ENSEEIHT

Projet long

Sonde logicielle pour IA du réseau 5G

AMHAIDA Sabir
FETTOUHI Seifeddin
LAGLIL Oussama
LAHYAN Salah Eddine
NACHIT Btissam

20 mars 2021

Table des matières

1 Introduction **4**

2 Spécifications des données à monitorer **5**

2.1 données Basée sur les paquets 5

2.1.1 Adresses IP source et destination 5

2.1.2 Port source et destination 6

2.1.3 Contenu du paquet 6

2.1.4 La taille du paquet 6

2.1.5 Quantité du paquet 6

2.2 Données au niveau de flux 6

2.2.1 Le comptage de flux 7

2.2.2 La taille de flux 7

2.2.3 La direction de flux 7

2.2.4 La durée de flux 7

2.2.5 Débit de flux 7

2.3 Données basées sur la connexion 8

2.3.1 La durée de la connexion 8

2.3.2 La taille de la connexion 8

2.3.3 Nombre de connexions 8

2.3.4 Le type de la connexion 8

3 Les platformes de visualisation possible **9**

3.1 Outils de surveillance d'un réseau 9

3.1.1 Nagios XI 9

3.1.2 Zabbix 10
3.1.3 DataDog 10
3.2 Outils de visualisation des données 11
3.2.1 Power BI 11
3.2.2 Google Data Studio 11
3.2.3 Geckoboard 11

4 Conclusion du recherche **12**

5 Contexte de réalisation **13**

6 Analyse et conception **14**
6.1 Planification et architecture du project 14
6.1.1 Organisation 14
6.1.2 Taches à réaliser : 14
6.2 Diagramme de classe du projet 15

7 solution logicielle et tests **17**
7.1 Les technologies utilisé 17
7.2 Serveur back-end : 17
7.2.1 Le serveurs des commandes 17
7.2.2 Les sniffers 19
7.2.3 Les REST API's 20
7.3 Classification : 20
7.3.1 Entraînement du modèle ML 20
7.3.2 Préparation des données 21

7.3.3 Évaluation des performances 21
7.4 Méthodes de test : 22
7.5 Application web Front-end : 26
7.5.1 Page d'accueil 26
7.5.2 Information sur les machines 26

8 Document d'installation (Ubuntu 16.04 +) 29

8.1 installation base de donnée 29
8.2 Installation et lancement de l'application Web Angular - Flask 29
8.2.1 Installation Angular 29
8.2.2 Installation Flask 29
8.2.3 Lancement de l'application Web 29
8.3 Lancement du serveur de commande et web 30
8.4 introduire les clients et les sniffers 30

9 Conclusion et Perspectives 31

1 Introduction

La visualisation des données devient de plus en plus demandée pour surveiller et comprendre le comportement d'un système, c'est bien le cas des réseaux informatiques où les individus et les entreprises cherchent a garder des traces de ce qui se passe dans leur réseau. Ces données peuvent être utiles dans la détection des pannes, équilibrage du trafic, pour garantir un niveau de QoS stable et la détection des attentat des attaques informatique.

D'ici vient l'intérêt de faire une étude sur les différentes techniques de surveillance de réseaux, de voir leur état d'avancement et de distinguer leur avantages.

Dans un première temps, nous allons présenter les différentes propriétés du réseaux qui seront utiles par la suite dans le diagnostic de l'état du réseaux, avant de mettre l'accent sur les point commun entre les réseaux réels et les réseaux virtuels qui sont de plus en plus utilisés avec l'émergence du 5G et la quantité énorme de donnée qui circule dans l'internet.

Cette phase a pour but de chercher les principales caractéristiques d'un réseau et d'étudier leur pertinence sur les réseaux virtuels. Un autre sous-objectif sera aussi de chercher comment on peut acquérir ces informations, et de trouver à quel point seront-ils utiles pour décrire l'état global du réseau.

Dans un deuxième lieu, il faut regrouper et structurer ces donnée pour pouvoir les visualiser d'une façon efficace et compréhensible, d'où l'intérêt donc à faire des recherches sur les plateformes de visualisation qui existent et voir lesquels parmi eux vont être utiles pour représenter les données spécifiées dans la première phase, Ainsi que les restrictions causées par ces outils sur l'objectif final de la surveillance du réseau, Vu que la quantité des données a l'entrée est importante.

L'objectif donc de ce rapport est de présenter les différentes caractéristiques d'un réseaux et de voir leur effets sur l'état globale du réseaux, puis voir des exemples des outils de surveillance de réseaux pour voir comment ces caractéristiques peuvent être représentées et enfin les potentiels outils qu'on va utiliser lors du phase de visualisation des données collectées et traitées.

2 Spécifications des données à monitorer

Dans cette section, nous présentons les différentes types de données à monitorer, et leurs applications et usages. [Xuy] [Fro]

2.1 données Basée sur les paquets

Les paquets transportent beaucoup de données utiles pour la visualisation et l'analyse du trafic réseau. Cette méthode de collecte de données est la plus courante car plusieurs appareils l'utilisent, allant des téléphones portables aux réseaux d'entreprise à grande échelle.

Afin de transmettre une information d'une machine à une autre sur un réseau, celle-ci est divisée au niveau du paquet (couche réseau), encapsulée en paquets et associée à des adresses et des en-têtes avant d'être envoyée à la machine de destination où elle est décodée et exécutée. Ces en-têtes sont utilisés pour guider le paquet dans un réseau et peut être utilisé pour identifier et filtrer les paquets tels que les adresses IP, les ports et les protocoles.[SN12]

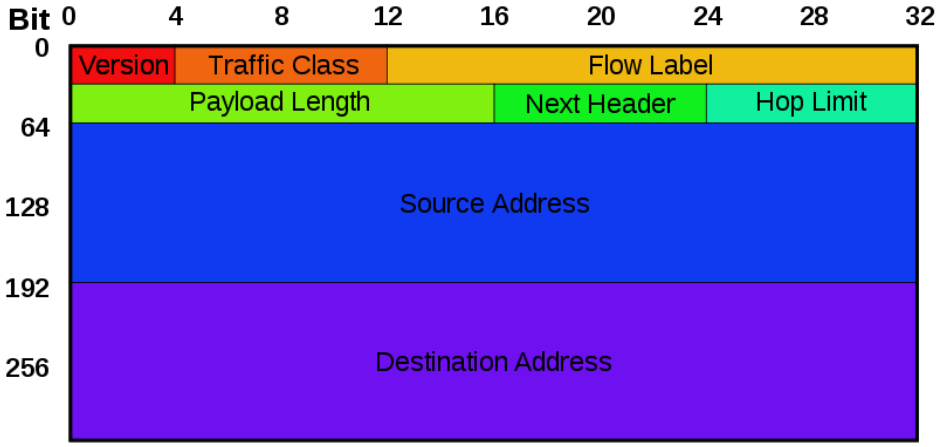

FIGURE 1 – Exemple : En-tête de paquet IPv6.

Étant donné la complexité et la variété de ces données au niveau des paquets, les composants suivants peuvent être les plus utiles à capturer et à examiner lorsqu'une surveillance ou visualisation du trafic réseau est nécessaire :

2.1.1 Adresses IP source et destination

L'adresse IP source/destination est l'aspect principale des protocoles de transmission de données, elles représentes les addresses du destinataire/dessinateur. ces données peuvent servir à definir une base de trafic "normal", c'est à dire que l'examinassions de ces addresses IP par exemple peut pointer vers attaques par déni de service distribuées

(abr. DDoS attack pour Distributed Denial of Service attack). où les adresses IP source du trafic se produisent en dehors des plages IP normales ou sont plus concentrées par rapport aux utilisateurs légitimes.[al12]

2.1.2 Port source et destination

Les paquets sont à destination vers différents ports dépendamment de leur objectif ou leur protocole. cette donnée peut être utile pour surveiller les anomalies. Par exemple lorsqu'il y a un trafic inhabituel vers ou depuis un port, les analystes de sécurité peuvent utiliser ce comportement réseau étrange pour déclencher des recherches supplémentaires.

2.1.3 Contenu du paquet

le contenu des paquets peut être trés utile dans le but d'identifier si un code malveillant est présent ou si des commandes d'application inhabituelles sont incluses. Les protocoles comme HTTP et DNS incluent des informations sur les paquets envoyées «en clair», alors que le trafic envoyé via Secure Sockets Layer (SSL) et Transport Layer Security (TLS) crypte les données, rendant l'analyse du contenu des paquets plus lente et plus difficile.

2.1.4 La taille du paquet

La taille d'un paquet est également utile pour la surveillance du réseau ou l'analyse de la sécurité. Comme pour les autres données basées sur les paquets.

2.1.5 Quantité du paquet

Un dernier type de données basé sur les paquets et qui peut être utile pour le monitoring d'un réseau est la quantité de paquets envoyés vers/depuis un réseau. par exemple la quantité d'activité réseau organisée par les différents types de paquets et de protocoles peut, par rapport aux adresses IP, être utilisée pour rechercher un comportement réseau malveillant si elle tombe en dehors de l'activité normale.

2.2 Données au niveau de flux

Les données au niveau des paquets ne permet que voir le réseau à partir d'un niveau très micro. de plus avec la 5G, la vitesse va être très élevée et avec les environnements d'application complexes et un cryptage des données, le monitoring basé sur les données au niveau du paquet manque de nombreuses données environnementales et contextuelles qui peuvent faciliter la surveillance et l'analyse.

Par conséquent, il faut utiliser également des données au niveau du flux, qui fournissent une vue un peu macro-niveau de l'activité du réseau, où des groupes de paquets qui partagent la même destination, source, type de protocole ou autres attributs similaires répertoriés dans l'en-tête du paquet sont analysés ensemble. ce regroupement peut aider à surveiller les performances du réseau, la santé de l'application ou l'activité de l'hôte, ainsi que le signalement d'un trafic réseau inhabituel qui peut indiquer une intrusion potentielle.

Une fois collectées, les données au niveau du flux peuvent être organisées et examinées selon les types de classification suivants :

2.2.1 Le comptage de flux

Le comptage de flux est très utile pour la surveillance d'un réseau, car il traque la fréquence des différents types de flux circulant dans le réseau, Par exemple, les données peuvent être organisées par adresse source de paquet, adresse de destination ou type de protocole (par exemple, DNS, TCP, HTTP) afin que le comportement du réseau puisse être évalué.

2.2.2 La taille de flux

la taille du flux regroupe la quantité et la taille du contenu des paquets collectés.

2.2.3 La direction de flux

La direction du flux peut aider les analystes réseau à surveiller la vitesse et la quantité de données entrante ou sortante d'un réseau.

2.2.4 La durée de flux

la durée du flux est la durée entre l'heure d'arrivée du premier paquet et l'heure d'expiration du flux

2.2.5 Débit de flux

Le débit est défini comme le nombre de paquets de transmission d'un flux par unité de temps. Dans un flux donné, les paquets partagent un ou plusieurs des mêmes attributs. Le débit a une spécificité plus élevée que le débit de paquets, mais ils ont pour la plupart des usages similaires.

2.3 Données basées sur la connexion

Les données basée sur le flux et sur les paquets fournissent des informations réseau plus complètes à examiner, mais les données basée sur la connexion peuvent aider à comprendre plus profondément la nature du trafic réseau entre des périphériques réseau spécifiques.

En particulier, les données basées sur la connexion sont l'agrégat du trafic réseau entre deux adresses IP spécifiques, une interne et une externe.

La surveillance du trafic entre deux points spécifiques peut être fait par plusieurs méthodes différentes :

2.3.1 La durée de la connexion

Les données basées sur la connexion peuvent être visualiser en fonction de leur durée, par exemple la quantité de données transmises entre une période de temps spécifique ou de manière globale pendant certaines périodes horaires sur plusieurs La durée de la connexion jours ou semaines. Ces informations peuvent être utiles pour surveiller le trafic d'un réseau.

2.3.2 La taille de la connexion

Les données de connexion peuvent également être visualiser et triées en fonction de la taille des paquets individuels transmis ou de la taille globale du flux de données entre deux machines. Le suivi de ce type de données peut aider à faire la distinction entre les flux de données normaux et ceux inhabituels.

2.3.3 Nombre de connexions

Le nombre de connexions d'une adresse IP particulière attribuée à un hôte peut aider à surveiller le nombre de connexions entre cet hôte et d'autres dans une certaine période. Un changement dans le nombre de connexions ou de nouvelles connexions inexpliquées peut être le signal de la nécessité d'une enquête plus approfondie.

2.3.4 Le type de la connexion

Enfin, les données de connexion peuvent être visualiser en fonction de type ou de protocole du trafic transmis, y compris TCP, UDP et ICMP. En comparant ces informations avec d'autres données telles que les données basées sur le flux ou au niveau des paquets.

3 Les platformes de visualisation possible

Dans ce chapitre, nous allons spécifier les outils et les méthodes pour mettre en forme et afficher les données.

3.1 Outils de surveillance d'un réseau

Les réseaux sont la pierre angulaire de toute entreprise moderne, et les ralentissements et les brèches sont coûteux. La surveillance consiste à surveiller le réseau interne dans son ensemble, y compris les appareils, le trafic et les serveurs. Cela permet d'identifier et de résoudre les problèmes potentiels au fur et à mesure qu'ils surviennent, évitant ainsi les problèmes de réseau. Pour presque toutes les entreprises, cette surveillance se fait à l'aide de systèmes logiciels.

Les systèmes de surveillance de réseau, dans leur forme la plus élémentaire, sont des outils qui aident les administrateurs à surveiller leurs réseaux plus efficacement. Nous allons citer dans cette section les principales outils qui aident à la supervision des réseaux.

3.1.1 Nagios XI

C'est l'un des rares outils qui permet une extrême flexibilité (du fait de son adaptabilité aux plug-ins) sur ce qui est surveillé et alerté pour un faible coût. Nagios XI se concentre sur la surveillance. Les principaux composants informatiques que Nagios XI surveille sont le réseau, l'infrastructure et la base de données. Bien que le logiciel soit facile à installer, il ne détecte pas automatiquement les périphériques. En effet, il faudra configurer chaque appareil qui doit être surveillé avec un fichier de configuration.[**Nagios**]

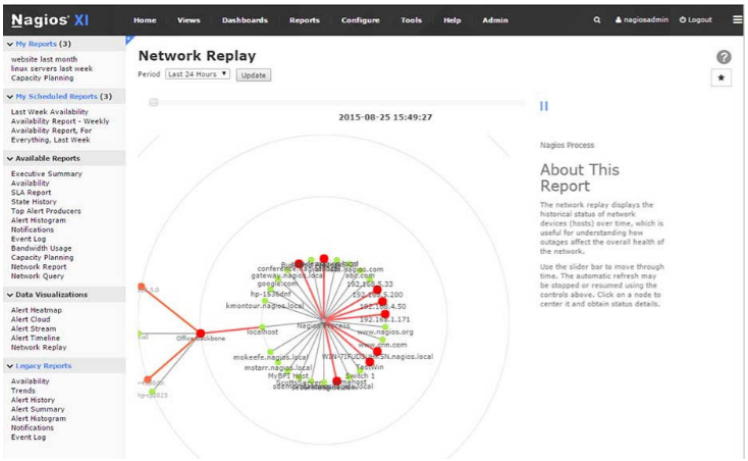

FIGURE 2 – Interface de Nagios XI.

3.1.2 Zabbix

Zabbix est un outil de surveillance open source. Il est populaire pour son interface graphique Web facile à utiliser et entièrement configurable. Zabbix se concentre sur la surveillance et les fonctionnalités de tendance. Ce logiciel est fréquemment utilisé pour surveiller les serveurs et le matériel réseau. L'un des points forts de Zabbix est qu'il peut prédire les tendances de votre trafic. Zabbix peut prévoir le comportement futur en fonction des données historiques.

Puisqu'il est open source, il dispose d'une communauté d'utilisateurs active répartie dans le monde entier et d'une bonne documentation. Zabbix donne la liberté d'utiliser la solution open-source sans verrouillage du fournisseur (y compris tous les composants).

Zabbix est puissant pour les réseaux SMB de moins de 1000 nœuds. Au-delà de cela, le logiciel peut devenir plus lent et ses performances diminuées. Un autre inconvénient est qu'il n'inclut pas les tests et les rapports en temps réel. [Zab]

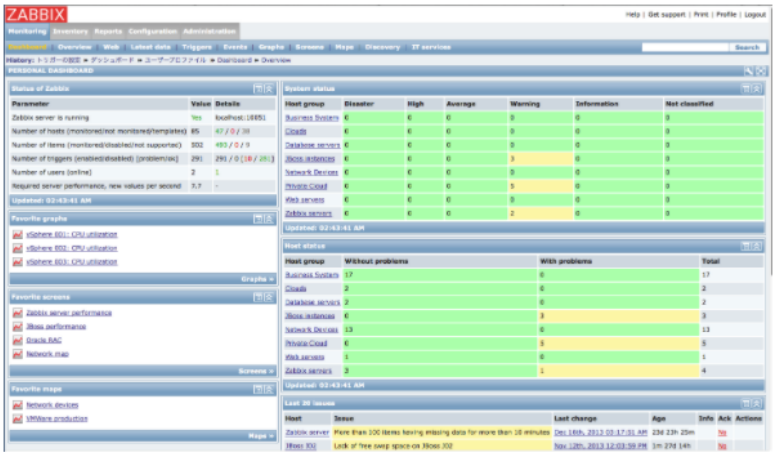

FIGURE 3 – Interface de Zabbix.

3.1.3 DataDog

Le module Infrastructure Datadog fait partie d'une suite d'outils de gestion des ressources informatiques basés sur le cloud. Le système Datadog peut être amélioré avec plus de 400 intégrations . Ces intégrations sont des procédures et des écrans spécifiques au produit qui se concentrent sur les produits d'infrastructure et les applications les plus vendus actuellement en cours d'utilisation. L'un d'eux est la gamme d'équipements Cisco Meraki.

Datadog Infrastructure est spécialisé dans la surveillance des équipements utilisés pour l'informatique. Les commutateurs, routeurs, points d'accès sans fil et pare-feu de la gamme Cisco Meraki font partie de cette catégorie. Datadog Infrastructure utilise le protocole SNMP (Simple Network Management Protocol) pour communiquer avec les agents de périphérique et obtenir des rapports d'état réguliers. Les agents SNMP sont préchargés

sur tous les produits Cisco Meraki. Le système Datadog effectue tous ses traitements sur les serveurs Datadog. C'est également le siège du tableau de bord système du client. Cependant, une partie de ce système de surveillance doit être installée. Il s'agit d'un programme d'agent et il peut être installé sur Windows , Linux ou Mac OS . [Dat]

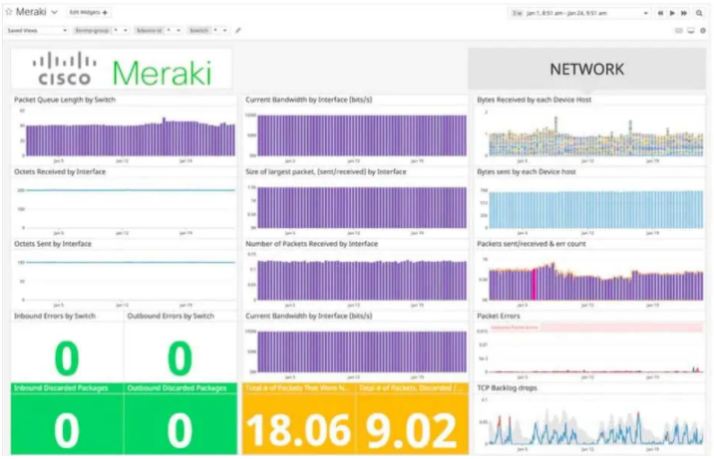

FIGURE 4 – Interface de DataDog.

3.2 Outils de visualisation des données

Dans cette partie nous présentons l'environnement de travail et les outils de développement utilisés. Ces outils qui permettent la manipulation, la visualisation et le traitement des données.[Kir]

3.2.1 Power BI

Power BI est un outil qui permet la connexion, et la visualisation des données afin de découvrir ce qui est important facilement. Il offre aussi des capacités d' entrepôt de données, notamment l'analyse avec des tableaux de bord interactifs.[kdn]

3.2.2 Google Data Studio

Un outil qui permet la manipulation des données de plusieurs ressources (MySql, google Analytics ..), cet outil offre la capacité de manipuler ces données et de les visualiser grâce à un Dashbord personalisé et intelligent.[goo]

3.2.3 Geckoboard

Geckoboard est un logiciel d'analyse des données qui facilite le traitement, la ségmentation et la restitution des données.[gec]

4 Conclusion du recherche

Dans ce rapport nous avons représenté l'environnement de développement de notre projet. Pour cela, nous avons présenté, en premier lieu, l'architecture générale des données à monitorer, et les différents aspects que nous allons traiter durant la phase de développement. En second lieu, nous avons élaboré une brève présentation des différents outils que nous allons utiliser durant la phase de traitement, la transformation et la visualisation des données.

Après avoir lister les objectifs de ce projet, il reste des questions qui se posent[ELB], à quel degré faut-il réduire le nombre de données collectées ? Est-ce une partie de cette réduction pourra être faite par les outils de visualisation directement ? Dans quelle étape donc faudra-t-il introduire les données à ces outils ? Comment peut-on arriver à visualiser ce qui se passe en temps réel ?

5 Contexte de réalisation

Depuis leur apparition dans les années 60, les réseaux d'interconnexions ont connu une évolution rapide autant au niveau de la taille qu'au niveau de la complexité. Et avec l'évolution rapide des réseaux tant en termes de taille que volume de trafic acheminé, l'utilisation traditionnelle de trafics rend difficile l'exploration des données et la surveillance de l'état du réseau surtout que ces données sont massives et collectées sur les réseaux en grande quantité. Par exemple, l'interface d'une machine est définie par plusieurs champs tels que l'adresse MAC et l'adresse IP, le nombre de paquets, l'état et bien d'autres.

Ensuite, avoir une vision claire de ce qui se passe dans un réseau devient une nécessité avec l'émergence des réseaux et services virtuels, parce que ces termes ont rendu difficile pour les développeurs de suivre ce qui se passe dans leur réseau et comment les services qu'ils fournissent interagissent les uns avec les autres. Par exemple, avec l'évolution rapide du réseau et des services, il devient plus difficile de trouver la source d'un bug en raison des nombreuses interconnexions et de la difficulté à suivre chaque lien.

Par conséquent, Le couplage des techniques de visualisation semble être une solution prometteuse pour résoudre ce problème. En adoptant cette approche, il pourrait être possible d'exploiter les capacités visuelles humaines pour explorer les grandes données provenant des réseaux. Les techniques de visualisation de l'information surmontent les difficultés liées à la lecture, à l'analyse et à l'interprétation de grandes quantités de données.

En plus, la visualisation d'informations est un processus de représentation graphique des données afin d'identifier les tendances et les corrélations qui ne peuvent pas être facilement vues dans les données brutes ou textuelles. C'est pourquoi ces techniques sont devenues un outil d'exploration de données populaire.

6 Analyse et conception

6.1 Planification et architecture du project

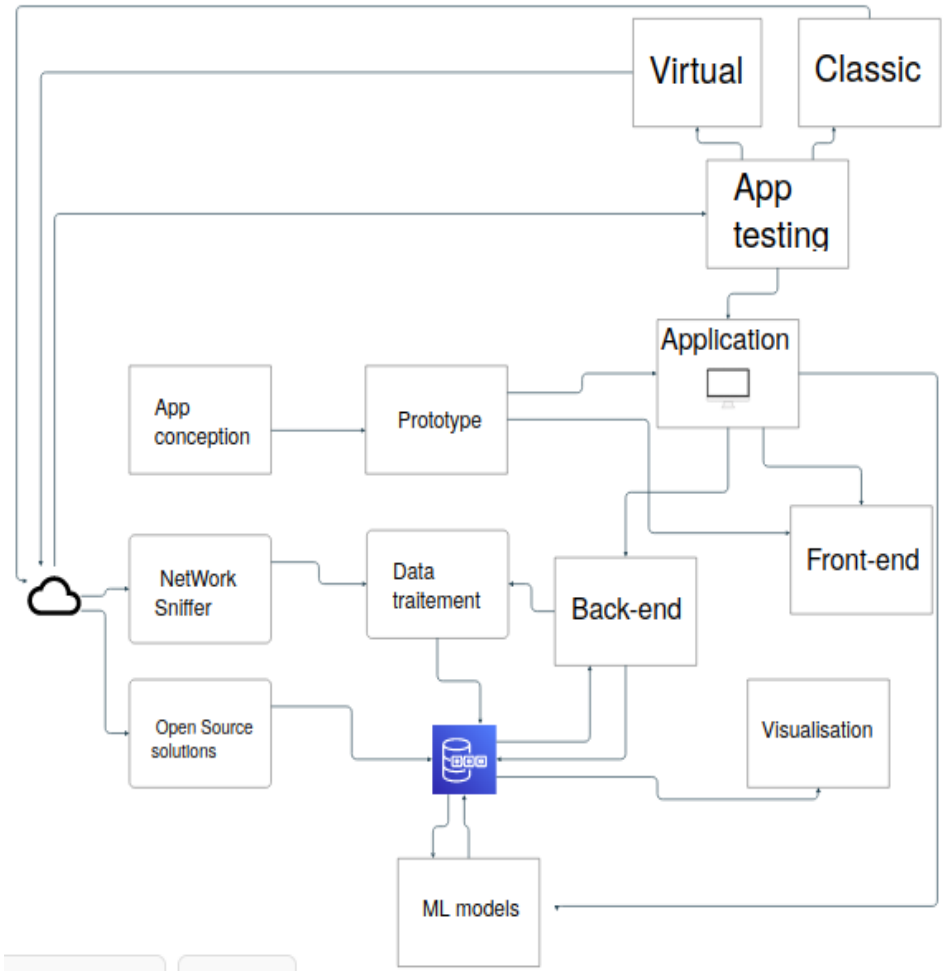

FIGURE 5 – Architecture du projet.

6.1.1 Organisation

Après avoir contextualisé le problème, l'étape suivante consiste à proposer l'architecture du logicielle à développer, comme le montre la figure 5. Certaines des tâches ont fait l'objet d'un travail collectif en raison de leur importance pour l'ensemble du projet. La conception de l'architecture et le processus de travail du logiciel en font partie.

6.1.2 Taches à réaliser :

La première tâche a été la création du prototype de l'application et le développement de la partie front-end. C'était la mission d'une sous-équipe de deux personnes.

La deuxième partie est le serveur back-end et les sniffers de réseau. Leur objectif principal est de collecter des informations à partir de nœuds spécifiques du réseau et de stocker les données dans une base de données. Et de donner à l'utilisateur administrateur un moyen de visualiser ce qui se passe dans le réseau et quelques statistiques basées sur ce qui a été capturé.

La troisième partie est une partie indépendante qui se concentre sur l'analyse de données et les modèles d'apprentissage automatique qui classifient le trafic capturé. Le résultat final est une information plus utile qui peut également être visualisée par l'utilisateur administrateur.

La dernière tâche est un plugin de test pour vérifier la fonctionnalité de la solution assemblée. Deux scénarios existent, l'un est un test dans un réseau réel, l'autre est l'utilisation d'un environnement virtuel pour tester et installer l'ensemble du logiciel. La raison de ces tests est de valider les exigences fonctionnelles, et de faciliter l'utilisation de ce logiciel.

6.2 Diagramme de classe du projet

Le diagramme de classes permet de présenter la structure des différents modules du projet, il constitue un élément très important de la modélisation, il permet de définir quelles seront les composantes du système final.

