**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 v](#__RefHeading___Toc1384_459577603)

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

[2.5. Connection Timings ix](#__RefHeading___Toc1930_543899845)

[x](#__RefHeading___Toc1932_543899845)

[3. SIEM Rules xi](#__RefHeading___Toc1025_543899845)

[3.1. Internal BotNet, Data Exfiltration and C&C Activities xi](#__RefHeading___Toc1934_543899845)

[3.2. Anomalous External Destinations xv](#__RefHeading___Toc1033_543899845)

[3.3. External Users using the corporate public services in an anomalous way xvii](#__RefHeading___Toc1049_1445242905)

[4. Test of SIEM Rules and Identification of Devices xviii](#__RefHeading___Toc1051_1445242905)

[4.1. Internal BotNet Activities, Data Exfiltration using HTTPS and/or DNS and C&C Activities using DNS xviii](#__RefHeading___Toc1053_1445242905)

[4.2. Anomalous External Destinations xix](#__RefHeading___Toc1057_1445242905)

[4.3. External Users using the Corporate Public Services in an Anomalous way xix](#__RefHeading___Toc1059_1445242905)

[5. Conclusion xix](#__RefHeading___Toc1061_1445242905)

# 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.

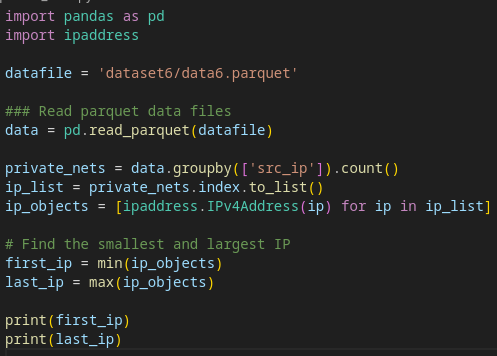
The network behavior was profiled based on source and destination IPs, protocols, traffic volume, and geolocation, enabling the formulation of SIEM rules grounded in empirical evidence. These rules were tested against known-safe and mixed-behavior datasets to validate their effectiveness in identifying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

According to the value of our mecanographic values the datasets used were the **sixth**.

# Analysis of Non-Anomalous Behavior

## Identify Private Networks

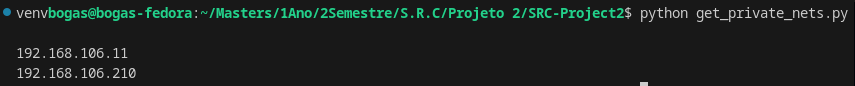
In order to identify the internal private networks used by the corporate infrastructure, a script was developed to analyze the source IP addresses from the network flow logs in the data6.parquet file. The dataset contains non-anomalous traffic from a full day of operation, serving as a baseline for typical 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.

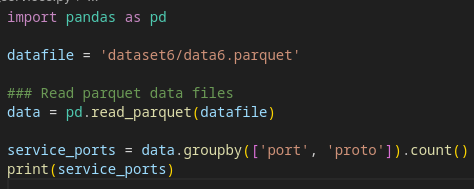
The smallest and largest source IPs found in the dataset were: **192.168.106.11** and **192.168.106.210** respectively.

## Identify Internal Services/Servers

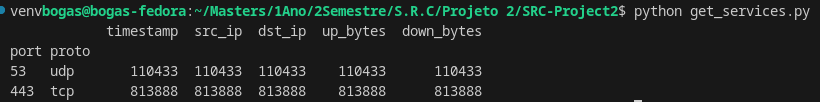
Figure 2: Result of Script get\_private\_nets

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.

