UDP_AES256_RFC761

September 28, 2021

[ ]:
```
import pandas as pd
from numpy import arange
import matplotlib.pyplot as plt
%matplotlib inline
```

[ ]:
```
# Read from CSV

df = pd.read_csv('Outputs/packets-udp-rfc761-encrypted-dinamico-2021.09.
↪27-processed')
```

[ ]:
```
# Setting global var

bytesize = 32
proto = 'UDP'
dstport = 20001
encoding = 'AES256'
text = 'RFC761'
```

[ ]:
```
# Add a new column to the end called 'flow'

df['flow'] = df['srcip'] + ':' + df.srcport.map(str) + ' -> ' + df['dstip'] + ':
↪' + df.dstport.map(str)
# Read a specific location (R,C)
print('Example of flow {}'.format(df.iloc[5,10]))
```

```
Example of flow 127.0.0.1:59973 -> 127.0.0.1:20001
```

[ ]:
```
# Sort dataframe by an index (column) and show

df = df.sort_values(['payload_size','flow'])
print(df.iloc[:,6:11])
```

```
     payload_size   shannon      bien     tbien  \
81              1  1.000000  0.468917  0.759649
158             1  1.000000  0.468917  0.759649
159             1  1.000000  0.468917  0.759649
160             1  1.000000  0.468917  0.759649
161             1  1.000000  0.468917  0.759649
..            …         …         …         …
```

```
153          1032  0.726410  0.915664  0.933403
154          1032  0.745865  0.951134  0.965099
155          1032  0.788910  0.937308  0.951416
156          1032  0.695160  0.950402  0.967231
157          1032  0.706955  0.245024  0.930477

                                       flow
81   127.0.0.1:49792 -> 127.0.0.1:49791
158  127.0.0.1:49792 -> 127.0.0.1:49791
159  127.0.0.1:49792 -> 127.0.0.1:49791
160  127.0.0.1:49792 -> 127.0.0.1:49791
161  127.0.0.1:49792 -> 127.0.0.1:49791
..                                    …
153  127.0.0.1:59973 -> 127.0.0.1:20001
154  127.0.0.1:59973 -> 127.0.0.1:20001
155  127.0.0.1:59973 -> 127.0.0.1:20001
156  127.0.0.1:59973 -> 127.0.0.1:20001
157  127.0.0.1:59973 -> 127.0.0.1:20001

[162 rows x 5 columns]
```

[ ]:
```
# Filtering by port

is_port = df['dstport']==dstport
print(is_port.head())
df = df[is_port]
```

```
81     False
158    False
159    False
160    False
161    False
Name: dstport, dtype: bool
```

[ ]:
```
# Filtering by the number of packets of chosen size

is_bytes = df['payload_size']>2
print(is_bytes.head())
df = df[is_bytes]
```

```
94    True
0     True
1     True
2     True
3     True
Name: payload_size, dtype: bool
```

```
[ ]: # Minimize number of displayed columns

     # pd.set_option("display.max.columns", None)
     # df.head()
```

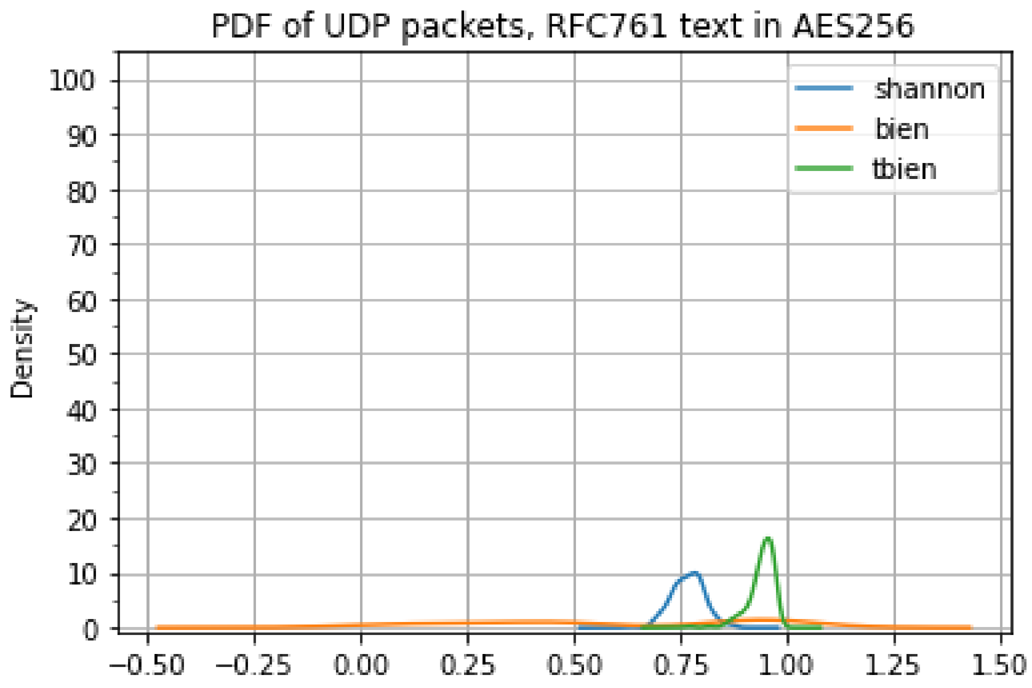
```
[ ]: # Aggregation by flow and each entropies mean

     df[['flow','shannon','bien','tbien','payload_size']].groupby('flow').mean().
      ↪sort_values('tbien', ascending=False)
```

```
[ ]:                                      shannon      bien     tbien  payload_size
     flow
     127.0.0.1:59973 -> 127.0.0.1:20001  0.768053  0.585095  0.940738   1026.859873
```

