Introduction à la détection d'intrusion

12 août 2019

Table des matières

1. « Dura lex, sed lex » 1
2. Les IDS à la rescousse 2
2.1. Les signatures ou le délit de sale gueule 3

L'ordinateur se faisant chaque jour plus présent dans nos vies quotidiennes, la question de la sécurité informatique prend elle aussi de l'importance. Avec l'essor d'Internet et du « tout connecté », nous dépendons de plus en plus de la fiabilité de nos appareils, sans même parfois nous en rendre compte. Je vous propose, grâce à cet article, de faire un « premier pas » dans cet univers un peu à part en vous présentant les notions clefs et le vocabulaire de base de la détection d'intrusions.

Cet article est basé, de manière directe et indirecte, sur les cours de Ludovic Mé, dont vous pouvez retrouver une présentation extrêmement intéressante ici (clôture du SSTIC 2013).

1. « Dura lex, sed lex »

En sécurité informatique, il convient en premier lieu « d'exprimer une politique de sécurité ». D'une certaine manière, il s'agit de définir les propriétés que l'on veut voir respecter au sein de notre système d'information. Le système d'information est l'ensemble des composants informatiques qui peuvent interagir entre eux. Prenons un exemple : pour une entreprise, un système d'information peut potentiellement ressembler à quelque chose comme ça :

1. tout le matériel possédé par l'entreprise (postes de travail des employés, serveurs, etc.) ;
2. tous les logiciels utilisés par l'entreprise (intranet, extranet, application de pose de congés, etc.) ;
3. tous les utilisateurs (on parle souvent de *sujets*) amenés à se connecter ;
4. les données produites et échangées sur le réseau.

Une manière beaucoup plus schématique de voir un système d'information est d'identifier trois composants primaires :

— les *sujets* du système (ses acteurs, humains ou non) ;
— les *objets* de ce système (globalement, les données qui sont manipulées) ;
— les *actions* que les sujets peuvent effectuer sur les objets.

La politique de sécurité va permettre de spécifier qui peut faire quoi sur quoi dans un système d'information. Dans un premier temps, il ne s'agit pas de configurer telle ou telle solution de sécurité, mais bien de spécifier les besoins en terme de sécurité, de la même façon que l'on peut concevoir un logiciel ou écrire le script d'un film, finalement.

Une fois cette politique exprimée, il faut s'assurer de son respect. Cela passe par la bonne configuration des outils utilisés[1], mais pas seulement. Il est aussi nécessaire d'utiliser des mécanismes de sécurité afin de structurer la façon dont les utilisateurs accèdent et interagissent aux différents services de votre système d'information. Il s'agit du rôle des mécanismes de contrôle d'accès. Un tel mécanisme doit déterminer si un *sujet* peut accéder à un *objet* d'un système. Prenons un nouvel exemple : celui du système d'exploitation. Dans Linux, on peut dire en simplifiant les choses que :

— les *sujets* sont les utilisateurs du système[2] pour lesquels sont exécutés les programmes ;
— les *objets* sont les fichiers stockés sur le disque dur ;
— les *actions* sont la lecture, l'écriture et l'exécution d'un fichier.

Sous Linux, le *Discretionary Access Control* (DAC)[3] est le mécanisme de contrôle d'accès utilisé par défaut. Il se manifeste par le biais des commandes `chmod` et `chown`, ou encore l'affichage de `ls -l` (les fameux rwx). Je ne vais pas rentrer dans les détails du fonctionnement du DAC, pas plus que dans ceux de ses nombreux concurrents. Retenez juste l'idée principale : le contrôle d'accès permet de déterminer si un *sujet* peut effectuer une *action* sur un *objet*.

un « compte par défaut » avec un mot de passe connu. Ce compte peut posséder certains droits et le fait de ne pas le supprimer peut potentiellement permettre à un attaquant d'entrer dans le système.

discrétionnaire](http ://fr.wikipedia.org/wiki/Contr%C3%B4le_d%27acc%C3%A8s_discr%C3%A9tionnaire) en bon français est un mécanisme de contrôle d'accès qui définit pour chaque objet du système les droits (lecture, écriture, exécution par exemple) des entités de ce système. Il est dit discrétionnaire, car le sujet qui possède l'objet peut décider de déléguer ses droits.

