de contrôle + mauvais numéro de séquence) puis SYN] ;
- FIN/RST *spoofing* avec mauvaise somme de contrôle ;
- *Data spoofing* avec mauvaise somme de contrôle ;
- FIN/RST *spoofing* avec TTL court ;
- insertion de données avec un TTL court, etc.

Placement de l'IDS. Sans entrer dans les détails, au niveau du placement de l'IDS (implémentation et design), il est intéressant de faire de la détection d'intrusion dans la zone démilitarisée (attaques contre les systèmes publics), dans le (ou les) réseau(x) privé(s) (intrusions vers ou depuis l'intérieur) et derrière le pare-feu (détection des signes parmi tout le trafic entrant et sortant). Chacun de ces positionnements a ses avantages et inconvénients. L'important est de bien identifier les ressources à protéger (risques d'affaires majeurs) et ce qui est le plus susceptible d'être attaqué [9]. Il convient alors d'implémenter précautionneusement l'IDS (paramétrage, etc.) en fonction du placement choisi [13].

Pollution/surcharge. Les IDS peuvent être pollués ou surchargés, par exemple par la génération d'un trafic important (le plus difficile et lourd à analyser possible). Une quantité importante d'attaques bénignes peut également être envoyée afin de surcharger les alertes de l'IDS [3]. Des conséquences possibles de cette surcharge peuvent être la saturation de ressources (disque, CPU, mémoire), la perte de paquets, le déni de service partiel ou total [5].

- Consommation de ressources : outre la taille des fichiers de *logs* (de l'ordre du Go), la détection d'intrusion est excessivement gourmande en ressources ;
- Perte de paquets (limitation des performances) : les vitesses de transmission sont parfois telles qu'elles dépassent largement la vitesse d'écriture des disques durs les plus rapides du marché, ou même la vitesse de traitement des processeurs. Il n'est donc pas rare que des paquets ne soient pas reçus par l'IDS, et que certains d'entre eux soient néanmoins reçus par la machine destinataire.
- Vulnérabilité aux dénis de service : un attaquant peut essayer de provoquer un déni de service au niveau du système de détection d'intrusion, ou pire au niveau du système d'exploitation de la machine supportant l'IDS. Une fois l'IDS désactivé, l'attaquant peut tenter tout ce qui lui plait. Par exemple, l'attaque Stick est une tentative de déni de service contre les IDS (en particulier contre ISS RealSecure) surchargeant de travail l'IDS au point de le désactiver ou au moins de le rendre moins efficace.

Ainsi une attaque réelle furtive glissée dans ce trafic aura du mal à être identifiée si le flot d'informations généré est suffisant, et risque donc de ne pas être traitée ! Tout outil permettant de générer du trafic peut être utilisé à cette fin : SynFlooder, Smurf, Targa, Tcpreplay, Nmap[12], etc. [13].

Contournement/évasion. Les IDS peuvent également être contournés ou outrepassés. Dans le cas d'une attaque par évasion, le système de détection d'intrusion rejette un paquet qui sera pourtant accepté par la destination [5]. Il se peut, par exemple, qu'une différence de systèmes d'exploitation entre la machine supportant l'IDS et la machine surveillée fasse que certains paquets rejetés par le système de détection d'intrusion soient acceptés par la destination (comme des paquets UDP avec une somme de contrôle erronée, rejetés par la plupart des systèmes d'exploitation sauf les plus anciens) [5].

[3] distingue six techniques de contournement :

- Insertion (ajout de données aux flux présents) : ce peut être, par exemple, l'insertion de paquets avec des TTL courts.
- Elimination : consiste à rendre l'IDS inutile ou inexploitable.
- Substitution : ce procédé échange tout ou une partie du contenu incriminable.
- Fragmentation (découper un contenu ou des opérations) : elle aboutit en général à une modification de signature, par exemple l'envoi de deux fragments qui se recouvrent. C'est le cas par exemple des attaques de type Teardrop[13], du chevauchement des en-têtes (pour modifier l'IP source, le port de destination… au dernier moment), ou encore des attaques au niveau applicatif qui peuvent être fragmentées pour leurrer le moteur d'analyse par *pattern matching* de l'IDS.

