Part 5 Examples, Instructions and Exercises

Step by step guide for creating a virtual machine with Plaso, Nginx, Elasticsearch, Logstash and Kibana followed by exercises for visualization of log2timeline data using Kibana.

Python for Data Analysis introduction

Master of Advanced Studies in Digital Forensics & Cyber Investigation

Data Analytics and Visualization for Digital Forensics

(c) Hans Henseler, 2022

1. Introduction

We are using VirtualBox from Oracle to work with a virtual machine that runs Plaso tools, Elasticsearch, Logstash and Kibana. You will need to download and install Oracle Virtual Box. Here are instructions for installing on Windows 10 :

https://download.virtualbox.org/virtualbox/6.1.36/VirtualBox-6.1.36-152435-Win.exe

After finishing the installation you should also install VirtualBox extension pack on your windows host. This executable file can be downloaded here:

https://download.virtualbox.org/virtualbox/6.1.36/Oracle_VM_VirtualBox_Extension_Pack-6.1.36a-152435.vbox-extpack

You can try step 2 first. If that works you can skip steps 3 and 4 and continue with step 5. If step 2 doesn't work (possibly because of hardware differences or network issues) I recommend to install a fresh VM. This is described in steps 3 and 4.

*** note To perform this exercise you need at least 10GB of free space on your harddrive.***

If you are using VirtualBox on a Mac or other OS please check the Oracle website for instructions.

2. Create VM based on the davday3.ova appliance

With these steps you can create the davday3 vm based on an appliance file (davday3.ova):

1. Download the davday3.ova appliance and mus2019ctf_2021.plaso file (version of mus2019ctf created with older version of plaso). Both files can be downloaded via a two-step process. First access the following link:
   https://drive.google.com/drive/folders/1TTJGxGCFBdtYGCH9crdATFlanOrIHEzj?usp=sharing
2. You have no permission to access this folder. If you Ask Permission, the owner (me) will receive an email and I will grant you permission. **Please do this on Tuesday** so you have enough time to download the .ova file. Once you have received permission, go ahead and download the file.
3. Start virtualbox (Download and install virtual box if not already done so)
4. Under the file menu select import appliance. Select the davday3.ova file. Click next. Check location. Click import. This will take 1-5 minutes.
5. Start the davday3 vm and login with user plaso and password plaso

6. start a web browser on you windows (host) and go to the following url:
   http://127.0.0.1:8080/
7. login with user kibanaadmin and password pyenv

If you can login that means the VM is working as expected and you can continue with the exercises in section 3. If the VM is not working you can try to create a new VM following the instructions in section 4 and 5 below.

3. Exercises

When you get to this point you should have a working ELK VM. Let's start loading some data. We can do that via the web interface but we can also use psort.py. In order to use psort.py we need to get a plaso file on the VM. This can be one of the plaso files you generated on day 2 in Exercise 3.

3.1 Access the Kibana webinterface

Open a browser at the VM host (your computer) and enter:

http://127.0.0.1:8080

The browser will prompt you for credentials. You can login with user kibanaadmin and password kibana. Skip the elasticsearch dialog and you should see the following:
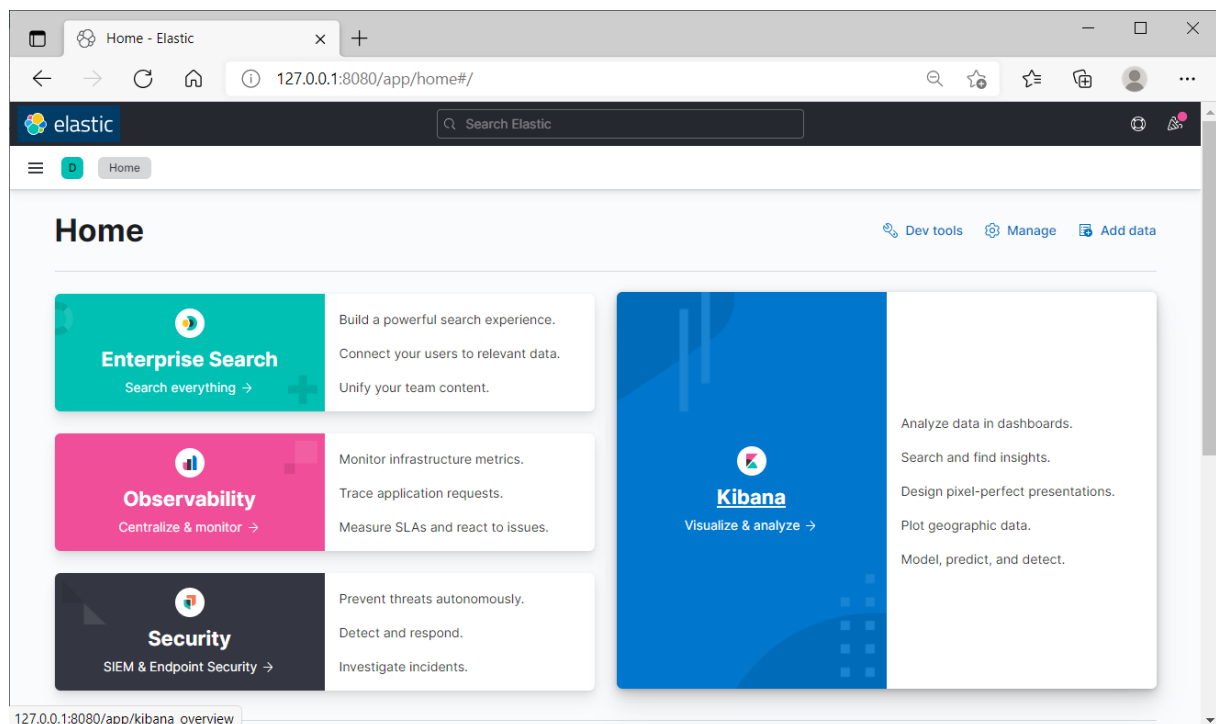
Click on Kibana in the blue area on the right. In the Kibana dashboard click "Add your data". Then select "Index management". You will then see that there are no indexes yet that you can manage. We first need to create an index with our data. GO to the next exercise 5.2.

3.2 Set up a shared folder with the host

If you want to use data is on your windows machine you need access that data. To get files on from your windows machine you can share a folder and mount it in your VM. Here is an example how to mount a folder 'shared' on your host computer in the VM. Make sure your VM is switched off.

1. Open VirtualBox

2. Right-click your VM, then click Settings
3. Go to Shared Folders section
4. Add a new shared folder
5. On Add Share prompt, select the Folder Path in your host that you want to be accessible inside your VM.
6. In the Folder Name field, type 'day3_data'
7. Uncheck Read-only and Auto-mount, and check Make Permanent
8. Start your VM
9. Create "data" directory in your home
   mkdir ~/data
10. Mount the shared folder from the host to your ~/data directory:
    sudo mount -t vboxsf day3_data ~/data
11. The host folder should now be accessible inside the VM.
12. cd ~/data
13. At the command prompt list the contents of this folder with 'ls -l'. It should be empty.
14. On the windows host: copy or move the 'mus2019ctf_2021.plaso' (that you downloaded together with davday3.ova file) to the 'day3_data' folder.
15. Repeat step 13. The 'ls -l' command should know list the 'mus2019ctf_2021.plaso' file with size is app. 430MB.
16. Check with pinfo.py the contents of the 'mus2019ctf_2021.plaso' file. Just check the sessions report:

    Pinfo.py --sections sessions mus2019ctf_2021.plaso   q

3.3 Extract events for the plaso file and output to Elasticsearch

Next step is the export of events from the plaso file into elasticsearch. Execute the following command

