

TP SNORT

Auteur : Guillaume Arcas
Version : 1.1

Ce document présente un cas pratique d'utilisation du logiciel Snort comme IDS. L'objectif est d'illustrer les notions théoriques abordées dans le cours Snort.

INTRODUCTION

Nous nous appuierons dans ce cours sur le schéma suivant :

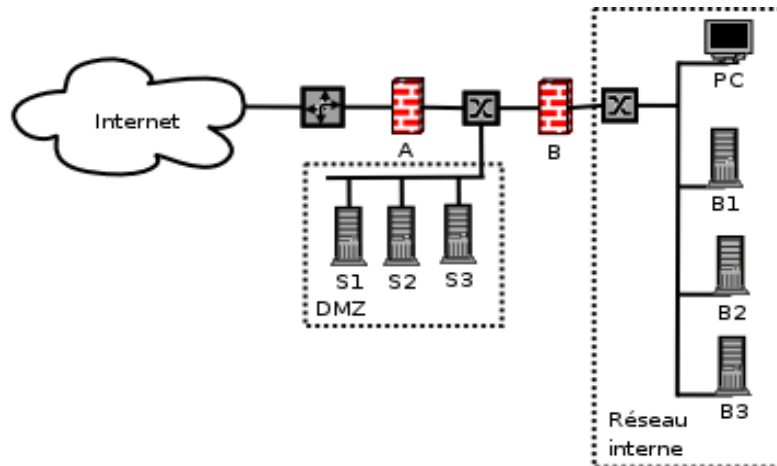


Figure 1 - Schéma du SI

Afin d'augmenter la visibilité sur le niveau de sécurité et d'exposition du S.I., il a été décidé de déployer un système de détection d'intrusions réseau (NIDS). Le logiciel retenu pour mettre en oeuvre cette fonctionnalité est Snort.

Voici le plan d'adressage de notre S.I. :

Internet	192.168.1.0/24 (En réalité il s'agira des IP publiques fournies par l'ISP.)
DMZ	192.168.2.0/24
Réseau interne	192.168.3.0/24

Les points d'écoute suivants ont été choisis :

Zone Internet	Une sonde Snort est branchée en amont du pare feu A Elle permet de quantifier (nombre) et de qualifier (type) les attaques qui viennent d'Internet et dirigées contre le S.I.. Cette mesure est effectuée devant le pare feu.
---------------	--

<p>Zone DMZ</p>	<p>Une sonde Snort est branchée sur le commutateur (switch) entre les pare feux A et B. Cette sonde analyse le trafic en provenance d'Internet vers la DMZ et en provenance du réseau interne vers le DMZ.</p> <p>L'objectif est double :</p> <ul style="list-style-type: none"> - valider que le filtrage mis en œuvre sur le pare feu A est efficace et protège donc les réseaux DMZ et Interne des attaques qui viennent de l'extérieur ; - identifier les attaques ou les anomalies qui viendrait du réseau Interne vers la DMZ ou vers l'extérieur (dans ce dernier cas, on cherchera à détecter les traces liées à des infections virales).
<p>Zone Interne</p>	<p>Une sonde Snort est branchée en aval du pare feu B.</p> <p>Elle analyse le trafic sur le réseau interne et permet, par exemple, de détecter une machine infectée par un ver ou un virus. Elle est aussi capable de détecter des attaques menées depuis l'intérieur du réseau interne vers d'autres machines de ce réseau.</p>

Le schéma ci-dessous indique le positionnement des sondes :

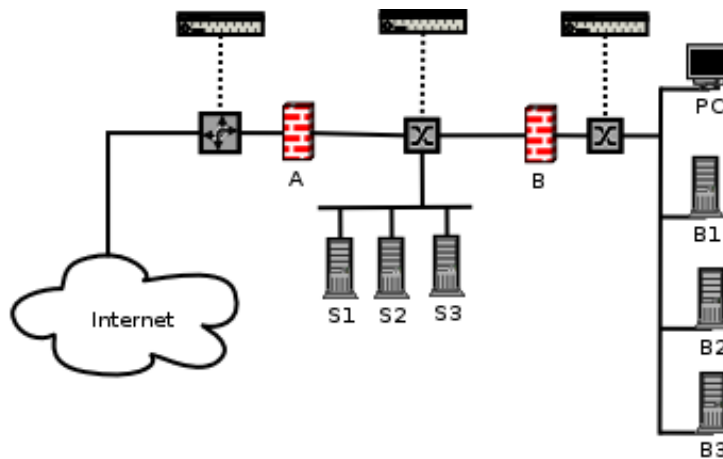


Figure 2 - Positionnement des sondes

ARCHITECTURE DE L'IDS

L'IDS se compose :

- de trois sondes Snort ;
- d'un serveur de base de données qui est utilisé pour stocker les alertes de manière centralisée ;
- d'une console utilisée pour consulter la base d'alertes et pour administrer les sondes ;
- d'un réseau entièrement dédié à l'IDS et indépendant des autres réseaux.

Le schéma ci-dessous présente l'IDS intégré au S.I. :

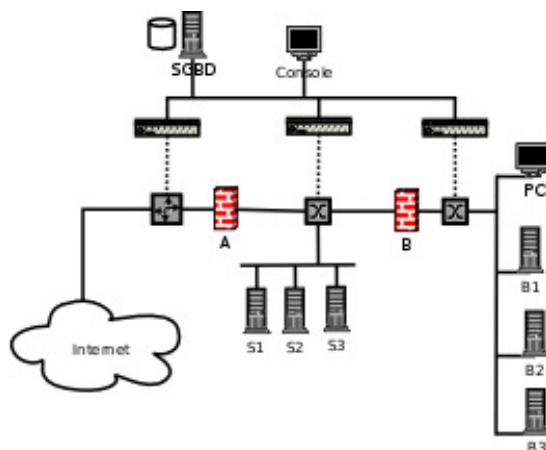




Figure 3 - Architecture de l'IDS

Note : Il est possible de fusionner certains composants sur une seule machine. Par exemple le serveur SGBD et la console.

Système d'exploitation

Le système d'exploitation retenu pour les sondes et pour les éléments de l'IDS (SGBD et console) est la distribution Debian avec un noyau Linux 2.6. On aurait très bien pu choisir une distribution Linux Fedora Core 4 (FC4). Cette distribution est l'héritière de la distribution Red Hat 9. Elle présente l'avantage d'être construite sur un noyau Linux 2.6 beaucoup plus performant que les précédents noyaux 2.4 en matière de traitement du trafic réseau.

Dans la suite de ce document, les logos suivants seront utilisés pour illustrer les commandes spécifiques à chaque distribution :

	Commandes spécifiques à la distribution Linux Fedora Core.
	Commandes spécifiques à la distribution Debian/Linux

Méthode d'installation des logiciels

Pour installer les logiciels utilisés pour construire l'IDS nous utilisons les systèmes de paquetages propres à chaque distribution.

Pour Debian les paquetages seront installés à l'aide de la commande `apt-get`. Pour FC4 nous utiliserons les paquetages RPM et la commande `rpm`.

Les sondes

Chaque sonde est constituée d'une machine de type Intel équipée de deux cartes réseau :

- une interface activée en mode "stealth", c'est-à-dire sans adresse IP ;
- une interface activée sur un réseau dédié à l'IDS. Ce réseau permet la remontée des alertes vers une base de données unique ainsi que l'administration des sondes.

Le logiciel utilisé est Snort en version 2.4.4.

Interface en mode stealth

Une interface en mode stealth est une interface réseau active mais à laquelle aucune adresse IP n'a été attribuée.

Le trafic réseau peut être lu depuis cette interface mais elle ne peut pas en générer.

Sous Linux, pour activer une interface sans lui attribuer d'adresse IP, il faut utiliser la commande `ifconfig` de la manière suivante :

```
# ifconfig eth1 up
```



L'interface `eth1` est alors montée mais sans adresse IP :

```
eth1      Lien encap:Ethernet  HWaddr 00:0D:88:4C:97:27
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
```

Pour vérifier que le trafic est bien visible depuis cette interface, vous pouvez utiliser le logiciel `tcpdump` :

```
# tcpdump -nvx -i eth1
tcpdump: listening on eth1
18:33:33.850365 192.168.2.21.ssh > 192.168.2.100.1195: P
28486852:928486916(64) ack 2719771859 win 15456 (DF) [tos 0x10] (ttl 64,
d 26305, len 104)
                                4510 0068 66c1 4000 4006 4df5 c0a8 0215
                                c0a8 0264 0016 04ab 3757 95c4 a21c 6cd3
                                5018 3c60 5eff 0000 0fe3 ddb7 69d1 4ca6
                                9729 40f1 09ff d63e e801 b6ae 350b 0101
                                4e66 e0bb c776 1230 5c6a 2ce8 e0e3 00b8
```

Note : les commandes précédentes sont exécutées en tant que `root`.

	<p>Pour activer au démarrage de la machine cette configuration sous FC4, le fichier <code>/etc/sysconfig/network-scripts/ifcfg-eth1</code> ne doit contenir que les lignes suivantes :</p> <pre>DEVICE=eth1 BOOTPROTO=none ONBOOT=yes</pre>
	<p>Pour activer au démarrage de la machine cette configuration sous Debian, le fichier <code>/etc/network/interfaces</code> doit contenir les lignes suivantes :</p> <pre>DEVICE=eth1 BOOTPROTO=none ONBOOT=yes</pre>

Sécurité

La sécurité de la sonde s'appuie sur :

- une désactivation du routage IP au niveau du noyau ; dans le fichier `/etc/sysctl.conf`, la valeur du paramètre `net.ipv4.ip_forward` doit être 0. Si nécessaire, il est toujours possible de recompiler le noyau Linux en retirant la fonctionnalité `ip_forwarding`.
- Une règle NetFilter/IPTables qui interdit toute émission de paquet sur l'interface activée en mode stealth (`iptables -A OUTPUT -o eth1 -j DROP`).

Ces deux principes rendent la sonde invisible depuis les points d'écoute. Cela ne signifie cependant pas qu'elle est à l'abri d'une attaque en aveugle (cas où un attaquant, soupçonnant la présence d'une sonde, envoie du trafic volontairement malformé dans le but de leurrer la sonde ou de la rendre inopérante. Voir Cours IDS / Snort).

Le réseau dédié à l'IDS

Le réseau IDS (en adressage RFC1918) est utilisé pour administrer les sondes (SSH), remonter les alertes depuis les sondes vers le serveur SGBD et mettre à jour les règles.

Il doit être sécurisé de manière à ne pas compromettre la sécurité du S.I. en faisant office de passerelle à un éventuel attaquant. Dans le cas présent, nous avons choisi de dédier un réseau non interconnecté aux autres zones du S.I.. Il est cependant possible de relier le réseau IDS à une autre zone du S.I. en passant par un pare feu. La sécurité de cette interconnexion doit alors être soigneusement validée et surveillée.

Il est recommandé d'adopter la configuration décrite sur le schéma suivant pour ce réseau particulièrement sensible. Le pare feu permet d'isoler et de contrôler les flux entre chaque composant. La connexion de la console à un autre réseau peut alors être envisagée de manière plus sereine.

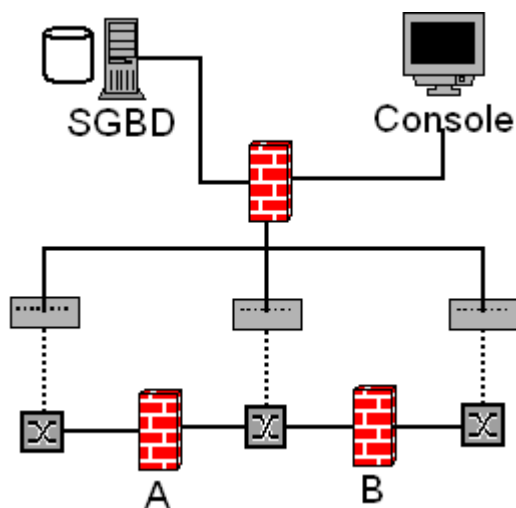


Figure 4 - Réseau IDS

Serveur SGBD

Comme nous utilisons trois sondes, il est judicieux de centraliser les alertes sur une seule machine. De même, l'utilisation d'une base de données présente de nombreux avantages et offre la possibilité de consulter les alertes depuis une interface dynamique et conviviale.

