Détection des intrusions et des vulnérabilités

1. Les systèmes IDS

a. Les limitations des pare-feu

Les pare-feu dans leur fonctionnement historique filtrent les paquets sur les valeurs contenues dans les en-têtes de couche réseau ou transport, et donc sur les adresses IP ou les ports utilisés. Pour contourner la protection apportée par les pare-feu, de nombreuses applications utilisent des ports courants (tcp 80 notamment) pour faire passer leur propre trafic applicatif. Les pare-feu, souvent configurés pour laisser passer les flux sur ces ports courants, n'y voient que du feu.

Pour assurer un meilleur contrôle, il faut utiliser un équipement plus élaboré, capable de regarder et d'analyser le trafic applicatif, directement et sans se faire tromper par l'annonce d'un port erroné. Ces équipements sont appelés « sondes » en français parce que sondant l'intérieur des paquets, ou encore IDS (*Intrusion Detection System*).

b. Techniques d'analyse

Pour identifier les trafics malicieux, les IDS disposent de trois techniques : la détection d'anomalies, l'analyse de protocoles et l'analyse de signatures.

La détection d'anomalies a pour objet de détecter un comportement anormal, comme par exemple un volume ICMP démesuré, qui indiquerait que l'on est la cible ou l'émetteur d'une attaque par dénis de service.

L'analyse de protocole ne cherche pas à repérer une action réellement malicieuse, mais plutôt un trafic applicatif qui ne respecterait pas à la lettre les règles de fonctionnement des protocoles employés. C'est un peu l'histoire du braqueur de banque qui se fait arrêter bêtement parce que ses pneus sont lisses.

Enfin, l'analyse de signatures permet d'identifier des attaques ou comportements malsains déjà référencés. C'est la technique la plus efficace et qui n'est pas sujette à erreur, puisqu'on ne gère que des attaques ou intrusions ayant déjà eu lieu chez un tiers, et donc dûment identifiées.

c. Sources d'information

Les techniques d'analyse, qu'il s'agisse d'analyse de signatures, de protocoles ou de détections d'anomalies s'appuient sur des informations qui évoluent avec le temps. Il est évident que l'analyse de signature ne peut s'appliquer que si l'IDS connaît la signature de l'attaque en cours. De plus, la nature des menaces peut évoluer. Par exemple, un hôte qui aurait envoyé de gros volumes de trafics SMTP dans les années 80 indiquerait qu'un serveur de messagerie fonctionne bien. La même situation aujourd'hui pourrait montrer que l'hôte en question est infecté par un cheval de Troie et qu'il envoie de gros volumes de SPAM.

Les IDS doivent impérativement récupérer à intervalle régulier les mises à jour de leurs techniques d'analyse ainsi que les bases de signatures. Les éditeurs d'IDS doivent systématiquement maintenir leurs bases d'informations à jour, et les administrateurs des IDS doivent tout aussi régulièrement télécharger ces bases.

De nombreux organismes, associations et entreprises permettent de se tenir au courant des évolutions en matière de techniques d'intrusion et de nuisance. Il est recommandé de connaître l'existence des principaux, et dans le cadre d'une administration réseau avec prise en compte de la sécurité, d'assurer une veille technologique sur ces domaines.

Principaux organismes de veille et de recherche	
Bugtraq	Liste de diffusion dédiée à l'annonce des vulnérabilités, leur exploitation et leur correction.
CERT	*Computer Emergency Response Team*. Cette organisation étudie les vulnérabilités, effectue de la recherche sur les évolutions en terme de réseaux et de sécurité, et propose des services liés à la sécurité.
CIAC	*Computer Incident Advisory Capability*. Organisme de veille et de recherche géré par le U.S. Department Of Energy.

2. SNORT

© ENI Editions - All rights reserved - benoit bermond

a. Les composants

Snort est le plus connu des IDS libre. Il analyse tout trafic et apporte un complément de sécurité appréciable, voire indispensable sur un réseau. Snort est composé d'un moteur d'analyse, et d'un ensemble de règles.

Snort est composé d'un service et de fichiers de configuration généralement situés sous **/etc/snort**. Le fichier de configuration principal est **snort.conf**. Les règles appliquées sont situées dans un sous-répertoire **rules**.

Snort dispose également d'une commande **oinkmaster** de mise à jour des règles qui trouve sa configuration dans un fichier **oinkmaster.conf**.

b. Gestion des sources d'information

SNORT exploite des fichiers de règles qui doivent être téléchargés sur le site web de l'éditeur.

Déclaration d'un fichier de règles dans oinkmaster.conf

```
url = http://www.snort.org/snort-rules/fichier_règles
```

Où *fichier_règles* représente le fichier des règles au format tar.gz. Il est nécessaire d'être abonné auprès de l'éditeur mais d'autres sites web proposent des fichiers de mise à jour gratuits. Naturellement, la qualité du suivi dépend des gestionnaires de ces fichiers de règles.

Après toute modification du fichier de définition des signatures, et par la suite à intervalle régulier par une planification cron, il faut demander à snort de télécharger ses nouvelles règles. Cette opération se réalise avec la commande **oinkmaster**.

Chargement des règles

```
oinkmaster -o rep_règles
```

Où *rep_règles* représente le répertoire qui contient les règles de fonctionnement de snort, souvent **/etc/snort/rules**. Les fichiers de règles doivent être appelés dans le fichier **snort.conf** par le paramètre **include**, ce qui est le cas avec les paramètres par défaut et les signatures de l'éditeur.

c. Gestion des alertes

Quand Snort détecte un trafic malicieux, il laisse une trace dans un fichier journal via **syslog**, et envoie une copie du paquet dans un fichier au format tcpdump (format libpcap, visible avec wireshark par exemple). Il a aussi la possibilité d'envoyer les informations vers une base de données (Oracle, MySQL, et PostGreSQL sont entre autres supportés).

Exemple de déclaration d'utilisation de syslog dans snort.conf

Cette déclaration indique que les éléments doivent être envoyés vers un serveur syslog dont l'adresse IP est ip_serveur, sous la catégorie « alerte ».

```
output  alert_syslog: host=ip_serveur, LOG_ALERT
```

3. OpenVAS

OpenVAS pour *Open Vulnerability Assessment scanner* est une variante libre du scanner de vulnérabilités Nessus.

a. Le serveur OpenVAS

Le serveur est le cœur de la suite applicative OpenVAS, il scanne et analyse les hôtes du réseau à la recherche de vulnérabilités connues (NVT : *Network Vulnerability Tests*).

b. Les clients OpenVAS

© ENI Editions - All rights reserved - benoit bermond

Les clients OpenVAS sont des éléments logiciels en ligne de commande ou avec une interface graphique qui assurent l'analyse des hôtes du réseau à la recherche de vulnérabilités pour renvoyer les résultats au serveur.

c. Récupération des vulnérabilités

OpenVas propose une source publique de vulnérabilités connues sous le nom OpenVas NVT Feed. Il permet aux serveurs de se tenir au courant des dernières vulnérabilités connues, et contient plus de 15000 NVT (*Network Vulnerability Tests*).

© ENI Editions - All rights reserved - benoit bermond