TCP_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-tcp-rfc761-encrypted-dinamico-2021.09.
↪27-processed')
```

[ ]:
```
# Setting global var

bytesize = 32
proto = 'TCP'
dstport = 8088
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 127.0.0.1:25542 -> 127.0.0.1:8088
```

[ ]:
```
# 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  \
314           177  0.831955  0.941973  0.954247
315           177  0.820160  0.950739  0.971726
312           217  0.423659  0.609142  0.982823
313           217  0.423659  0.609142  0.982823
1            1024  0.343135  0.549149  0.981660
..            …       …       …       …
```

```
304          1024  0.339667  0.579375  0.980456
306          1024  0.342845  0.566062  0.980161
308          1024  0.339159  0.665713  0.982216
310          1024  0.336670  0.569435  0.981033
311          1024  0.341459  0.574278  0.980143

                                              flow
314  192.168.0.119:54322 -> 239.255.255.250:1900
315   192.168.56.1:54323 -> 239.255.255.250:1900
312            127.0.0.1:25542 -> 127.0.0.1:8088
313            127.0.0.1:8088 -> 127.0.0.1:25542
1              127.0.0.1:25542 -> 127.0.0.1:8088
..                                            ...
304            127.0.0.1:8088 -> 127.0.0.1:25542
306            127.0.0.1:8088 -> 127.0.0.1:25542
308            127.0.0.1:8088 -> 127.0.0.1:25542
310            127.0.0.1:8088 -> 127.0.0.1:25542
311            127.0.0.1:8088 -> 127.0.0.1:25542

[316 rows x 5 columns]
```

[ ]:
```
# Filtering by port

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

```
314    False
315    False
312     True
313    False
1       True
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]
```

```
312    True
1      True
3      True
5      True
7      True
Name: payload_size, dtype: bool
```

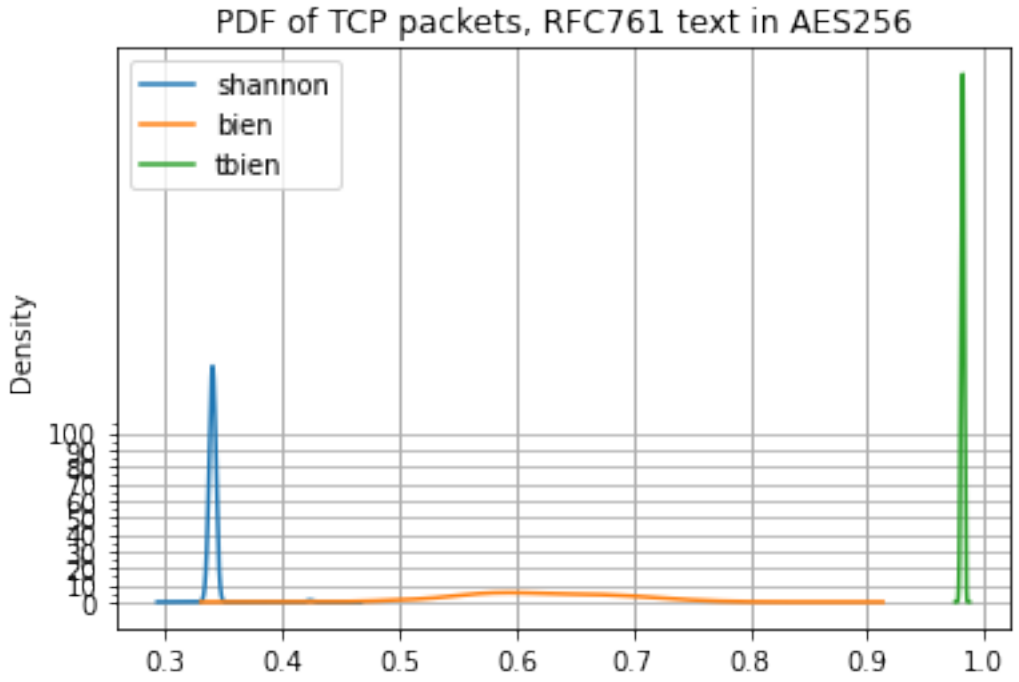
[ ]:
```
# Minimize number of displayed columns

# 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)
```

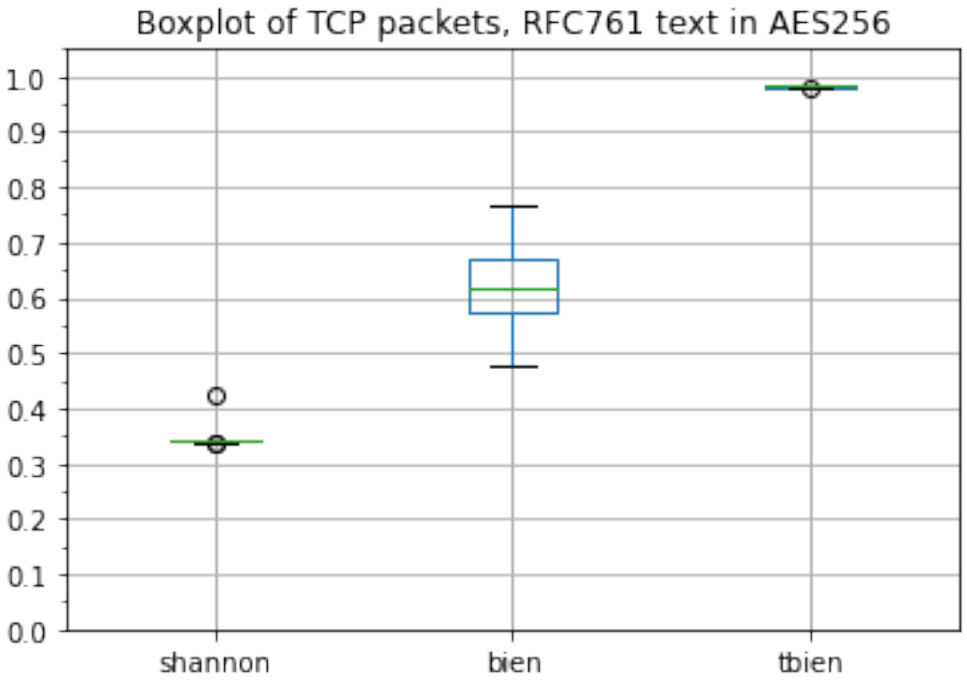
[ ]:

flow	shannon	bien	tbien	payload_size
127.0.0.1:25542 -> 127.0.0.1:8088	0.340986	0.622648	0.981691	1018.859873

[ ]:
```
# 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   25542.0   8088.0   1018.859873    0.340986    0.622648    0.981691
```

std	0.0	0.0	64.405612	0.006788	0.063926	0.001235
min	25542.0	8088.0	217.000000	0.336670	0.476254	0.978415
25%	25542.0	8088.0	1024.000000	0.339598	0.573042	0.980906
50%	25542.0	8088.0	1024.000000	0.340483	0.617221	0.981726
75%	25542.0	8088.0	1024.000000	0.341420	0.667989	0.982560
max	25542.0	8088.0	1024.000000	0.423659	0.767617	0.984395

[ ]:
```
# Exporting new data

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