psort.py -o elastic --server localhost --port 9200 --elastic_mappings /usr/share/plaso/elasticsearch.mappings --index_name mus2019ctf mus2019ctf_2021.plaso
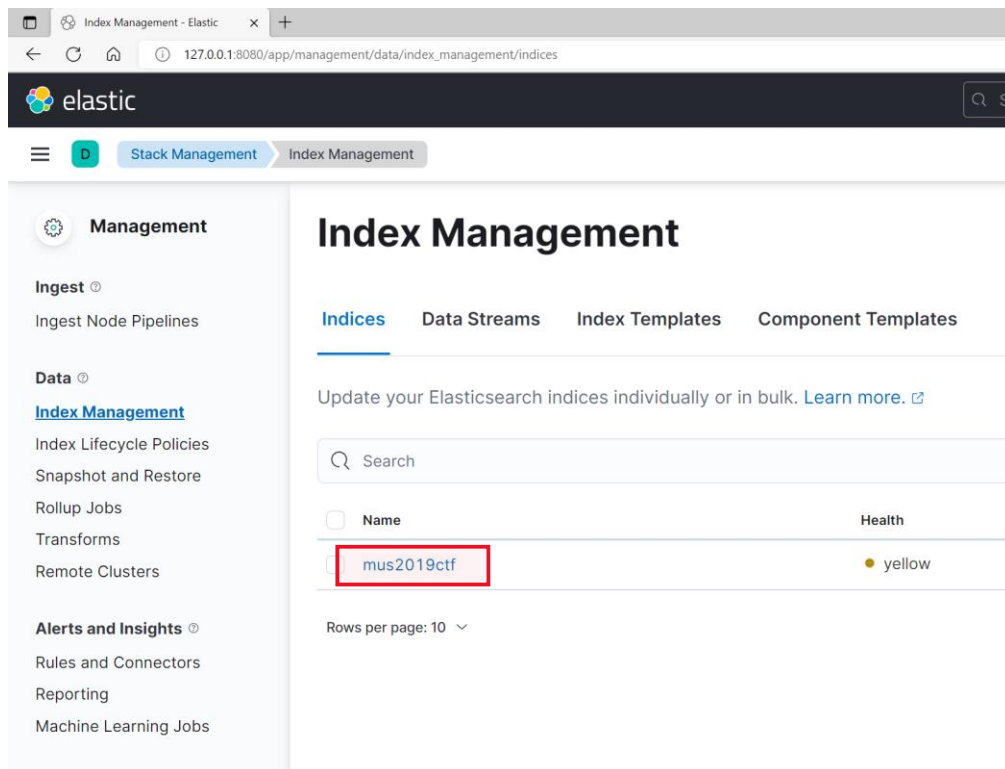

Psort.py starts running and is displaying statistics. Depending on the host computer this can take some time (10-20 minutes for 809.284 events). While Psort.py is busy, go back in Kibana. You will now see that there is an index with the name that we put in the --index_name option (mus2019ctf).

3.4 Create an index pattern

Now that the index is there, when you select "Add your data" in the Kibana screen, Kibana has detected that you have data and will prompt you to create an Index Pattern:
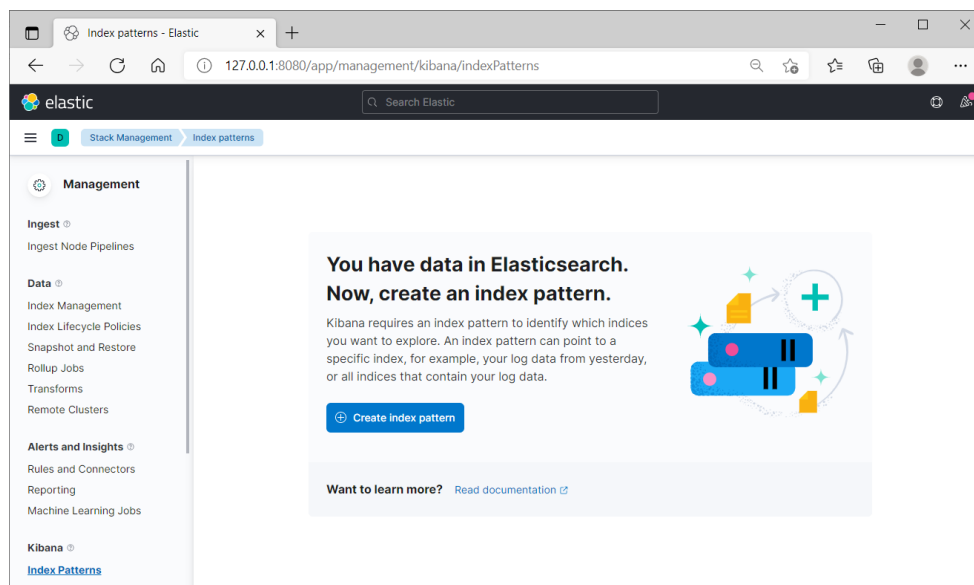

