In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# Step 2: Load Dataset
```

In [4]:
```
df = pd.read_csv("cybersecurity_data.csv")

df.info()
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 282 entries, 0 to 281
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   bytes_in             282 non-null    int64 
 1   bytes_out            282 non-null    int64 
 2   creation_time        282 non-null    object
 3   end_time             282 non-null    object
 4   src_ip               282 non-null    object
 5   src_ip_country_code  282 non-null    object
 6   protocol             282 non-null    object
 7   response.code        282 non-null    int64 
 8   dst_port             282 non-null    int64 
 9   dst_ip               282 non-null    object
 10  rule_names           282 non-null    object
 11  observation_name     282 non-null    object
 12  source.meta          282 non-null    object
 13  source.name          282 non-null    object
 14  time                 282 non-null    object
 15  detection_types      282 non-null    object
dtypes: int64(4), object(12)
memory usage: 35.4+ KB
```

Out[4]:

	bytes_in	bytes_out	creation_time	end_time	src_ip	src_ip_country_code
0	5602	12990	2024-04-25T23:00:00Z	2024-04-25T23:10:00Z	147.161.161.82	AE
1	30912	18186	2024-04-25T23:00:00Z	2024-04-25T23:10:00Z	165.225.33.6	US
2	28506	13468	2024-04-25T23:00:00Z	2024-04-25T23:10:00Z	165.225.212.255	CA
3	30546	14278	2024-04-25T23:00:00Z	2024-04-25T23:10:00Z	136.226.64.114	US
4	6526	13892	2024-04-25T23:00:00Z	2024-04-25T23:10:00Z	165.225.240.79	NL

In [5]:
```
# Remove Duplicate Rows
```

In [6]:
```
df_unique = df.drop_duplicates()
```

In [7]:
```
df_unique['bytes_in'] = df_unique['bytes_in'].fillna(df_unique['bytes_in'].media
df_unique['bytes_out'] = df_unique['bytes_out'].fillna(df_unique['bytes_out'].me
```

In [8]:
```
df_unique.dropna(subset=['src_ip', 'dst_ip'], inplace=True)
```

In [10]:
```
df_unique['creation_time'] = pd.to_datetime(df_unique['creation_time'])
df_unique['end_time'] = pd.to_datetime(df_unique['end_time'])
df_unique['time'] = pd.to_datetime(df_unique['time'])
```

In [11]:
```
df_unique['src_ip_country_code'] = df_unique['src_ip_country_code'].str.upper()
```

In [12]:
```
df_unique.info()