Dans notre cas, nous utilisons un serveur de type Intel sous Linux et un serveur de bases de données MySQL.

Console

La console est elle aussi une machine de type Intel sous Linux. Un serveur HTTP Apache avec l'interpréteur PHP est utilisé pour consulter le contenu de la base de données hébergée sur le serveur SGBD. On utilise l'interface BASE pour cela.

Administration

Sur chaque machine qui compose l'IDS, un démon SSH OpenSSH est utilisé pour les tâches d'administration courantes.

Schéma des flux

Le tableau suivant illustre les flux entre les composants.

Source	Port	Destination	Port	Protocole	Type
Sonde	>1024	SGBD	3306	TCP	MySQL Stockage des alertes
Console	>1024	SGBD	3306	TCP	Consultation de la base de données des alertes (Interface BASE)
Console	>1024	SGBD Sondes	22	TCP	Flux d'administration (SSH)
Sondes	>1024	Console	22	TCP	Transferts de fichiers, sauvegardes (SSH)



INSTALLATION DU SERVEUR SGBD

La première étape de la construction de notre IDS va consister à installer le serveur SGBD qui sera utilisé pour stocker les alertes.

Nous allons utiliser une base de données de type MySQL pour stocker les alertes générées par les sondes.

Le logiciel retenu est MySQL dans sa version 4.

Installation de MySQL

	Pour installer le serveur MySQL sous Fedora Core 4 il faut se procurer les paquetages suivants : Mysql 4.1.16 Mysql Server 4.1.16 Mysql Devel 4.1.16 Ces paquetages sont disponibles sur le site du projet Fedora. L'installation de la sonde sous Fedora Core se résume donc à celle de ces paquetages.
	Pour installer MySQL 4 sous Debian, nous allons utiliser la commande <code>apt-get</code> : <pre>apt-get install mysql-server-4.1 mysql-common \ mysql-client-4.1</pre>

Sécurisation

Par défaut, certains paramètres de MySQL sont inutiles ou trop permissifs. Il est donc nécessaire de les modifier.

Les actions suivantes doivent être entreprises avant la mise en production effective du serveur :

- Destruction des comptes et bases de test ;
- Renforcement du contrôle d'accès au compte root : cet utilisateur ne doit pouvoir se connecter à la base que depuis la machine en locale (IP source : 127.0.0.1). Pour les opérations d'administration à distance, on créera un autre compte (par exemple : myadmin) dont les droits seront plus élevés que ceux d'un utilisateur non privilégié mais plus restreints que ceux de root. Cet utilisateur pourra créer des bases, les administrer, mais ne pourra pas créer de nouveaux utilisateurs ni modifier les droits des utilisateurs existants ni intervenir sur les processus MySQL.

Configuration du serveur MySQL

MySQL utilise un système de gestion de comptes Utilisateurs pour définir les droits attribués sur les objets MySQL (c'est-à-dire : les bases de données, les tables dans les bases) et les commandes associées (création de base, de tables, utilisation des ordres SQL INSERT, UPDATE, DELETE, etc.).

Comme sous Unix, le super utilisateur MySQL s'appelle root et par défaut ce compte n'est pas sécurisé : il n'a en effet pas de mot de passe.

La première étape de la configuration du serveur MySQL va donc consister à attribuer un mot de passe à cet utilisateur root :

- Vérifier que le démon `mysqld` est bien lancé ou bien le démarrer ;
- Lancer la commande suivante depuis la ligne de commande :

```
# mysqladmin -u root password "mypassword"
```


mypassword doit bien entendu être remplacé par un mot de passe sûr et gardé secret.

- Vérifier qu'il est maintenant nécessaire d'utiliser ce mot de passe pour les connexions MySQL :
mysql -u root -p
Enter password : *****
mysql>
- Créer la base snortdb et les utilisateurs autorisés à s'y connecter :
mysql> CREATE DATABASE snortdb ;
mysql> GRANT CREATE, INSERT, SELECT, DELETE, UPDATE\
ON snortdb.* TO snorty IDENTIFIED BY 'snorty' ;
mysql> FLUSH PRIVILEGES ;

Le schéma de la base snortdb est fourni par le paquetage FC4 et le code source. Il est contenu dans le fichier create_mysql.

Il faut charger ce schéma dans la base snortdb pour la rendre opérationnelle :

```
# mysql -u snorty -p snortdb < create_mysql  
Enter password: *****
```

Pour vérifier que le schéma est bien présent :

```
# mysql -u snorty -p snortdb  
Enter password: *****
```

```
mysql> show tables;
```



Nous pouvons maintenant passer à l'étape suivante : installer les sondes.

INSTALLATION DES SONDES SNORT

Ce paragraphe décrit l'installation d'une sonde Snort. Les opérations décrites ci-dessous s'appliquent donc aux trois sondes de notre IDS.

Pré requis

Que le système d'exploitation utilisé sur les sondes soit Debian ou Fedora Core, il est recommandé qu'il ait été installé de manière minimale, c'est-à-dire avec le moins de paquetages et de logiciels possibles.

	<p>Pour installer l'OS Fedora Core de manière minimale, il faut au moment de l'installation choisir l'option <code>Minimal</code>. Les paquetages additionnels seront téléchargés depuis les sites des différents projets et installés à l'aide de la commande <code>rpm</code>.</p>
	<p>Pour l'OS Debian, on choisira l'installation <code>Net-Install</code>. Les paquetages additionnels seront ajoutés grâce à la commande <code>apt-get</code> si la machine est connectée à Internet. Dans le cas contraire, ils seront téléchargés et installés à l'aide de la commande <code>dpkg</code> avec l'option <code>-i</code> :</p> <pre>dpkg -i nom_du_paquetage.deb</pre>



Dans la suite de ce document, nous nous contenterons d'indiquer le nom et la version des logiciels à installer, ainsi que l'adresse des sites sur lesquels les paquetages sont disponibles pour chaque distribution.

Installation de Snort 2.4.4

Le logiciel Snort doit être installé sur chaque sonde. La version retenue est la 2.4.4, dernière version stable disponible en avril 2006.

Ce logiciel s'appuie sur les fonctionnalités de la bibliothèque `libpcap` pour capturer les paquets. Il faut donc installer cette bibliothèque avant d'installer Snort. La version de la `libpcap` à jour en avril 2006 est la 0.9.2.

De la même façon, Snort s'appuie sur les fonctionnalités de la bibliothèque `PCRE` (Perl Compatible Regular Expressions) pour rechercher des chaînes de caractères dans les paquets. La dernière version à jour en avril 2006 est la 6.4.

	<p>Libpcap 0.9.4 Pcre 6.3</p> <p>Ces paquetages sont disponibles sur le site du projet Fedora.</p> <p>Snort 2.4.4 : le paquetage est disponible sur le site du projet Snort.</p> <p>L'installation de la sonde sous Fedora Core se résume donc à celle de ces paquetages.</p>
	<p>Pour installer snort 2.4.4 sous Debian, nous allons utiliser la commande <code>apt-get</code> pour installer les deux bibliothèques <code>LibPcap</code> et <code>LibPCRE</code> et leurs dépendances :</p> <pre>apt-get install libnet1 libnet1-dev libpcre3 \ libpcre3-dev autoconf automake1.9 \ libpcap0.8 libpcap0.8-dev libmysqlclient14-dev gcc make</pre> <p>Comme il n'existe pas de paquetage pour la version 2.4.4, nous allons devoir la compiler à partir du code source de la manière suivante :</p> <p>1^{ère} étape : Compilation</p> <p>Note : l'option <code>-with-mysql</code> est requise pour que la binaire snort puisse utiliser une base de données MySQL pour stocker les alertes.</p> <pre># cd /usr/local/src</pre>

	<pre># wget http://www.snort.org/dl/current/snort-2.4.4.tar.gz # tar xvzf snort-2.4.4.tar.gz # cd snort-2.4.4 # ./configure --with-mysql # make # make install</pre> <p>2ème étape : Création des repertoires et de l'environnement de travail</p> <pre># mkdir /etc/snort # mkdir /var/log/snort # groupadd snort # useradd -g snort snort # chown snort:snort /var/log/snort # cd /etc/snort # mv /root/snortrules-snapshot-2.4.tar.gz . # tar xvzf snortrules-snapshot-2.4.tar.gz # cp /usr/local/src/snort-2.4.4/etc/*.conf* . # cp /usr/local/src/snort-2.4.4/etc/*.map .</pre>
--	--

Validation

Pour vérifier que le binaire snort fonctionne, lancez la commande suivante :

```
# snort -V
```

La sortie est la suivante :

```
o''_~ -*> Snort! <*-
      ~ Version 2.4.4 (Build 28)
      '''' By Martin Roesch & The Snort Team:
http://www.snort.org/team.html
      (C) Copyright 1998-2005 Sourcefire Inc., et al.
NOTE: Snort's default output has changed in version 2.4.1!
      The default logging mode is now PCAP, use "-K ascii" to activate
      the old default logging mode.
```

Utilisation de Snort comme Analyseur de trafic

Snort peut être utilisé pour lire le trafic réseau sur une interface de la manière suivante :

```
# snort -dev -i eth0
```

Cette commande et ces options affichent à l'écran le trafic qui transite sur le réseau auquel est connectée l'interface eth0 :

```
Running in packet dump mode
Initializing Network Interface eth0
      ==== Initializing Snort ====
Initializing Output Plugins!
Decoding Ethernet on interface eth0
      ==== Initialization Complete ====
o''_~ -*> Snort! <*-
      ~ Version 2.4.4 (Build 28)
      '''' By Martin Roesch & The Snort Team:
http://www.snort.org/team.html
      (C) Copyright 1998-2005 Sourcefire Inc., et al.
NOTE: Snort's default output has changed in version 2.4.1!
      The default logging mode is now PCAP, use "-K ascii" to activate
      the old default logging mode.

05/08-09:24:43.948350  0:C:29:4F:B5:ED  ->  0:12:79:60:79:F0  type:0x800
len:0x86
```

192.16.9.194:22 -> 192.16.9.122:4923 TCP TTL:64 TOS:0x10 ID:36821 IpLen:20 DgmLen:120 DF

AP Seq: 0xFE1A0AD3 Ack: 0x6F89665 Win: 0x3300 TcpLen: 20
8B 10 48 76 C8 D1 57 7C 5D B3 D8 C4 67 63 38 5C ..Hv..W|]...gc8\
ED 49 30 FE E1 1D 0F A2 0D 17 CC 48 FA F9 31 A8 .IO.....H..1.
EB A1 42 51 A0 06 02 B4 73 D3 5C EF 9A 72 5B 1E ..BQ....s.\.r[.
31 D4 CE 28 DD 93 E7 76 4F 7A 78 B2 BE 92 46 5C 1..(...vOzx...F\
AA D4 A5 9F E1 DF 84 54 78 A0 F7 7D F6 F1 B3 47Tx..}...G

=====

05/08-09:24:43.957692 0:12:79:60:79:F0 -> 0:C:29:4F:B5:ED type:0x800 len:0x3C

192.16.9.122:4923 -> 192.16.9.194:22 TCP TTL:128 TOS:0x0 ID:46203 IpLen:20 DgmLen:40 DF

A* Seq: 0x6F89665 Ack: 0xFE1A0B23 Win: 0xF8B0 TcpLen: 20

=====

[Control+C]

=====
Snort received 92 packets
Analyzed: 89(96.739%)
Dropped: 3(3.261%)
=====

Breakdown by protocol:

TCP: 8 (8.696%)
UDP: 0 (0.000%)
ICMP: 0 (0.000%)
ARP: 0 (0.000%)
EAPOL: 0 (0.000%)
IPv6: 0 (0.000%)
ETHLOOP: 0 (0.000%)
IPX: 0 (0.000%)
FRAG: 0 (0.000%)
OTHER: 0 (0.000%)
DISCARD: 0 (0.000%)
=====

Action Stats:

ALERTS: 0
LOGGED: 0
PASSED: 0
=====

Snort exiting

Il est également possible de capturer le trafic réseau et de le stocker sur disque dans un fichier au format Pcap :

snort -b -i eth0 -L snort.pcap

Le trafic est stocké dans le fichier préfixé snort.pcap et suffixé par un "timestamp" Unix dans le répertoire /var/log/snort :

ls /var/log/snort/
snort.pcap.1147077043

Snort peut alors être utilisé pour lire ce fichier et en afficher le contenu :

snort -dev -r /var/log/snort/ snort.pcap.1147077043

Utilisation de Snort comme NIDS

Nous voulons utiliser Snort comme sonde de détection Réseau.

Pour cela, il nous faut :

- éditer et modifier les paramètres du fichier de configuration snort.conf ;
- charger les règles Snort à jour et choisir les types de règles à utiliser ;
- lancer Snort en mode démon.

Configuration

Le fichier de configuration de Snort se trouve dans `/etc/snort` sous FC4 et Debian. Il s'agit d'un simple fichier texte qui obéit aux règles standard des fichiers de configuration du monde Unix :

- toute ligne qui commence par le caractère # est un commentaire
- les variables sont appelées en faisant précéder leur nom du caractère \$
- d'une manière générale les paramètres sont définis sur une seule ligne et il est possible, pour une meilleure lisibilité de passer à la ligne en insérant le caractère \.

Dans le fichier snort.conf, voici les paramètres les plus significatifs :

HOME_NET : ce paramètre désigne les adresses IP internes de nos réseaux, c'est-à-dire les adresses cibles des attaques.

Dans notre cas, ce paramètre prendra les valeurs suivantes :

Sonde A	Plage d'adresses publiques de notre réseau <code>var HOME_NET 192.168.1.0/24</code>
Sonde B	Plage d'adresses privées de la DMZ et du réseau interne <code>var HOME_NET [192.168.2.0/24, 192.168.3.0/24]</code>
Sonde C	Plage d'adresses privées du réseau interne. <code>var HOME_NET 192.168.3.0/24</code>

EXTERNAL_NET : il s'agit des adresses des réseaux dits « externes », c'est-à-dire les adresses des sources des attaques d'une manière générale.

Une pratique courante consiste à déclarer ce paramètre comme étant l'inverse du paramètre HOME_NET :

```
var EXTERNAL_NET !$HOME_NET
```

Dans notre cas, il prendra les valeurs suivantes :

Sonde A	Plage d'adresses publiques externes de notre réseau : <code>var EXTERNAL_NET !\$HOME_NET</code>
Sonde B	Plage d'adresses n'appartenant pas à la DMZ (par contre elles peuvent appartenir au réseau interne) : <code>var EXTERNAL_NET !192.168.2.0/24</code>
Sonde C	Plage d'adresses privées du réseau interne (car nous voulons sur cette sonde détecter les attaques en provenance du réseau interne à destination du réseau interne) : <code>var EXTERNAL_NET 192.168.3.0/24</code>

Les autres paramètres importants concernent le mode de stockage des alertes et les règles chargées au lancement du démon snort.

Nous utilisons plusieurs sondes mais nous stockerons les alertes sur une base de données centralisée. Cette base sera de type MySQL.

Par précaution, il est recommandé d'utiliser, en plus de cette base de données, les mode de stockage suivants sur chaque sonde :

- tcpdump : chaque paquet qui génère une alerte sera conservé sur disque sur chaque sonde. En effet, l'interface BASE que nous allons utiliser pour visualiser le contenu de la base de données MySQL ne permet pas d'afficher toutes les informations contenues dans les paquets capturés. Notamment les informations de niveau 2 (couche Ethernet) ne sont pas lisibles. Or dans certains cas (attaques depuis le même segment réseau avec usurpation de l'adresse IP de l'attaquant) les informations relatives à cette couche peuvent être très utiles : c'est le cas par exemple des adresses MAC des machines sources.
- CSV : ce mode stocke les alertes dans un fichier texte au format CSV (chaque alerte est stockée sur une ligne, chaque donnée qui compose l'alerte est séparée des autres par une virgule). Ce format permet l'injection facile dans une base de données. Il servira de « backup » en cas de problème sur la base de données (perte de la liaison entre la sonde et le serveur SGBD, corruption de la base, etc.).
- Syslog : chaque alerte donnera lieu à l'insertion d'une ligne dans les journaux du système.

Les paramètres suivants sont utilisés pour cela :

Sylog :

```
output alert_syslog: LOG_AUTH LOG_ALERT
```

TcpDump :

```
output log_tcpdump: tcpdump.log
```

CSV :

```
output alert_csv : /var/log/snort/alert.csv
```

Base MySQL :

```
output database: log, mysql, user=snort password=test \  
dbname=snortdb host=192.168.4.10 sensor_name=nom_de_la_sonde
```

Notes :

L'option sensor_name est importante. C'est cette valeur qui sera utilisée dans l'interface BASE pour indiquer la sonde sur laquelle une alerte a été générée.

La valeur des paramètres user, password, dbname et host sera identique sur chaque sonde. Dbnome est le nom de la base MySQL utilisée pour stocker les alertes, Host l'adresse IP sur serveur SGBD.

Les signatures

Snort utilise des signatures pour détecter les tentatives d'intrusion ou les anomalies dans le trafic réseau.

Ces signatures sont fournies par le projet Snort et le projet Bleeding-Edge.

Elles se présentent sous la forme de fichiers texte qui contiennent une signature par lignes. Elles sont généralement regroupées par types et sont stockées sur les sondes dans le répertoire `/etc/snort/rules`.

Dans le fichier de configuration `snort.conf`, les signatures sont activées (chargées au lancement du démon snort) en décommentant la ligne qui indique quels sont les fichiers de signatures à utiliser :

```
# include $RULE_PATH/info.rules  
# include $RULE_PATH/icmp-info.rules  
include $RULE_PATH/virus.rules
```

Dans l'exemple ci-dessus, les signatures du fichier `/etc/snort/rules/virus.rules` sont chargées, mais pas celles des fichiers `info.rules` ni `icmp-info.rules`.

La variable `RULE_PATH` désigne le répertoire où sont stockés les fichiers de signatures :

```
var RULE_PATH /etc/snort/rules
```

Lancement du démon Snort

Une fois le fichier de configuration snort.conf modifié selon les besoins, le démon Snort est lancé à l'aide du script /etc/init.d/snortd sous FC4 ou bien depuis la ligne de commande :

```
# /usr/local/bin/snort -Dq -u snort -g snort -c /etc/snort/snort.conf
```



INSTALLATION DE LA CONSOLE

La console est le dernier élément qu'il nous faut installer.



Elle s'appuie sur un serveur Apache et l'interpréteur PHP. Ces deux logiciels nous permettront d'installer l'interface BASE pour visualiser le contenu de la base de données dans laquelle sont stockées les alertes.

Installation du serveur Apache

Nous allons utiliser un serveur Apache en version 2 ainsi que l'extension SSL qui nous permettra de sécuriser les consultations.

	<p>Pour installer le serveur Apache sous Fedora Core 4 il faut se procurer les paquetages suivants :</p> <p>Httpd 2.0.54 Httpd Devel 2.0.54 Mod_ssl 2.0.54</p> <p>Ces paquetages sont disponibles sur le site du projet Fedora.</p> <p>L'installation de la sonde sous Fedora Core se résume donc à celle de ces paquetages.</p>
	<p>Pour installer Apache sous Debian, nous allons utiliser la commande <code>apt-get</code> :</p> <pre>apt-get install apache-ssl apache-common libssl-dev</pre>

Installation de l'interpréteur PHP

	<p>Pour installer PHP sous Fedora Core 4 il faut se procurer les paquetages suivants :</p> <p>PHP 5.0.4 PHP-devel 5.0.4 PHP-gd 5.0.4 PHP-MySQL 5.0.4 PHP-Pear 5.0.4</p> <p>Ces paquetages sont disponibles sur le site du projet Fedora.</p> <p>L'installation de la sonde sous Fedora Core se résume donc à celle de ces paquetages.</p>
	<p>Pour installer PHP sous Debian, nous allons utiliser la commande <code>apt-get</code> :</p> <pre>apt-get install libapache-mod-php4 php4-mysql php4-gd \ php4-pear libphp-adodb</pre>

Installation de l'interface BASE

BASE est une interface pour Snort écrite en PHP. La dernière version à jour en avril 2006 est la 1.2.4.

Note : dans la suite de ce paragraphe, on partira du principe que le DocumentRoot du serveur Apache est le répertoire /var/www.

Pré requis

BASE utilise deux bibliothèques : AdoDB pour ses fonctions de connexion et d'interrogation à des bases de données de plusieurs types, dont MySQL, et GD pour ses fonctions graphiques. L'installation de ces deux bibliothèques peut s'effectuer à partir des paquetages ou du code source, sachant que ce dernier est une simple archive Tar qu'il faut décompresser dans le DocumentRoot du serveur Apache.

Installation de BASE

L'interface BASE est fournie elle-aussi sous forme d'une archive Tar. L'installation de BASE consiste à décompresser cette archive et l'extraire dans le DocumentRoot du serveur Apache.

Une fois extraite, il est recommandé de créer un lien symbolique qui pointe sur le répertoire base-1.2.4 :

```
# ln -s base-1.2.4 base
```

Configuration de BASE

La configuration de l'interface BASE se fait depuis un navigateur :

- vérifier que le serveur Apache est bien lancé sur le serveur où BASE a été installée, le démarrer le cas échéant ;
- depuis un navigateur Web, ouvrir l'URL `http://adresse_du_serveur/base`

Il suffit alors de suivre les écrans et les instructions qu'ils contiennent :

Basic Analysis and Security Engine (BASE) Setup Program

The following pages will prompt you for set up information to finish the install of BASE.
If any of the options below are red, there will be a description of what you need to do below the chart.

Settings	
Config Writeable:	Yes
PHP Version:	4.3.10-16

[Continue](#)

Cliquer sur Continue.

Note : Si le champ Config Writeable est à No, il faut modifier les attributs des fichiers du répertoire /var/www/base :

```
$ chown -R apache:apache /var/www/base
```

Choisir à l'étape suivante la langue retenue et renseigner le chemin d'accès aux librairies PHP AdoDB :

Basic Analysis and Security Engine (BASE) Setup Program

Step 1 of 5	
Pick a Language:	english ▾ [?]
Path to ADOdb:	/var/www/adodbl [?]
<input type="button" value="Envoyer"/>	

A l'étape suivante, renseigner le nom de la base de données utilisée pour stocker les alertes générées par les sondes (Champ Database Name), le nom ou l'adresse IP du serveur qui héberge le serveur MySQL (Champ Database Host), le nom d'utilisateur MySQL et le mot de passe pour accéder à la base (Champs Database User Name et Database Password) :

Basic Analysis and Security Engine (BASE) Setup Program

Step 2 of 5	
Pick a Database type:	MySQL [?]
Database Name:	snortdb
Database Host:	localhost
Database Port: Leave blank for default!	
Database User Name:	snorty
Database Password:	
<input type="checkbox"/> Use Archive Database[?]	
Archive Database Name:	
Archive Database Host:	
Archive Database Port: Leave blank for default!	
Archive Database User Name:	
Archive Database Password:	
<input type="button" value="Envoyer"/>	

Cocher ensuite la case Use Authentication System et renseigner les champs Admin User Name, Password et Full Name :

Basic Analysis and Security Engine (BASE) Setup Program

Step 3 of 5	
<input type="checkbox"/> Use Authentication System [?]	
Admin User Name:	
Password:	
Full Name:	
<input type="button" value="Envoyer"/>	

Basic Analysis and Security Engine (BASE) Setup Program

Step 3 of 5	
<input checked="" type="checkbox"/> Use Authentication System [?]	
Admin User Name:	basemaster
Password:	thisisbase
Full Name:	BASE Administrator
<input type="button" value="Envoyer"/>	

Basic Analysis and Security Engine (BASE) Setup Program

Step 4 of 5		
Operation	Description	Status
BASE tables	Adds tables to extend the Snort DB to support the BASE functionality	Create BASE AG
Search Indexes	(Optional) Adds indexes to the Snort DB to optimize the speed of the queries	DONE

Basic Analysis and Security Engine (BASE) Setup Program

Successfully created 'acid_ag'
 Successfully created 'acid_ag_alert'
 Successfully created 'acid_ip_cache'
 Successfully created 'acid_event'
 Successfully created 'base_roles'
 Successfully INSERTED Admin role
 Successfully INSERTED Authenticated User role
 Successfully INSERTED Anonymous User role
 Successfully INSERTED Alert Group Editor role
 Successfully created 'base_users'
 Successfully created user.

Step 4 of 5		
Operation	Description	Status
BASE tables	Adds tables to extend the Snort DB to support the BASE functionality	DONE
Search Indexes	(Optional) Adds indexes to the Snort DB to optimize the speed of the queries	DONE

The underlying Alert DB is configured for usage with BASE.

Additional DB permissions

In order to support Alert purging (the selective ability to permanently delete alerts from the database) and DNS/whois lookup caching, the DB user "snorty" must have the DELETE and UPDATE privilege on the database "snortdb@localhost"

Now continue to [step 5...](#)

Basic Analysis and Security Engine (BASE)

Login:

Password:

Envoyer

BASE 1.2.1 (kris) (by Kevin Johnson and the BASE Project Team
 Built on ACID by Roman Danyliw)

Utilisation de l'interface BASE

L'interface BASE se présente de la manière suivante :

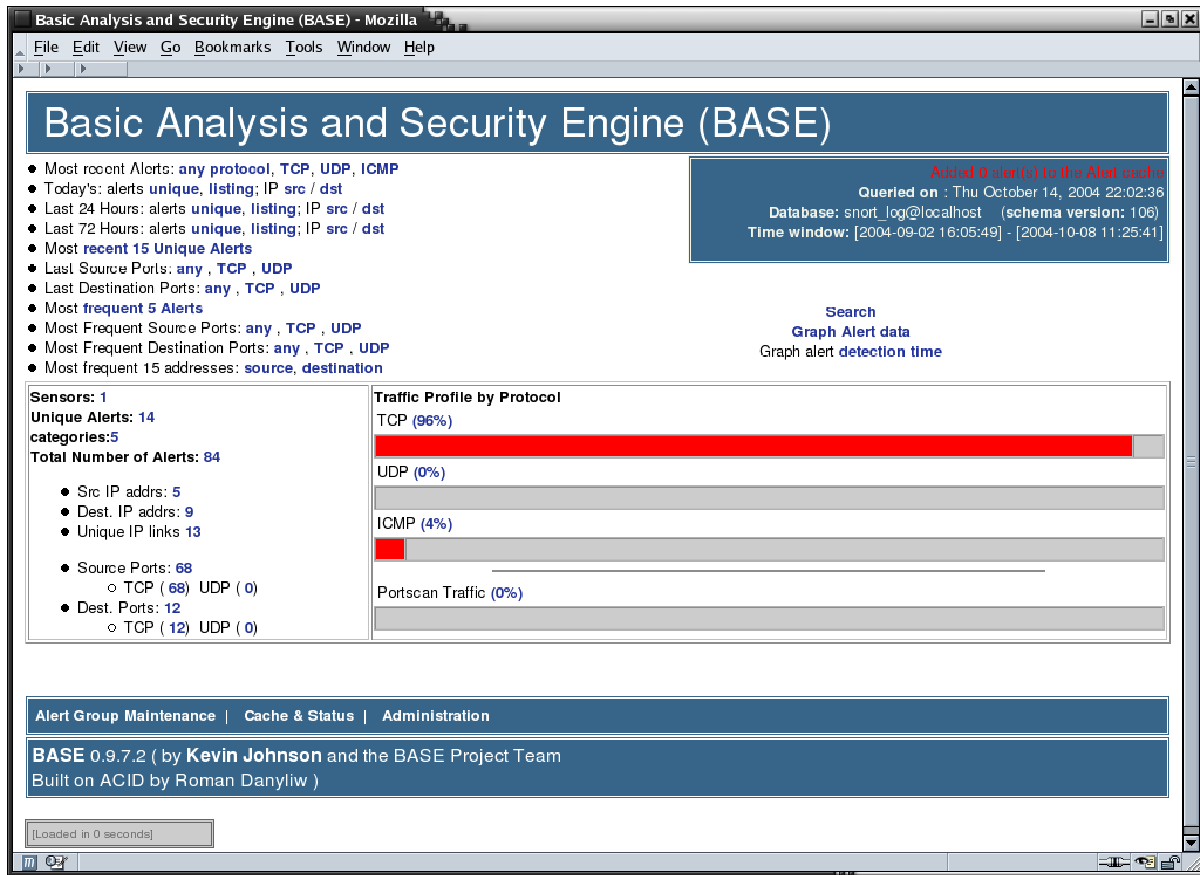


Figure 5 - BASE - Page d'accueil

The screenshot shows the 'Basic Analysis and Security Engine (BASE): Alert - Mozilla' window. The alert details are as follows:

ID #	Time	Triggered Signature
1 - 84	2004-10-08 11:25:41	[snort] NETBIOS SMB IPC\$ share unicode access

Meta

Sensor	name	interface	filter
	192.168.1.4	eth0	none

Alert Group: none

IP

source addr	dest addr	Ver	Hdr Len	TOS	length	ID	flags	offset	TTL	chksum
192.168.1.100	192.168.1.4	4	5	0	122	14356	0	0	128	16049

FQDN

Source Name	Dest. Name
Unable to resolve address	kevinanddenise.homelinux.com

Options: none

TCP

source port	dest port	R	R	U	A	P	R	S	F	seq #	ack	offset	res	window	urp	chksum
		1	0	G	K	H	T	N	N							
1613	139			X	X					990153569	3328611299	5	0	63669	0	32195