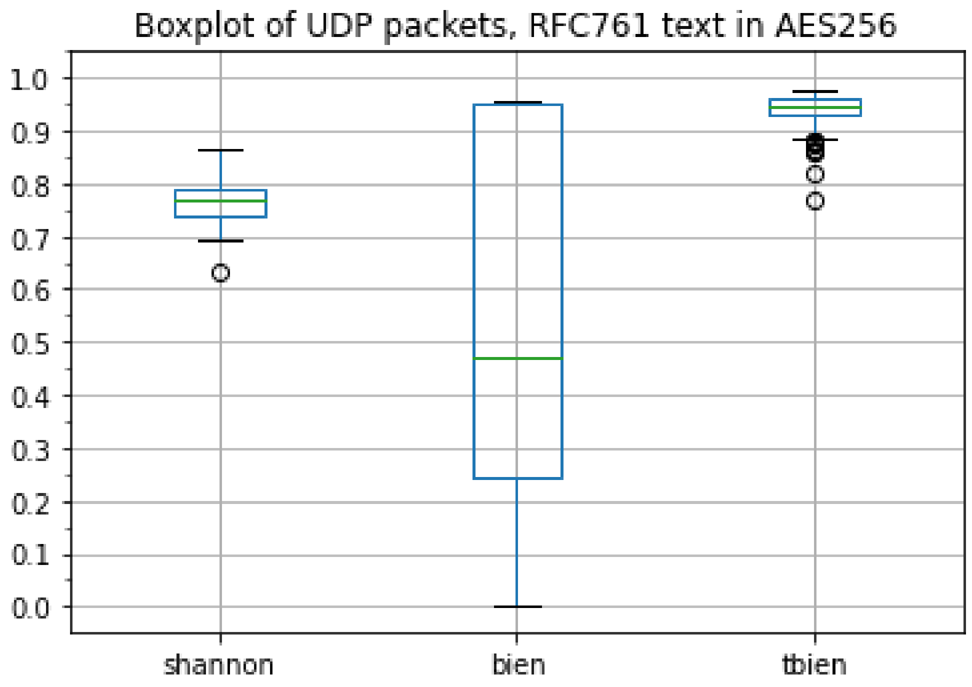
```
[ ]: # Plot 1

     title = 'PDF of {} packets, {} text in {}'.format(proto, text, encoding)
     ax = df.plot(x='payload_size',␣
      ↪y=['shannon','bien','tbien'],kind='density',title=title, grid=True)
     ax.xaxis.grid(True, which='major', linestyle='-', linewidth=1)
     ymajortick = arange(0,110,10)
     yminortick = arange(0,110,5)
     ax.set_yticks( ymajortick, minor=False )
     ax.set_yticks( yminortick, minor=True )
     ax.grid('on', which='both', axis='x' )
     plt.savefig('Plots/rfc761/{}{}{}density.png'.format(proto, encoding, text),␣
      ↪transparent=False)
```



```
[ ]: # Plot 2

     title = 'Boxplot of {} packets, {} text in {}'.format(proto, text, encoding)
     ax = df.plot(x='payload_size',␣
      ↪y=['shannon','bien','tbien'],kind='box',title=title, grid=True)
     ax.xaxis.grid(True, which='major', linestyle='-', linewidth=1)
     ymajortick = arange(0,1.1,0.1)
     yminortick = arange(0,1.1,0.05)
     ax.set_yticks( ymajortick, minor=False )
     ax.set_yticks( yminortick, minor=True )
     ax.grid('on', which='both', axis='x' )
     plt.savefig('Plots/rfc761/{}{}{}box.png'.format(proto, encoding, text),␣
      ↪transparent=False)
```



```
[ ]: # Table of data

     df = df.describe()
     print(df)
```

```
       srcport  dstport  payload_size     shannon        bien       tbien
count    157.0    157.0    157.000000  157.000000  157.000000  157.000000
mean   59973.0  20001.0   1026.859873    0.768053    0.585095    0.940738
```

```
std        0.0      0.0     64.405612    0.037786    0.342145    0.031205
min    59973.0  20001.0    225.000000    0.631099    0.000932    0.766737
25%    59973.0  20001.0   1032.000000    0.738205    0.244817    0.930630
50%    59973.0  20001.0   1032.000000    0.769455    0.471157    0.947941
75%    59973.0  20001.0   1032.000000    0.788910    0.948545    0.962728
max    59973.0  20001.0   1032.000000    0.863205    0.953414    0.974843
```

[ ]:
```
# Exporting new data

filename = 'Outputs/RFC761/{}{}{}data.csv'.format(proto, encoding, text)
df.to_csv(filename,',')
```