df_unique.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 282 entries, 0 to 281
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   bytes_in             282 non-null    int64
 1   bytes_out            282 non-null    int64
 2   creation_time        282 non-null    datetime64[ns, UTC]
 3   end_time             282 non-null    datetime64[ns, UTC]
 4   src_ip               282 non-null    object
 5   src_ip_country_code  282 non-null    object
 6   protocol             282 non-null    object
 7   response.code        282 non-null    int64
 8   dst_port             282 non-null    int64
 9   dst_ip               282 non-null    object
 10  rule_names           282 non-null    object
 11  observation_name     282 non-null    object
 12  source.meta          282 non-null    object
 13  source.name          282 non-null    object
 14  time                 282 non-null    datetime64[ns, UTC]
 15  detection_types      282 non-null    object
dtypes: datetime64[ns, UTC](3), int64(4), object(9)
memory usage: 35.4+ KB
```

Out[12]:

	bytes_in	bytes_out	creation_time	end_time	src_ip	src_ip_country_co
0	5602	12990	2024-04-25 23:00:00+00:00	2024-04-25 23:10:00+00:00	147.161.161.82	
1	30912	18186	2024-04-25 23:00:00+00:00	2024-04-25 23:10:00+00:00	165.225.33.6	
2	28506	13468	2024-04-25 23:00:00+00:00	2024-04-25 23:10:00+00:00	165.225.212.255	
3	30546	14278	2024-04-25 23:00:00+00:00	2024-04-25 23:10:00+00:00	136.226.64.114	
4	6526	13892	2024-04-25 23:00:00+00:00	2024-04-25 23:10:00+00:00	165.225.240.79	

In [13]:
```
# visualize the distribution of bytes in and bytes out
```

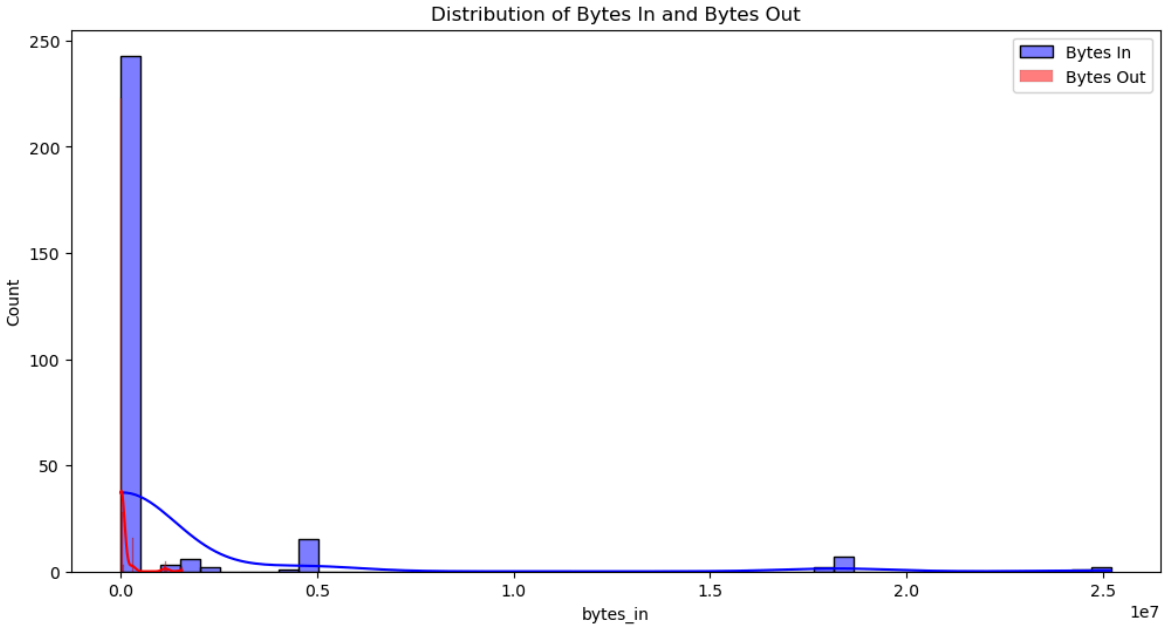
In [14]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [15]:
```
Distribution plot for Bytes In and Bytes Out
```

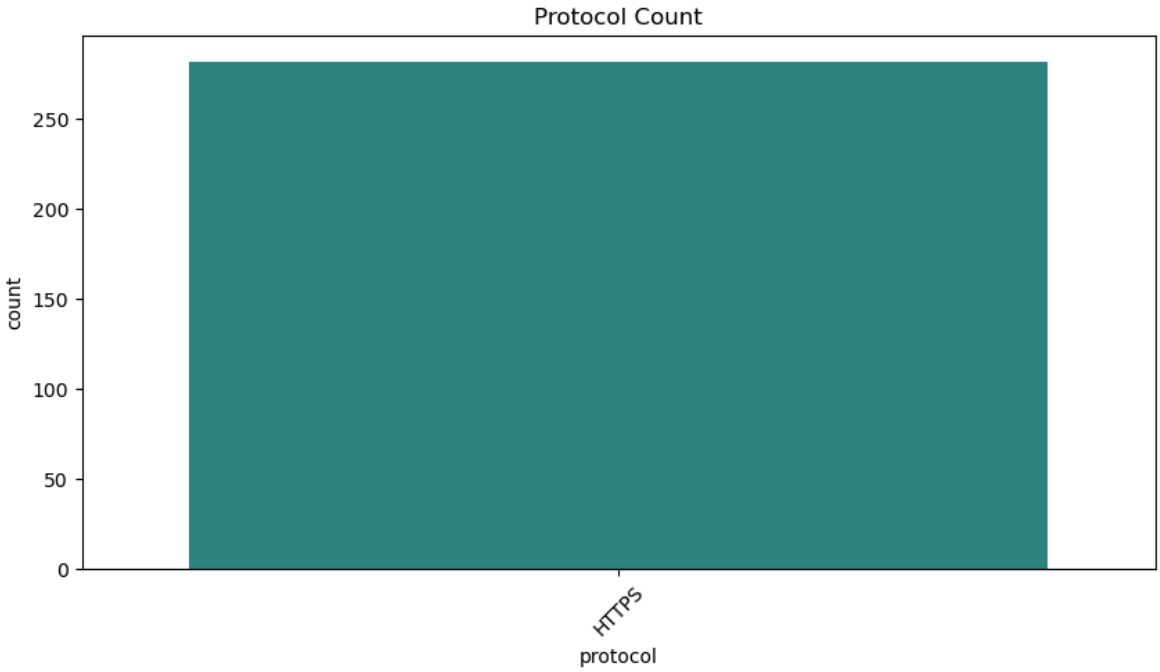
```
  Cell In[15], line 1
    Distribution plot for Bytes In and Bytes Out
                  ^
SyntaxError: invalid syntax
```

In [16]:
```
# Distribution plot for Bytes In and Bytes Out
```

In [17]:
```
plt.figure(figsize=(12, 6))
sns.histplot(df_unique['bytes_in'], bins=50, color='blue', kde=True, label='Byte
sns.histplot(df_unique['bytes_out'], bins=50, color='red', kde=True, label='Byte
plt.legend()
plt.title('Distribution of Bytes In and Bytes Out')
plt.show()
```

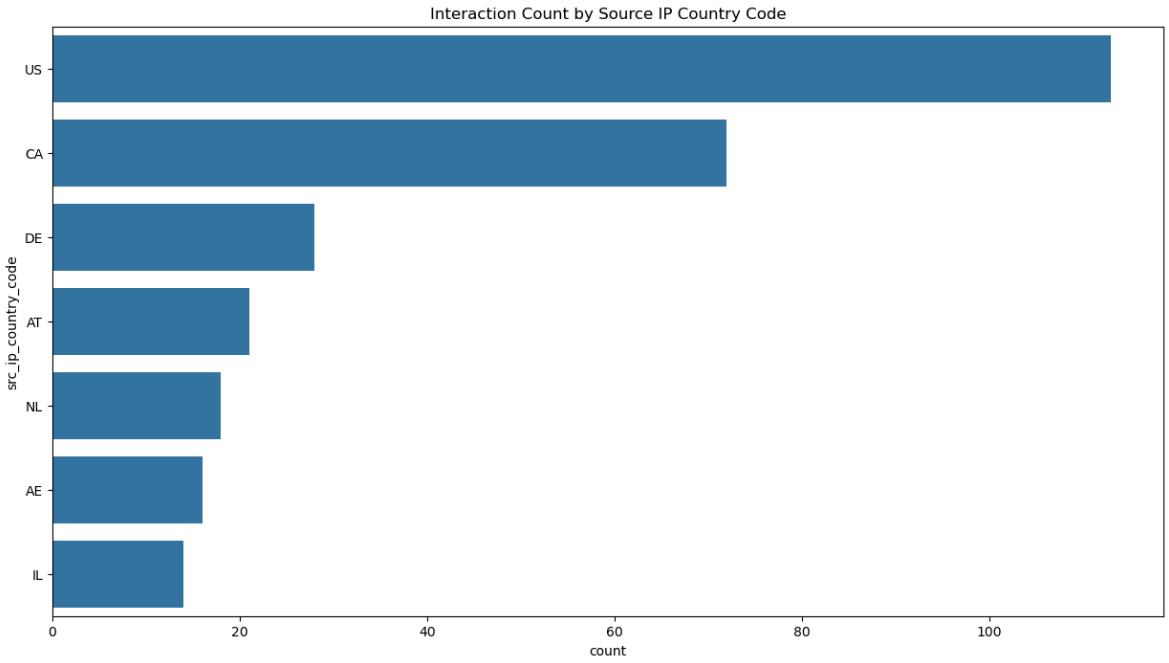


In [18]:
```
plt.figure(figsize=(10, 5))
sns.countplot(x='protocol', hue='protocol', data=df_unique, palette='viridis', l
plt.title('Protocol Count')
plt.xticks(rotation=45)
plt.show()
```



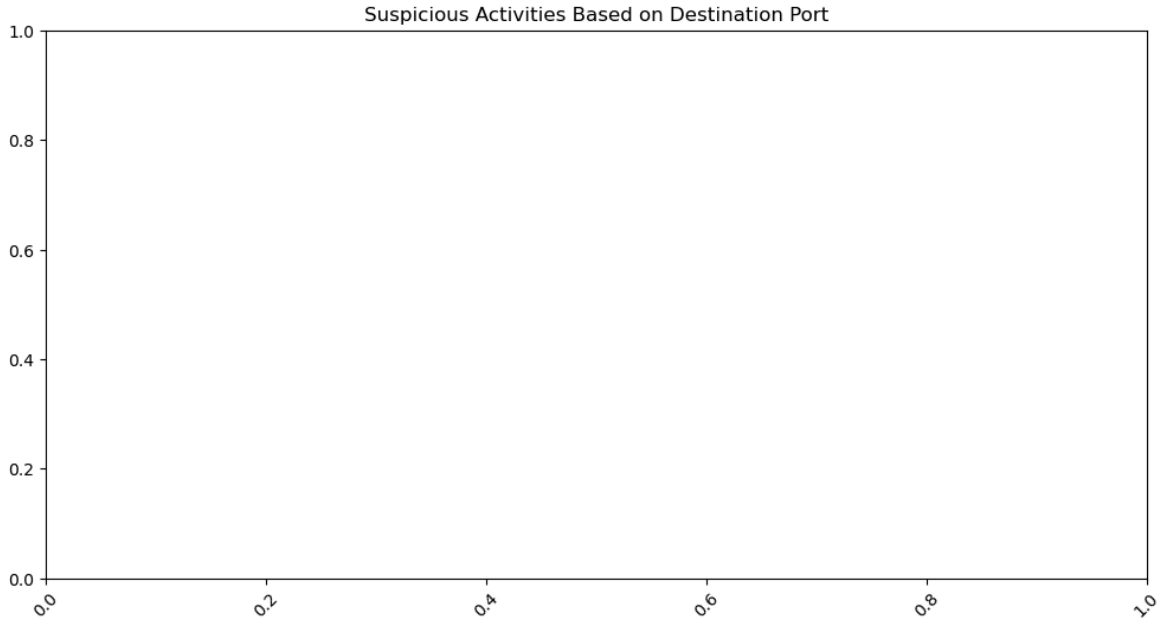
In [19]:
```
# Country-Based Interaction Count
```

In [20]:
```
plt.figure(figsize=(15, 8))
sns.countplot(y='src_ip_country_code', data=df_unique,
              order=df_unique['src_ip_country_code'].value_counts().index)
plt.title('Interaction Count by Source IP Country Code')
plt.show()
```



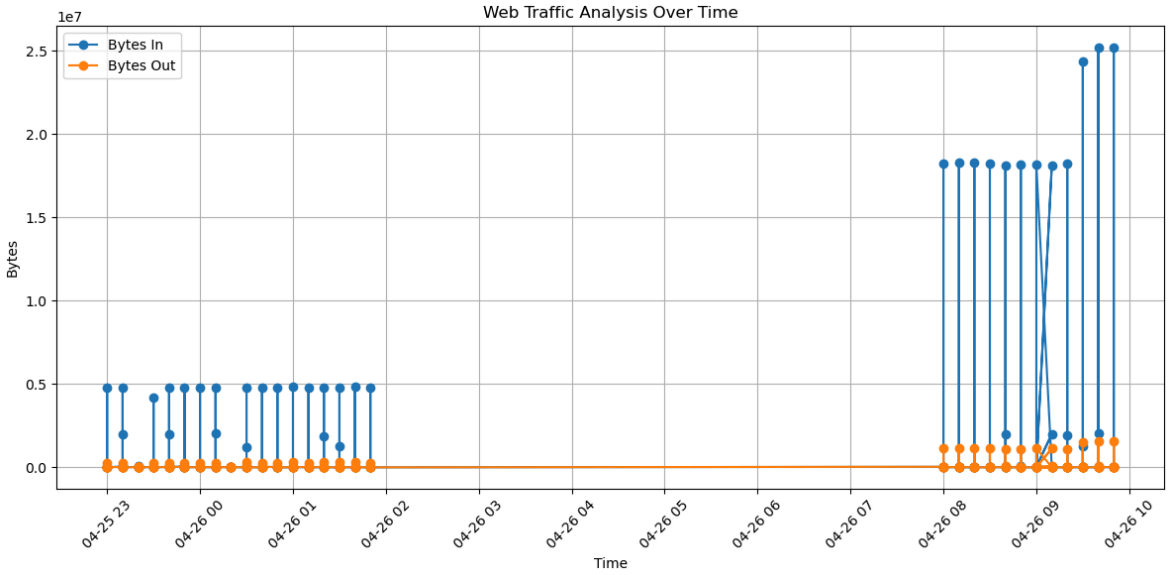
In [21]:
```
# Suspicious Activities by Destination Port
```

In [22]:
```
plt.figure(figsize=(12, 6))
sns.countplot(x='dst_port',
              data=df_unique[df_unique['detection_types'] == 'Suspicious'],
              palette='coolwarm')
plt.title('Suspicious Activities Based on Destination Port')
plt.xticks(rotation=45)
plt.show()
```

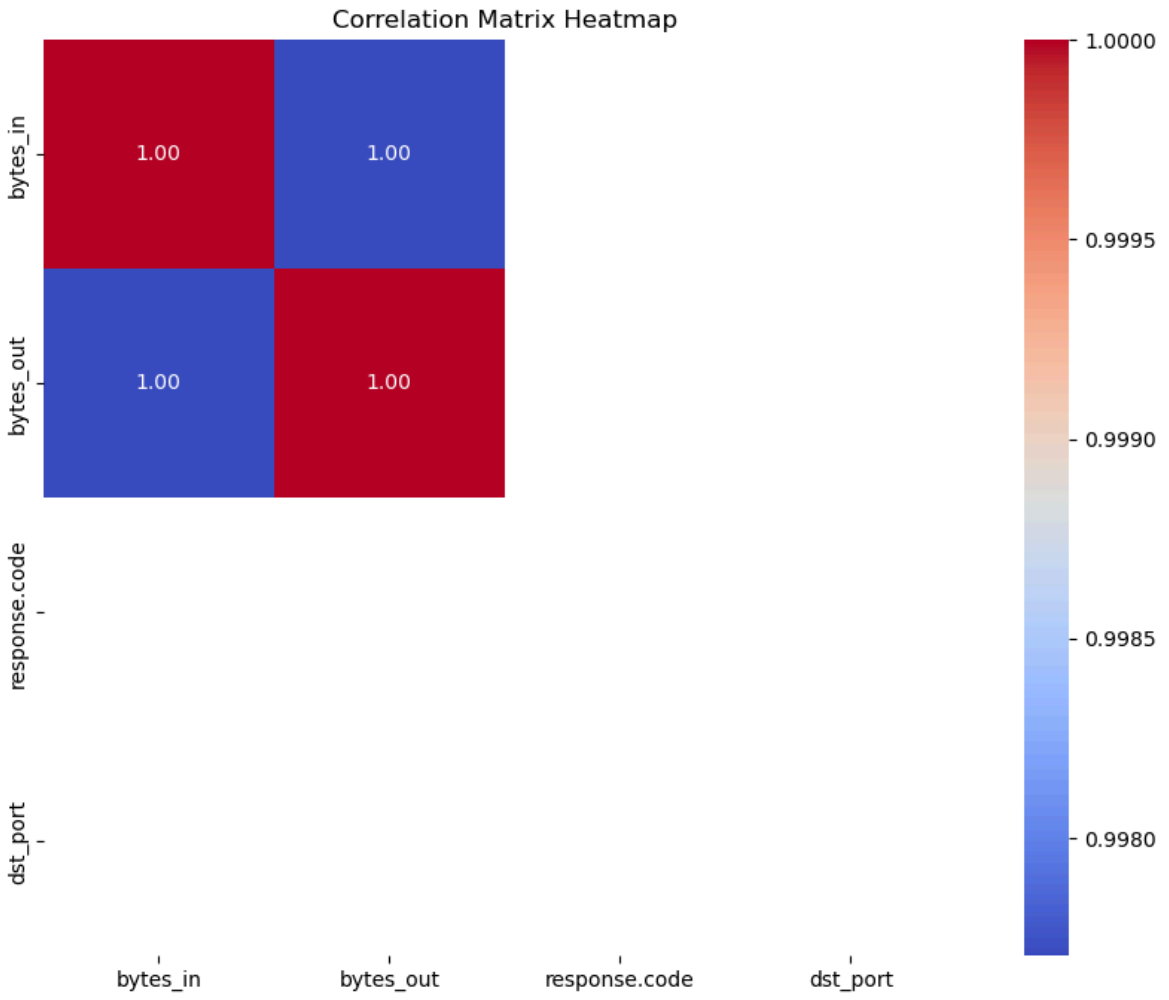


In [23]:
```
df_unique['creation_time'] = pd.to_datetime(df_unique['creation_time'])

plt.figure(figsize=(12, 6))
plt.plot(df_unique['creation_time'], df_unique['bytes_in'], label='Bytes In', ma
plt.plot(df_unique['creation_time'], df_unique['bytes_out'], label='Bytes Out',
plt.title('Web Traffic Analysis Over Time')
plt.xlabel('Time')
plt.ylabel('Bytes')
plt.legend()
```

```
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



In [24]:
```
numeric_df = df_unique.select_dtypes(include=['float64', 'int64'])
correlation_matrix_numeric = numeric_df.corr()

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix_numeric, annot=True, fmt=".2f", cmap='coolwarm')
plt.title('Correlation Matrix Heatmap')
plt.show()
```



In [25]:
```
from sklearn.preprocessing import StandardScaler, OneHotEncoder
```

In [26]:
```
df_unique['session_duration'] = (df_unique['end_time'] - df_unique['creation_tim
```

In [27]:
```
df_unique['avg_packet_size'] = (df_unique['bytes_in'] + df_unique['bytes_out'])
```

In [28]:
```
scaler = StandardScaler()
scaled_features = scaler.fit_transform(df_unique[['bytes_in', 'bytes_out', 'sess
scaled_columns = ['scaled_bytes_in', 'scaled_bytes_out', 'scaled_session_duratio
```

In [29]:
```
from sklearn.preprocessing import OneHotEncoder

encoder = OneHotEncoder(sparse_output=False)
encoded_features = encoder.fit_transform(df_unique[['src_ip_country_code']])
encoded_columns = encoder.get_feature_names_out(['src_ip_country_code'])
encoded_df = pd.DataFrame(encoded_features, columns=encoded_columns, index=df_un
```

In [30]:
```
df_transformed = pd.concat([df_unique, scaled_df, encoded_df], axis=1)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[30], line 1
----> 1 df_transformed = pd.concat([df_unique, scaled_df, encoded_df], axis=1)

NameError: name 'scaled_df' is not defined
```

In [32]:
```
scaled_df = pd.DataFrame(scaled_features, columns=scaled_columns, index=df_uniqu
```

```
encoder = OneHotEncoder(sparse_output=False)
encoded_features = encoder.fit_transform(df_unique[['src_ip_country_code']])
encoded_columns = encoder.get_feature_names_out(['src_ip_country_code'])
encoded_df = pd.DataFrame(encoded_features, columns=encoded_columns, index=df_un
```

In [33]:
```
df_transformed = pd.concat([df_unique, scaled_df, encoded_df], axis=1)
```

In [34]:
```
df_transformed.head()
```

Out[34]:

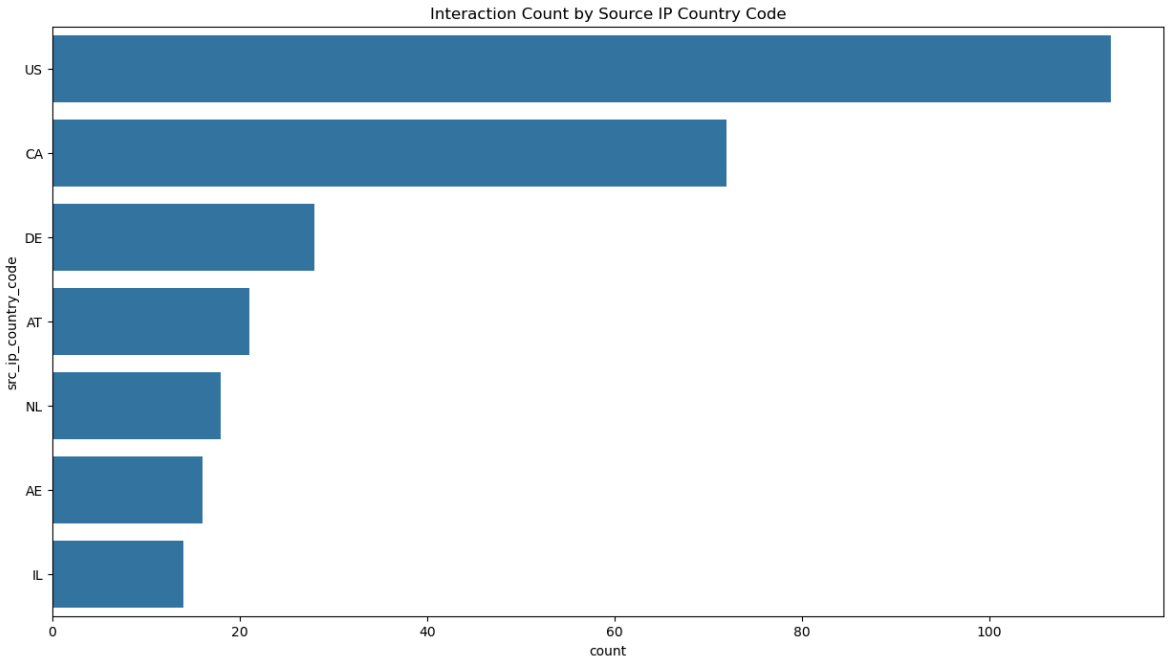
	bytes_in	bytes_out	creation_time	end_time	src_ip	src_ip_country_co
0	5602	12990	2024-04-25 23:00:00+00:00	2024-04-25 23:10:00+00:00	147.161.161.82	
1	30912	18186	2024-04-25 23:00:00+00:00	2024-04-25 23:10:00+00:00	165.225.33.6	
2	28506	13468	2024-04-25 23:00:00+00:00	2024-04-25 23:10:00+00:00	165.225.212.255	
3	30546	14278	2024-04-25 23:00:00+00:00	2024-04-25 23:10:00+00:00	136.226.64.114	
4	6526	13892	2024-04-25 23:00:00+00:00	2024-04-25 23:10:00+00:00	165.225.240.79	

5 rows × 29 columns

In [35]:
```
# visualization
```

In [36]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

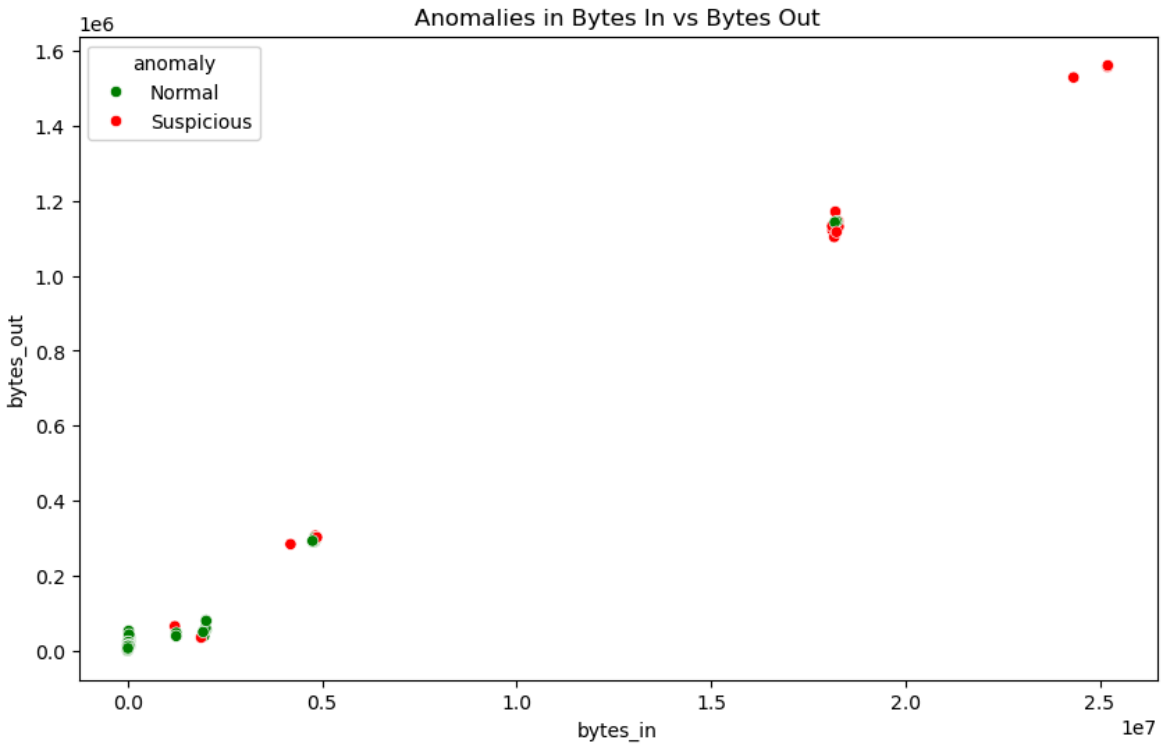
In [37]:
```
plt.figure(figsize=(15, 8))
sns.countplot(
    y='src_ip_country_code',
    data=df_transformed,
    order=df_transformed['src_ip_country_code'].value_counts().index
)
plt.title('Interaction Count by Source IP Country Code')
plt.show()
```



In [38]:
```
from sklearn.ensemble import IsolationForest

features = df_transformed[['bytes_in', 'bytes_out', 'session_duration', 'avg_pac

model = IsolationForest(contamination=0.05, random_state=42)

df_transformed['anomaly'] = model.fit_predict(features)

df_transformed['anomaly'] = df_transformed['anomaly'].apply(lambda x: 'Suspiciou

print(df_transformed['anomaly'].value_counts())
```

```
anomaly
Normal        267
Suspicious     15
Name: count, dtype: int64
```

In [39]:
```
plt.figure(figsize=(10, 6))
sns.scatterplot(
    x='bytes_in',
    y='bytes_out',
    hue='anomaly',
    data=df_transformed,
    palette={'Normal': 'green', 'Suspicious': 'red'}
)
plt.title('Anomalies in Bytes In vs Bytes Out')
plt.show()
```



In [40]:
```
# Green dots → Normal traffic

# Red dots → Suspicious connections detected by Isolation Forest
```

In [41]:
```
df_transformed.to_csv("cybersecurity_transformed_nidhi.csv", index=False)

print("File saved as cybersecurity_transformed.csv")
```

```
File saved as cybersecurity_transformed.csv
```

In [ ]: