Institut Universitaire de Technologie,
Aix-Marseille Université

RAPPORT DE STAGE
Diplôme Universitaire de Technologie
Spécialité Réseaux et Télécommunications

Déploiement d'un Système de détection d'intrusion

Bastien BADOT

Med Europe Terminal

Responsable entreprise : Aurélie VIGNES

Responsable académique : Ivan MADJAROV

2021

Introduction

Dans l'optique de valider ma dernière année de **DUT*** (Diplôme universitaire de technologie) en Réseaux et Télécommunications, j'ai eu l'opportunité de pouvoir réaliser mon stage d'une durée de 10 semaines en tant qu'administrateur systèmes et réseaux, au sein du service informatique de l'entreprise de manutention portuaire : Med Europe Terminal.

Suite à une violente **cyberattaque*** datant d'il y a un an, il m'a été demandé de mettre en place un **IDS***(Intrusion détection system), afin d'analyser le trafic réseaux de toute l'entreprise et détecter de potentielles cyberattaques, aussi bien internes qu'externes.
De plus, il me fallait également trouver un moyen d'alerter le service informatique pour qu'il puisse réagir rapidement en cas de détection.

Dans ce document, je vais commencer par présenter en détail l'entreprise Med Europe Terminal et ses nombreux partenaires, afin de poser le cadre dans lequel mon stage s'est déroulé.
Par la suite, je discuterai de la technologie IDS et de ses applications dans le monde professionnel.
Enfin je terminerai par une description complète de ma réalisation lors de ce stage.

1 Présentation de l'entreprise

1.1 Présentation

MedEurope Terminal est une société travaillant dans le domaine de la manutention portuaire, c'est-à-dire qu'elle met en œuvre des processus logistiques afin de décharger des porte-conteneurs et autres bateaux de transport pour par la suite charger la marchandise qui sera transportée par voie terrestre vers sa destination.

Son domaine d'activité se situe sur le grand port maritime de Marseille s'étendant du 2ème arrondissement jusqu'au 16ème arrondissement, l'entreprise dispose de plusieurs sites à des endroits stratégiques du port.
De cette façon, les employés du service informatique doivent se déplacer pour naviguer entre les différentes zones du port et les différents sites, pour cela une voiture de fonction leur est nécessaire.

La criticité du domaine d'activité de Med Europe Terminale fait d'elle une zone à accès restreint : en effet, pour pouvoir rentrer sur le port il faut déjà un laissez-passer et il existe également un deuxième point de contrôle, où il faut une autre autorisation pour pouvoir accéder aux terrains de l'entreprise. Par ailleurs, de nombreuses caméras sont réparties sur l'ensemble du terminal afin de pouvoir détecter toute intrusion ou activité suspecte.

Elle dispose du commandement unique de toute la manutention quai et parc du port depuis 2011, faisant de Med Europe Terminal un acteur important des activités portuaires de Marseille.
De cette façon, ses partenaires sont nombreux : Med Europe Terminal assure le déchargement de toute les lignes maritimes de la figure 1:

Figure 1 : lignes maritimes assurées par Med Europe Terminal

De par son domaine d'activité, Med Europe Terminal se doit d'être opérationnelle 7 jours sur 7 et lorsque des bateaux sont en attente de déchargement, ses horaires d'activités s'étendent de 6h15 jusqu'à 3h30 du matin.
Pour cette raison, le service informatique se doit d'être disponible à tout moment quelque soit le jour et l'heure, afin de pouvoir réagir en cas de problème sur l'infrastructure.
Pour cela, un système d'astreinte a été mis en place permettant que quelqu'un puisse réagir à tout imprévu.

L'entreprise est une **PME*** accueillant une centaine d'employés, répartis dans une multitude de postes.
Voici avec la figure 4, un organigramme complet de l'entreprise avec le service informatique entouré en rouge :

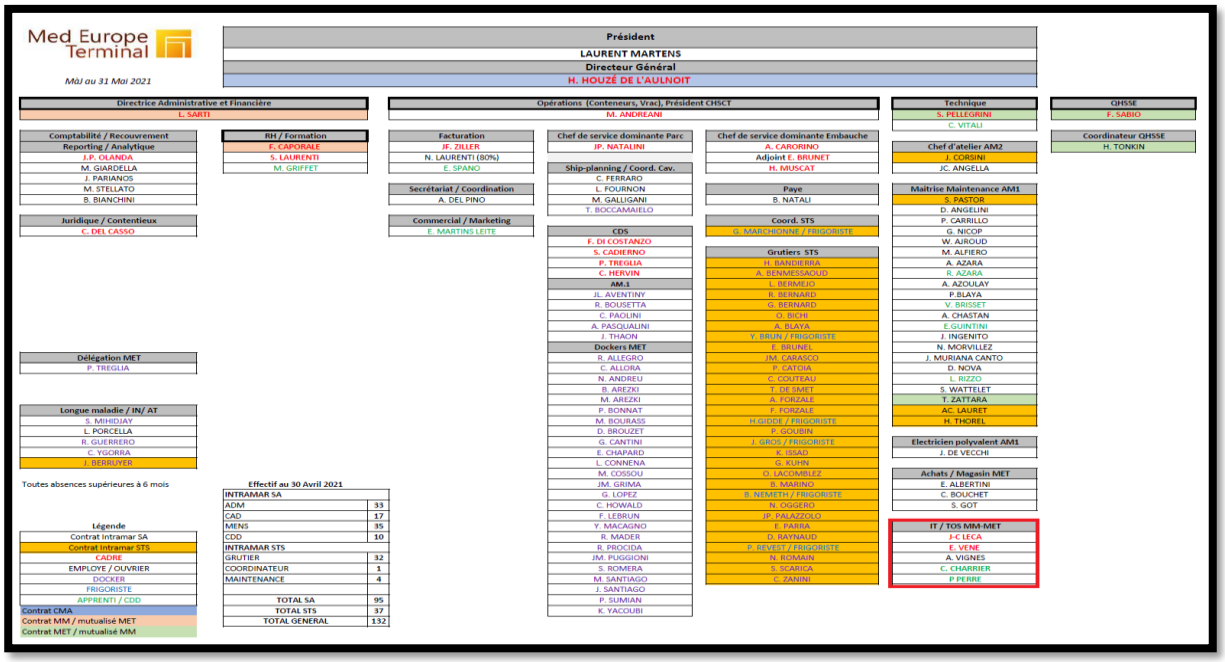

Figure 2 : Organigramme complet de Med Europe Terminal

De façon plus précise, ci-dessous est présenté sur la figure 3 l'organigramme décrivant l'organisation au sein du service informatique de Med Europe Terminal :

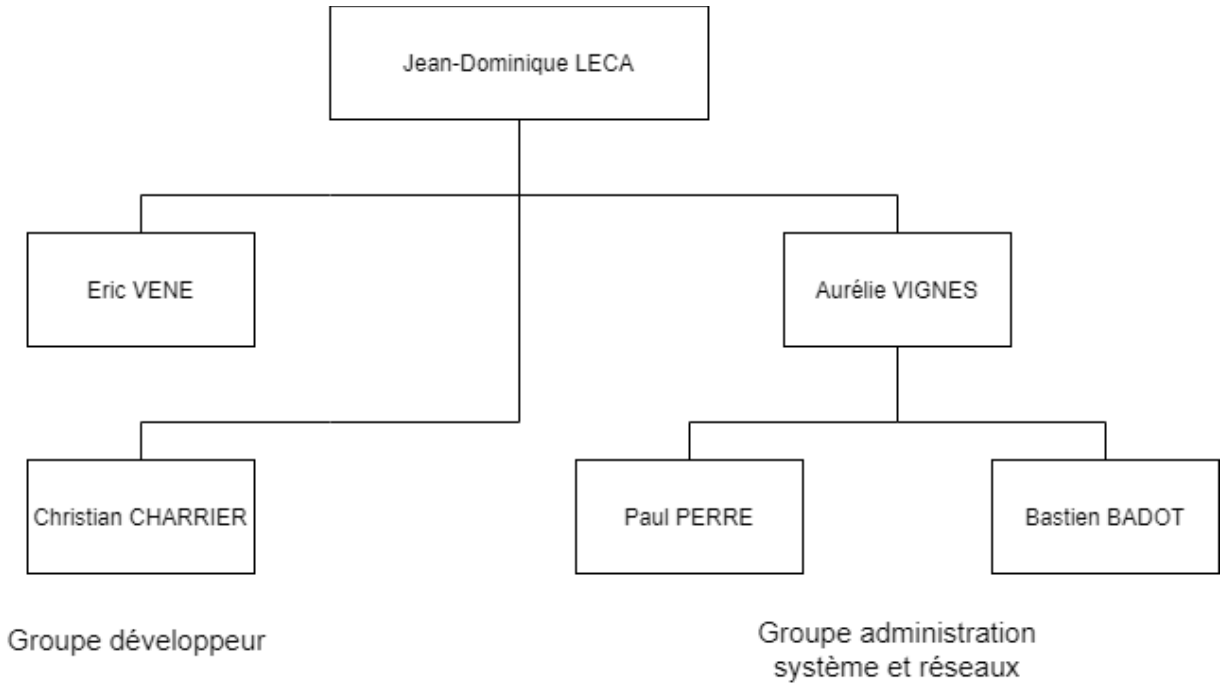

Figure 3 : Organigramme du service informatique de Med Europe Terminal

On peut voir sur la figure 3 que le service informatique est scindé en 2 parties : une spécialisée dans le développement de logiciel et l'autre dédiée à l'administration systèmes et réseaux.

En réalité, la partie administration est en sous-effectif : en plus de devoir maintenir et améliorer l'infrastructure existante, elle doit également s'occuper du support utilisateur qui est très chronophage notamment pour une entreprise d'une taille assez conséquente.
Par ailleurs, les utilisateurs ne sachant pas nécessairement formés aux outils de l'informatiques, le support se doit de les accompagner au mieux.

1.2 Historique

Voici l'histoire de Med Europe Terminal en quelque dates :

-1815 Jean-Baptiste Régulus Savon, portefaix, fonde une entreprise de manutention maritime

-1956 Naissance de la Société Industrielle de Trafic Maritime (Intramar).

-1983 Intramar est repris par STIM et MEDIACO

-1989 Création de MGM (CGM – STIM – MEDIACO ACCONAGE)

-1992 STIM rachetée par CGM

-1994 Accord de partenariat entre CGM, LEON VINCENT et SOCOMA

-1996 Cession des actions de CGM dans MGM à SOCOMA et LEON VINCENT

-1999 SOCOMA et LEON VINCENT (Groupe Sea Invest) cèdent 20% de leurs actions à EGIS

-2000 Arrivée des Chariots Cavaliers

-2001 EGIS PORTS devient actionnaire de MGM à 51%

-2003 Cession d'EGIS PORTS à PORTSYNERGY France (50% DP World / 50 % Terminal Link)

-Juin 2010 Lancement Plan de relance d'Intramar SA

-Sept. 2010 Changement de nom pour l'activité Containers d'Intramar : Med Europe Terminal

-2011 Transfert des activités **RoRo*** vers le groupe Roro Marseille. Intramar se concentre totalement sur son activité conteneurs & vrac. Le Groupe CMA CGM via CMA-T detient 100% du capital d'INTRAMAR SA.

1.3 Contexte

Dans la nuit du vendredi 13 Mars 2020, la ville de Marseille a été victime d'une terrible cyberattaque qui a mis toutes ses infrastructures à l'arrêt : 1300 serveurs et 6000 ordinateurs inutilisables, c'est l'étendue des dégâts qu'elle a subie.

Cette même nuit, Med Europe Terminal a été victime de cette attaque qui a paralysé son infrastructure en moins de quelques heures.
Une grande partie des données de l'entreprise s'est faite **encrypter*** par les attaquants, une rançon d'un montant d'environ 50 000 € était exigée pour se les voir restituer.

Med Europe Terminal venait de subir une cyberattaque de type **ransomware***,
L'attaquant avait utilisé la technique dite du **phishing*** par mail piégé qui a constitué un avant-poste pour infecter tout le reste du réseau, très probablement en **crackant*** les mots de passe administrateur domaine dont la politique de sécurité était extrêmement faible à cette période.

Cette attaque avait été minutieusement préparée par les attaquants, le moment de l'attaque a été choisie pour avoir lieu quelques jours avant le premier confinement et les élections municipales.
Cependant, grâce à la réactivité du service **IT*(Information technologie)** et à une architecture réseau bien construite, les opérations sur le port n'ont pas été impactées par cette attaque, évitant ainsi une catastrophe qui aurait pu provoquer des millions d'euros de pertes.
Med Europe Terminal a refusé de payer la rançon, l'immense majorité des données de l'entreprise ont été perdues, seules 10% ont pu être récupérées grâce aux ordinateurs éteints lors de l'attaque.
De plus, toute l'infrastructure touchée a été déconnectée du réseau et mise en quarantaine afin d'éviter une résurgence du virus.

Lors de mon arrivé, les cicatrices de cette attaque étaient encore profondes, de nombreuses machines n'étaient toujours pas remises en production et la peur d'une nouvelle attaque était encore vive dans l'esprit des employés du service.

2 Présentation du sujet du stage

2.1 Objectif du stage

Afin d'éviter une nouvelle attaque, il m'a été demandé de mettre en place un système permettant de détecter les tentatives d'intrusions d'acteurs malveillants.
Pour cela ma mission consistait à déployer un IDS dans le réseau existant, en respectant quelques contraintes :

- il fallait faire en sorte que le système n'ait pas d'effet négatif sur le réseau et ne ralentisse pas le débit des utilisateurs;
- la machine qui allait héberger l'IDS devait être virtualisée sur un **Hyperviseur***,
- le système devait alerter automatiquement les administrateurs par mail.

Tout cela implique que :

- Il fallait être prudent lors des modifications de l'architecture réseau;
- La machine virtualisé ne devait pas empiéter sur les autres machines de l'Hyperviseur;
- Le système devait utiliser un système de traitement de donnée capable d'automatiser l'envois de mail en cas de détection.

Cette mission ayant des contraintes fortes, elle requière des compétences dans le domaine de l'administration systèmes Windows, Linux, l'administration réseaux ainsi que le traitement de données.
Cette mission nécessite également une coopération entre les membres du service car il faut faire en sorte de ne pas empiéter sur le travail des autres, tout en pouvant avancer dans sa mission.

2.2 NIDS

Les IDS peuvent être classés sous différentes catégories : il en existe deux principales , les **NIDS*(Network intrusion détection system)** et les **HIDS*(Host intrusion détection system).**
Dans le cas des NIDS, on place le système dans un point stratégique du réseau et on analyse les données réseau de l'infrastructure à la recherche de requêtes suspectes.
Dans le cas d'un HIDS, on place le système directement sur une machine et on analyse ses processus et actions afin de détecter un comportement suspect, comme par exemple une élévation de privilège (passage d'utilisateur normal à administrateur par des moyens anormaux).

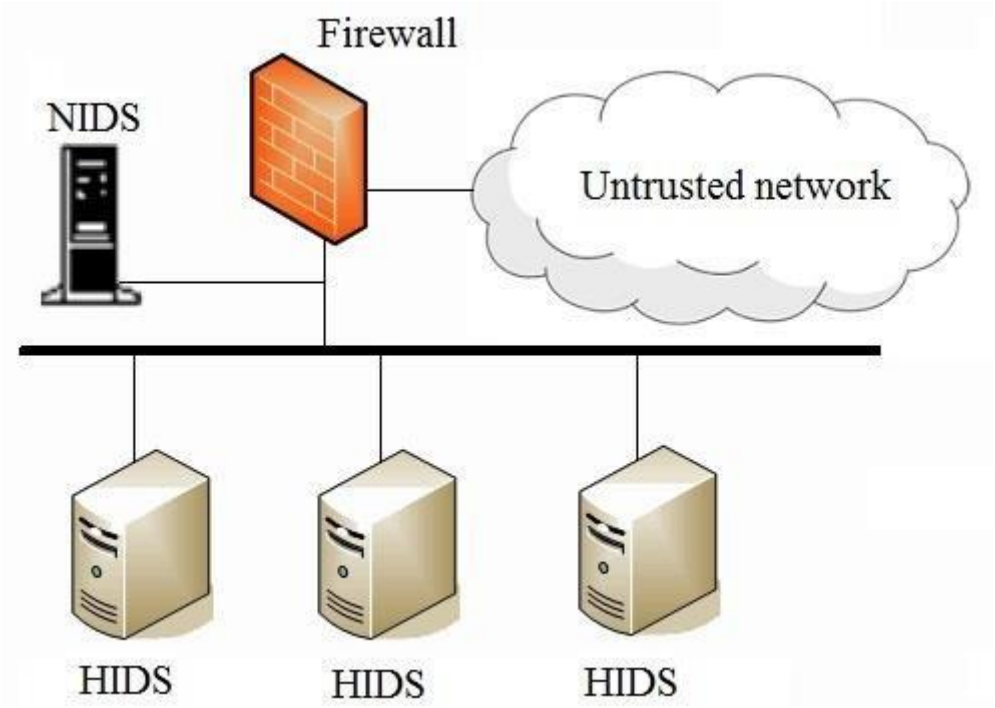

Figure 4 Implémentation d'HIDS et d'un IDS

Dans la figure 4, on peut clairement voir les différentes utilisations des IDS ainsi que leur complémentarité.

Il est intéressant de noter que leur fonctionnement est très similaire, ils utilisent tous les deux une approche de détections par signatures.
Cette approche consiste à utiliser des bibliothèques de description des attaques appelées signatures que l'on comparera aux informations récoltés pour déterminer leur nature : ainsi, chaque évènement est analysé et comparé à la table des signatures.
Il est donc crucial de garder à jour cette table afin d'être capable de détecter les dernières attaques en date.
Dans le cadre de mon stage, Med Europe Terminal avait déjà implémenté un système HIDS sur tous ses postes et serveurs en utilisant la solution Trend Apex One.
Ce faisant, il me fallait déployer un NIDS afin de compléter leur solution IDS.

2.3 Snort

Il a été décidé que j'utiliserai la solution Snort afin d'implémenter mon NIDS.
En effet, Snort est un NIDS gratuit et open source avec une communauté de développeurs très actifs. De plus, sa légèreté et son efficacité à traiter des données en temps réel, ainsi que ses documentations nombreuses, font de lui une solution de choix pour l'implémentation d'un NIDS dans le réseau d'une PME.
Il faut également noter que Cisco participe au développement du projet Snort et propose des bibliothèques de signatures, faisant de Snort une solution fiable et approuvée par les experts.

figure 5 logo de snort

Snort est complètement gratuit, cependant, il est possible de payer 30$ afin de bénéficier des mises à jours des tables de signatures un mois à l'avance.
Il existe également une option à 400$ pour bénéficier de tables de signature à la pointe écrites par les équipes de Cisco pour des clients aux besoins de sécurité très élevée.

3 Présentation des travaux réalisés

3.1 Préparation

La première étape de ma mission a consisté à faire de la recherche d'informations pour me préparer et bien comprendre les subtilités de la mise place Snort.
Pour cela, j'ai commencé à lire la documentation officielle de Snort et les guides d'installations proposés.
Quand je pensais être prêt, j'ai mis en place un laboratoire avec plusieurs **machines virtuelles*** afin de m'entrainer à installer le système puis à le tester dans un environnement contrôlé.

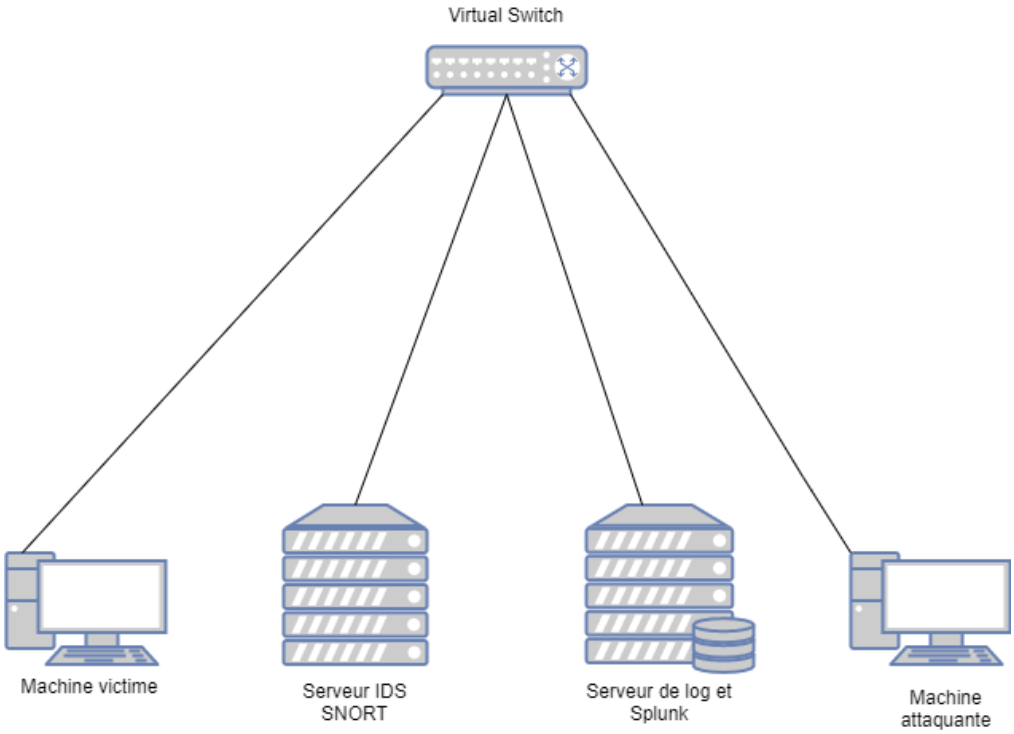

Figure 6 laboratoire de test

Ce laboratoire est composé de 4 éléments comme on peut le voir dans la figure 6.

Toutes les machines de ce laboratoire tournent sous Debian, une distribution Linux.

Tout d'abord, on trouve bien évidemment le serveur IDS qui récolte les requêtes de toutes les autres machines et les analyse pour y détecter des attaques potentielles.

Ensuite, il y a le serveur de log, où seront stockées toutes les alertes détectées par SNORT.
Il est possible d'installer le serveur de log sur la même machine que le serveur IDS, cependant pour une optimisation de l'organisation, il est plus intéressant de les séparer : le serveur de log pouvant être utilisé pour plusieurs applications, et , sur le serveur de log se trouve un **SIEM*(security information and event management)**, Splunk.
Son objectif consiste à récupérer les alertes produites par Snort et de traiter toutes ses données pour mettre en place une interface utilisateur claire, permettant en quelque secondes d'avoir une bonne idée de l'état du réseau.

Pour finir il y a 2 ordinateurs supplémentaires qui me permettront de simuler une attaque dans le réseau de telle sorte à ce que la machine attaquante tente de prendre le contrôle de la machine victime.

3.2 Installation de Snort

Une des premières difficultés que j'ai rencontrées avec l'installation de Snort, c'est le choix de sa version : en effet, un mois avant le début de mon stage Snort3 était disponible au public, et la documentation sur cette version était encore très limitée, de plus les **gestionnaires de paquets*** ne possédait pas cette version dans leur liste.
De cette façon, au début de mes tentatives, j'ai installé une vieille version de Snort en utilisant les gestionnaires de paquets.
Quand je m'en suis aperçu, j'ai décidé de tout recommencer afin de m'assurer que le système que j'allais mettre en place soit à jour et reste supporté par l'équipe derrière Snort.

Ainsi, l'installation de Snort3 ne s'est pas faite avec les gestionnaires de paquets mais directement avec les codes sources fournis par les développeurs.
Tout d'abord pour installer Snort, il faut s'assurer que son système d'exploitation et ses **librairies*** soient à jour.

Pour cela, on peut utiliser le gestionnaire de paquet avec cette commande :

```
sudo apt-get install -y build-essential autotools-dev libdumbnet-dev libluajit-5.1-dev
libpcap-dev zlib1g-dev pkg-config libhwloc-dev cmake liblzma-dev openssl libssl-dev cpputest
libsqlite3-dev libtool uuid-dev git autoconf bison flex libcmocka-dev libnetfilter-queue-dev
libunwind-dev libmnl-dev ethtool
```

Pour continuer, Snort3 a besoin de librairies particulière qu'il faut installer à la main.
Pour cela, il faut commencer par télécharger le code source de la librairie, le décompresser puis le **compiler*** de telle sorte à ce que le programme soit lisible par l'ordinateur.
Voici un exemple de la marche à suivre :

```
wget https://github.com/rurban/safeclib/releases/download/v02092020/libsafec-02092020.tar.gz
tar -xzvf libsafec-02092020.tar.gz
cd libsafec-02092020.0-g6d921f
./configure
make
make install
```

A noter que la compilation des librairies est parfois assez longue, allant jusqu'à une trentaine de minutes selon leur complexité.

Lors de cette étape, j'ai rencontré venant de certaines librairies dépendantes d'autres librairies pour fonctionner.
Elles n'étaient pas installées sur ma machine, lors de l'échec de l'installation, les erreurs affichées n'étaient pas explicites et il était très difficile de savoir ce qui posait problème, ainsi il n'était pas rare que je passe plusieurs heures à chercher une solution sur des **forums*.**

Une fois toutes les librairies installées, il reste à installer Snort3, à l'identique des librairies, il faut également compiler le code source de Snort3 fourni par les développeurs et l'installer.

Il est important de noter qu'il existe plusieurs versions de Snort3 et que les dernières versions du logiciel peuvent poser certains problèmes de par leur nouveauté.

Une fois l'installation terminée, on peut vérifier le bon fonctionnement grâce à la commande :

```
Snort -V
```

Qui renvoi ce qu'on peut voir sur la figure 7

```
   ,,_     -*> Snort++ <*-
  o"  )~   Version 3.1.0.0
   ''''    By Martin Roesch & The Snort Team
           http://snort.org/contact#team
           Copyright (C) 2014-2020 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using DAQ version 3.0.0
           Using LuaJIT version 2.1.0-beta3
           Using OpenSSL 1.1.1f  31 Mar 2020
           Using libpcap version 1.9.1 (with TPACKET_V3)
           Using PCRE version 8.44 2020-02-12
           Using ZLIB version 1.2.11
           Using FlatBuffers 1.12.0
           Using Hyperscan version 5.3.0 2021-01-19
           Using LZMA version 5.2.4
```

Figure 7 installation réussi de Snort3

A présent que snort est installé, il faut le configurer pour le rendre opérationnel :

Pour commencer, il est à noter que Snort fonctionne en utilisant les interfaces réseaux de sa machine en mode promiscuité.
Cela signifie que l'interface est autorisée à écouter du trafic qui ne lui est pas destiné.
Ainsi il faut activer ce mode de façon permanente, pour cela il existe une commande qui active le mode promiscuité :

```
ip link set eth0 promisc on
```

De plus il faut désactiver certaines propriétés de la carte réseau :

- le mode GRO (Generic receive offload)
- le mode LRO (large receive offload)

qui sont des techniques d'optimisation de traitements de paquets incompatibles avec Snort.
On peut utiliser ces commandes pour les désactiver :

```
ethtool -K ens3 lro off
ethtool -K ens3 gro off
```

Pour éviter toute perte de configuration au redémarrage, on va indiquer au serveur d'appliquer cette commande à chaque démarrage.
Pour cela, il faut créer un script dans le répertoire « /lib/systemd/system/ » qui sera exécuté au démarrage de la machine.

Afin de personnaliser Snort, il faut modifier le fichier de configuration « snort.lua »

On remarquera son extension en .lua, qui est un langage que je n'avais jamais rencontré auparavant. La plupart des fichiers de configuration finissent en .conf, mais Snort3 a été écrit dans le langage de programmation Lua pour le rendre le plus efficace possible, son extension prend donc le nom de ce langage.

Dans le fichier de configuration, il y a de nombreuses options à modifier, je ne vais donc citer que les plus importantes :

- la variable HOME_NET où : il faut rentrer l'adresse IP de son réseau de telle sorte à ce que les règles sachent quel est le réseau à protéger.

- la variable blacklist permet de donner une liste d'adresse IP qui déclencheront automatiquement une alerte.

le site de Snort propose une liste d'adresses IP connues pour être malveillantes mise à jour régulièrement.

- la partie filter, cette option est très importante car elle permet d'ignorer certaines règles connues pour être des faux positifs. En effet au vu de la quantité astronomique de requêtes traitées par Snort, de nombreuses faux positifs ont lieu et c'est le travail du SIEM de faire le tri. Cependant, l'espace disque du server log n'étant pas illimité, il est important de ne pas sauvegarder les alertes inutiles.

Une fois la configuration de Snort terminé, il faut faire en sorte qu'il se lance automatiquement au démarrage.
Le plus simple consiste à créer un service dédié à snort, qui sera exécuté au démarrage comme nous l'avons fait auparavant avec ethtool.

3.3 Mises à jour des règles avec PulledPork

Comme énoncé dans la partie 2.2, un IDS n'est efficace que si sa base de signature est mise à jour régulièrement. Ainsi les dernières attaques en date sont détectables, grâce au site officiel de Snort qui met à disposition des ensembles de signatures téléchargeables.
Pour automatiser ce processus de mise à jour, il existe un script maintenu par les équipes de Cisco : « PulledPork ».
Ce script, écrit en Perl, permet de récupérer directement les règles sur le site de Snort et les rajouter à notre fichier de règles. Il faut noter qu'il est possible de créer son propre script afin d'effectuer cette tâche.
Cependant, le fait que PulledPork soit utilisé par une grande quantité d'utilisateurs de Snort et supporté par Cisco font de « PulledPork » la solution par défaut pour l'automatisation des mises à jours.

Pour son installation, Snort nécessite d'avoir installé Perl sur le système. Ainsi on peut utiliser le gestionnaire de paquet pour l'installer :

```
apt install -y libcrypt-ssleay-perl liblwp-useragent-determined-perl
```

Par la suite, il faut se procurer les fichiers d'installation de PulledPork disponibles sur le site officiel. Ces fichiers étant déjà compilés, il n'y a pas besoin de les recompiler pour installer PulledPork : il suffit simplement de copier les fichiers dans leur répertoire respectif et PulledPork sera installé sur le système :

```
cp pulledpork.pl /usr/local/bin
chmod +x /usr/local/bin/pulledpork.pl
mkdir /usr/local/etc/pulledpork
cp etc/*.conf /usr/local/etc/pulledpork
```

Maintenant que PulledPork est installé, il faut le configurer pour lui spécifier où se trouvent les règles à télécharger, où il doit les installer et leur donner un code utilisateur disponible sur son profil sur le site de Snort.

```
# ..u can specify one or as many rule_urls as you like, they
# must appear as http://what.site.com/|rulesfile.tar.gz|1234567.  You can specify
# each on an individual line, or you can specify them in a , separated list
# i.e. rule_url=http://x.y.z/|a.tar.gz|123,http://z.y.z/|b.tar.gz|456
# note that the url, rule file, and oinkcode itself are separated by a pipe |
# i.e. url|tarball|123456789,
rule_url=https://www.snort.org/downloads/registered|snortrules-snapshot-3130.tar.gz| "Oinkcode"
# NEW Community ruleset:
#rule_url=https://snort.org/downloads/community/|snort3-community-rules.tar.gz|Community
# NEW For IP Block lists! Note the format is urltofile|IPBLOCKLIST|<oinkcode>
# This format MUST be followed to let pulledpork know that this is a blocklist
rule_url=https://snort.org/downloads/ip-block-list|IPBLOCKLIST|open
# THE FOLLOWING URL is for emergingthreats downloads, note the tarball name change!
# and open-nogpl, to avoid conflicts.
#rule_url=https://rules.emergingthreats.net/|emerging.rules.tar.gz|open-nogpl
# THE FOLLOWING URL is for etpro downloads, note the tarball name change!
# and the et oinkcode requirement!
```

Figure 8 fichier de configuration de PulledPork

A noter que le Oinkcode présent sur la figure 8 est le code utilisateur spécifié sur le site de Snort.
On peut également voir que PulledPork télécharge également une IPBLOCKLIST qui correspond aux adresses IP connues comme étant malveillantes et qui déclencheront donc automatiquement une alerte quand snort les rencontrera.

Une fois que tout est configuré, on peut utiliser la commande :

```
/usr/local/bin/pulledpork.pl -c /usr/local/etc/pulledpork/pulledpork.conf -l /mnt/snortlog/ -
P -E -H SIGHUP
```

Cette commande permet non seulement de mettre à jour le fichier de règles de Snort mais elle permet aussi de redémarrer automatiquement Snort de telle sorte à ce que les nouvelles règles soient bien prises en compte.

Il est possible de manuellement lancer cette commande de temps en temps.
Cependant on préférera lancer automatiquement cette commande journalièrement ou hebdomadairement, de telle sorte à ce que notre système soit à jour le plus rapidement possible.
Pour cela il faut utiliser le programme natif de Linux : Cron qui permet d'exécuter une commande automatiquement, selon un cycle défini.
Ainsi, j'ai décidé d'exécuter la commande tous les jours en configurant Cron comme spécifié dans la figure 9 :

```
# m h  dom mon dow   command
00 16 * * * /usr/local/bin/pulledpork.pl -c /usr/local/etc/pulledpork/pulledpork.conf -l -P -E -T -H SIGHUP
```

Figure 9: fichier de configuration de cron

Ici, la commande de mises à jour des règles est exécutée tous les jours à 16h00, l'heure où la commande est effectuée n'a que peu d'importance car PulledPork étant très rapide on ne perd que quelques secondes, voire millisecondes, de détection. Il est alors hautement improbable que Snort manque une attaque pendant ce laps de temps.

3.4 Analyse des données avec Splunk

En production, Snort sauvegarde énormément d'alertes : environ 7000 détections par heure sont enregistrées. Il est à noter qu'un pré filtrage a déjà été effectué dans les fichiers de configuration de Snort, il est donc impossible pour un humain d'analyser cette quantité de données brutes.
Il faut alors utiliser ce qu'on appelle un SIEM pour analyser ces données et en tirer des conclusions quant à l'état du réseau.
Ainsi, j'ai décidé d'utiliser Splunk, un logiciel propriétaire conseillé dans le guide d'installation officiel de Snort afin de mettre en place un SIEM.

Son installation est très simple : il suffit de se procurer les paquets d'installation Debian sur leur site officiel puis de l'installer directement avec ces commandes :

```
dpkg -i splunk-8.*.deb
chown -R splunk:splunk /opt/splunk
/opt/splunk/bin/splunk start --answer-yes --accept-license
```

Maintenant que Splunk est opérationnel, on y accède en utilisant un navigateur WEB et en se connectant sur le server, avec le port 8000 :

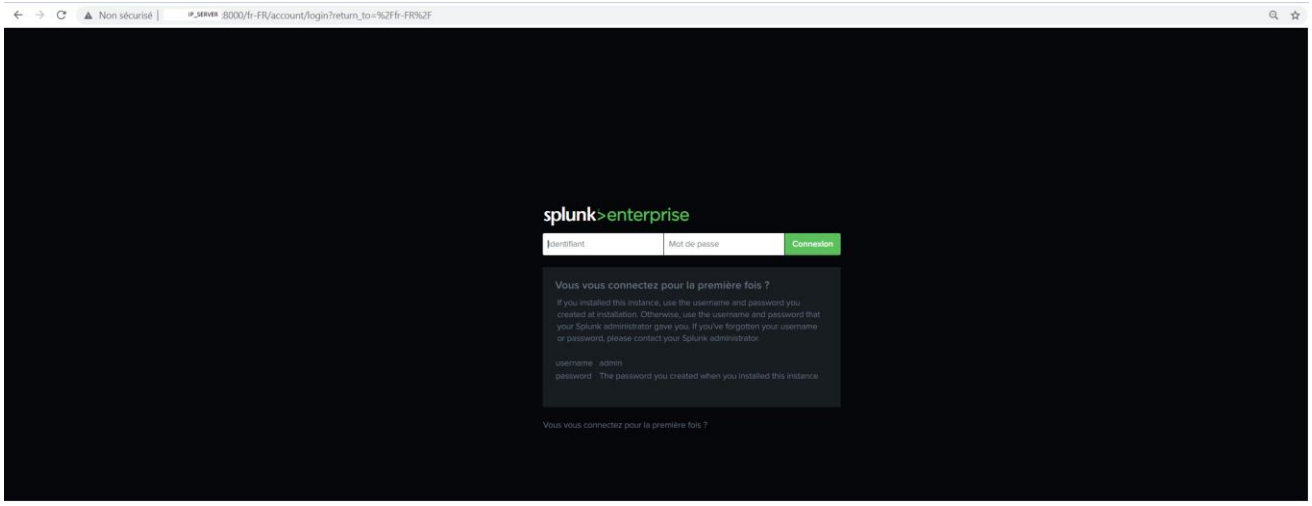

Figure 10: Affichage de Splunk

Avec la version d'essai, on ne dispose que d'un seul utilisateur qui est administrateur, Cependant en acquérant la licence il est possible d'en avoir plusieurs, si toutefois la version d'essai venais à expirer, le site serait accessible sans aucune authentification, ce qui représente une énorme faille de sécurité.

Ainsi je suis rentré en contact avec les experts de Splunk pour discuter du prix de la solution qui serait la plus adaptée à notre utilisation.

Splunk est un outil complet possédant une multitude de possibilités d'application dans de nombreux domaines.
Toutefois, dans le cadre de mon projet, je n'utilise qu'une fraction de ses capacités d'analyses.

Splunk possède un onglet recherche et analyse permettant avec une certaine syntaxe de récupérer très précisément les informations nécessaires à notre analyse, par exemple :

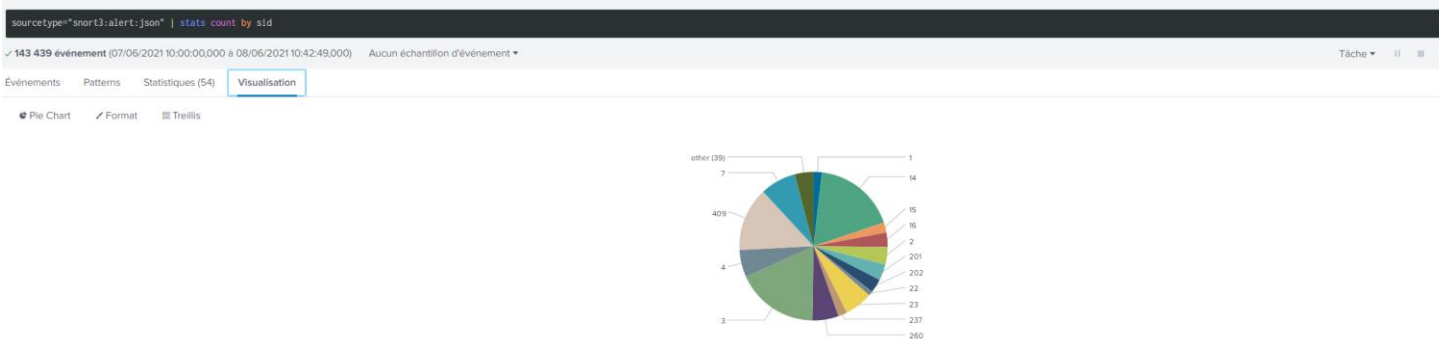

Figure 11 : Graphique des règles les plus détectés

Comme on peut le voir dans la figure 11, avec la commande :

```
sourcetype="snort3:alert:json" | stats count by sid
```

On peut compter les alertes les plus répandues des dernières 24h et l'afficher sous la forme d'un graphique camembert.

« sourcetype="snort3:alert.json" » désigne les alertes stockées dans le format JSON par SNORT puis « | » permet d'envoyer le résultat de la première commande vers une deuxième comme sur Linux.
Enfin « stats count by sid » permet de compter le nombre de sid qui correspond à chaque requête. Un SID est un numéro d'identification qui permet de retrouver quelle règle de snort a été déclenchée.

Ainsi, on peut également choisir la fenêtre temporelle de l'analyse avec son formatage pour l'afficher par exemple comme un graphique ou bien sous une forme de tableau.

Bien maitriser ces requêtes est crucial pour proprement analyser les informations récoltées par Snort. J'ai alors passé un certain temps à me renseigner sur la syntaxe de Splunk pour maitriser correctement cet outil.

A partir de ces requêtes, il est possible de configurer des alertes qui se déclencheront et effectueront une action quand certaines conditions seront remplies. Par exemple, on peut envoyer un mail à un administrateur si on remarque un nombre de détection anormalement élevé.

3.5 Test du système

Afin de m'assurer que mes installations étaient fonctionnelles et que mon système était robuste, j'ai effectué un certain nombre de tests en simulant une cyberattaque pour voir comment l'IDS allait réagir.
La machine attaquante de la figure 6 a commencé à scanner la machine victime puis, pour pousser la simulation le plus loin possible, j'ai installé un serveur WEB vulnérable , Juice Shop , sur la machine victime et j'ai effectué une grande quantité de cyberattaques depuis la machine attaquante.

Pour cela j'ai utilisé 3 principaux outils illustrés dans la figure 12 :

Figure12 : outils utilisés pendant l'attaque

- Nmap : un outil permettant de scanner les ports ouverts d'une cible afin de découvrir les services correspondants et trouver potentiellement une porte d'entrée pour des attaques plus destructrices.

- Nikto : Un scanneur de site web pour découvrir les failles de sécurités les plus communes et ainsi d'avoir un point d'attaque facilement exploitable.

- Sqlmap : Un outil pour tester des **injections SQL*** sur un site web afin d'exfiltrer les informations de la base de données qui lui est associée.

Après avoir fait tourner ces 3 outils, j'ai pu étudier les résultats que l'on pouvait observer dans Splunk comme on peut le voir dans la figure 13:

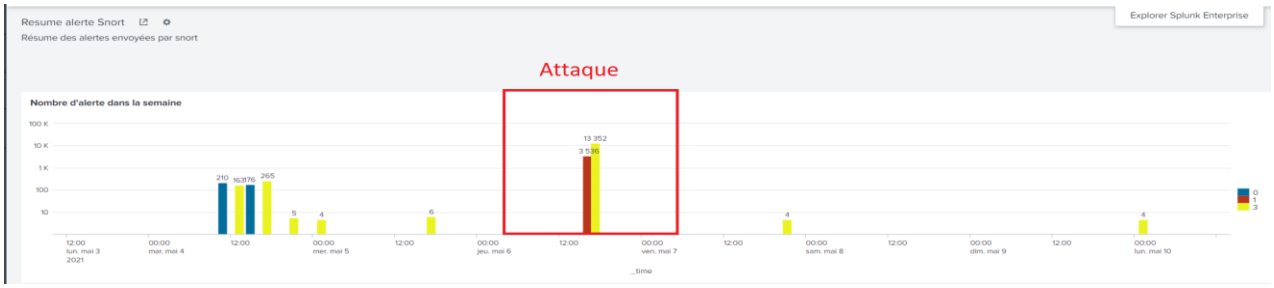

Figure13 : résultat des détections lors de la simulation d'attaque

Les résultats étaient très clairs : l'attaque avait bien été détectée et la gravité et la quantité des détections permettaient sans aucune forme de doute de conclure que le système avait bien été attaqué.

Dans la figure 13, l'échelle est sous format logarithmique. On pourrait visuellement croire que le nombre de détections n'est pas si grand mais les chiffres affichés au-dessus des barres permettent de voir la différence.
J'ai décidé d'utiliser ce format logarithmique afin qu'il soit possible de voir les petits nombres de détections. En effet , parfois certaines attaques ne laissent pas un nombre énorme d'alertes comme les injections SQL mais il faut tout de même pouvoir les observer.
Suite à ces résultats j'ai conclu que mon système remplissait sa fonction et j'étais prêt à l'installer en production.

3.6 Mise en production

Mon système était fonctionnel dans un laboratoire contrôlé, cependant il fallait maintenant l'installer dans une infrastructure réelle.

On m'a fixé quelques contraintes à respecter :
Mon système devait fonctionner dans un serveur de virtualisation HyperV que je devais installer à partir d'un ancien serveur IBM X3650m3 ayant été piraté lors de l'attaque informatique.

Figure14 :Un server IBM X3650M3

Cette étape a été source de nombreuses difficultés pour moi. En effet, les machines n'ayant pas été testées depuis leur mise hors service, de nombreux serveurs était défaillants.

Le premier serveur qui m'a été fourni semblait posséder des défauts à sa carte vidéo. De ce fait, il ne produisait aucun affichage et était donc inutilisable.

Le deuxième serveur que l'on m'a fourni avait des problèmes de ventilateur le rendant instable et provoquant des arrêts intempestifs du système le rendant également inutilisable.

Finalement, le troisième serveur que l'on m'a fourni fonctionnait. Toutefois, il ne possédait pas suffisamment d'espace où brancher des disques. J'ai dû démonter un des serveurs défaillants pour y récupérer son **contrôleur RAID*** et le mettre en place sur le serveur fonctionnel.
Cette étape était délicate puisque les composants des serveurs sont parfois difficilement accessibles.

Une fois le matériel prêt, j'ai mis à jour le **firmware*** du système pour qu'il soit prêt à accueillir mon installation. Pour ce faire, j'ai utilisé le logiciel Xclarity fourni par les constructeurs permettant de graver un installateur de mises à jour sur une clef USB **bootable*** afin d'installer les mises à jour automatiquement.
Ces mises à jours ont été une source de problèmes pour moi car, de par l'ancienneté du server, les clefs USB 3.0 n'étaient pas utilisables, il m'a donc fallu passer du temps à rechercher dans la documentation d'IBM et leur forums de discussions la source de mes problèmes.

Une fois le système à jour, il a été nécessaire de créer un nouveau profil dans le contrôleur RAID afin de mettre en place une architecture de disques respectant les standards et les besoins de ma solution.

Ainsi, j'ai opté pour utiliser un RAID1 d'une capacité de 66Go qui allait être accueillir le système d'exploitation de telle sorte à ce que si un des disques du RAID1 venait à lâcher, un autre pourrait prendre le relais immédiatement, empêchant toute panne du système d'exploitation.
Concernant le stockage des données, j'ai utilisé un RAID5 d'environ 3To pour accueillir les machines virtuelles qui allaient tourner sur l'Hyperviseur.

J’ai également réservé un disque en tant que **HOTSPARE*** global, de telle sorte à ce que si l’un des disques du système venait à tomber, il prendrait automatiquement le relais en le remplaçant.

Une fois tout terminé, j’ai installé Windows Server 2019 sur la machine, l’ai mis à jour et ai rajouté les fonctionnalités nécessaires pour le bon fonctionnement d’Hyper-V.

Ainsi, tout semblait prêt pour virtualiser mon système d’IDS.
Cependant, il restait encore quelques problèmes à résoudre. En effet , il fallait encore acheminer tout le trafic du réseau vers l’hôte qui hébergera l’IDS: pour cela, il a été décidé que j’utiliserai SPAN pour copier tout le trafic du réseau vers mon IDS.

Il faut cependant noter plusieurs choses : pour fonctionner, HyperV utilise un système de switch virtuel qui permet de convertir une des interfaces physiques de la machine en un switch relié aux machines virtuelles comme on peut le voir dans la figure 15 :

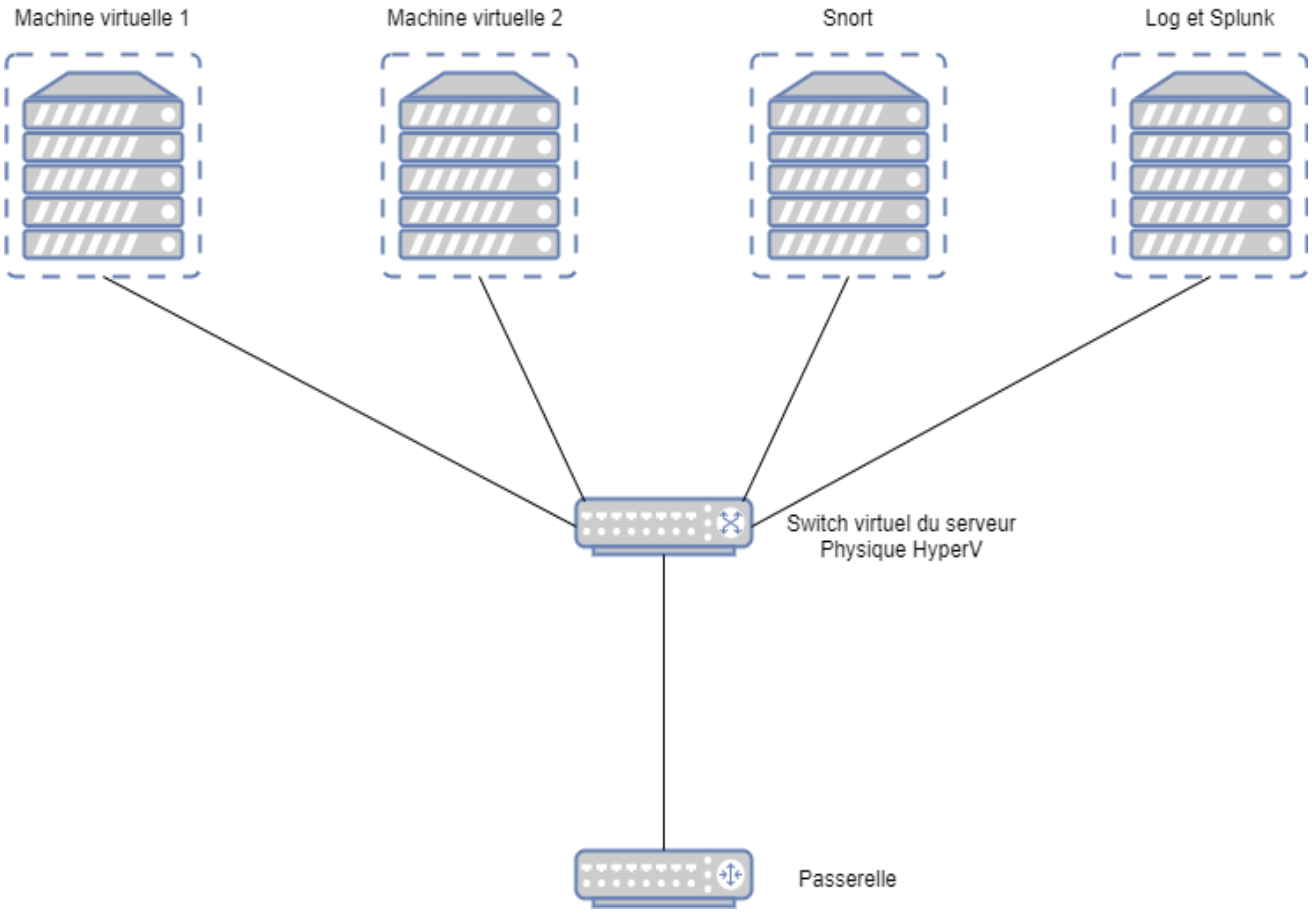

Figure15 : schéma de fonctionnement d’un switch virtuel

Cette architecture m’a posé un problème car, si une liaison est utilisée par SPAN elle ne peut pas être utilisé pour accéder au reste du réseau, et si on appliquait SPAN dans la figure 15, on couperait la connexion entre le switch virtuel et la passerelle rendant toute connexion impossible depuis le réseau.

J’ai alors développé une autre architecture réseau qui rendrait possible l’implémentation de mon IDS dans un HyperV. A cet effet, j’ai utilisé 2 switchs virtuels, dont un dédié à Snort comme on peut le voir dans la figure 16 :

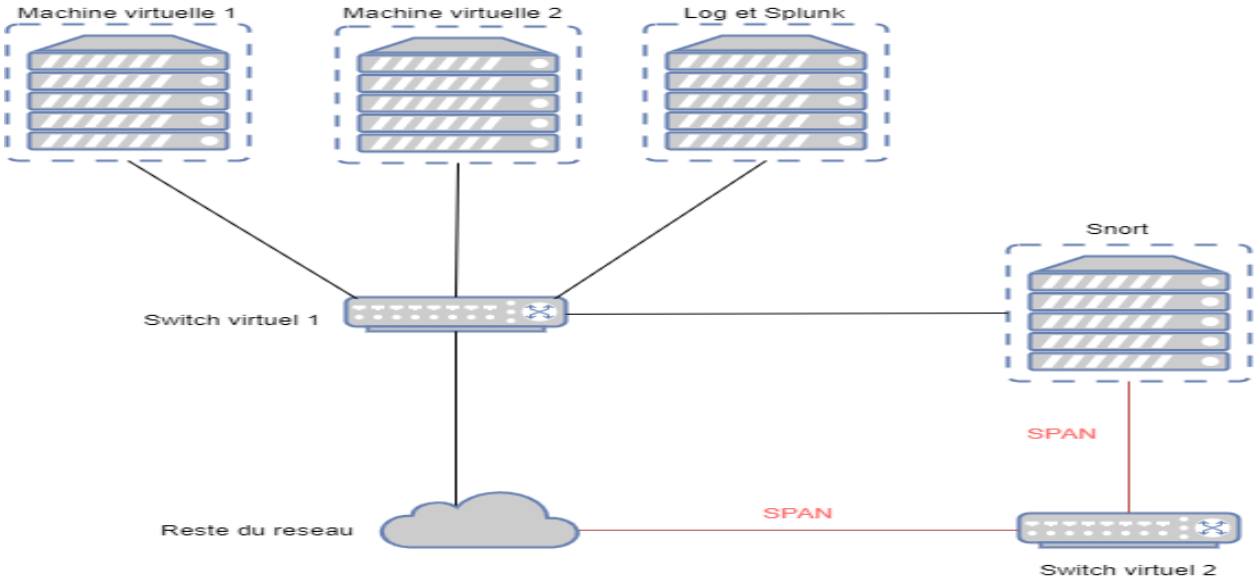

Figure16 : mon architecture réseau

Cette architecture permet d’accéder à Snort par le Switch Virtuel 1, tout en laissant la liaison SPAN passer par le Switch virtuel 2.

De cette façon, il est toujours possible pour Snort d’accéder au server de Log et Splunk par le switch virtuel 1.

Il reste cependant un problème à régler : il faut savoir qu’un switch n’envoie à une machine que les données qui lui sont destinées. Or, Snort devait recevoir toutes les données de tout le réseau et, avec un switch virtuel il ne recevrait aucune donnée à analyser.

Heureusement, HyperV possède une fonction de **port mirroring***, qui permet de désigner une source de trafic et une destination.
Ainsi j’ai désigné comme source de trafic le port externe du Switch Virtuel 2 relié au reste du réseau, puis, j’ai désigné comme destination le port interne relié à Snort.
De cette façon, tout le trafic acheminé par Span a pu être transmis à Snort qui , de son côté, n’a plus qu’à analyser le trafic puis stocker les résultats sur le serveur de log où Splunk qui à son tour analysera les données enregistrées.

A partir de ce moment, l’IDS était fonctionnel et analysait tous les paquets du réseau.

Cependant, par la quantité de requêtes reçues, Snort détectait énormément de faux positifs, et écrivait environ 1Go de Log par jour, ce qui allait saturer l’espace disque du serveur. J’ai rapidement dû filtrer dans les fichiers de configurations de Snort environ 90% des signatures, qui n’apportaient aucune information.

Le système était certes opérationnel, mais mon travail n’était pas terminé. En effet,
l’une des problématiques majeures pour un administrateur réseau est d’être capable d’effectuer des maintenances et de maintenir les infrastructures déjà mises en place.
Il est alors crucial que les installations et les modes de fonctionnement de tous les appareils soient documentés.
Pour cette raison, j’ai rédigé une documentation décrivant en détail toutes les procédures nécessaires à l’installation et la maintenance de mon système IDS. J’ai de plus rédigé des guides d’utilisations pour Splunk.

3.7 Résultats obtenus

Aujourd'hui, Snort a tourné pendant plusieurs semaines , il a détecté une centaine d'attaques provenant d'acteurs extérieurs, que nous avons pu confirmer comme étant malveillants, La plupart provenant de pays d'Asie comme l'Inde ou bien la Chine, visible dans la figure 17, les administrateurs du réseau vont après vérification bannir toutes les IP commettant ces actes malveillants rendant le réseau un peu plus résistant face aux cybermenaces.

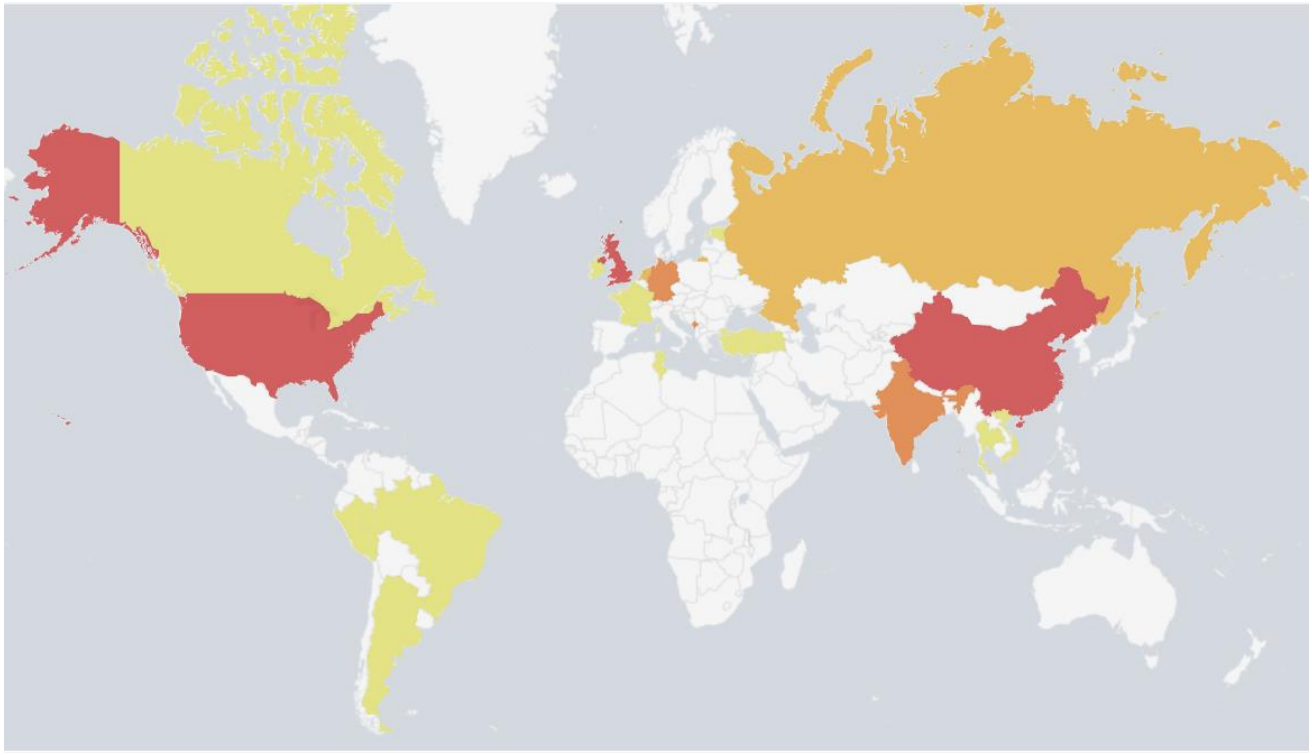

Figure17 : Carte de la provenance des attaques

Ces résultats ont été pour moi assez surprenants, je ne m'attendais pas à ce que l'entreprise soit victime d'autant de tentatives d'attaques.
Cependant comme la plupart de ces attaques sont automatisées et ciblent une énorme quantité de réseaux différents, on peut espérer que l'entreprise ne soit pas prise pour cible en particulier.

Snort a également servi à détecter une panne dans l'un des équipements du réseau : un matin, l'équipement en question s'est mis à émettre environ 800 000 requêtes en 30 minutes, ce qui a été détecté par Snort, Par la suite, nous avons pu cibler l'équipement problématique grâce à son IP et régler le problème.
Cette surprise a permis de mettre en avant une utilité inattendue du système IDS.

Conclusion

Pour conclure, ce stage en tant qu'administrateur systèmes et réseaux m'a été très profitable pour plusieurs raisons.

En 1ER lieu, sur le plan technique, j'ai pu réaliser ma mission et mettre en place une solution de cybersécurité IDS qui a su convaincre mon maître de stage.
Cela m'a permis d'en apprendre plus sur les différentes façons de protéger un réseau informatique et m'a aidé à mieux comprendre le fonctionnement des solutions de cybersécurité.

Puis, cette première expérience professionnelle m'a permis d'améliorer mon relationnel et ma pédagogie : j'ai pu à l'occasion aider mes camarades stagiaires et documenter en détail ma solution,
De plus cette expérience m'aura permis de mieux comprendre comment fonctionne le monde du travail en entreprise.
Bien que Med Europe Terminal opère dans un domaine assez particulier, les expériences vécues ici me seront sans aucun doute utiles dans la suite de mon parcours professionnel.

Par ailleurs ce stage qui conclut ma formation à l'IUT est déjà un atout dans la poursuite de ma scolarité.
En effet, les compétences et expériences acquises lors de ces 10 semaines me permettent d'être plus attractif auprès des entreprises que je contacte pour trouver une alternance en école d'ingénieur.

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage et qui m'ont aidé lors de la rédaction de ce rapport.

Tout d'abord, j'adresse mes remerciements à tous mes professeurs, du département de Réseaux et Télécommunications de Luminy pour m’avoir transmis les compétences nécessaires au bon déroulement de ce stage.

De plus, je tiens également à remercier vivement mon maître de stage, Mme Aurèlie Vignes, responsable adjointe du service informatique au sein de l'entreprise MedEurope Terminal, pour son accueil, le temps passé ensemble et le partage de son expertise au quotidien. Grâce aussi à sa confiance j'ai pu m'accomplir totalement dans mes missions et développer mes compétences aussi bien sociales que techniques.

Je remercie également plus généralement toute l'équipe du service informatique de MedEurope Terminal pour leur accueil, leur esprit d'équipe et en particulier Mr Paul Perré et Mr Jean-Dominique Leca, qui m'ont beaucoup aidé à comprendre la culture et les problématiques au sein du service.

Enfin, je tiens à remercier mes camarades de promotion qui m'ont conseillé et relu lors de la rédaction de ce rapport de stage.

Glossaire

Cyberattaque : Acte malveillant envers un dispositif informatique via un réseau cybernétique.
DUT : Diplôme Universitaire de Technologie.
IDS : Intrusion détection system est le diminutif anglais de système de détection d'intrusion.
PME* : Petite et moyenne entreprise
Portefaix : Celui dont le métier consiste à porter des fardeaux.
RoRo : navire roulier utilisés pour transporter des véhicules.
Encryptions : Procédé de cryptographie ayant pour but de rendre la compréhension d'un document impossible à toute personne n'ayant pas la clé de déchiffrement.

Ransomware : Logiciel malveillant qui prend en otage des données personnelles. Pour ce faire, un rançongiciel chiffre des données personnelles puis demande à leur propriétaire d'envoyer de l'argent en échange de la clé qui permettra de les déchiffrer.

Phishing : La technique consiste à se faire passer pour un tier de confiance afin de faire cliquer le destinataire sur un lien ou lui faire ouvrir une pièce jointe infecté.

Cracker : processus de récupération de mots de passe à partir de données stockées ou transmises par un système informatique, une approche commune consiste à tester de nombreux mot de passe connus stockés dans un dictionnaire et les comparer aux informations récoltées, on parle d'attaque par force brute.

IT : désigne les systèmes informatiques.
Hyperviseur : Une plate-forme de virtualisation permettant à plusieurs systèmes d'exploitation de travailler sur une même machine physique en même temps
Machine virtuelle : désigne un appareil informatique simulé par un logiciel.
SIEM : L'objectif du SIEM est d'apporter une solution pour analyser en temps réel les données récoltés par de multiple source de telles sortes à pouvoir en tirer des conclusions quant à l'état du système analysé.

Librairie : Fonctions utilisées par les programmes informatiques pour fonctionner.
Gestionnaire de paquet : serveur distant possédant de nombreux logiciel permettant l'automatisation du processus d'installation et de désinstallation des logiciels.
Compilation : Traduction d'un programme écris en langage de programmation compréhensible par un humain en un langage machine compréhensible par un ordinateur.
Forums : Espace public virtuel dédié à l'échange de message sur un thème donnés.
HOTSPARE : Un disque de secours.
Injection SQL : Exploitation de faille de sécurité d'une application interagissant avec une base de donnée
Contrôleur raid : Un périphérique qui gère les unités de disque physiques et les présente à l'ordinateur en tant qu'unité logiques.
Firmware : programme intégré dans un matériel informatique pour qu'il puisse fonctionner
Boot : Procédure de démarrage d'un ordinateur
Port mirroring : fonction de surveillance permet de copier des paquets transitant par le commutateur réseau, configuré pour cet usage, vers un port de destination choisi

Bibliographie

IBM. *Guide d'installation et d'utilisation de l'IBM X3650m3*
[en ligne] :https://www.ibm.com/docs/en/STAV45/com.ibm.sonas.doc/69y4099.pdf

Lenovo. *Documentation Xclarity*
[en ligne]: https://sysmgt.lenovofiles.com/help/index.jsp?topic=%2Fcom.lenovo.lxca.doc%2Faug_product_page.html

Snort. *Site officiel*
[En ligne]: https://snort.org/

Snort. *Guide d'installation pour Ubuntu*
[En ligne] : https://snort.org/documents/snort-3-1-0-0-on-ubuntu-18-20

Med Europe Terminal. *Site officiel*
[En ligne]: https://www.med-europe-terminal.com/

Abuse IPDB. *Base de donnée regroupant des IP malveillante*
[En ligne] : https://www.abuseipdb.com/

Ilir, K. *Thèse sur l'utilisation de Snort dans une PME*
[En ligne] : https://doc.rero.ch/record/327843/files/IlirKadriu_TB.pdf

Cisco. *Documentation sur l'utilisation de SPAN*
[En ligne] : https://www.cisco.com/c/fr_ca/support/docs/switches/catalyst-6500-series-switches/10570-41.html

Splunk. *Site officiel*
[En ligne] : https://www.splunk.com/fr_fr

Splunk. *Documentation sur Splunk Entreprise*
[En ligne] : https://docs.splunk.com/Documentation/Splunk

Wikipedia. *Page relative aux IDS*
[En ligne] : https://fr.wikipedia.org/wiki/Système_de_détection_d'intrusion