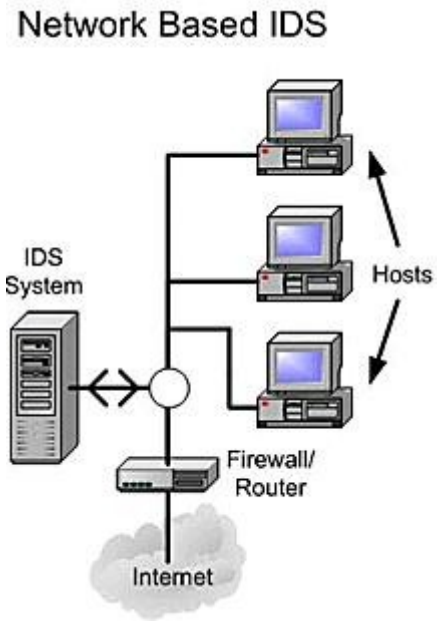
# Introduction aux IDS

## Présentation

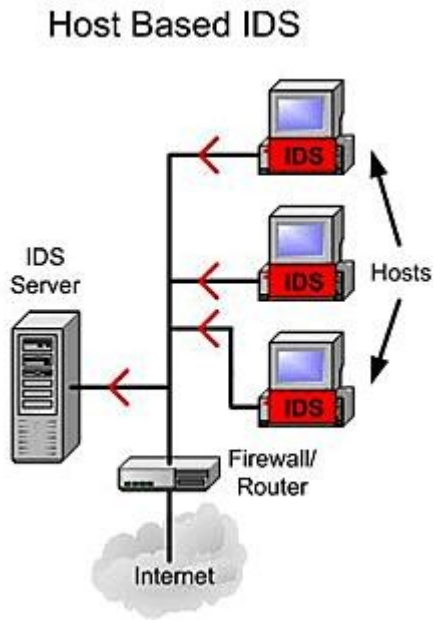
Les IDS sont des outils permettant de détecter les attaques/intrusions du réseau sur lequel il est placé. C'est un outil complémentaire aux firewall, scanneurs de failles et anti virus.

Il existe deux niveaux d'IDS : les IDS systàmes et les IDS réseaux.

- Les IDS systèmes (Host IDS) analysent le fonctionnement et l'état des machines sur lesquels ils sont installés afin de détecter les attaques en se basant sur des démons (tels que syslogd par exemple). L'intégrité des systèmes est alors vérifiée périodiquement et des alertes peuvent êtres levées. Par nature, ces IDS sont limités et ne peuvent détecter les attaques provenant des couches réseaux (tels que les attaques de type DOS).




- Les IDS réseaux (Network IDS), quant à eux, analysent en temps réel le trafic qu'ils aspirent à l'aide d'une sonde (carte réseau en mode "promiscuous"). Ensuite, les paquets sont décortiqués puis analysés. En cas, de détection d'intrusion, des alertes peuvent être envoyées.




## Les différents types d'IDS

Les IDS disposent de deux approches différentes, afin de déceler les intrusions :

- les IDS à signature :

  Généralement, les IDS réseaux se basent sur un ensemble de signatures qui représentent chacune le profil d'une attaque. Cette approche consiste à rechercher dans l'activité de l'élément surveillé (un flux réseau) les empreintes d'attaques connues, à l'instar des anti virus.

  Une signature est habituellement définie comme une séquence d'événements et de conditions relatant une tentative d'intrusion. La reconnaissance est alors basée sur le concept de "pattern matching" (analyse de chaînes de caractères présente dans le paquet, à la recherche de correspondance au sein d'une base de connaissance). Si une attaque est détectée, une alarme peut être remontée (si l'IDS est en mode actif, sinon, il se contente d'archiver l'attaque).

- Les IDS comportementaux :

  Les IDS comporteaux ont pour principale fonction la détection d'anomalie. Leur déploiement nécessite une phase d'apprentissage pendant laquelle l'outil va apprendre le comportement "normal" des fluxs applicatifs présents sur son réseau.

  Ainsi, chaque flux et son comportement habituel doivent îtres déclarés ; l'IDS se chargera d'émettre une alarme, si un flux anormal est détecté, et ne pourra bien entendu, spécifier la criticité de l'éventuelle attaque.

  Les IDS comportementaux sont apparus bien plus tard que les IDS à signature et ne bénéficient pas encore de leur maturité. Ainsi, l'utilisation de tels IDS peut s'avérer

délicate dans le sens où les alarmes remontées contiendront une quantité importante de fausses alertes. Ce problème peut être résolus en généralisant la déclaration des flux mais cette opération peut entrainer une transparence de l'IDS face à la détection de certaines attaques.

