

# Supervision des intrusions

ENCADRÉ PAR: MR RAISS

REALISÉ PAR: ABDELKARIM BARRANE

ANNÉE UNIVERSITAIRE: 2018/2019

# Plan de la présentation:

1. Introduction
2. Les systèmes de détection des intrusions IDS
3. Implémentation de la Solution SNORT

# Introduction

**L'internet** a facilité le flux **d'informations**, personnelles, financières et autres. En même temps, il a également promu autant de dangers. Les utilisateurs malveillants recherchent des proies vulnérables comme les systèmes sans correctifs. **Des alarmes** sont nécessaires pour prévenir les administrateurs et les membres de l'équipe de sécurité qu'une effraction s'est produite afin qu'ils puissent répondre en temps réel au danger.

**Les systèmes de détection d'intrusions** ont été conçus pour jouer le rôle d'un tel système d'alarme.

# La Détection des intrusions:

La détection des intrusions est le processus de **surveillance** des événements qui se trouvent dans un système des ordinateurs ou du réseau et les analysant pour **déetecter** les signes des intrusions, défini comme des tentatives pour compromettre la confidentialité, l'intégrité, la disponibilité ou éviter des mécanismes de sécurité de l'ordinateur ou du réseau.

## Les Intrusions:

Une intrusion est définie comme une **faute malveillante** d'origine interne ou externe résultant d'une **attaque** qui a réussi à exploiter une **vulnérabilité** dans le système de sécurité. Il vise par exemple à accorder ou à gagner des priviléges supplémentaires dont ils n'ont pas été autorisés.

# Les systèmes de Détection des intrusions IDS

# Les IDS:

Les IDS(Intrusion Detection Systems) sont des outils permettant de **repérer** tout type de trafic malveillant(les attaques ou intrusions, débits trop important, trafic suspect...) du réseau ou système sur lequel il est placé. Ils peuvent également **surveiller** l'activité du réseau ou du système, **analyser** les configurations du système ou du réseau contre toute vulnérabilité, analyser **l'intégrité** de données et bien plus.

# Les IDS:

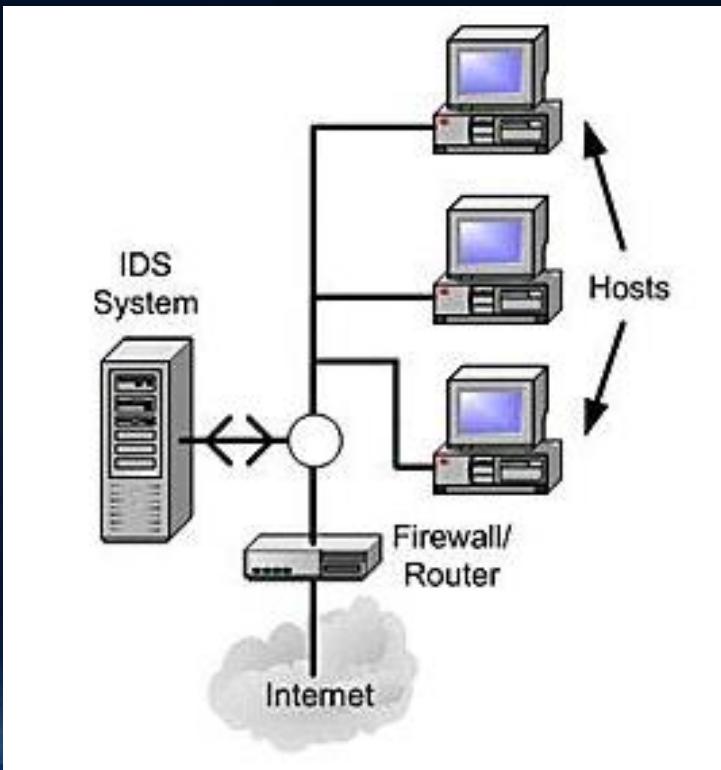
Il existe différents genres d'IDS que l'on classe comme suit:

# Les IDS:

Les **NIDS**(Network IDS) : Les IDS réseaux surveillent l'état de la sécurité du réseau, il **analyse** en temps réel le **trafic**(paquets circulant dans le réseau) qu'il aspire. Ensuite il est décortiqué et analysé et génère une **alerte** si des paquets semblent dangereux et ordonne des actions de **blocage** d'un flux.

# Les NIDS:

Les **NIDS**(Network IDS) en terme d'architecture :



**Se situe sur un réseau isolé**

**Ne voit qu'une copie du trafic du réseau à surveiller**

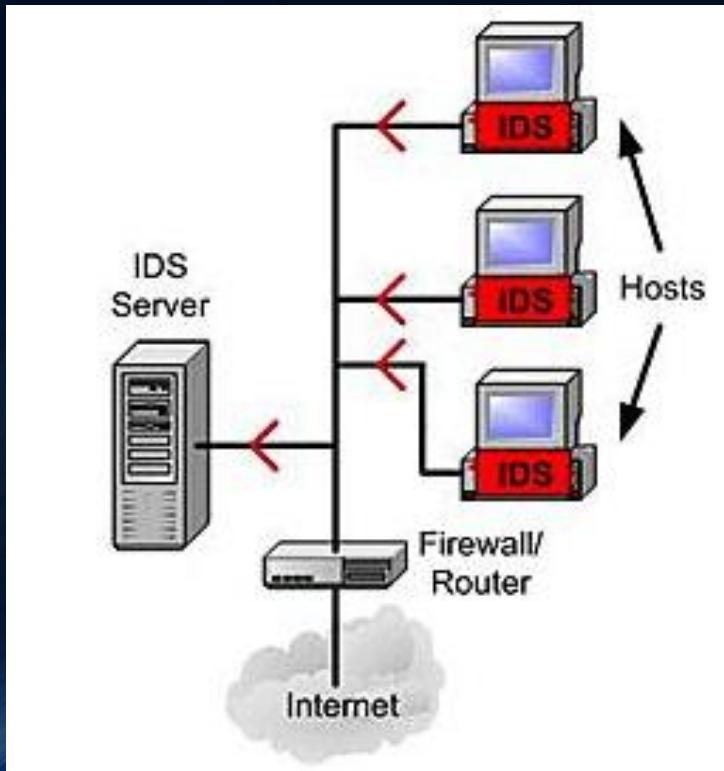
**Ne communique pas avec le réseau surveillé**

# Les IDS:

Les **HIDS**(Host IDS) : Les IDS systèmes surveillent l'état de sécurité des **hôtes** selon différents **critères**: il surveille l'activité de la **machine**(nombre et listes des processus, nombre d'utilisateurs, ressources consommées...), l'activité des **utilisateurs** de la machine(horaires et durée des connexions, commandes utilisées, programmes activés...) et l'activité potentiel lié au **programmes suspect**(virus , cheval de Troie, vers...). L'**intégrité** du système est alors vérifiée périodiquement et des alertes sont élevés.

# Les IDS:

Les **HIDS**(Host IDS) en terme d'architecture :



Récupère les informations remontés par les machines hôtes.

Analyse ses informations.

Surveille l'état de sécurité des machines hôtes.

# Les IDS:

Les **IDS Hybrides** : Ces IDS rassemblent les caractéristiques des NIDS et HIDS. Ils permettent, en un seul outil, de surveiller le **réseau** et les **terminaux**. Ils fonctionnent selon leurs **emplacement**. Ainsi l'architecture des IDS est distribué, où chaque composant unifie son format d'envoie, cela permet de communiquer et extraire de alertes plus pertinentes.

# Les Méthodes de détections des IDS:

## Les IDS à signature

Ils s'appuie sur un modèle constitué des **sections interdites** dans le système d'information, ce modèle s'appuie sur la connaissance des **techniques** employés par les attaquants: on tire des scénarios d'attaque et on recherche dans les traces d'audit leur éventuelle survenue.

# Les Méthodes de détections des IDS:

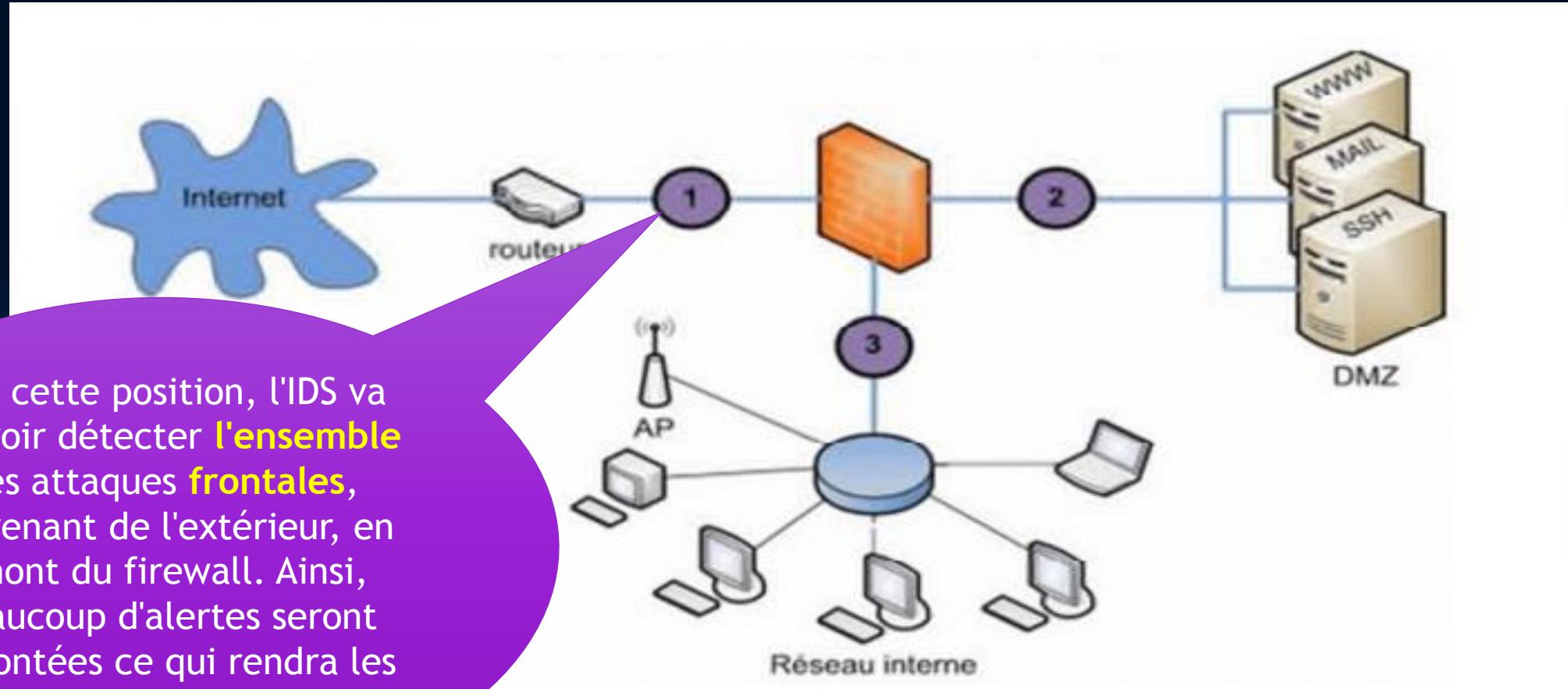
## Les IDS comportementaux

Ils s'appuient sur l'hypothèse selon lequel nous pouvons définir un **comportement normal** de l'utilisateur et que toute déviation par rapport à celui-ci est potentiellement **suspect**. Donc leur déploiement nécessite une phase d'apprentissage pendant laquelle l'outil va **apprendre** le comportement "normal" des flux applicatifs présents sur son réseau ou système.

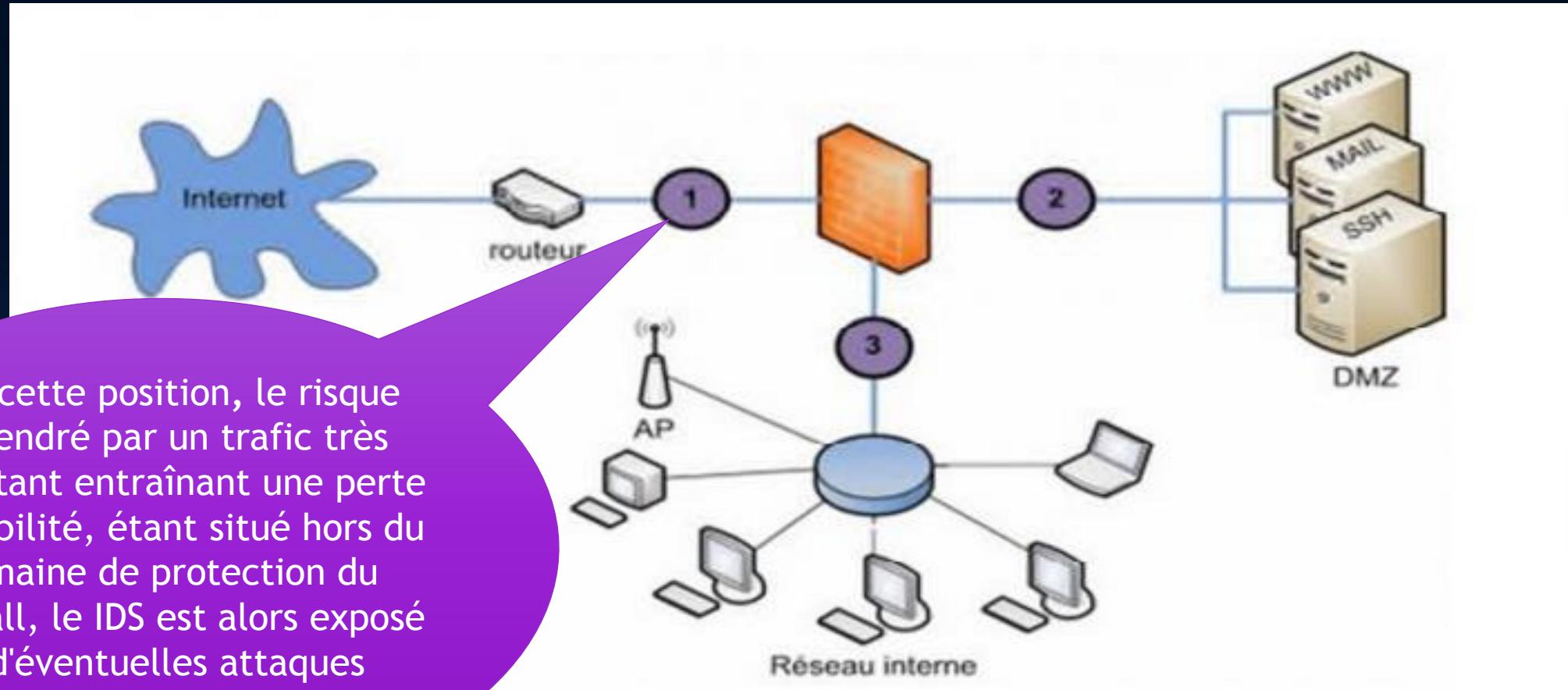
# Où positionner son IDS ?:

L'emplacement physique de la sonde du IDS sur le réseau a un impact considérable sur son efficacité. Dans le cas d'une architecture classique, composée d'un **Firewall** et d'une **DMZ**, trois positions sont généralement envisageables.

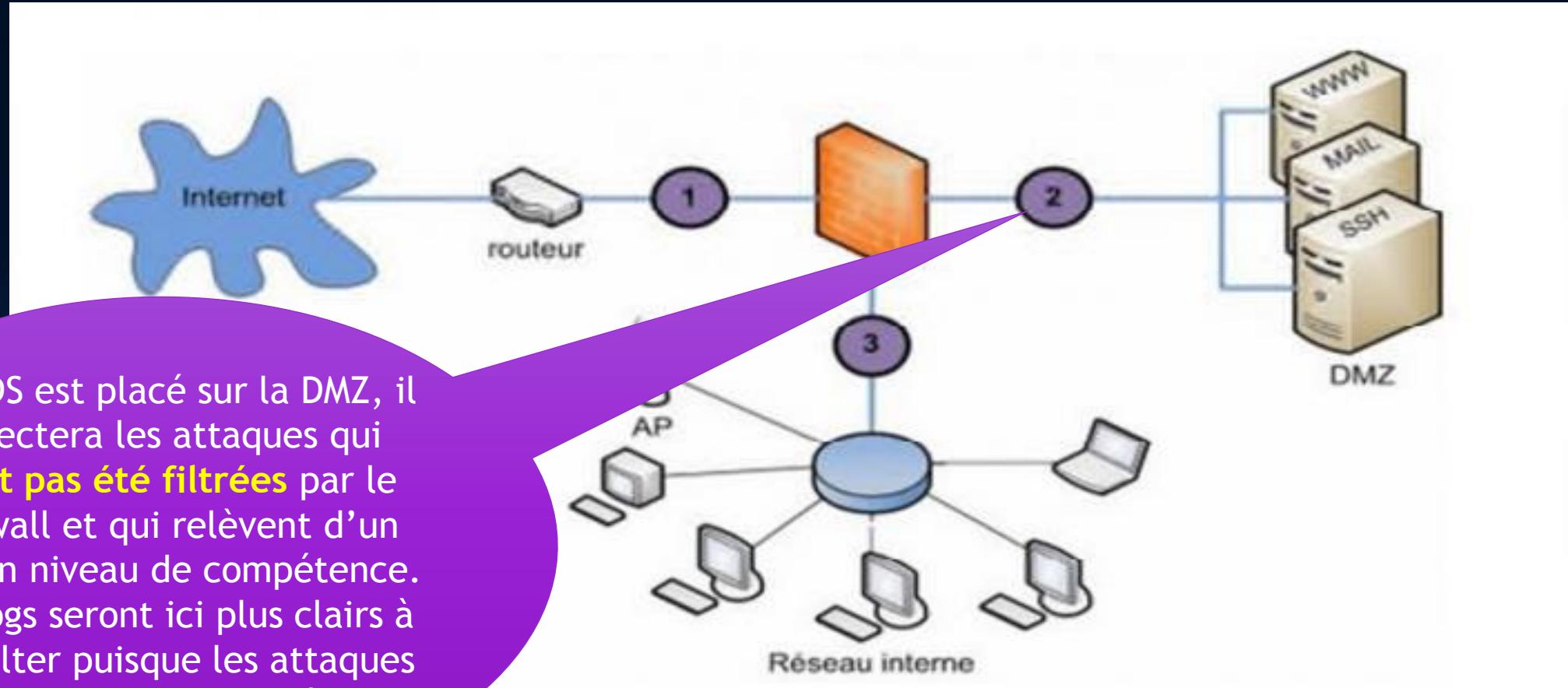
# Où positionner son IDS ?:



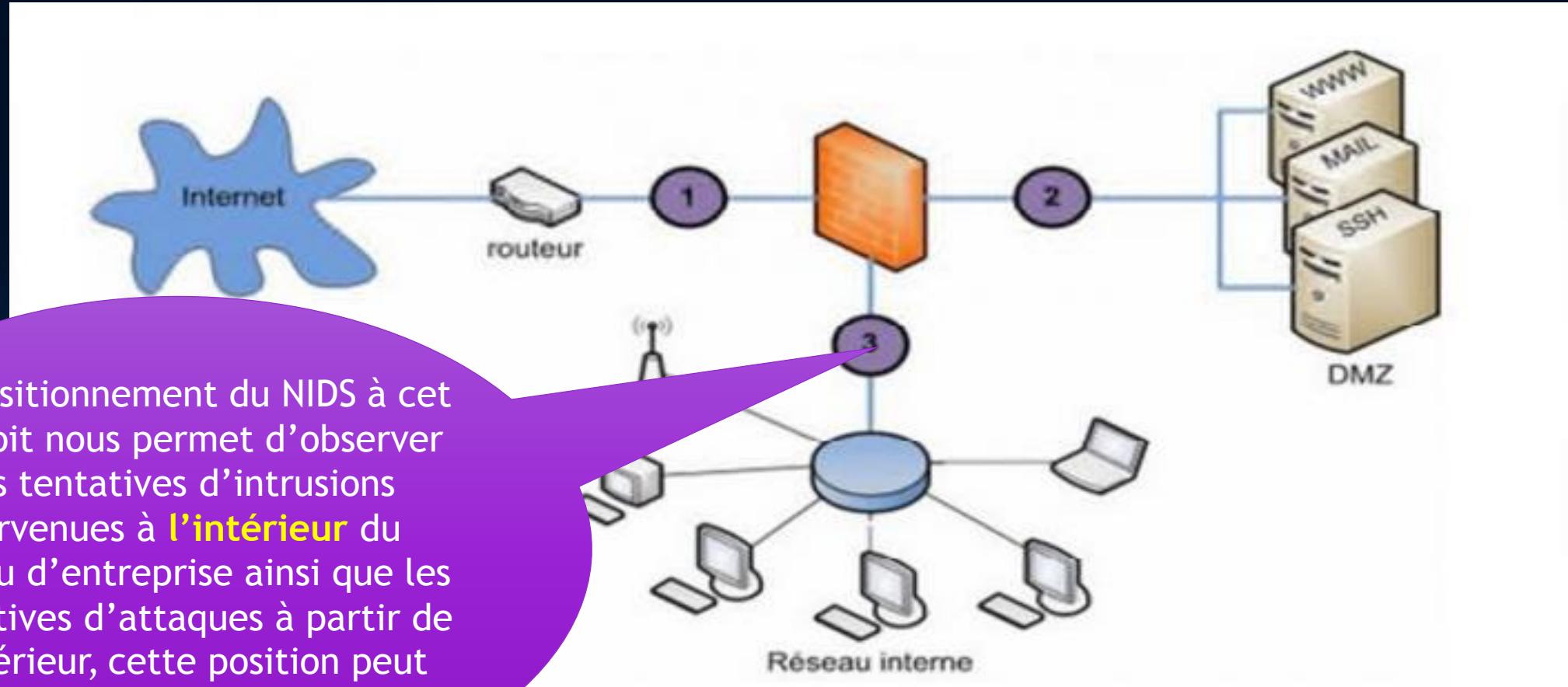
# Où positionner son IDS ?:



# Où positionner son IDS ?:



# Où positionner son IDS ?:



le positionnement du NIDS à cet endroit nous permet d'observer les tentatives d'intrusions parvenues à **l'intérieur** du réseau d'entreprise ainsi que les tentatives d'attaques à partir de l'intérieur, cette position peut revêtir un intérêt primordial.

## Des outils IDS :

Il existe plusieurs outils IDS parmi eux :

**ISS RealSecure**

**Enterasys DRAGON**

**SNORT**

## Des limites d'un IDS :

Parmi les faiblesses des IDS on trouve :

- 1)** Nombreux faux positifs(alerte en l'absence d'attaque).
- 2)** Configuration complexe et longue.
- 3)** Les attaques applicatives sont difficilement détectables.
- 4)** Attaques contre l'IDS lui-même.
- 5)** Ils ne peuvent pas compenser les trous de sécurité dans les protocoles réseaux.
- 6)** Pollution de l'IDS(Perte de paquet, attaques qui passent inaperçues, consommation des ressources..).

# Implémentation de la Solution SNORT

# SNORT:

SNORT est un logiciel **open source** écrit par Martin Roesch, disponible sous licence GNU. SNORT a la capacité d'effectuer **l'analyse du trafic** en temps réel et effectuer l'analyse de protocole, et la recherche de contenu. Il utilise une **sonde** pour détecter les attaques, le débordement système, les scans de port etc.



# SNORT:

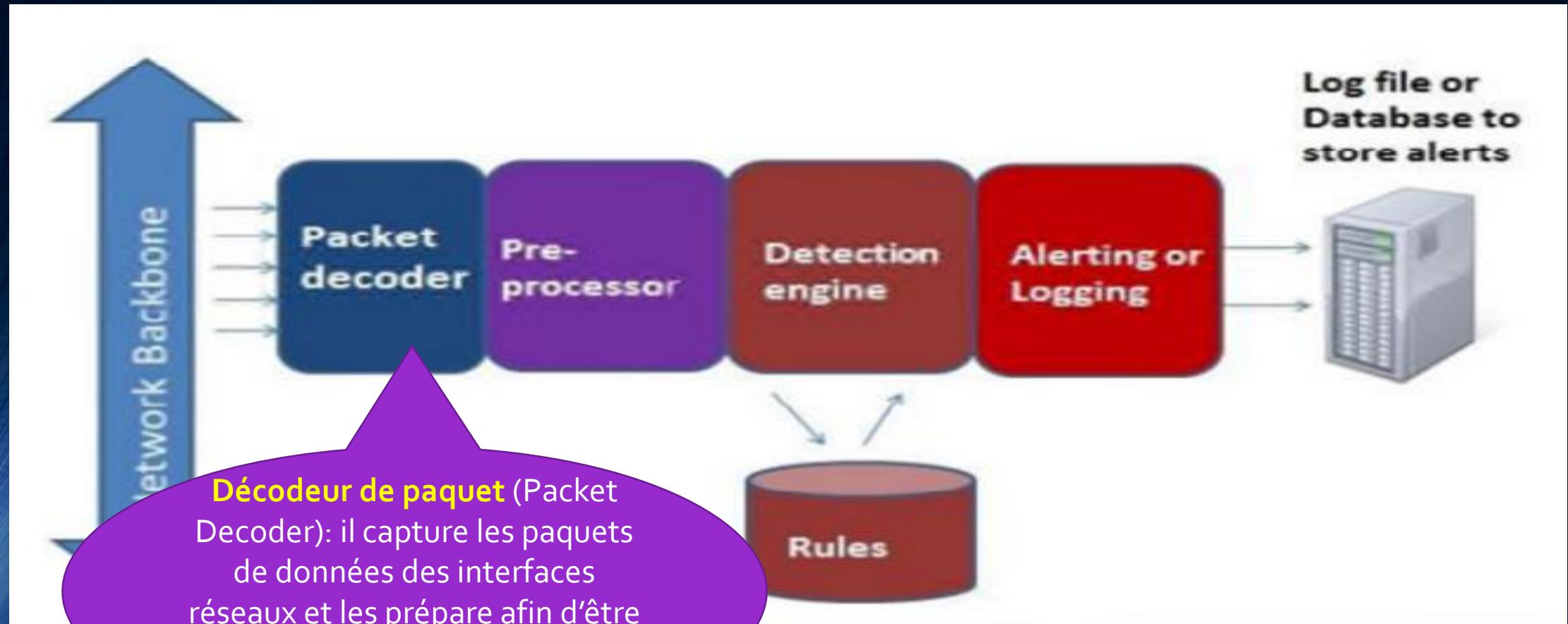
SNORT peut être configuré pour fonctionner en trois modes:

**Le mode sniffer** : dans ce mode, SNORT lit les paquets circulant sur le réseau et les affiche d'une façon continue sur l'écran.

**Le mode « packet logger »** : dans ce mode SNORT journalise le trafic réseau dans des répertoires sur le disque.

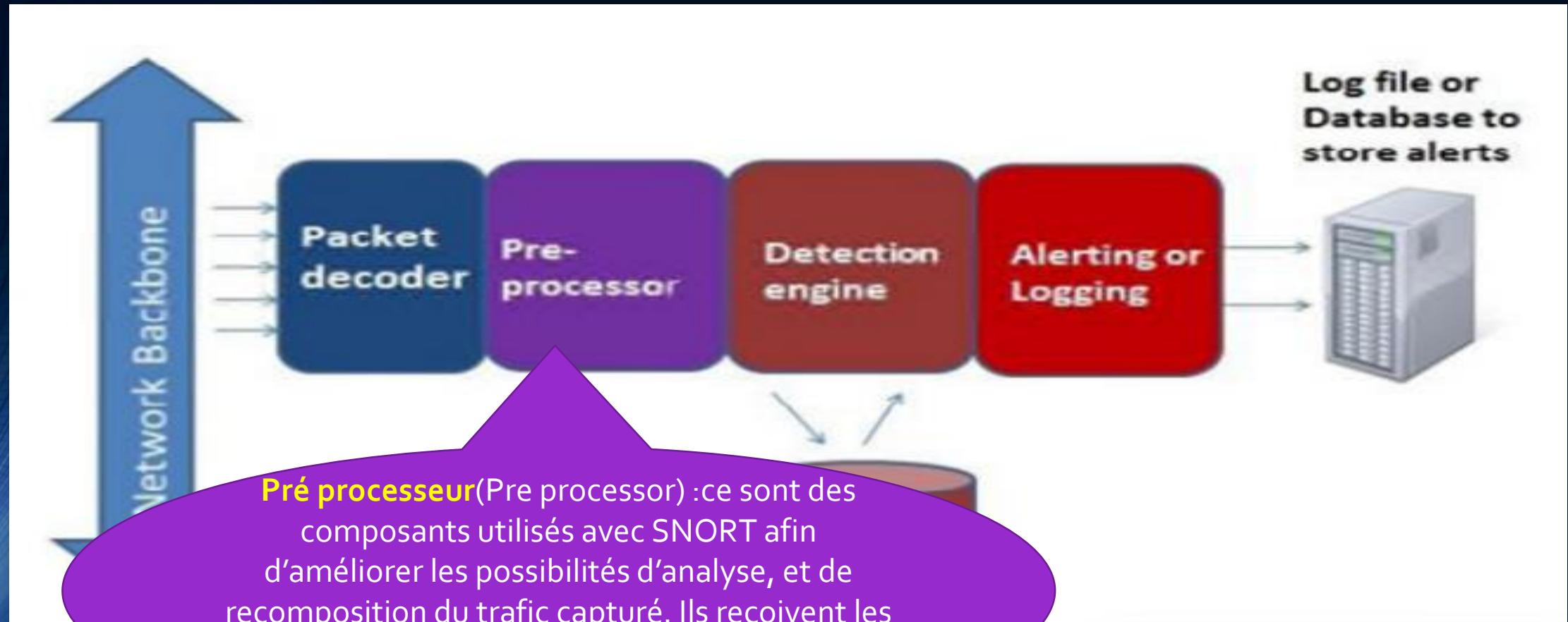
**Le mode détecteur d'intrusions réseau (NIDS)** : dans ce mode, SNORT analyse le trafic du réseau, compare ce trafic à des règles déjà définies par l'utilisateur et établi des actions à exécuter.

# Architecture de SNORT:



**Décodeur de paquet** (Packet Decoder): il capture les paquets de données des interfaces réseaux et les prépare afin d'être prétraitées ou envoyées au moteur de détection.

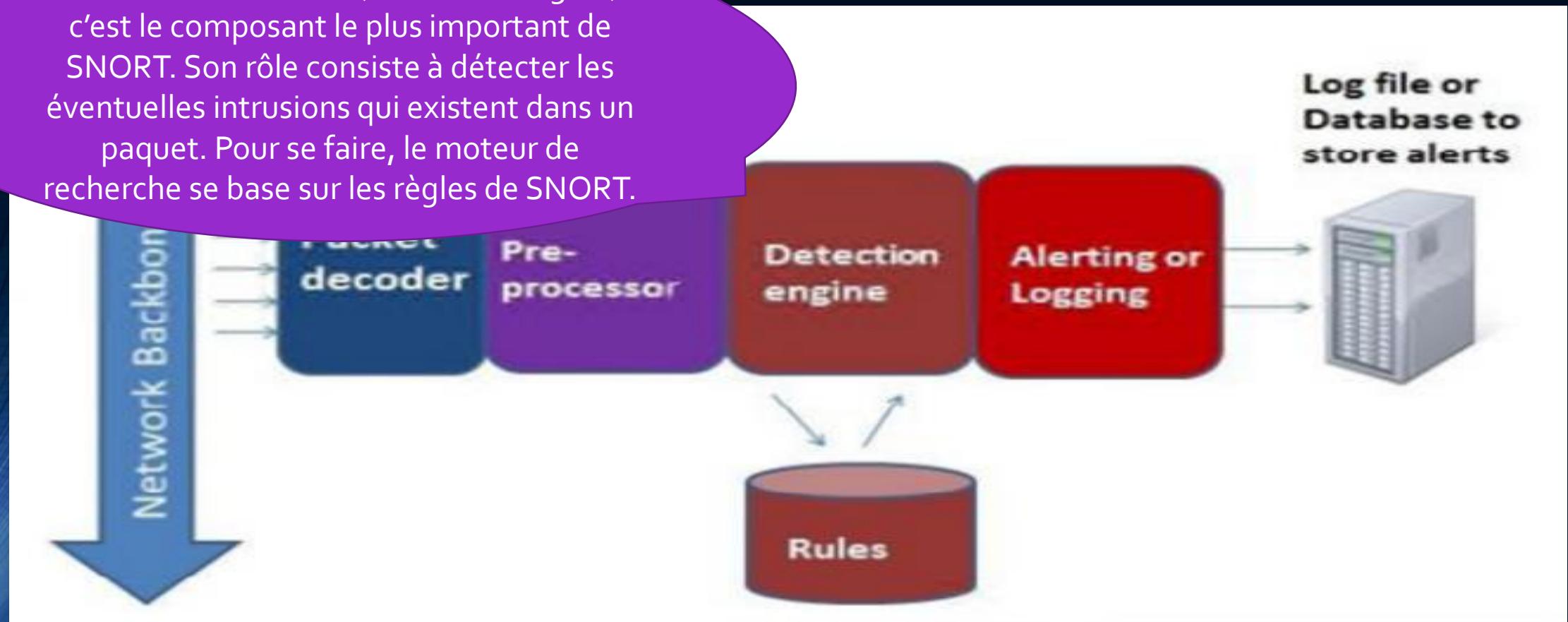
# Architecture de SNORT:



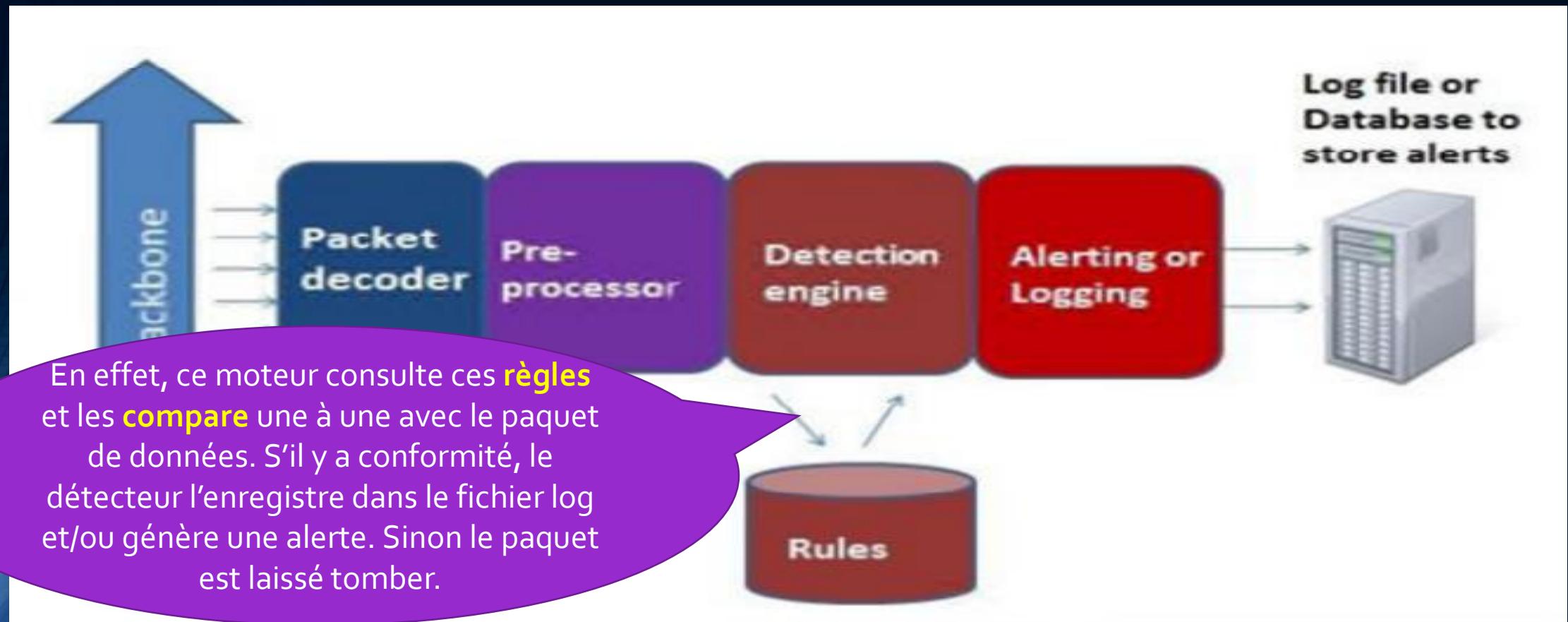
# Architecture de SNORT:

## Moteur de détection (Detection Engine):

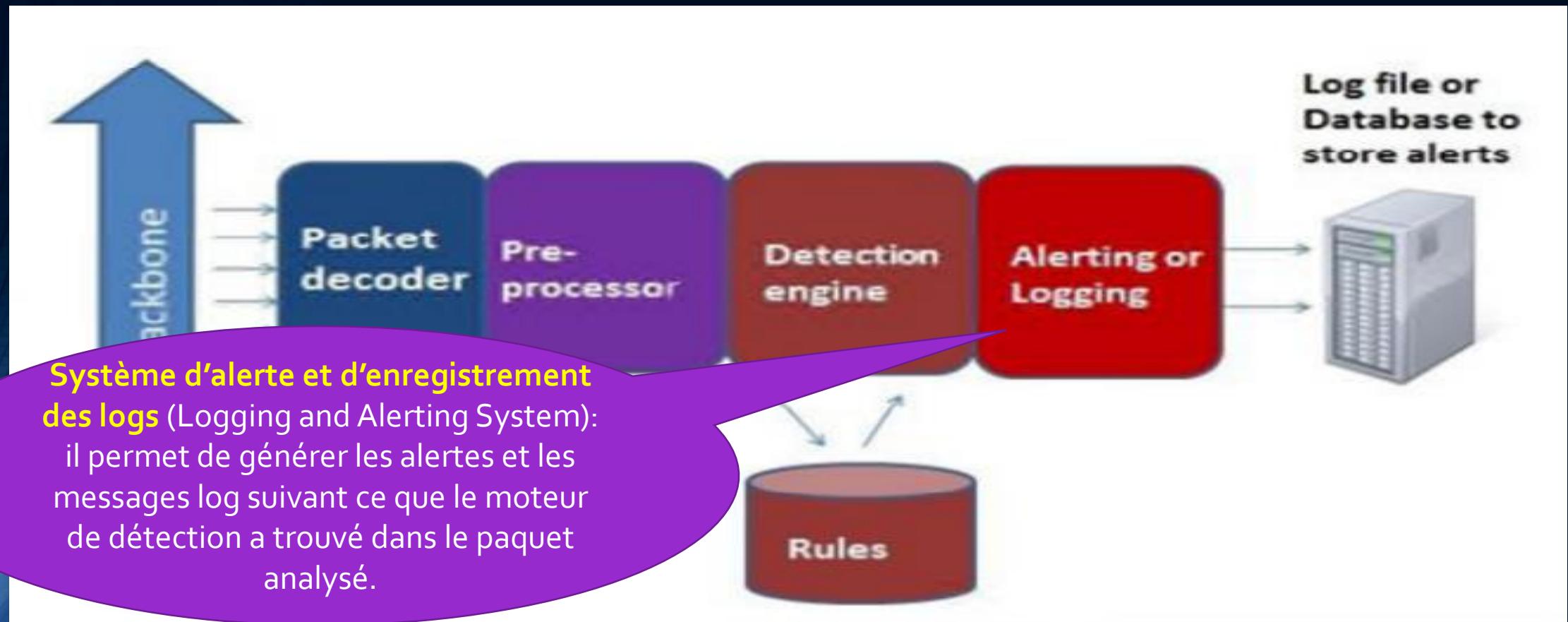
c'est le composant le plus important de SNORT. Son rôle consiste à détecter les éventuelles intrusions qui existent dans un paquet. Pour se faire, le moteur de recherche se base sur les règles de SNORT.



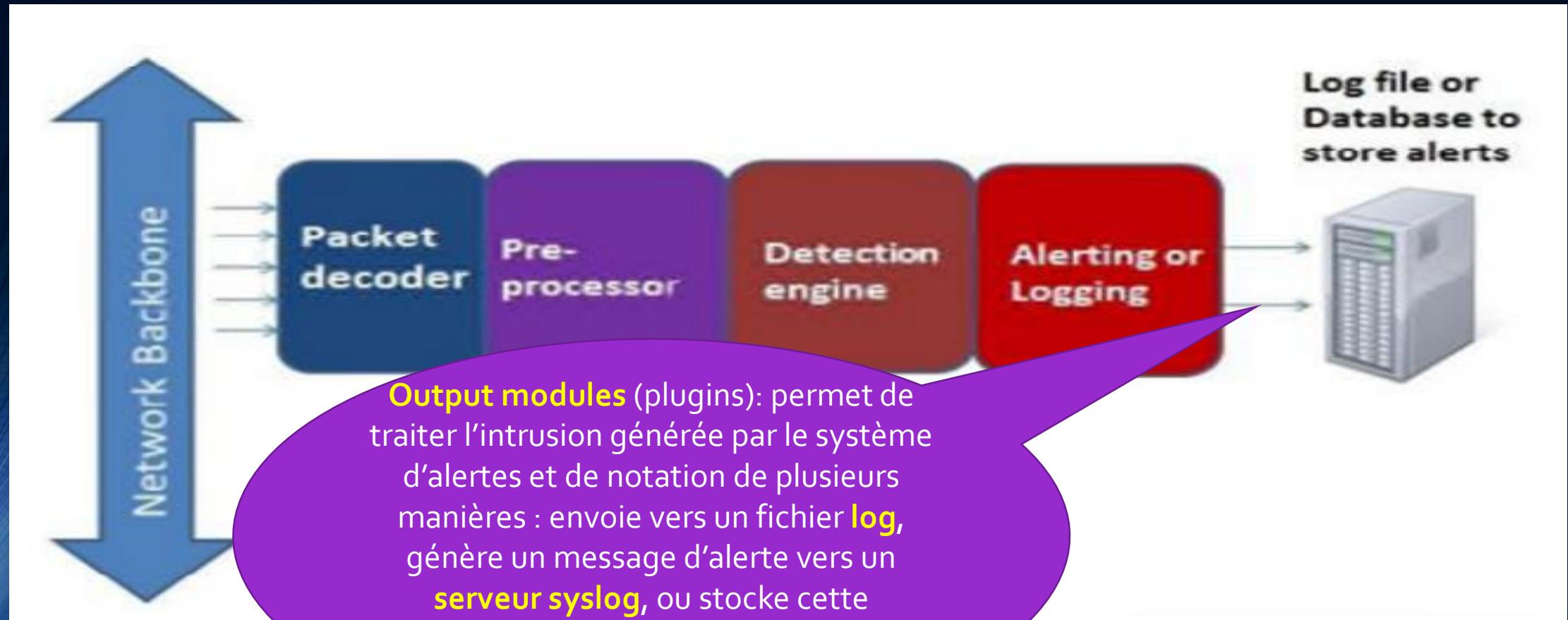
# Architecture de SNORT:



# Architecture de SNORT:



# Architecture de SNORT:



# Installation de SNORT:

Installer les conditions préalables des dépôts Ubuntu:

**sudo apt-get install -y build-essential libpcap-dev libpcre3-dev  
libdumbnet-dev bison flex zlib1g-dev**

Modification de **/etc/network/interfaces** comme un admin:

**sudo pico /etc/network/interfaces**

Rajouter les deux lignes suivantes pour chaque interface réseau :

**post-up ethtool -K eth0 gro off**  
**post-up ethtool -K eth0 lro off**

Création d'un répertoire pour enregistrer les fichiers téléchargé:

**mkdir ~/snort\_src**  
**cd ~/snort\_src**

## Installation de SNORT:

Snort utilise la bibliothèque d'acquisition de données (DAQ) aux appels abstraits par paquets bibliothèques de capture. DAQ est téléchargé et installer à partir du site Snort:

```
cd ~/snort src
wget https://www.snort.org/downloads/snort/daq-2.0.6.tar.gz
tar -xvzf daq-2.0.6.tar.gz
cd daq-2.0.6
./configure
make
sudo make install
```

# Installation de SNORT:

On installe Snort :

cd ~/snort src

wget https://www.snort.org/downloads/snort/snort-2.9.12.tar.gz

tar -xvzf snort-2.9.12.tar.gz

cd snort-2.9.12

./configure --enable-sourcefire --disable-open-appid

make

sudo make install

```
make[3]: Leaving directory `/home/karim/snort/snort-2.9.12/tools/u2spewfoo'  
make[3]: Entering directory `/home/karim/snort/snort-2.9.12/tools'  
make[3]: Nothing to be done for `all-am'.  
make[3]: Leaving directory `/home/karim/snort/snort-2.9.12/tools'  
make[2]: Leaving directory `/home/karim/snort/snort-2.9.12/tools'  
make[2]: Entering directory `/home/karim/snort/snort-2.9.12'  
make[2]: Leaving directory `/home/karim/snort/snort-2.9.12'  
make[1]: Leaving directory `/home/karim/snort/snort-2.9.12'
```

# Installation de SNORT:

Exécuter la commande suivante pour mettre à jour des bibliothèques partagées :  
**sudo ldconfig**

Placer le binaire local de Snort dans /usr/local/bin/snort et créer un lien symbolique vers /use/sbin/snort :  
**sudo ln -s /usr/local/bin/snort /usr/sbin/snort**

Tester que le binaire Snort fonctionne avec la commande suivante :  
**/usr/sbin/snort -V**

```
root@karim:~# snort -V

      ,,-> Snort! <*-  
o" )~ Version 2.9.6.0 GRE (Build 47)  
'``` By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-team  
Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.  
Copyright (C) 1998-2013 Sourcefire, Inc., et al.  
Using libpcap version 1.5.3  
Using PCRE version: 8.31 2012-07-06  
Using ZLIB version: 1.2.8
```

```
root@karim:~# █
```

# Installation de SNORT:

Création de l'utilisateur Snort et le groupe :

**sudo groupadd snort**

**sudo useradd snort -r -s /sbin/nologin -c SNORT\_IDS -g snort**

Création des répertoire de Snort:

**sudo mkdir /etc/snort**

**sudo mkdir /etc/snort/rules**

**sudo mkdir /etc/snort/rules/iplists**

**sudo mkdir /etc/snort/preproc\_rules**

**sudo mkdir /usr/local/lib/snort\_dynamicrules**

**sudo mkdir /etc/snort/so\_rules**

# Installation de SNORT:

Création des fichiers qui stocke les règles:

**sudo touch /etc/snort/rules/iplists/black\_list.rules**  
**sudo touch /etc/snort/rules/iplists/white\_list.rules**  
**sudo touch /etc/snort/rules/local.rules**

Création des répertoires de journalisation:

**sudo mkdir /var/log/snort**  
**sudo mkdir /var/log/snort/archived\_logs**

Régler les autorisations :

**sudo chmod -R 5775 /etc/snort**  
**sudo chmod -R 5775 /var/log/snort**  
**sudo chmod -R 5775 /var/log/snort/archived\_logs**  
**sudo chmod -R 5775 /etc/snort/so\_rules**  
**sudo chmod -R 5775 /usr/local/lib/snort\_dynamicrules**