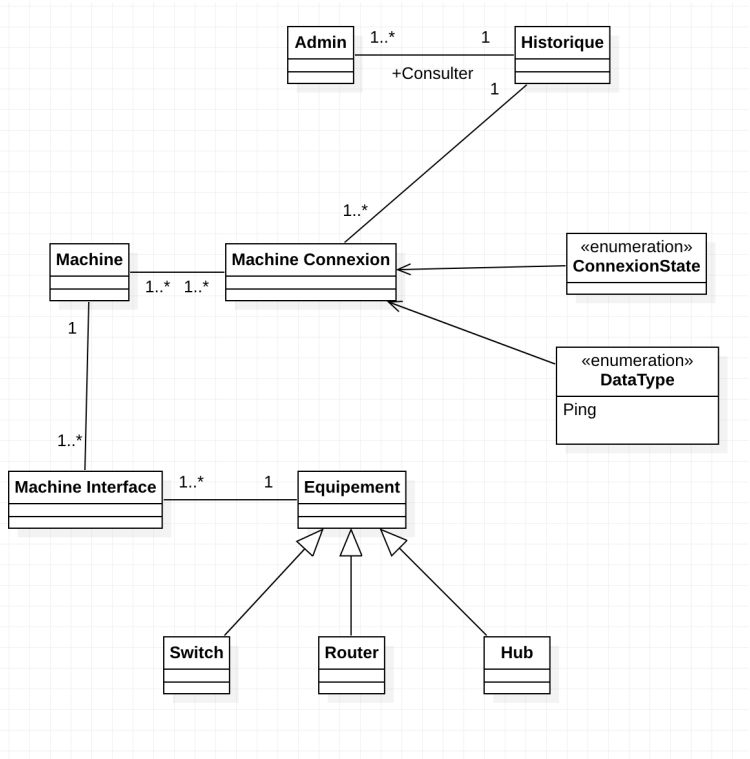

FIGURE 6 – UML diagramme de classe.

— **Admin :** Présente un type des utilisateurs de la plateforme, ayant comme rôle la supervision.

— **Historique :** L'ensemble d'historique des connexions entre les machines.

— **Machine Connexion :** Englobe les différentes connexions établies entre les machines.

— **Machine :** Présente les différents informations des machines dans le réseau

— **Machine Interface :** Présente L'ensemble des interfaces réseau lié à chaque machine.

— **Équipement :** Définit les informations de l'équipement(Hub, Router, Switch) lié à une interface réseau.

7 solution logicielle et tests

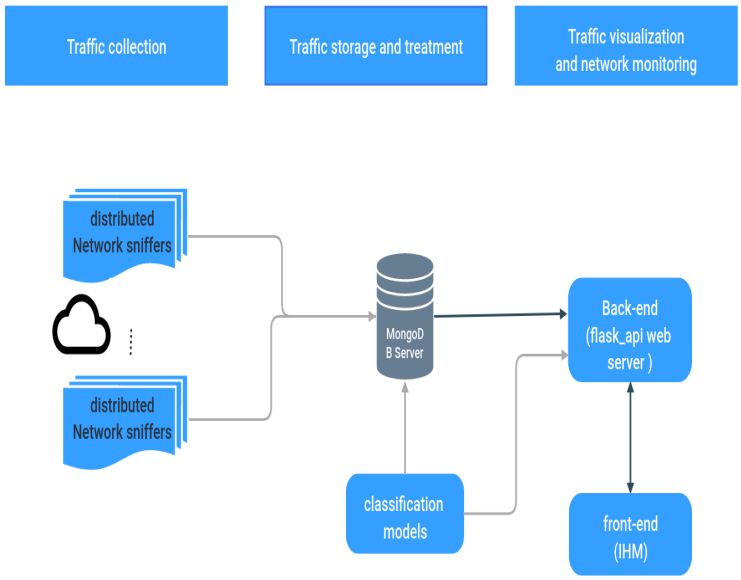

FIGURE 7 – Proposed monitoring System

Le système proposé est composé de trois phases :

- ***Collecte de trafic*** : cette phase vise à collecter le trafic réseau de différents hôtes en utilisant un système de sniffing distribué.
- ***Stockage et traitement du trafic*** : le but de cette phase est de collecter les données de trafic réseau dans la base de données NoSql afin de les visualiser ou de les utiliser dans nos modèles de classification.
- ***Visualisation du trafic et surveillance du réseau :*** cette phase vise à visualiser le trafic réseau à l'aide d'une application web

7.1 Les technologies utilisé

7.2 Serveur back-end :

7.2.1 Le serveurs des commandes

Tout commence par le déploiement du logiciel. Le processus de sniffer est illustré dans la Fig 9. Les sniffers sont installés sur certains des hôtes importants du réseau. Toutes les informations collecter sont envoyées à la base de données centrale pour un traitement ultérieur. Notez que l'accès au serveur est nécessaire avant de commencer l'insertion dans la base de données.

Un port est ouvert dans le serveur pour recevoir des commandes des clients (Dans ce

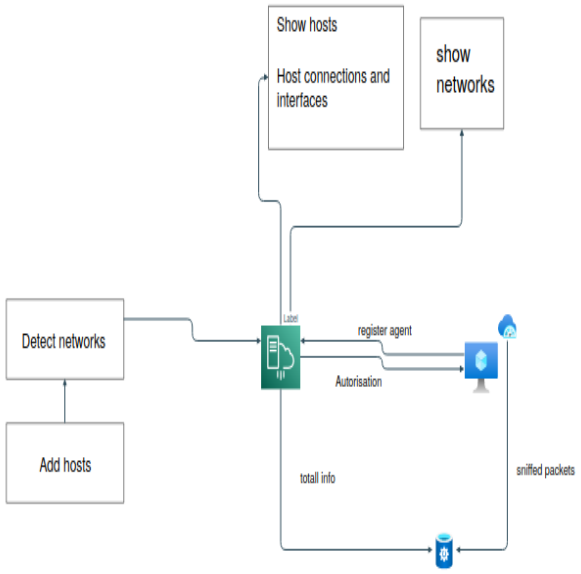

FIGURE 8 – Réception des commandes

cas les Hosts qui sniff), cette partie est développée pour avoir un contact direct via des socket avec le serveur central.

Exemple de ce contact : l'agent envoie une requête "add" au serveur, le serveur ajoute le sous-réseau dans lequel se trouve l'agent. Ensuite, le serveur demande à l'agent de commencer le sniffing (voir Figure 8). D'autres commandes ont été développées.

7.2.2 Les sniffers

Afin de collecter les données du réseau, un sniffer distribué a été adopté. Dans chaque hôte, il y a un sniffer qui utilise le multi-threading afin de collecter les données de trafic à partir de différentes interfaces.

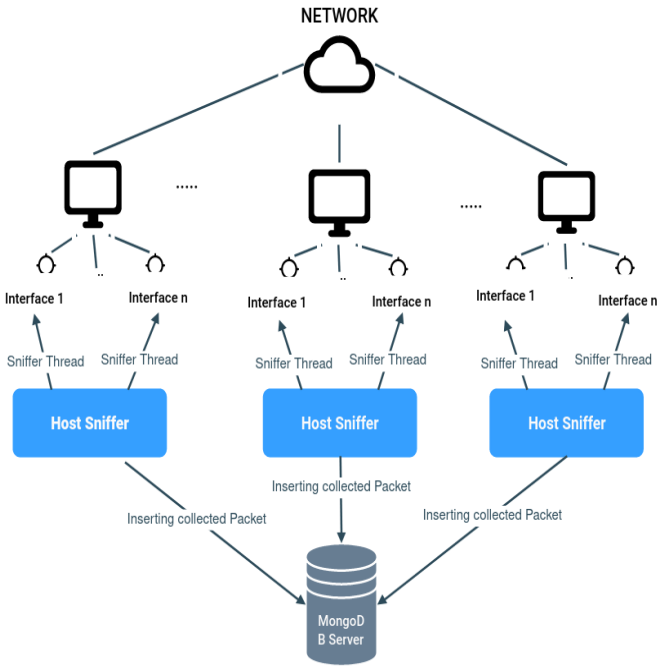

FIGURE 9 – Distributed multi threading network sniffer

Le sniffing distribué conçu se compose de plusieurs sniffers installés dans tous les nœuds. ces sniffers utilisent des threads pour paralléliser la collection des paquets à partir des différentes interfaces d'hôte, puis les insérer dans la base de données.

Les données suivantes ont été collectées :

— l'heure du paquet entrant / sortant IP source et destination de l'élément
— source et port de destination du paquet d'article
— taille du paquet
— type d'élément du protocole utilisé (tcp, udp, ...)
— nom de l'interface et le nom d'hôte à partir duquel le paquet a été capturé
— couche la plus élevée du paquet

Afin de simplifier l'interaction entre l'application Web et la base de données mongoDb, une interface de programmation d'application (API) a été utilisée.

7.2.3 Les REST API's

Comme représenté dans la figure 10, nous avons utilisé flask qui est un micro framework écrit en Python pour développer l'API qui servira d'intermédiaire entre le serveur web et la base de données.

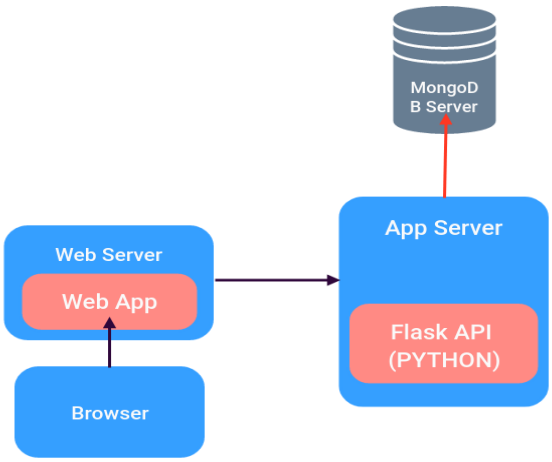

FIGURE 10 – API Architecture

7.3 Classification :

La solution proposée était de former les deux algorithmes d'apprentissage automatique. Dans cette section, un ensemble de données associé a été sélectionné, nettoyé et préparé pour les modèles ML. Les algorithmes ML sont appliqués pour regrouper et étiqueter l'ensemble de données.

7.3.1 Entraînement du modèle ML

Pour cette partie, le fichier qui contient des données «Flux de trafic réseau IP, étiqueté avec 87 applications» de la base de données *Kaggle* a été utilisé. Cet ensemble de données correspondait parfaitement à nos objectifs et répondait aux trois principaux composants d'un bon ensemble de données, qui sont réels, substantiels et diversifiés. Après cela, le but est d'appliquer les deux algorithmes sur notre propre ensemble de données (partie non réalisée).

Cet ensemble de données est composé de 3 577 296 instances et 87 fonctionnalités et a été conçu à l'origine pour la classification des applications. Chaque ligne représente un flux de trafic d'une source vers une destination et chaque colonne représente les caractéristiques des données de trafic.

	Flow.ID	Source.IP	Source.Port	Destination.IP	Destination.Port	Protocol	Timestamp	Flow.Duration	Total.Fwd.Packets	Total.Backward.Pack
0	172.19.1.46-10.200.7.7-52422-3128-6	172.19.1.46	52422	10.200.7.7	3128	6	26/04/201711:11:17	45523	22	
1	172.19.1.46-10.200.7.7-52422-3128-6	10.200.7.7	3128	172.19.1.46	52422	6	26/04/201711:11:17	1	2	
2	10.200.7.217-50.31.185.39-38848-80-6	50.31.185.39	80	10.200.7.217	38848	6	26/04/201711:11:17	1	3	
3	10.200.7.217-50.31.185.39-38848-80-6	50.31.185.39	80	10.200.7.217	38848	6	26/04/201711:11:17	217	1	
4	192.168.72.43-10.200.7.7-55961-3128-6	192.168.72.43	55961	10.200.7.7	3128	6	26/04/201711:11:17	78068	5	

FIGURE 11 – Les 5 premières lignes de l'ensemble de données

7.3.2 Préparation des données

Il existe des fonctionnalités telles que les adresses MAC source et de destination et les adresses de port, la durée du flux, le nombre d'octets de flux, le nombre de paquets de flux et la taille moyenne des paquets ...

Dans le processus de nettoyage des données, plusieurs opérations doivent être effectuées avant qu'il ne soit prêt pour la formation du modèle d'apprentissage automatique. S'il y a des instances en double dans l'ensemble de données, cela entraînera un biais dans l'algorithme d'apprentissage automatique.

De plus, certains modèles ML ne peuvent pas gérer les entrées de données manquantes. Dans ce cas, les lignes avec des données manquantes doivent être supprimées de l'ensemble de données ou les remplir avec les valeurs proches de la moyenne de cette entité. Dans cet ensemble de données, il existe plusieurs entités contenant différents types de données.Mais certains modèles ML ne peuvent fonctionner qu'avec des valeurs numériques. Pour utiliser ces types de données pour l'apprentissage du modèle ML, il est nécessaire de convertir ou de réaffecter des valeurs numériques pour représenter ses corrélations avec d'autres fonctionnalités. Ensuite, la normalisation Min / Max a été utilisée pour normaliser les caractéristiques avec une variance élevée. Tout cela a été traité dans le fichier datas-preparation.ipynb puis nous extrayons le fichier que nous appelons «KaggleImbalanced.csv»

7.3.3 Évaluation des performances

Des données étiquetées ont été utilisées pour former des modèles de classification. Il existe plusieurs modèles de classification disponibles et chaque modèle classe les données avec différents modèles mathématiques. Par conséquent, les résultats de chaque modèle peuvent être différents les uns des autres. Certains modèles peuvent être plus performants et certains modèles fonctionnent mal. En d'autres termes, il est préférable de former et de tester plusieurs modèles de classification pour savoir quel modèle convient le mieux au

projet. Nous classons les données en classes qui sont les noms de protocole et les deux modèles testés sont **Random Forest** sur le fichier déja préparé *KaggleImbalanced.csv* et **Artificial neural network (ANN)** sur le fichier source *Dataset-Unicauca-Version2-87Atts.csv*.

	precision	recall	f1-score	support
AMAZON	1.00	0.96	0.98	8375
DEEZER	0.00	0.00	0.00	2
DROPBOX	1.00	0.99	0.99	7057
EASYTAXI	0.00	0.00	0.00	7
EBAY	0.11	0.06	0.08	17
FACEBOOK	0.62	0.52	0.56	552
FTP_DATA	0.00	0.00	0.00	3
GMAIL	0.11	0.07	0.08	459
GOOGLE	0.83	0.91	0.87	176419
GOOGLE_MAPS	0.05	0.00	0.01	221
HTTP	0.95	0.93	0.94	29252
HTTP_CONNECT	0.87	0.86	0.86	38199
HTTP_PROXY	0.68	0.76	0.72	20443
INSTAGRAM	0.61	0.28	0.39	67
MICROSOFT	0.49	0.28	0.36	569
MQTT	0.22	0.08	0.12	25
MSN	0.16	0.15	0.16	204
NETFLIX	0.00	0.00	0.00	2
NFS	0.00	0.00	0.00	1
OFFICE_365	0.62	0.28	0.39	57
OSCAR	0.00	0.00	0.00	2
SKYPE	0.23	0.13	0.17	376
SSL	0.71	0.71	0.71	39540
SSL_NO_CERT	0.36	0.43	0.39	238
TWITTER	0.20	0.08	0.12	365
UPNP	0.00	0.00	0.00	2
WHATSAPP	0.50	0.28	0.36	68
WHOIS_DAS	0.00	0.00	0.00	4
WIKIPEDIA	0.00	0.00	0.00	8
WINDOWS_UPDATE	0.08	0.07	0.07	15
YAHOO	0.39	0.08	0.13	389
YOUTUBE	0.51	0.15	0.23	23636
accuracy			0.82	346574
macro avg	0.35	0.28	0.30	346574
weighted avg	0.80	0.82	0.80	346574

FIGURE 12 – Rapport de classification ANN

— Les modèles de classification ont été analysés plus en détail à l'aide de matrices de confusion pour vérifier l'exactitude et les figures suivantes montrent les résultats pour chaque modèle.

— À partir des matrices de confusion, nous pouvons voir que les deux modèles montrent de bons résultats et sont capables de classer la plupart des paquets réseau.

7.4 Méthodes de test :

L'objectif donc était d'installer les sniffer dans des machines spécifiques du réseau pour regarder les paquets échangées avec ces noeud, la figure 16 montre clairement cette approche.

	precision	recall	f1-score	support
AMAZON	0.96	0.94	0.95	2701
APPLE	0.95	0.88	0.92	498
APPLE_ITUNES	0.95	0.64	0.77	64
CLOUDFLARE	0.98	0.98	0.98	864
CONTENT_FLASH	0.99	0.97	0.98	147
DNS	1.00	0.98	0.99	57
DROPBOX	0.99	0.92	0.95	1090
FACEBOOK	0.98	0.96	0.97	1188
GMAIL	0.96	0.85	0.90	1474
GOOGLE	0.99	1.00	0.99	41072
HTTP	1.00	1.00	1.00	36839
HTTP_CONNECT	0.99	1.00	1.00	15564
HTTP_PROXY	0.98	1.00	0.99	22789
IP_ICMP	1.00	1.00	1.00	80
MICROSOFT	0.97	0.94	0.96	3520
MSN	0.98	0.82	0.89	578
MS_ONE_DRIVE	0.94	0.74	0.83	100
NETFLIX	0.98	0.77	0.86	114
OFFICE_365	0.97	0.79	0.87	272
SKYPE	0.97	0.88	0.92	1275
SSL	0.99	1.00	0.99	22395
TWITTER	0.95	0.79	0.87	883
WHATSAPP	0.96	0.63	0.76	138
WIKIPEDIA	0.99	0.78	0.87	174
WINDOWS_UPDATE	0.99	0.97	0.98	2257
YAHOO	0.94	0.88	0.91	1079
YOUTUBE	0.96	0.98	0.97	7431
accuracy			0.99	164643
macro avg	0.98	0.89	0.93	164643
weighted avg	0.99	0.99	0.99	164643

FIGURE 13 – Rapport de classification Random Forest

On a utilisé docker pour tester le fonctionnement des composants du projet, une architecture réseau de test a été crée (Voir figure 17), deux réseaux de clients ont été simulés et un réseaux interne qui contient le serveur et la base de donnée.

Pour créer cette architecture on a utilisé DOCKER-COMPOSE pour créer les différents réseaux virtuels, les différent services et les Hosts. l'architecture dans la Figure 22 explique les étapes à suivre pour obtenir un déploiement correcte des éléments du projet. Une image docker a été crée dans un premier lieu pour installer toutes les bibliothèques nécessaire pour le bon fonctionnement des composants de la solution, ensuite des images sont faites pour les différents services (BD, serveurs, clients).

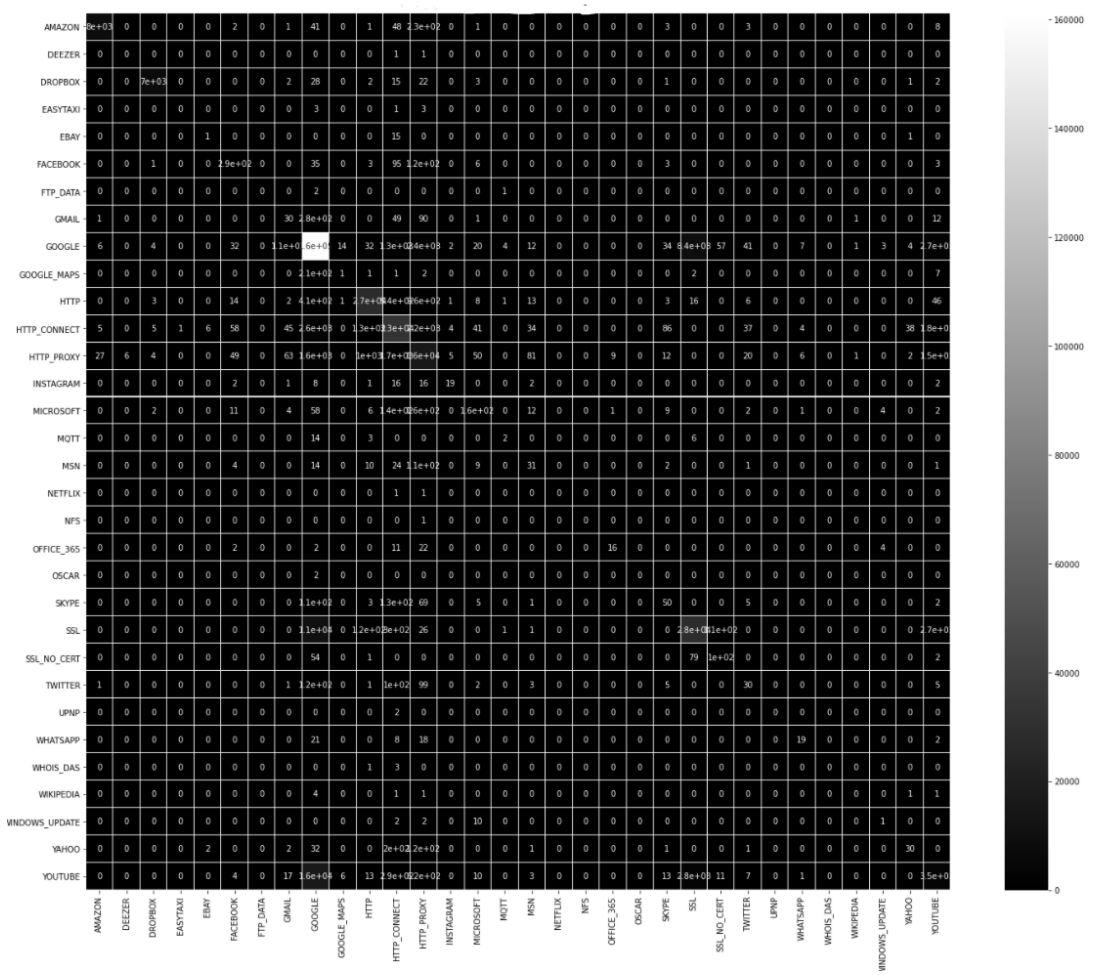

FIGURE 14 – Matrice de confusion pour l'algorithme ANN sur 3 000 000 lignes

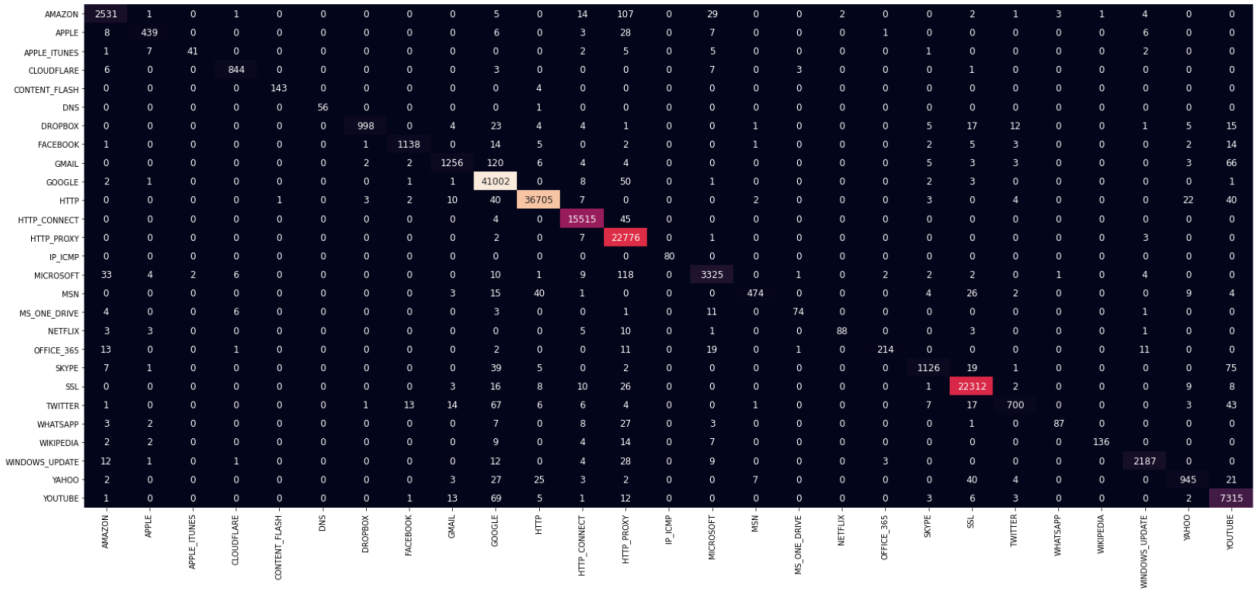

FIGURE 15 – Matrice de confusion pour l'algorithme Random Forest sur 500 000 lignes

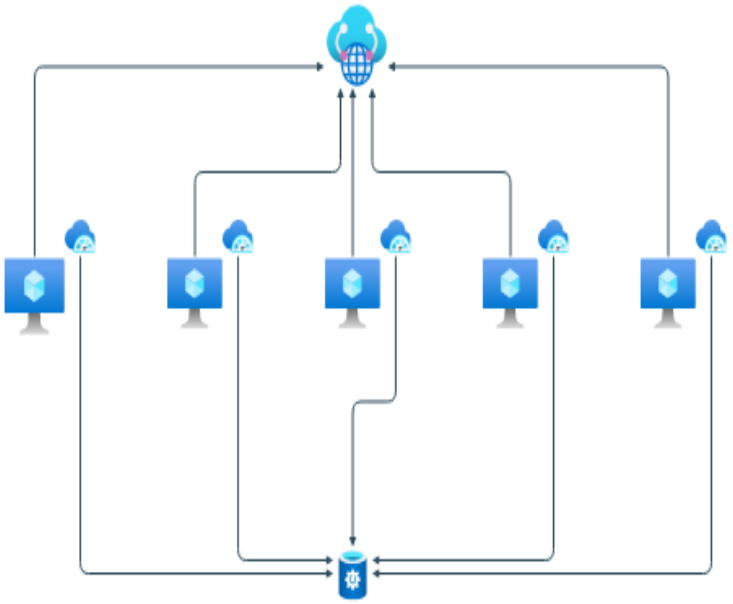

FIGURE 16 – Multiple Sniffers

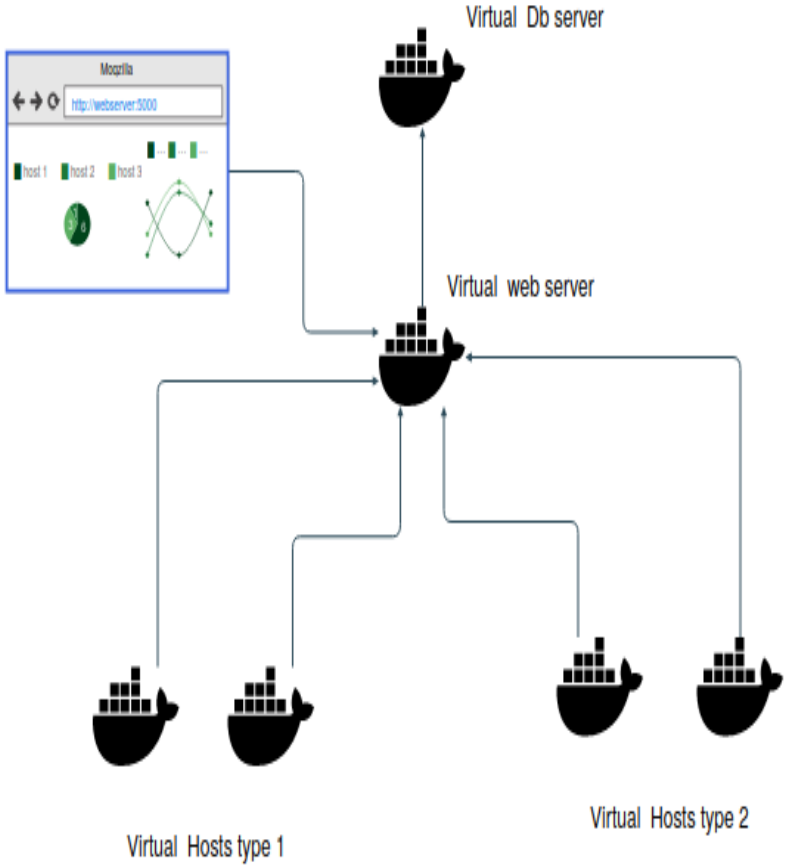

FIGURE 17 – Architecture Virtuel

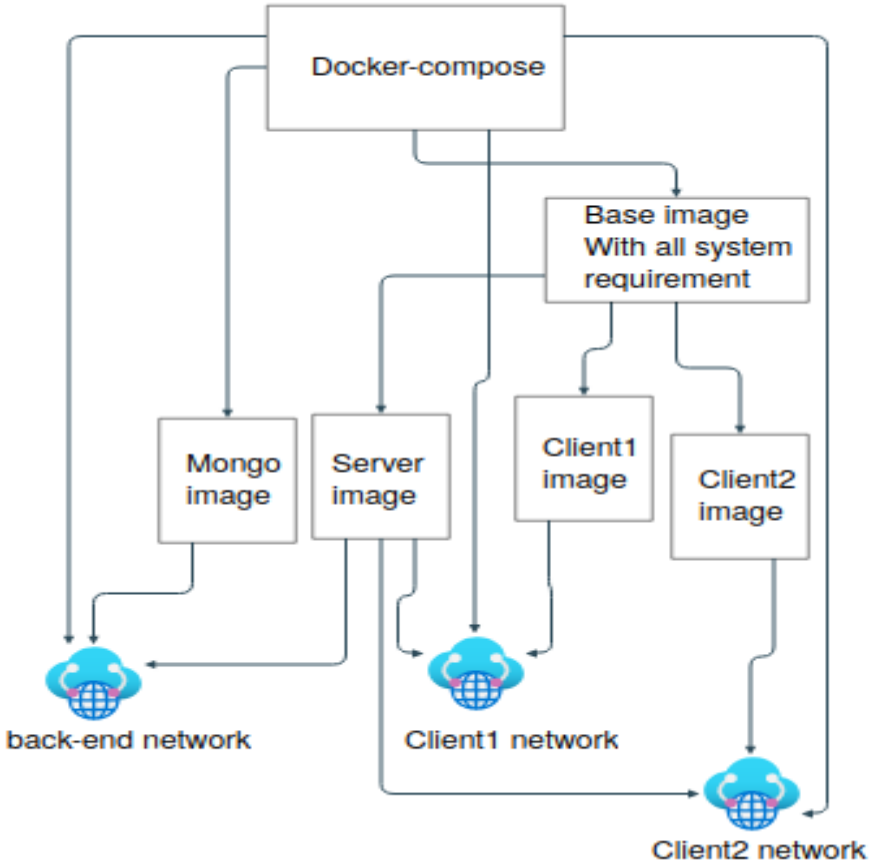

FIGURE 18 – Docker-compose architecture

7.5 Application web Front-end :

7.5.1 Page d'accueil

Dès que l'utilisateur accède à son propre espace (Administrateur). La Figure 14 concerne la page d'accueil d'un administrateur, qui comporte une liste des réseaux , le nombre des machines, les paquets et protocoles échangés dans le système. Ainsi, un graphe [Figure15] qui montre les protocoles utilisés dans l'échange des paquets.

7.5.2 Information sur les machines

Dans cette page Figure16, nous remarquons les informations obtenus sur chaque machine existante dans le réseau, avec une liste des interfaces réseau et leurs états(UP, Down, Unknown). Au Dessous de la page [Figure17] nous avons un graphe qui démontre l'utilisation des paquets pour chaque interface réseau.

