```
# This Python 3 environment comes with many helpful analytics
libraries installed
# It is defined by the kaggle/python Docker image:
https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/"
directory
# For example, running this (by clicking run or pressing Shift+Enter)
will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/)
that gets preserved as output when you create a version using "Save &
Run All"
# You can also write temporary files to /kaggle/temp/, but they won't
be saved outside of the current session
```

```
/kaggle/input/cyber-security-attacks/README.md
/kaggle/input/cyber-security-attacks/cybersecurity_attacks.csv
```

Import necessary libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Load the dataset

```
df=pd.read_csv("/kaggle/input/cyber-security-attacks/
cybersecurity_attacks.csv")
```

Basic Data Exploration

```
df.head()
```

```
             Timestamp Source IP Address Destination IP Address
Source Port  \
0  2023-05-30 06:33:58     103.216.15.12           84.9.164.252
31225
1  2020-08-26 07:08:30    78.199.217.198         66.191.137.154
17245
2  2022-11-13 08:23:25      63.79.210.48          198.219.82.17
16811
3  2023-07-02 10:38:46     163.42.196.10        101.228.192.255
20018
4  2023-07-16 13:11:07     71.166.185.76        189.243.174.238
6131

   Destination Port Protocol  Packet Length Packet Type Traffic
Type  \
0             17616     ICMP            503        Data         HTTP

1             48166     ICMP           1174        Data         HTTP

2             53600      UDP            306     Control         HTTP

3             32534      UDP            385        Data         HTTP

4             26646      TCP           1462        Data          DNS


                                        Payload Data  ... Action Taken
\
0  Qui natus odio asperiores nam. Optio nobis ius...  ...       Logged

1  Aperiam quos modi officiis veritatis rem. Omni...  ...      Blocked

2  Perferendis sapiente vitae soluta. Hic delectu...  ...      Ignored

3  Totam maxime beatae expedita explicabo porro l...  ...      Blocked

4  Odit nesciunt dolorem nisi iste iusto. Animi v...  ...      Blocked


   Severity Level User Information  \
0             Low    Reyansh Dugal
1             Low       Sumer Rana
2             Low     Himmat Karpe
3          Medium       Fateh Kibe
4             Low     Dhanush Chad

                                  Device Information Network
Segment  \
0  Mozilla/5.0 (compatible; MSIE 8.0; Windows NT ...       Segment A

1  Mozilla/5.0 (compatible; MSIE 8.0; Windows NT ...       Segment B
```

```
2  Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...       Segment C

3  Mozilla/5.0 (Macintosh; PPC Mac OS X 10_11_5; ...       Segment B

4  Mozilla/5.0 (compatible; MSIE 5.0; Windows NT ...       Segment C


    Geo-location Data Proxy Information Firewall Logs IDS/IPS
Alerts  \
0  Jamshedpur, Sikkim      150.9.97.135      Log Data            NaN

1  Bilaspur, Nagaland               NaN      Log Data            NaN

2   Bokaro, Rajasthan    114.133.48.179      Log Data     Alert Data

3  Jaunpur, Rajasthan               NaN           NaN     Alert Data

4  Anantapur, Tripura     149.6.110.119           NaN     Alert Data


  Log Source
0     Server
1   Firewall
2   Firewall
3   Firewall
4   Firewall

[5 rows x 25 columns]

df.info() # Dataset Information

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40000 entries, 0 to 39999
Data columns (total 25 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Timestamp               40000 non-null  object
 1   Source IP Address       40000 non-null  object
 2   Destination IP Address  40000 non-null  object
 3   Source Port             40000 non-null  int64
 4   Destination Port        40000 non-null  int64
 5   Protocol                40000 non-null  object
 6   Packet Length           40000 non-null  int64
 7   Packet Type             40000 non-null  object
 8   Traffic Type            40000 non-null  object
 9   Payload Data            40000 non-null  object
 10  Malware Indicators      20000 non-null  object
 11  Anomaly Scores          40000 non-null  float64
