Global Cybersecurity Threats Analysis

Contents

1 Setup

1.1 Import Packages

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter
import seaborn as sns
```

1.2 Load data

```python
pd.set_option('display.max_columns', 50, 'display.width', 200)
df = pd.read_csv('data/Global_Cybersecurity_Threats_2015-2024.csv')
df.head()
```

```
|    | Country | Year | Attack Type       | Target Industry    | Financial Loss (in Million
↪  $) | Number of Affected Users | Attack Source | Security Vulnerability Type | Defense
↪  Mechanism Used | Incident Resolution Time (in Hours) |
|---+---------+------+-------------------+--------------------+--------------------------------+--------
| 0  | China   | 2019 | Phishing          | Education          |
↪  80.53 |                    773169 | Hacker Group  | Unpatched Software          | VPN
↪  |                              63 |
| 1  | China   | 2019 | Ransomware        | Retail             |
↪  62.19 |                    295961 | Hacker Group  | Unpatched Software          |
↪  Firewall                |                              71 |
| 2  | India   | 2017 | Man-in-the-Middle | IT                 |
↪  38.65 |                    605895 | Hacker Group  | Weak Passwords              | VPN
↪  |                              20 |
| 3  | UK      | 2024 | Ransomware        | Telecommunications |
↪  41.44 |                    659320 | Nation-state  | Social Engineering          |
↪  AI-based Detection      |                               7 |
| 4  | Germany | 2018 | Man-in-the-Middle | IT                 |
↪  74.41 |                    810682 | Insider       | Social Engineering          | VPN
↪  |                              68 |
```

2 Trend Analysis

2.1 Financial Losses

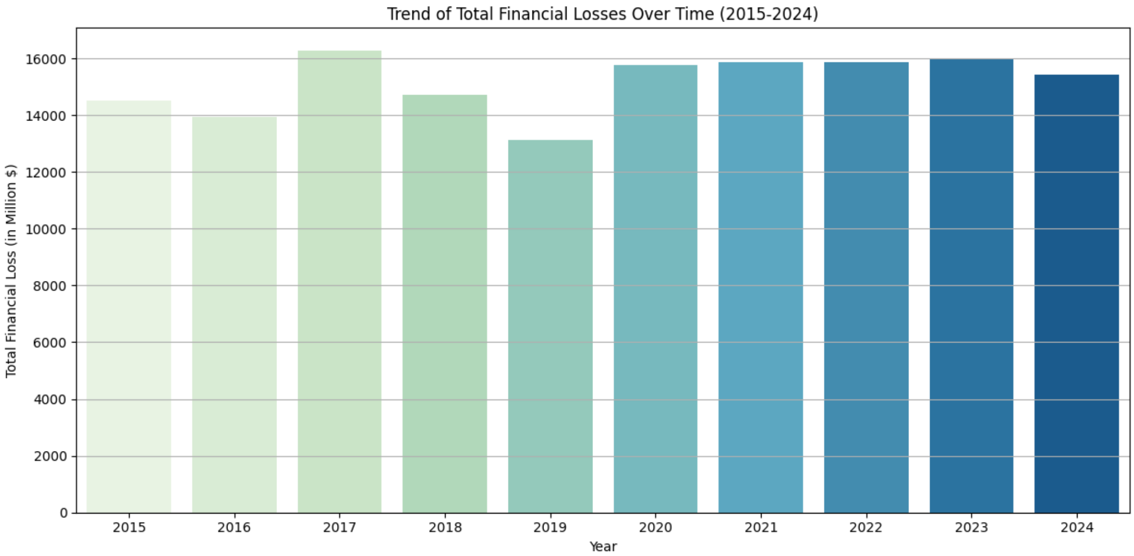
2.1.1 Overall

```
trend_df = df.groupby('Year')['Financial Loss (in Million $)'].sum()

plt.figure(figsize=(12, 6))
sns.barplot(x=trend_df.index,y=trend_df, palette="GnBu")

plt.title('Trend of Total Financial Losses Over Time (2015-2024)')
plt.xlabel('Year')
plt.ylabel('Total Financial Loss (in Million $)')
plt.grid(axis='y')

plt.tight_layout()
plt.show()
```



2.1.2 Breakdown by Countries

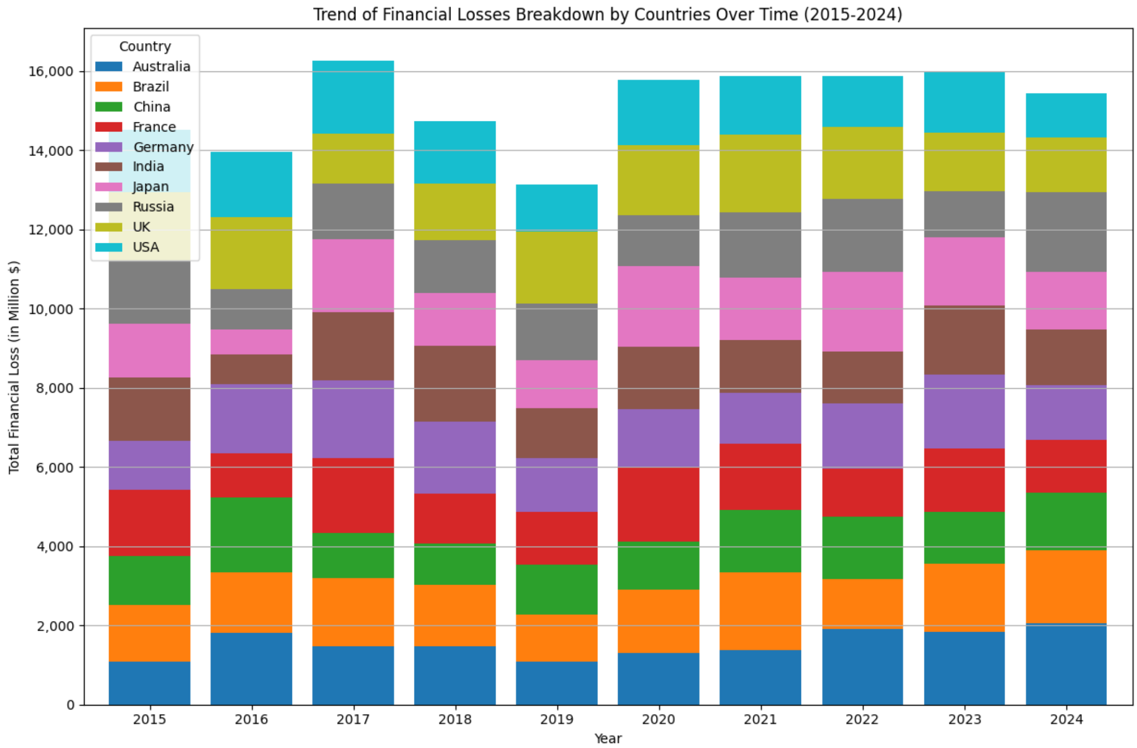
```
trend_df = df.groupby(['Year', 'Country'])['Financial Loss (in Million
↪ $)'].sum().unstack(fill_value=0)
trend_df
```

```
| Year | Australia | Brazil |              China |              France |              Germany
↪ |  India |              Japan |             Russia |                 UK |
↪ USA |
|------+-----------+---------+--------------------+--------------------+--------------------+---------+
| 2015 |    1083.64 | 1433.53 |            1230.41 |             1678.81 | 1245.1399999999999
↪ | 1588.71 |            1346.21 |            1608.85 | 1729.6200000000001 |
↪ 1565.29 |
```

```
| 2016 |    1823.38 | 1507.33 |             1890.11 |             1126.03 |             1752.33
↪  |  739.05 |              626.25 |             1014.82 |             1837.54 |
↪  1630.42 |
| 2017 |    1472.65 |  1711.9 |             1146.21 |             1891.72 |              1974.1
↪  | 1701.02 |             1847.73 |             1404.13 |             1278.32 |
↪  1833.9 |
| 2018 |    1483.63 | 1533.38 |             1054.21 |             1251.98 |             1815.61
↪  | 1918.73 |             1331.31 |             1346.33 |             1419.53 |
↪  1565.77 |
| 2019 |    1090.43 | 1183.03 |             1258.51 |             1343.36 |             1350.34
↪  | 1254.71 | 1215.3700000000001 |             1431.94 |             1808.93 |
↪  1198.0700000000002 |
| 2020 |    1291.54 | 1611.79 |             1220.73 | 1859.8899999999999 |             1480.43
↪  |  1580.0 |             2038.73 |             1260.93 |              1789.7 |
↪  1634.21 |
| 2021 |    1372.29 | 1977.69 |             1572.86 |              1669.3 |             1278.91
↪  | 1334.33 |             1586.41 | 1638.1100000000001 |             1973.24 |
↪  1470.27 |
| 2022 |    1897.56 | 1270.53 | 1564.6499999999999 |             1216.86 |             1649.13
↪  | 1306.11 |             2018.56 |             1841.52 |             1822.47 |
↪  1283.47 |
| 2023 |    1841.43 |  1709.4 |             1313.26 |             1599.39 |             1877.72
↪  |  1725.2 |             1732.56 |             1171.56 |             1475.31 |
↪  1512.25 |
| 2024 |    2046.45 | 1844.04 |             1463.52 |             1334.94 |             1369.53
↪  | 1418.26 |             1454.21 |             2016.54 |             1368.33 |
↪  1118.47 |
```

```python
# Plotting the data as a bar chart
trend_df.plot(kind='bar', stacked=True, figsize=(12, 8), width=0.8)

plt.title('Trend of Financial Losses Breakdown by Countries Over Time (2015-2024)')
plt.xlabel('Year')
plt.ylabel('Total Financial Loss (in Million $)')
plt.xticks(rotation=0)  # Keep x-axis labels horizontal
plt.legend(title='Country')
plt.grid(axis='y')
plt.gca().yaxis.set_major_formatter(FuncFormatter(lambda x, _: f'{int(x):,}')) # Apply
↪ number formatting
plt.tight_layout()
plt.show()
```

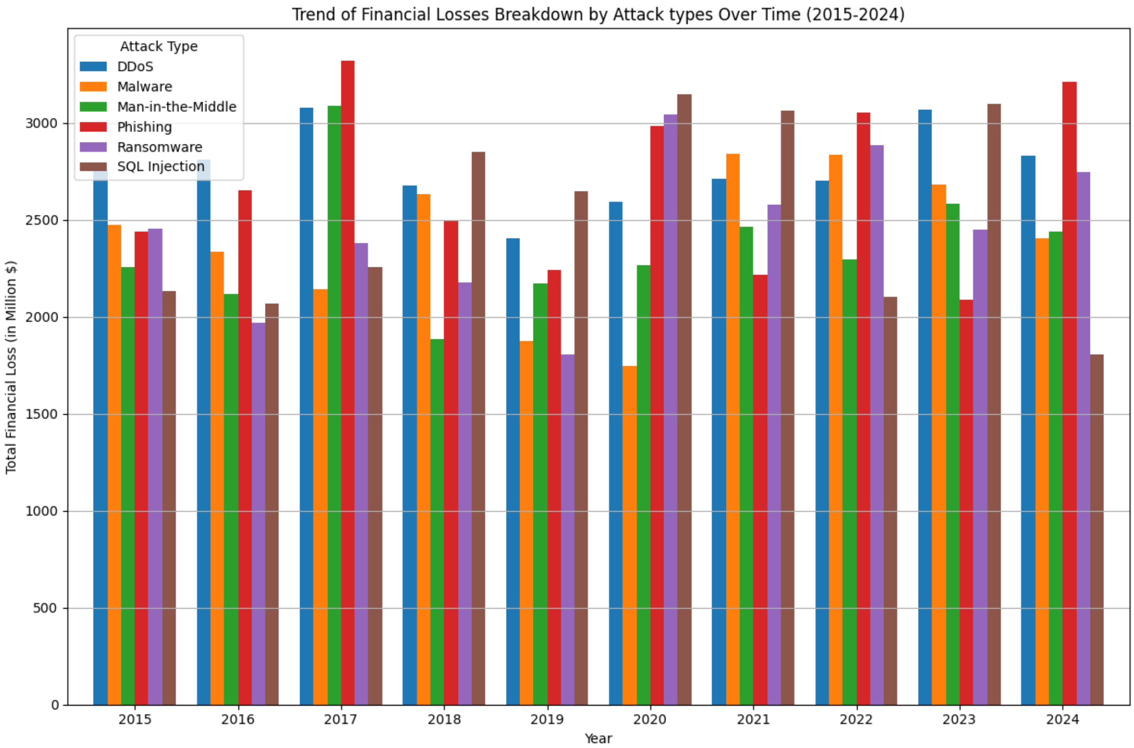


2.1.3 Breakdown by Attack Types

```
# Group by Year and Attack Type, summing financial losses
trend_df = df.groupby(['Year', 'Attack Type'])['Financial Loss (in Million
↪ $)'].sum().unstack(fill_value=0)

# Plotting the data as a bar chart
trend_df.plot(kind='bar', figsize=(12, 8), width=0.8)

plt.title('Trend of Financial Losses Breakdown by Attack types Over Time (2015-2024)')
plt.xlabel('Year')
plt.ylabel('Total Financial Loss (in Million $)')
plt.xticks(rotation=0)  # Keep x-axis labels horizontal
plt.legend(title='Attack Type')
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```

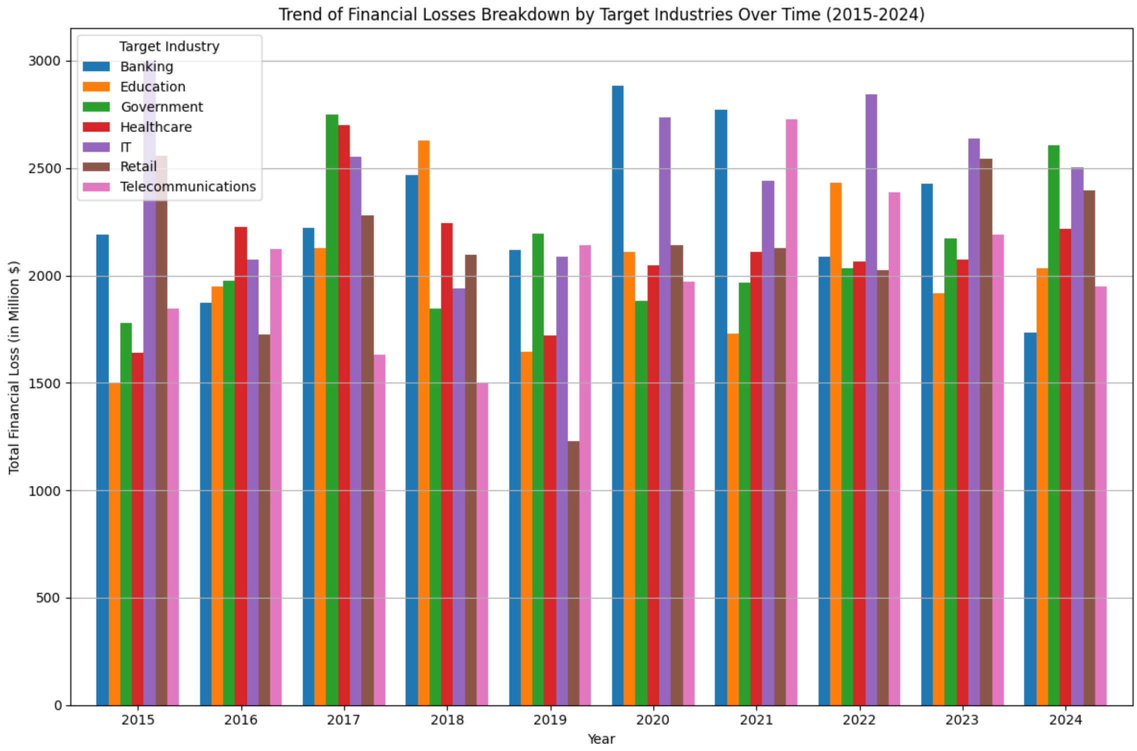


2.1.4 Breakdown by Target Industries

```
# Group by Year and Target Industry, summing financial losses
trend_df = df.groupby(['Year', 'Target Industry'])['Financial Loss (in Million
↪  $)'].sum().unstack(fill_value=0)

# Plotting the data as a bar chart
trend_df.plot(kind='bar', figsize=(12, 8), width=0.8)

plt.title('Trend of Financial Losses Breakdown by Target Industries Over Time (2015-2024)')
plt.xlabel('Year')
plt.ylabel('Total Financial Loss (in Million $)')
plt.xticks(rotation=0)  # Keep x-axis labels horizontal
plt.legend(title='Target Industry')
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```

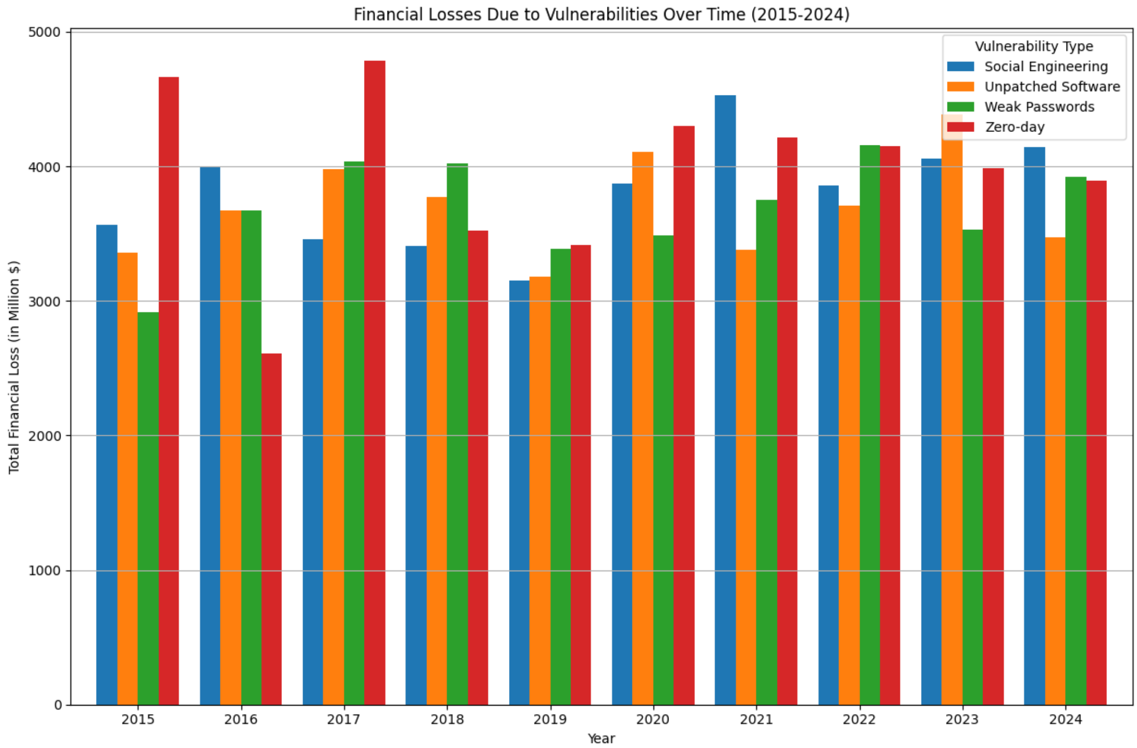


2.1.5 Breakdown by Vulnerability Types

```
# Group by Year and Security Vulnerability Type, summing financial losses
vulnerability_df = df.groupby(['Year', 'Security Vulnerability Type'])['Financial Loss (in
↪  Million $)'].sum().unstack(fill_value=0)

# Plotting the data as a bar chart
vulnerability_df.plot(kind='bar', figsize=(12, 8), width=0.8)

plt.title('Financial Losses Due to Vulnerabilities Over Time (2015-2024)')
plt.xlabel('Year')
plt.ylabel('Total Financial Loss (in Million $)')
plt.xticks(rotation=0)  # Keep x-axis labels horizontal
plt.legend(title='Vulnerability Type')
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```



2.2 Number of Affected Users

2.2.1 Overall

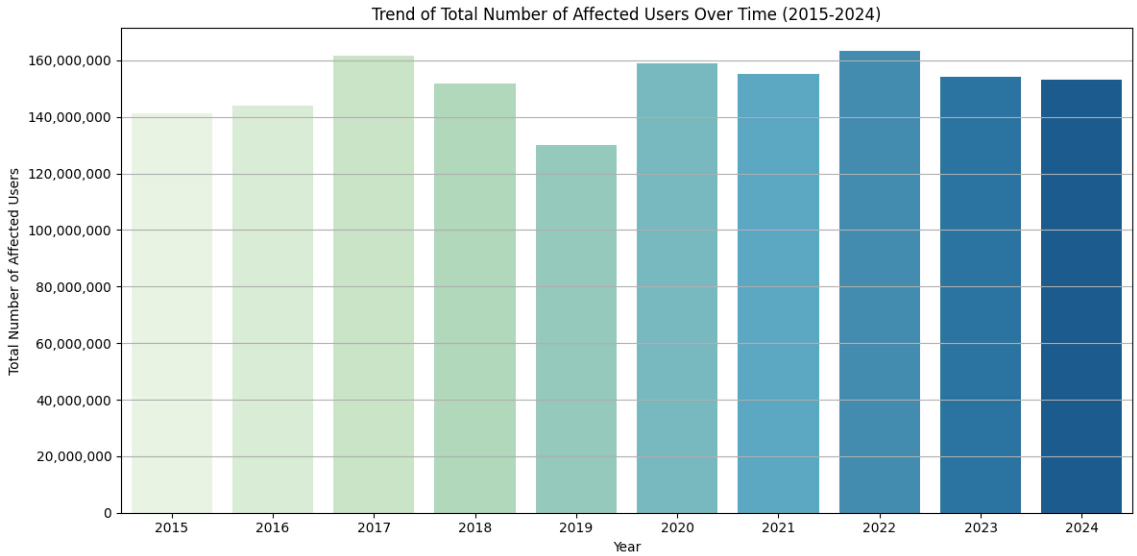
```
trend_df = df.groupby('Year')['Number of Affected Users'].sum()

plt.figure(figsize=(12, 6))
sns.barplot(x=trend_df.index,y=trend_df, palette="GnBu")

plt.title('Trend of Total Number of Affected Users Over Time (2015-2024)')
plt.xlabel('Year')
plt.ylabel('Total Number of Affected Users')
plt.grid(axis='y')

# Apply number formatting
plt.gca().yaxis.set_major_formatter(FuncFormatter(lambda x, _: f'{int(x):,}'))

plt.tight_layout()
plt.show()
```



2.2.2 Breakdown by Countries

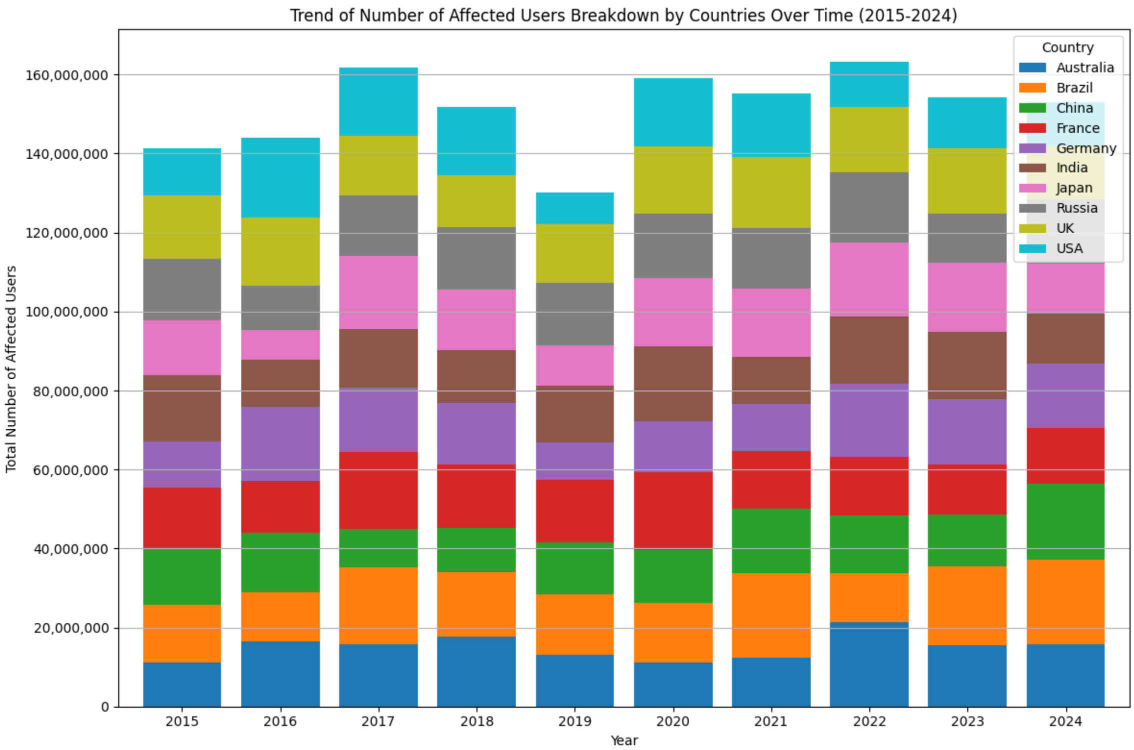
```
trend_df = df.groupby(['Year', 'Country'])['Number of Affected
↪ Users'].sum().unstack(fill_value=0)
trend_df
```

Year	Australia	Brazil	China	France	Germany	India	Japan	Russi
2015	11071355	14625687	14315760	15389278	11814325	16632245	13971922	1560130
2016	16522223	12412484	14958638	13274133	18645691	12085161	7457248	1110971
2017	15829968	19398597	9732124	19444809	16239690	14927191	18421875	1547537
2018	17636180	16441947	11024271	16231968	15503497	13355822	15287249	1588555
2019	12995593	15514625	12943762	15916554	9420383	14398125	10207336	1575812
2020	11011231	15196610	13332023	19716201	12919945	19110451	17114862	1631422
2021	12383861	21467873	16158640	14749463	11751905	12091545	17272985	1518303
2022	21277139	12476964	14653503	14739242	18523873	17033943	18722065	1793242
2023	15472616	19925490	13188428	12589421	16617789	17031851	17499894	1250099
2024	15811664	21346703	19273789	14178073	16238260	12512325	12756378	1643109

```
# Plotting the data as a bar chart
trend_df.plot(kind='bar', stacked=True, figsize=(12, 8), width=0.8)

plt.title('Trend of Number of Affected Users Breakdown by Countries Over Time (2015-2024)')
plt.xlabel('Year')
plt.ylabel('Total Number of Affected Users')
plt.xticks(rotation=0)  # Keep x-axis labels horizontal
plt.legend(title='Country')
plt.grid(axis='y')
plt.gca().yaxis.set_major_formatter(FuncFormatter(lambda x, _: f'{int(x):,}')) # Apply
↪ number formatting
```

```
plt.tight_layout()
plt.show()
```

