MS BIG DATA : GESTION ET ANALYSES DES DONNÉES MASSIVES

INF 726 : SÉCURITÉ POUR LE BIG DATA

Julien Dreano

PROJET : MISE EN PLACE ET MANIPULATION D'UNE STACK ELK

Andres SOTO

2021-2022

1. Introduction

Avec l'utilisation toujours plus croissante des objets connectés, la quantité de données générées explose aujourd'hui et atteint des niveaux sans précédents. Cette explosion des données est également accompagnée d'une explosion des cyber-attaques et des vols des données personnelles et professionnelles. Ce phénomène n'est pas sans conséquences pour les entreprises car il peut entraîner de lourdes répercussions telles que des pertes financières, des dépenses judiciaires, des amandes ou encore des sanctions.

Pour toutes ces raisons, il est devenu urgent de changer d'approche en matière d'infrastructure technologique et penser à de nouvelles technologies capables de mieux surveiller les services informatiques et ainsi permettre une détection plus rapide des situations anormales sur les réseaux.

Plus précisément, l'architecture d'une solution de cybersécurité en environnement Big Data doit répondre aux exigences suivantes :

- La solution doit être capable de se connecter à n'importe quelle source de données, c'est-à-dire être capable de lire les données de n'importe quel format
- être capable de stocker et conserver sans limitation de durée toutes les données acquises
- permettre d'effectuer des analyses, des modélisations et des croisements entre les données en temps quasi-réel
- permettre de faire du reporting et de l'analyse visuelle pour la présentation des résultats des analyses

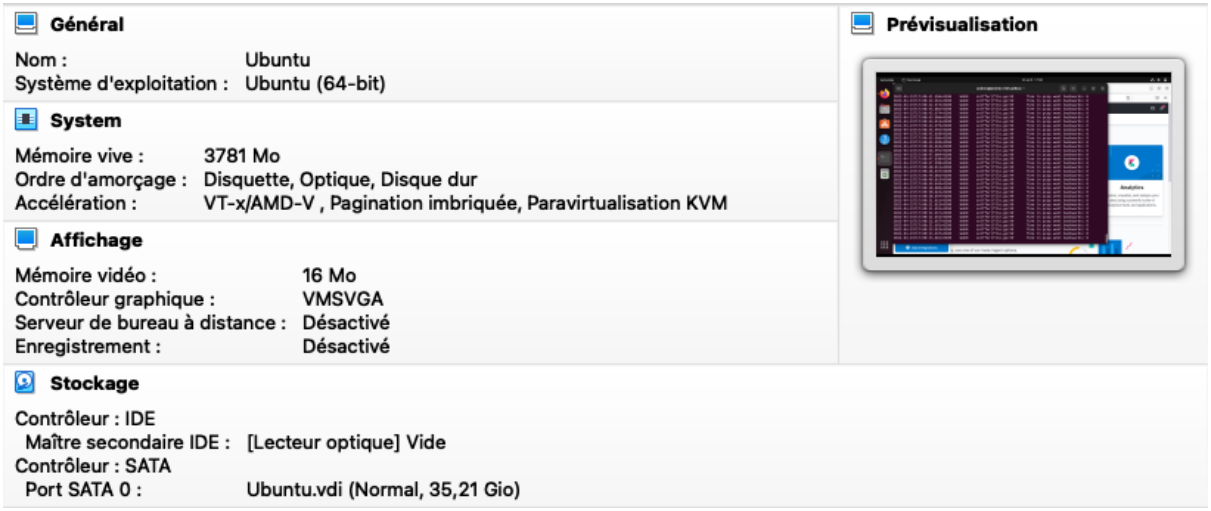
C'est dans ce contexte que s'inscrit le présent projet qui a pour objectif la construction d'un environnement et la mise en place d'une stack ELK pour l'analyse de fichiers réseaux.

2. Choix de l'infrastructure

L'environnement technique utilisé pour la mise en place de la stack ELK est la suivante :

- Machine virtuelle VirtualBox :
  - OS : Ubuntu 20.04
  - 35 Go de mémoire de stockage
  - 3 Go de mémoire RAM
  - Processeur 1 coeur

Dans le but de permettre une meilleure fluidité de la stack ELK j'ai décidé d'allouer pratiquement la moitié de la mémoire RAM disponible sur mon ordinateur personnel soit 3,7 Go sur les 8 Go. Cependant, cette configuration reste limitée du fait des besoins gourmands de la stack ELK en termes de ressources calculatoire. La figure ci-dessous vous présente plus en détail les différentes configurations de la VM.



3. La stack ELK

3.1 La stack ELK

Les logs sont l'une des informations les plus précieuses en matière de gestion et de surveillance des systèmes informatiques. Comme ils enregistrent toutes les actions qui ont eu lieu sur une machine, ils fournissent des informations dont on a besoin pour repérer les problèmes susceptibles d'avoir un impact sur les performances, la conformité et la sécurité des éléments d'une infrastructure. C'est pourquoi la gestion des logs est très importante et devrait faire partie de toute infrastructure de surveillance.