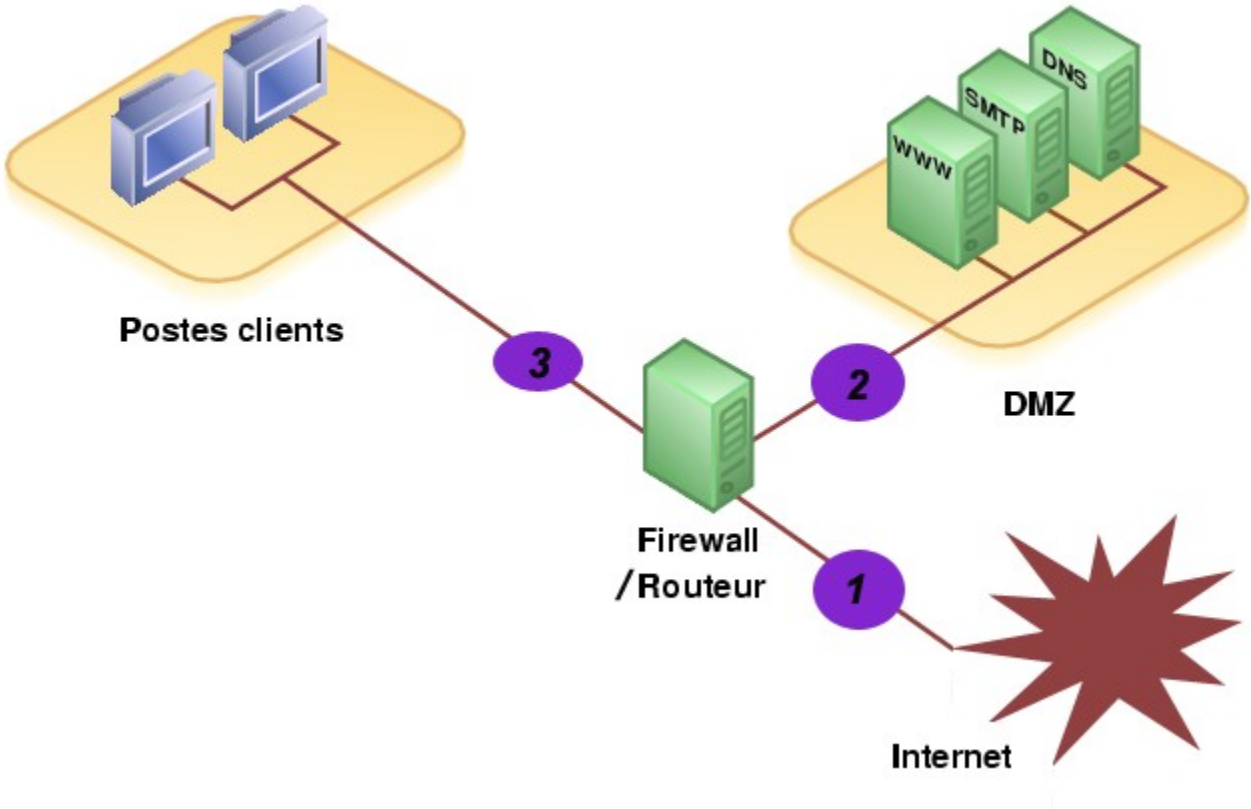
# Les IDS par la pratique (gratuite) : Snort

## Mise en place d'un IDS

## Où positionner son IDS ??

Il existe plusieurs endroits stratégiques où il convient de placer un IDS.

Le sché ma suivant illustre un réseau local ainsi que les trois positions que peut y prendre un IDS :




- **Position ( 1 ):** Sur cette position, l'IDS va pouvoir détecter l'ensemble des attaques frontales, provenant de l'extérieur, en amont du firewall. Ainsi, beaucoup (trop?) d'alertes seront remontées ce qui rendra les logs difficilement consultables.
- **Position ( 2 ):** Si l'IDS est placé sur la DMZ, il détectera les attaques qui n'ont pas été filtrées par le firewall et qui relèvent d'un certain niveau de compétence. Les logs seront ici plus clairs à consulter puisque les attaques bénines ne seront pas recensées.
- **Position ( 3 ):** L'IDS peut ici rendre compte des attaques internes, provenant du réseau local de l'entreprise. Il peut être judicieux d'en placer un à cet endroit étant donné le fait que 80% des attaques proviennent de l'intérieur. De plus, si des trojans ont

contaminé le parc informatique (navigation peu méfiante sur internet) il pourront êtres ici facilement identifiés pour être ensuite éradiqués.

