7. Kibana Dashboards et Canvas

On travaille sur les logs d’accès

7.1 Création de jobs

Créer 3 jobs stockant ses résultats dans le même index dédié :

- Analyse du trafic global
  Distinct count on clientip
- Détecter les comportements suspects des clients en ce qui concerne le volume de demandes par code de réponse.
  high_count by « response.keyword » over « clientip.keyword »
- Détection des hôtes ayant un trafic anormalement haut
  *high_count over « clientip.keywor*d »

7.2 Tableau de bord

Configurer un index pattern vers l’index précédemment créé.

Con struire les visualisations suivantes :

7.2.1 Time Series Visual Builder

Dans l’onglet panel *Options*, indiquer l’index pattern et le timestamp

Dans l’onglet *Annotations*, indiquer l’index pattern et le timestamp ainsi qu’une requête isolant les anomalies critiques

Ajouter une autre annotation indiquant les anomalies majeures

7.2.2 Heat Map

Choisir l’index pattern .ml-anomalies*

1 bucket de type Date Histogram

1 sub-bucket de type term sur le job id

Metrics : Max anomaly_score

Color schema : Rouge

7.2.3 Timelion

Expression :

```
.es(index=logstash-apache*, metric=avg:bytes)
  .divide(.es(index=logstash-apache*)).yaxis(1),
  .es(index='.ml-anomalies-custom-apache*',timefield=timestamp,
```

```
metric=max:anomaly_score)
   .points(symbol=cross).yaxis(2).if(lt, 50, null)
```

Assembler ses visualisations dans un tableau de bord Kibana et ajouter des URLs dans les jobs permettant de pointer directement vers les tableaux de bord Kibana

7.3 Canvas

Démarrer un workpad

7.3.1 Data Table

Ajouter une data table et la configurer comme suit :

- *Index*: Index pattern des 3 jobs
- *Query*: job_id:<traffic_global> AND result_type:bucket
- *Sort Field*: anomaly_score
- *Sort Order*: Descending
- *Fields*: job_id , timestamp , and anomaly_score.

7.3.2 Markdown

Ajouter un *Markdown* qui affiche le nombre d'anomalies ayant un score > 80

Essayer d'appliquer du css

7.3.3 Graphique à barre

Afficher les mêmes informations dans un graphique à barre