Project 2 Report

Segurança em Redes de Comunicações

Universidade de Aveiro

Carlos Costa (88755)
Leandro Rito (92975)

2023/2024

Departamento de Eletrónica, Telecomunicações e Informática

Contents

1 - Introduction

This project's objective is to analyze the network traffic of a company that aims to implement a reliable Cybersecurity system in order to identify unusual behaviors, leading to the implementation of alert rules based on possible attacks.

To do that, we were given three files, *data0.parquet* with the data of the typical behavior of the network device, *test0.parquet* with data that may contain anomalous behaviors resulting from illicit activities within the network and *servers0.parquet* with the data of external accesses to the corporation servers from a small set of clients in the same network that may contain external users interacting with the corporation servers in an anomalous way. All three files contain one full day's worth of data.

Our first step is to understand the difference in flows between the *data0* and *test0* files by making queries to these files and creating plots to visualize this difference.

As shown by the number of flows over time in Figure 1.1, *test0* shows an increase in the number of flows.

This result leads us to analyze these datasets further and the definition of SIEM rules for the detection and alert of unusual behaviors.

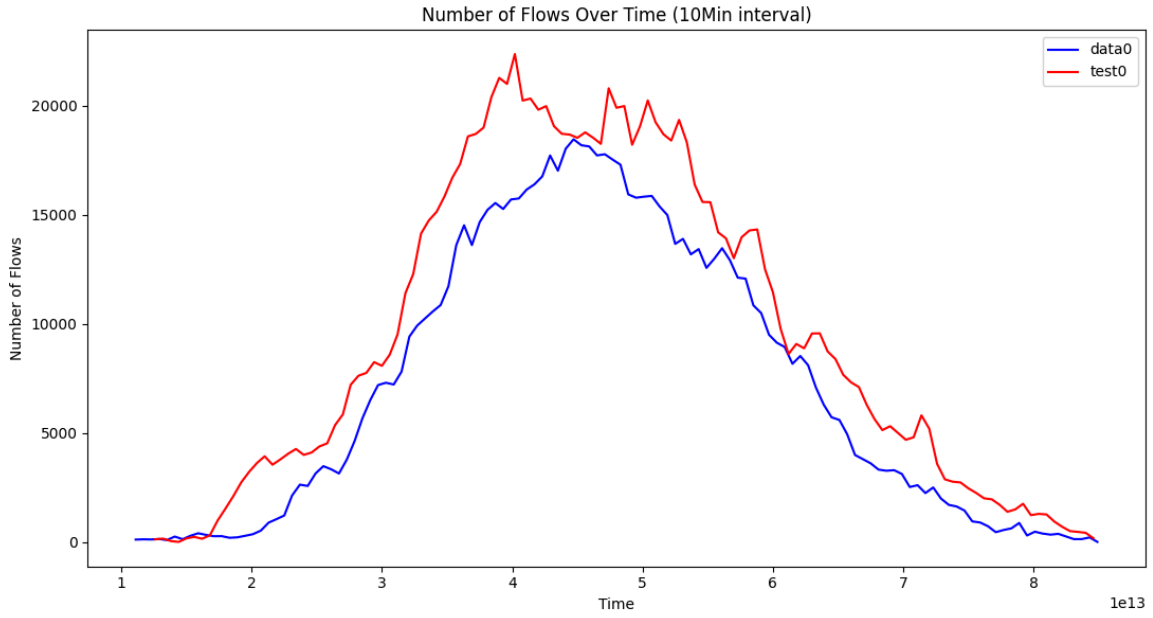

Figure 1.1: Number of Flows Over Time

2 - Normal behavior analysis

An overall analysis of the normal flow for each country depicted in Figure 2.1.

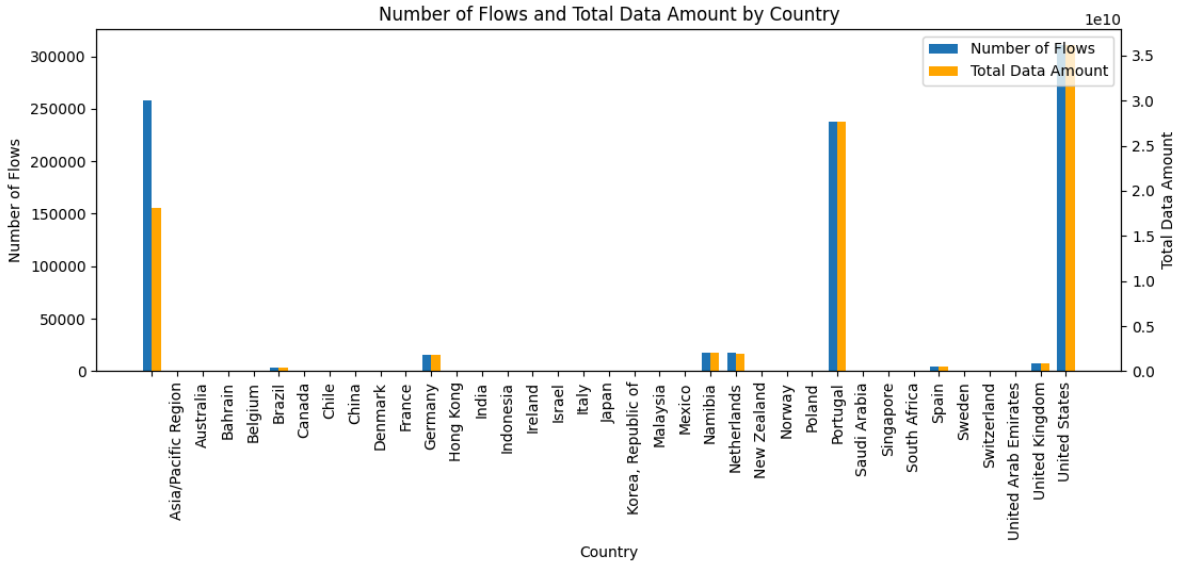

Figure 2.1: Number of Flows and Data Transferred for All Countries in *data0.parquet*

From these flows, the most significant addresses in terms of data bytes transferred are depicted in Figure 2.2 and Figure 2.3, first for up_bytes and then for down_bytes.

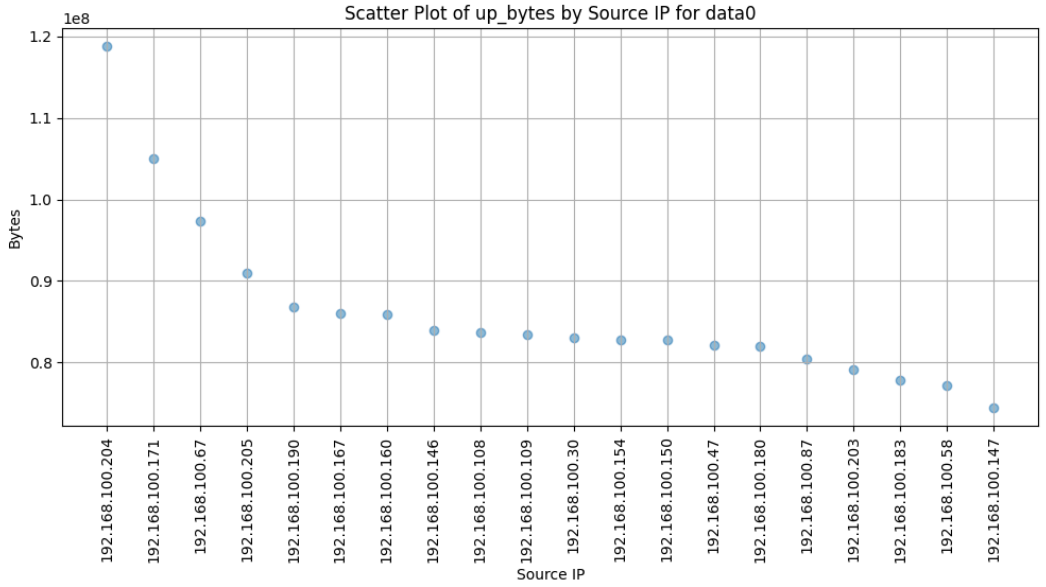

Figure 2.2: Amount of bytes, in total, sent upstream by IP address

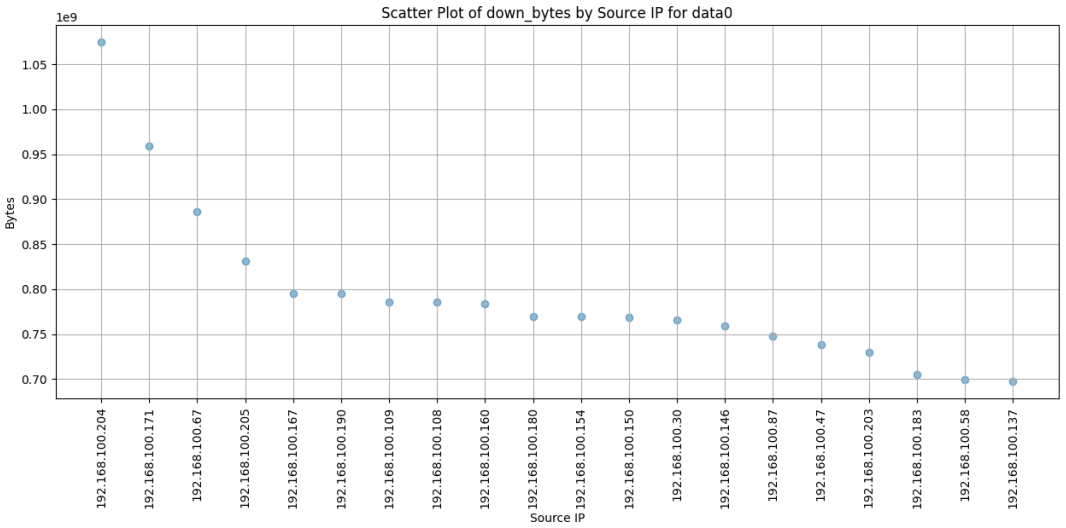

Figure 2.3: Amount of bytes, in total, sent downstream by IP address

By analyzing the flow of the machines, we concluded 192.168.100.225 and 192.168.100.234 are addresses to DNS servers because they only receive requests through UDP on port 53 and 192.168.100.239 is either a Web or a Mail server because it only receives requests through TCP on port 443.

This analysis was made by filtering all communications between private addresses and then counting the number of data flows, taking into consideration the address, the protocol and the port. The result is as seen in Table 2.1.

dst_ip	proto	port	count
192.168.100.255	udp	53	197
192.168.100.234	udp	53	197
192.168.100.239	tcp	443	197

Table 2.1: Network Data

3 - Anomalous behavior analysis

3.1 - Data exfiltration

When analyzing the flow of data upstream, we detected what might have been an issue with data exfiltration. We first calculated the amount of bytes being sent upstream in both contexts and concluded that, in the illicit scenario, the amount of data being sent upstream being sent by the addresses 192.168.100.35 and 192.168.100.57 was much higher than the rest of the other addresses.

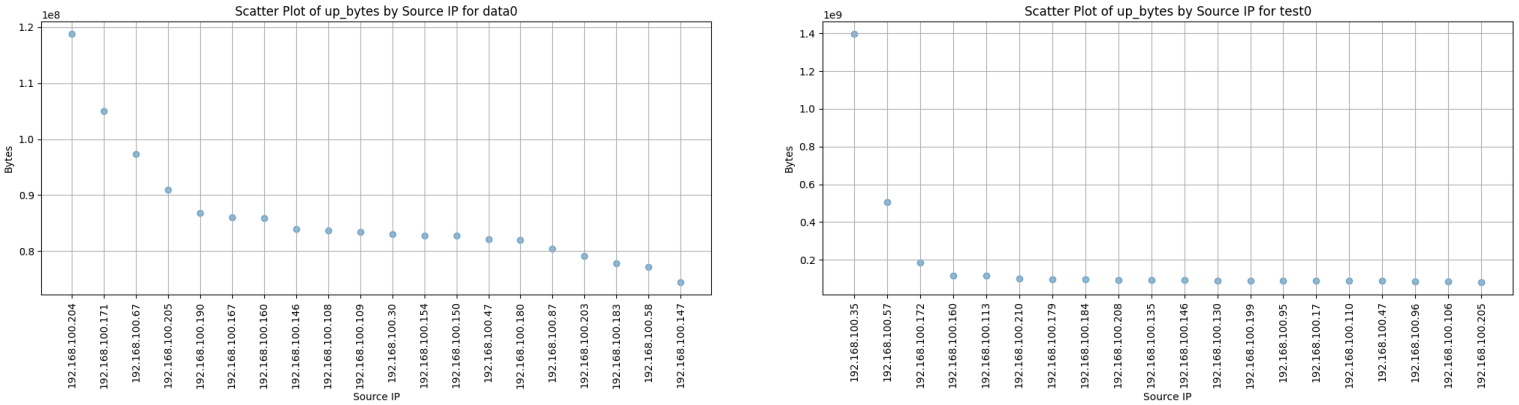

Figure 3.1: Amount of bytes, in total, sent upstream by IP addresses

It is worth noting that the Y axis for *test0* is one order of magnitude larger than in *data0.* Applying the same order of magnitude on both plots would render the plot for *data0* unreadable.

With these suspicious addresses in mind, and considering the context they're in, we proceeded to try and understand if this was a case of data exfiltration.

We took the top 2 addresses that stood out the most (192.168.100.35 and 192.168.100.67) and for both of them we checked out the destination IP address for the largest upstream transactions to understand who the target was.

For the address 192.168.100.35 (Table 3.1), the destination address was 142.250.184.196, which was traced back to Google LLC (GOGL), so it was most likely an exfiltration of the Google Drive services.

For the address 192.168.100.57 (Table 3.2), the destination address was 13.107.42.17, which was traced back to Microsoft Corporation (MSFT), so it was most likely an exfiltration of the OneDrive services.

Timestamp	src_ip	dst_ip	proto	port	up_bytes	down_bytes
3417865	192.168.100.35	142.250.184.196	tcp	443	435000698	5429053
3537833	192.168.100.35	142.250.184.196	tcp	443	129798244	2028025
3777637	192.168.100.35	142.250.184.196	tcp	443	460103308	6126627
4017827	192.168.100.35	142.250.184.196	tcp	443	162107172	1707721

Table 3.1: Some of the traffic to Google from 192.168.100.35

Timestamp	src_ip	dst_ip	proto	port	up_bytes	down_bytes
3816863	192.168.100.57	13.107.42.17	tcp	443	228004998	2623275
3936801	192.168.100.57	13.107.42.17	tcp	443	131457964	1925704
4176780	192.168.100.57	13.107.42.17	tcp	443	83764615	790940
4056877	192.168.100.57	13.107.42.17	tcp	443	52718022	803704

Table 3.2: Some of the traffic to Microsoft from 192.168.100.57

3.2 - Unusual communications

The data from *test0* gave us an insight on new countries to where data was being sent, apart from the ones registered in *data0*. We grouped the number of flows and amount of data transferred in total for each new country and obtained the result depicted in Figure 3.2.

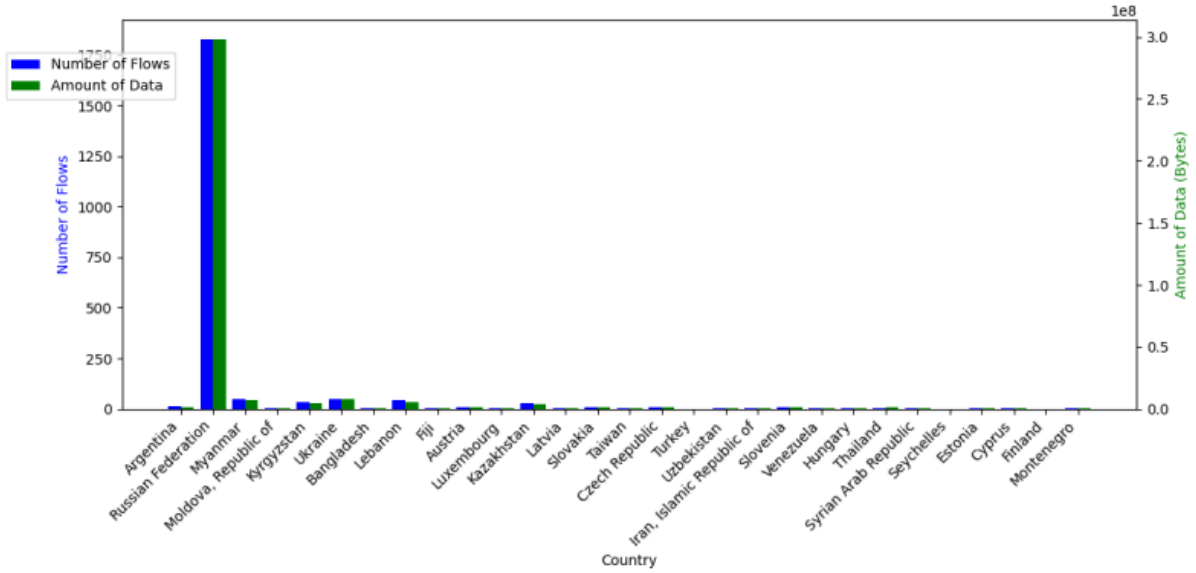

Figure 3.2: Number of Flows and Data Transferred for New Countries

A quick analysis tells us that the country with most flows and amount of data transferred was by far the Russian Federation (1830 flows and 298 476 091 bytes transferred). The whole data for all the suspicious countries is presented in Table 3.3.

Because of the distinctive behavior portrayed by the Russian Federation, we gathered those communications and concluded that they are all TCP in port 443.

country	data_flows	data_bytes
Argentina	13	1181151
Russian Federation	1830	298476091
Myanmar	51	7160098
Republic of Moldova	3	726360
Kyrgyzstan	33	4385761
Ukraine	51	8267783
Bangladesh	6	539513
Lebanon	43	5706331
Fiji	5	831779
Austria	11	1643291
Luxembourg	5	592320
Kazakhstan	27	4068161
Latvia	2	477958
Slovakia	7	1420211
Taiwan	6	680143
Czech Republic	9	1119327
Turkey	1	164834
Uzbekistan	2	253613
Islamic Republic of Iran	7	704008
Venezuela	2	322619
Hungary	5	784137
Thailand	5	1200336
Syrian Arab Republic	2	445128
Seychelles	1	161857

Estonia	3	284001
Cyprus	5	678980
Finland	1	70122
Montenegro	2	207799

Table 3.3: New Countries Data

3.3 - Botnets

Botnets are characterized by devices that become puppets and perform undesired actions behind the curtains. One of the issues can be privilege escalation within a network or DDoS attacks, and the communication between private addresses is expected.

To detect this, we analyzed the communication between local machines on *test0*, like we did for *data0* to identify regular services like DNS and Web servers.

This time, apart from the 3 services, we found 3 other communication flows (Table 3.5) that, because they weren't in the regular communications data file, we can assume that they are not regular services from the corporate but, in fact, in the context of this specific analysis, they could be botnets.

To gain further confidence, we calculated the ratio between upload and download on the flows involving the suspicious addresses, and they all presented a ratio close to 1, as seen in Table 3.6.

Because the communication between the infected devices is usually the sharing of information and instructions or coordinating attacks, the traffic in both directions tends to be balanced, hence the ratio of 1.

dst_ip	protocol	port	count
192.168.100.148	tcp	443	2
192.168.100.225	udp	53	200
192.168.100.234	udp	53	200
192.168.100.239	tcp	443	200
192.168.100.46	tcp	443	2
192.168.100.97	tcp	443	2

Table 3.5: Possible Botnets

src_ip	dst_ip	up_bytes	down_bytes	ratio	diff
192.168.100.97	192.168.100.46	378527	379879	0.996441	0.003559
192.168.100.148	192.168.100.46	127103	126547	1.004394	0.004394
192.168.100.148	192.168.100.97	149972	147130	1.019316	0.019316
192.168.100.46	192.168.100.97	74514	76827	0.969893	0.030107
192.168.100.97	192.168.100.148	416517	404196	1.030483	0.030483
192.168.100.46	192.168.100.148	91994	98527	0.933693	0.066307

Table 3.6: Botnet Data Transfer Statistics

3.4 - Command & Control

Command & Control attacks can sometimes be disguised with DNS encapsulation, the so-called DNS Tunneling. Knowing the DNS servers of the corporate (Table 2.1), we gathered the flows that targeted their addresses, and retrieved the source addresses with more requests, as depicted in Figure 3.3.

The addresses 192.168.100.208, 192.168.100.61 and 192.168.100.64 clearly stood out from the rest, with an abnormal amount of requests to a DNS server, which right away puts them under suspicion.

Considering the context of the analysis, it is fair to assume that these were targets of C&C attacks.

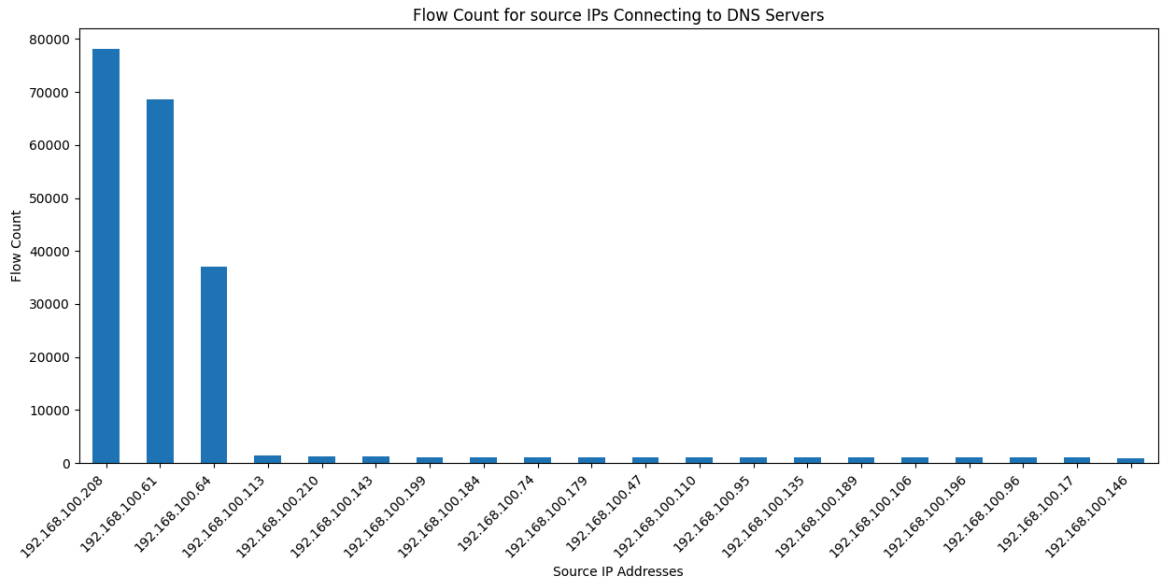

Figure 3.3: Number of Flows connecting to DNS Servers

3.5 - External client behaviors

As mentioned before, the server0 file contains a record of the communications from external clients to the corporation servers that may be interacting in an anomalous way.

In order to identify anomalous behavior, we started by sorting the clients according to their ratio of uploads and downloads and selected the 20 which had the bigger ratio, making these the most suspicious addresses.

ip address	ratio(up/down)
82.155.120.34	0.120154
82.155.120.210	0.119952
82.155.120.86	0.119238
82.155.120.99	0.119091
82.155.120.134	0.118925
82.155.120.155	0.118899
82.155.120.180	0.118843
82.155.120.159	0.118709
82.155.120.167	0.118670
82.155.120.52	0.118648
82.155.120.59	0.118647
82.155.120.55	0.118611
82.155.120.168	0.118595
82.155.120.122	0.118502
82.155.120.181	0.118487
82.155.120.90	0.118437
82.155.120.158	0.118426
82.155.120.137	0.118418
82.155.120.96	0.118391
82.155.120.100	0.118376

Table 3.7: External Clients Upload and Download Ratio

Then by analyzing their ratio every 30 minutes in a time frame of 10 hours we concluded that address 82.155.120.34 was an example of the expected behavior of a normal client since it presented a more spaced out access to the servers.

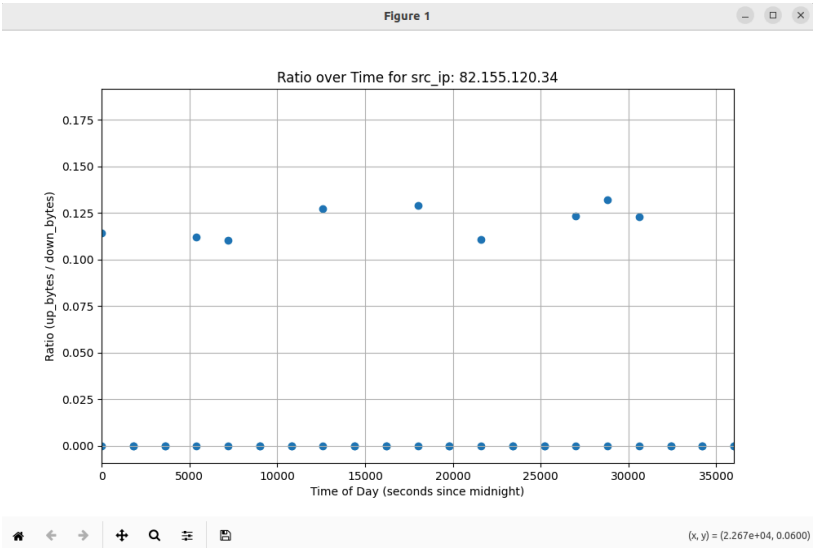

Figure 3.4: Ratio for ip address 82.155.120.34

On the other hand, the results for addresses like 82.155.120.52 and 82.155.120.168 showed a continuous ratio, which meant that there was always a continuous flow of requests trying to access the servers, leading to the conclusion that these could be malicious clients.

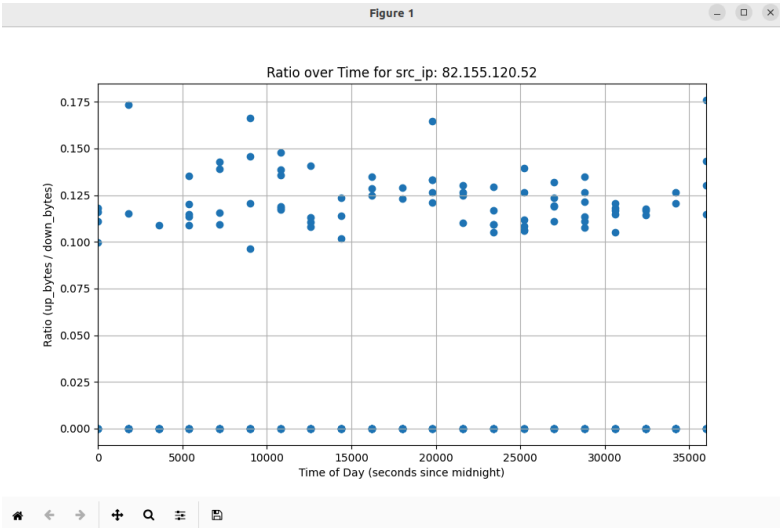

Figure 3.6: Ratio for ip address 82.155.120.52

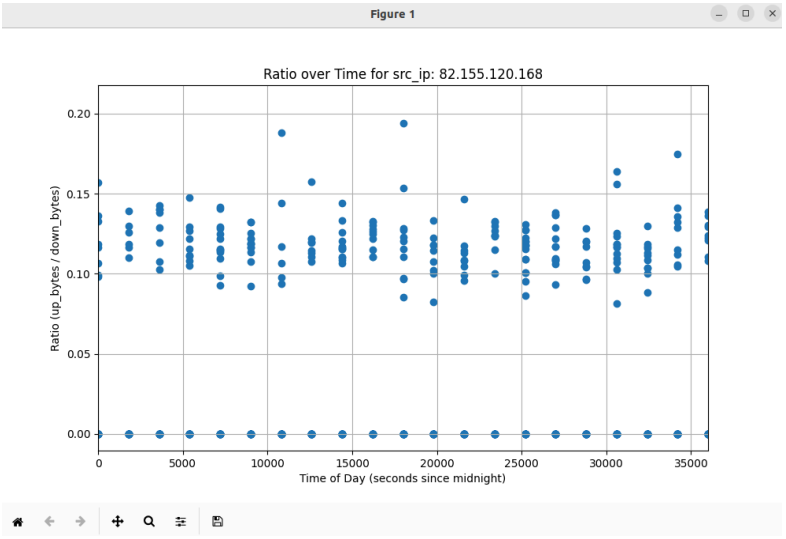

Figure 3.7: Ratio for ip address 82.155.120.52

4 - SIEM Rules Definition

4.1 - Unusual Traffic on Port 443

This rule focuses on traffic on port 443, which is commonly used for secure web communication. By monitoring the amount of outgoing traffic (down_bytes) on this port, it is possible to identify cases where the volume exceeds normal patterns, potentially indicating exfiltration attempts.

4.1.1 - Conditions

- Protocol is TCP or UDP
- Destination port is 443 (HTTPS)
- Downstream bytes (outgoing traffic) is significantly higher than the average of port 443

4.1.2 - Actions

- Generate an alert for potential exfiltration activity

```
# Get the file names from the CLI
f1 = sys.argv[1]
f2 = sys.argv[2]

# Read data and test datasets
df1 = pd.read_parquet(f1)
df2 = pd.read_parquet(f2)

# Group the DataFrame by src_ip and dst_ip and calculate the sum of up_bytes and down_bytes
grouped_df = df2.groupby(['src_ip', 'dst_ip']).agg({'up_bytes': 'sum', 'down_bytes': 'sum'})

# Calculate the total sum of bytes for each pair of src_ip and dst_ip
grouped_df['total_bytes'] = grouped_df['up_bytes'] + grouped_df['down_bytes']

# Sort the DataFrame by the total sum of bytes in descending order
grouped_df = grouped_df.sort_values('up_bytes', ascending=False)

mean_data = grouped_df['up_bytes'].mean()

print(mean_data)

# Display the resulting DataFrame
print(grouped_df.head(50))
```

4.2 - Communication with New Countries

This rule focuses on identifying communications with IP addresses located in countries that have not been previously communicated with. By maintaining a database or list of known IP ranges for each country, it is possible to compare the destination IP address of network traffic with those ranges to determine if it falls within the IP range of a previously uncommunicated country.

4.2.1 - Conditions

- Protocol is TCP or UDP
- Destination IP is outside the known IP ranges of previously communicated countries.

4.2.2 - Actions

- Generate an alert for potential communication with new countries

```
# Get the file names from the CLI
f1 = sys.argv[1]
f2 = sys.argv[2]

# Read data and test datasets
df1 = pd.read_parquet(f1)
df2 = pd.read_parquet(f2)

# Add the 'country' column based on 'dst_ip' for df1
df1['country'] = df1['dst_ip'].apply(get_country)

# Get a list of unique countries from df1
unique_countries = df1['country'].unique()

# Add the 'country' column based on 'dst_ip' for df2
df2['country'] = df2['dst_ip'].apply(get_country)

# Find flows in df2 with new countries
new_flows = df2[~df2['country'].isin(unique_countries)]
```

4.3 - Command and Control (C&C) Communications with Abnormal DNS Requests

This rule monitors the number of DNS requests made from your network within a specified time window. A high volume of DNS requests can be an indicator of C&C communication, as botnets and other malicious infrastructures often use DNS as a covert communication channel

4.3.1 - Conditions

- Protocol is TCP or UDP
- Abnormally high number of DNS requests within a specific time window

4.3.2 - Actions

- Generate an alert for potential Command and Control (C&C) communication with abnormal DNS requests

```
# Get the file names from the CLI
f1 = sys.argv[1]
f2 = sys.argv[2]

# Read data and test datasets
df1 = pd.read_parquet(f1)
df2 = pd.read_parquet(f2)

# Define the DNS servers
dns_servers = ["192.168.100.225", "192.168.100.234"]

# Filter the flows where src_ip connects to the DNS servers
dns_flows = df2[df2["dst_ip"].isin(dns_servers)]

# Count the number of flows for each src_ip
flow_counts = dns_flows["src_ip"].value_counts()

# Print the flow counts
print(flow_counts)

top_flow_counts = flow_counts.head(20)
```

4.4 - Botnet Detection with Unusual Communication with Internal Machines

This rule detects potential botnet activity by analyzing suspicious network behavior which can be an unusually high number of connections from a single source IP within a time window or unusual traffic patterns, such as sudden surges in volume or packet counts. Botnets exhibit distinct behavior that deviates from normal traffic. By monitoring these behaviors, it is possible to detect botnet activity.

4.4.1 - Conditions

- Protocol is TCP or UDP
- High number of connections to multiple internal machines within a specified time window

4.4.2 - Actions

- Generate an alert for potential botnet communication with unusual communications with internal machines

```
# Get the file names from the CLI
f1 = sys.argv[1]
f2 = sys.argv[2]

# Read data and test datasets
df1 = pd.read_parquet(f1)
df2 = pd.read_parquet(f2)

# Filter rows with private source and destination IP addresses
private_ips = df2[(df2['src_ip'].apply(lambda ip: ipaddress.ip_address(ip).is_private))
                  & (df2['dst_ip'].apply(lambda ip: ipaddress.ip_address(ip).is_private))]

# Get unique pairs of private source and destination IP addresses
unique_pairs = private_ips[['src_ip', 'dst_ip', 'proto', 'port']].drop_duplicates()

unique_pairs = unique_pairs.groupby(['dst_ip', 'proto', 'port']).size().reset_index(name='count')

# Print the results
print(unique_pairs)
```

4.5 - Suspicious Activity of External Users Using the Corporate Public Services in an Anomalous Way

This rule detects any type of suspicious activity coming from external users using the corporate public services. A constant flow of upstreams and downstreams over same periods of time can indicate odd behaviors and potential users with bad intentions.

4.5.1 - Conditions

- Protocol is TCP or UDP
- User has an external IP
- Constant number of flows over the same periods of time

4.5.2 - Actions

- Generate an alert of anomalous behavior coming from an external IP address using corporate public services

```
datafile='./servers0.parquet'

data=pd.read_parquet(datafile)

print(data.head(20))

a1=data.loc[((data['port']==443))].groupby(['src_ip'])['up_bytes'].sum()
a2=data.loc[((data['port']==443))].groupby(['src_ip'])['down_bytes'].sum()

ratios = a1 / a2.replace(0, pd.NA)
a3 = pd.DataFrame(ratios, columns=['ratio'])

upS=a3.groupby(['src_ip'])['ratio'].sum().sort_values(ascending=False)

first20 = upS.head(20)

# Get top 20 src_ip addresses
top20_src_ips = first20.head(20).index


filtered_data = data[data['src_ip'].isin(top20_src_ips)]
print("Filtered data based on top 20 src_ip values:")
print(filtered_data)
```

5 - Conclusion

The project has demonstrated the critical importance of robust SIEM rules in monitoring a network and identifying potential threats. By analyzing a day's worth of typical network traffic, we gained valuable insights into the normal network usage and used it to identify deviations that could indicate potential threats.

In testing these SIEM rules against the *test0.parquet* and the *server0.parquet* datasets, we found that our rules were effective in highlighting anomalous network behaviors, confirming the value of a data-driven approach. Despite the complexity of the task, the use of pandas for data analysis, the use of databases for geo-localization based on IPv4 addresses helped the creation of the SIEM rules.

Future work should aim to refine the rules based on ongoing analysis of the network. Furthermore, integrating machine learning algorithms could potentially enhance the efficiency and effectiveness of detecting anomalous behaviors