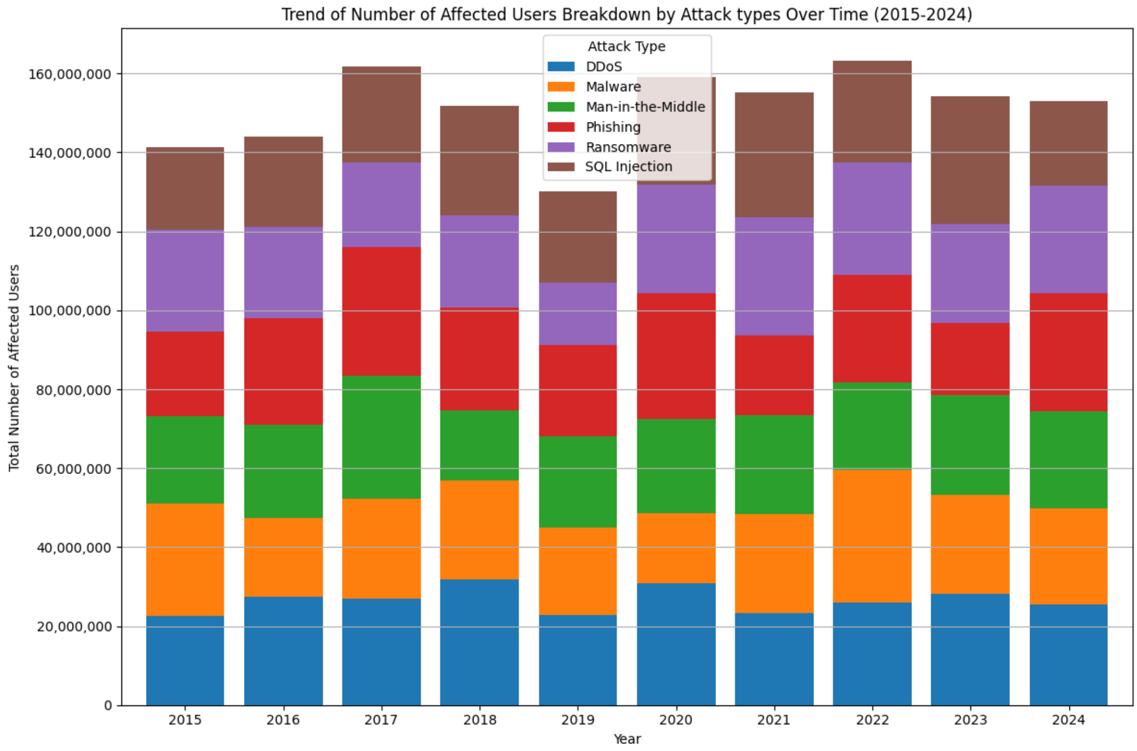


2.2.3 Breakdown by Attack Types

```
trend_df = df.groupby(['Year', 'Attack Type'])['Number of Affected
↪ Users'].sum().unstack(fill_value=0)

# Plotting the data as a bar chart
trend_df.plot(kind='bar', stacked=True, figsize=(12, 8), width=0.8)

plt.title('Trend of Number of Affected Users Breakdown by Attack types Over Time
↪ (2015-2024)')
plt.xlabel('Year')
plt.ylabel('Total Number of Affected Users')
plt.xticks(rotation=0)  # Keep x-axis labels horizontal
plt.legend(title='Attack Type')
plt.grid(axis='y')
plt.gca().yaxis.set_major_formatter(FuncFormatter(lambda x, _: f'{int(x):,}')) # Apply
↪ number formatting
plt.tight_layout()
plt.show()
```

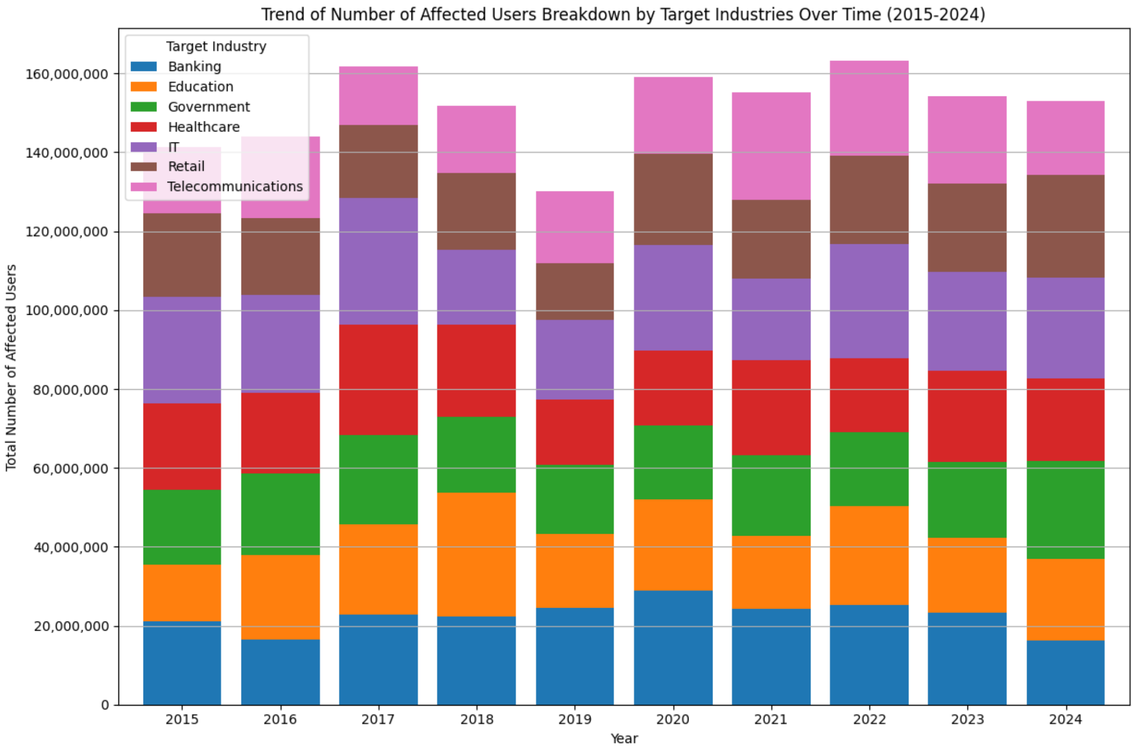


2.2.4 Breakdown by Target Industries

```
trend_df = df.groupby(['Year', 'Target Industry'])['Number of Affected
↪ Users'].sum().unstack(fill_value=0)

# Plotting the data as a bar chart
trend_df.plot(kind='bar', stacked=True, figsize=(12, 8), width=0.8)

plt.title('Trend of Number of Affected Users Breakdown by Target Industries Over Time
↪ (2015-2024)')
plt.xlabel('Year')
plt.ylabel('Total Number of Affected Users')
plt.xticks(rotation=0)  # Keep x-axis labels horizontal
plt.legend(title='Target Industry')
plt.grid(axis='y')
plt.gca().yaxis.set_major_formatter(FuncFormatter(lambda x, _: f'{int(x):,}')) # Apply
↪ number formatting
plt.tight_layout()
plt.show()
```

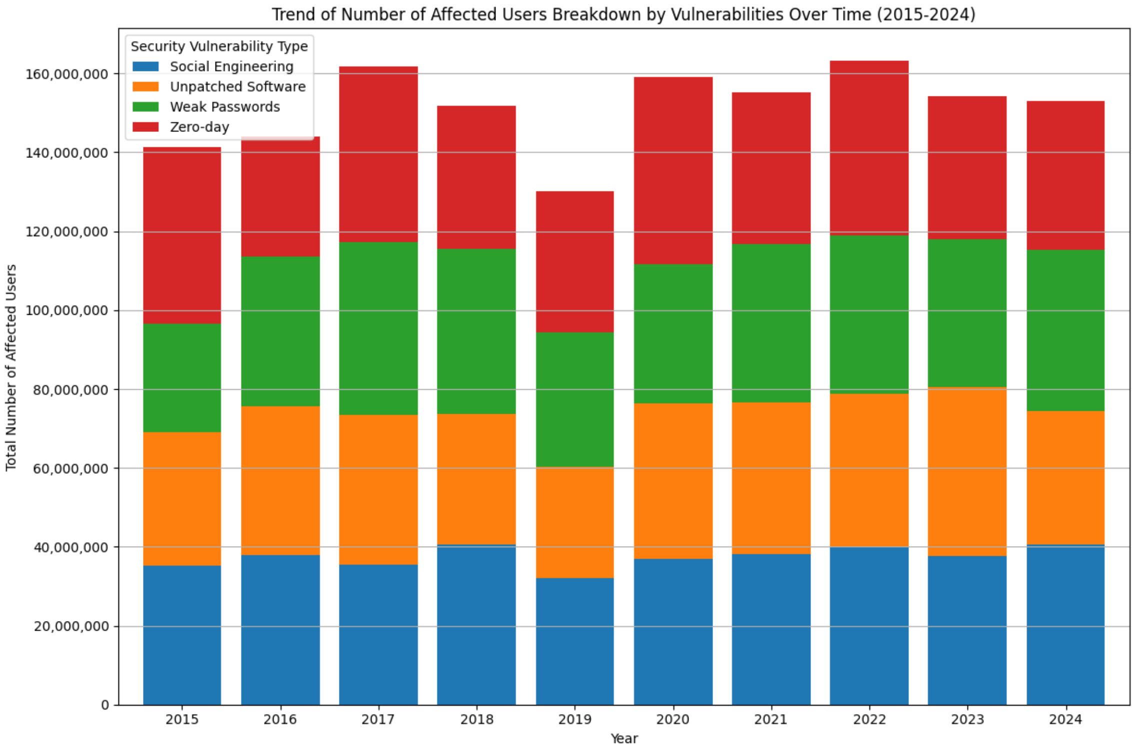


2.2.5 Breakdown by Vulnerability Types

```
trend_df = df.groupby(['Year', 'Security Vulnerability Type'])['Number of Affected
↪ Users'].sum().unstack(fill_value=0)

# Plotting the data as a bar chart
trend_df.plot(kind='bar', stacked=True, figsize=(12, 8), width=0.8)

plt.title('Trend of Number of Affected Users Breakdown by Vulnerabilities Over Time
↪ (2015-2024)')
plt.xlabel('Year')
plt.ylabel('Total Number of Affected Users')
plt.xticks(rotation=0)  # Keep x-axis labels horizontal
plt.legend(title='Security Vulnerability Type')
plt.grid(axis='y')
plt.gca().yaxis.set_major_formatter(FuncFormatter(lambda x, _: f'{int(x):,}')) # Apply
↪ number formatting
plt.tight_layout()
plt.show()
```



3 Geographical Analysis

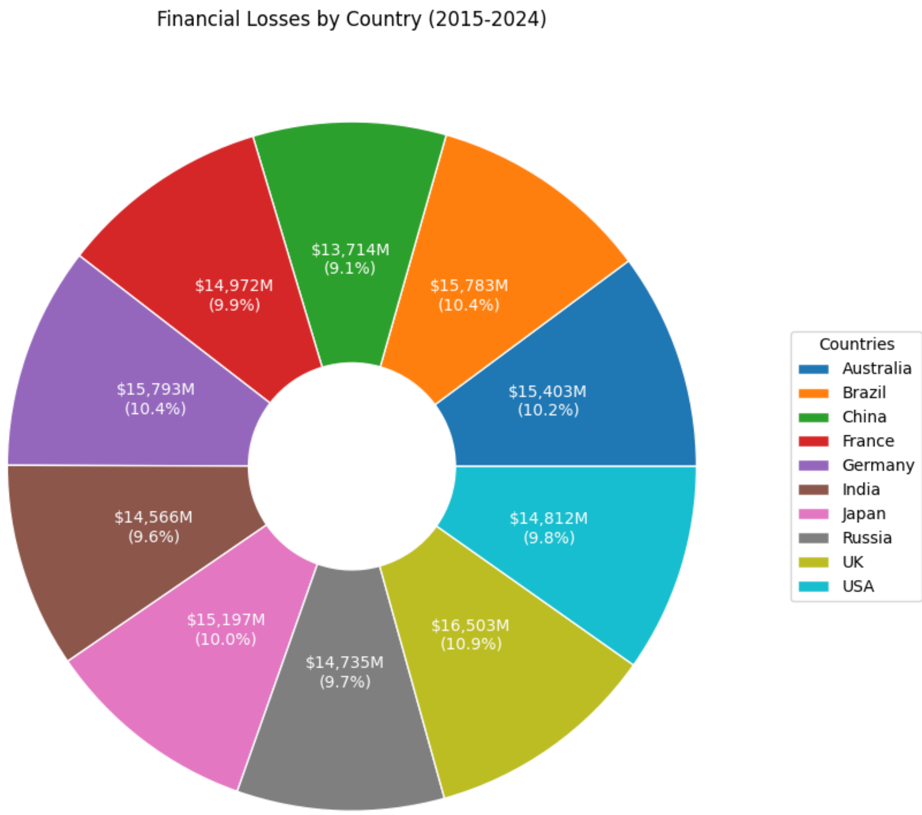
- Compare the frequency and impact of cyberattacks across different countries.
- Identify which countries are most affected by specific attack types.

3.1 Financial Losses

3.1.1 Overall

```
grouped_df = df.groupby('Country')['Financial Loss (in Million $)'].sum()

# Plotting the pie chart
plt.figure(figsize=(10, 8))
wedges, texts, autotexts = plt.pie(grouped_df, autopct=lambda
↪ pct:f"${round(grouped_df.sum()*pct/100):,}M\n({pct:.1f}%)", textprops=dict(color="w"),
↪ wedgeprops=dict(width=0.7, edgecolor='w'))
plt.legend(wedges, grouped_df.index, title="Countries", loc="center left",
↪ bbox_to_anchor=(1, 0, 0.5, 1))
plt.title('Financial Losses by Country (2015-2024)')
plt.tight_layout()
plt.show()
```



3.1.2 Breakdown by Attack Types

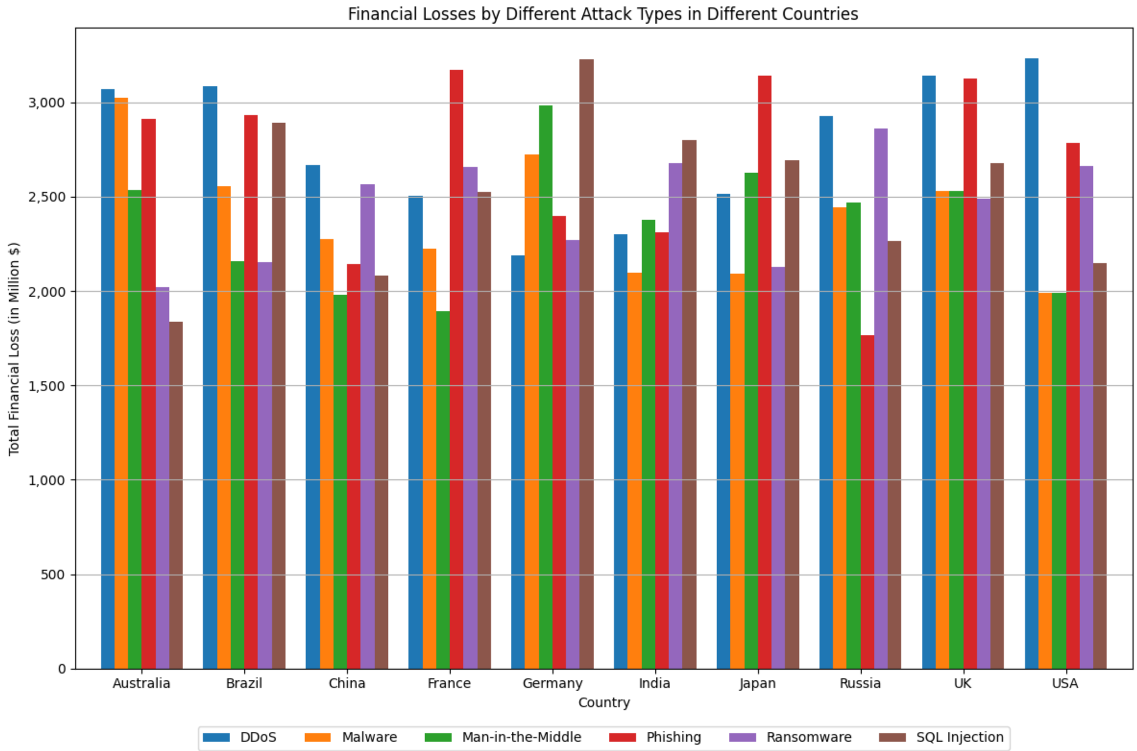
```
grouped_df = df.groupby(['Country', 'Attack Type'])['Financial Loss (in Million
↪ $)'].sum().unstack(fill_value=0)
grouped_df
```