Idéalement, on placerait des IDS sur les trois positions puis on délèguerait la consultation des logs à l'application "acid" (cf http://acidlab.sourceforge.net/) qui permet d'analyser les alertes et d'en présenter clairement les résulats via une interface web complète. Si une seule machine peut être déployée, autant la mettre sur la position 2, crutiale pour le bons fonctionnement des services.

## Installation et configuration de Snort

Snort est un IDS gratuit disponible dans sa version 2.2.20 (www.snort.org). A l'origine, ce fut un sniffer qui connnu une telle évolution qu'il fut vite adopter et utiliser dans le monde de la détection d'intrusion en s'appuyant sur une base de signature régulièrement enrichie par le "monde du libre".

Sous Linux (comme sous windows) son installation est simple et se résume (pour linux) par les commandes suivantes, une fois l'archive téléchargée dans le répertoire "/usr/local/snort"

```
cd /usr/local/snort
tar -xvf SNORT-2.2.*.tar.gz
./configure --mysql=/usr/lib/mysql
make
make install
```

Ainsi, on effectue le lien entre snort et mysql afin d'utiliser une base de donnée pour la détection d'intrusion. L'outil sera alors bien plus riche et réactif. Il est possible d'utiliser d'autres solutions de bases de données en renseignant les variables d'environnement corespondantes :

```
--with-postgresql=$PATH_POSTGRE : pour une base PostegreSQL
--with-oracle=$ORACLE_HOME : pour une base Oracle
-with-odbc=$PATH_ODBC : pour une base de données Microsoft SQL server
```

Afin d'indiquer à snort la base où il doit envoyer ses alertes, il convient de modifer la ligne suivante, dans le fichier de configuration "snort.conf":

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

par

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

Pour l'exemple, l'utilisateur "user" a pour mot de passe "snort_pwd", et le nom de la base MySQL utilisée par snort est "snort" (le serveur concerné est la machine où tourne snort).

Sous MySQL, il faut ensuite créer la base SNORT ainsi que l'utilisateur user en lui indiquant les bon paramètres par la commande suivante :

```
insert into user values('localhost', 'user_snort', password('password')
, 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y',
'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', '', '',
'', '', 'Y', 'Y', 'Y');
grant ALL PRIVILEGES ON SNORT.* TO user@localhost IDENTIFIED BY
'password' WITH GRANT OPTION;
```

Il est alors possible de lancer l'outil snort par la commande suivante, si nous voulons utiliser la base de donnée (au lieu de simple fichier texte de log, cf commande bis) :

```
/usr/local/snort*/src/snort -c /etc/snort/snort.conf

commande bis :
/usr/local/snort*/src/snort -c /etc/snort/snort.conf -i eth0 -D
```

## Lancement de Snort

Snort dispose de plusieurs modes de fonctionnements qui sont les suivants :

- **Mode écoute :** Ce mode permet de lancer snort en mode sniffer et permet d'observer les paquets que l'IDS perçoit ("snort -v")
- **Mode "log de paquets" :** Le log de paquet permet l'archivage des paquets circulant sur le réseau de l'IDS. Il permet, grâce à ses arguments des opérations intéressantes permettant de limiter les logs à certains critères, comme une plage d'adresse IP (ex : "snort -l ../log/snort -h 192.168.0.0/24")
- **Mode "détection d'intrusion" :** Le mode IDS permet à snort d'adopter un comprtement particulier en cas de détection d'une (succession) de chaînes de caractères dans les paquets interceptés ; selon les règles définies dans les fichiers d'extension ".rules" du répertoire /rules ("snort -A full -d -l ../log -c $SNORTPATH/snort.conf").

## Fonctionnement des règles de Snort

Les règles de snort sont décrites dans un langage simple et suivent le schéma suivant :

**l'en-tête de règle** qui contient

- l'action de la règle (la réaction de snort);
- le protocole qui est utilisé pour la transmission des données (snort en considère trois: TCP, UDP et ICMP);
- les adresses IP source et destination et leur masque;
- les ports source et destination sur lesquels il faudra vérifier les paquets.

**les options de la règle**(entre parenthèse) qui contiennent

- le message d'alerte;

- les conditions qui déterminent l'envoi de l'alerte en fonction du paquet inspecté.

L'exemple de règle suivant est simple et permet de détecter les tentatives de login sous l'utilisateur root, pour le protocole ftp (port 21) :

```
alert tcp any any -> 192.168.1.0/24 21 (content: "USER root"; nocase;
msg: "Tentative d'accès au FTP pour l'utilisateur root";)
```

Les messages en direction de cette plage d'adresse IP effectuant une tentative de login root ("USER root" contenu dans le paquet) auront pour conséquence la génération de l'alerte "Tentative d'accès au FTP pour l'utilisateur root".

Ainsi, il s'agit de renseigner ces variables par les champs que l'on pourrait trouver dans les paquets propres aux intrusion tels que les "shell code" que les "exploits" utilisent afin d'insérer des instructions malicieuses dans des programmes sujets aux "buffer overflows". Ainsi, ils obtiennent des accès privilégiés sur la machine et peuvent en prendre le contrôle.

Pour comprendre le fonctionnement des règles de snort, un exemple simple sera employé. Il s'agit ici de détecter la présence d'un ping provenant d'une station de type Windows et de lever une alerte, lorsque celle ci est détectée.

Pour cela, il nous faut récolter une trace que pourrait laisser une telle station. Il convient alors d'effectuer un ping à partir de cette station, tout en sniffant les paquets (tcpdump ou snort -v) afin d'avoir sa trace complète

Une fois les paquets identifiés, il s'agit de trouver les chaînes redondantes contenues dans ce paquets

La trace suivante montre un paquet typique provenant d'un tel ping :

```
[root@localhost etc]# tcpdump icmp -vv -X
tcpdump: listening on eth0, link-type EN10MB (Ethernet),
capture size 96 bytes 14:27:41.472192
IP (tos 0x0, ttl 128, id 12102, offset 0,
flags [none], length: 60) windows > 192.168.0.101: icmp 40:
echo request seq 24300

0x0000 4500 003c 2f46 0000 8001 895e c0a8 0064 E..</F.....^...d
0x0010 c0a8 0068 0800 ea5b 0400 5f00 6162 6364 ...h...[.._.abcd
0x0020 6566 6768 696a 6b6c 6d6e 6f70 7172 7374 efghijklmnopqrst
0.0030 7576 7761 6263 6465 6667 6869           uvwabcdefghi
...
...
```

Ainsi, nous pouvons remarquer que la chaîne "abcdefghij..." est présente dans tous les paquets générés par les ping lancés pour la reconnaissance. On peut alors détecter de tels paquets en créeant la règle snort correspondante.

Sa syntaxe est simple :

```
alert icmp any any -> any any (msg:"Ping Windows détecté";\
content:"abcdefghijklmnop"; depth:16;)\
```

Enormément d'options sont disponibles afin d'affiner au mieux l'identification des paquets véhiculés dans le réseau.

### Réactions de Snort

Les alertes émises par snort peuvent être de différentes nature. Par exemple, on peut spécifier à snort de rediriger l'intégralité des alarmes sur la sortie standard et ainsi observer l'évolution des attaques. Cependant, ceci nécessite une présence attentive devant un écran, ce qui peut parraître rebutant.

Snort ne permet pas d'envoyer de mail directement, étant donné son rôle premier de sniffer qui est gourmand en ressource. L'envoi de mail d'alerte ralentirait snort d'une telle manière que beaucoup de paquets seraient "droppés" (éjectés). Qu'à cela ne tienne, Snort a été conçu pour interagir facilement avec le deamon syslogd afin que ce dernier génère les futurs logs qui peuvent être instantannément parsés par d'autres applications telles que "logsurfer" ou encore "swatch" respectivement :
http://www.obfuscation.org/emf/logsurfer/snort.txt
http://www.theadamsfamily.net/~erek/snort/snort-swatch.conf.txt.

Ces derniers permettent d'envoyer un mail avec les logs attachés en pièces jointes, et donc aussi des sms, si l'entreprise dispose d'un tel serveur.

Snort est aussi capable d'adopter des comportements visant à interdire l'accès à certaines adresses IP, dans le cas où ces dernières auraient tenté de pénétrer le réseau. L'IDS peut alors interagir avec le firewall afin qu'il mette à jour ses règles d'accès pour empêcher tout contact avec l'éventuel pirate.

Il faut cependant ce méfier de cette possibilité puisqu'en cas de mauvaise configuration, elle peut facilement entrainer la coupure totale du réseau. Il convient alors d'utiliser une solution robuste, telle que "snortsam" (www.snortsam.net) et de lire attentivement les documentations.

# Précision sur les IDS

### Les limites des IDS

### Problématique

Par nature, les IDS vont remonter énormément d'alertes, si ils ne sont pas configurés convenablement. L'essentiel réside dans le compromis effectué entre la quantité d'alertes remontés et la finesse de ces dernières. Il faut donc prendre soin d'inclure dans le fichier de configuration le fichier ".rule" nécessaire, en fonction des règles établies par le firewall. Par exemple, si un service est totalement interdit, il est presque inutile d'inclure les signatures associées.

## L'art du scann

### *Dillution temporelle des scanns ...*

Les IDS se basent sur certains critères autres que les signatures, afin de détecter les attaques réseaux. Prenons l'exemple d'un scann de port banal ciblant une machine du réseau de l'IDS. Ce dernier sera automatiquement détecté étant donné la fréquence des scanns et une alerte sera remontée. Cependant, les outils de scann disposent de plus en plus d'options et certains, comme "nmap" dispose de précaution (mode "Paranoïd", "Sneaky") qui permettent d'adapter la fréquence de ces scanns, afin de leurrer l'IDS.

Par exemple, l'option -T permet de changer la fréquence des scann afin de passer en dessous du seuil qui déclenche l'adresse. La commande suivante permet d'effectuer des scanns furtifs et discrets: "#nmap -P0 -sS **-T Sneaky** -p 21, 20 ADDERSS_IP"

### *Le revers mapping*

De même, certaines techniques de scann telles que le "reverse mapping" disposent d'une élaboration accrue. Elle consiste à envoyer des paquets à un ensemble de machines, avec le flag RST positionné à "1". Ainsi, si l'on reçoit en retour le message ICMP "Host Unreachable" d'une machine on sait qu'elle n'est pas présente. Par contre, si l'on ne reçoit rien, il y a de grande chance pour que la machine visée existe. L'outil "hping" (option "-R -c 1 -p IP_MACHINE") permet de telles opérations. Le mauvais trafic peut alors être détecté selon la configuration de snort (car le port source par défaut des paquets générés par snort est 0). En utilisant l'argument -s, on peut éviter d'être reconnu par l'IDS en mettant un port normal).

### *Inconvénients du mode promiscuous*

Par nature, les IDS doivent mettre leur carte réseau en mode "promiscuous" ce qui va leur permettre de 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 affranchi la machine de la couche 2 et le filtrage sur les adresses MAC n'est plus activé. Il se peut alors que la machine répond à certains messages (icmp echo request généré avec l'outil "nemesis"). Si la machine n'est pas en mode "promiscuous", elle ignorera le paquet sinon, elle répondra.

Ce mode génère des accès mémoire et processeur important et 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 tels que le "sniffing". On pourra alors s'appuyer sur ces données pour valider la présence d'un IDS.

### *Surchage de l'IDS*

Il est aussi possible d'envoyer une quantité importante d'attaques bénignes afin de surcharger les alertes de l'IDS, et ainsi glisser une attaque plus furtive qui aura du mal à être identifiée, si le flot d'informations généré est suffisant.

### *Scan Zombie*

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 adéquates afin de se faire passer pour une autre machine, puis de déterminer le comportement à adopté, en fonction de la réaction de la cible. Dans tous les cas, l'assaillant ne peut être directement découvert par l'IDS.

# Conclusion

Nous l'avons vu, la sécurisation d'un réseau est une étape délicate permettant de protéger une entreprise des risques les plus courants, émanant aussi bien de l'internet que de son propre réseau local.

Une gamme "complète" de solutions telles que les anti virus, scanneurs de failles et firewall permet d'obtenir une sécurité presque convenable face aux attaques les plus courantes. Ces dernières sont d'ailleurs en évolution quotidiennes et de nombreuses failles sont découvertes et exploitées chaque jour.

Il faut alors prendre au sérieux les risques provenant du réseau et analyser régulièrement ses flux, afin d'y déceler les anomalies, pouvant prouver une intrusion. Cette tâche serait aussi rebutante que fastidieuse si les IDS n'avaient pas été développés et certains, comme "Snort" font preuve d'une maturité exemplaire.

Cependant, le monde des pirates est un milieu où les failles découvertes (et exploits associés) sont diffusés, parfois à grande échelle. Cependant, ces fuites sont peu fréquentes, et il convient de consulter plusieurs forums "underground" afin d'être (à peu près) à jour sur les dernières failles. Malheureusement, quand une information émane de ces milieux, elle est souvent désuée et non exploitable.

On peut alors imaginer les **risques potentiels** étant donné les limites de chaque "rempart" du réseau (exploits sur firewall, anti virus détourné, attaques camouflées pour contourner l'IDS...).

De plus, un facteur essentiel est à prendre en considération : La réactivité de l'administrateur. En effet, la solidité de chaque solution employée dépend essentiellement de ses mises à jour qui doivent être régulièrement effectuées.

Il est important de noter que **le risque nul d'être piraté n'existe pas** et il faut s'avoir s'appuyer au mieux sur les outils (nouvellement) disponibles afin de tendre vers cet idéal.