Understanding Cyber Attacks on Networks

Joshua Joseph

Problem Statement

- The number of cyber attacks on system networks have dramatically increased over the past several years. As a result, there is a growing need to better understand and mitigate these threats for business and consumers alike.

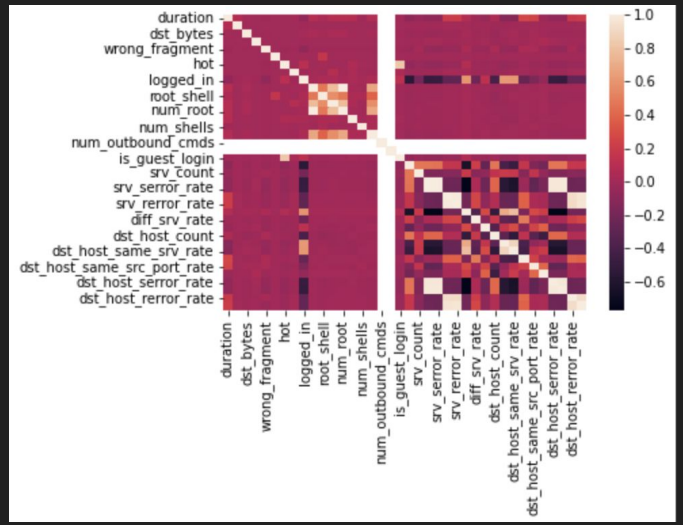
- Goal: Better understanding network threats, their characteristics, and potentially predict future threats using a ML model

Dataset

- Simulated a network with various attacks
- Date created 2018-10-09
- Derived from Kaggle
- Dataset contains
    - 25,192 rows & 42 columns
- Link: https://www.kaggle.com/sampadab17/network-intrusion-detection

Data Preprocessing/Analysis

- Removing null values
- Understanding relationship between features
- Identify potentially influential features for the “class” column



class
normal
normal
anomaly
normal
normal
anomaly
anomaly
anomaly
anomaly
anomaly
anomaly
anomaly
anomaly
normal
anomaly

```
+-------+-----+
|  class|count|
+-------+-----+
| normal|13449|
|anomaly|11743|
+-------+-----+
```

Analyzing the dataset

Schema

```
root
 |-- duration: integer (nullable = true)
 |-- protocol_type: string (nullable = true)
 |-- service: string (nullable = true)
 |-- flag: string (nullable = true)
 |-- src_bytes: integer (nullable = true)
 |-- dst_bytes: integer (nullable = true)
 |-- land: integer (nullable = true)
 |-- wrong_fragment: integer (nullable = true)
 |-- urgent: integer (nullable = true)
 |-- hot: integer (nullable = true)
 |-- num_failed_logins: integer (nullable = true)
 |-- logged_in: integer (nullable = true)
 |-- num_compromised: integer (nullable = true)
 |-- root_shell: integer (nullable = true)
 |-- su_attempted: integer (nullable = true)
 |-- num_root: integer (nullable = true)
 |-- num_file_creations: integer (nullable = true)
 |-- num_shells: integer (nullable = true)
 |-- num_access_files: integer (nullable = true)
 |-- num_outbound_cmds: integer (nullable = true)
 |-- is_host_login: integer (nullable = true)
 |-- is_guest_login: integer (nullable = true)
 |-- count: integer (nullable = true)
 |-- srv_count: integer (nullable = true)
 |-- serror_rate: double (nullable = true)
 |-- srv_serror_rate: double (nullable = true)
 |-- rerror_rate: double (nullable = true)
 |-- srv_rerror_rate: double (nullable = true)
 |-- same_srv_rate: double (nullable = true)
 |-- diff_srv_rate: double (nullable = true)
 |-- srv_diff_host_rate: double (nullable = true)
 |-- dst_host_count: integer (nullable = true)
 |-- dst_host_srv_count: integer (nullable = true)
 |-- dst_host_same_srv_rate: double (nullable = true)
 |-- dst_host_diff_srv_rate: double (nullable = true)
 |-- dst_host_same_src_port_rate: double (nullable = true)
 |-- dst_host_srv_diff_host_rate: double (nullable = true)
 |-- dst_host_serror_rate: double (nullable = true)
 |-- dst_host_srv_serror_rate: double (nullable = true)
 |-- dst_host_rerror_rate: double (nullable = true)
 |-- dst_host_srv_rerror_rate: double (nullable = true)
 |-- class: string (nullable = true)
```

General statistics of dataset

	duration	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	num_failed_logins	logged_in	num_compromised	...	dst_host_count	dst_host_srv_count
count	25192.000000	2.519200e+04	2.519200e+04	25192.000000	25192.000000	25192.00000	25192.000000	25192.000000	25192.000000	25192.000000	...	25192.000000	25192.000000
mean	305.054104	2.433063e+04	3.491847e+03	0.000079	0.023738	0.00004	0.198039	0.001191	0.394768	0.227850	...	182.532074	115.063036
std	2686.555640	2.410805e+06	8.883072e+04	0.008910	0.260221	0.00630	2.154202	0.045418	0.488811	10.417352	...	98.993895	110.646850
min	0.000000	0.000000e+00	0.000000e+00	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000
25%	0.000000	0.000000e+00	0.000000e+00	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	...	84.000000	10.000000
50%	0.000000	4.400000e+01	0.000000e+00	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	...	255.000000	61.000000
75%	0.000000	2.790000e+02	5.302500e+02	0.000000	0.000000	0.00000	0.000000	0.000000	1.000000	0.000000	...	255.000000	255.000000
max	42862.000000	3.817091e+08	5.151385e+06	1.000000	3.000000	1.00000	77.000000	4.000000	1.000000	884.000000	...	255.000000	255.000000

8 rows × 38 columns

dst_host_same_srv_rate	dst_host_diff_srv_rate	dst_host_same_src_port_rate	dst_host_srv_diff_host_rate	dst_host_serror_rate	dst_host_srv_serror_rate	dst_host_rerror_rate	dst_host_srv_rerror_rate
25192.000000	25192.000000	25192.000000	25192.000000	25192.000000	25192.000000	25192.000000	25192.000000
0.519791	0.082539	0.147453	0.031844	0.285800	0.279846	0.117800	0.118769
0.448944	0.187191	0.308367	0.110575	0.445316	0.446075	0.305869	0.317333
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.050000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.510000	0.030000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1.000000	0.070000	0.060000	0.020000	1.000000	1.000000	0.000000	0.000000
1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Random Forest Classifier

1) Turn “class” columns into numeric values
    i) (normal = 0 and anomaly = 1)
2) Assembler
3) RFC =
4) Train-test split
5) Training and testing on data
6) Finding accuracy

normal	0.0
normal	0.0
anomaly	1.0
normal	0.0
normal	0.0
anomaly	1.0
anomaly	1.0
anomaly	1.0

```
inputCols=["serror_rate", "count", "same_srv_rate", "dst_host_srv_count",
           "dst_host_srv_serror_rate", "dst_host_rerror_rate", "dst_host_srv_rerror_rate"],
outputCol="features")
```

```
(labelCol='classIndex', featuresCol= 'features')
```

```
train, test = indexed.randomSplit([0.7, 0.3]
```

```
print(acc)
```

```
0.9272097053726169
```

Future Works

1) Different model = better results?
2) Changing input columns to see if it improves score