Country	DDoS	Malware	Man-in-the-Middle	Phishing	Ransomware
Australia	3071.53	3026.17	2534.89	2911.85	2019.07
Brazil	3083.66	2557.21	2160.92	2934.49	2152.6
China	2665.38	2276.29	1981.22	2143.7400000000002	2566.17
France	2504.98	2223.92	1891.18	3170.87	2657.22
Germany	2188.92	2726.08	2986.04	2396.63	2269.19
India	2299.43	2097.0	2376.06	2313.18	2678.64
Japan	2515.52	2094.56	2626.23	3141.36	2125.44
Russia	2925.27	2446.15	2471.18	1767.5	2859.2400000000002
UK	3143.2	2528.71	2531.63	3128.33	2491.42
USA	3233.03	1991.86	1991.52	2785.34	2660.33

```
# Plotting the data as a bar chart
```

```
grouped_df.plot(kind='bar', figsize=(12, 8), width=0.8)

plt.title('Financial Losses by Different Attack Types in Different Countries')
plt.xlabel('Country')
plt.ylabel('Total Financial Loss (in Million $)')
plt.xticks(rotation=0)  # Keep x-axis labels horizontal
plt.legend(ncol=6, loc="upper center", bbox_to_anchor=(0.5,-0.08))
plt.grid(axis='y')
plt.gca().yaxis.set_major_formatter(FuncFormatter(lambda x, _: f'{int(x):,}')) # Apply
↪ number formatting
plt.tight_layout()
plt.show()
```



3.1.3 Breakdown by Target Industries

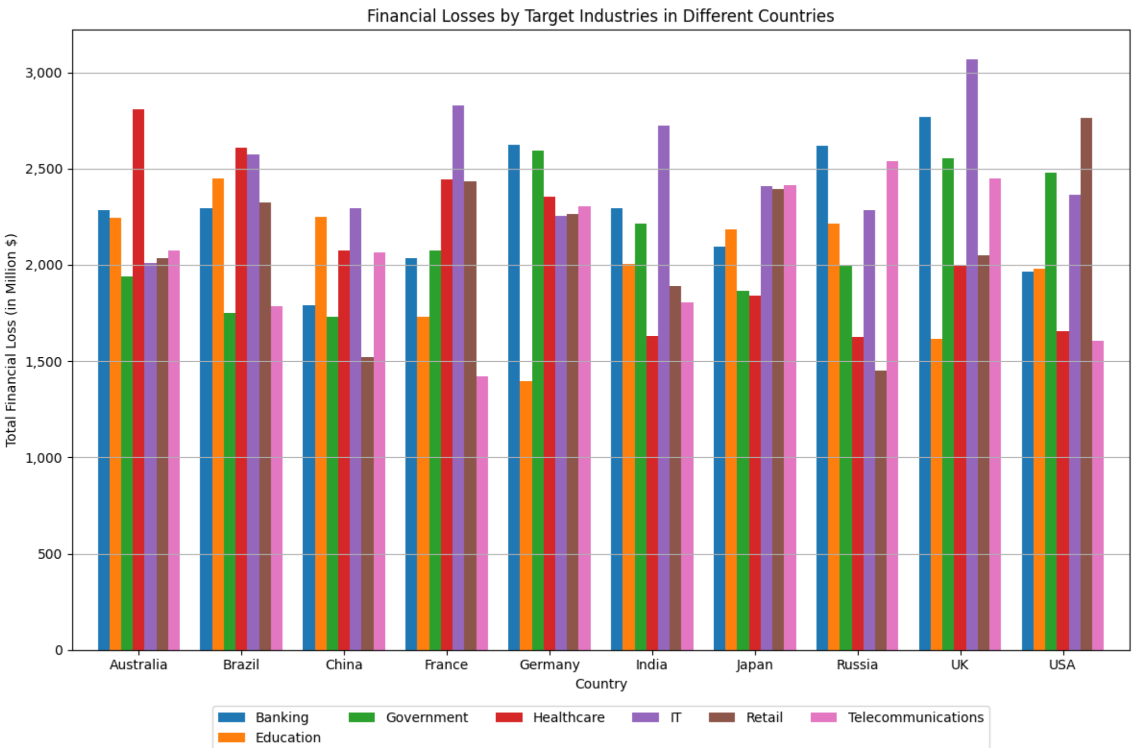
```
grouped_df = df.groupby(['Country', 'Target Industry'])['Financial Loss (in Million
↪ $)'].sum().unstack(fill_value=0)
grouped_df
```

```
# Plotting the data as a bar chart
grouped_df.plot(kind='bar', figsize=(12, 8), width=0.8)

plt.title('Financial Losses by Target Industries in Different Countries')
plt.xlabel('Country')
plt.ylabel('Total Financial Loss (in Million $)')
plt.xticks(rotation=0)  # Keep x-axis labels horizontal
```

Country	Banking	Education	Government	Healthcare	IT	R
Australia	2285.81	2243.91	1941.91	2808.7	2011.6399999999999	203
Brazil	2292.02	2449.51	1749.89	2609.41	2575.09	23
China	1788.04	2247.96	1728.75	2073.64	2293.12	152
France	2036.86	1728.59	2076.87	2445.17	2827.31	243
Germany	2622.86	1397.1	2591.92	2356.53	2256.15	226
India	2296.45	2004.3799999999999	2215.0	1631.46	2723.52	189
Japan	2092.3	2186.71	1864.9	1838.63	2406.81	239
Russia	2620.95	2214.73	2000.22	1627.85	2284.95	144
UK	2769.94	1616.99	2554.67	1996.82	3067.59	204
USA	1967.16	1981.55	2481.2	1653.08	2363.65	276

```
plt.legend(ncol=6, loc="upper center", bbox_to_anchor=(0.5,-0.08))
plt.grid(axis='y')
plt.gca().yaxis.set_major_formatter(FuncFormatter(lambda x, _: f'{int(x):,}')) # Apply
↪ number formatting
plt.tight_layout()
plt.show()
```



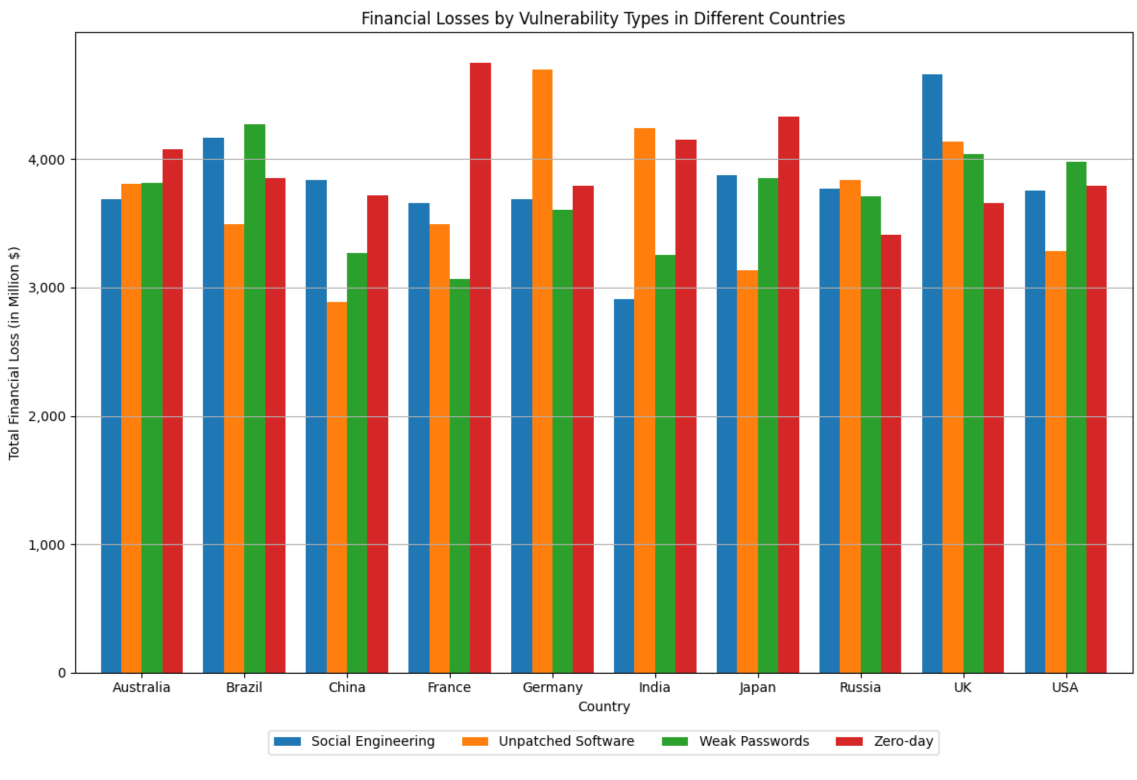
3.1.4 by Target Industries

```
grouped_df = df.groupby(['Country', 'Security Vulnerability Type'])['Financial Loss (in
↪ Million $)'].sum().unstack(fill_value=0)
grouped_df
```

Country	Social Engineering	Unpatched Software	Weak Passwords	Zero-day
Australia	3692.44	3811.46	3819.65	4079.45
Brazil	4166.16	3491.26	4272.7699999999995	3852.43
China	3836.52	2887.97	3269.9	3720.08
France	3659.89	3493.17	3067.7	4751.52
Germany	3689.08	4702.51	3606.09	3795.56
India	2911.76	4244.57	3256.32	4153.47
Japan	3879.2599999999998	3134.11	3852.43	4331.54
Russia	3773.9700000000003	3839.04	3711.48	3410.24
UK	4664.55	4139.15	4040.76	3658.53
USA	3752.91	3281.19	3982.3	3795.72

```
# Plotting the data as a bar chart
grouped_df.plot(kind='bar', figsize=(12, 8), width=0.8)

plt.title('Financial Losses by Vulnerability Types in Different Countries')
plt.xlabel('Country')
plt.ylabel('Total Financial Loss (in Million $)')
plt.xticks(rotation=0)  # Keep x-axis labels horizontal
plt.legend(ncol=6, loc="upper center", bbox_to_anchor=(0.5,-0.08))
plt.grid(axis='y')
plt.gca().yaxis.set_major_formatter(FuncFormatter(lambda x, _: f'{int(x):,}')) # Apply
↪ number formatting
plt.tight_layout()
plt.show()
```

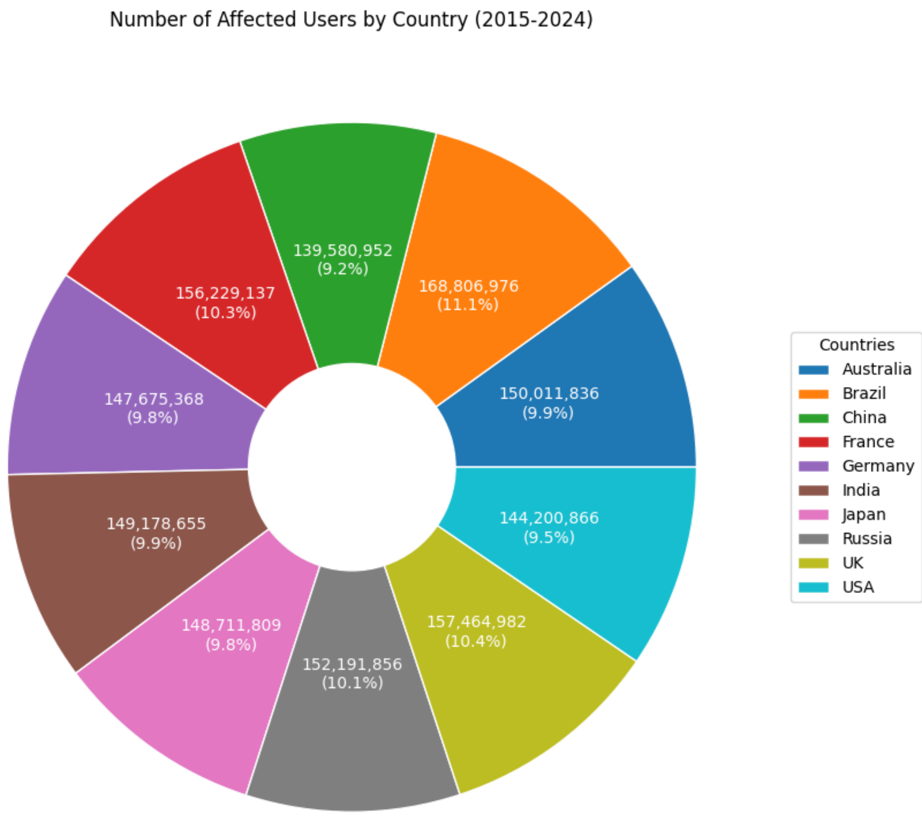


3.2 Number of Affected Users

3.2.1 Overall

```
country_df = df.groupby('Country')['Number of Affected Users'].sum()

# Plotting the pie chart
plt.figure(figsize=(10, 8))
wedges, texts, autotexts = plt.pie(country_df, autopct=lambda
↪ pct:f"{round(country_df.sum()*pct/100):,}\n({pct:.1f}%)", textprops=dict(color="w"),
↪ wedgeprops=dict(width=0.7, edgecolor='w'))
plt.legend(wedges, country_df.index, title="Countries", loc="center left",
↪ bbox_to_anchor=(1, 0, 0.5, 1))
plt.title('Number of Affected Users by Country (2015-2024)')
plt.tight_layout()
plt.show()
```



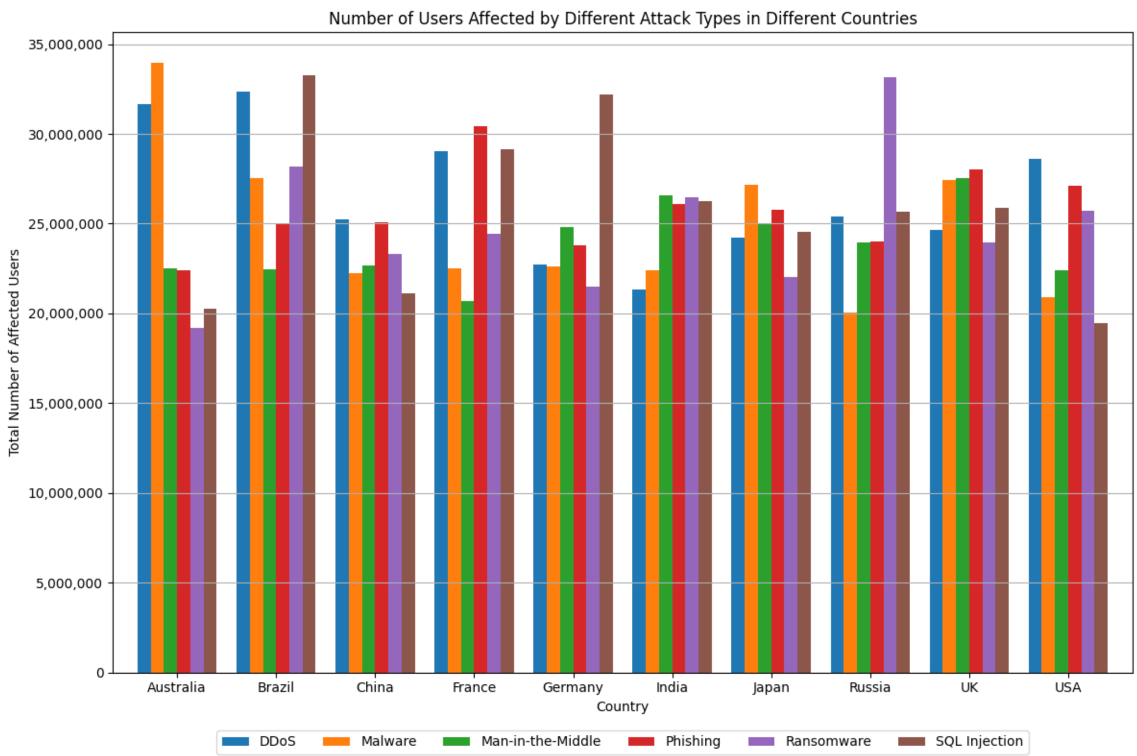
3.2.2 Breakdown by Attack Types

```
grouped_df = df.groupby(['Country', 'Attack Type'])['Number of Affected
↪ Users'].sum().unstack(fill_value=0)
grouped_df
```

Country	DDoS	Malware	Man-in-the-Middle	Phishing	Ransomware	SQL Injection
Australia	31680693	33960220	22518909	22390813	19177903	20283292
Brazil	32331598	27529028	22472326	25045910	28170653	33257465
China	25216140	22234189	22677828	25066462	23290803	21095516
France	29042198	22498584	20683824	30408495	24458280	29137761
Germany	22714784	22593199	24835152	23820513	21487419	32224291
India	21347569	22395214	26568402	26105954	26491982	26269538
Japan	24214976	27181900	25028943	25750354	22015956	24519685
Russia	25381101	20041990	23953238	24013027	33144185	25658294
UK	24662842	27415131	27565197	28015585	23944806	25861422
USA	28609364	20908958	22405704	27100862	25710920	19465062

```python
# Plotting the data as a bar chart
grouped_df.plot(kind='bar', figsize=(12, 8), width=0.8)

plt.title('Number of Users Affected by Different Attack Types in Different Countries')
plt.xlabel('Country')
plt.ylabel('Total Number of Affected Users')
plt.xticks(rotation=0)  # Keep x-axis labels horizontal
plt.legend(ncol=6, loc="upper center", bbox_to_anchor=(0.5,-0.08))
plt.grid(axis='y')
plt.gca().yaxis.set_major_formatter(FuncFormatter(lambda x, _: f'{int(x):,}')) # Apply
↪ number formatting
plt.tight_layout()
plt.show()
```



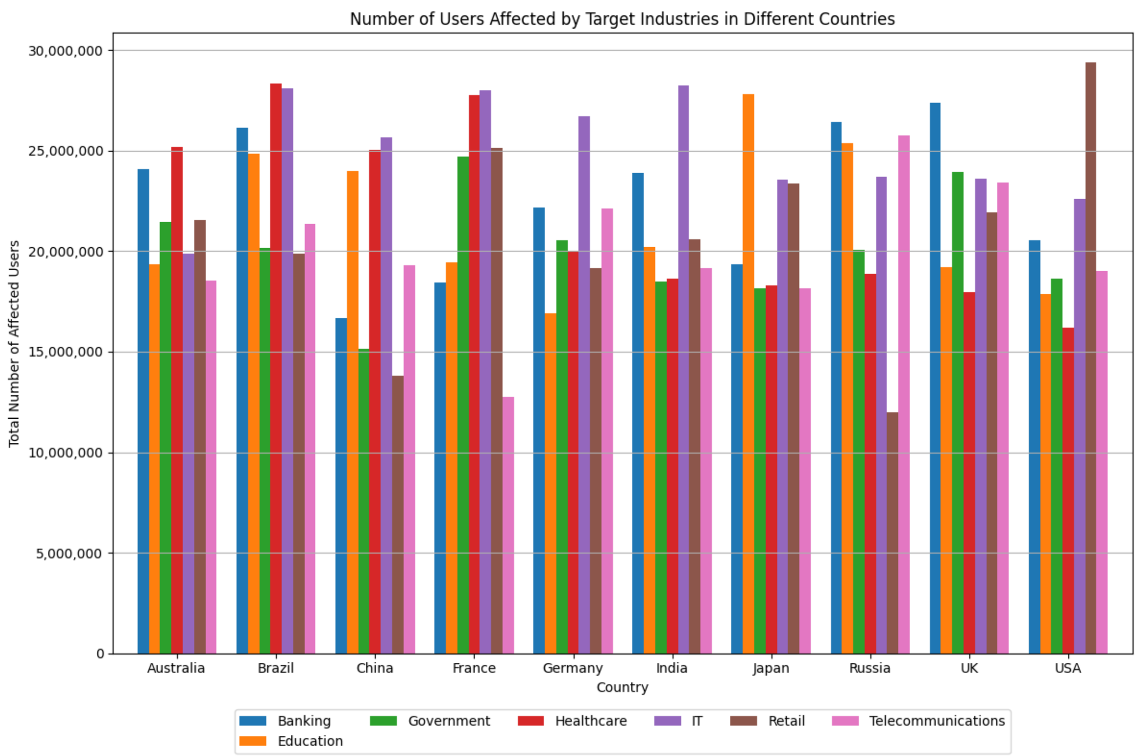
3.2.3 Breakdown by Target Industries

```
grouped_df = df.groupby(['Country', 'Target Industry'])['Number of Affected
↪ Users'].sum().unstack(fill_value=0)
grouped_df
```

Country	Banking	Education	Government	Healthcare	IT	Retail	Telecommun
Australia	24091736	19353282	21431052	25171156	19880173	21540256	18
Brazil	26155727	24832426	20148467	28345995	28113529	19873671	21
China	16655683	23978380	15119967	25036722	25654123	13816525	19
France	18432132	19438280	24701233	27765020	27984331	25145910	12
Germany	22179754	16906535	20548248	20001882	26734222	19169177	22
India	23877886	20211347	18497871	18616483	28243125	20595810	19
Japan	19328408	27833081	18163470	18311429	23573823	23358354	18
Russia	26445149	25388211	20084391	18869355	23682282	11980441	25
UK	27385750	19217857	23921622	17977995	23622708	21909685	23
USA	20546181	17845333	18622709	16175879	22606513	29386557	19

```
# Plotting the data as a bar chart
grouped_df.plot(kind='bar', figsize=(12, 8), width=0.8)

plt.title('Number of Users Affected by Target Industries in Different Countries')
plt.xlabel('Country')
plt.ylabel('Total Number of Affected Users')
plt.xticks(rotation=0)  # Keep x-axis labels horizontal
plt.legend(ncol=6, loc="upper center", bbox_to_anchor=(0.5,-0.08))
plt.grid(axis='y')
plt.gca().yaxis.set_major_formatter(FuncFormatter(lambda x, _: f'{int(x):,}')) # Apply
↪ number formatting
plt.tight_layout()
plt.show()
```



3.2.4 Breakdown by Vulnerabilities

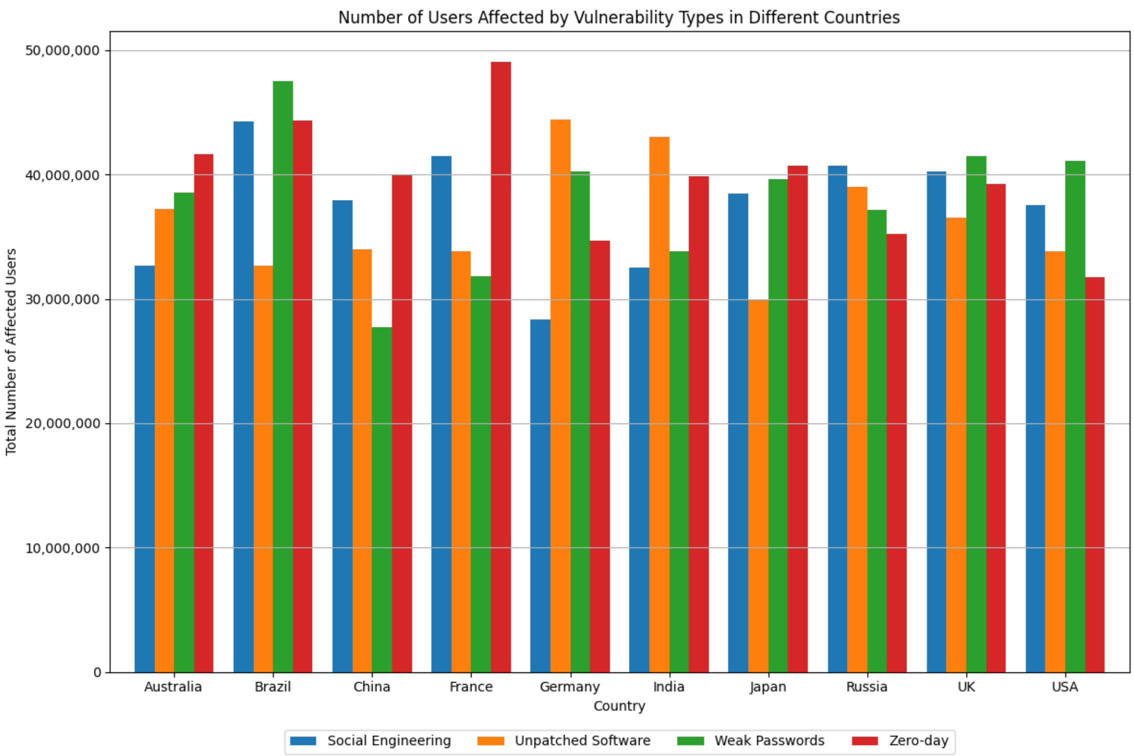
```
grouped_df = df.groupby(['Country', 'Security Vulnerability Type'])['Number of Affected
↪ Users'].sum().unstack(fill_value=0)
grouped_df
```

Country	Social Engineering	Unpatched Software	Weak Passwords	Zero-day
Australia	32655628	37239422	38511547	41605233
Brazil	44264096	32665808	47545440	44331636
China	37903166	33993101	27718498	39966173
France	41497066	33848891	31830875	49052310
Germany	28330489	44417172	40267316	34660381
India	32481881	43014921	33843651	39838206
Japan	38440588	29961591	39611830	40697805
Russia	40737980	39032497	37186290	35235068
UK	40211515	36562515	41485989	39204964
USA	37576452	33803605	41116265	31704548

```
# Plotting the data as a bar chart
grouped_df.plot(kind='bar', figsize=(12, 8), width=0.8)

plt.title('Number of Users Affected by Vulnerability Types in Different Countries')
plt.xlabel('Country')
```

```python
plt.ylabel('Total Number of Affected Users')
plt.xticks(rotation=0)  # Keep x-axis labels horizontal
plt.legend(ncol=6, loc="upper center", bbox_to_anchor=(0.5,-0.08))
plt.grid(axis='y')
plt.gca().yaxis.set_major_formatter(FuncFormatter(lambda x, _: f'{int(x):,}')) # Apply
↪  number formatting
plt.tight_layout()
plt.show()
```



4 Financial Impact Analysis

- Assess the total financial losses caused by cyberattacks per year or country.
- Analyze the correlation between attack types and financial losses.

5 Industry Analysis

- Determine which industries are most frequently targeted by cyberattacks.
- Assess the impact of attacks on different sectors, such as healthcare, finance, and education.

6 Vulnerability Analysis

- Identify common security vulnerabilities exploited in attacks.

- Analyze the effectiveness of various defense mechanisms used against attacks.

7 User Impact Analysis

- Assess how many users are affected by different attack types or in different countries.
- Explore the relationship between the number of affected users and financial losses.

8 Response Time Analysis

- Analyze the incident resolution times based on attack types or countries.
- Identify any patterns in response effectiveness.

9 Defensive Mechanism Effectiveness

- Evaluate the success rates of different defense mechanisms against various attack types.