Ces mécanismes qui visent à assurer le respect de la politique de sécurité sont donc « bloquants » et sont censés protéger un système d'information de potentielles attaques. Néanmoins, ils restent des programmes comme les autres et sont sujets eux-mêmes aux vulnérabilités et aux failles. C'est pourquoi des détecteurs d'intrusions (ou IDS pour *Intrusion Detection System*) ont été développés, ils sont un peu les agents de sécurité « de la dernière chance ».

2. Les IDS à la rescousse

Un détecteur d'intrusions peut être vu comme une caméra de sécurité[4]. Une caméra de sécurité est placée à un endroit stratégique et elle capture tous les mouvements à cet endroit. Elle peut ainsi filmer un homme en train de voler la Joconde, mais elle ne peut en aucun cas l'arrêter. C'est une caméra, elle n'a pas de bras[5]. La seule chose qu'elle peut faire, c'est analyser les images qu'elle capture et prévenir la sécurité en cas de comportement suspect.

1. Petit exemple, comme cela. Sur certaine solution logicielle, il existe parfois
2. Au sens UNIX, ce qui veut dire qu'il ne s'agit pas forcément d'un humain. Ainsi, il n'est pas rare qu'un serveur web possède son propre compte utilisateur !
3. Le *Discretionary access control* ou [contrôle d'accès
4. En réalité, on parle plus volontiers de « sondes » dans le cadre de la

détection d'intrusions. L'idée reste néanmoins la même.

Si vous avez compris cela, vous avez compris l'utilité d'un IDS.

Il existe plusieurs types de détecteurs fonctionnant à différents niveaux d'abstractions. On parlera de **N**IDS pour un détecteur surveillant les paquets réseaux, de **H**IDS pour un détecteur s'assurant du bon fonctionnement de l'hôte (*host*) comme le système d'exploitation et l'on parlera plus rarement de **A**IDS pour un détecteur dédié à une application en particulier.

Il est possible de classer les différents IDS selon deux approches majeures : la détection *signature-based* et la détection comportementale.

2.1. Les signatures ou le délit de sale gueule

Si vous utilisez Windows comme système d'exploitation, il y a de grandes chances pour que vous ayez installé un antivirus. Si vous avez prêté un minimum d'attention à ce dernier, vous avez pu parfois voir s'afficher des messages du genre « La base de données de signatures a été mise à jour. » Ces signatures caractérisent l'approche majoritairement utilisée dans les solutions commerciales d'antivirus et de détecteurs d'intrusion actuelles, on parle d'ailleurs de détection *signature-based*.

Une signature peut être assimilée à un avis de recherche. Elle décrit précisément ce que le détecteur doit trouver : les preuves de l'exploitation d'une faille de sécurité. Pour les antivirus, il va s'agir du code binaire des *malwares* s'étant ajouté au contenu d'un exécutable jusqu'alors sain. Il existe des bases de données répertoriant ces codes binaires malicieux, régulièrement mises à jour. Votre antivirus va « simplement » regarder le contenu de chaque exécutable à la recherche de code malveillant. Malheureusement, une menace inconnue restera sous le radar. C'est pour cela que votre antivirus se met à jour aussi souvent que possible : afin d'éviter d'être vulnérable trop longtemps aux virus qui sont lâchés dans la nature chaque jour[6].

cybercriminels ne prenant pas leurs week-ends.

Il existe une autre approche, suivant une logique complètement inversée, qui vise à pouvoir reconnaître des attaques dont elle ne sait rien *a priori*.

2.1.1. Un exemple : Snort

http://assets.sourcefire.com/2009/03/23/snort-

FIGURE 2. – Le logo Snort, un cochon

5. Pas de chocolat.
6. Ce qui représente un nombre non négligeable, les vilains

Publié sous licence libre (GPL), Snort est un *Network-based* IDS[7] (ou NIDS). Son travail se résume assez simplement : placé à l'entrée de votre réseau, il analyse tous les paquets qu'il récupère, à la recherche d'anomalies. Ces anomalies, il les repère grâce à des signatures textuelles qui ressemblent à ça :

seule différence entre un IDS et un IPS, c'est que là où le premier va lever une alerte pour ce qu'il pense être une action malveillante, le second va la bloquer.

7. Il est aussi présenté comme un IPS (*Intrusion Prevention System*). La