Etude et déploiement d'une solution de sécurité de type IDS

09.2018 - 01.2019

FALL Papa Alioune

4A Informatique / Electronique

Parcours Sécurité & Qualité des Réseaux

Esirem -Dijon 21000

Tuteur

M. Nader MBAREK

TABLE DES MATIÈRES

Liste des Tableaux

Liste des figures

REMERCIEMENTS

Avant de commencer je tiens à remercier le tuteur de mon projet M. Nader MBAREK qui m'a aidé et encadré durant ce projet. Les conseils qu'il m'a fournis, les échanges par email, les réunions, les rapports hebdomadaire et son assistance m'a faciliter de mener à bien ce projet et la rédaction du compte rendu.
Par ailleurs je remercie aussi les collègues de 5éme année pour les orientations de recherches qu'ils m'ont fournis.

Introduction générale:

Aujourd'hui les réseaux informatiques occupent une place importante dans les activités quotidiennes des individus. Le développement de ces systèmes d'information et leurs expansions, notamment de l'internet sont très bénéfique pour les structures professionnelles et les entreprises. Cependant les systèmes informatiques des entreprises sont confrontés à des attaques informatiques qui entraînent des pertes de données et d'informations. De ce fait la mise en place d'un politique de sécurité tel que les systèmes de détection d'intrusion sont primordiaux car il nous permettra de détecter une intrusion en temps réel selon des règles définies.

I. Etat de l'art sur les solutions de type IDS: Classification, principe de fonctionnement, technologies

I.1. Définition:

Les IDS (Intrusion Detection System), ou systèmes de détection d'intrusions, sont des mécanismes mises en place au sein d'une infrastructure (entreprise) pour détecter les activités anormales. En cas de menaces une alerte pourra éventuellement être envoyé à l'administrateur système.

I.2. Classification des IDS:

On effectue la classification des IDS en fonction de leurs domaines de surveillance. On distingue deux familles d'IDS que nous allons donner les fonctionnement ci-dessous:

- **Détection d'intrusion au niveau du réseau(N-IDS):** repose essentiellement sur l'analyse et l'interprétation du trafic dans le réseau. Il se met à l'écoute du réseau pour analyser le trafic afin de détecter les éventuelles attaques. Ensuite, les paquets sont décortiqués puis analysés.
- **Détection d'intrusion au niveau des hôtes(H-IDS):** analysent le fonctionnement de l'état des machines sur lesquelles ils sont installés afin de détecter les attaques. L'intégrité des systèmes est alors vérifiée périodiquement et des alertes peuvent être levées.
- **Détection d'intrusion Hybrides:** Les IDS hybrides sont basés sur une architecture distribué. On y retrouve à la fois les caractéristiques des NIDS et HIDS. Ils permettent de surveiller le réseau et les hôtes à la fois par le biais d'un seul système. Le système agit comme un **H-IDS** ou **N-IDS** suivant son placement dans le réseau.

Types	Avantages	Inconvénients
N-IDS	-Plus adapté pour les réseaux de grande taille, contrôle un grand nombre d'hôtes -identifier les attaques de plusieurs hôtes. -assurer la sécurité contre les attaques puisqu'il est invisible	-difficulté de fonctionner dans des environnements cryptés -ne permet pas d'assurer si une tentative d'attaque est couronnée de succès

H-IDS	-Plus précis sur les types d'attaques subies (ils n'ont pas à contrôler le trafic du réseau) -Plus de précision sur la zone d'intrusion -capable de fonctionner dans des environnements cryptés	-Grande quantités de données générées, Taille des fichiers de rapport d'alerte conséquent -la difficulté de déploiement et de gestion, surtout lorsque le nombre d'hôte qui ont besoin de protection est large. -incapable de détecter des attaques contre de multiples cibles dans le réseau.

Tableau 1: Comparaison entre NIDS et HIDS

I.3. Critères de choix:

La mise en place d'un système de détection d'intrusion dans les entreprises est devenu quasi indispensable face au menace informatiques. Le choix d'outil à adopter dépendra en grande partie du contexte d'étude:
Fiabilité: repose particulièrement sur les alertes envoyés. On de s'assurer que les alertes envoyés sont considérés comme des attaques et que les toutes attaques doivent être détectés.
Réactivité: capacité à détecter une intrusion le rapidement possible, d'où la nécessité de de le garder à jour.
Performance: l'IDS ne doit pas affecter le trafic du réseau surveillé
Facilité de mise en œuvre et adaptabilité: il doit être facile à mettre en œuvre et doit être adapté au trafic du réseau qu'il est installé.
Multicanal: utilisation de différentes canaux d'alerte pour garantir une sécurité sur tous les attaques possibles.

I.4. Etude des logiciels et plateformes d'implémentation de solution de type IDS

Actuellement il existe plusieurs solutions de type IDS sur le marché. Nous allons voir quelques un dans cette partie:

- **Snort:** est un outil de détection d'intrusion(IDS) libre. Il permet l'analyse du trafic réseau en temps réel et la sauvegarde des paquets afin d'avoir une analyse détaillée. Il surveille les paquets émis et reçus sur une interface réseau et permet de détecter alors les tentatives d'intrusions sur les segments du réseau. Pour détecter les intrusions et déterminer les types de trafic réseau, Snort utilise des règles qui sont dites "règles de Snort". Ces règles sont appliquées sur les protocoles ICMP, TCP et UDP.
- **Bro IDS:** est un IDS de type réseau (N-IDS), open Source et disponible pour les systèmes d'exploitation de type Unix tels que Linux. Il analyse le trafic du réseau en recherchant les activités suspectes. L'analyse se fait de manière passive et transparente. La détection d'intrusion se fait en deux étapes.
    - **Premièrement** il capte le trafic réseau et décode les différentes couches protocolaires
    - **Deuxièmement** il analyse les événements générés lors de la première étape par des scripts d'analyse. Ces scripts comparent ces événements par rapport à des motifs caractérisant des comportements réputés anormaux. Cette analyse permet à la fois la détection d'attaques connues au préalable (qui sont décrites en termes de signatures ou d'événements) et d'anomalies

(par exemple, la présence de connexions de certains utilisateurs vers s certains services ou l'occurrence de tentatives de connexions infructueuses).

- **OSSEC**: est un système de détections d'intrusion de type HIDS (Système de détection d'intrusion basé sur les hôtes). Il fonctionne sur sur des serveurs de plusieurs environnements tels que: Linux, Solaris, Windows, Mac... Il offre les fonctionnalités suivantes:
    - Vérification de l'intégrité des fichiers systèmes.
    - Supervision, analyse et corrélation des logs du système.
    - Prévention active en cas de détection d'attaques.
    - Détection des outils de dissimulation d'activité appelés souvent rootkit.

- **Prélude:** est un IDS hybride et Open Source de détection d'intrusions et d'anomalies. La détection d'intrusion par Prélude-IDS est réalisée par l'analyse du trafic réseau, l'utilisation de signatures d'évènements hostiles et par l'analyse des fichiers de logs. Prélude IDS collecte les événements de sécurité, les normalise, les trie, les corrèle et affiche tous ces événements indépendamment des types d'équipements et des système surveillés. Le point fort du Prélude est qu'il offre une solution complète de détection d'intrusions systèmes(H-IDS) et réseaux(N-IDS). Cependant, son point faible est qu'il ne peut pas être installé sous Windows, en plus sa base de données des signatures d'attaques demeure pauvre par rapport aux outils les plus utilisés comme SNORT.
- **SURICATA:** Suricata est un outil de détection d'intrusion gratuit et open source développée depuis 2008 par la fondation OISF. Il est basé sur des signatures et offre des possibilités intéressantes en termes d'analyse protocolaire et de suivi de l'activité réseau. La première version stable de Suricata date de 2010. L'objectif de cette version était d'avoir un moteur d'IDS/IPS multithread supportant le langage de signatures de Snort [SNORT], ce qui permettait de conserver l'existant en termes de signatures.

Sur la base de ces informations on peut faire un tableau comparatif pour les différents outils qu'on vient de voir.

Nom IDS	**Type**	**acquisition**	**Méthode de détection d'intrusion**
Snort	N-IDS	Open Source	● Analyse du trafic réseau en temps réel ● Sauvegarde des paquets afin d'avoir une analyse détaillée ● utilisation des règles
Bro	N-IDS	Open Source	● Analyse le trafic du réseau en recherchant les activités suspectes ● Analyse passive et transparente
OSSEC	H-IDS	Open Source	● Analyse et corrélation des logs du système ● Supervision des logs du système ● Détection des outils de dissimulation d'activité
Prelude	N-IDS / H-IDS	Open Source	● Analyse du trafic réseau, ● Utilisation de signatures d'évènements hostiles ● Analyse des fichiers de logs

SURICATA	N-IDS	Open source	● Analyse le trafic réseau en comparant les paquets à un ensemble de signatures cherchant des motifs ressemblant à des attaques ● capture, détection et création d'alertes ou de fichiers de journalisation

Tableau 2: Comparaison des différents solutions IDS

II. Choix et déploiement d'une solution de type IDS

II.1. Choix du logiciel

La première étape de ce projet m'a permis de faire un choix concernant le type d'IDS à choisir. J'ai adopté d'implémenter une solution de type N-IDS vu qu'il nous offre plus de possibilités et plus de couverture pour mieux assurer la détection d'intrusion d'un environnement. Sur cette base le logiciel SNORT est ma préférence du fait qu'il est plus répandu parmi les logiciels de détection d'intrusion. En plus snort est un système open source, fournit l'analyse complète du trafic et journalise le contenu des paquets entrants et sortants.
Pour implémenter la solution choisie, on utilisera la distribution unix qui est gratuit dans lequel on va installera le système d'exploitation Linux. Ce système a été adopté parce qu'il m'est le plus familier et possède des versions stables.

II.2. Propositions de scénarios d'attaques de sécurité

Les attaques que peuvent subir un système d'information sont nombreux. De ce fait on peut faire la détection des attaques selon différents niveaux, parmi lesquels nous avons:

- **Détection au niveau des protocoles (IP, ICMP, UDP, TCP):** appelé aussi attaque réseau. Il se base principalement sur des failles liées aux protocoles ou leur implémentation. Pour chaque protocole défini on peut définir des règles à appliquer lorsque ce protocole est sollicité dans un trafic.
- **Détection par déni de service:** couramment appelé attaque **DOS,** il vise à rendre inaccessible un service. Il peut se faire par une surcharge du réseau rendant ainsi la machine totalement injoignable ou de manière applicative en crachant l'application à distance.
- **Détection par activités anormales:** on peut aussi détecter d'autres intrusion en définissant nos propres règles en fonction du trafic de notre réseau et du besoin de la structure.

II.3. Installation de SNORT

Avant de commencer l'installation de SNORT, il nous faut d'abord préparer le système hôte en s'assurant qu'il est à jour et que tous les logiciels installés exécutent la dernière version.
La procédure d'installation est fournie au niveau de la documentation officielle sur le site de **SNORT.** Cependant il nous faut installer un certains nombres de dépendances requises sur notre système:

```
sudo apt-get install -y build-essential autotools-dev libdumbnet-dev
libluajit-5.1-dev libpcap-dev \ libpcre3-dev zlib1g-dev pkg-config
libhwloc-dev
```

Figure 1: Installation des pré-requis de snort

- **Build-essential** : fournit les outils de construction (GCC et similaires) au logiciel de compilation.
- **Bison, flex** : analyseurs requis par DAQ (DAQ est installé plus tard).
- **Libpcap-dev** : bibliothèque pour la capture de trafic réseau requise par Snort.
- **Libpcre3-dev** : Bibliothèque de fonctions pour prendre en charge les expressions régulières requises par Snort.
- **Libdumbnet-dev** : la bibliothèque libdnet fournit une interface portable simplifiée à plusieurs routines de réseau de bas niveau. De nombreux guides pour installer Snort installent cette bibliothèque à partir de la source, bien que cela ne soit pas nécessaire.
- **Zlib1g-dev** : une bibliothèque de compression requise par Snort.
- **Liblzma-dev** : fournit une décompression des fichiers swf (Adobe Flash)
- **OpenSSL** et libssl-dev : Fournit des signatures de fichiers et SHA MD5 La bibliothèque finale requise par Snort est la bibliothèque de développement de Nghttp2 : une bibliothèque HTTP / 2 C qui implémente l'algorithme de compression d'en-tête HPAC

Après avoir installé les paquets requises, nous allons suivre les étapes renseigné en [ANNEXE 1] pour l'installation et la configuration de **SNORT.** Nous pouvons vérifier si l'installation a bien réussi sur la figure suivante:

root@fall-HP-ENVY-TS-15-Notebook-PC:~# /usr/local/bin/snort -V

```
root@fall-HP-ENVY-TS-15-Notebook-PC:~# /usr/local/bin/snort -V
   ,,_       -*> Snort++ <*-
  o"  )~   Version 3.0.0 (Build 250) from 2.9.11
   ''''      By Martin Roesch & The Snort Team
             http://snort.org/contact#team
             Copyright (C) 2014-2018 Cisco and/or its affiliates. All rights
reserved.
             Copyright (C) 1998-2013 Sourcefire, Inc., et al.
             Using DAQ version 2.2.2
             …
             Using ZLIB version 1.2.11
             Using FlatBuffers 1.9.0
             Using LZMA version 5.2.2
```

Figure 2: Vérification de l'installation de Snort

Pour finir l'installation il est bien de créer un lien local sur **/usr/local/sbin** pour **SNORT**.

II.4. Réalisation des scénarios d'attaque

Dans cette partie nous allons présenter deux scénarios d'attaque d'intrusion dans un réseau. Cependant ces attaques sont très basiques et simples à mettre en œuvre et sont de loin les méthodes qu'utilisent les pirates. Néanmoins il nous permet de voir le comportement d'un IDS face à une attaque.

- **Détection d'intrusion de type Ping**

La première consiste à simuler une alerte dans un environnement réseau lorsqu'on effectue un **"PING".** Le scénario proposé est constitué d'une machine attaquant, d'un modem et d'une machine jouant le rôle de serveur avec l'outils **SNORT.**