Creating an index pattern is a 2 step process:

1. Define an index pattern
   a. Type mus2019ctf in the search box. The must2019ctf index show show up. Then select next.
2. After selecting the index you have to configure settings
   a. Select the Time field. This is easy since there is only field available: datetime. Then select Create index pattern

In the next field you can edit Field properties. Note that there are 250 fields in our index! For now we won't change anything. We will leave the Management section and go back to home.

3.5 Discover data with Kibana

From the Home page dashboard (http://127.0.0.1:8080/app/home#/) we select Kibana. In Kibana we select Discover (Search and find insights). By default Kibana is filtering for recent events which are not present in our data. You can change the date filter in the upper right corner. If you click on the period you can set absolute dates. Let's pick the 12-03-2019 until 22-03-2019 period. You should now see that the windows becomes populated with data.
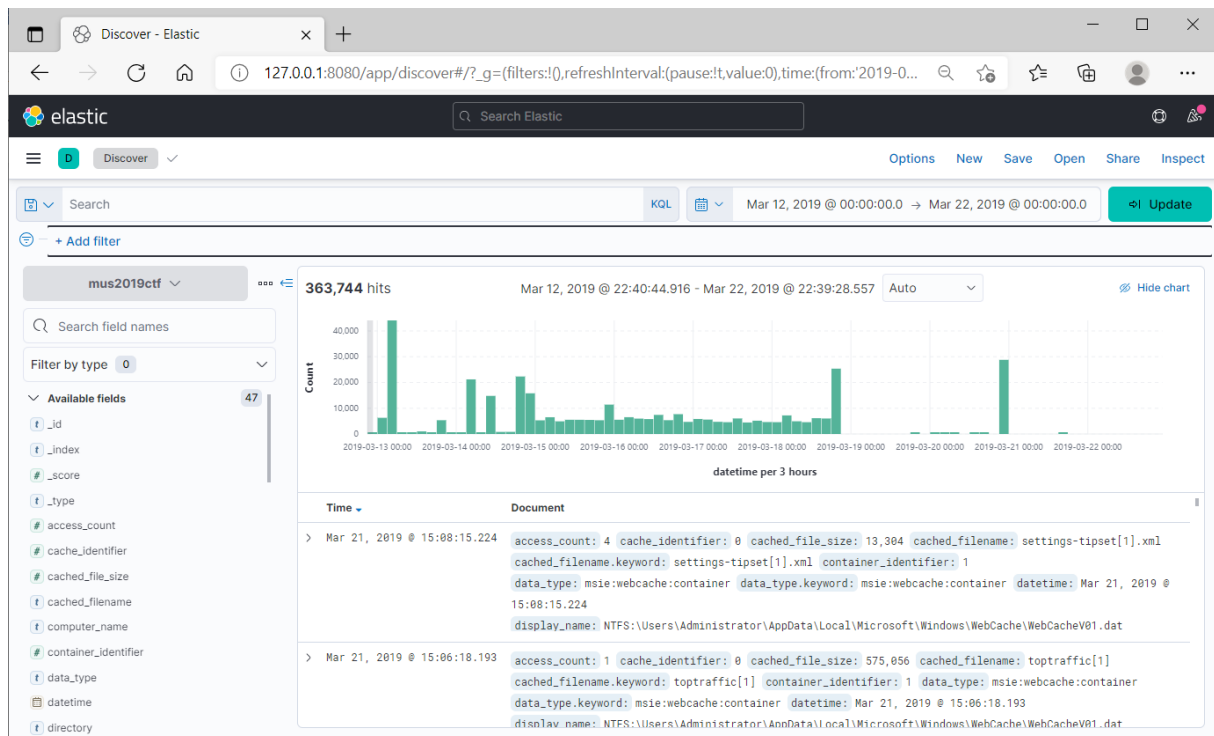

On the left side you see there are 47 different fields available in this time frame. You can search for names. Type in date and select the date_type field. There is a pop-up showing you the values for this field.
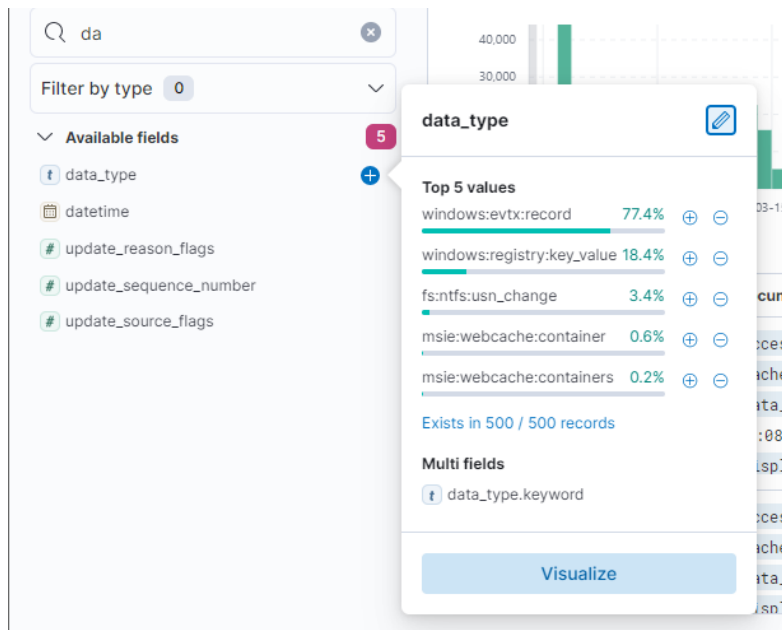
This works so much easier then the Elasticsearch json query syntax that we tried on Day 2 😉 Click on the – sign to eliminate a type from our dashboard. You will see it is added to the top left at NOT data_type: windows:evtx:record
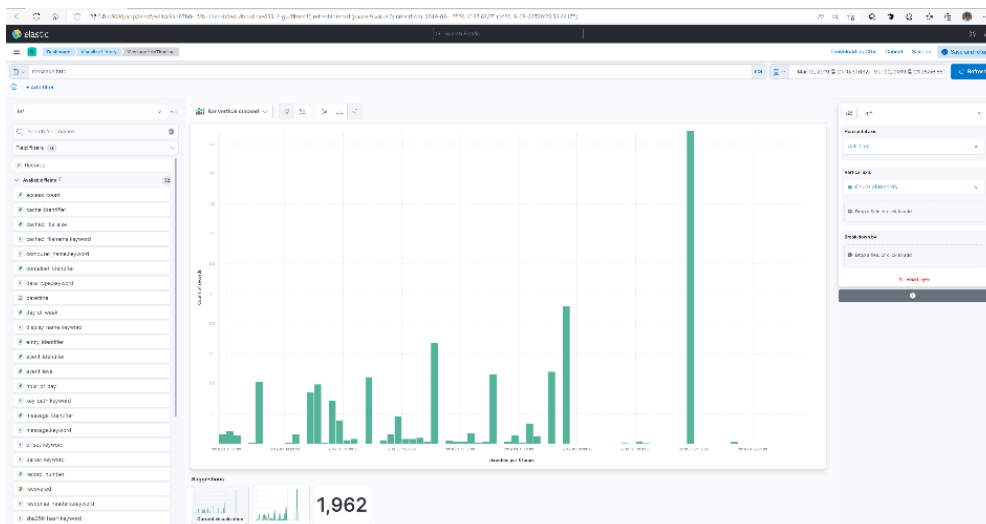
3.6 Visualize data with Kibana

Go back to Kibana and choose Dashboard. Kibana will prompt you to create your first dashboard:
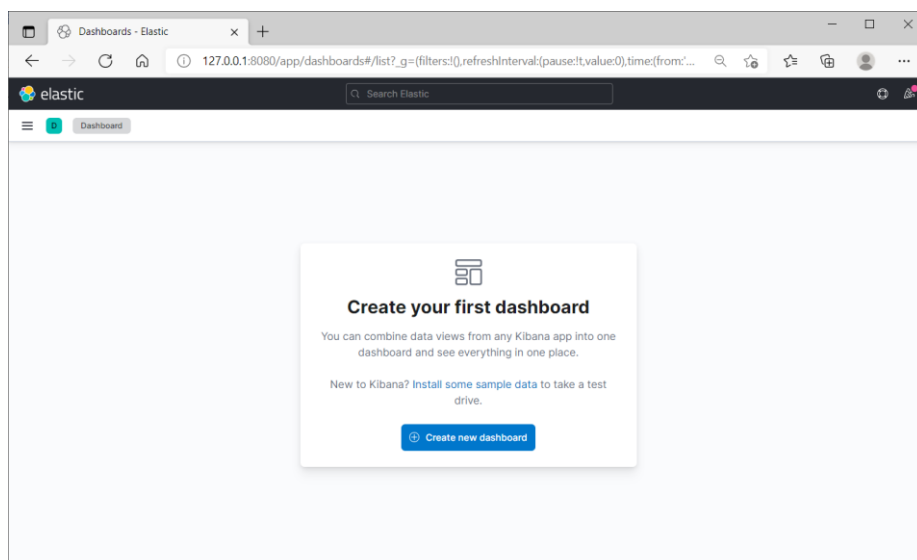
Now try to create some visualisations. Create a dashboard with:

A: Bar chart

Create a bar chart with date of time for the filtered time frame and filtered against message:http*

B: Heatmap

Create a heatmap with number of events per Day of Week vs Hour of Day. In order to do this you need to define scripted values under Index Patterns. See hint below.

Hint for Heatmap:

Scripted fields can be defined in the Index Patterns section under Management. See screenshot below:

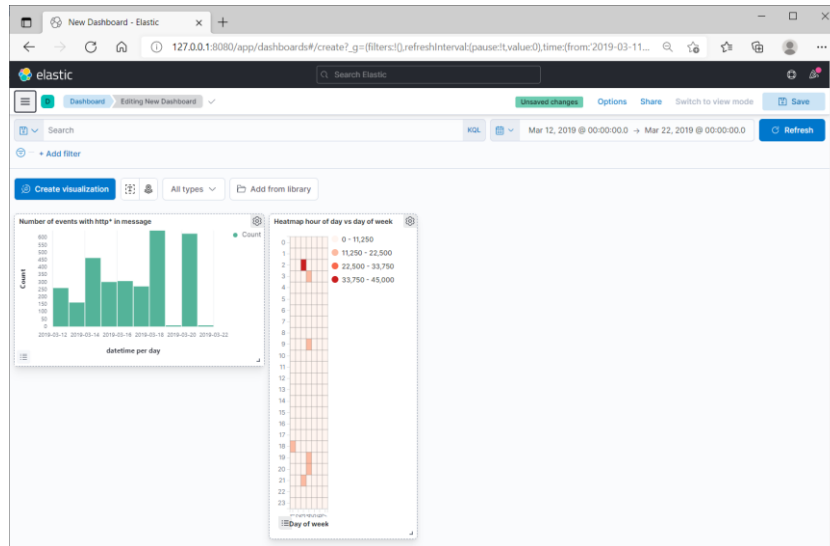The day_of_week field has the following script:

doc['datetime'].value.dayOfWeek

The hour_of_day field has the following script:

doc['datetime'].value.hourOfDay

(hint: here is web page with some explanation how to do this https://discuss.elastic.co/t/bar-chart-filtered-by-day-of-week/155743/2)

The final result can look like this:

C: Treemap

Create a treemap based on source_short and source_long with count values filtering out FILE, WinEvtx and Windows registry key. Tip: Use the Lense option in the visualization menu and in Lens choose Treemap.