Lorsqu'on parle d'analyse de logs, le plus important défi consiste à regrouper, normaliser, visualiser et analyser les logs dans un emplacement unique et accessible. L'analyse de logs est un processus qui permet de donner du sens aux messages de logs générés par un système dans le but d'utiliser ces données pour améliorer ou résoudre des problèmes de performances au sein d'une application ou d'une infrastructure. Dans une vue d'ensemble, les entreprises analysent les logs pour atténuer les risques de manière proactive et réactive, se conformer aux politiques de sécurité, aux audits et aux réglementations, et permet également de mieux comprendre le comportement des utilisateurs utilisant les applications.

Lorsqu'une entreprise tire parti de l'analyse des logs, elle peut détecter les problèmes avant ou lorsqu'ils surviennent et éviter le gaspillage de temps, les retards inutiles et les coûts supplémentaires qui vont avec. Ainsi, les équipes peuvent intervenir et résoudre les problèmes plus rapidement, leur permettant ainsi de se concentrer davantage sur l'amélioration des fonctionnalités existantes et sur l'ajout de nouvelles fonctionnalités aux produits et services qu'ils créent au lieu de passer du temps à dépanner manuellement.

C'est dans ce contexte qu'a émergé la stack (ou pile) ELK qui est une combinaison de 3 projets open source : Elasticsearch, Kibana et Logstash. Ces briques logicielles ont toutes étés développées par l'entreprise Elasticsearch pour le stockage et l'analyse des fichiers de logs réseaux. Plus précisément, la stack ELK a été conçue pour permettre aux utilisateurs d'analyser et visualiser en temps réel tous types de données réseau indépendamment du format des données collectées.

La stack ELK permet de répondre aux besoins des administrateurs en fournissant les fonctionnalités essentielles pour l'automatisation des processus de traitement des logs. Ainsi, ELK permet :

- ➢ Collecte : la collecte des logs de différentes sources
- ➢ Traitement : le traitement des données car dans leurs états brutes les logs ne peuvent pas être analysés
- ➢ Stockage : les données sont collectées et centralisées à un endroit précis

- ➢ Indexation : l'indexation permet de faciliter la recherche d'un log et il est nécessaire pour le requêtage des données
- ➢ Visualisation : la création de dashboards facilite la compréhension et l'analyse des logs ce qui permet de prendre des bonnes décisions rapidement

Nous verrons plus en détail dans ce qui suit le rôle de chacune des composantes de la stack ELK.

3.2 ElasticSearch

Elasticsearch est un moteur de recherche et d'analyse de données open source distribué, basé sur Apache Lucene et développé en Java. Le projet a commencé comme une version extensible (scalable) du framework de recherche open-source Lucene. La capacité d'étendre horizontalement les indices Lucene a ensuite été ajoutée. Elasticsearch permet de stocker, rechercher et analyser de larges volumes de données rapidement et presque en temps réel. Les réponses sont transmises en quelques millisecondes.

Cette vitesse est liée au fait qu'Elasticsearch recherche un index plutôt que de rechercher directement le texte. Sa structure est basée sur les documents plutôt que sur des tableaux et des schémas. Les REST API permettent de stocker et d'explorer les données. En résumé, Elasticsearch est un serveur capable de traiter les requêtes JSON et de retourner des données JSON. Au sein de l'architecture ELK, Elasticsearch est en charge du stockage et de l'indexation des données.

3.3 Kibana

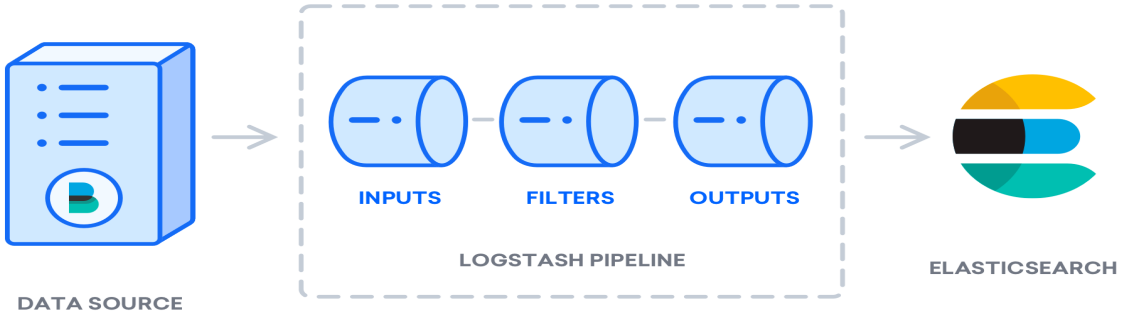
Kibana est l'outil de visualisation de données de la stack ELK. C'est une interface web open source qui permet de rechercher et de visualiser les données indexées dans Elasticsearch. L'un des principaux avantages de Kibana est qu'il offre la possibilité de créer des tableaux de bords (dashboard) interactifs à partir de données sélectionnées dans la base Elasticsearch. Ces dashboards donnent la possibilités aux utilisateurs de les personnaliser avec tous types de visualisations (graphique en barre, nuages de points, histogrammes, cartographies…). Par ailleurs, Kibana est également utilisé pour réaliser des requêtes complexes et afficher les résultats sous des formes plus compréhensibles et intelligibles. En outre, Kibana dispose également de fonctionnalités de recherches avancées pour affiner, filtrer, et agréger les données stockées dans la base Elasticsearch.

3.4 Logstash

Logstash est un outil open source de traitement de données et de logs côté serveur. Logstash établie une pipeline de collecte de données et alimente la base de données Elasticsearch.
Le rôle de Logstash est de recueillir des données provenant d'une multitude de sources pour les analyser, les filtrer, les transformer, les enrichir, et enfin les transmettre à un autre système pour traitements additionnels ou stockage, dans notre cas Elasticsearch. Pour ce faire,

Logstash effectue une opération de “parsing” pour extraire les informations contenues dans les données brutes puis une opération de nettoyage pour faciliter l’analyse et la visualisation des connaissances dans Kibana. Les 3 des composants les plus importants de Logstash sont :

- Entrée : transmission des logs pour le traitement dans un format compréhensible par la machine
- Filtres : ensemble de conditions pour effectuer une action ou un évènement particulier
- Sortie : restitution des données nettoyées, filtrées et agrégées



3.5 PacketBeat

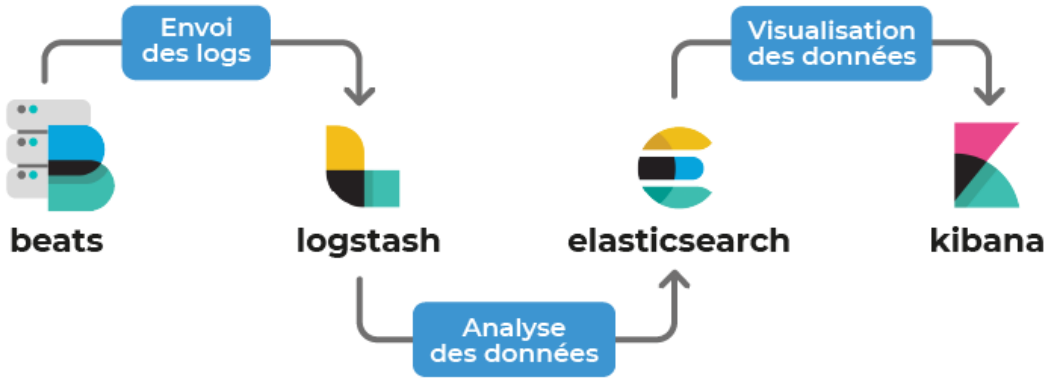
Packetbeat est un agent de la suite Beats permettant l’envoi des logs. C’est un outil de surveillance réseau qui permet de collecter des fichiers de capture réseau. L’objectif de Packetbeat est de parser les données contenues dans un fichier de capture réseau afin d’en extraire les informations importantes sous un format compréhensible pour Elasticsearch. Cet outil de pipeline de données est souvent utilisé en supplément de Logstash lorsqu’il s’agit d’effectuer des opérations de nettoyage, et d’agrégation de données plus complexes avant de pouvoir les charger dans la base Elasticsearch. Autrement dit, Packetbeat allège le travail de Logstash en effectuant le gros du travail de parsing des données.

3.6 Architecture

L’architecture ELK introduit les fonctionnalités suivantes :

- **Collecte et Agrégation** : La capacité de collecter et d’extraire auprès de multiples sources de données différentes des informations contenues dans les fichiers de capture réseau.
- **Traitement** : La possibilité de transformer les logs réseau dans un format de données compréhensible pour une analyse facile.

- **Stockage** : La possibilité de centraliser à un endroit les données collectées qui seront utilisées ultérieurement pour l'analyse et la surveillance de l'utilisation des applications à des fins de sécurité informatique.
- **Analyse** : La capacité de faire parler les données en les interrogeant et en créant des visualisations et des tableaux de bords interactifs.



4. Installations de l'environnement

4.1 Prérequis

Avant de débuter ce tutoriel, il est important de rappeler que pour aller au bout de celui-ci votre machine a besoin des éléments suivants :

- ➢ Un serveur Ubuntu 20.04 avec au minimum 4 Go de mémoire RAM et 1 processeur configuré (dans notre cas nous avons utilisé une VM)
- ➢ Installer Java sur le système d'exploitation
- ➢ Installer le serveur Web Nginx sur le serveur Ubuntu afin d'obtenir un accès contrôlé par mot de passe au dashboard Kibana

4.2 Installation

Dans cette partie, l'objectif est de détailler les différentes étapes d'installation de chaque composante de la stack ELK avec notamment la configuration des ports qui seront utilisés par chacune d'elles. Ces composantes sont les suivantes :

- Elasticsearch
- Kibana
- Logstash
- Packetbeat

4.2.1 ElasticSearch

On commence par importer la clé publique GPG d'ElasticSearch, cela ajoute un repository Elasticsearch qui permet d'accéder à tous les logiciels open-source de la stack ELK :

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
```

Ensuite on ajoute le repository Elasticsearch à la liste repository du système, on met à jour les packages du système, et on installe ElasticSearch :

```
echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a
/etc/apt/sources.list.d/elastic-7.x.list
sudo apt-get update
sudo apt-get install elasticsearch
```

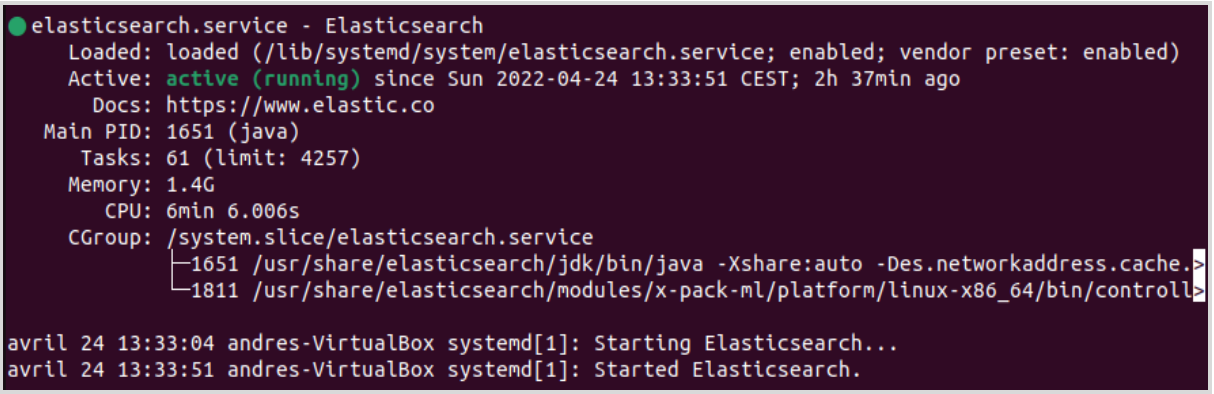
On modifie le fichier de configuration *elasticsearch.yml* pour définir le port utilisé par Elasticsearch mais également le format du cluster (single-node ou multi-node) :

```
sudo nano /etc/elasticsearch/elasticsearch.yml
```

On lance ensuite Elasticsearch :

```
sudo systemctl start elasticsearch
sudo systemctl status elasticsearch
```

output :



On vérifie qu'Elasticsearch soit bien en cours d'exécution en envoyant la requête HTTP suivante, cela nous permet également de voir les informations de base sur notre noeud local :

```
curl -X GET "localhost:9200"
```

output :

```
andres@andres-VirtualBox:~$ curl -X GET "localhost:9200"
{
  "name" : "andres-VirtualBox",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "Ew_kDHE1QDWApqoQXFt22A",
  "version" : {
    "number" : "7.17.3",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "5ad023604c8d7416c9eb6c0eadb62b14e766caff",
    "build_date" : "2022-04-19T08:11:19.070913226Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

Maintenant qu'Elasticsearch est opérationnel, installons Kibana le prochain composant de la stack ELK.

4.2.2 Kibana

➔ À noter que selon la documentation officielle, l'installation de Kibana doit se faire après celle d'Elasticsearch, car cet ordre garantit que les composants dont chaque produit dépend sont correctement en place.

L'installation de Kibana est assez similaire :

```
sudo apt-get install kibana
```

À l'instar que pour Elasticsearch on modifie le fichier de configuration *kibana.yml* :

```
 GNU nano 6.2                          /etc/kibana/kibana.yml *
server.port: 5601
server.host: "localhost"
elasticsearch.hosts: ["http://localhost:9200"]
```

On lance ensuite Kibana :

```
sudo systemctl start kibana
sudo systemctl status kibana
```

```
● kibana.service - Kibana
     Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2022-04-30 12:23:08 CEST; 2 days ago
       Docs: https://www.elastic.co
   Main PID: 26237 (node)
      Tasks: 11 (limit: 4257)
     Memory: 239.1M
        CPU: 15min 48.538s
     CGroup: /system.slice/kibana.service
             └─26237 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/../src/cli/dist --l

avril 30 12:23:08 andres-VirtualBox systemd[1]: Started Kibana.
```

Maintenant que Kibana est configuré, on installe le composant suivant : Logstash.

4.3.2 Logstash

Installation de Logstash :

```
sudo apt-get install logstash
```

Les fichiers de configuration de Logstash se trouvent dans le répertoire /etc/logstash/conf.d. Logstash étant une pipeline qui prend les données en entrée, les traite, et les envoie à sa destination (dans notre cas la destination est Elasticsearch), elle comporte deux éléments importants, une entrée et une sortie. Pour ce faire, on commence par créer un fichier de configuration *02-beats-input.conf* dans lequel on y configure le port d'entrée de la pipeline (dans notre cas c'est celui utilisé par Packetbeat) :

```
sudo nano /etc/logstash/conf.d/02-beats-input.conf
```

output :

```
  GNU nano 6.2                 /etc/logstash/conf.d/02-beats-input.conf *
input {

   beats {

      port => 5044

   }

}
```

On crée ensuite un nouveau fichier de configuration *30-elasticsearch-output.conf* dans lequel on configure le port de sortie (dans notre cas les données seront stockés dans Elasticsearch) :

```
sudo nano /etc/logstash/conf.d/30-elasticsearch-output.conf
```

output :

```
  GNU nano 6.2              /etc/logstash/conf.d/30-elasticsearch-output.conf *
output {
        elasticsearch {
                hosts => ["localhost:9200"]
                manage_template => false
                index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
        }
}
```

Avec ces configurations Logstash comprend qu'il doit récupérer les données collectées par Packetbeat et qu'il doit ensuite les envoyer à Elasticsearch pour le stockage des données prétraitées.