---

[12] http://tcpreplay.sourceforge.net; http://www.insecure.org/nmap/

[13] Attaque par fragmentation : le principe de l'attaque *Teardrop* consiste à insérer dans des paquets fragmentés des informations de décalage erronées. Ainsi, lors du réassemblage il existe des vides ou des recoupements (*overlapping*), pouvant provoquer une instabilité du système.

- Distribution (répartition des sources) : la distribution peut permettre une attaque coordonnée (concept présenté et géré, par exemple, dans Shadow Novell NetWare Crack[14] avec l'outil Dscan -*distributed scanner*-).
- Confusion : consiste à rendre le contenu incompréhensible.

[5] ajoute à ces techniques de contournement les attaques temporelles : la détection comprenant fréquemment une valeur de seuil, l'attaquant peut s'arranger pour délayer ses attaques dans le temps en prenant garde à ne pas générer une activité dépassant les seuils fixés.

D'autres attaques peuvent être mises en œuvre pour outrepasser les IDS, notamment les attaques des protocoles à bas niveau (IP, TCP), Whisker[15] (*scanner* de vulnérabilités du protocole de haut niveau http), URL encodées, etc.

En outre, la source d'une attaque n'est pas toujours identifiable [15] :

- Par exemple, le "scan zombie" permet de s'appuyer sur une autre machine afin de leurrer l'IDS sur la provenance de l'attaque. En effet, Nmap dispose de l'option -sI qui permet de forger des paquets adéquats et de se faire passer pour une autre machine, puis de déterminer le comportement à adopter en fonction de la réaction de la cible. Dans tous les cas, l'assaillant ne peut être directement découvert par l'IDS [3].
- Attaque d'origine masquée : il est possible d'utiliser un relais mal configuré ou anonyme sur Internet (option –p de Nmap), ou d'utiliser pour chaque requête un relais http tiré aléatoirement dans une liste[16], outils hping ou idlescan[17], etc.
- Attaque "noyée" (option decoy de Nmap) en envoyant des paquets similaires mais avec de fausses adresses sources. L'adresse d'origine est une parmi plusieurs ! [13].

Il existe de nombreuses autres techniques anti-IDS non abordées ici par mesure de concision. Pour de plus amples développements sur ces techniques, nous renvoyons le lecteur à des analyses telles que [11] ou [5].

Temps de détection. Le temps de détection est un élément capital pour un IDS : la détection des intrusions se fait-elle en temps réel ou nécessite-t-elle un délai ? Quel délai (quelques jours…) ? [13]. L'expérience montre qu'il faut habituellement un certain laps de temps afin de déceler ou de reconstituer une attaque (temps d'analyse, de réaction...) [9].

Communication cachées (cover channels). Les communications cachées sont typiquement des communications entre un pirate et la machine compromise. Par exemple, il est possible d'encapsuler des données malicieuses dans des commandes TCP. Les attaques

---

[14] Voir par exemple http://www.tenebril.com/src/info.php?id=130322275

[15] Voir http://www.securityfocus.com/tools/727)

[16] Option -B 4 : *distributed proxy scanning* : des listes de plus de 500 relais anonymes sont disponibles.

[17] Respectivement http://www.hping.org/; http://www.insecure.org/nmap/idlescan.html

par canaux de communication cachée (de type Loki, *reverse www shell*, *covert TCP*…) sont très difficiles à détecter. A l'heure actuelle, aucune solution n'existe contre ce danger potentiel, puisque même l'interception en profondeur est inefficace si la cryptographie est utilisée pour camoufler les données en transit. Seule une analyse rigoureuse des changements sur le disque dur d'une machine associée à l'observation éventuelle de trafic inexpliqué peut permettre de découvrir un canal de communication cachée en activité [5].

Mode "promiscuous". L'utilisation du mode "*promiscuous*" présente quelques inconvénients, notamment [3] :

- Réponse involontaire du système : par nature, les IDS doivent mettre leur carte réseau en mode "*promiscuous*" afin de pouvoir recevoir l'intégralité des trames circulant sur le réseau. Ainsi, l'IDS ne générera généralement aucun trafic et se contentera d'aspirer tous les paquets. Cependant, ce mode spécial désactive la couche 2 "liaison" de la machine (le filtrage sur les adresses MAC n'est plus activé). Il se peut alors que la machine réponde à certains messages (ICMP echo request généré avec l'outil Nemesis[18]).
- Mise en évidence de la présence d'un IDS : le mode "*promiscuous*" génère des accès mémoire et processeur importants ; il est possible de détecter de telles sondes en comparant les latences de temps de réponse avec celles des machines du même brin LAN (ou proche). Des temps de réponse trop importants sont significatifs d'une activité gourmande en ressources telle que le *sniffing*, validant possiblement la présence d'un IDS.
- L'utilisation du mode "*promiscuous*" implique d'installer une sonde par réseau commuté.

Réponse de l'IDS. Enfin, des difficultés et désaccords existent sur la réponse à donner par l'IDS aux différentes attaques, et sur le contenu des retours et actions que l'IDS devrait exécuter [13]. Selon ses possibilités intrinsèques et sa configuration, un IDS peut être *passif* ou *réactif* : en tant que système passif, l'IDS détecte une faille potentielle de sécurité, note l'information et génère une alerte. En tant que système réactif, l'IDS répond à l'activité suspecte, par exemple en déconnectant un utilisateur ou en reprogrammant le pare-feu afin qu'il bloque le trafic réseau provenant de la source malveillante suspectée.

Aujourd'hui, tous les problèmes ne sont pas résolus, les outils commencent à apparaître pour répondre à ces limites et contrer les attaques contre les IDS [13] (mais c'est une boucle infinie, cf conclusion).

Limites spécifiques à la détection d'abus. Les principaux défis actuels de cette technique sont les suivants.

[18] http://nemesis.sourceforge.net/

Définition et maintenance des signatures. Toutes les attaques ne sont pas détectées, sselon les fonctionnalités du système, la définition de la signature, la mise à jour de la base, la charge du système, etc. [13] :

- Limites “humaines” : signatures pas à jour ou mal conçues [13]. La détection d’abus a pour impératifs une bonne conception des signatures d’attaques et une mise à jour continue de la liste des signatures [3].
- Contexte d’utilisation : parfois la technologie est basée sur des signatures qui ne reposent pas sur le contexte d’utilisation. La conséquence est double : de nombreux faux positifs[19] et une dégradation importante des performances du système.
- Même si la méthode des signatures de corps (y compris les signatures de chaîne) semble être assez sûre, il y a moyen de les contourner [11] (voir paragraphe contournement/évasion ci-dessus).
- Vulnérabilité aux mutations : de par son manque de flexibilité, la détection par signatures d'attaques est très vulnérable aux mutations. D’une part, pour pouvoir définir une signature, il faut avoir déjà été confronté à l'attaque considérée. D'autre part, certaines de ces signatures se basent sur des caractéristiques “volatiles” d'un outil, comme par exemple le port qu'un cheval de Troie ouvre par défaut ou la valeur d’ISN (*Initial Sequence Number*) choisie par certains outils de piratage. Or ces logiciels sont souvent soit hautement configurables, soit *open source* donc librement modifiables. Les caractéristiques retenues pour définir la signature sont donc fragiles, et les signatures extrêmement sensibles aux mutations [5].
- Faute de définition, les *nouvelles attaques* passent l’IDS sans être détectées.

