Elasticsearch Lab 2

Objectif : In this lab, you will create and start a machine learning job on the e-commerce sample dataset. Finally you will create users with role-based access control.

1. Open the main menu in Kibana, then click **Machine Learning**. Notice that you need to start the trial license to enable all of the features. Click the **Start trial** button near the bottom of the page, which opens the **License management** page. Click **Start trial** again to start a 30-day trial.
2. Go back to the **Machine Learning** app in Kibana under the **Analytics** heading and you should see more options now. Click the **jobs** button under the **Anomaly Detection** heading on the **Overview** tab.
3. In the first step of the **Create job** wizard, select the **kibana_sample_data_ecommerce** index pattern.
4. Notice there is a preconfigured machine learning job for this dataset, but you are going to define your own. Start the **Single metric** wizard:
    1. Step 1 of the wizard allows you to select a time range for training the ML model. Click the **Use full kibana_sample_data_ecommerce data** button to use all of the existing data, then click the **Next** button.
    2. For the single metric, select **Mean(taxful_total_price)** which is the average value of the total amount of money spent on orders.
    3. Set the **Bucket span** to **1h** (which is 1 hour), then click the **Next** button.
    4. On the **Job details** step of the wizard, enter **average_total_price** for the **Job ID**. Also, add **ecommerce** to the **Groups** text field, which is a way to tag your jobs for easy filtering later (super helpful when you have dozens or hundreds of ML jobs).
    5. Click **Next** to have your setting validated, then click **Next** again to view the **Summary** of your new ML job. If everything looks correct, click the **Create job** button near the bottom of the webpage.

6. The data gets processed and the model gets built for your ML job. Click the **View results** button to continue.

7. If you view the earlier dates in your results, you will see two critical anomalies - they appear red in both the line graph and the table of anomalies.

8. Expand an anomaly in the table view to see more details about the anomaly.

9. Go back to the **Anomaly Detection** tab to view all of your defined ML jobs. Notice that your new ML job is created, but it is not running right now.

10. To start an ML job, you can click the three dots in the **Actions** column and select **Start datafeed**. (This ecommerce sample dataset is not generating new events, so you do not need to start your ML job.)

Now that the ML part is done let's look for some security options!

5. Let's start with creating a user with restricted access to your Elasticsearch cluster.

   a. Begin by creating a new role. From the Kibana navigation menu, select **Stack Management** (under Management). The Security section has options to configure users, roles and API keys. Create a new role named **read_only** that satisfies the following criteria:

      b. the user has no cluster privileges.

      c. the user has access to indices that match the pattern *.

      d. the index privileges are only read.

   e. Create a new user named **read_only_user** that satisfies the following criteria:

      f. password is "nonprodpwd"

      g. enter **Read Only User** for the name of the user

      h. use your own email address

      i. assign the user to two roles: read_only and kibana_admin

      j. Make sure to add the kibana_admin role, otherwise you won't be able to log in to Kibana with read_only_user.

    k. Log out of Kibana and login again as **read_only_user**. Navigate to the Console and run the commands below.

    l. Notice that the only successful command is the _search request, as it only reads data.

```
Unset
GET /
```

```
Unset
GET _search
```

```
Unset
PUT new_index/_doc/1
{
  "security_test": "this will fail"
}
```

6. Log out of Kibana and login again as **elastic.**
7. Now under the **Security** heading, click on **Manage,** select **Rules** and import the pre-build security detection rules of Elastic.
8. Each rule is created to match a specific abnormal or malicious security behavior.
9. Elastic rules are queries written in either Elastic Query Language or Kibana Query Language.
10. To explore how this detections work let’s do the following steps:
    a. Under the **Management** heading, click on the **Fleet** section**.**
    b. Select **Add a fleet server.**

c. Include all your localhost information then copy the code for agent installation.

d. Open a local terminal in your machine, past the command and run it.

e. An **elastic agent** will be installed and data from your system will start getting ingested to your elastic cluster.

f. In the **Discover** section under the **Analytics** heading, change the **Data view** to **logs-***. You will notice that the **Data view** contains some of your system data.

g. Now go back to the **Rules,** display the Tags list and select the one that matches your **OS** type.

h. Enable the filtered rules and if any abnormal behavior is matched in your system logs you will notice Alerts generated (to see security Alerts go to **Alerts** table)

Probably You are asking yourselves why an Alert was Triggered (if any), Well, To have the answer, Investigate the alert ^_^ !