 12  Alerts/Warnings         19933 non-null  object
 13  Attack Type             40000 non-null  object
```

```
 14  Attack Signature        40000 non-null  object
 15  Action Taken            40000 non-null  object
 16  Severity Level          40000 non-null  object
 17  User Information        40000 non-null  object
 18  Device Information      40000 non-null  object
 19  Network Segment         40000 non-null  object
 20  Geo-location Data       40000 non-null  object
 21  Proxy Information       20149 non-null  object
 22  Firewall Logs           20039 non-null  object
 23  IDS/IPS Alerts          19950 non-null  object
 24  Log Source              40000 non-null  object
dtypes: float64(1), int64(3), object(21)
memory usage: 7.6+ MB
```

```
df.shape  # Dataset Shape
```

```
(40000, 25)
```

```
df.describe()   # Statistics
```

	Source Port	Destination Port	Packet Length	Anomaly Scores
count	40000.000000	40000.000000	40000.000000	40000.000000
mean	32970.356450	33150.868650	781.452725	50.113473
std	18560.425604	18574.668842	416.044192	28.853598
min	1027.000000	1024.000000	64.000000	0.000000
25%	16850.750000	17094.750000	420.000000	25.150000
50%	32856.000000	33004.500000	782.000000	50.345000
75%	48928.250000	49287.000000	1143.000000	75.030000
max	65530.000000	65535.000000	1500.000000	100.000000

Data Cleaning

```
df.isnull().sum()  # Missing Values
```

```
Timestamp                       0
Source IP Address               0
Destination IP Address          0
Source Port                     0
Destination Port                0
Protocol                        0
Packet Length                   0
Packet Type                     0
Traffic Type                    0
Payload Data                    0
Malware Indicators          20000
Anomaly Scores                  0
Alerts/Warnings             20067
Attack Type                     0
```

```
Attack Signature              0
Action Taken                  0
Severity Level                0
User Information              0
Device Information            0
Network Segment               0
Geo-location Data             0
Proxy Information         19851
Firewall Logs             19961
IDS/IPS Alerts            20050
Log Source                    0
dtype: int64
```

```
df.dropna(inplace=True)  # Handling Missing Values
```

```
df.isnull().sum()
```

```
Timestamp                 0
Source IP Address         0
Destination IP Address    0
Source Port               0
Destination Port          0
Protocol                  0
Packet Length             0
Packet Type               0
Traffic Type              0
Payload Data              0
Malware Indicators        0
Anomaly Scores            0
Alerts/Warnings           0
Attack Type               0
Attack Signature          0
Action Taken              0
Severity Level            0
User Information          0
Device Information        0
Network Segment           0
Geo-location Data         0
Proxy Information         0
Firewall Logs             0
IDS/IPS Alerts            0
Log Source                0
dtype: int64
```

```
df.shape
```

```
(1237, 25)
```

```
df.duplicated().sum()
```

```
0
```

```
df.head()
```

```
                Timestamp Source IP Address Destination IP Address  \
2     2022-11-13 08:23:25      63.79.210.48          198.219.82.17
7     2023-02-12 07:13:17      11.48.99.245         178.157.14.116
46    2023-05-16 13:01:56    170.211.138.30         172.97.181.148
97    2021-10-25 04:23:15   129.189.216.143         197.202.27.160
105   2022-10-30 05:51:47      62.75.113.77         216.196.28.158

     Source Port  Destination Port Protocol  Packet Length Packet Type
\
2          16811             53600      UDP            306     Control

7          34489             20396     ICMP           1022        Data

46         25022              6593      TCP            554     Control

97         19199             27928     ICMP           1178        Data

105        42864             48696     ICMP            765     Control


    Traffic Type                                           Payload
Data  ...  \
2           HTTP  Perferendis sapiente vitae soluta. Hic
delectu...  ...
7            DNS  Amet libero optio quidem praesentium libero.
E...  ...
46           DNS  Voluptate mollitia cupiditate necessitatibus
n...  ...
97          HTTP  Eaque deserunt nemo ad voluptate. Aliquid
rem ...  ...
105          DNS  Dolores vitae neque velit maiores.\
nReprehende...  ...

    Action Taken  Severity Level User Information  \
2        Ignored             Low     Himmat Karpe
7         Logged            High     Yuvaan Dubey
46       Blocked            High   Aradhya Kamdar
97       Ignored            High      Ira Kapadia
105      Blocked             Low      Arnav Krish

                                    Device Information Network Segment
\
2    Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...       Segment C

7    Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_7_6...       Segment A

46   Mozilla/5.0 (iPod; U; CPU iPhone OS 3_3 like M...       Segment A

97   Mozilla/5.0 (compatible; MSIE 8.0; Windows 95;...       Segment C
```

```
105  Mozilla/5.0 (iPod; U; CPU iPhone OS 3_0 like M...       Segment C


            Geo-location Data Proxy Information Firewall Logs IDS/IPS
Alerts  \
2           Bokaro, Rajasthan    114.133.48.179      Log Data
Alert Data
7    Phagwara, Andhra Pradesh      192.31.159.5      Log Data
Alert Data
46           Amravati, Kerala     95.170.137.42      Log Data
Alert Data
97       Tirunelveli, Gujarat    57.192.174.154      Log Data
Alert Data
105            Ranchi, Sikkim     199.194.2.180      Log Data
Alert Data

    Log Source
2     Firewall
7     Firewall
46    Firewall
97      Server
105     Server

[5 rows x 25 columns]
```

Data Visualization

```
df['Payload Data'].dtype

dtype('O')

# Convert your_data to a string

text = str(df['Payload Data'])
```

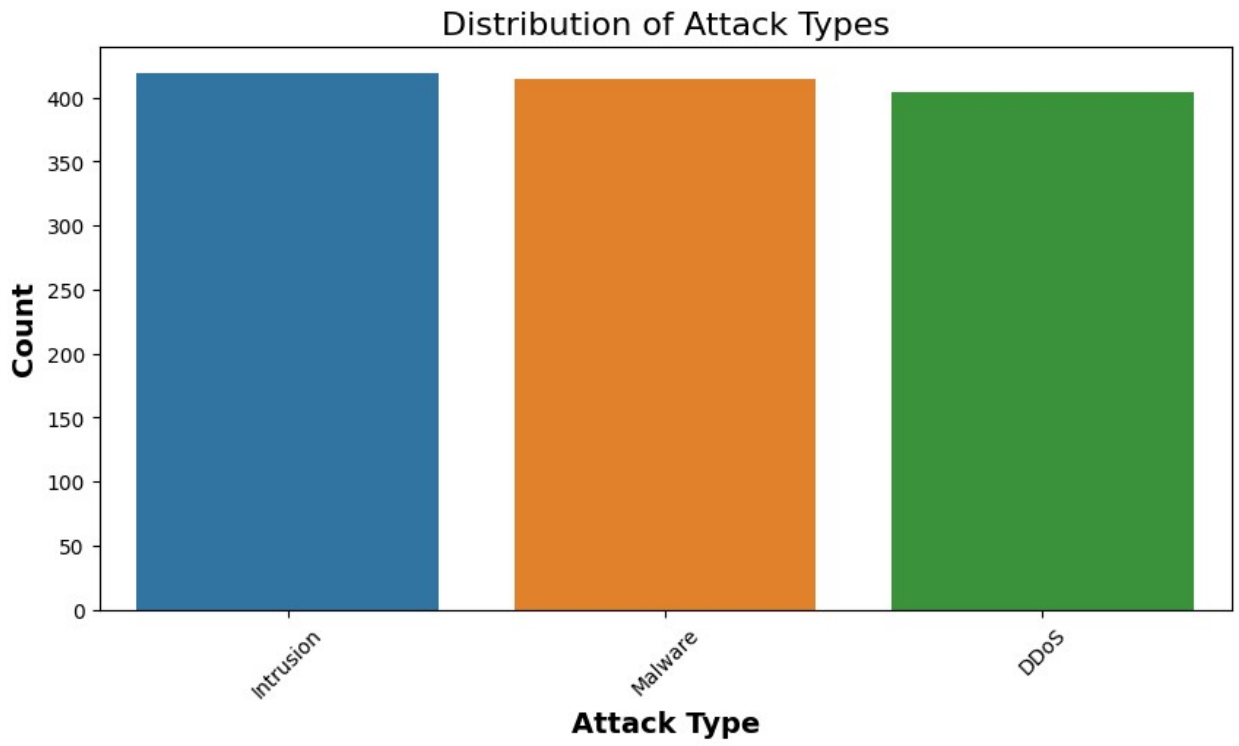
Word Cloud

```
from wordcloud import WordCloud

# Generate the word cloud
wordcloud = WordCloud(width=800, height=400,
background_color='white').generate(text)

# Display the word cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
```

```
plt.axis("off")
plt.show()
```

Bar Chart

```
# Visualize the distribution of attack types

attack_counts = df['Attack Type'].value_counts()

plt.figure(figsize=(10, 5))
sns.barplot(x=attack_counts.index , y=attack_counts)

plt.xlabel('Attack Type',fontsize=14, fontweight='bold')
plt.ylabel('Count',fontsize=14, fontweight='bold')
plt.title('Distribution of Attack Types', fontsize=16)

plt.xticks(rotation=45)
plt.show()

print(attack_counts)
```




```
Attack Type
Intrusion    419
Malware      414
DDoS         404
Name: count, dtype: int64
```

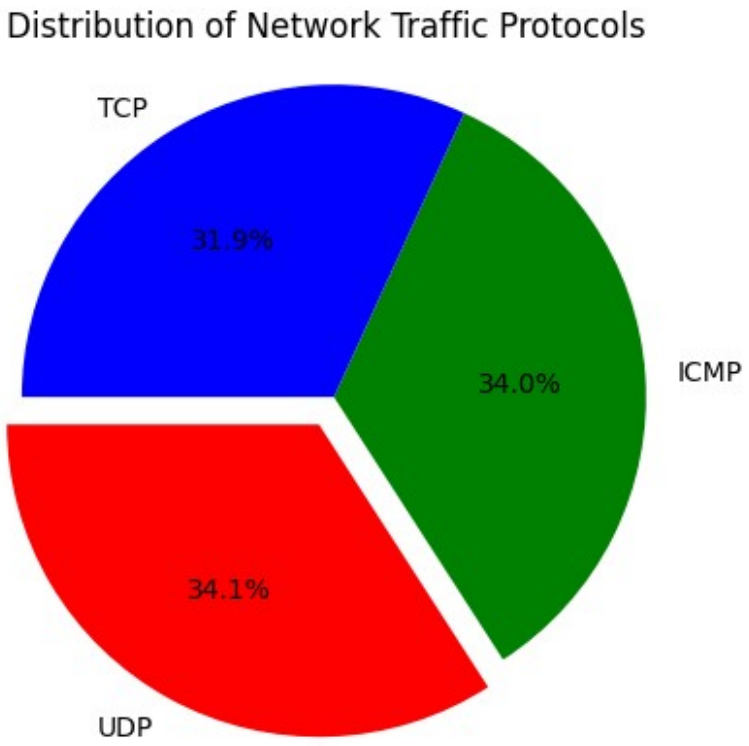
Pie Charts

```
df['Protocol'].value_counts()
```

```
Protocol
UDP     422
ICMP    421
TCP     394
Name: count, dtype: int64
```

```
# Data for the pie chart


labels = ['UDP', 'ICMP', 'TCP']
sizes = df['Protocol'].value_counts()  # Proportional sizes of each
category
colors = ['red', 'green', 'blue']  # Color for each category segment
explode = (0.1, 0, 0)  # Explode a slice if needed (0 means no
explosion)
```

```
# Create a pie chart
plt.pie(sizes, labels=labels, colors=colors, explode=explode,
autopct='%1.1f%%', startangle=180)


plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a
circle.
plt.title('Distribution of Network Traffic Protocols')


# Display the pie chart
plt.show()
```

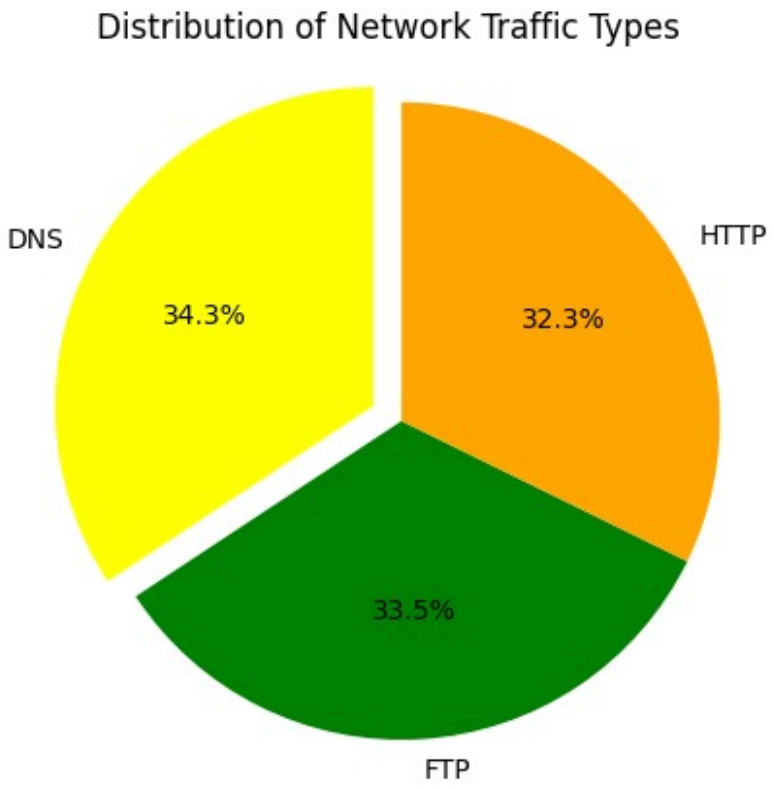



```
df['Traffic Type'].value_counts()
```

```
Traffic Type
DNS     424
FTP     414
HTTP    399
Name: count, dtype: int64
```

```
# Data for the pie chart

labels =['DNS','FTP','HTTP']
sizes = df['Traffic Type'].value_counts()
colors = ['yellow', 'green', 'orange']
```

```
explode = (0.1, 0, 0)

# Create a pie chart
plt.pie(sizes, labels=labels, colors=colors, explode=explode,
autopct='%1.1f%%', startangle=90)

plt.axis('equal')
plt.title('Distribution of Network Traffic Types')

# Display the pie chart
plt.show()
```

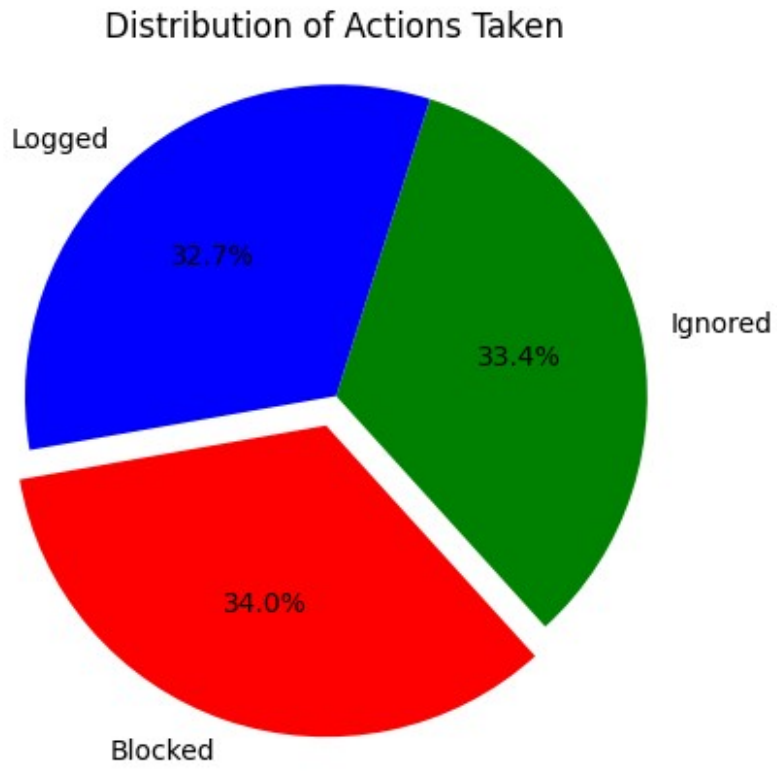



```
df['Action Taken'].value_counts()
```

```
Action Taken
Blocked    420
Ignored    413
Logged     404
Name: count, dtype: int64
```

```
labels =['Blocked','Ignored','Logged']
sizes =df['Action Taken'].value_counts()
colors = ['Red', 'green', 'blue']
explode = (0.1, 0, 0)
```

```
# Create a pie chart
plt.pie(sizes, labels=labels, colors=colors, explode=explode,
autopct='%1.1f%%', startangle=190)

plt.axis('equal')
plt.title('Distribution of Actions Taken')

# Display the pie chart
plt.show()
```

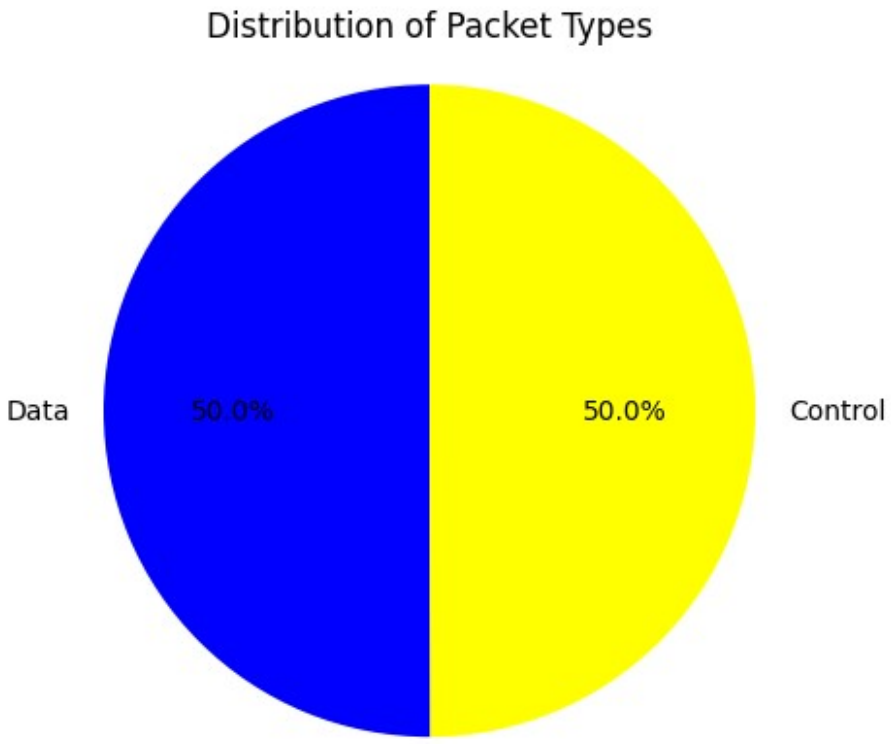



```
df['Packet Type'].value_counts()
```

```
Packet Type
Data       619
Control    618
Name: count, dtype: int64
```

```
# Data for the pie chart
labels =['Data','Control']
sizes = df['Packet Type'].value_counts()
colors = ['blue', 'yellow']
explode = (0, 0)

# Create a pie chart
```

```
plt.pie(sizes, labels=labels, colors=colors, explode=explode,
autopct='%1.1f%%', startangle=90)


plt.axis('equal')
plt.title('Distribution of Packet Types')

# Display the pie chart
plt.show()
```

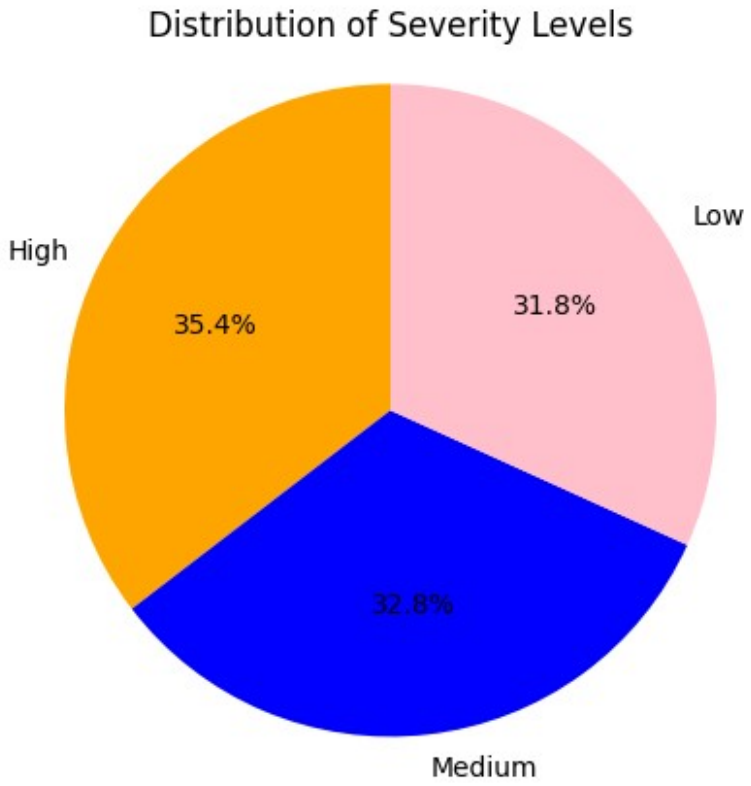



```
df['Severity Level'].value_counts()
```

```
Severity Level
High      438
Medium    406
Low       393
Name: count, dtype: int64
```

```
# Data for the pie chart
labels =['High','Medium','Low']
sizes =df['Severity Level'].value_counts()
colors = ['orange', 'blue','pink']
explode = (0, 0,0)

# Create a pie chart
```

```
plt.pie(sizes, labels=labels, colors=colors, explode=explode,
autopct='%1.1f%%', startangle=90)

plt.axis('equal')
plt.title('Distribution of Severity Levels')

# Display the pie chart
plt.show()
```




```
df['Log Source'].value_counts()
```

```
Log Source
Server      623
Firewall    614
Name: count, dtype: int64
```

```
labels =['Server','Firewall']
sizes =df['Log Source'].value_counts()
colors = ['blue', 'orange']
explode = (0, 0)

plt.pie(sizes, labels=labels, colors=colors, explode=explode,
autopct='%1.1f%%', startangle=270)
```

```
plt.axis('equal')
plt.title('Distribution of Log Sources')

# Display the pie chart
plt.show()
```