COMP90073 - Assignment 2

Machine learning based cyberattack detection

Daniel Gil <Student Id: 905923>

First, import the necessary packages to load and process data

In [4]:

```
import pandas as pd
import sys
sys.path.append("../src")
import cconfig
import utils
```

The iForest model seems to be a good choice to analyze final results, the isolation technique was analyzed as a good approach for real-time applications and results shows consistency across both type of flows. To generate the final result the bi-directional flow was chosen in order to identify clearly the attackers and victims.

First, we can load the generated output of anomalies for this particular algorithm and type of low:

In [17]:

```
df=utils.load("../outputs/BiFlow/BIFLOW_df_anomalies_iforest")
df=df.sort_values(by=['flow_start'])
```

Since we are interested in attacker and victim, we can create a new column in data to consolidate easier and get the most frequent host participating in anomalies:

In [18]:

```
df['conversation']=df[['src_ip', 'dst_ip']].apply(lambda x: '->'.join(x), axis=1
)
df.conversation.value_counts()[:15]
```

Out[18]:

```
162.242.240.67->192.168.10.50     10208
169.54.33.166->192.168.10.50       8556
146.20.128.223->192.168.10.50      8427
146.20.128.189->192.168.10.50      8350
151.101.192.194->192.168.10.50     8304
13.59.43.55->192.168.10.50         8253
104.16.26.35->192.168.10.50        7911
13.58.146.190->192.168.10.50       5136
104.117.102.33->192.168.10.50      4584
107.22.224.100->192.168.10.50      4020
104.97.137.26->192.168.10.50       3962
192.168.10.16->199.244.48.55        384
199.244.48.55->192.168.10.16        383
162.213.33.50->192.168.10.51        354
192.168.10.17->199.167.65.25        209
Name: conversation, dtype: int64
```

The top eleven account for the majority of the frequencies, it can be seen the host 19.216.10.50 as the victim and several hosts as attacker. Now let's filter out the top 11 and consolidate the data for final result using the final columns required as output

In [19]:

```
selected_columns=['flow_start','flow_finish','src_ip','dst_ip','src_port','dst_p
ort','protocol','tcp_stream','conversation']
df=df[selected_columns]
df.head()
```

Out[19]:

	flow_start	flow_finish	src_ip	dst_ip	src_port	dst_port	protocol	tc
0	2017-07-07 17:00:00.014	2017-07-07 17:01:15.253000	192.168.10.17	172.217.10.34	43060	443	TCP	
2	2017-07-07 17:00:00.026	2017-07-07 17:01:45.255000	172.217.10.46	192.168.10.17	80	55848	TCP	
5	2017-07-07 17:00:00.085	2017-07-07 17:00:35.453000	192.168.10.19	151.101.118.2	42918	443	TCP	
8	2017-07-07 17:00:00.148	2017-07-07 17:05:41.280000	192.168.10.16	172.217.10.130	38518	443	TLSv1.2	
7	2017-07-07 17:00:00.202	2017-07-07 17:05:45.649000	192.168.10.16	172.217.10.130	38518	443	TCP	

In [22]:

```
# create a ddos file with all malicious conversations and the required fields
malicious_conversations=df.conversation.value_counts()[:11]
ddos_conversations=list()
ddos_summary=list()
for index,row in df.iterrows():
    if row.conversation in malicious_conversations.index:
        ddos_conversations.append(row)
df_ddos = pd.DataFrame(ddos_conversations)
```

In [35]:

```
df_ddos.head()
```

Out[35]:

	flow_start	flow_finish	src_ip	dst_ip	src_port	dst_port	protocol	tc
6892	2017-07-07 17:14:05.497	2017-07-07 17:14:11.505	107.22.224.100	192.168.10.50	35550	80	TCP	
6896	2017-07-07 17:14:11.541	2017-07-07 17:14:11.876	107.22.224.100	192.168.10.50	35552	80	TCP	
6898	2017-07-07 17:14:11.877	2017-07-07 17:14:11.878	107.22.224.100	192.168.10.50	35554	80	TCP	
6911	2017-07-07 17:14:17.054	2017-07-07 17:14:23.060	107.22.224.100	192.168.10.50	35560	80	TCP	
6913	2017-07-07 17:14:23.099	2017-07-07 17:14:23.486	107.22.224.100	192.168.10.50	35562	80	TCP	

In [24]:

```
# save results in file

df_ddos.to_csv("../outputs/ddos.csv")
```