**University of Aveiro**

**Master’s in Cybersecurity**

**Security in Communications Networks**

**School Year 2024/25**

**Second Project**

**Made in: 2025/06/02**

**Carlos Ferreira 108822**

**Tomás Bogalho 124224**

**Index**

[1. Introduction iii](#__RefHeading___Toc2034_1304377139)

[2. Analysis of Non-Anomalous Behavior iv](#__RefHeading___Toc1015_543899845)

[2.1. Identify Private Networks iv](#__RefHeading___Toc1017_543899845)

[2.2. Identify Internal Services/Servers iv](#__RefHeading___Toc1019_543899845)

[2.3. Describe and Quantify Internal Traffic Exchanges vi](#__RefHeading___Toc1048_3023196523)

[2.4. Describe and Quantify External Traffic Exchanges ix](#__RefHeading___Toc1023_543899845)

[3. SIEM Rules and Identification of Devices xii](#__RefHeading___Toc1025_543899845)

[3.1. Data Exfiltration using HTTPS/DNS and C&C Activities xii](#__RefHeading___Toc1934_543899845)

[3.2. BotNet Activities xiv](#__RefHeading___Toc1050_3023196523)

[3.3. Anomalous External Destinations xvii](#__RefHeading___Toc1033_543899845)

[3.4. External Users using the corporate public services in an anomalous way xx](#__RefHeading___Toc1052_3023196523)

[xx](#__RefHeading___Toc1932_543899845)

[4. Conclusion xxii](#__RefHeading___Toc1948_3023196523)

# Introduction

This report presents the configuration and validation of a rule-based SIEM detection system for identifying anomalous network behavior and compromised internal devices. The primary objective of this project is to define, implement, and evaluate detection rules capable of identifying security threats such as internal botnet activity, data exfiltration, remote command and control (C&C) communication, and anomalous external interactions with public corporate servers.

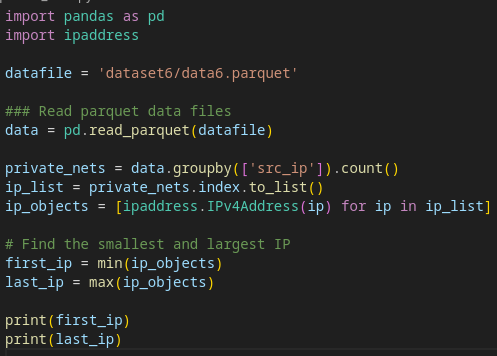
The analysis was conducted using Python on a dataset of network flow logs exported in Parquet format, simulating real-world enterprise environments. Data from normal and potentially compromised traffic were analyzed to establish baselines and detect deviations.

The network behavior was profiled based on source and destination IPs, protocols, traffic volume, and geolocation, enabling the formulation of SIEM rules grounded in empirical evidence. These rules were tested against known-safe and mixed-behavior datasets to validate their effectiveness in identifying malicious patterns without generating excessive false positives.

# Analysis of Non-Anomalous Behavior

## Identify Private Networks

In order to identify the internal private networks used by the corporate infrastructure, a script was developed to analyze the source IP addresses from the network flow logs in the data10.parquet file. The dataset contains non-anomalous traffic from a full day of operation, serving as a baseline for typical network behavior.

Figure 1: get\_private\_nets Script

Using Python and the ipaddress library, the script extracts and processes all unique source IP addresses, which correspond to internal devices. These IPs are then converted to IPv4Address objects to determine their numeric range and identify the lowest and highest addresses in use.

The smallest and largest source IPs found in the dataset were: **192.168.110.11** and **192.168.110.210** respectively.

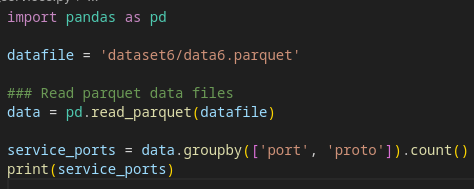
## Identify Internal Services/Servers

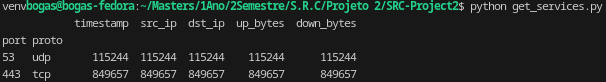
Figure 2: Result of Script get\_private\_nets

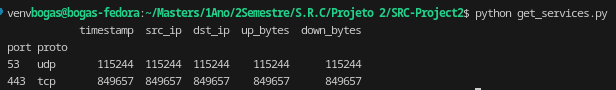
Figure 3: Result of get\_private\_nets Script

To identify the internal servers and services operating within the private corporate network, two complementary analyses were performed based on the traffic flows present in the data and servers datasets.

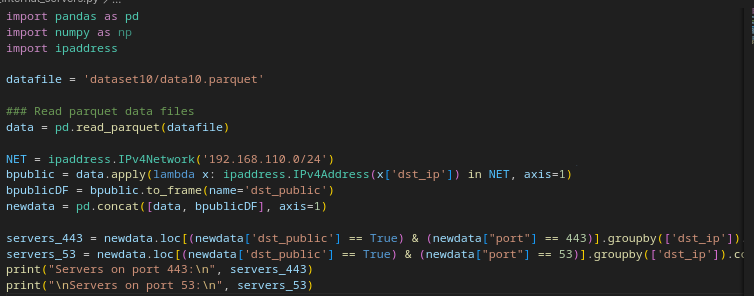
First, using the data file which includes internal traffic only a grouping operation by destination port and transport protocol was executed. This allowed for the identification of commonly accessed ports and services used internally, such as web port 54 using UDP and port 443 using TCP.

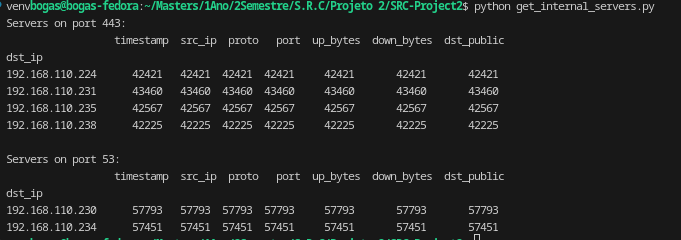
Figure 3: get\_services.py



Figure 4: Result of Script get\_services.py

Second, we filtered connections with destination IPs inside the private network (192.168.110.0/24) to identify actual internal servers. Each flow was labeled as internal or not based on whether the destination IP belonged to this subnet. We then focused on ports 443 and 53 (found in the previous script) to find devices frequently receiving internal traffic over HTTPS and DNS. These IPs are likely internal servers responding to internal client requests. Specifically, IPs with a high number of flows to port 443 were interpreted as HTTPS servers, while those on port 53 were considered DNS servers.

Figure 5: Script get\_internal\_servers.py

Figure 6: Result of get\_internal\_servers.py Script

The results show which internal IP addresses are likely acting as servers based on the number of connections they received on specific ports.

For port 443 (HTTPS), four internal IPs **192.168.110.224, .231, .235, and .238** received a large number of connections from other internal devices. This indicates these are likely internal HTTPS servers.

For port 53 (DNS), two IPs **192.168.110.230 and .234** had a high number of incoming flows, meaning they are probably internal DNS servers.

## Describe and Quantify Internal Traffic Exchanges

To analyze internal network behavior and characterize data flows between internal devices and both internal and external servers, the dataset data was used.

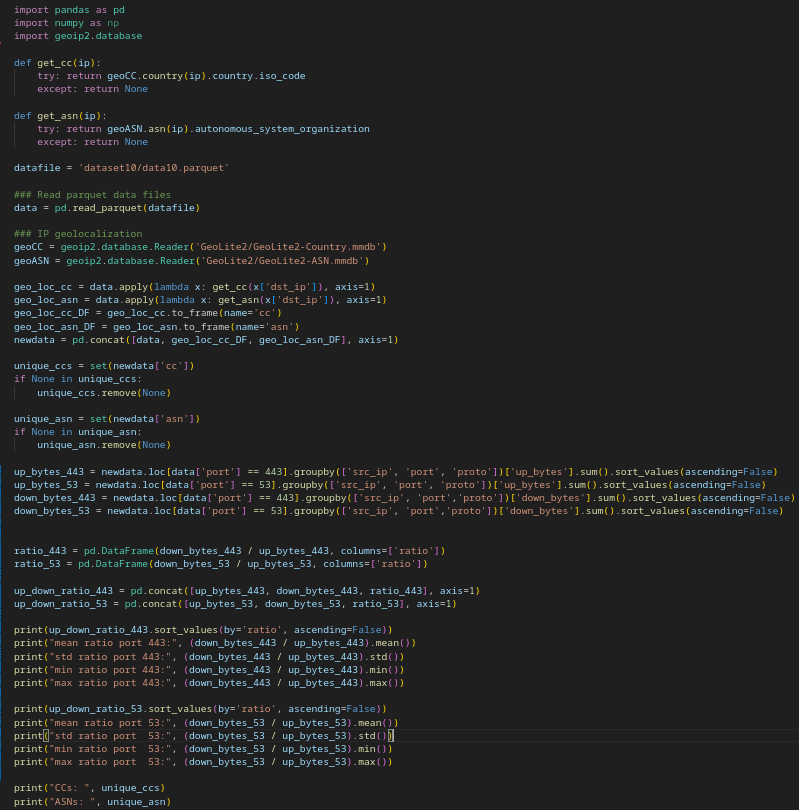
Each internal IP (source address) was evaluated in terms of total uploaded and downloaded bytes by port and protocol. By grouping traffic by source IP, port and protocol and summing the up\_bytes fields and down\_bytes, the volume and directionality of data exchanges were quantified. Additionally, the download-to-upload ratio was calculated to assess usage behavior, distinguishing endpoints with typical client-like activity (high download) from potential servers or upload-heavy devices.

To identify interactions with external services, each destination IP address was geolocated using the GeoLite2 Country and ASN databases. The results revealed a set of distinct destination countries that internal devices communicated with, providing insights into the geographical spread of external dependencies.

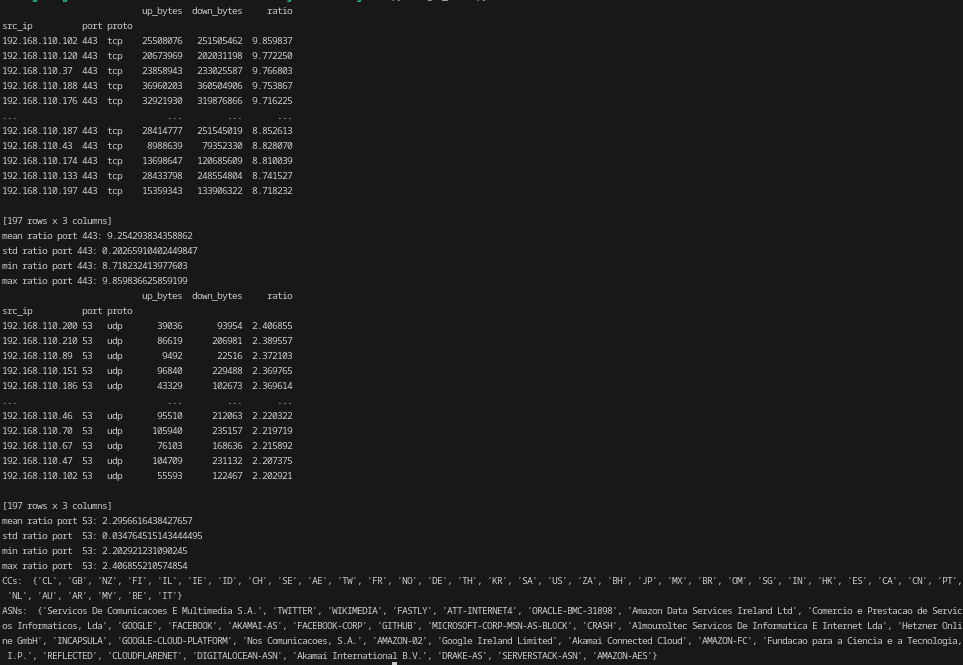
The calculated metrics included:

* Total **upload** and **download** volume per internal IP
* **Download-to-upload ratio** per internal IP
* **Mean** and **standard deviation** of download/upload ratios across all devices
* Maximum and minimum values

Figure 8: get\_stats Script

Figure 7: Script get\_stats.py

The following table presents the statistic values of the download/upload ratio for the services available in the ports 443 and 53.

Figure 9: Result of Script get\_stats-py

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **Mean Ratio** | **Standard Deviation Ratio** | **Minimum Ratio** | **Maximum Ratio** |
| **443** | 9.25 | 0.20 | 8.71 | 9.85 |
| **53** | 2.29 | 0.034 | 2.20 | 2.40 |

The observed destination countries included:

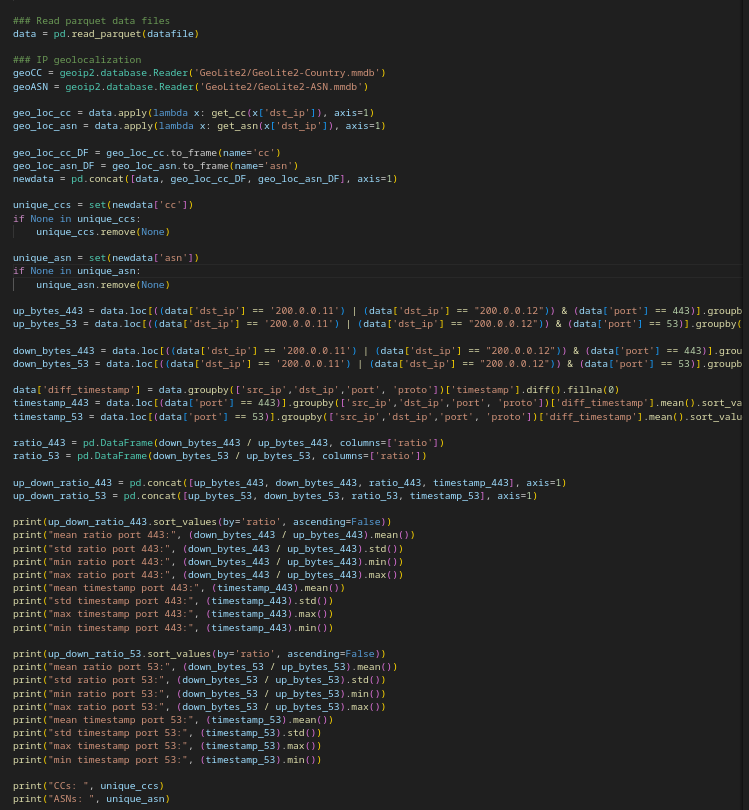
* North America: **US, CA, MX**
* Europe: **DE, FR, NL, GB, PL, SE, CH, NO, IT, BE, PT, IE**
* Asia-Pacific: **JP, CN, IN, SG, KR, HK, TW, MY, ID, AU**
* Middle East: **AE, SA, OM, BH, IL**
* Africa: **ZA**
* South America: **BR, CL**

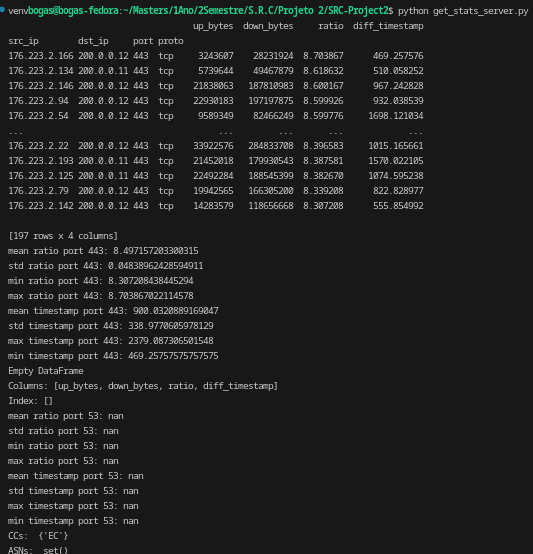
Notable ASNs included:

* Cloud Providers: GOOGLE, AMAZON-02, GOOGLE-CLOUD-PLATFORM, ORACLE-BMC-31898, MICROSOFT-CORP-MSN-AS-BLOCK, DIGITALOCEAN-ASN, SERVERSTACK-ASN
* CDNs: CLOUDFLARENET, AKAMAI-AS, Akamai Connected Cloud, INCAPSULA, FASTLY
* Content Platforms: FACEBOOK, TWITTER, WIKIMEDIA, GITHUB
* National/ISP Providers: Nos Comunicacoes, S.A., Fundacao para a Ciencia e a Tecnologia, I.P., Servicos De Comunicacoes E Multimedia S.A.
* Others: REFLECTED, ATT-INTERNET4, Hetzner Online GmbH, DRAKE-AS, Almouroltec Servicos De Informatica E Internet Lda

## Describe and Quantify External Traffic Exchanges

To understand how external clients interact with the corporate network, the servers10.parquet dataset was analyzed. This dataset contains traffic flows from external IPs to public-facing corporate servers located in the 200.0.0.0/24 network specifically 200.0.0.11 and 200.0.0.12 since they are the servers addresses.

Figure 10: Script get\_stats\_server.py

Figure 11: Result of Script get\_stats\_server.py

For each external IP and port, the total uploaded and downloaded ratio and timestamp was calculated. The download-to-upload ratio was computed to assess the nature of each connection with higher ratios indicating data retrieval behavior, which is typical for external clients accessing web or API services.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Port | Mean Ratio | Standard Deviation Ratio | Minimum Ratio | Maximum Ratio |
| 443 | 8.49 | 0.04 | 8.30 | 8.70 |
| 53 | - | - | - | - |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Port | Mean Timestamp Diff. | Standard Timestamp Diff. | Minimum Timestamp Diff. | Maximum Timestamp Diff. |
| 443 | 900.03 | 338.97 | 2379.02 | 469.25 |
| 53 | - | - | - | - |

These values indicate moderate variation, with some clients showing steady access patterns, while others exhibit more sporadic or burst-like behavior. Although there no data using the port 53 was found.

# SIEM Rules and Identification of Devices

## Data Exfiltration using HTTPS/DNS and C&C Activities

To detect potential internal botnet infections, data exfiltration via DNS, and command and control (C&C) activities, SIEM rules were formulated based on deviations from baseline traffic patterns established using the clean dataset data.

For the normal internal traffic, the following distribution of HTTPS-to-DNS connection ratios was observed across 197 internal IPs:

* Mean ratio: 7.41
* Standard deviation: 0.61
* Minimum: 5.39
* Maximum: 9.63

This profile reflects typical enterprise behavior, where HTTPS traffic dominates due to web access and secure services, with relatively fewer DNS queries.

Using the test dataset the same HTTPS/DNS ratio was calculated, with the following results:

* Mean ratio: 7.35
* Standard deviation: 1.26
* Minimum: 0.077
* Maximum: 10.35

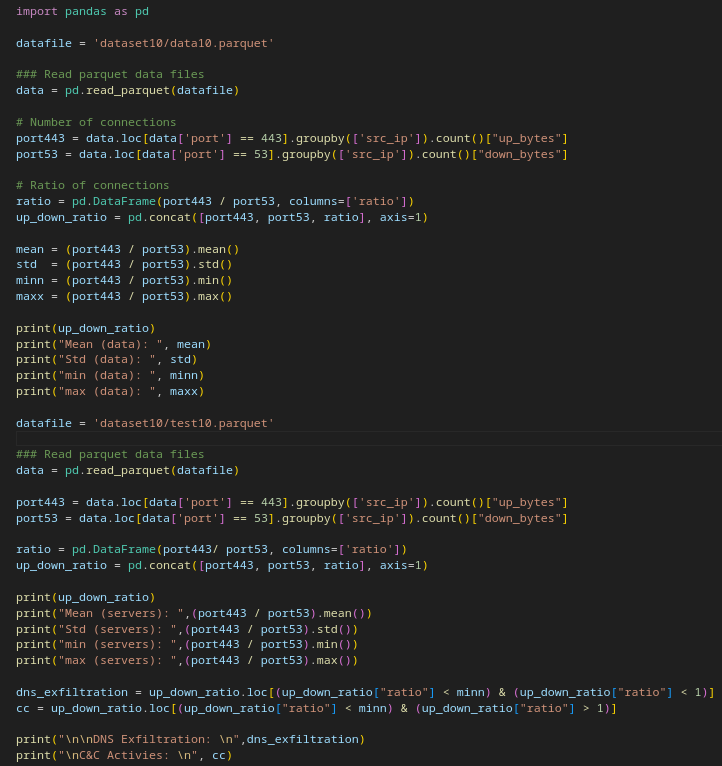
By comparing the values we could conclude that the standard deviation and the minimum were the values that most diverge from the clean results. When analyzing the ratio, a low value means that a lot more data was consumed/obtained by external devices then by the servers, which **could mean data extrafiltration**, w**hen the difference isn’t as high it could still represent C&C activities**. These values are alarming since the client consumes more than what it sends.

Any internal host with a significantly lower ratio than the baseline minimum was flagged for suspicious behavior. Such a pattern may indicate:

* Control activity using frequent DNS queries
* DNS tunneling or data exfiltration (e.g., sensitive data hidden in DNS payloads)
* Suppressed web activity while maintaining background communications

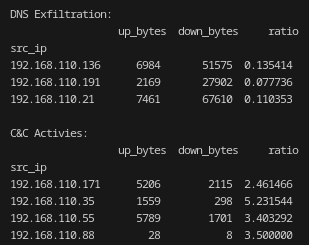
So the resulting rule was:

* **Any value below the minimum obtained from the clean dataset is considered anomalous**, then when the ratio **is bigger then 1 we considered it as DNS Exfiltration** if it is **lower then 1 then we considered it C&C Activites**

Figure 12: Script http\_dns\_exfiltration

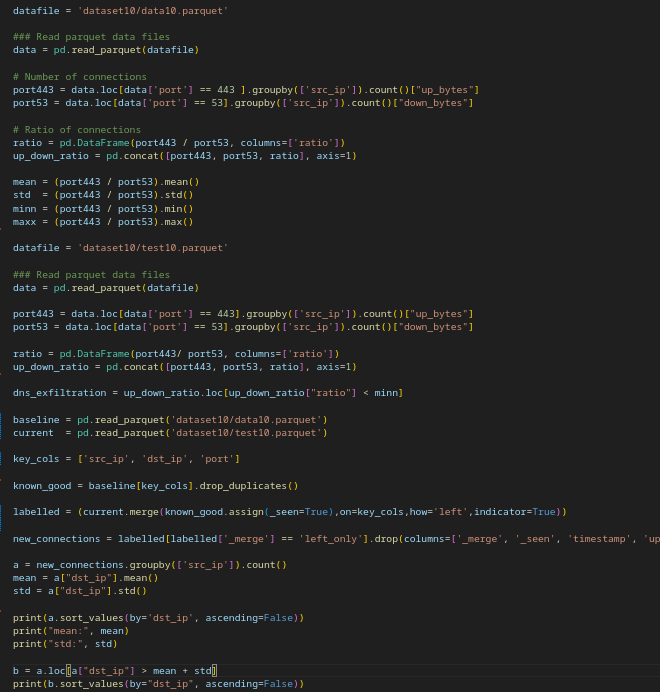
When applied, the rule found the following addresses:

* For **DNS Exfiltration – 192.168.110.136, 192.168.110.191 and 192.168.110.21**
* For **C&C Activities – 192.168.110.171, 192.168.35, 192.168.110.55 and 192.168.110.88**

Figure 13: Result of the Script http\_dns\_exfiltration

## BotNet Activities

The group identified new connections in the test data by comparing them to the clean dataset. We flagged source IPs that connected to previously unseen (src\_ip, dst\_ip, port) combinations, especially those with a count significantly above the average plus standard deviation. These may represent compromised devices establishing new internal communications with internal computers, which is a normal behavior of a botnet.

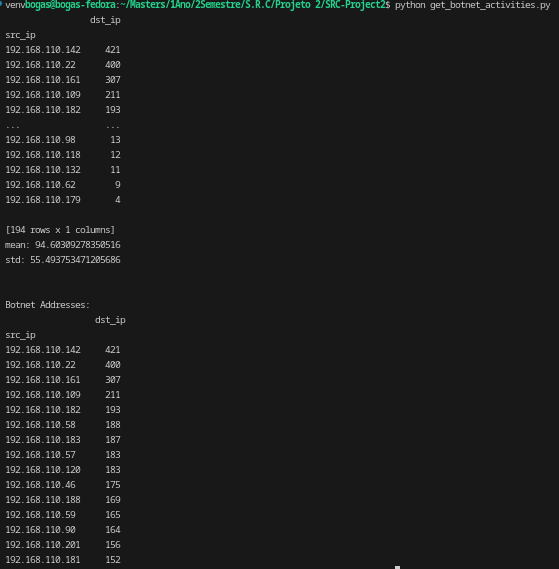
Figure 14: Script get\_botnet\_activities

The resulting rule was:

* **New connections where the number of connections established is above the the mean plus the standard deviation**.

The internal IP addresses identified as part of the botnet due to a high number of new and anomalous connections are:

* **192.168.110.142, 192.168.110.22, 192.168.110.161, 192.168.110.109, 192.168.110.182, 192.168.110.58, 192.168.110.183, 192.168.110.57, 192.168.110.120, 192.168.110.46, 192.168.110.188, 192.168.110.59, 192.168.110.90, 192.168.110.201, and 192.168.110.181.**

Figure 15: Result of Script get\_botnet\_activities.py

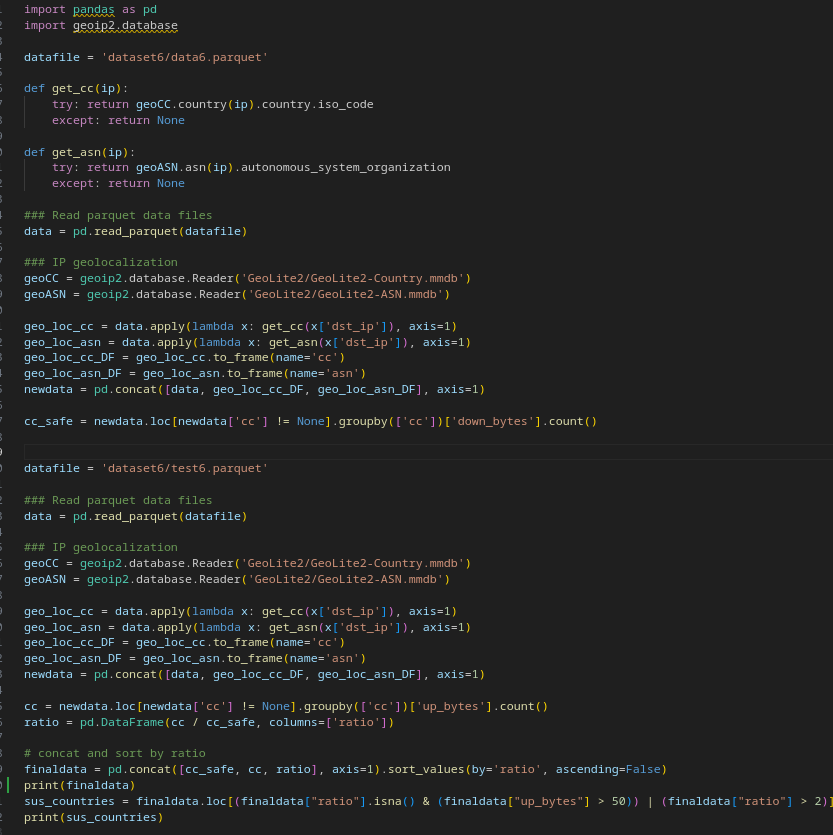
## Anomalous External Destinations

To identify anomalous behavior from external users accessing the corporate public servers, a rule was implemented to analyze traffic patterns at a country level. The methodology leveraged geolocation data for destination IP addresses, obtained using the GeoLite2 Country and ASN databases. First, the script reads the known-clean dataset (data10.parquet) to establish a baseline of normal traffic distribution across countries. Each destination IP (dst\_ip) is geolocated to extract the corresponding country code (cc) and autonomous system name (asn). These are appended as new columns to the original dataframe.

From the clean dataset, the total number of download flows per country is calculated and stored (cc\_safe). This reflects the expected distribution of traffic to various countries under non-anomalous conditions. The same process is repeated on the test dataset (test10.parquet), this time recording the number of upload flows per country (cc). The logic behind this approach is that a significantly increased volume of uploads to a particular country, relative to what was seen in the clean dataset.

The ratio of upload flows in the test dataset to download flows in the clean dataset is then computed for each country. This ratio provides a comparative measure of deviation from normal behavior. Countries are flagged as anomalous under two conditions: either they do not appear in the clean dataset but generate more than 50 upload flows in the test dataset (suggesting new, potentially malicious destinations), or their upload ratio is more than double the historical baseline..

The rules applied here are:

Figure 16: get\_suspicious\_country\_code Script

* **if the ratio is Nan, which means that the country never made a request before, and has more then at least 50 requests or the ratio of requests has at least doubled when comparing to the data from the normal dataset.**

This script originated a lot of addresses so not all will be written, the countries that presented suspicious activies were:

* China – 292 addresses
  + **1.0.13.5, 1.45.71.108, 101.1.2.239, 103.103.207.97, 103.105.23.165, 103.107.47.91, 103.110.117.178, 103.112.142.87, 103.114.4.20, 103.114.5.2, 103.115.59.73, 103.116.151.94, and 103.119.0.113...**
* United Arab Emirates – 18 addresses
  + **13.105.61.21, 13.105.61.69, 13.105.61.81, 13.105.61.87, 13.105.61.90, 13.105.61.93, 13.105.61.98, 15.230.219.15, 15.230.219.20, 15.230.219.234, 15.230.219.235, 15.230.219.241, 15.230.219.34, 15.230.219.57, 15.230.219.58, 15.230.219.7, 15.230.219.81, and 15.230.219.98**
* Chile – 5 addresses
  + **40.97.14.105, 40.97.14.252, 40.97.14.83, 40.97.14.92, and 40.97.14.97.**
* Bahrain – 21 addresses
  + **13.34.78.255, 15.177.87.197, 15.177.87.247, 15.177.87.77, 150.222.7.130, 150.222.7.144, 150.222.7.161, 150.222.7.17, 150.222.7.192, 150.222.7.198, 150.222.7.200, 150.222.7.207, 150.222.7.22, 150.222.7.221, 150.222.7.231, 150.222.7.243, 150.222.7.71, 99.77.236.10, 99.77.236.14, 99.77.236.220, and 99.77.236.92.**
* Russia – 527
  + **94.137.70.118, 95.108.146.163, 46.8.52.133, 62.192.245.242,91.206.116.50, 85.31.127.36, 87.117.0.83, 81.28.187.175, 81.89.113.199, 5.3.80.2, 109.172.35.251, and 217.25.218.169…**
* Iran – 97 addresses.
  + **103.130.144.83, 109.230.223.187, 164.138.144.250, 176.102.237.152, 178.239.147.51, 185.103.129.29, 185.109.244.229, 185.11.90.205, 185.111.12.117, 185.112.168.66, 193.148.66.159, 193.246.201.17, 193.3.31.177, 194.56.148.96, 195.191.23.20, 212.46.45.186, 213.176.29.38, 217.218.68.178, 217.24.146.252, 31.14.92.193, 31.47.61.137, 45.150.89.120, 45.158.121.132, 45.86.5.11, 45.87.6.221, 5.212.168.146, 5.22.199.236, 5.253.96.49, 77.245.227.75, and 78.3…**

Since the result of the script is very large no picture will be shown.

## External Users using the corporate public services in an anomalous way

One of the points asked in the assignment was “External Users using the corporate public services in an anomalous way”and to be able to do this the group also analyzed the connection timings, this was the only field that was significant to analyze since all the remaining fields were normal with no anomalies, in order to obtain the base values, we first obtained the differences between timestamps for each source address and calculated the mean, standard deviation, maximum and minimum, then did the same for the servers data. These value values will later allow to define rules to identify anomalous addresses.

Figure 17: Script conn\_timings.py

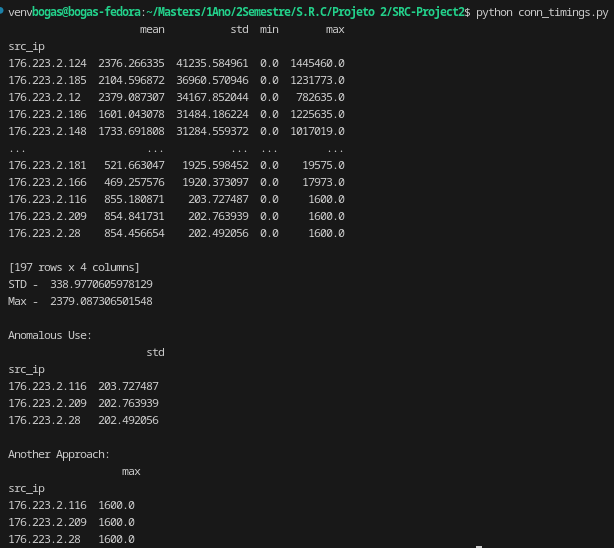
## 

So we knew that the mean standard deviation was 338 hundredths of a second, we considered this value as the minimum to be considered not anomalous which originated the rule:

* **Any source address with a lower standard deviation than the mean standard deviation would be flagged as anomalous.**

Applying this rule found the following addresses:

* **176.223.2.116, 176.223.2.209 and 176.223.2.28**

Figure 18: Result of Script conn\_timings.py

And as an extra we verified the maximum time the addresses flagged took and noticed they all have the exact same maximum value of difference between timestamps, which is also an anomalous characteristic that helped us determine that they were in fact anomalous.

# Conclusion