Faux positifs. Normalement, l'avantage de la détection d'abus devrait être un faible taux de *faux positifs* (fausses alarmes) puisque les critères des signatures peuvent être définis de manière précise [11]. Néanmoins, selon les sources d’information, on lira qu’il y a de peu à beaucoup de faux positifs résultant de cette technique. [9] concilie « *qu’il y en a de moins en moins avec l’évolution de la technique mais qu’il reste encore du travail à faire* », notamment concernant :

- La sensibilité/spécificité de l’IDS : par nature, les IDS vont remonter énormément d'alertes s’ils ne sont pas configurés convenablement. Toute l’attention doit être portée à la création des règles de signature [11]. Le compromis effectué entre la quantité d'alertes remontée (sensibilité ou *accurancy*) et la finesse de ces dernières (spécificité ou *precision*) est déterminant [15]. Il faut donc prendre soin d'inclure dans le fichier de configuration le fichier “.rule” nécessaire, en fonction des règles établies par le pare-feu. Par exemple, si un service est totalement interdit, il est presque inutile d'inclure les signatures associées [3].

[19] Faux positif : alerte pour une attaque qui n'a pas eu lieu (par opposition à faux négatif : une attaque a eu lieu mais n’a pas été détectée).

- La pertinence de l'information (en considérant le taux de faux positifs résultant d'une analyse) [3].
- Le comportement de l'IDS : le système se comporte-t-il bien ? Autrement dit, capture-t-il toutes (ou la plupart) des intrusions ? Capture-t-il des intrusions réelles ? [13]

Limites spécifiques à la détection d'anomalie. Cette technique comporte elle-aussi de nombreux problèmes complexes à résoudre ; voici les plus couramment évoqués.

Apprentissage/configuration de l'IDS. L'apprentissage du comportement « normal » n'est pas aisé. Automatiser le raisonnement conduisant à penser que le comportement est "déviant" par rapport à celui connu est une tâche difficile. Par contre, cette technique est appliquée par défaut (la plupart du temps) par les administrateurs réseau ou système : lorsque quelque-chose paraît inhabituel (par exemple, des pics de bande passante, des services qui tombent, des systèmes de fichiers qui se remplissent plus vite qu'à l'accoutumée, etc.), l'usage veut que des recherches plus poussées soient entreprises [9]. Par ailleurs, toute anomalie ne correspond pas forcément à une attaque, ce peut être un changement de comportement de l'utilisateur [15] ou un changement de la configuration du réseau [3]. En règle générale, la convergence vers un modèle comportemental "normal" est plutôt longue [5].

Lors du paramétrage de l'IDS, toute la difficulté pour une détection efficace réside dans le choix des métriques, des modèles de comportement et dans la définition des différents profils. Pour toutes ces raisons, les IDS fonctionnant par détection d'anomalie sont reconnus comme étant très longs et fastidieux à configurer [13].

Même après une configuration efficace, rien n'empêche un pirate se sachant surveillé de "rééduquer" un tel système en faisant évoluer progressivement son modèle de convergence vers un comportement anormal pour l'analyste, mais tout-à-fait "normal" d'un point de vue statistique [5].

Faux positifs. La détection d'anomalie est capable de détecter les attaques inconnues ; toutefois, elle n'est pas aussi efficace que la détection d'abus pour les attaques connues. Notamment, un fort taux de faux positifs peut être rencontré si le paramétrage de l'IDS n'a pas été réalisé avec soin [9].

Vie privée (privacy). Enfin, les IDS basés sur la détection d'anomalie de comportement enregistrent toutes les actions des utilisateurs ; en ce sens, cette technique pose la question de l'atteinte à la vie privée [15, 13].

Pour tenter de dépasser ces limites (et d'autres encore), la détection d'anomalie fait l'objet de nombreuses recherches à l'heure actuelle.

2 La prévention d'intrusion

La *prévention d'intrusion* est un ensemble de technologies de sécurité ayant pour but d'anticiper et de stopper les attaques [3]. La prévention d'intrusion est appliquée par quelques IDS récents et diffère des techniques de détection d'intrusion décrites précédemment : au lieu d'analyser les *logs* du trafic, c'est-à-dire découvrir les attaques après qu'elles se soient déroulées, la prévention d'intrusion essaie de prévenir ces attaques. Là où les systèmes de détection d'intrusion se contentent de donner l'alerte, les systèmes de prévention d'intrusion bloquent le trafic jugé dangereux.

2.1 Les systèmes de prévention d'intrusion (IPS)

Le principe de fonctionnement d'un IPS est symétrique à celui d'un IDS (IPS hôte et IPS réseau), ajoutant à cela l'analyse des contextes de connexion, l'automatisation d'analyse des *logs* et la coupure des connexions suspectes. Contrairement aux IDS classiques, aucune signature n'est utilisée pour détecter les attaques. Avant toute action, une décision en temps réel est exécutée (i.e., l'activité est comparée à un ensemble de règles). Si l'action est conforme à l'ensemble de règles, la permission de l'exécuter sera accordée et l'action sera exécutée. Si l'action est illégale (c'est-à-dire si le programme demande des données ou veut les changer alors que cette action ne lui est pas permise), une alarme est donnée. Dans la plupart des cas, les autres détecteurs du réseau (ou une console centrale) en seront aussi informés dans le but d'empêcher les autres ordinateurs d'ouvrir ou d'exécuter des fichiers spécifiques. Le diagramme ci-après illustre le fonctionnement d'un IPS.

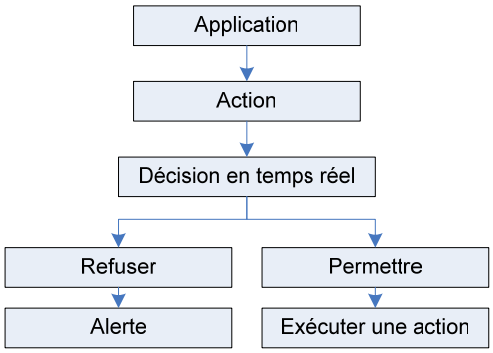

Fig. 2. Fonctionnement d'un IPS [11]

Les IPS fournissent les fonctionnalités suivantes [11] :

- La surveillance du comportement d'application se rapproche des IDS basés sur une application, c'est-à-dire que le comportement de l'application est analysé et

noté (quelles données sont normalement demandées, avec quels programmes elle interagit, quelles ressources sont requises, etc.).

- La création de règles pour l'application : dérivé de la surveillance du comportement d'application, cet ensemble de règles donne des informations sur ce que peut faire ou non une application (par exemple, quelles ressources peut-elle demander ?).
- La fonctionnalité d'alerte suite aux violations permet d'envoyer une alerte en cas de déviation (c'est-à-dire lorsqu'une attaque est détectée). L'alerte peut aller d'une simple entrée dans un journal à un blocage de ressources, par exemple.
- La corrélation avec d'autres événements implique un partage d'informations entre des senseurs coopératifs, afin de garantir une meilleure protection contre les attaques.
- L'interception d'appels au système : avant qu'un appel au système (*rootkit*) soit accepté, il doit être complètement vérifié (par exemple, quel programme a demandé l'appel au système, sous quelles autorisations d'utilisateur tourne le processus *-root...-*, à quoi l'appel système essaie-t-il d'accéder, etc.). Cette fonctionnalité permet la surveillance des essais de modification d'importants fichiers du système ou de la configuration.
- D'autres fonctionnalités sont possibles, comme la compréhension des réseaux IP (architecture, protocoles, etc.), la maîtrise des sondes réseau/analyse des *logs*, la défense des fonctions vitales du réseau, la vitesse d'analyse et un mode "*stateful inspection*".

La prévention d'intrusion est une technique relativement nouvelle par comparaison aux autres techniques. Cette approche fait interagir des technologies hétérogènes : pare-feu, VPN, IDS, anti-virus, anti-spam, etc.

2.2 Les limites des IPS

Néanmoins le décalage entre le concept commercial (cf Cisco : « *Des réseaux capables de se défendre tout seuls* », etc.) et la réalité peut être important ! Un article du Magazine 01net.[20] met en lumière que « *[…] pour l'entreprise, un IPS constitue, à première vue, un investissement pour la sécurité plus judicieux qu'un simple IDS. Contrairement à celui-ci, il pourra fonctionner sans une surveillance constante. Autre avantage, un IPS est capable de stopper des attaques réseau et, surtout, applicatives, les plus courantes aujourd'hui […] Cependant, si un IDS -qui est passif- peut se permettre de se tromper, un IPS -actif- n'a pas le droit à l'erreur.* »

Les principales limites et contraintes des IPS à ce jour semblent être leur mise en place délicate, leur administration rebutante, la possibilité de bloquer tout le réseau en cas de

[20] Article « IPS : pour une sécurité active » de Jérôme Saiz, 22/11/2004 (258845), Décision Informatique 01net., www.01net.com.

fausse alerte, ainsi que l'inexistence d'un standard actuel ; l'IPS est en quelque sorte encore assimilé à un concept marketing (*buzzword*) [3].

Conclusion

De manière générale, l'efficacité d'un système de détection d'intrusion dépend de sa "configurabilité" (possibilité de définir et d'ajouter de nouvelles spécifications d'attaque), de sa robustesse (résistance aux défaillances) et de la faible quantité de faux positifs (fausses alertes) et de faux négatifs (attaques non détectées) qu'il génère. Les paragraphes qui précèdent ont pour objectifs à la fois d'illustrer la technicité des attaques actuelles, de montrer la complexité d'une détection d'intrusion et d'expliquer les limites des IDS actuels. Une lutte entre techniques d'intrusion et IDS s'est engagée, les IDS ayant pour conséquence une plus grande technicité des attaques sur IP, et les attaques actuelles imposant aux IDS d'être plus complets et plus puissants [13].

Attention toutefois à l'importance accordée aux IDS ! Selon SecuriteInfo.com (www.securiteinfo.com), les IDS sont actuellement des produits mûrs et aboutis. Ils continuent d'évoluer pour répondre aux exigences technologiques du moment mais offrent d'ores et déjà un éventail de fonctionnalités capable de satisfaire les besoins de tous les types d'utilisateurs. Néanmoins, comme tous les outils techniques, ils ont des limites que seule une analyse humaine peut compenser. A la manière des pare-feu, les détecteurs d'intrusion s'améliorent chaque jour grâce à l'expérience acquise, mais ils deviennent aussi de plus en plus sensibles aux erreurs de configuration et de paramétrage. Par conséquent, il est plus que fondamental de former correctement les personnes chargées de la mise en œuvre et de l'exploitation des IDS. Malheureusement, il semble que subsiste là une grande partie de la difficulté. A ce jour, aucun outil (Nessus, ISS Internet Scanner, NetSonar, Cybercop[21], etc.) ne permet de remplacer l'être humain dans un test d'intrusion.

Enfin, attention également aux discours commerciaux ! [15] montre qu'en réalité, si la détection d'abus fonctionne aujourd'hui plus ou moins correctement (par exemple, Snort), la détection d'anomalie en revanche n'est pas encore fiable (autrement dit, elle ne fonctionne pas).

Si les IDS au niveau réseau ont déjà été énormément étudiés, depuis peu ils le sont aussi au niveau service ; les articles produits sur les IDS réseau sont assez directement applicables au niveau service. Un seul et unique modèle de détection d'anomalie au niveau service est fiable à ce jour[22].

Pour conclure ce paragraphe, « *les IDS/IPS apportent un plus indéniable aux réseaux dans lesquels ils sont placés. Cependant, leurs limites ne permettent pas de garantir une*

---

[21] Respectivement http://www.nessus.org/; http://www.iss.net/; http://www.cisco.com/; http://www.mcafee.com/us/products/mcafee/managed_services/cybercop_asap.htm

[22] Voir l'article de [8] décrivant le seul système IDS sur serveur Web qui fonctionne à ce jour (testé chez Google).

sécurité à 100%, impossible à obtenir. Il faut alors y tendre… Le futur de ces outils permettra de combler ces lacunes en évitant les "faux positifs" (pour les IDS) et en affinant les restrictions d'accès (pour les IPS) » [3].

Références bibliographiques

[1] J.P. Anderson, "Computer Security Threat Monitoring and Surveillance", *Technical Report*, James P. Anderson Company, Fort Washington, Pennsylvania, April 1980.

[2] H. Chang, S.F. Wu, Y.F. You, "Real-time protocol analysis for detecting link-state routing protocol", *ACM Transactions on Information and System Security*, Vol. 4, N° 1, February 2001, pp. 1-36.

[3] F. Cikala, R. Lataix, S. Marmeche, « Les IDS/IPS. Intrusion Detection/Prevention Systems », *Présentation*, 2005.

[4] D.E. Denning, "An Intrusion-Detection Model", *IEEE Transactions on Software Engineering*, Vol. SE-13, N° 2, February 1987, pp. 222-232.

[5] M. Huin, *Pattern matching et détection d'intrusion*, Mémoire DEA Informatique, Loria, Université Henri Poincaré, Nancy1, 2001.

[6] K. Ilgun, R.A. Kemmerer, P.A. Porras, "State transition analysis: a rule-based intrusion detection approach", *IEEE Transactions on Software Engineering*, Vol. 21, N° 3, March 1995, pp.181-199.

[7] H.S. Javits, A. Valdes, "The NIDES statistical component: description and justification", *Technical Report*, SRI International, Computer Science Laboratory, 1993.

[8] C. Kruegel, T. Toth, E. Kirda, "Service Specific Anomaly Detection for Network Intrusion Detection", *17th ACM Symposium on Applied Computing (SAC)*, ACM Press, Madrid, Spain, March 2002, pp. 201-208.

[9] F. Meunier, « Détection d'intrusions: notions avancées de NIDS axées sur le logiciel ManHunt (Recourse Technologies) », *Rapport*, Watch4net, Août 2002.

[10] K. Müller, « IDS - Systèmes de Détection d'Intrusion, Partie I », May 2003, http://www.linuxfocus.org/Francais/May2003/article292.shtml.

[11] K. Müller, « IDS - Systèmes de Détection d'Intrusion, Partie II », July 2003, http://www.linuxfocus.org/Francais/July2003/article294.shtml.

[12] P.A. Porras, P.G. Neumann, "EMERALD: Event Monitoring Enabling Response to Anomaly Live Disturbances", *20th National Information Systems Security*

Conference, National Institute of Standards and Technology, Galthersburg, October 1997.

[13] Herve Schauer Consultants, « La détection d'intrusion… », *Présentation : extrait du cours sécurité TCP/IP du Cabinet HSC*, Mars 2000.

[14] S.E. Smaha, "Haystack: an intrusion detection system", *4th Aerospace Computer Security Applications Conference*, December 1988.

[15] R. State, « Sécurité avancée des réseaux dynamiques » et « Sécurité des systèmes communicants », *cours de Master2 Informatique*, Université Henri Poincaré, Nancy1, Novembre 2005.

[16] S.F. Wu, H.C. Chang, F. Jou, F. Wang, F. Gong, C. Sargor, D. Qu, R. Cleaveland, "JiNao: design and implementation of a scalable intrusion detection system for the OSPF routing protocol", *IEEE Workshop on Information Assurance and Security*, West Point NY, June 2001, pp.91-99.

[17] J. Zimmermann, L. Mé, C. Bidan, "An Improved Reference Flow Control Model for Policy-Based Intrusion Detection", *8th European Symposium on Research in Computer Security (ESORICS)*, October 2003.