Une fois les configurations réalisées, on lance Logstash :

```
sudo systemctl start logstash
sudo systemctl status logstash
```

output :

```
● logstash.service - logstash
     Loaded: loaded (/etc/systemd/system/logstash.service; disabled; vendor preset: enabled)
     Active: active (running) since Sun 2022-04-24 16:43:08 CEST; 27s ago
   Main PID: 5876 (java)
      Tasks: 14 (limit: 4257)
     Memory: 280.4M
        CPU: 24.770s
     CGroup: /system.slice/logstash.service
             └─5876 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSweepGC -XX:CM>

avril 24 16:43:08 andres-VirtualBox systemd[1]: Started logstash.
avril 24 16:43:08 andres-VirtualBox logstash[5876]: Using bundled JDK: /usr/share/logstash/jdk
avril 24 16:43:09 andres-VirtualBox logstash[5876]: OpenJDK 64-Bit Server VM warning: Option UseC>
```

4.2.4 PacketBeat

Installation de Packetbeat :

```
sudo apt-get install packetbeat
```

On modifie le fichier de configuration *packetbeat.yml* dans lequel on commente la ligne ***output.elasticsearch*** et on décommente la ligne ***output.logstash***, en y ajoutant le port de Logstash, cela permet de spécifier à Packetbeat que les données collectées doivent être envoyé à Logstash pour que ce dernier puisse effectuer les opérations de traitement sur les données :

```
sudo nano /etc/packetbeat/packetbeat.yml
```

```
  GNU nano 6.2                          /etc/packetbeat/packetbeat.yml *
# ============================== Network device ================================

packetbeat.interfaces.device: any
packetbeat.interfaces.internal_networks:
  - private

# ================================== Flows =====================================

packetbeat.flows:
  timeout: 30s
  period: 10s

# ========================== Transaction protocols =============================

packetbeat.protocols:
- type: icmp
  enabled: true

[...]

- type: sip
  ports: [5060]

# ======================= Elasticsearch template setting =======================

setup.template.settings:
  index.number_of_shards: 1
```

```
  GNU nano 6.2                        /etc/packetbeat/packetbeat.yml *
# ================================== Dashboards =====================================

setup.dashboards.enabled: true

# =================================== Kibana ========================================

setup.kibana:
  host: "localhost:5601"

# ================================== Outputs ========================================

# ---------------------------- Elasticsearch Output ----------------------------

#output.elasticsearch:
  #hosts: ["localhost:9200"]

# ------------------------------ Logstash Output -------------------------------
output.logstash:
  hosts: ["localhost:5044"]

# ================================= Processors ======================================

processors:
```

Afin de vérifier si nos configurations fonctionnent, on lance la commande suivante :

`sudo packetbeat test output`

output :

```
[sudo] Mot de passe de andres :
elasticsearch: http://localhost:9200...
  parse url... OK
  connection...
    parse host... OK
    dns lookup... OK
    addresses: 127.0.0.1
    dial up... OK
  TLS... WARN secure connection disabled
  talk to server... OK
  version: 7.17.3
```

On crée ensuite un index dans Elasticsearch :

`sudo packetbeat setup`

output :

```
Overwriting ILM policy is disabled. Set `setup.ilm.overwrite: true` for enabling.

Index setup finished.
Loading dashboards (Kibana must be running and reachable)
Loaded dashboards
```

On peut maintenant lancer Packetbeat :

`sudo systemctl start packetbeat`

`sudo systemctl status packetbeat`

output :

