

## 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.

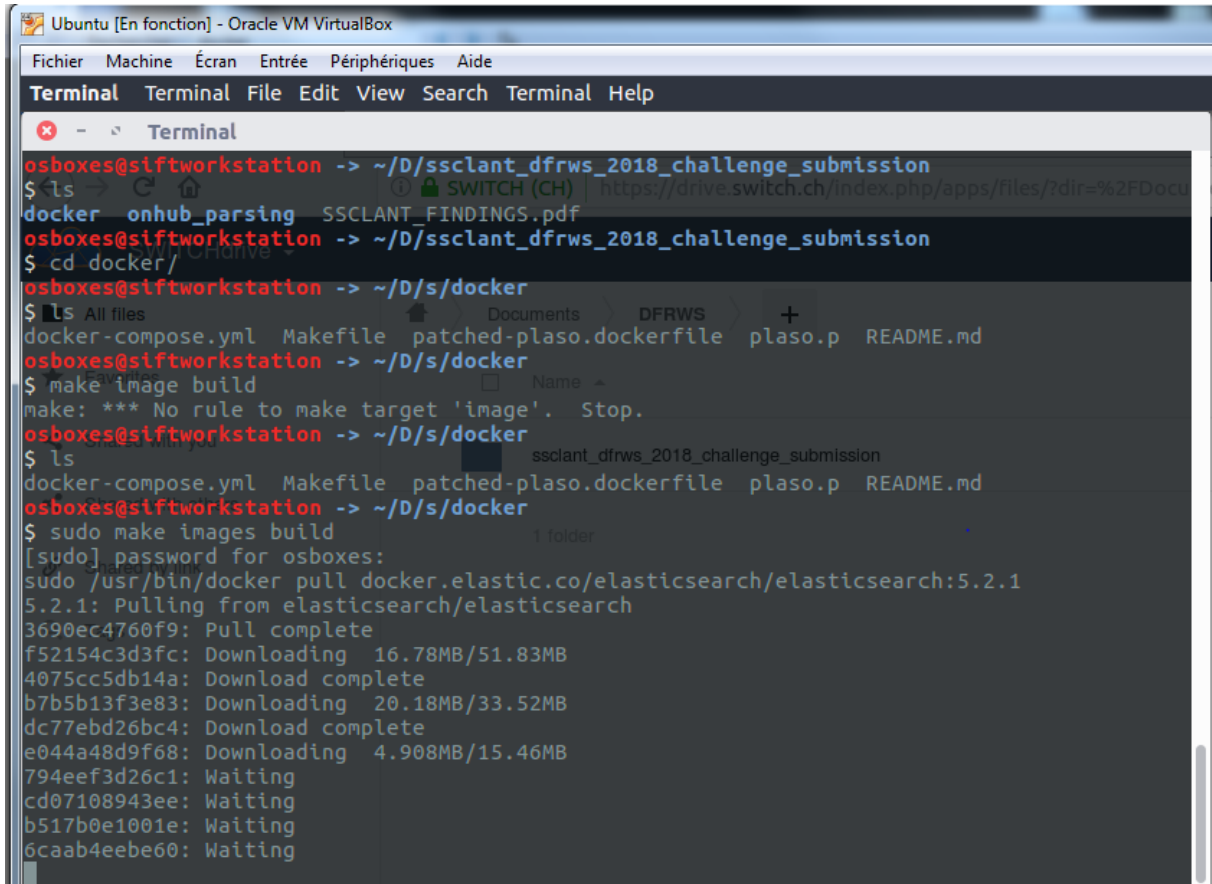
**Note:** Configuring 8GB of RAM and additional video memory will enhance performance of the system.

### 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:



```
osboxes@siftworkstation -> ~/D/ssclant_dfrws_2018_challenge_submission
$ ls
docker onhub_parsing SSCLANT_FINDINGS.pdf
osboxes@siftworkstation -> ~/D/ssclant_dfrws_2018_challenge_submission
$ cd docker/
osboxes@siftworkstation -> ~/D/s/docker
$ ls
docker-compose.yml Makefile patched-plaso.dockerfile plaso.p README.md
osboxes@siftworkstation -> ~/D/s/docker
$ make image build
make: *** No rule to make target 'image'. Stop.
osboxes@siftworkstation -> ~/D/s/docker
$ ls
docker-compose.yml Makefile patched-plaso.dockerfile plaso.p README.md
osboxes@siftworkstation -> ~/D/s/docker
$ sudo make images build
[sudo] password for osboxes:
sudo /usr/bin/docker pull docker.elastic.co/elasticsearch/elasticsearch:5.2.1
5.2.1: Pulling from elasticsearch/elasticsearch
3690ec4760f9: Pull complete
f52154c3d3fc: Downloading 16.78MB/51.83MB
4075cc5db14a: Download complete
b7b5b13f3e83: Downloading 20.18MB/33.52MB
dc77ebd26bc4: Download complete
e044a48d9f68: Downloading 4.908MB/15.46MB
794eef3d26c1: Waiting
cd07108943ee: Waiting
b517b0e1001e: Waiting
6caab4eebe60: Waiting
```

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 xzf 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
```

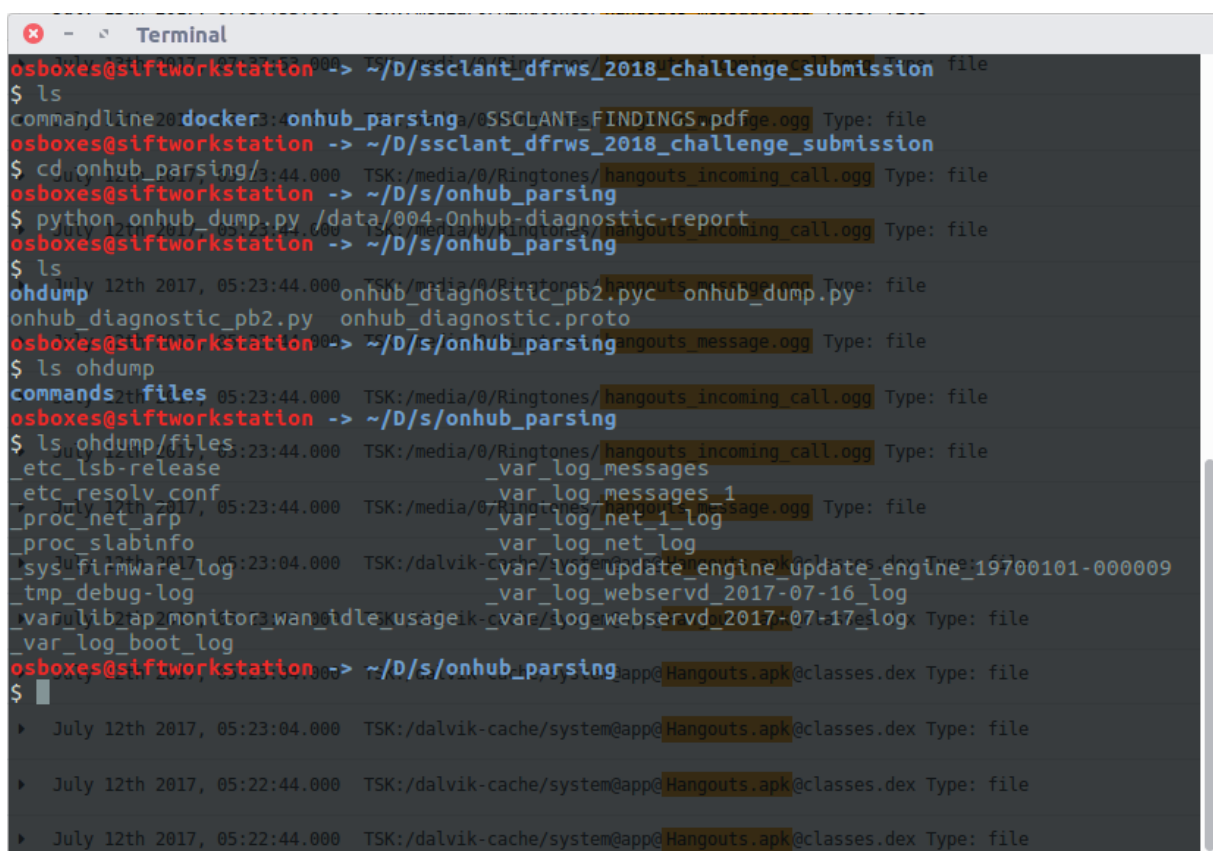
5) Mount the DFRWS IoT Forensic Challenge datasets within the virtual machine:

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
```



```
osboxes@siftworkstation -> ~/D/ssclant_dfrws_2018_challenge_submission
$ ls
commandline20170717_05:23:44.000 onhub_parsing_05:23:44.000 FINDINGS.pdf
osboxes@siftworkstation -> ~/D/ssclant_dfrws_2018_challenge_submission
$ cd onhub_parsing/
osboxes@siftworkstation -> ~/D/s/onhub_parsing
$ python onhub_dump.py /data/004-Onhub-diagnostic-report
osboxes@siftworkstation -> ~/D/s/onhub_parsing
$ ls
ohdump onhub_diagnostic_pb2.pyc onhub_dump.py
onhub_diagnostic_pb2.py onhub_diagnostic.proto
osboxes@siftworkstation -> ~/D/s/onhub_parsing
$ ls ohdump
commands files
osboxes@siftworkstation -> ~/D/s/onhub_parsing
$ ls ohdump/files
_etc_lsb-release _var_log_messages
_etc_resolv.conf _var_log_messages_1
_proc_net_arp _var_log_net_1_log
_proc_slabinfo _var_log_net_log
_sys_firmware_log _var_log_update_engine_update_engine_19700101-000009
_tmp_debug-log _var_log_webservd_2017-07-16_log
_var_lib_app_monitor_wan_idle_usage _var_log_webservd_2017-07-17_log
_var_log_boot_log
osboxes@siftworkstation -> ~/D/s/onhub_parsing
$
```

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, specifying 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).

The screenshot shows the Kibana web interface. On the left is a sidebar with navigation links: Discover, Visualize, Dashboard, Timelion, Graph, Dev Tools, Monitoring, and Management (which is highlighted). The main content area has a header 'Management / Kibana' and a sub-header 'Index Patterns'. Below this, there's a list of index patterns with 'dfrws2018' selected. The main heading is 'Configure an index pattern'. Below the heading is a description: 'In order to use Kibana you must configure at least one index pattern. Index patterns are used to id analytics against. They are also used to configure fields.' There are two checkboxes: 'Index contains time-based events' (checked) and 'Use event times to create index names [DEPRECATED]' (unchecked). Below these is the 'Index name or pattern' section with a text input field containing 'dfrws2018'. Underneath is the 'Time-field name' section with a text input field containing 'datetime' and a 'refresh fields' link. At the bottom is a 'Create' button.

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.

The screenshot shows the Kibana Discover interface. The left sidebar contains navigation links: Discover, Visualize, Dashboard, Timelion, Graph, Dev Tools, Monitoring, and Management. The main area displays search results for the query 'hangouts\*'. The top right shows the time range 'Last 2 years' and a 'Go' button. The search results are displayed as a bar chart and a list of messages.

**Search Query:** hangouts\*

**Time Range:** From: April 28th 2016, 15:41:04.596 To: Now

**Selected Fields:** t\_message

**Available Fields:** t\_id, t\_index, #\_score, t\_type, ? body, t\_data\_type, @ datetime, t\_display\_name, # file\_entry\_type, # file\_size, t\_file\_system\_type, t\_filename, # inode, @ is\_allocated

**Bar Chart:** The chart shows the count of messages over time, with a peak around July 2017. The x-axis is labeled 'datetime per week' and the y-axis is labeled 'Count'.

**Message List:**

Time	message
July 17th 2017, 06:05:40.000	Sender: John Macron Body: Who the fuck do you think k you are!
July 17th 2017, 06:05:25.000	Sender: Hallym Betty Body: Its over dont msg me
July 17th 2017, 06:05:22.000	Sender: John Macron Body: ...
July 17th 2017, 06:05:10.000	Sender: Hallym Betty Body: I cannot take it anymore
July 17th 2017, 06:04:45.000	Sender: John Macron Body: You6#39;re just paranoid....
July 17th 2017, 06:04:37.000	Sender: Hallym Betty Body: It feels like they are warching us
July 17th 2017, 06:04:19.000	Sender: Hallym Betty Body: Ewryone suspectsbs