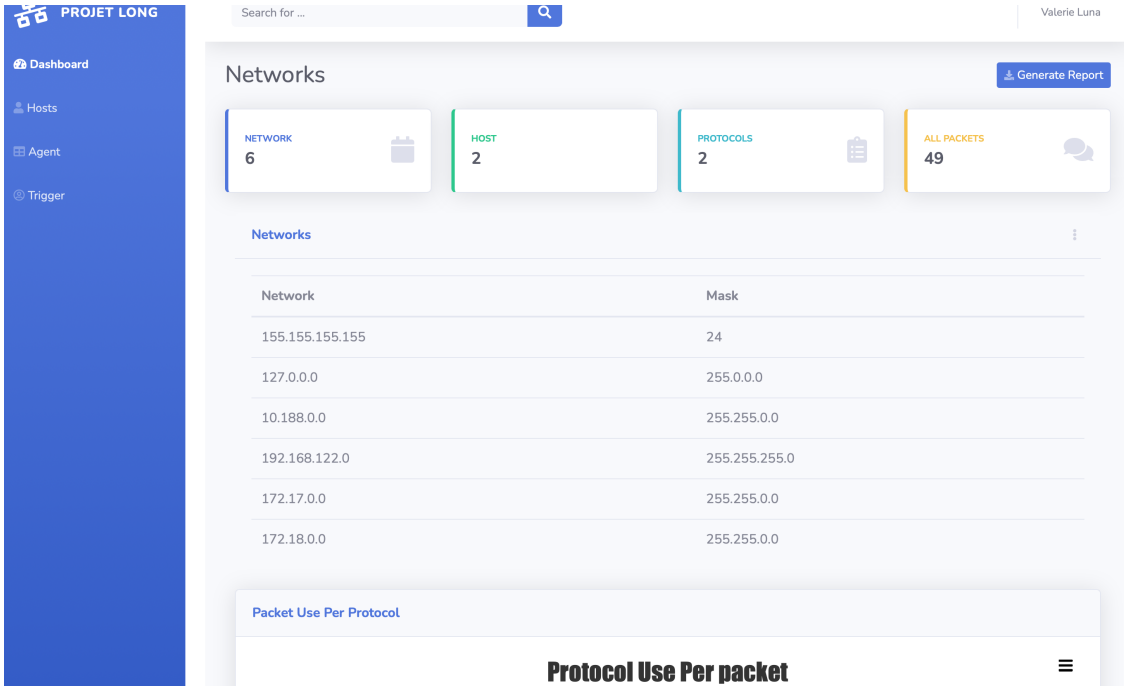

FIGURE 19 – Page d'accueil - 1

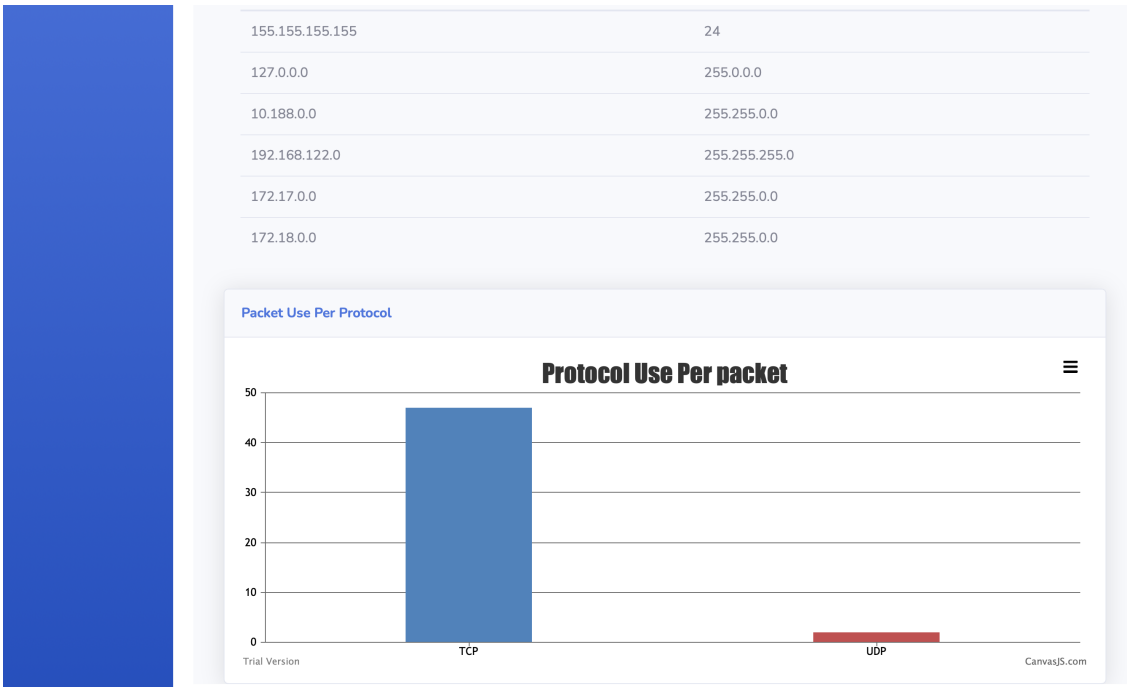

FIGURE 20 – Page d'accueil - 2

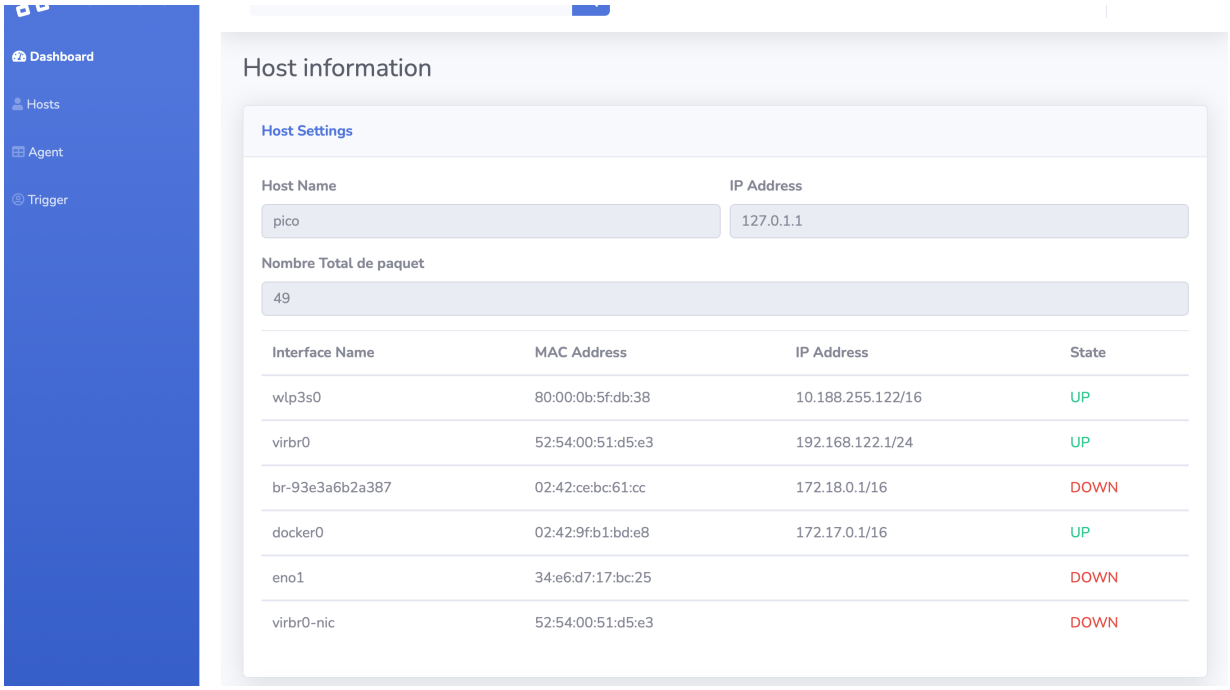

FIGURE 21 – Host Informations

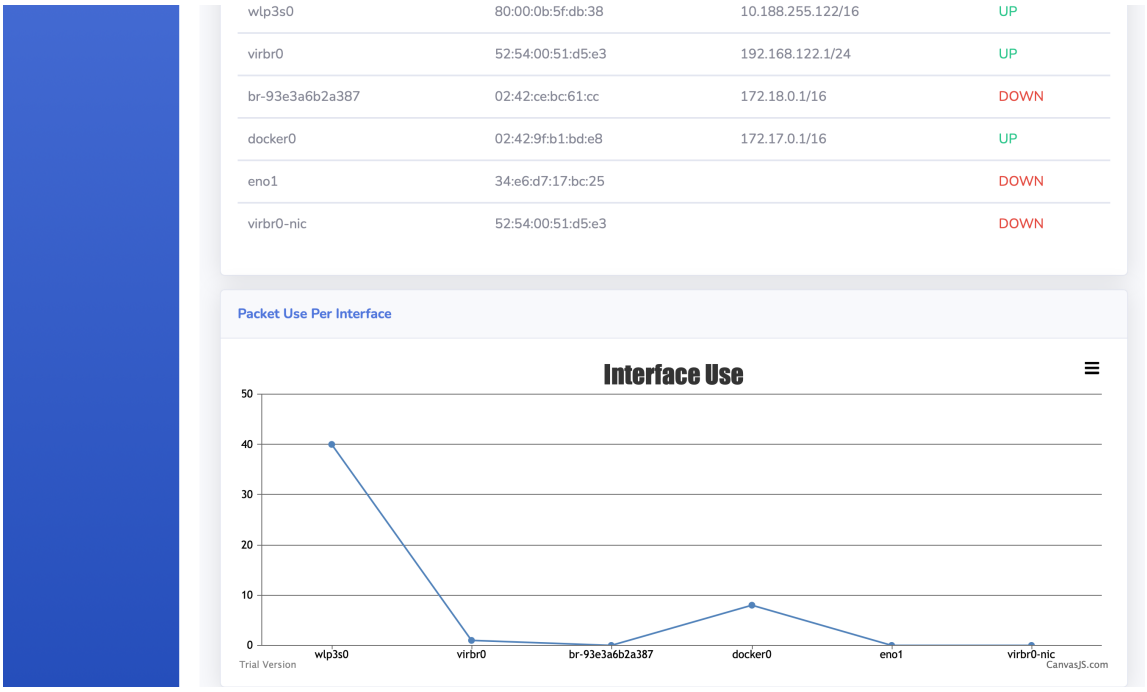

FIGURE 22 – Host Informations

8 Document d'installation (Ubuntu 16.04 +)

8.1 installation base de donnée

Deux méthodes sont possible :

— Méthode classique d'installation mongoDB(tutorial : `https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/`)

— un scripts bash **(mongo.sh)** qui lance la base de donnée dans un container via un docker file.

8.2 Installation et lancement de l'application Web Angular - Flask

8.2.1 Installation Angular

Méthode classique d'installation Angular 9 :

```
sudo npm install -g @angular/cli@9
```

si la commande npm n'existe pas ! vous pouvez l'installer à l'aide de la commande :

```
sudo apt install npm
```

8.2.2 Installation Flask

le tutorial d'installation de flask est diponible sur ce site `https://linuxize.com/post/how-to-install-flask-on-ubuntu-18-04/`

il faut uniquement s'assurer que vous avez installé python3. vous pouvez vérifier ceci en tapant :

```
python3 -V
```

8.2.3 Lancement de l'application Web

— Lancement de l'application depuis le répertoire front-end à l'aide de la commande **ng serve** sur le port 4200.

```
ng serve
```

— Lancement de serveur flask à l'aide de la commande **flask run**

```
flask run
```

8.3 Lancement du serveur de commande et web

Deux méthodes sont possibles :

— Directement dans un server web : lancer le script **(websnif.sh)**
  * Le serveur web prend le port 5000.
  * Le serveur des commands prend le port 15000.

— un scripts bash **(server.sh)** qui lance les serveurs dans un container via un docker file.

Pour les deux serveurs si la base de donnée est lancé dans une autre machine il faut la changer dans le ficher **sniffer_app.py** et **DBController.py**

8.4 introduire les clients et les sniffers

Pour Lancer un host dans une machine il faut lancer le script **(client.sh)** qui prend en paramètre l'adresse ip du serveur. Ce script lance une demande d'inscription au serveur puis commence le sniffing et il envoie les données collecter au serveur qui les stock.

deux types de sniffer existe : un qui envoi les donnée directement au base de donnée **sniffingHost.py**, un autre qui les envoyez au serveur via le port 15000 et le serveur qui communique avec la base de donnée **clintsfiles/sniffingHost.py**.

Pour l'instant le rôle du client est d'envoyer une requête "add" pour s'ajouter au base de donnée puis envoyer les donnée au serveur.

Vous pouvez ajoutez également d'autres requête au client en cas de besoin, une exemple sera demande des informations spécifiques. un perspective est de pouvoir commander le client via le serveur pour arrêter le sniffing.

Pour tester des clients virtuel, un script **(start.sh)** qui prend comme paramètres, le répertoire de travaille, Nb1 de clients dans un réseaux 1, Nb2 de clients dans un réseaux 2.

9 Conclusion et Perspectives

Les réseaux sont la pierre angulaire de toute entreprise moderne, et les ralentissement-set les brèches sont coûteux. La surveillance consiste à surveiller le réseau interne dans son ensemble, y compris les appareils, le trafic et les serveurs. Cela permet d'identifier et de résoudre les problèmes potentiels au fur et à mesure qu'ils surviennent, évitant ainsi les problèmes de réseau. Pour presque toutes les entreprises, cette surveillance se fait à l'aide de systèmes logiciels.

Les systèmes de surveillance de réseau, dans leur forme la plus élémentaire, sont des outils qui aident les administrateurs à surveiller leurs réseaux plus efficacement.

Dans ce rapport nous avons représenté l'environnement de développement de notre projet. Pour cela, nous avons présenté, en premier lieu, l'architecture générale des données à monitorer, et les différents aspects que nous allons traiter durant la phase de développement. En second lieu, nous avons élaboré une brève présentation des différents outils que nous allons utiliser durant la phase de traitement, la transformation et la visualisation des données

Références

[SN12] P. Lorier S. Alcock et R. Nelson. "Libtrace : A packet capture andanalysis library". In : *inProc. ACM SIGCOMM Comput. Commun. Rev.* (2012), p. 42-48.

[Dat] DataDog. *Outil de surveillance DataDog.* URL : https://www.datadoghq.com/dg/monitor/network/gen/?utm_source=Advertisement&utm_medium=Advertisement&utm_campaign=PCWorld-NetworkMonitoring.

[ELB] Meryem ELBAHAM. "Visualisation de trafic de réseau en temps réel". In : *ÉCOLE DE TECHNOLOGIE SUPÉRIEURE, UNIVERSITÉ DU QUÉBEC* ().

[Fro] Andrew Froehlich. *Network visibility and monitoring tools now amp up security.* URL : https://searchsecurity.techtarget.com/tip/Network-visibility-and-monitoring-tools-now-amp-up-security.

[gec] geckoboard. *geckoboard.* URL : http://developer-custom.geckoboard.com/#introduction.

[goo] google. *Options disponibles dans Data Studio.* URL : https://support.google.com/datastudio/answer/6283323?hl=fr.

[kdn] kdnuggets. *kdnuggets.* URL : https://www.kdnuggets.com/2018/02/comparative-analysis-top-6-bi-data-visualization-tools-2018.html.

[Kir] [2] Leena Ajay Gokhale Kirti Nilesh Mahajan. "Comparative Study of Data Visualization Tools". In : *Institute of Management and Entrepreneurship Development (IMED), Bharati Vidyapeeth Deemed to be University,* ().

[Xuy] Zheng Yan Xuyang Jing. "Security Data Collection and Data Analyticsin the Internet : A Survey". In : *IEEE COMMUNICATIONS SURVEYS TUTORIALS, VOL. 21, NO. 1, FIRST QUARTER 2019* ().

[Zab] Zabbix. *Outils de surveillance Zabbix.* URL : https://www.zabbix.com/.

[al12] S. Yuet al. "Discriminating DDoS attacks from flash crowds usingflow correlation coefficient". In : *IEEE Trans. Parallel Distrib. Syst.* 23, no. 6 (Jun 2012), p. 1073-1080.