## Installation de SNORT:

Changement de propriété sur les dossiers :

**sudo chown -R snort:snort /etc/snort**

**sudo chown -R snort:snort /var/log/snort**

**sudo chown -R snort:snort /usr/local/lib/snort\_dynamicrules**

Déplacer les fichiers suivants vers /etc/snort avec les commandes suivantes :

**cd ~/snort/snort-2.9.12/etc/**

**sudo cp \*.conf\* /etc/snort**

**sudo cp \*.map /etc/snort**

**sudo cp \*.dtd /etc/snort**

**cd ~/snort\_src/snort-2.9.12/src/dynamic-preprocessors/build/usr/local/lib/snort\_dynamicprocessor/**

**sudo cp \* /usr/local/lib/snort\_dynamicprocessor/**

## Installation de SNORT:

Modification du fichier de configuration Snort. Le fichier de configuration Snort est stocké à /etc/snort/snort.conf, et contient tous les paramètres que Snort va utiliser quand il est exécuté en mode NIDS.

Mettre en commentaire toutes les règles de Snort avec la commande suivante :

**sudo sed -i 's/include \\$RULE\_PATH/#include \\$RULE\_PATH/' /etc/snort/snort.conf**

Modifier quelques lignes dans le fichier snort.conf, on ouvre le fichier Snort.conf avec la commande suivante :

**sudo pico /etc/snort/snort.conf**

## Installation de SNORT:

ipvar HOME\_NET 192.168.1.0/24

Insertion des chemins et des répertoires que nous avons créés plus tôt comme suit :

var RULE\_PATH /etc/snort/rules

var SO\_RULE\_PATH /etc/snort/so\_rules

var PREPROC\_RULE\_PATH /etc/snort/preproc\_rules

var WHITE\_LIST\_PATH /etc/snort/rules/iplists

var BLACK\_LIST\_PATH /etc/snort/rules/iplists

# Installation de SNORT:

```
karim@karim: ~/snort/snort-2.9.12/src/dynamic-preprocessors/build/usr/local/lib/snort_dynamicprep
GNU nano 2.2.6          File: /etc/snort/snort.conf          Modified

# List of file data ports for file inspection
portvar FILE_DATA_PORTS [$HTTP_PORTS,110,143]

# List of GTP ports for GTP preprocessor
portvar GTP_PORTS [2123,2152,3386]

# other variables, these should not be modified
ipvar AIM_SERVERS [64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.$

# Path to your rules files (this can be a relative path)
# Note for Windows users: You are advised to make this an absolute path,
# such as: c:\snort\rules
var RULE_PATH /etc/snort/rules
var SO_RULE_PATH /etc/snort/so_rules
var PREPROC_RULE_PATH /etc/snort/preproc_rules
```

## Installation de SNORT:

Activer les règles qu'on veut utiliser en enlevant le « # » qui se trouve en début de la ligne, maintenant on va éditer un seul fichier règles local.rules et pour que Snort l'utilise on doits lui enlevé le # qui rend la ligne comme commentaire

include \$RULE\_PATH/local.rules

## Installation de SNORT:

Pour s'assurer du bon fonctionnement de Snort on exécute la commande suivante :

**sudo snort -T -c /etc/snort/snort.conf**

Pour écrire sur le fichier local.rules on ouvre d'abord le fichier à l'aide de la commande :

**sudo pico /etc/snort/rules/local.rules**

Puis on écrit les règles qui nous intéresse le plus, exemple :  
**alert ip any any -> any any (msg:"ICMP test"; sid:10000001;**  
**rev:001;)**

Cette règle dit que pour tous les paquets IP qu'il voit dans tout le réseau, génère une alerte avec le test IP texte

# Installation de SNORT:

Apres avoir apporté des modifications aux fichiers qui reniflent des charges, on doit tester le fichier de configuration à nouveau:

**sudo snort -T -c /etc/snort/snort.conf**

```
karim@karim: ~/snort_src/snort-2.9.12/src/dynamic-preprocessors/build/usr/local/lib/snort_dynamicpr
Using libpcap version 1.5.3
Using PCRE version: 8.31 2012-07-06
Using ZLIB version: 1.2.8

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 3.0 <Build 1>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_POP Version 1.0 <Build 1>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>

snort successfully validated the configuration!
snort exiting
karim@karim:~/snort_src/snort-2.9.12/src/dynamic-preprocessors/build/usr/local/lib/snort_dy
namicprocessor$ █
```

# Installation de SNORT:

On lance SNORT avec la commande suivante:

**sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf**

```
karim@karim: ~/snort_src/snort-2.9.12/src/dynamic-preprocessors/build/usr/local/lib/snort_dynamicpr
12/17-06:05:33.251300  [**] [1:10000001:1] IP Packet detected [**] [Priority: 0] {UDP} fe
30::f0ae:8bd9:27fa:688d:54967 -> ff02::c:1900
^C*** Caught Int-Signal
12/17-06:05:37.252719  [**] [1:10000001:1] IP Packet detected [**] [Priority: 0] {UDP} fe
30::f0ae:8bd9:27fa:688d:54967 -> ff02::c:1900
<carim@karim:~/snort_src/snort-2.9.12/src/dynamic-preprocessors/build/usr/local/lib/snort_
dynamicprocessor$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf
12/17-06:08:02.298897  [**] [1:10000001:1] IP Packet detected [**] [Priority: 0] {UDP} 19
2.168.137.1:1900 -> 239.255.255.250:1900
12/17-06:08:02.299072  [**] [1:10000001:1] IP Packet detected [**] [Priority: 0] {UDP} fe
30::f0ae:8bd9:27fa:688d:1900 -> ff02::c:1900
12/17-06:08:02.353879  [**] [1:10000001:1] IP Packet detected [**] [Priority: 0] {UDP} 19
2.168.137.1:1900 -> 239.255.255.250:1900
12/17-06:08:02.354187  [**] [1:10000001:1] IP Packet detected [**] [Priority: 0] {UDP} fe
30::f0ae:8bd9:27fa:688d:1900 -> ff02::c:1900
12/17-06:08:03.270719  [**] [1:10000001:1] IP Packet detected [**] [Priority: 0] {UDP} fe
30::f0ae:8bd9:27fa:688d:54967 -> ff02::c:1900
12/17-06:08:07.271405  [**] [1:10000001:1] IP Packet detected [**] [Priority: 0] {UDP} fe
30::f0ae:8bd9:27fa:688d:54967 -> ff02::c:1900
^C*** Caught Int-Signal
12/17-06:08:10.271144  [**] [1:10000001:1] IP Packet detected [**] [Priority: 0] {UDP} fe
30::f0ae:8bd9:27fa:688d:54967 -> ff02::c:1900
<carim@karim:~/snort_src/snort-2.9.12/src/dynamic-preprocessors/build/usr/local/lib/snort_
dynamicprocessor$
```