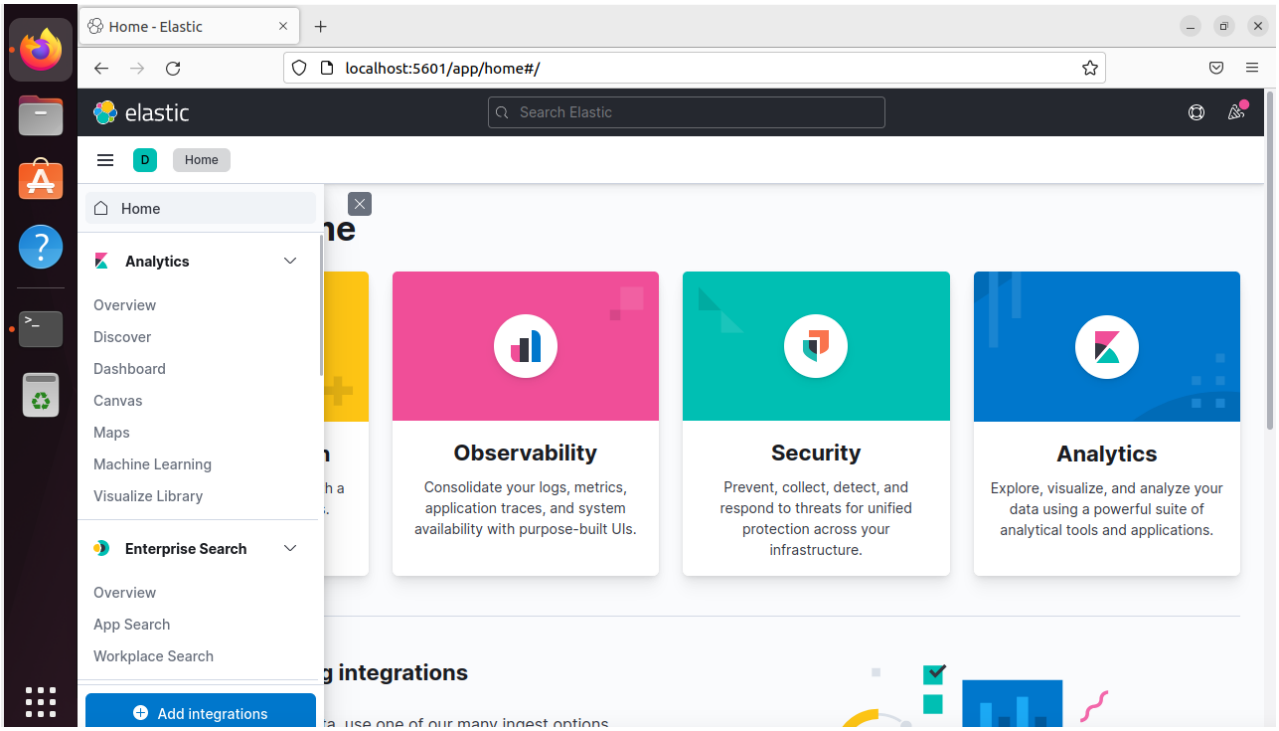
```
● packetbeat.service - Packetbeat analyzes network traffic and sends the data to Elasticsearch.
     Loaded: loaded (/lib/systemd/system/packetbeat.service; disabled; vendor preset: enabled)
     Active: active (running) since Sun 2022-04-24 17:12:58 CEST; 8s ago
       Docs: https://www.elastic.co/beats/packetbeat
   Main PID: 6191 (packetbeat)
      Tasks: 5 (limit: 4257)
     Memory: 80.1M
        CPU: 638ms
     CGroup: /system.slice/packetbeat.service
             └─6191 /usr/share/packetbeat/bin/packetbeat --environment systemd -c /etc/packetbeat/pac>

avril 24 17:13:01 andres-VirtualBox packetbeat[6191]: 2022-04-24T17:13:01.773+0200        INFO        >
avril 24 17:13:01 andres-VirtualBox packetbeat[6191]: 2022-04-24T17:13:01.775+0200        INFO        >
avril 24 17:13:01 andres-VirtualBox packetbeat[6191]: 2022-04-24T17:13:01.779+0200        INFO        >
avril 24 17:13:01 andres-VirtualBox packetbeat[6191]: 2022-04-24T17:13:01.783+0200        INFO        >
avril 24 17:13:01 andres-VirtualBox packetbeat[6191]: 2022-04-24T17:13:01.784+0200        INFO        >
avril 24 17:13:01 andres-VirtualBox packetbeat[6191]: 2022-04-24T17:13:01.786+0200        WARN        >
avril 24 17:13:01 andres-VirtualBox packetbeat[6191]: 2022-04-24T17:13:01.790+0200        INFO        >
avril 24 17:13:01 andres-VirtualBox packetbeat[6191]: 2022-04-24T17:13:01.791+0200        INFO        >
avril 24 17:13:01 andres-VirtualBox packetbeat[6191]: 2022-04-24T17:13:01.857+0200        INFO        >
avril 24 17:13:02 andres-VirtualBox packetbeat[6191]: 2022-04-24T17:13:02.141+0200        INFO        >
```