Options: none

Payload

length = 82

```

000 : 00 00 00 4E FF 53 4D 42 75 00 00 00 00 18 07 C8 ...N.SMBu.....
010 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF FE .....
020 : 64 00 C0 00 04 FF 00 4E 00 08 00 01 00 23 00 00 d.....N.....#..
030 : 5C 00 5C 00 4C 00 4F 00 52 00 49 00 45 00 4E 00 \.\.L.O.R.I.E.N.
040 : 5C 00 49 00 50 00 43 00 24 00 00 00 3F 3F 3F 3F \.I.P.C.$...???
```

Figure 6 - BASE - Visualisation d'une alerte

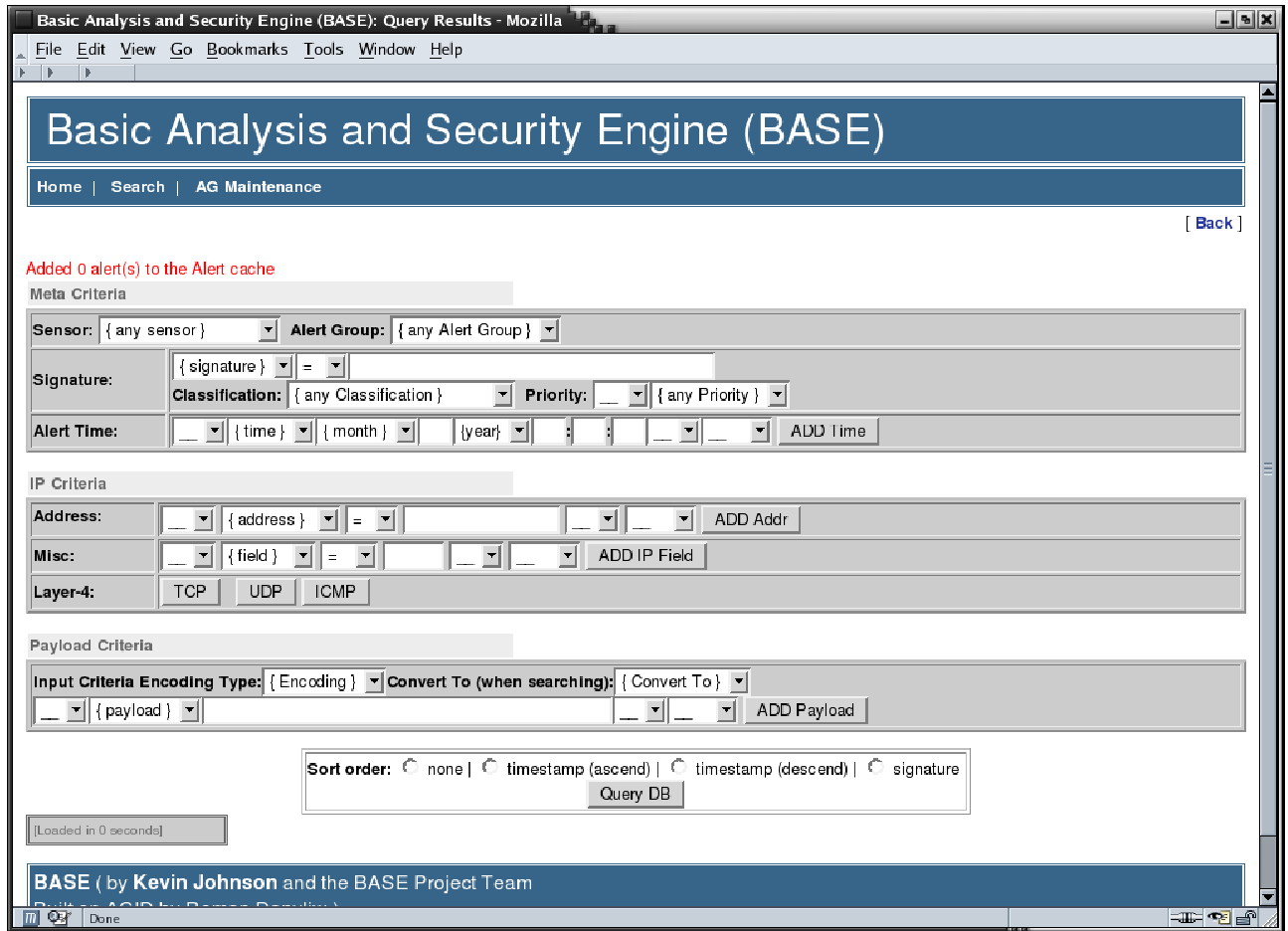


Figure 7 - BASE - Moteur de recherche

Basic Analysis and Security Engine (BASE): Query Results - Mozilla

File Edit View Go Bookmarks Tools Window Help

Basic Analysis and Security Engine (BASE)

Home | Search | AG Maintenance [Back]

Added 0 alert(s) to the Alert cache

Queried DB on : Thu October 14, 2004 22:04:44

Meta Criteria	any
IP Criteria	any
TCP Criteria	any
Payload Criteria	any

Summary Statistics

- Sensors
- Unique Alerts (classifications)
- Unique addresses: source | destination
- Unique IP links
- Source Port: TCP | UDP
- Destination Port: TCP | UDP
- Time profile of alerts

Displaying alerts 1-50 of 81 total

<input type="checkbox"/>	ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
<input type="checkbox"/>	#0-(1-84)	[snort] NETBIOS SMB IPC\$ share unicode access	2004-10-08 11:25:41	192.168.1.100:1613	192.168.1.4:139	TCP
<input type="checkbox"/>	#1-(1-83)	[snort] NETBIOS SMB IPC\$ share unicode access	2004-10-08 11:25:31	192.168.1.100:1608	192.168.1.4:139	TCP
<input type="checkbox"/>	#2-(1-82)	[snort] NETBIOS SMB IPC\$ share unicode access	2004-10-08 11:25:05	192.168.1.100:1601	192.168.1.4:139	TCP
<input type="checkbox"/>	#3-(1-80)	[snort] (http_inspect) OVERSIZE CHUNK ENCODING	2004-10-04 22:25:41	192.168.1.4:42164	67.19.245.228:80	TCP
<input type="checkbox"/>	#4-(1-81)	[snort] (http_inspect) OVERSIZE CHUNK ENCODING	2004-10-04 22:25:41	192.168.1.4:42163	67.19.245.228:80	TCP

Figure 8 - BASE - Liste d'alertes