# Assignment 3:

In this assignment, I try to train a model to detect DNS entries for attempts of cyber-attacks.

The entire experiment has been divided into the following steps:

1. Data Analysis and data Engineering and Data Cleaning
2. Feature Filtering/Selection
3. Model
4. Evaluation

## Data Analysis and Data Cleaning/Engineering:

In data analysis, I looked at the imbalance in the data set. I found that the data has a slight skew but the difference is not enough for the need of balancing techniques. The Data is in a 55%-45% split which is not a lot [cell 9].

Next, I tried some preprocessing for the non-numeric values.

I found that there are 3 features which have such values - 'longest_word', 'sld', 'timestamp'.

Timestamp is one feature which doesn't give any statistical knowledge about the data so I drop it without much thought. One thing that can be considered as a future area of investigation is that whether there are certain hours of the day when the attacks are maximum and is it feasible to reduce security during those hours. Or the hour of the day can be divided into time sections and can be used as a feature.

The feature longest word has mostly numbers, but in a few instances the words are given so taking a len of those strings makes sense and that's what I've done. I've converted all values to int through this method.

The sld values were very interesting. First, I checked the number of unique one values. The number was quite high. Then I tried to check the frequency of each and the frequency of each having the target attack as 1. At this step, because I was trying to correlate things to the label variable, I've divided the data into train, validation and test (60, 25, 15). Looking at the number of slds for which there were Target Attack as 1, I found that the number is very less. Let's call all the slds which don't have Target Attack as 1 as whitelisted lsds. I first wanted to whitelist the entire dataset based on this frequency but it carried a few risks. For the list of whitelisted slds, there were a few which occurred only once. This didn't give me enough confidence to trust these slds everytime. So, I plotted distribution of the values and divided them into 4 parts based on the percentile frequency of each sld. [cell 22]. The ones which occur the least have the least confidence which is represented by 4 and the ones which occur the most on the list get a score of 1. I 've given a score of 5 to all slds which are not in this list a 5. The number is based on the difference in confidence. I wanted to preserve the difference between ones with the most confidence, 1 and one's with the lest confidence (the one's not on the list).

## Feature Selection:

For feature selection, I've used 2 methods:

1. A statistical method – correlation [cell 30]
   Looking at the correlation matrix, I eliminated the features which were the most similar to each other but keeping just on to represent the bunch.
2. Model based method - Random Forest Feature Selection [cell 36]
   I've used a random forest feature selection because I plan on using some tree-based methods as models. The algorithm gives us the importance of each feature present. I've eliminated the one's with the least importance.

## Model:

For the model, I've used 2 algorithms:

1. RandomForestClassifier [cell 46]
2. ExtraTreesClassifier [cell 50]

I've done hyperparameter tuning using RandomizedSearchCV and then used the hyperparameters for training the models.

## Evaluation:

I've used f1 score as a evaluation metric throughout this experiment because it is a good representative of a true Positives and true negatives. In a field like Cybersecurity, this matters a lot because the cost of both can be very high.