In [4]:

```
#Log Read
import json
import pandas as pd

f = pd.read_json('C:\\Users\\sahil\\OneDrive\\Desktop\\Fall 2020\\Security Analytics\\nflog.json', line
s=True)
f.shape
```

Out[4]:

```
(42766, 20)
```

In [6]:

```
#ip validation
import re

pat = re.compile("^(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0
-9]?)$")
test = pat.match("10.10.10.ab")

if test:
   print ("Acceptable ip address")
else:
   print ("Unacceptable ip address")
```

```
Unacceptable ip address
```

In [28]:

```
#internal IP check
import ipaddress

ipa = ipaddress.ip_address("192.168.1.52")
ipa.is_private
```

Out[28]:

```
True
```

In [8]:

```
#ASN Check
import pyasn

asndb = pyasn.pyasn("C:\\Users\\sahil\\OneDrive\\Desktop\\Fall 2020\\Security Analytics\\ipasn_2020102
6.dat")
asndb.lookup('8.8.8.8')[0]
```

Out[8]:

```
15169
```

In [6]:

```
f.head()
```

Out[6]:

	timestamp	flow_id	in_iface	event_type	src_ip	src_port	dest_ip	dest_port	proto	app_proto	pkts_toserver	pkts_
0	2020-04-07 12:00:21	1.778733e+15	None	flow	10.5.55.1	56860	239.255.255.250	1900	UDP	failed	22.0	
1	2020-04-07 12:00:21	1.778733e+15	None	flow	10.0.1.57	56860	192.168.1.34	1900	UDP	failed	5.0	
2	2020-04-07 12:00:30	7.170903e+14	None	flow	10.0.1.99	56860	192.168.1.130	1900	UDP	failed	293.0	
3	2020-04-07 12:00:30	7.170903e+14	None	flow	172.16.0.2	62650	192.168.0.50	1971	UDP	failed	1.0	
4	2020-04-07 12:00:32	1.672783e+15	eth3	tls	172.16.0.2	57944	65.55.44.109	443	TCP	None	NaN	

In [11]:

```
#src_ip_internal
import re
import ipaddress

pat = re.compile("^(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0
-9]?)$")

for i in range(f.shape[0]):
    test = pat.match(f.iloc[i,4])
    if test:
        ipa = ipaddress.ip_address(f.iloc[i,4])
        src_ip_internal = ipa.is_private
        f['src_ip_internal'].loc[i] = src_ip_internal

f.head()
```

```
C:\Users\sahil\anaconda3\lib\site-packages\pandas\core\indexing.py:671: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexin
g.html#returning-a-view-versus-a-copy
  self._setitem_with_indexer(indexer, value)
```

Out[11]:

	timestamp	flow_id	in_iface	event_type	src_ip	src_port	dest_ip	dest_port	proto	app_proto	...	bytes_toserver
0	2020-04-07 12:00:21	1.778733e+15	None	flow	10.5.55.1	56860	239.255.255.250	1900	UDP	failed	...	10122.0
1	2020-04-07 12:00:21	1.778733e+15	None	flow	10.0.1.57	56860	192.168.1.34	1900	UDP	failed	...	87593175.0
2	2020-04-07 12:00:30	7.170903e+14	None	flow	10.0.1.99	56860	192.168.1.130	1900	UDP	failed	...	18779436.0
3	2020-04-07 12:00:30	7.170903e+14	None	flow	172.16.0.2	62650	192.168.0.50	1971	UDP	failed	...	330.0
4	2020-04-07 12:00:32	1.672783e+15	eth3	tls	172.16.0.2	57944	65.55.44.109	443	TCP	None	...	NaN

5 rows × 22 columns

In [10]:

```
#dest_ip_internal
import re
import ipaddress

pat = re.compile("^(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0
-9]?)$")

for i in range(f.shape[0]):
    test = pat.match(f.iloc[i,6])
    if test:
        ipa = ipaddress.ip_address(f.iloc[i,6])
        dest_ip_internal = ipa.is_private
        f['dest_ip_internal'].loc[i] = dest_ip_internal

f.head()
```

```
C:\Users\sahil\anaconda3\lib\site-packages\pandas\core\indexing.py:671: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexin
g.html#returning-a-view-versus-a-copy
  self._setitem_with_indexer(indexer, value)
```

Out[10]:

	timestamp	flow_id	in_iface	event_type	src_ip	src_port	dest_ip	dest_port	proto	app_proto	...	bytes_toserver
0	2020-04-07 12:00:21	1.778733e+15	None	flow	10.5.55.1	56860	239.255.255.250	1900	UDP	failed	...	10122.0
1	2020-04-07 12:00:21	1.778733e+15	None	flow	10.0.1.57	56860	192.168.1.34	1900	UDP	failed	...	87593175.0
2	2020-04-07 12:00:30	7.170903e+14	None	flow	10.0.1.99	56860	192.168.1.130	1900	UDP	failed	...	18779436.0
3	2020-04-07 12:00:30	7.170903e+14	None	flow	172.16.0.2	62650	192.168.0.50	1971	UDP	failed	...	330.0
4	2020-04-07 12:00:32	1.672783e+15	eth3	tls	172.16.0.2	57944	65.55.44.109	443	TCP	None	...	NaN

5 rows × 22 columns

In [24]:

```
f.shape
```

Out[24]:

```
(42766, 22)
```

In [15]:

```
#dest_ip_company
import pyasn

asndb = pyasn.pyasn("C:\\Users\\sahil\\OneDrive\\Desktop\\Fall 2020\\Security Analytics\\ipasn_2020102
6.dat")
#asndb.lookup('8.8.8.8')[0]

microsoftASN = [13811,6182,8075,23468,20046,8069,8072,26222,8068,3598]
googleASN = [36492,36040,22577,45566,41264,15169,36384]
amazonASN = [16509,38895,39111,14618]
facebookASN = [32934]

#ASN = [13811,6182,8075,23468,20046,8069,8072,26222,8068,3598,36492,36040,22577,45566,41264,15169,3638
4,16509,38895,39111,14618,32934]

for i in range(f.shape[0]):
    testASN = asndb.lookup(f.iloc[i,6])[0]
    if testASN in microsoftASN:
        f['dest_ip_company'].loc[i] = "Microsoft"
    elif testASN in googleASN:
        f['dest_ip_company'].loc[i] = "Google"
    elif testASN in amazonASN:
        f['dest_ip_company'].loc[i] = "Amazon"
    elif testASN in facebookASN:
        f['dest_ip_company'].loc[i] = "Facebook"
    else:
        f['dest_ip_company'].loc[i] = "Other"

f.head()
```

```
C:\Users\sahil\anaconda3\lib\site-packages\pandas\core\indexing.py:671: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexin
g.html#returning-a-view-versus-a-copy
  self._setitem_with_indexer(indexer, value)
```

Out[15]:

	timestamp	flow_id	in_iface	event_type	src_ip	src_port	dest_ip	dest_port	proto	app_proto	...	bytes_toclient	
0	2020-04-07 12:00:21	1.778733e+15	None	flow	10.5.55.1	56860	239.255.255.250	1900	UDP	failed	...	0.0	0
1	2020-04-07 12:00:21	1.778733e+15	None	flow	10.0.1.57	56860	192.168.1.34	1900	UDP	failed	...	87661354.0	0
2	2020-04-07 12:00:30	7.170903e+14	None	flow	10.0.1.99	56860	192.168.1.130	1900	UDP	failed	...	1447.0	0
3	2020-04-07 12:00:30	7.170903e+14	None	flow	172.16.0.2	62650	192.168.0.50	1971	UDP	failed	...	0.0	0
4	2020-04-07 12:00:32	1.672783e+15	eth3	tls	172.16.0.2	57944	65.55.44.109	443	TCP	None	...	NaN	

5 rows × 23 columns

In [18]:

```
import json

flowj = f.to_json(orient='records', lines=True)
# save it
with open('C:\\Users\\sahil\\OneDrive\\Desktop\\netflow.json','w') as file:
    file.write(flowj)
```

In [ ]: