In [60]:
```
!pip install gmplot
import pandas as pd
import ipaddress
import matplotlib.pyplot as plt
import sys
import pycountry
import gmplot
```

```
Requirement already satisfied: gmplot in c:\users\lucas\anaconda3\lib\site-pa
ckages (1.4.1)
Requirement already satisfied: requests in c:\users\lucas\anaconda3\lib\site-
packages (from gmplot) (2.28.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\lucas\anaconda3
\lib\site-packages (from requests->gmplot) (2022.9.14)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\lucas\ana
conda3\lib\site-packages (from requests->gmplot) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\lucas\anacon
da3\lib\site-packages (from requests->gmplot) (1.26.11)
Requirement already satisfied: idna<4,>=2.5 in c:\users\lucas\anaconda3\lib\s
ite-packages (from requests->gmplot) (3.3)
```

In [61]:
```
data = pd.read_csv("AlienVault_IP_reputation.csv")
data.head(10)
```

Out[61]:

	Unnamed: 0	IP	Reliability	Risk	Type	Country	Locale	
0	0	222.76.212.189	4	2	Scanning Host	cn	Xiamen	24.4797992706,118.(
1	1	222.76.212.185	4	2	Scanning Host	cn	Xiamen	24.4797992706,118.(
2	2	222.76.212.186	4	2	Scanning Host	cn	Xiamen	24.4797992706,118.(
3	3	5.34.246.67	6	3	Spamming	us	NaN	;
4	4	178.94.97.176	4	5	Scanning Host	ua	Merefa	49.8230018616,36.05
5	5	66.2.49.232	4	2	Scanning Host	us	Union City	37.5962982178,-122.0(
6	6	222.76.212.173	4	2	Scanning Host	cn	Xiamen	24.4797992706,118.(
7	7	222.76.212.172	4	2	Scanning Host	cn	Xiamen	24.4797992706,118.(
8	8	222.76.212.171	4	2	Scanning Host	cn	Xiamen	24.4797992706,118.(
9	9	174.142.46.19	6	3	Spamming	NaN	NaN	24.4797992706,118.(

In [62]:
```
#task 1
print(data.describe())
for col in data.select_dtypes(include='object'):
    print(f"Column: {col}")
    print(data[col].value_counts())
    print(data[col].unique())
```

```
        Unnamed: 0   Reliability          Risk
count  10000.00000  10000.000000  10000.000000
mean    4999.50000      4.004800      2.545900
std     2886.89568      0.920033      0.776372
min        0.00000      1.000000      1.000000
25%     2499.75000      4.000000      2.000000
50%     4999.50000      4.000000      2.000000
75%     7499.25000      4.000000      3.000000
max     9999.00000     10.000000      6.000000
Column: IP
222.76.212.189     1
195.226.218.127    1
195.226.218.113    1
195.226.218.133    1
195.226.218.132    1
                  ..
58.59.162.52       1
58.59.162.112      1
58.59.162.107      1
```

In [63]:
```
#task 2
data['subnet'] = data['IP'].apply(lambda x: '.'.join(x.split('.')[:3]) + '.0')
unique_subnets = data['subnet'].nunique()
print(f"There are {unique_subnets} unique 24-bit subnet addresses in the datas
subnet_count = data[data['subnet'] == '222.76.212.0'].shape[0]
print(f"There are {subnet_count} IP addresses in the CIDR block 222.76.212.0/2
subnet_count = data[data['subnet'] == '5.34.246.0'].shape[0]
print(f"There are {subnet_count} IP addresses in the CIDR block 5.34.246.0/24.
```

```
There are 1037 unique 24-bit subnet addresses in the dataset.
There are 22 IP addresses in the CIDR block 222.76.212.0/24.
There are 3 IP addresses in the CIDR block 5.34.246.0/24.
```

In [64]:
```
#task 3
data['ip_address'] = data['IP'].apply(ipaddress.IPv4Address)
cidr_block = ipaddress.IPv4Network('222.76.212.0/24')
data.loc[data['ip_address'].apply(lambda x: x in cidr_block), 'Risk'] = data['
avg_risk_score = data['Risk'].mean()

print(f"The average risk score for the IP addresses in CIDR block 222.76.212.0
```

```
The average risk score for the IP addresses in CIDR block 222.76.212.0/24 is
2.55.
```

In [65]:
```
#Task 4
us_df = data[data['Country'] == 'us']
cn_df = data[data['Country'] == 'cn']
us = us_df['Risk'].mean()
cn = cn_df['Risk'].mean()
risk_score_gap = us_avg_risk_score - cn_avg_risk_score

print(f"The average risk score for 'US' geo-located IP addresses is {us:.2f}."
print(f"The average risk score for 'CN' geo-located IP addresses is {cn:.2f}."
print(f"The gap in risk scores between 'US' and 'CN' geo-located IP addresses
```

```
The average risk score for 'US' geo-located IP addresses is 2.33.
The average risk score for 'CN' geo-located IP addresses is 2.43.
The gap in risk scores between 'US' and 'CN' geo-located IP addresses is -0.1
0.
```

In [66]:
```
#task 5
countr_name_list=[]
country_name_list=[]
for cc in data['Country']:
    cc=str(cc).lower()
    if not (cc=='nan'):
        country_name = pycountry.countries.lookup(cc).name
        country_name_list.append(country_name)
    else:
        country_name_list.append('nan')

data['country_name']= country_name_list
```

In [67]:
```
countries = ['Ukraine', 'Russian Federation', 'United Kingdom', 'Mexico', 'Pak
df = data[data['country_name'].isin(countries)]
for country in countries:
    count = len(df[df['country_name'] == country])
    print(f"There are {count} IP addresses in the dataset associated with {cou
```

```
There are 322 IP addresses in the dataset associated with Ukraine.
There are 210 IP addresses in the dataset associated with Russian Federation.
There are 266 IP addresses in the dataset associated with United Kingdom.
There are 145 IP addresses in the dataset associated with Mexico.
There are 165 IP addresses in the dataset associated with Pakistan.
```

In [68]:
```
#task 6
countries = ['United States', 'Russian Federation']
filtered_df = data[data['country_name'].isin(countries)]
us_rel = filtered_df[filtered_df['country_name'] == 'United States']['Reliabil
ru_rel = filtered_df[filtered_df['country_name'] == 'Russian Federation']['Rel
rel_gap = abs(us_rel - ru_rel);

print(f"The average reliability score for the United States is {us_rel:.2f}.")
print(f"The average reliability score for the Russian Federation is {ru_rel:.2
print(f"The reliability gap between United States and Russian Federation is {r
```

```
The average reliability score for the United States is 4.31.
The average reliability score for the Russian Federation is 4.28.
The reliability gap between United States and Russian Federation is 0.03.
```

In [69]:
```
#Task 7
lat_list = []
long_list= []

for coord in data['Coords']:
    lat_long = coord.split(',')
    lat= lat_long[0]
    lon = lat_long[1]
    lat_list.append(float(lat))
    long_list.append(float(lon))

data['lat'] = lat_list
data['lon'] = long_list
```

In [70]:
```
australia_df = data[data['Country'] == 'Australia']
gmap = gmplot.GoogleMapPlotter(-25.2744, 133.7751, 4)
for i, row in australia_df.iterrows():
    gmap.marker(row['lat'], row['lon'], title=row['IP Address'])

gmap.draw('australia_map.html')
```

In [71]:
```
#Task 8
def attack_types(country):
    country_data = data[data['country_name'] == country]
    attack_types = country_data['Type'].value_counts().head(3)
    return attack_types

us = attack_types('United States')
ger = attack_types('Germany')
china = attack_types('China')

print('Top 3 attack types in the United States:')
print(us)
print('Top 3 attack types in Germany:')
print(ger)
print('Top 3 attack types in China:')
print(china)
```

```
Top 3 attack types in the United States:
Scanning Host    1758
Spamming          267
Malware IP         73
Name: Type, dtype: int64
Top 3 attack types in Germany:
Scanning Host     381
Malware Domain     20
Malware IP         14
Name: Type, dtype: int64
Top 3 attack types in China:
Scanning Host     2053
Malicious Host      37
Malware Domain      26
Name: Type, dtype: int64
```

In [72]:
```
#Task 9
cr = data.groupby('country_name')['Risk'].mean()

cr_sorted= country_risk.sort_values(ascending=False)

top_five = cr_sorted.head(5)

print("Top 5 countries with the highest average risk scores:")
print(top_five)
```

```
Top 5 countries with the highest average risk scores:
country_name
Finland                   9.0
Hungary                   6.5
Cyprus                    6.0
Sri Lanka                 6.0
Virgin Islands, British   6.0
Name: Reliability, dtype: float64
```

In [73]:
```
#task 10
crel = data.groupby('country_name')['Reliability'].mean()

crel_sorted = country_rel.sort_values(ascending=False)

top_five = crel_sorted.head(5)

print("Top 5 countries with highest average Reliability scores:")
print(top_five)
```

```
Top 5 countries with highest average Reliability scores:
country_name
Finland                   9.0
Hungary                   6.5
Cyprus                    6.0
Sri Lanka                 6.0
Virgin Islands, British   6.0
Name: Reliability, dtype: float64
```

In [74]:
```
#extra question
def get_subnet(ip):
    ip_list = ip.split('.')
    subnet_list = ip_list[:3]
    subnet_list.append('0')
    subnet = '.'.join(subnet_list)
    return subnet

data['subnet'] = data['IP'].apply(get_subnet)
unique_subnets = data['subnet'].nunique()
print("There are", unique_subnets, "unique 24 bit subnet addresses in the data
subnet_risk = data.groupby('subnet')['Risk'].mean().reset_index()
subnet_risk.columns = ['subnet', 'subnet_risk']
print(subnet_risk.head())
```

```
There are 1037 unique 24 bit subnet addresses in the dataset.
       subnet  subnet_risk
0   1.0.232.0          4.0
1  1.26.119.0          3.0
2  1.36.226.0          2.0
3    1.93.4.0          2.0
4   1.93.45.0          2.0
```