# Network Security

Project 2: Log Analysis

0858612 魏嘉豪

**Implementation:**

**(1) Data Preprocessing**

**Step 1**

Parse the Json files, including the winlogbeat.json and packetbeat.json.

**Step 2**

Observe the logs to see what may contain the information related to the user behavior and benefits distinguishing between users.

**Step 3**

After extracting datum, I found that each user has a different distribution of **"event ID"** s. So I decided to train our model according to the **"event ID"** . For every users, I extract the event IDs from the winlogbeat.json and calculate the probability of each event's occurrence. There are 35 types of event IDs as follow.

```
User 1's event distribution:
[(4624, 11), (4656, 4569), (4658, 2167), (4663, 919), (4672, 11), (4688, 79), (4689, 78), (4690, 1061), (4702, 1), (4703, 22), (47
98, 1), (4799, 4), (5379, 26), (7040, 4), (10016, 1)]

User 2's event distribution:
[(4624, 4), (4656, 5859), (4658, 4533), (4663, 2135), (4672, 4), (4688, 58), (4689, 56), (4690, 2244), (4703, 15)]

User 3's event distribution:
[(26, 1), (4624, 4), (4656, 943), (4658, 1855), (4660, 2), (4663, 731), (4670, 8), (4672, 4), (4688, 79), (4689, 75), (4690, 919),
(4698, 1), (4703, 28), (4798, 5), (5156, 357), (5158, 167), (7045, 1)]

User 4's event distribution:
[(15, 2), (4624, 19), (4625, 1), (4634, 8), (4648, 2), (4656, 3571), (4658, 7193), (4660, 4), (4663, 2632), (4672, 21), (4688, 10
6), (4689, 101), (4690, 3625), (4702, 1), (4703, 26), (4719, 6), (4798, 25), (5058, 1), (5061, 1), (5156, 790), (5158, 146), (537
9, 1), (5381, 2), (5382, 5), (6416, 17), (10016, 1), (16384, 1), (16394, 1)]

User 5's event distribution:
[(1001, 1), (4624, 8), (4656, 4958), (4658, 2640), (4663, 1114), (4672, 8), (4688, 74), (4689, 81), (4690, 1299), (4703, 22), (479
9, 2), (5379, 13), (5382, 3)]
```

**(2) Model**

I apply a simple classifier model based on pytorch framework, consisting of two fully-connected layers and a ReLU activation function. I use Adam with default settings except the learning rate setup as 0.01 as the optimizer and apply Cross Entropy as the loss function. The input of our model is a set of normalized event distributions of the following shape (batch_size, num_events) where variable

num_events represents how much event types the total training set contains.The output of our model is the likelihood estimation of shape (batch_size, 5). Then the prediction of test case i is determined by finding the index of the biggest value upon output[i].

**Problems or interesting things found in process:**

**(1) Problems:**

In the testing data, there are 37 types of event IDs but there are only 35 types of event IDs in training data which cause dimension cannot match. So I decided to write another function to count testing data event ID depend on the event IDs which have appeared in training data and the dimension can match finally.

**(2) Interesting things:**

I found that the logs contains a large proportion of useless information. According to the method I implemented, only about 20% of logs are somehow meaningful which the other 80% aren't required. It is out of my expectation. In the past, I learned to gather as much information as I could while logging which causes a lot of problem for saving the big data in limited storage.