CP302 - Capstone Project

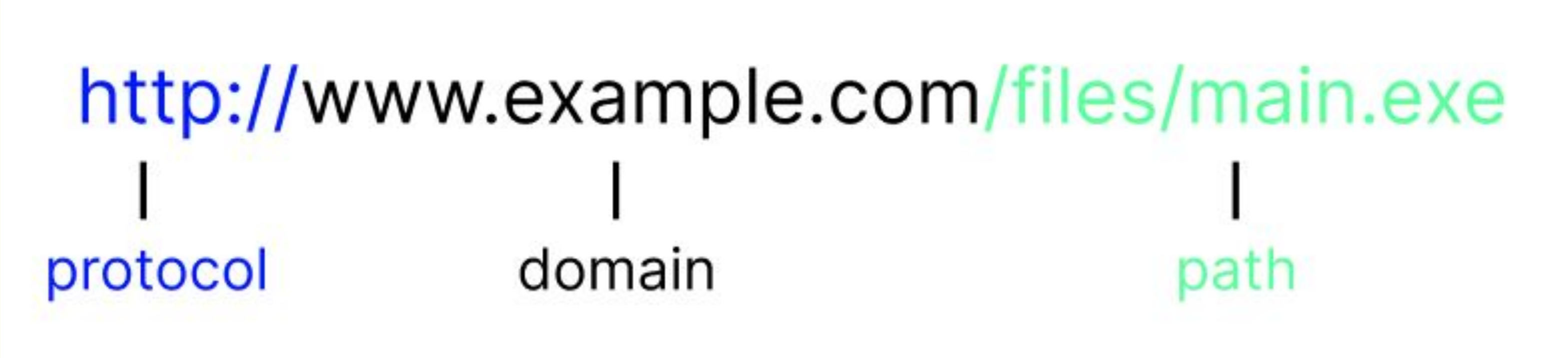
Under supervision of Dr. Sudeepta Mishra

- P Rajasekhar (2019CSB1105)
- Kshitiz arora (2019CSB1095)

Project Description

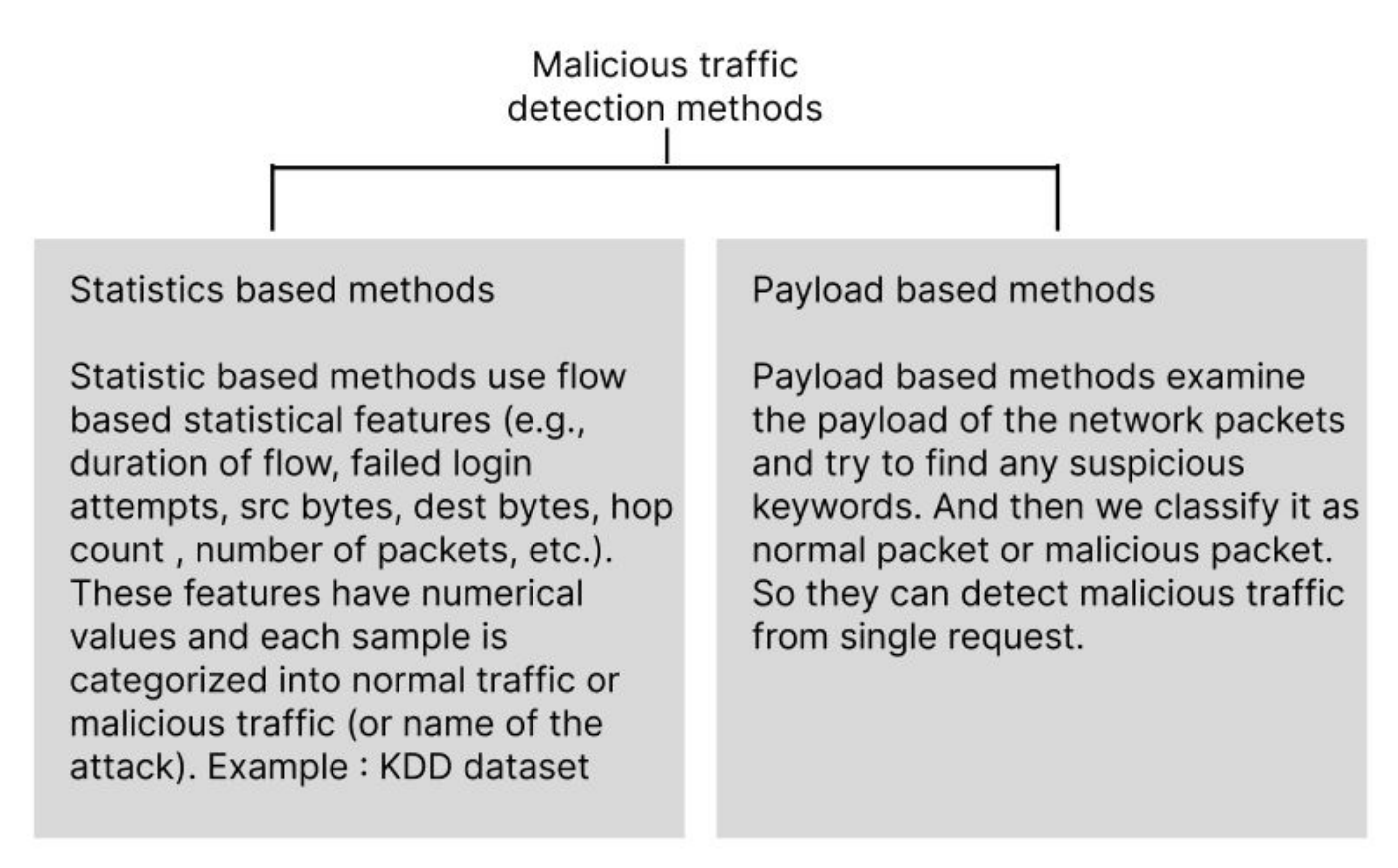
Payload based malicious network traffic detection using machine learning models BERT and RoBERTa:

- Malicious network traffic refers to any suspicious link, file, connection that is created or received over the network



- It has major impact on organisation's network security.

- We have machine learning techniques to improve network security.



Malicious traffic detection methods

Statistics based methods

Statistic based methods use flow based statistical features (e.g., duration of flow, failed login attempts, src bytes, dest bytes, hop count , number of packets, etc.). These features have numerical values and each sample is categorized into normal traffic or malicious traffic (or name of the attack). Example : KDD dataset

Payload based methods

Payload based methods examine the payload of the network packets and try to find any suspicious keywords. And then we classify it as normal packet or malicious packet. So they can detect malicious traffic from single request.

[Most of the existing research papers are on statistics based methods. There are few papers on “payload based methods”]

Limitations of statistics based approach:

- ➔ Suspicious keywords inside the payload are ignored. So it is difficult to detect malicious traffic which doesn’t exhibit significant flow patterns.
- ➔ For extracting some of the features, they need longer periods of monitoring time. So this method is not suitable for detecting malicious traffic from single http request (which may contain malware files etc.)

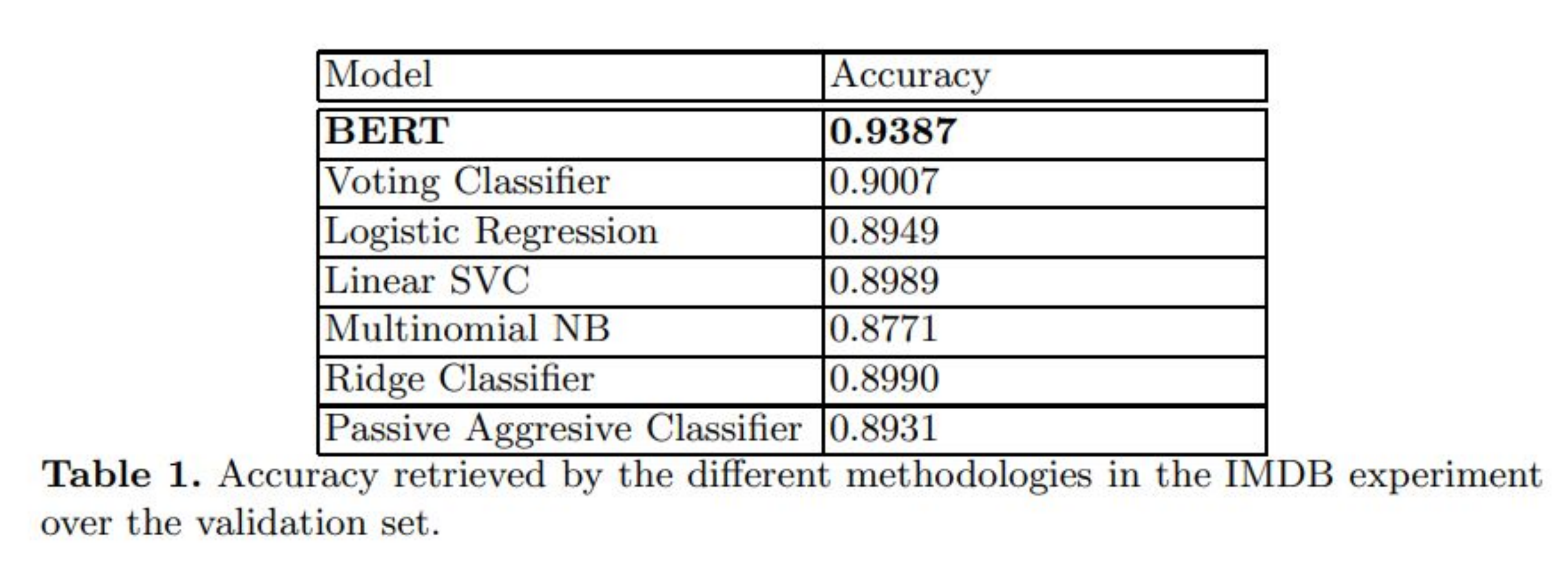
In this project we are focussing on payload based malicious traffic detection.

- We are implementing BERT (**B**idirectional **E**ncoders **R**epresentations from **T**ransformers) and RoBERTa (**R**obustly **o**ptimised **BERT** pre training **a**pproach) models in this project.
- BERT is a state of the art machine learning model published by researchers at Google (2018). It has far better performance in various NLP tasks. RoBERTa is improved version of BERT published by researchers at facebook.
- Other text classification algorithms:
  - Support Vector Machines
  - Naive Bayes Classifier
  - Random forest

How BERT and RoBERTa are better than other algorithms ?

- As opposed to directional models (which read the input sequentially), it reads the entire sequence of words at once. This allows the model to learn the context of the word from both left and right side of the word.
- Since BERT is used extensively in google search across all countries in multiple languages, this evidence is good enough to prove the efficiency of BERT
- RoBERTa is an improved version of BERT

How BERT and RoBERTa are better than other algorithms ?



Model	Accuracy
BERT	**0.9387**
Voting Classifier	0.9007
Logistic Regression	0.8949
Linear SVC	0.8989
Multinomial NB	0.8771
Ridge Classifier	0.8990
Passive Aggresive Classifier	0.8931

Table 1. Accuracy retrieved by the different methodologies in the IMDB experiment over the validation set.

Src: https://arxiv.org/pdf/2005.13012.pdf

Dataset we need

- Malicious informations of web attacks can be detected from request url.
- Payload of the packet contains this url information
- Raw payload of a packet looks like this:

```
b'GET/media/images/layout/inside/3e1b5f9a4a/?f=s&k=2189008969040714
HTTP/1.1\r\naccept-encoding: pack200-gzip, gzip\r\ncontent-type:
application/x-java-archive\r\nUser-Agent: Mozilla/4.0 (Windows 7 6.1)
Java/1.6.0_25\r\nHost: www.lapostgroup.com\r\nAccept: text/html,
image/gif, image/jpeg, *; q=.2, */*; q=.2\r\nConnection: keep-alive\r\n\r\n'
```

When we extract the URL from raw payload it looks like this:

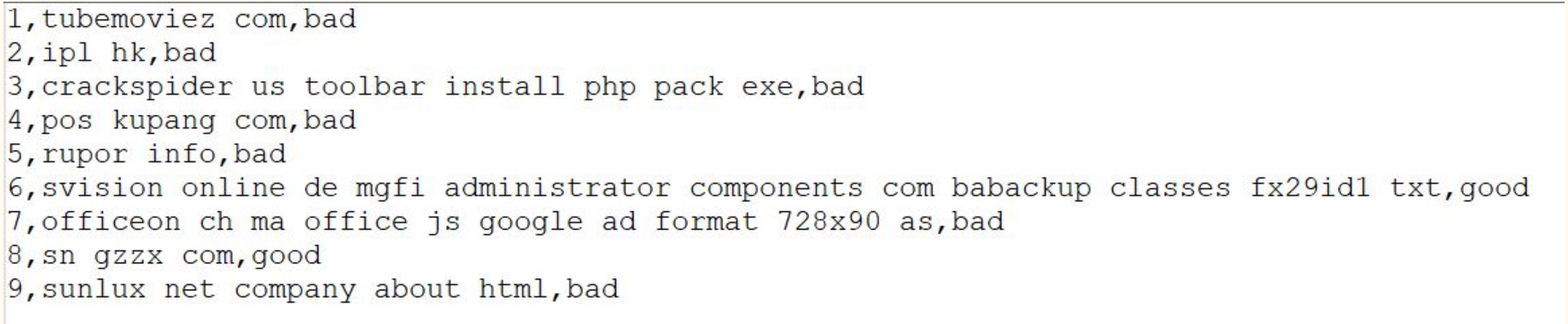
```
www.lapostgroup.com/media/images/layout/inside/3e1b5f9a4a/?f=s&k=2189008969040714
```

- We will remove the special characters from the url .

```
www lapostgroup com media images layout inside 3e1b5f9a4a f s k 2189008969040714
```

- In the dataset, we need url (payload) in the above format in one column and in other column we need whether the url is normal url or malicious url.
- And then we will train BERT and RoBERTa machine learning models on this dataset.
- BERT and RoBERTa are Text classification algorithms. Here we consider the above url as text and implement these algorithms.

- Csv file should be in this format.



```
1,tubemoviez com,bad
2,ipl hk,bad
3,crackspider us toolbar install php pack exe,bad
4,pos kupang com,bad
5,rupor info,bad
6,svision online de mgfi administrator components com babackup classes fx29id1 txt,good
7,officeon ch ma office js google ad format 728x90 as,bad
8,sn gzzx com,good
9,sunlux net company about html,bad
```

Work done so far

Task: Create dataset to train model (unsupervised learning)

Dataset requirements:

- Large dataset: at least 4000-5000 packets
- Equal number of malicious and non malicious packets

Problem with dataset collection:

- Network traffic can be captured using softwares like wireshark but the number of malicious packets in that data would be less, hence, unsuitable for training our model.

Approach

Since we could not use network traffic we searched for pcap files available online which contain malicious packets.

We found a set of such pcap files with malicious packets.

Now our task was left to labelling these packets to verify our model at the end.

Labelling packets:

> *Metadefender* is a website that tells whether a packet is malicious or not. So we decided to use it to get labels for our packets
>
> Since manually uploading urls from the packets to check their state is not feasible for a large dataset we automated the process

Automating packet labelling

Language/libraries used: Python, Selenium, Scapy, BeautifulSoup, Pandas

Method:

- Read pcap files using scapy and merge them to one file
- Run the code on merged pcap file, which extracts packet header, payload and url for each packet
- Use the extracted url to check if packet is malicious by automating the verification procedure on *Metadefender* website using selenium and beautifulsoup
- Add the returned label along with other packet information to a csv file

Problems faced with this approach

- The time taken to check one packet is approximately 10 seconds
- Even with the given pcap files, malicious packets were only ~5% of all the packets which meant that we needed to check ~40,000 packets
- Also, the website only allowed 500 requests per system each day

Hence, it was not feasible to create the required dataset with this approach.

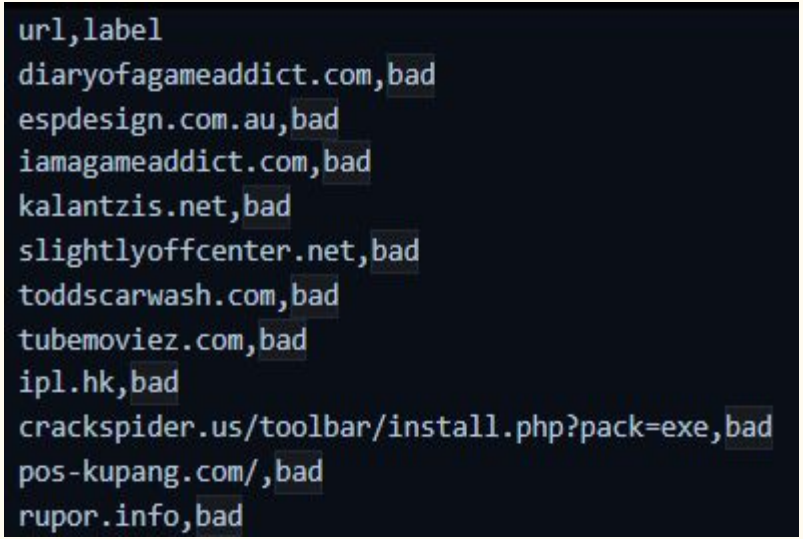
Now, we started looking at the dataset used for doing similar research.

We found the following dataset which would fulfil our requirements:

https://github.com/faizann24/Using-machine-learning-to-detect-malicious-URLs/blob/master/data/data.csv

Overview of the dataset

Total URLs	420464
Good / Non-Malicious URLs	344821
Bad / Malicious URLs	75643
Percentage of Malicious URLs	17.99%



```
url,label
diaryofagameaddict.com,bad
espdesign.com.au,bad
iamagameaddict.com,bad
kalantzis.net,bad
slightlyoffcenter.net,bad
toddscarwash.com,bad
tubemoviez.com,bad
ipl.hk,bad
crackspider.us/toolbar/install.php?pack=exe,bad
pos-kupang.com/,bad
rupor.info,bad
```

Screenshot of dataset values

Applications

In practice, the malicious HTTP payload detection model can be deployed as follows:

- A user consults a web application and sends a request to the web-server.
- Before the request arrives at the server, it will be processed by a WAF (Web Application Firewall), which takes as input all the HTTP requests and processes them using the proposed malicious traffic detection model.
- If the request is malicious then it will be rejected automatically, else it will be sent to the web-server

Thank you