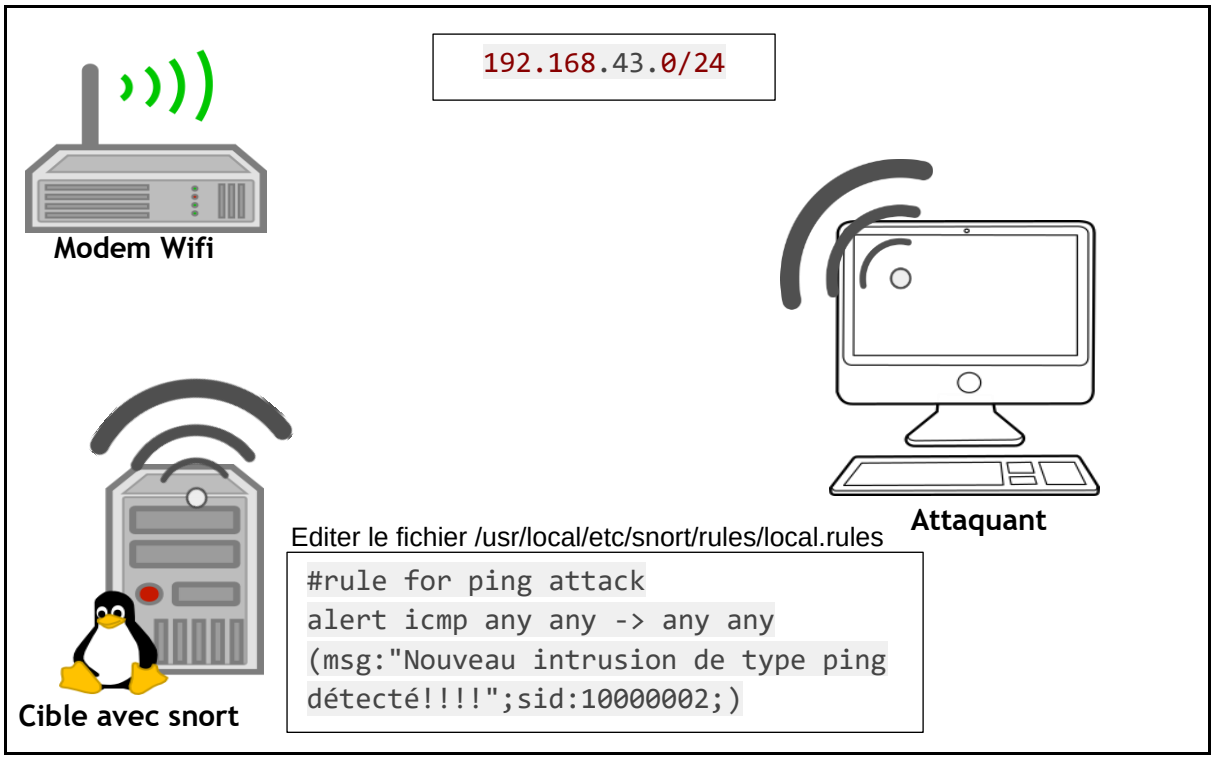

Figure 3: Premier scénario d'attaque

Nous allons d'abord afficher les configurations **IP** de la machine à attaquer avec la commande **ifconfig.** Puisque nous somme connecté en sans-fil on regarde l'adresse IP de l'interface **wlo1.**

```
fall@fall-HP-ENVY-TS-15-Notebook-PC:~$ ifconfig
eno1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
      ether ac:e2:d3:53:5b:91  txqueuelen 1000  (Ethernet)
      RX packets 0  bytes 0 (0.0 B)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 0  bytes 0 (0.0 B)
      TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
```

```
      inet 127.0.0.1  netmask 255.0.0.0
      inet6 ::1  prefixlen 128  scopeid 0x10<host>
      loop  txqueuelen 1000  (Local Loopback)
      RX packets 439  bytes 30041 (30.0 KB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 439  bytes 30041 (30.0 KB)
      TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.168.43.6  netmask 255.255.255.0  broadcast 192.168.43.255
      inet6 fe80::8020:95f4:1dda:caee  prefixlen 64  scopeid 0x20<link>
      ether 40:9f:38:b5:ac:9b  txqueuelen 1000  (Ethernet)
      RX packets 12043  bytes 15930418 (15.9 MB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 8479  bytes 862341 (862.3 KB)
      TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 4: Configuration ip de la machine cible.

Au niveau de la machine pirate, on lance la commande Ping avec l'adresse IP de la machine ciblée.

```
fall@fall-HP-ENVY-TS-15-Notebook-PC:~$ ping 192.168.43.6
PING 192.168.43.6 (192.168.43.6) 56(84) bytes of data.
64 bytes from 192.168.43.6: icmp_seq=1 ttl=64 time=0.065 ms
64 bytes from 192.168.43.6: icmp_seq=2 ttl=64 time=0.154 ms
64 bytes from 192.168.43.6: icmp_seq=3 ttl=64 time=0.089 ms
64 bytes from 192.168.43.6: icmp_seq=4 ttl=64 time=0.063 ms
64 bytes from 192.168.43.6: icmp_seq=5 ttl=64 time=0.061 ms
64 bytes from 192.168.43.6: icmp_seq=6 ttl=64 time=0.067 ms
```

Figure 5: Attaque Ping de la cible depuis la machine pirate

Au niveau de la machine avec le serveur on lance la commande suivante au niveau du terminal. Ceci permet de mettre snort en activité passive et écoute le trafic du réseau. Une fois un trafic de type **ICMP** détecté une alerte sera envoyée comme on le voit sur la figure suivante.

```
fall@fall-HP-ENVY-TS-15-Notebook-PC:~$ sudo snort -c
/usr/local/etc/snort/snort.lua -R /usr/local/etc/snort/rules/local.rules \-i
wlo1 -A alert_fast -s 65535 -k none
--------------------------------------------------
o")~   Snort++ 3.0.0-250
--------------------------------------------------
Loading /usr/local/etc/snort/snort.lua:
Finished /usr/local/etc/snort/snort.lua.
Loading builtin:
Finished builtin.
Loading rules:
Loading /usr/local/etc/snort/rules/local.rules:
```

```
Finished /usr/local/etc/snort/rules/local.rules.
Finished rules.
-------------------------------------------------
rule counts
      total rules loaded: 474
            text rules: 1
            builtin rules: 473
            option chains: 474
            chain headers: 2
-------------------------------------------------
port rule counts
             tcp    udp    icmp   ip
      any    473    0      1      0
   total     473    0      1      0
-------------------------------------------------
pcap DAQ configured to passive.
Commencing packet processing
++ [0] lo
12/13-14:04:04.516676 [**] [116:151:1] "(decode) same src/dst IP" [**]
[Priority: 3] {ICMP} 192.168.43.6 -> 192.168.43.6
12/13-14:04:04.516676 [**] [1:10000002:0] "Nouveau intrusion de type ping
détecté!!!!" [**] [Priority: 0] {ICMP} 192.168.43.6 -> 192.168.43.6
12/13-14:04:04.516697 [**] [116:151:1] "(decode) same src/dst IP" [**]
[Priority: 3] {ICMP} 192.168.43.6 -> 192.168.43.6
12/13-14:04:04.516697 [**] [1:10000002:0] "Nouveau intrusion de type ping
détecté!!!!" [**] [Priority: 0] {ICMP} 192.168.43.6 -> 192.168.43.6
```

Figure 6: Détection de l'attaque Ping par snort

Le résultat obtenu montre que **SNORT** a détecté les requêtes ping avec plusieurs informations comme: le type de paquet, la priorité, le port, l'adresse IP source et destination.

- **Détection d'intrusion de type DOS**

Dans ce deuxième scénario nous allons voir une des attaques les plus connues dans le monde de l'informatique le **DOS** (Denial Of Service). Le but de ce dernier est de rendre indisponible un service ou des ressources d'une cible durant une période. Ce type d'attaque n'engendre pas le vol, la suppression ou la modification des données.

Il existe plusieurs méthodes pour la réalisation d'une attaque par déni de service. Dans ce projet nous allons utiliser la méthode dite le **SYN FLOOD** en utilisant l'utilitaire **hping3.** Ce dernier envoie des paquets **TCP SYN** à la cible. Il s'agit des demandes de connexions en boucles sans effectuer d'acquittement de ces connexions. De ce fait, au niveau de la cible on a des réservations de ressources qui attendent d'être acquitté alors qu'elle n'en recevra jamais.

Notre scénario d'attaque est composé d'une machine attaquant, d'une machine cible hébergeant un serveur Apache et d'un modem wifi, et on tentera de rendre indisponible le serveur apache.

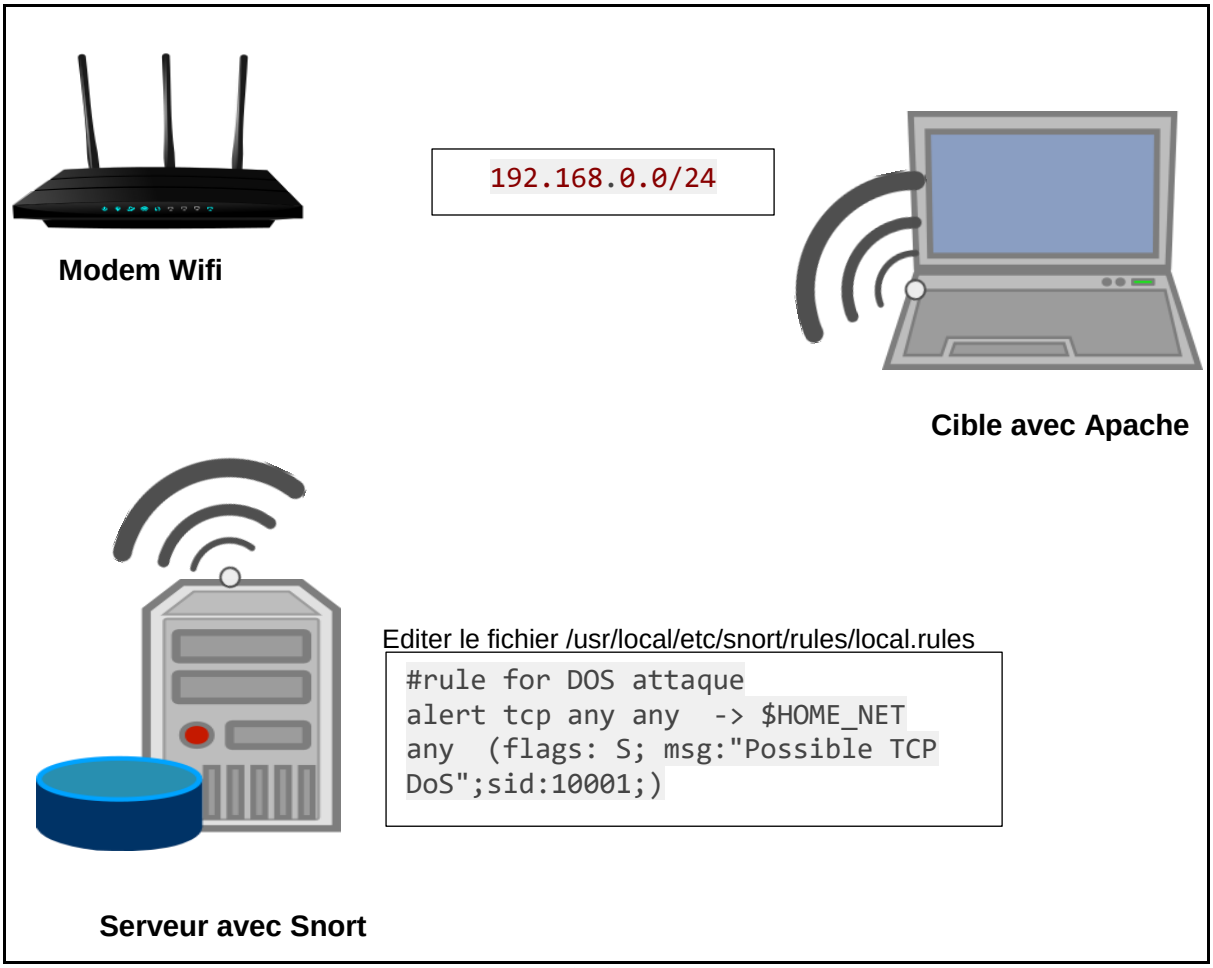

Figure 7: Détection de l'attaque Ping par snort

Nous allons d'abord scanner les ports ouverts sur la machine avec la commande **nmap.**

```
root@fall-HP-ENVY-TS-15-Notebook-PC:~# sudo nmap -sS -sV 192.168.0.14
Starting Nmap 7.60 ( https://nmap.org ) at 2019-01-10 00:24 CET
Nmap scan report for 192.168.0.14
Host is up (0.00052s latency).
Not shown: 998 closed ports
PORT  STATE SERVICE VERSION
80/tcp   open  http     Apache httpd 2.4.29 ((Ubuntu))
8080/tcp open  http     Apache httpd 2.4.29 ((Ubuntu))
```

Figure 8: Scan port ouvert sur la machine cible

La figure 8 nous montre que les ports 80 et 8080 sont ouvert et que le serveur apache les utilise.

Sachant sur quel port tourne le service apache on peut maintenant simuler l'attaque DOS avec la commande **hping3** en lui passant l'adresse ip de la cible**.**

```
fall@fall-HP-ENVY-TS-15-Notebook-PC:~$ sudo hping3 --flood -p 80 -S
192.168.0.14
HPING 192.168.0.14 (wlo1 192.168.0.14): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.0.14 hping statistic ---
826296 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Figure 9: Attaque SYN FLOOD

Au niveau de la machine sur lequel on a installé **SNORT**, on fait la même chose que pour le premier scénario en écoutant le trafic. On obtient le résultat suivant:

```
fall@fall-HP-ENVY-TS-15-Notebook-PC:~$ sudo snort -c /usr/local/etc/snort/snort.lua
-R /usr/local/etc/snort/rules/local.rules \-i wlo1 -A alert_fast -s 65535 -k none
--------------------------------------------------
o")~   Snort++ 3.0.0-250
--------------------------------------------------
Loading /usr/local/etc/snort/snort.lua:
...
--------------------------------------------------
pcap DAQ configured to passive.
Commencing packet processing
++ [0] wlo1
01/10-01:21:15.609977 [**] [1:10001:0] "Possible TCP DoS" [**] [Priority: 0] {TCP}
192.168.0.11:2508 -> 192.168.0.14:80
01/10-01:21:15.610036 [**] [1:10001:0] "Possible TCP DoS" [**] [Priority: 0] {TCP}
192.168.0.11:2509 -> 192.168.0.14:80
01/10-01:21:15.610054 [**] [1:10001:0] "Possible TCP DoS" [**] [Priority: 0] {TCP}
192.168.0.11:2510 -> 192.168.0.14:80
01/10-01:21:15.610069 [**] [1:10001:0] "Possible TCP DoS" [**] [Priority: 0] {TCP}
192.168.0.11:2511 -> 192.168.0.14:80
01/10-01:21:15.610083 [**] [1:10001:0] "Possible TCP DoS" [**] [Priority: 0] {TCP}
192.168.0.11:2512 -> 192.168.0.14:80
```

Figure 10 : Détection de l'attaque SYN FLOOD

Le résultat obtenu montre que **SNORT** a détecté l'attaque de type **SYN FLOOD** avec plusieurs informations comme pour le premier scénario à savoir: le type de paquet, la priorité, le port, l'adresse IP source et destination.

Conclusion générale:

La sécurité des systèmes d'information constitue un enjeu majeur dans le monde informatique. Les attaques sont de plus en plus fréquent, difficiles à maîtriser et presque imprévisible.

Ce projet nous a permis d'acquérir certaines notions sur la sécurité informatique et un moyen pour mettre en place un système de détection d'intrusion.

Les systèmes de détection d'intrusion sont actuellement mis en place dans les entreprises pour assurer la protection de leurs données. Ils peuvent être comparé à des alarmes qui se déclenchent lorsqu'il a une activité considérée comme intrusion, définis dans les configurations de l'outils utilisé.

On notera que les IDS ont tendance à envoyer enormement d'alertes s'ils ne sont pas bien configurés. Ceci engendre l'envoie de fausses alertes aux adminitrateurs. D'ou la nécessité de trouver un moyen complémentaire pour une protection efficace et fine de notre sytéme d'informations.

.

ANNEXE 1

Mettre à jour le système

```
sudo apt-get update && sudo apt-get dist-upgrade -y
```

creer un dossier pour les telechargement à faire

```
mkdir ~/snort_src
cd ~/snort_src
```

Installation des pre-requis de snort

```
sudo apt-get install -y build-essential autotools-dev libdumbnet-dev libluajit-5.1-dev
libpcap-dev \ libpcre3-dev zlib1g-dev pkg-config libhwloc-dev
```

Pour **Ubuntu 16 et 18**, installer cmake à partir du dépôt par défaut

```
# Ubuntu 16 and 18 (and greater) only:
sudo apt-get install -y cmake
```

Pour Ubuntu 14, installer cmake depuis la source, car la version du répertoire Ubuntu est trop ancienne pour compiler Snort

```
# Ubuntu 14 only sudo apt-get remove -y cmake cd ~/snort_src wget
https://cmake.org/files/v3.10/cmake-3.10.3.tar.gz tar -xzvf cmake-3.10.3.tar.gz cd
cmake-3.10.3 ./bootstrap make sudo make install
```

Ensuite, installation de certains logiciels optionnel (mais fortement recommandé) :

```
sudo apt-get install -y liblzma-dev openssl libssl-dev cpputest libsqlite3-dev \ uuid-dev
```

Construire la documentation la plus récente à partir de l'arborescence des sources, y compris Snort++ Developers Guide, installez les paquets suivants (purement facultatif).

```
sudo apt-get install -y asciidoc dblatex source-highlight w3m
```

Puisque nous installerons Snort depuis le dépôt Github, nous avons besoin de quelques outils :

```
sudo apt-get install -y libtool git autoconf
```

Installation des pré-requis pour la bibliothèque Snort DAQ (Data Acquisition):

```
sudo apt-get install -y bison flex
```

Si vous voulez exécuter Snort en mode inline en utilisant NFQ, installez les paquets requis (non requis pour le mode IDS ou en mode inline en utilisant afpacket). Si vous n'êtes pas sûr, vous devez installer ce paquet.

```
sudo apt-get install -y libnetfilter-queue-dev
```

Téléchargement et installation safec pour vérifier les limites d'exécution de certains appels de la bibliothèque C (facultatif mais recommandé) :

```
cd ~/snort_src
wget https://downloads.sourceforge.net/project/safeclib/libsafec-10052013.tar.gz
tar -xzvf libsafec-10052013.tar.gz cd libsafec-10052013
./configure
make
sudo make install
```

Téléchargez et installez gperftools 2.7, le thread-caching malloc de Google (utilisé en chrome).

```
cd ~/snort_src
wget https://github.com/gperftools/gperftools/releases/download/gperftools-2.7/gperftools-2.7.tar.gz
tar xzvf gperftools-2.7.tar.gz
cd gperftools-2.7
./configure
make
sudo make install
```

Téléchargez et installez Ragel.
Snort3 utilise Hyperscan pour l'appariement rapide des motifs. Hyperscan nécessite Ragel et les headers Boost : :

```
cd ~/snort_src
wget http://www.colm.net/files/ragel/ragel-6.10.tar.gz tar -xzvf ragel-6.10.tar.gz
cd ragel-6.10
./configure
make
sudo make install
```

Téléchargez les bibliothèques Boost 1.67.0, (mais n'installez pas) pour Hyperscan:

```
cd ~/snort_src
wget https://dl.bintray.com/boostorg/release/1.67.0/source/boost_1_67_0.tar.gz
tar -xvzf boost_1_67_0.tar.gz
```

Installation de Hyperscan 4.7.0

```
 cd ~/snort_src wget https://github.com/intel/hyperscan/archive/v4.7.0.tar.gz
tar -xvzf v4.7.0.tar.gz
mkdir ~/snort_src/hyperscan-4.7.0-build
cd hyperscan-4.7.0-build/
cmake -DCMAKE_INSTALL_PREFIX=/usr/local -DBOOST_ROOT=~/snort_src/boost_1_67_0/
../hyperscan-4.7.0
```

```
make
sudo make install
```

Tester que Hyperscan fonctionne, à partir du répertoire build:

```
cd ~/snort_src/hyperscan-4.7.0-build/
./bin/unit-hyperscan
```

Téléchargement et installation des bibliothèque de sérialisation pour **SNORT**

```
 cd ~/snort_src
wget https://github.com/google/flatbuffers/archive/v1.9.0.tar.gz -O flatbuffers-
v1.9.0.tar.gz
tar -xzvf flatbuffers-1.9.0.tar.gz mkdir flatbuffers-build
cd flatbuffers-build
cmake ../flatbuffers-1.9.0
make
sudo make install
```

Téléchargement et installation de la bibliothèque d'acquisition de données (DAQ) à partir du site Web de Snort. Notons que Snort 3 utilise un DAQ différent de la série Snort 2.9.x. x

```
cd ~/snort_src
wget https://www.snort.org/downloads/snortplus/daq-2.2.2.tar.gz
tar -xvzf daq-2.2.2.tar.gz
cd daq-2.2.2
./configure
make
sudo make install
```

Update shared libraries:

```
sudo ldconfig
```

Récupération et installation du dernier **Snort3** de sur Github :

```
cd ~/snort_src
git clone git://github.com/snortadmin/snort3.git
cd snort3 ./configure_cmake.sh --prefix=/usr/local --enable-tcmalloc
cd build
make
sudo make install
```

Bibliographie

https://www.securiteinfo.com/conseils/choix_ids.shtml
http://info2aaz.blogspot.com/2010/09/la-detection-dintrusion-principe-et.html
https://www.securiteinfo.com/conseils/choix_ids.shtml
http://igm.univ-mlv.fr/~duris/NTREZO/20032004/Baudoin-Karle-IDS-IPS.pdf
https://connect.ed-diamond.com/MISC/MISC-066/Presentation-de-l-IDS-IPS-Suricata2
https://bfmbusiness.bfmtv.com/01-business-forum/snort-3-0-outil-de-detection-dintrusion-open-source-sera-plus-performant-398895.html
https://korben.info/ids-windows-patriot.html
http://www.logicielsmalins.fr/2014/12/ossec-loutil-de-detection-dintrusion.html
https://www.zdnet.fr/actualites/deployer-un-systeme-de-detection-d-intrusion-2118189.htm
https://www.lemagit.fr/conseil/Construire-un-systeme-de-prevention-et-detection-dintrusions-pour-le-Cloud
http://www.kookyoo.net/dossier/Informatique/ids-snort-prelude-presentation-fonction-page2-00000016
https://www.debian-fr.org/t/ids-lequel-choisir/56845/14
https://www.supinfo.com/articles/single/4920-mise-place-snort-299x-ubuntu
https://wiki.monitoring-fr.org/securite/snort/snort-ubuntu-install
https://dbprog.developpez.com/securite/ids/#LII-B-4
https://www.commentcamarche.com/contents/237-systemes-de-detection-d-intrusion-ids
https://it-central.fr/snort-et-pfsense/
http://jacquesgoueth.blogspot.com/2017/07/comment-mettre-en-place-un-systeme-de.html
https://snort-org-site.s3.amazonaws.com/production/document_files/files/000/000/138/original/Snort_3.0.0-a4-245_on_Ubuntu_14_16_18.pdf?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIXACIED2SPMSC7GA%2F20181202%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20181202T203432Z&X-Amz-Expires=172800&X-Amz-SignedHeaders=host&X-Amz-Signature=03771a5626ab23703c8931e118c375ac3ada321b4833ae579619f8bdd38192ee
https://blog.rapid7.com/2017/01/11/how-to-install-snort-nids-on-ubuntu-linux/
https://www.supinfo.com/articles/single/4920-mise-place-snort-299x-ubuntu
https://homputersecurity.com/2016/09/15/comment-fonctionnent-les-attaques-ddos-demo-avec-hping/
https://openclassrooms.com/fr/courses/1240136-mise-en-place-des-serveurs-apache-et-dns
https://www.blackmoreops.com/2015/04/21/denial-of-service-attack-dos-using-hping3-with-spoofed-ip-in-kali-linux/
https://www.system-linux.eu/index.php?post/2008/09/10/DOS-avec-hping
https://www.blackmoreops.com/2015/04/21/denial-of-service-attack-dos-using-hping3-with-spoofed-ip-in-kali-linux/
https://piratercommeunnul.wordpress.com/2013/06/07/attaquer-par-deni-de-service-comme-un-null/
https://www.zdnet.fr/actualites/deployer-un-systeme-de-detection-d-intrusion-2118189.htm
https://www.lemagit.fr/conseil/Construire-un-systeme-de-prevention-et-detection-dintrusions-pour-le-Cloud
http://www.kookyoo.net/dossier/Informatique/ids-snort-prelude-presentation-fonction-page2-00000016
https://www.debian-fr.org/t/ids-lequel-choisir/56845/14
https://www.supinfo.com/articles/single/4920-mise-place-snort-299x-ubuntu
https://wiki.monitoring-fr.org/securite/snort/snort-ubuntu-install