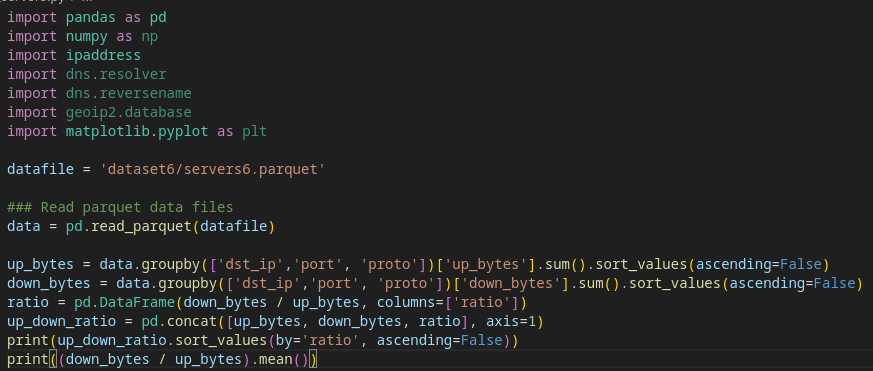
First, using the data file which includes internal traffic only a grouping operation by destination port and transport protocol was executed. This allowed for the identification of commonly accessed ports and services used internally, such as web port 54 using UDP and port 443 using TCP.

Figure 3: get\_services.py

Second, the servers file was analyzed to understand interactions with the **public-facing corporate servers** located in the 200.0.0.0/24 subnet. By aggregating up\_bytes and down\_bytes per destination IP, port, and protocol, the script calculates upload/download ratios to infer the role of each server.

Figure 4: Result of Script get\_services

This script identified two servers: 200.0.0.11 and 200.0.0.12, both using TCP in port 443.

Figure 5: get\_servers Script

## Describe and Quantify Internal Traffic Exchanges

Figure 6: Result of Script get\_servers

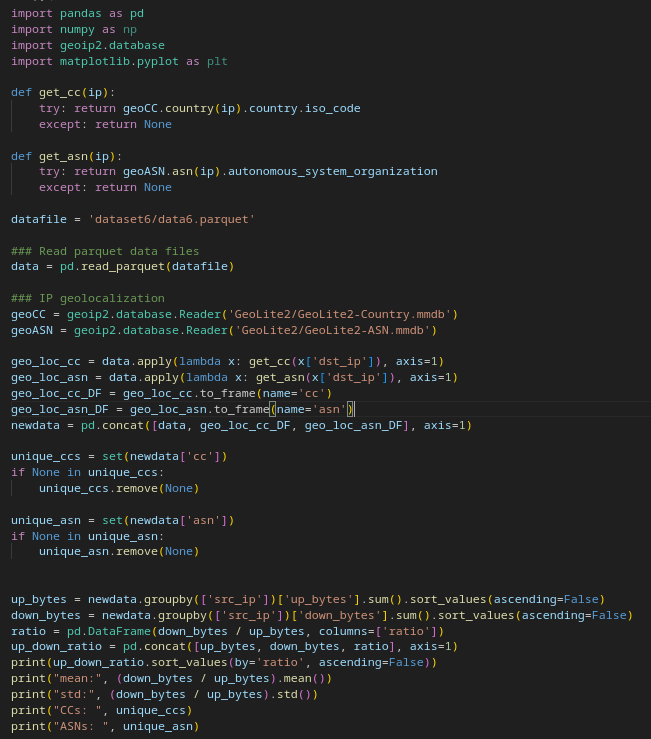
To analyze internal network behavior and characterize data flows between internal devices and both internal and external servers, the dataset data was used.

Each internal IP (source address) was evaluated in terms of total uploaded and downloaded bytes. By grouping traffic by source IP 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.

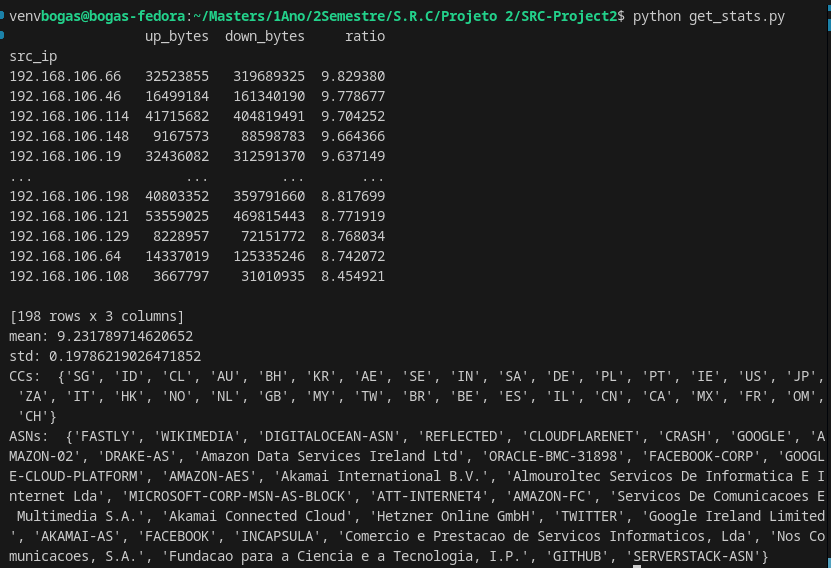
To identify interactions with external services, each destination IP address was geolocated using the GeoLite2 Country and ASN databases. The results revealed a set of distinct destination countries that internal devices communicated with, providing insights into the geographical spread of external dependencies.

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

Figure 7: get\_stats Script

The **average download-to-upload ratio** across 198 internal devices was **9.23**, with a **standard deviation of 0.20**, indicating a relatively uniform pattern of usage.

Figure 8: Result of Script get\_stats

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 servers6.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.

For each external IP, the total uploaded and downloaded data 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.

* The average download-to-upload ratio was 8.49
* The standard deviation was 0.047, indicating very consistent behavior among all clients

To evaluate how frequently external clients accessed services, the average time between consecutive flows was calculated.

* Mean inter-flow timestamp: 842.58 (1/100 seconds, ≈ 8.4 seconds)
* Standard deviation: 337.06

These values indicate moderate variation, with some clients showing steady access patterns, while others exhibit more sporadic or burst-like behavior.

## Connection Timings

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.

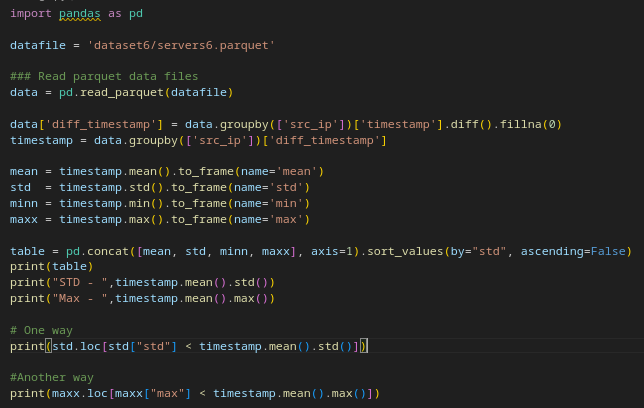
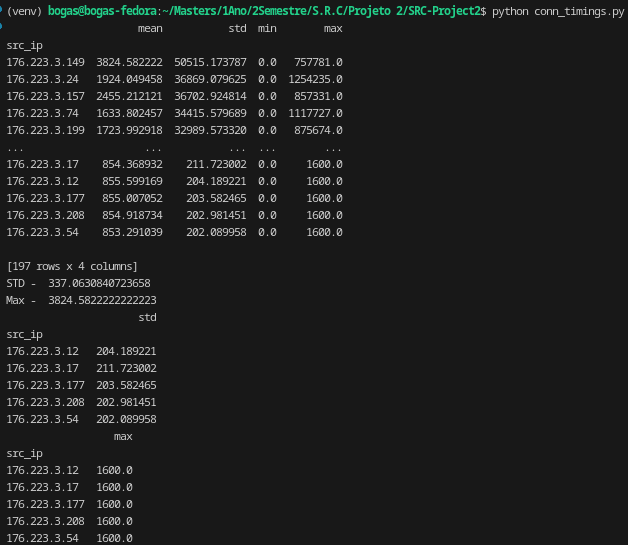


