Setup of SSCLANT DFRWS IoT Challenge Submission

Eoghan Casey

28 April 2018

The following steps were followed to setup the virtual environment created and submitted by the winning SSCLANT team in the DFRWS IoT Forensic Challenge. These steps follow instructions provided in the PDF report submitted by the SSCLANT team along with the tools they provided to process various data sources in the DFRWS IoT Forensic Challenge.

1) Setup an Ubuntu 16.04 virtual machine.

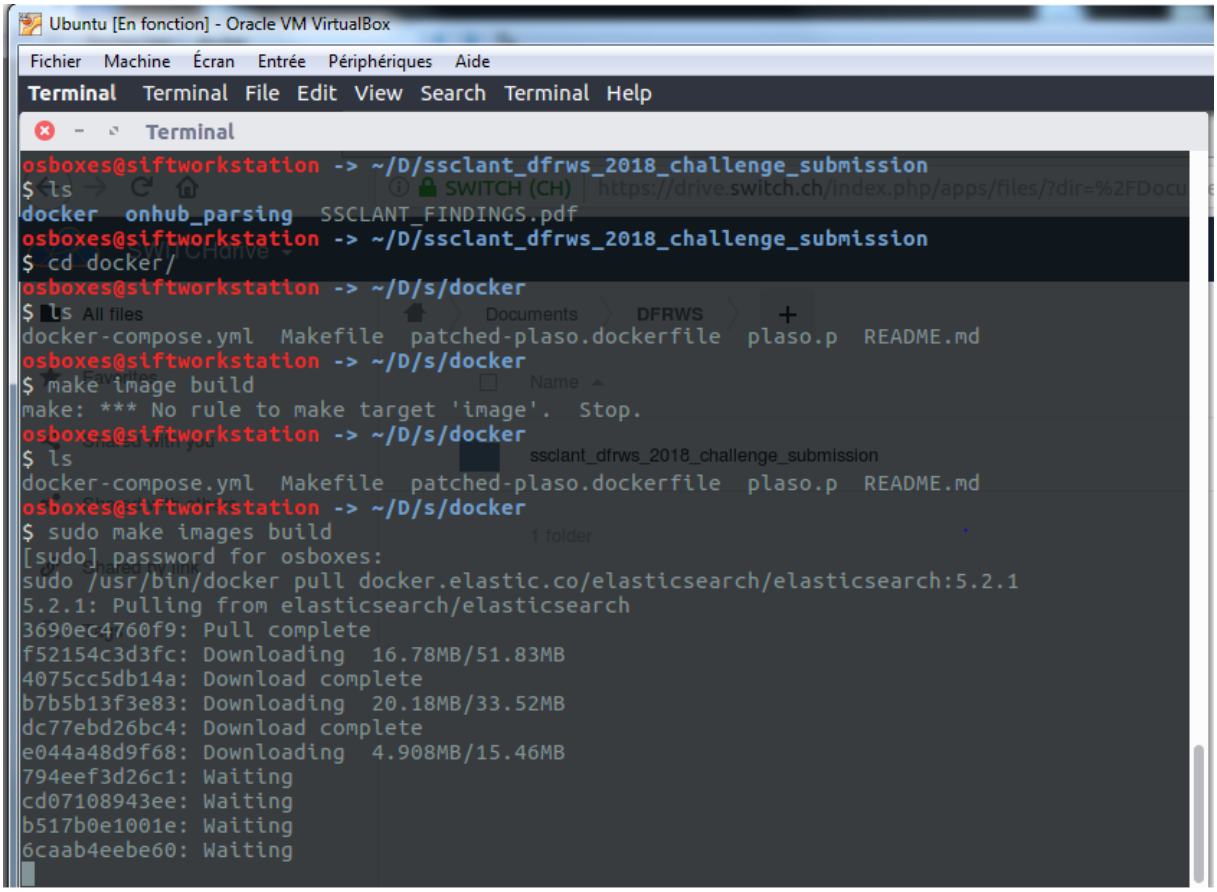
For instance, download a virtual machine with Ubuntu 16.04 from osboxes.org (https://www.osboxes.org/ubuntu/) and install the VirtualBox Extension Pack and Guest Additions.

Note: Configuring 8GB of RAM and additional video memory will enhance performance of the system.

2) Setup the SIFT workstation environment (https://github.com/sans-dfir/sift-cli/)

For example, downloading the SIFT configuration program into the Ubuntu 16.04 virtual machine (https://github.com/sans-dfir/sift-cli/releases/tag/v1.6.1)

3) Build the docker images in the SSCANT submission, as instructed in the PDF report and shown here:



4) Download and install protobufs bindings for python to support the onhub_dump.py tool.

```
wget \
https://github.com/google/protobuf/releases/download/v3.5.1/pr
otobuf-all-
3.5.1.tar.gz
tar zxf protobuf-all-3.5.1.tar.gz
cd protobuf-3.5.1
./configure
make
sudo make install
cd python
python setup.py build
sudo python setup.py install
```

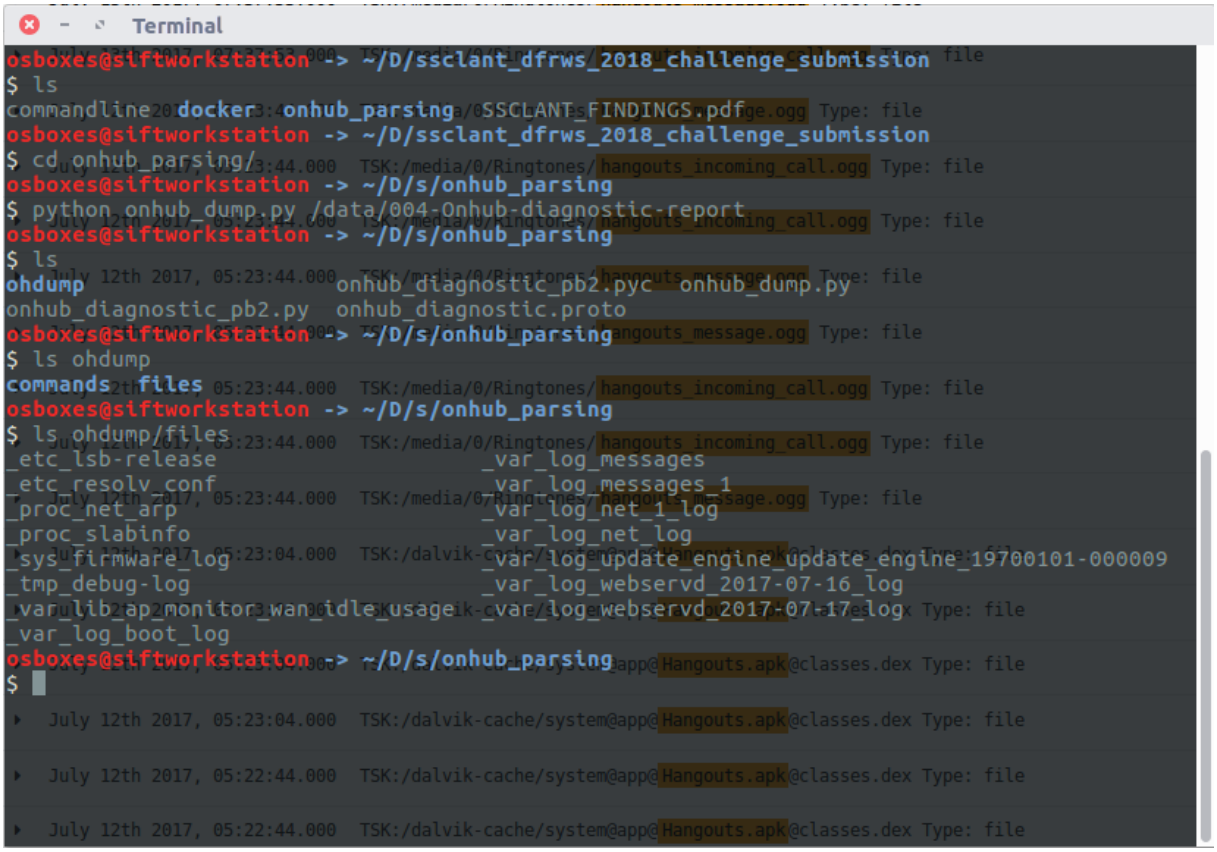
5) Mount the DFRWS IoT Forensic Challenge datasets within the virtual machine:

For example, create a Shared Folder within VirtualBox named challenge, and then mount it within the Ubuntu 16.04 virtual machine:

```
sudo mount -t vboxsf -o uid=1000,gid=1000 challenge /data
```

6) Run the script the onhub_dump.py tool to extract information from the OnHub diagnostic report.

```
python onhub_dump.py /data/004-Onhub-diagnostic-report
```



7) Run the plaso tool log2timeline to extract events from the DFRWS IoT Forensic Challenge

```
sudo docker run -v /data:/data iot-plaso:latest log2timeline
/data/dfrws-iot.plaso /data/002-BettyNote2Black/SHV-
E250L_Physical_20170717/SHV-
E250L_Physical_20170717_USERDATA.mdf

sudo docker run -v /data:/data iot-plaso:latest log2timeline -
- partitions all /data/dfrws-iot.plaso /data/001-SmartTV-
RaspberryPi/E001SmartTVMMC.000
```

Elastic and Kibana

8) Set `vm.max_map_count = 262144` in `/etc/sysctl.conf` as instructed in the docker README to support Elastic.

9) Start Elastic and Kibana using the Makefile provided in the docker folder:

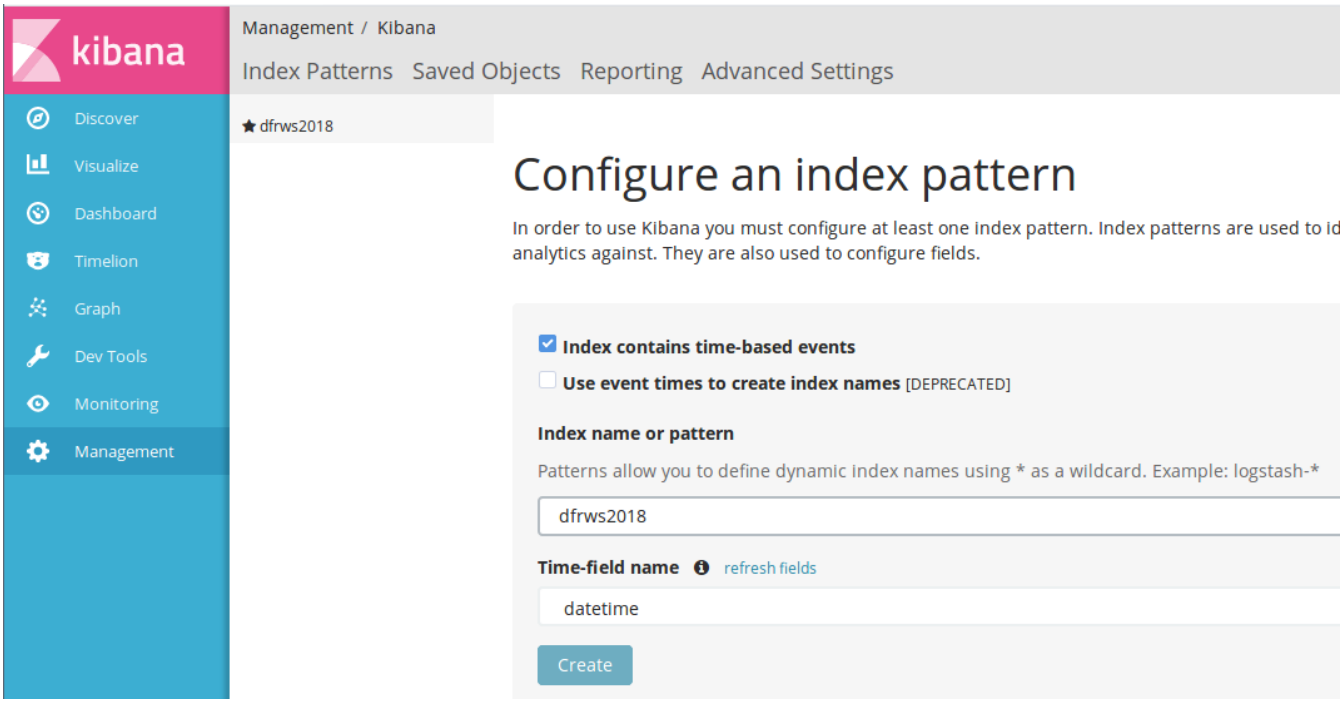
```
cd docker
make elastic-start
```

10) Run plaso to load events extracted using log2timeline into the Elastic database, specifzing the server IP address as the docker virtual switch IP address (172.17.0.1), the Elastic container IP address or the hosts IP address.

```
sudo docker run --rm -ti -v /data:/data iot-plaso:latest psort
-o elastic --raw_fields --index_name dfrws2018 --server
172.17.0.1 --elastic_user elastic /data/dfrws-iot.plaso
```

Note: when prompted, enter the Elastic password (e.g., changeme).

11) In Kibana, use the Management area to set an Index Pattern in Kibana that corresponds with the index created when loading data in the previous step (e.g. dfrws2018).



12) In Kibana, use the Discover area to select the time frame in the top right (e.g., Last 2 years) and explore the data.