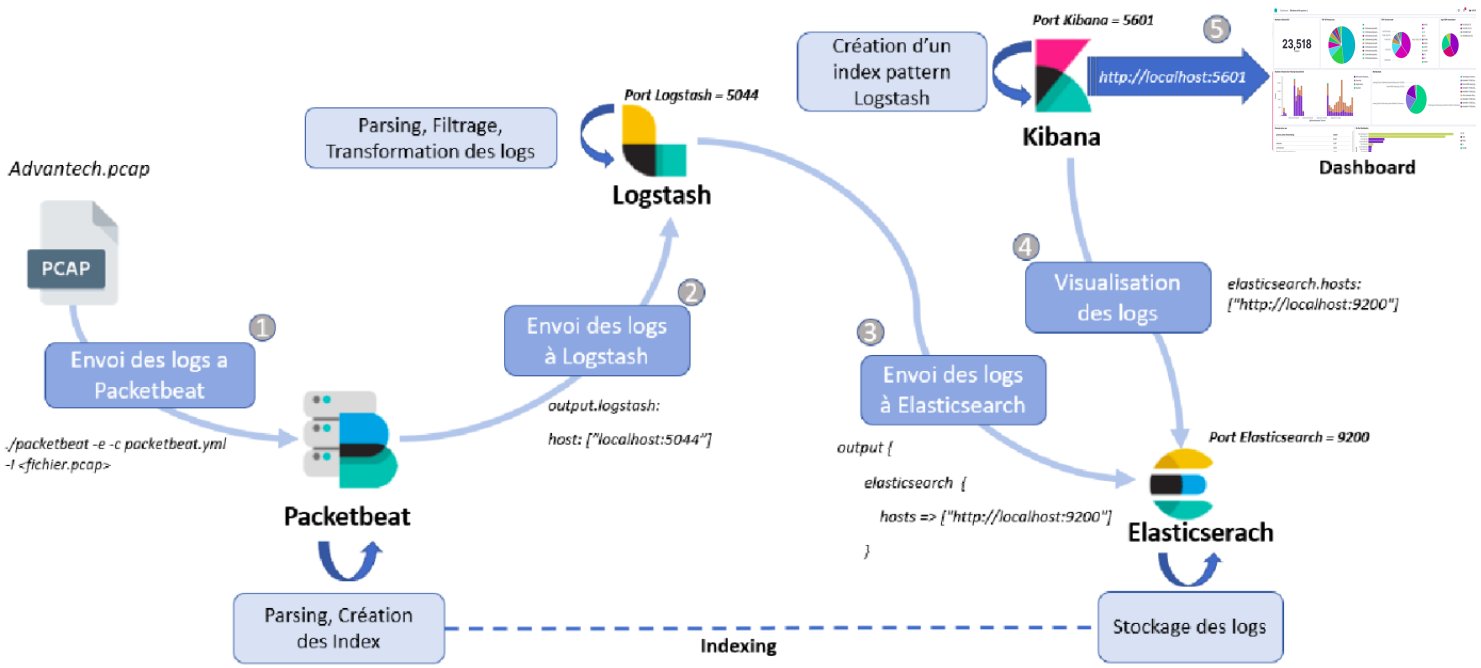
4.3 Accès à l'interface Web

Une fois que toute l'architecture est mise en place et après s'être assuré que tout fonctionne correctement, on peut accéder à l'interface web de Kibana en utilisant l'adresse URL suivante :

http://localhost:5601



5. WorkFlow des données



6. Dashboard Kibana

6.1 Fichier PCAP

L'extension .pcap de fichier est principalement associée au logiciel Wireshark qui est un programme utilisé pour l'analyse de réseaux. Les fichiers pcap sont des fichiers de données qui contiennent des paquets de données qui transitent au sein d'un réseau. Ces fichiers sont principalement utilisés pour surveiller le trafic d'un réseau et détecter des anomalies au sein d'un réseau. Dans mon cas, j'ai choisi un fichier pcap léger car les chargements de fichiers plus volumineux étés vraiment trop longs, ce fut d'ailleurs de loin l'étape la plus chronophage parmi toutes les étapes du projet.

Le fichier PCAP utilisé est disponible dans le lien suivant :
https://www.netresec.com/?page=PCAP4SICS

Pour analyser ce fichier, il nous faut d'abord le télécharger sur la machine virtuelle :

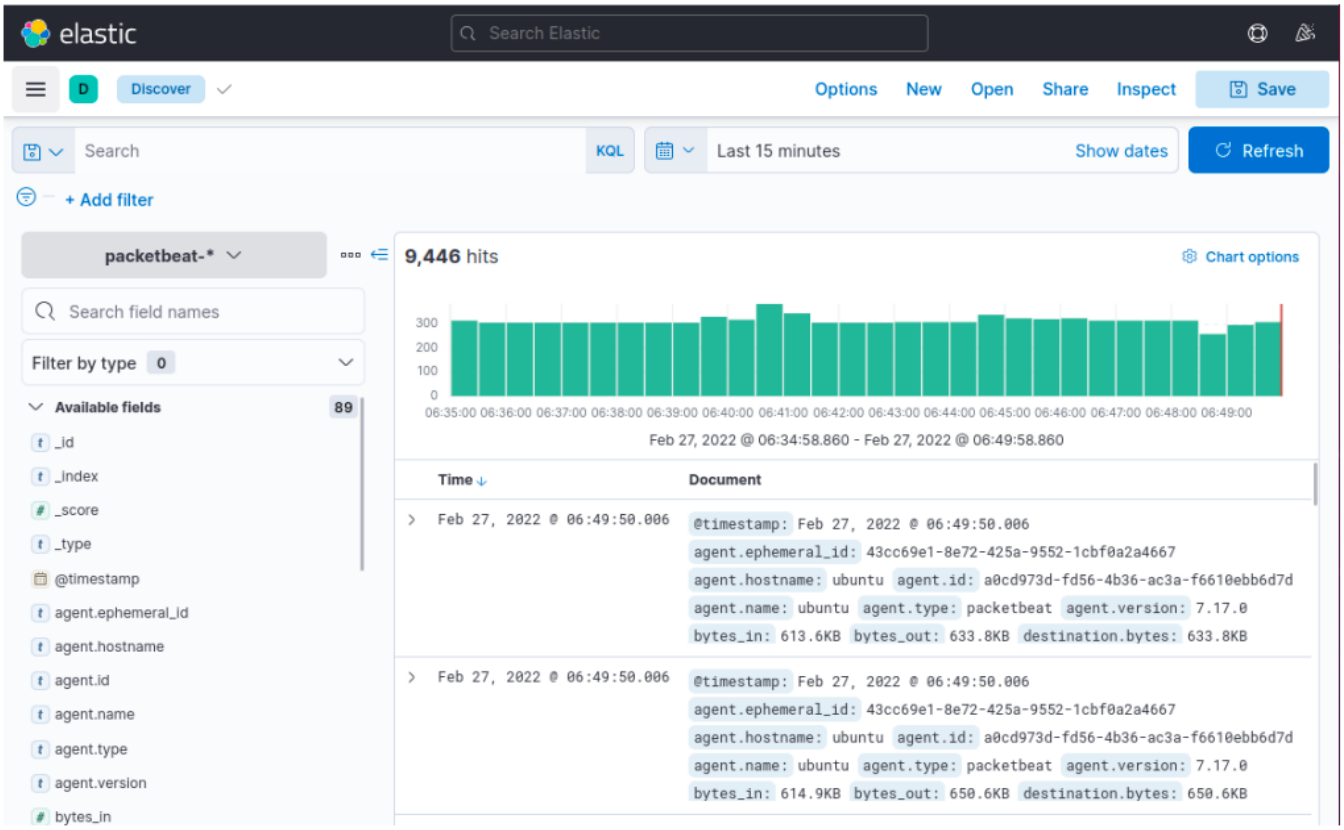
```
wget https://download.netresec.com/pcap/4sics-2015/4SICS-GeekLounge-151020.pcap
```

On demande ensuite à Packetbeat de lire le fichier de données PCAP :

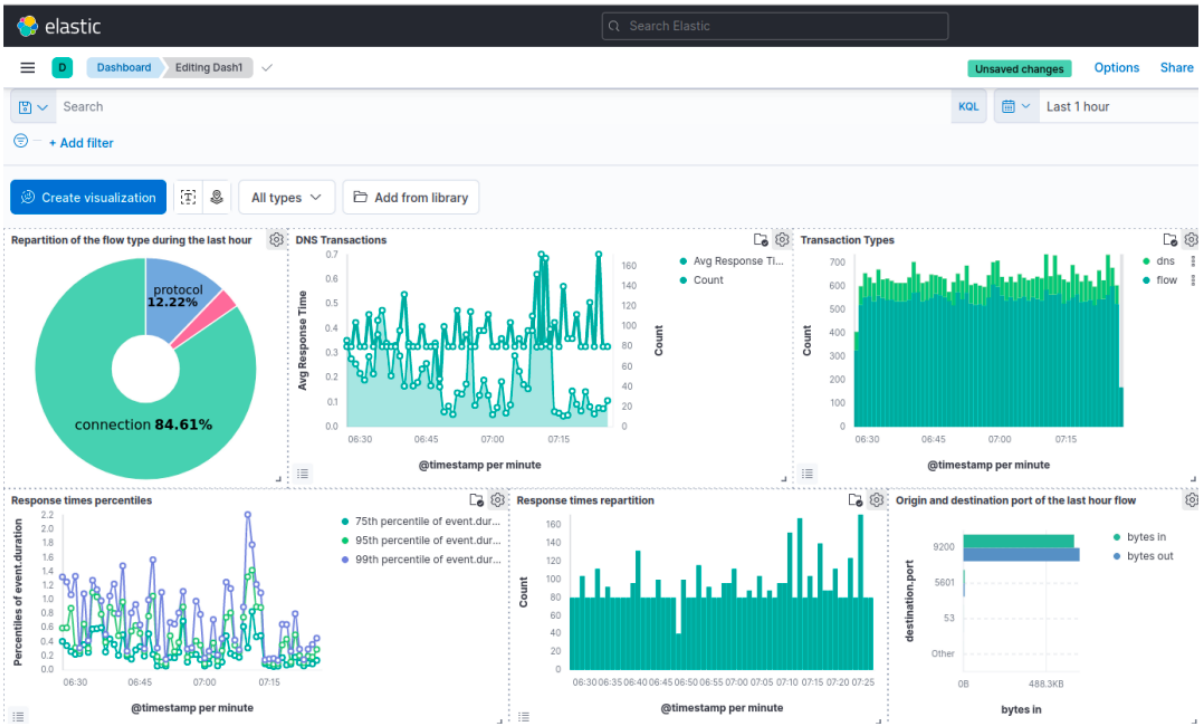
```
packetbeat -e -c /etc/packetbeat/packetbeat.yml -I 4SCIS-GeekLounge-151020.pcap -d "publish"
```

6.2 Dashboard

Une fois que Packetbeat a lu le fichier PCAP, le flux de données apparaît dans la table Discover de Kibana. L'interface permet d'explorer les données en filtrant notamment les informations pertinentes des données afin de mieux les visualiser. Voici une représentation de cet interface :



L'une des analyses que l'on peut effectuer est par exemple une analyse sur différents intervalles de temps : des temps de réponses, du type de transactions, du nombre de bytes généré etc…



7. Conclusion

Ce projet de sécurité pour le Big Data m'a permis d'apprendre à mettre en place une stack ELK pour l'analyse de fichiers réseaux. J'ai ainsi pu constater les avantages qu'offre la stack ELK pour l'analyse de grandes quantités de logs provenant de diverses sources. J'ai notamment pu réalisé les enjeux que représente la surveillance en temps réel des systèmes informatiques chez les entreprises, et à quel point la stack ELK joue un rôle très important dans ces enjeux. En outre, cela m'a également permis de constater les limites de ma machine virtuelle en termes de ressources mémoires face à la stack ELK, puisque ma machine virtuelle et ses 4 Go de RAM était clairement dépassée dans la sollicitation de ses ressources par la stack ELK. L'étape de lecture des fichiers PCAP par Packetbeat fut d'ailleurs très pénible en termes de temps puisqu'il m'aura fallu plusieurs jours pour effectuer cette étape. Finalement et malgré ces difficultés rencontrées, j'ai pu construire un Dashboard Kibana pour l'analyse des données réseaux.