Figure 9: conn\_timings Script

## 

# SIEM Rules

Figure 10: Result of Script conn\_timings

## Internal BotNet, Data Exfiltration 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 198 internal IPs:

* Mean ratio: 7.36
* Standard deviation: 0.60
* Minimum: 5.74
* Maximum: 9.16

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: 1.11
* Minimum: 0.100
* Maximum: 9.47

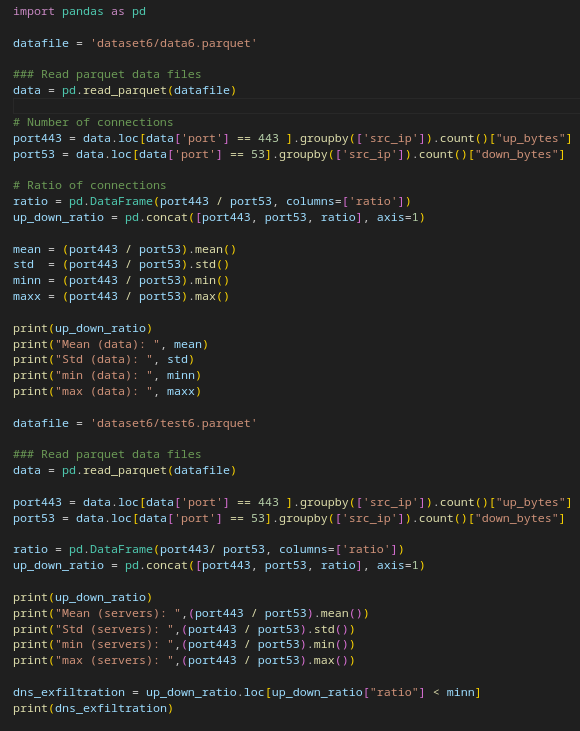
By comparing the values we could conclude that the standard deviation and the minimum were the values that most diverge from the clean results. When analyzing the ratio, a low value means that a lot more data was consumed/obtained by external devices then by the servers, which could mean data extrafiltration, when 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.

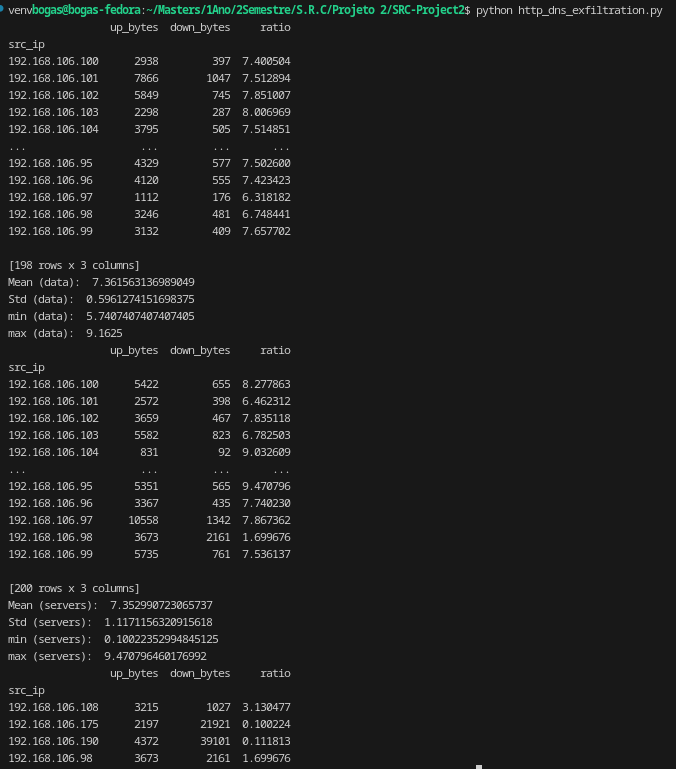
Any internal host with a significantly lower ratio than the baseline minimum (5.74) was flagged for suspicious behavior. Such a pattern may indicate:

* Botnet command & 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 rules was:

* **Any value below the minimum obtained from the clean dataset is considered anomalous**, this was our decision since

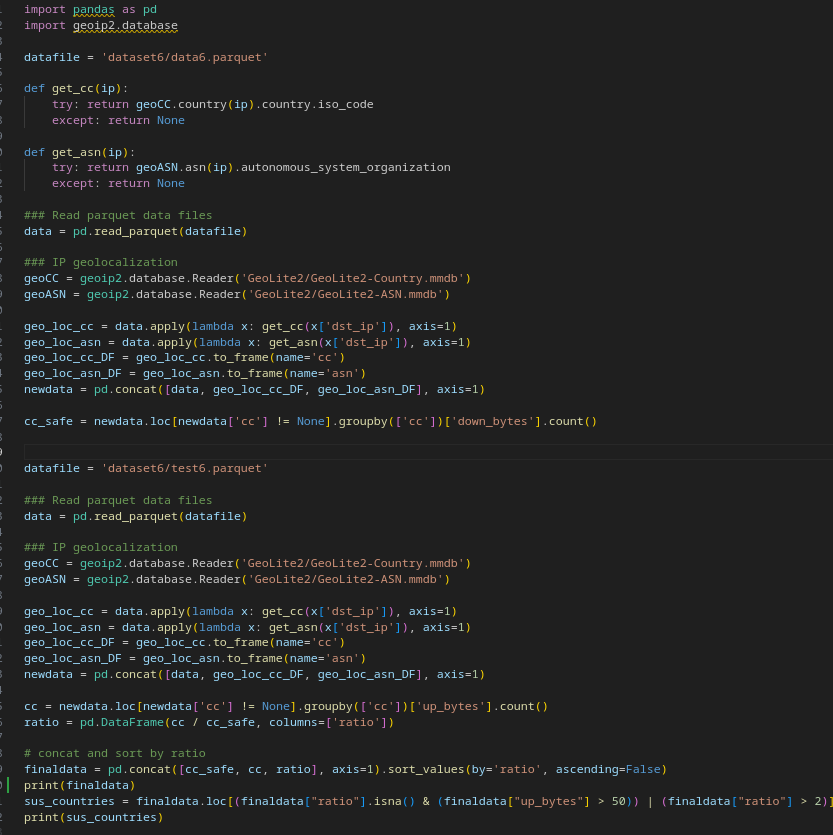
Figure 11: http\_dns\_extrafiltration Script

Figure 12: Result of Script http\_dns\_extrafiltration

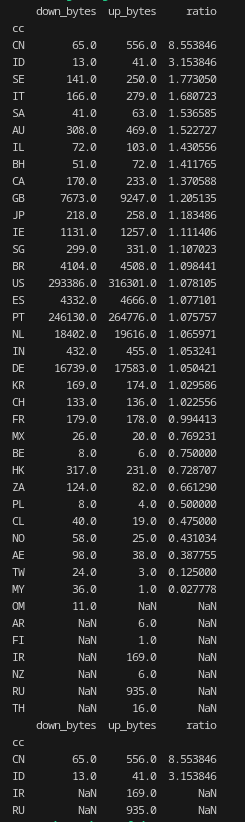
## Anomalous External Destinations

To be able to detect the suspicious countries, the group decided to first analyses the clean dataset and then compare to the values of the anomalous dataset, this allowed to verify new countries that never made contact with the internal servers and countries were the ratio of requests varied to the point that it should at least trigger a warnig.

The rules applied here are:

Figure 13: 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 request has at least doubled.**

Figure 14: Result of Script get\_suspicious\_country\_codes

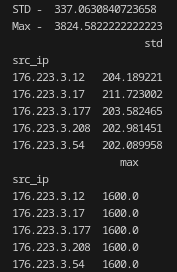
## External Users using the corporate public services in an anomalous way

As mentioned before, the script used to obtain the values and were the rules are applied is conn\_timings.py, here we couldn’t compare to the value of the clean dataset since this doesn’t have external users. So in order to understand the value of this dataset for this specific reason, we grouped all the timestamps per source address, and then obtained the mean, standard deviation, maximum and minimum. And after analyzing the value obtained we knew the goal was to find the external users behaving in an anomalous way, so the behavior had to be different from a normal user, like one from a bot, and we assumed that a bot would have a mechanic behavior with less variation, so we focused on the standard deviation.

So we knew that the mean standard deviation was 337 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.**

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.

Figure 15: Result of Script conn\_timings

# Test of SIEM Rules and Identification of Devices

## Internal BotNet Activities, Data Exfiltration using HTTPS and/or DNS and C&C Activities using DNS

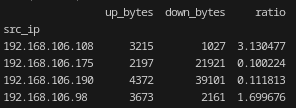
By comparing the HTTPS-to-DNS connection ratios between the clean dataset (data6.parquet) and the test dataset (test6.parquet), we observed that while the average ratio remained relatively consistent between the two datasets (around 7.36), the minimum observed value differed significantly. In the clean dataset, the lowest ratio recorded was 5.74, whereas in the test dataset it dropped sharply to 0.10.

This sharp decline in the minimum ratio indicates the presence of internal devices in the test dataset whose behavior deviates substantially from the baseline. Specifically, some devices are making a disproportionately high number of DNS queries relative to HTTPS connections — a pattern not observed in the normal traffic.

We identified two key anomaly scenarios based on the HTTPS-to-DNS ratio:

* Ratio < 1: This indicates that the number of DNS flows exceeds HTTPS flows. Such behavior is highly unusual in a typical enterprise environment and is a strong indicator of DNS-based data exfiltration. Malicious software may use DNS queries to covertly transmit information to an external server, bypassing traditional detection methods.
* 1 < Ratio < 5.74 (i.e., below the clean dataset’s minimum): While these devices still make more HTTPS connections than DNS queries, the ratio is significantly lower than expected. This pattern may point to command and control (C&C) communication, where a compromised host periodically contacts external infrastructure via DNS to receive instructions. Unlike exfiltration, these flows are typically smaller and more discrete but still abnormal.

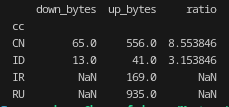
With the rule of detection being any values below the minimum the following addresses were found:

Figure 16: Internal BotNet Activities, Data Exfiltration using HTTPS and/or DNS and C&C Activities using DNS detected Addresses

## Anomalous External Destinations

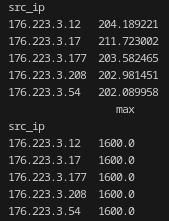
By comparing the quantity of connections per country between the clean dataset (data6.parquet) and the test dataset (test6.parquet), we observed that some new countries appeared in the anomalous dataset with a reasonable quantity of requests (more than 50) and besides this some countries appeared to increase the amount of requests, and we considered that if the ratio is bigger then 2 it should be flagged as anomalous.

This led to the following countries:

Figure 17: Anomalous External Destinations detected Addresses

## External Users using the Corporate Public Services in an Anomalous way

After analyzing the difference between the timestamps and as well as the standard deviation, it was decided that any address with a standard deviation below the average (337.06) would be flagged as anomalous since it doesn’t vary a lot which could indicate a more automatic or even mechanical like behavior, like one from a bot. This resulted in the following addresses being considered anomalous:

Figure 18: External Users using the Corporate Public Services in an Anomalous way detected Addresses

# Conclusion