TALLINN UNIVERSITY OF TECHNOLOGY

School of Information Technology

Illia Petrash 184659IVCM

USER BEHAVIOUR DURING COVID-19 FROM THE PERSPECTIVE OF A TELCO SERVER

Master's thesis

Supervisor

Olaf Manuel Maennel

PhD

Tallinn 2020

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Illia Petrash ....

(signature)

Date: 19.05.2020

Abstract

There is a problem of detecting anomalies in huge sets of data, thus many methods are available for dataset and pattern analysis. These methods mostly apply its techniques for predefined datasets with known anomalies, thus it makes patterns more obvious.

A unique dataset that is used for method development is taken from the very important time slot – before and after the COVID-19 pandemic has happened in Estonia. This dataset analysis is a valuable contribution to the currently ongoing research about an increased amount of malicious activities after COVID-19 lockdown.

Besides, this thesis provides a new method for detecting anomalies in big datasets with the help of selected tools and techniques. "Multiple login attempts from different locations at the same time" use-case is taken as it can help to build a good understanding of the log structure and log patterns. Also, it can be useful for future research in criminal case investigation by Estonian police.

This thesis also provides the analysis of existing parsing and visualizing tools, comparisons of technical characteristics of these tools. An enterprise system is described and techniques are summarized for further research detecting malicious logins and analyzing login patterns within an enterprise network. Since the current method uses a recently extracted dataset of logs from the real system environment of an enterprise company, it makes this methodology more interesting as it creates an analysis of different types of logs with different structures and data types. The created method includes feature selection, data grouping, data aggregation, and dataset cleaning.

The contribution made in this thesis aims to create a method for conducting anomaly detection analysis of a huge dataset taken from the very important time slot – during the COVID-19 pandemic lockdown in Estonia. This dataset consists of pre-COVID-19 and post-COVID-19 time slots, which can be analyzed for any interesting malicious activities. The output of the method will be a Top 10 emails before the lockdown has happened and Top 10 IP addresses after the lockdown has happened in Estonia. Besides, Top 5 emails with the highest number of failed logins and Top 5 emails with the highest number of successful logins will be described. A range of methodological steps will be completed to investigate the dataset. The dataset is taken from the real, live system without any predefined anomalies, which is currently applicable to only a specific use-case set, but the applied set of tools allows to adjust the current method to different types of use-case sets.

This thesis is written in English and is 103 pages long, including 6 chapters, 92 figures, and 1 table.

List of abbreviations and terms

ACL	Access Control List
IDS	Intrusion Detection Systems
APT	Advanced Persistent Threat
CLM	Credential-based Lateral Movement
LOF	Local Outlier Factor
ML	Machine Learning
PADM	PageRank-based Anomaly Detection Method
GLR	Generalized Likelihood Ratio
YMS	Yahoo monitoring service
EGADS	Extensible Generic Anomaly Detection System
IP	Internet Protocol
3W	Who, Where, and When technique
SQL	Structured Query Language
ELK	Elasticsearch, Logstash, Kibana
WEKA	Waikato Environment for Knowledge Analysis
URL	Uniform Resource Locator
ARFF	Attribute-Relation File Format
AGPL	Affero General Public License
GUI	Graphical User Interface
EDA	Exploratory Data Analysis
JSON	JavaScript Object Notation
CSV	Comma-Separated Values
GPU	Graphics Processing Unit
CUDA	Compute Unified Device Architecture
RAM	Random Access Memory
K-NN	K-Nearest-Neighbor

Table of Contents

List of Figures

List of Tables

1. Chapter 1

1.1 Introduction

A huge quantity of connected networks in most of the companies makes security even more complicated. There is always a problem of collecting data in one place and this place is needed to be secure enough.

TechRepublic source mentions, that during the COVID-19 pandemic which has happened in 2020, almost all of the small and big enterprises are suffering from data breaches and malicious attacks which include phishing and spamming, etc. This situation shows a diversity of possible attack vectors which is used by attackers, thus there is a possibility to analyze it and get an understanding of what attackers are trying to achieve if these will be found. Herjavec Group describes multiple kinds of attacks conducted during COVID-19.

As mentioned by Zhaoli Liu et al, usually, the situation follows the next scenario: an attacker steals and uses the credentials of one computer to get access to another computer in this chain. The idea is that an attacker moves between systems of all these computers until he/she will not get needed data/information.

According to Zhaoli Liu et al, most of the traditional Intrusion Detection Systems (IDS) detects some of the malicious activities that happen in the network traffic on the initial stage of the analysis.

Nevertheless, there are some access control policies and systems described by Zhaoli Liu et al. These policies include Access Control List and Active Directory which are usually failing to battle with credential-based attacks due to misconfigurations and over-entitlement of users regarding what they can access. These are just a few examples of what is lacking administration and improvement to combat malicious activities within the network. There is a possibility to get information about the needed user by simply getting access to his/her account in one of the enterprise's systems. These systems usually contain highly sensitive information about its users and customers. And by getting access to one of the user's account, it makes much easier to get access to another account of this user but on the different system. Usually, users reuse their passwords for different systems. It is also useful for the competitors to gain access to these accounts by doing spear phishing to accounts of other users.

According to the Zhaoli Liu et al. paper, there are also some misconfigurations between different systems, so the user gets access to another user account due to the system's misconfigurations.

Thus, it is important to keep an eye on what is currently happening in live systems as well as in data that is backed up, as automated Intrusion Detection tools are not able to properly react due to the high volume of data and complicated system infrastructures.

In this master thesis, dataset analysis is provided, which includes data from during the COVID-19 lockdown that has happened in Estonia. The dataset is extracted from the critical system which is also interesting for Estonian police criminal case investigations. Police queries needed data from the Hadoop ecosystem, thus the level of criticality of the system is very high.

Besides, a new method for detecting malicious logins and analyzing login patterns within enterprise networks is presented. This method is unique as it includes an understanding of the current system and it's logs structure, after that defining a combination of tools that will give a needed result – it will reveal Top 10 IP addresses into which the biggest amount of emails tried to log in before and after lockdown has happened in Estonia. Also, it will reveal Top 5 users with the highest amount of successful and failed login attempts. These use-cases were chosen to help to analyze login patterns for a defined set of tools. The bigger the company, the more customer data it needs to handle, thus a lot of users trying to log in at the same time to different systems. Multiple logins into one account at the same time shows some interesting patterns, especially if these logins are from different locations. This is one of the use-cases which is interesting for the Estonian police when trying to investigate criminal cases in the future. Identification of an account that was compromised by another user changes the result of the investigation, thus a defined method for pattern recognition and anomaly detection is needed.

COVID-19 pandemic gave another jolt towards improving current methods within the anomaly detection domain as many attacks are conducted by criminals during pandemic lockdown which shows how vulnerable big enterprise systems are.

1.2 Motivation

Most of the enterprises suffering from security breaches, thus it is important to find ways to protect user's sensitive data which is mentioned by TechRepublic source. Data leaks happen every day, companies losing their money and trust of the customers due to these data leaks.

According to Herjavec Group blog post, critical infrastructures also suffering from malicious activities. Needless to say, most of the criminal cases investigated by police are also consist of a mobile forensic part which is helping to resolve many cases.

There is no defined method and set of tools that will help to understand the structure of the logs and find possible guesses about anomalies in login patterns.

The motivation of this work is to provide a method for finding patterns of malicious actors during the COVID-19 pandemic time slot which is included in an analyzed dataset. Multiple login attempts from different locations at the same time is a good example of starting point for data investigation. To help to achieve this goal, some tools and data analysis libraries will be used.

A unique dataset, which is used for detecting any kind of anomalies includes data from two very important periods: before and after the COVID-19 pandemic happened. From the 1st of January 2020 until the 20th of April 2020. This dataset discloses other research questions about the anomaly detection domain for future investigations within the company and Estonian police. Millions of people are staying at home and there is the question if all of these people are legitimate and if there are any emails or IP addresses that are suspicious. COVID-19 dataset makes an analysis more interesting and discloses interesting behavioral patterns of the users.

1.3 Problem Statement and contribution

The importance of detecting anomalies in logs has increased, thus new methods are expecting to be involved in the process of anomaly detection and pattern analysis. Sometimes it happens that multiple users, login into account from different locations at the same time. It can be caused by different reasons. A unique dataset that is used for method development is taken from the very important timeslot – before and after the COVID-19 pandemic has happened in Estonia during 2020. This dataset analysis is a valuable contribution to the currently ongoing research about an increased amount of malicious activities after lockdown has happened in Estonia. This dataset is taken from a massive telecommunication company that includes millions of users using their services daily. It is a mission-critical organization, thus dataset which was extracted from such a big company during a pandemic situation in Estonia will be evaluated as highly interesting from the anomaly detection domain. According to Herjavec Group, there were already reported many cases of increased malicious activeness during COVID-19 pandemic in small as well as big enterprises around the world. These are including phishing emails and other malicious links. There is no research conducted regarding anomaly detection during the COVID-19 pandemic.

The research problem this study aims to resolve is to define a method of analyzing a huge quantity of logs, login patterns which is useful for further investigation of malicious activities within the system. The method includes plotting of different variations of data interpretations, analyzing time slots before and after COVID-19 pandemic has happened in Estonia. The idea is to conduct an analysis of the current situation and to describe a methodology used on the way of achieving this analysis. The proposed method helps to understand the reasons for multiple logins from different locations at the same time into one system. Top 10 IP addresses from which the biggest amount of users were logged into their emails before and after lockdown has happened will be listed. Top 5 users with the highest amount of successful logins and 5 users with the highest amount of failed logins will be listed as well. User's data will be anonymized as it is taken from the real, live system of an enterprise. As a consequence of this, the proposed method and defined set of tools is used for detecting anomalies in logs for future research in criminal case investigations by the Estonian police.

1.4 The scope

The goal of this study is to analyze a dataset from a telecommunication company, which consists of very sensitive information before and after the COVID-19 pandemic lockdown has happened in Estonia. In addition to that, to identify a method for defining login patterns in the set of logs, which will help to detect multiple logins from different locations at the same time.

The outcome will be an analysis of the dataset during COVID-19 lockdown and a method for identifying login patterns and detecting Top 10 IP addresses into which the biggest amount of emails tried to log in before and after lockdown has happened. Top 5 users with the highest amount of successful logins and 5 users with the highest amount of failed logins will be listed as well. The method will include a graphical representation of the activities of the users during a given time slot.

1.5 Limitations

A unique dataset, which is analyzed in this thesis is consists of valuable records before and after the COVID-19 lockdown has happened in Estonia. Thus, new threats are evolving and APT groups are actively involved in it. Limited information about new techniques and patterns which criminals are using makes an analysis more complicated and unique at the same time.

Not enough information about other systems that are behind the system currently used for anomaly detection in this work (more time and permissions are needed to discover other user's behavioral patterns). The dataset consists of a few vital features, thus some features such as what the user was doing after he/she logged in and into which system, duration of the session, these are absent. Another limitation, if the dataset is too big, additional clustering or summarization method will be applied in future work to comply with the allowed size of the dataset available in analyzing tools such as Google Collaboratory, WEKA or Rapid Miner 4.

1.6 Structure of the thesis

In Chapter 1, motivation, problem statement, and contribution are presented. In Chapter 2, state of art has been discussed and compared with the current method. In Chapter 3, the current dataset is described and the methodology is developed. In Chapter 4, the main focus is on the analysis of existing parsing and visualizing tools, comparisons of technical characteristics of these tools. An Enterprise system is described and techniques are summarized for further research detecting malicious logins and analyzing login patterns within an enterprise network. Since the current method uses a recently extracted dataset of logs from the real system environment of an enterprise company, it makes this methodology unique as it creates an analysis of different types of logs with different structures and data types. The created method includes feature selection, data grouping, data aggregation, and dataset cleaning.

The dataset is cleaned and plots are depicted with the help of Google Collaboratory environment.

In Chapter 5, results from the SpectX tool and Google Collaboratory are retrieved for further analysis. Analysis of the plots is provided and different kinds of plots are present different times lots. All methodology steps are completed in this chapter.

Chapter 6 concludes the research and summarizes the achievements of the analyzed dataset during the COVID-19 pandemic.

2. Chapter 2

2.1 Related work

There are plenty of papers with machine learning techniques, algorithms, and methods that describe anomaly detection in massive sets of logs. Most of the existing methods require a priori knowledge about previous attacks in the systems (known patterns of previous attacks), thus these methods are not capable of detecting new or unknown threats according to Zhaoli Liu et al. Method, described in this thesis does not use any dataset with previously known attacks – without any ground truth. This dataset consists of original logs taken from the real system, where real users login into it. The current state of the art anomaly detection approaches and methods are lacking scalable definitions of methods, use-case restrictions, the difficulty of use (which includes complicated explanations and low-level knowledge about algorithms and programming languages) and a large number of false positives which is also mentioned by Nikolay Laptev et al. Method described in this thesis works with the huge as well as a small set of data but it requires some new data summarization techinques to be applied in future. The methodology provides an overview of existing tools, and it is effective at finding anomalies, but there is still room for improvement. This method can be improved in the future for use by the Estonian police for criminal case investigations. As will be discussed later in this thesis, there are possibilities for police to use the current method as it is created based on the dataset components structure, they are using daily. But this information cannot be disclosed in detail due to confidentiality issues.

The uniqueness of this dataset is that it is taken from a very important period of our lifetime, during the COVID-19 pandemic situation which impacted a cybersecurity situation in the whole world. There are no anomaly detection research papers available that are considering a dataset of a huge telecommunication company during COVID-19. This makes this thesis a unique opportunity to share the analysis and to impact the current cybersecurity awareness situation.

Nevertheless, the number of research works that are committed to the general anomaly detection area increases. According to Xiaoben Yan et al. these works can be divided into two groups: the pre-prevented and the post-found methods. The main idea of pre-prevented methods is that these are trying to predict future events (if it is abnormal or not). On the other hand, post-found methods are more common as these are dealing with previously gathered logs. The method described in this thesis tries to solve the post-found problem.

Zhaoli Liu et al. in their paper describe a method that is also not using a dataset with previously known attacks, but it includes a highly complicated workflow of the proposed method which includes K-prototype clustering and the k-NN classifier. The method described in this thesis has a limited set of data, thus it requires a different approach. It will not be feasible to include machine learning techniques due to the limited quantity of features extracted from the dataset. Besides, the system which logs were extracted from is a very complicated one and there are other systems behind this system which are needed more investigation, these systems were not accessible due to confidentiality restrictions. There are not many papers that describe any similar methodology based on the limited dataset. Also, Zhaoli Liu et al. try to describe a feature selection of logging activities. This technique is used in the method described in this thesis as well. It gives a characterization of the user's behavior. To see, if the user logged in into the system from the usual host and in a usual time slot. Also, Zhaoli Liu et al describe a 3W technique (Who, Where, and When), which is useful for general dataset classification in the described method.
Domingos S. de O. Santos Júnior et al. in their paper describe some of the characteristics of the existing data mining platforms and tools. Domingos S. de O. Santos Júnior et al. paper is helpful to gain a basic idea of how tools can be useful, namely some of the information was collected for Table 1, which was modified and compares these tools and platforms.

The method presented in this thesis will be a combination of some statistics and log parsing techniques.

Leman Akoglu et al. in their quite extensive survey discuss graph-based anomaly detection techniques, The survey shows the problem-specific challenges of anomaly detection taking into account the structure of datasets, namely the unbalanced nature of the data. Anomalies are rare, thus only a very small amount of data can be labeled as abnormal. The cost of the mistake made while labeling as an anomaly or not depends on the type of the application. Another point is that the more adversaries know about current techniques to bypass intrusion detection algorithms, the more information they can gather to create new, sophisticated techniques and ways to bypass the detection and fit-in into the normal behavioral pattern. Leman Akoglu et al. survey is useful for defining what is anomaly regarding available datasets by using a specific set of tools. The graph-based approach is considered as a different technique which will not be covered in the method developed in this thesis.

There are some papers regarding Time Series Forecasting methods and it's anomaly detection implementations.

Feature selection is another important part of the anomaly detection method. It is essential to be able to choose the right features because the future analysis and method construction will depend on it. In addition to all mentioned above, Miao Rong et al. provide an overview of feature selection methods for big data mining. Current challenges and difficulties of mining valuable information from big data environments are discussed. Available methods are extremely specific, thus universal approach/method is needed. Miao Rong et al. assure that it is still an open question. There are a few problems of feature selection methods described by Miao Rong et al:

1. Traditional feature selection methods usually require large amounts of learning time. Thus, it is hard for the current speed to process according to the rapidly growing big data;
2. Big data includes a huge amount of irrelevant and redundant features in its datasets. Also, there is a potential existence of different kinds of noises, which is another pitfall for successful anomaly detection methods;
3. Some amount of data could be unreliable due to different losses, which consequently enhances the complexity of feature selection, and as a consequence quality of anomaly detection methods.

Three feature selection methods exist: filter methods, wrapper methods, and embedded methods. Anomaly detection method which is presented in this thesis takes only the best practices of the feature selection techniques from Miao Rong et al. work. In the current thesis methodology, feature selection is done manually which helps to learn more about the dataset.

V.Bolón-Canedo et al. research explains the evolution of feature selection techniques in different areas as well as feature selection costs. Verónica Bolón-Canedo et al. in their research paper deliver a review of feature selection methods on synthetic data.

All these papers provide an overview of complicated dataset feature selection techniques due to the large volume of data. Besides, all these methods help to get a better understanding of the anomaly detection domain. More experimental work on feature selection algorithms for comparative purposes can be found by Anil Jain at al. and Mineichi Kudo et al. Exploration of Big Data 3V's: Volume, Variety, and Velocity is described by Cheikh Kacfah Emani et al. Writers show the difference between different data analysis techniques and Big Data management.

The method described in this thesis has a simpler approach as there are not so many features available in the dataset. It is not very convenient to use complicated techniques for the current dataset, thus it is much easier and cost-effective to select features by analyzing what is available and what is needed to be achieved. The current method combines multiple tools for solving anomaly detection problems. Needless to say, the current method solves a specific use case. By using described in this thesis method it is possible to implement other solutions on the top of it. Thus, it makes the described method more flexible to new upcoming future techniques.

According to Mohiuddin Ahmed and his research findings, data summarization can be a useful and effective technique for network management, when it comes to analyzing the big amount of data and extract needed information from it. In most of the cases, the label shows if the behavior is normal or anomalous. In contrast, in this thesis real network traffic is used from the live system, thus labels are not available. Based on Mohiuddin Ahmed's paper, there are few methods: statistical, linguistic, and machine learning. This is a good paper to detect which type of data summarization is suitable for a particular dataset. Referring to Mohiuddin Ahmed's paper, data grouping and data aggregation ideas are useful to develop a current method, but the whole approach of this thesis consists of many other techniques and tools which are following a different approach, meaning aggregation and grouping are only small parts of the method. In another paper, Mohiuddin Ahmed proposes a summarization technique (SUCh technique) that can create summaries which, when used as input to anomaly detection algorithms, yield similar or better performance than anomaly detection applied to the original data. All these summarization techniques can be used in future work, for improving a feature selection in big datasets of logs. In contrast, the method described in this thesis, the whole dataset is taken to the analysis, and after that, the system is learned to understand which features are needed and which are not. The Current method consists of steps to complete to eradicate outliers and clean the data to the extent, that data analysis can be conducted. Yu Cui et al. research paper describes a 2D window matrix form instead of the accustomed vector. The difference between Yu Cui et al. proposed method and the current method is that proposed in this thesis method uses a set of tools, a combination of which allows to parse logs and conduct an anomaly detection analysis by using one tool - SpectX (in addition to Google Collaboratory platform and Pandas Python library). Needless to say, each method has different strengths toward different target systems. It is hard for developers and researchers to know which is the best method for their practical problems at hand. In the case of the current method, the Google Collaboratory platform and Pandas Python library allow conducting a basic analysis with a dataset, namely grouping, aggregating, and evaluation of the data.

Yu Cui et al. in their paper use the Log Key Extraction technique for log parsing whereas the current method uses the SpectX tool. SpectX tool works very fast with a big amount of data, thus it is one of the most efficient tools based on the comparison table provided in this thesis. Xuze Xia et al. researchers proposed two methods for anomaly detection based on the machine learning ensemble models. The mixture of Experts is another technique described by Xuze Xia et al. Method described by Xuze Xia et al. uses experiment data which includes training and testing datasets (it includes ground truth), whereas current method uses real, original dataset (without any testing datasets – as there is no information about previous possible anomalies) from the live system.

Xiaoben Yan et al. in their paper present a log anomaly detection method named PADM (PageRank-based Anomaly Detection Method) based on the graph computing algorithm. The method described by Xiaoben Yan et al. uses transformation from logs to graphs by using records and identifiers whereas in this thesis there are no defined identifiers and log structure varies from time to time depending on the event. In this case, it is better to learn and to understand what is the meaning of different rows in logs. The current method is good for analyzing log sequences as well as logs itself because sometimes the log itself could look normal but log sequence abnormal. The main idea is that after having a general picture of what is going on in logs, it is possible to construct another methodology to adjust it to the dataset.

According to the United States Computer Emergency Readiness Team, the COVID-19 pandemic has created an enormous amount of APT groups that are trying to compromise user's accounts and networks. Thus, the dataset, which is involved in analysis is a unique opportunity to discover possible new attack vectors and patterns.

Current approaches in automated anomaly detection suffer from a large number of false positives that prohibit the use of these systems in practice.

The main difference between the method described in this thesis and the above-related works is that it uses the original dataset without any predefined outliers/anomalies. This dataset is taken from a very important time slot before and after COVID-19 lockdown has happened in Estonia. As a consequence of this, the proposed method and defined set of tools could be potentially used for detecting anomalies in logs for future research in criminal case investigations by the Estonian police.

Besides, the current method applies to only a specific use-case set, but the applied set of tools allows us to adjust the current method to different types of use-case sets.

3. Chapter 3

3.1 Definition of the created method

Based on the information retrieved from the dataset, it is possible to describe the method which was used to achieve the results. Firstly, to disclose the use case of detecting multiple logins at the same time from different locations, it is needed to analyze the dataset.
Secondly, the characteristics of abnormal behavior are analyzed. Besides, session information is used as the main source of information about user behavior within the system. For login activity, the following fields are extracted: IP address, Email, Date, Time, Geolocation, Event (fail, success). The extracted information contains numerical and categorical attributes. The time slot of the analysis: from the 1st of January 2020 until the 20th of April 2020.
An important aspect is to get as much information as possible about the system users tried to login in. This information is useful as later it will allow getting a full overview of what logs are consist of. Feature extraction will help to do that.
Feature extraction description:

1. Login physical locations. This feature gives an understanding of where the user was located at the moment of login into the system. In this case, it means the city in which the user was located at the moment of connection. It can be done by comparing databases of IP addresses and physical addresses. Geo point is available and makes geolocation detection much easier. This feature can be extracted from the data source;
2. Date and time point of the established connection. This feature means a login of the user at some time point. It gives an exact date and time point. This feature can be extracted from the data source;
3. Email address. This feature describes into which account the user tried to login in. This feature can be extracted from the data source;
4. Several passwords failed times during the given time slot. This feature is used to get an overview if there are any cases with very frequent failed attempts during a short timeframe. This feature can be extracted from the data source;
5. The number of passwords succeeded times during the given time slot. This feature can be extracted from the data source.

3W (Who, Where, and When) technique describes a user's behavior by pointing out the date, time, location, IP address, email (or username). But not all the aspects could be used by the method described in this thesis, due to the limited dataset properties. Thus, the 3W technique is used in the method but to a limited extent.

General information about the dataset:

1. Data type information about dataset;
2. Names of the columns displaying;
3. List of the most frequently appeared IP addresses in the dataset;
4. Amount of days during which data were collected;
5. To calculate the quantity of successful logins in total (in numbers);
6. Plotting all successful and failed logins during the period of dataset collection;
7. To find an account with biggest amount of IP addresses tried to login into it;
8. To find how many different IP addresses successfully logged in into the system;
9. To find how many different IP addresses failed to log in into the system;
10. To find how many unique emails in the dataset;
11. To find how many unique IP addresses in the dataset.

Method consist of the following steps to accomplish:

1. Log collection related to chosen system. Feature selection;
2. Select logins from the beginning of January 2020 till the 20th of April 2020;
3. To select all failed and successful logins and unite them in one dataset. Dataset has been divided into two classes: failed logins and successful logins and united into one general class (which excludes outliers);
4. To filter out "Duplicate password submission" attempts. These kinds of attempts could be created due to the user's device failures because of the old password saved on the old email application. It will lower down the false alarm rate;
5. Geolocation of the users;
6. Calculation of how many different IP addresses successfully logged in into one account;
7. To depict geolocation of the users (based on the IP addresses) on the map for visual analysis;
8. To create a list of users, who tried to log in to multiple accounts. Meaning, into which accounts each user tried to log in with the same IP address.

Possible ways to analyze dataset taking into account time slots before and after COVID-19 pandemic situation has happened:

1. To analyze a frequency of failed and successful logins day/night time;
2. To visualize mean, min, max, quantiles of failed and successful logins before and after lockdown has happened in Estonia;
3. To visualize a distribution of the amount of failed and successful login attempts on the weekly basis during the whole period of time;
4. To visualize a distribution of the amount of unique login attempts;
5. To visualize a distribution of the average amount of unique locations per user daily;
6. To visualize a correlation of the total number of attempts to a total number of locations during the whole period;
7. To visualize on the world's map distribution of unique user's locations before lockdown has happened in Estonia;
8. To visualize on the world's map distribution of unique user's locations after lockdown has happened in Estonia;
9. To visualize a distribution of unique user's locations before and after lockdown on the scale of Europe (to take a closer look at the situation within Europe);
10. To visualize a distribution of the number of email addresses into which one IP address tried to login into (to see if there are any outliers) and to depict it on the world's map;
11. To visualize a distribution of outliers before and after lockdown has happened;

3.2 Output of the method

1. Detailed analysis of the dataset taken from the time slot before and after COVID-19 pandemic situation has happened in Estonia;
2. Top 10 IP addresses into which the biggest amount of emails tried to log in before lockdown has happened;
3. Top 10 IP addresses into which the biggest amount of emails tried to log in after lockdown has happened;
4. Top 5 users with the highest amount of failed login attempts;
5. Top 5 users with the highest amount of successful login attempts.

4. Chapter 4

4.1 Tools comparison and analysis

Tool selection is a very important step during the method implementation stage, thus this chapter will cover some of the most common tools for anomaly detection and data analysis. Log parsing tools and monitoring systems: SpectX, Graylog, Splunk, Plumbr.

SpectX: SpectX is a powerful log parser and at the same time query engine for investigating anomalies and incidents – raw logs when it comes to Big Data. It can manipulate data from multiple log sources: ELK, Hadoop, SQL-databases, etc.

According to SpectX development guide, this is an ideal parsing tool for the following tasks:

1. Large-scale log review;
2. Root cause investigation during incidents;
3. Historical log analysis;
4. Performing virtual SQL-join across multiple sources of raw data;
5. Analyzing legacy data;
6. Ad hoc queries on data dumps.

There is a free license to download a Desktop version of the SpectX. There are 4 cores available in the free version. To get an unlimited quantity of cores, a license is needed. For testing purposes, a free license is enough.
SpectX has been chosen as a tool that can parse a huge amount of data very fast. There are also other parsing tools available on the Internet. But this master thesis is a good opportunity to try a new tool for cleaning data and for looking for any interesting patterns, as it completely fits research needs, namely works with a huge amount of data. Besides, for this master thesis research project was provided unlimited SpectX version. Thus, SpectX was integrated with Hadoop to making data available for extraction. SpextX is a very useful and simple tool. Intuitively easy logs can be extracted and analyzed.

There are a few buttons on the main SpectX panel, it can be seen from Figure 1:

1. Button “New”. It is possible to create a new file to write a query;
2. Button “Run”. For running the query;
3. Button “Input Data”. To find needed source directory in HDFS and to get its path to the main query window;
4. Button “Insert”. To insert some JavaScript function, Timestamp or Init Block.

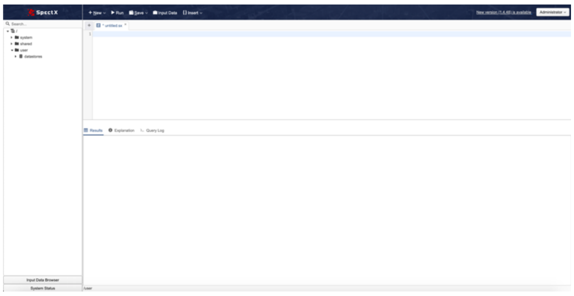


Figure 1. *SpectX interface.*

At the bottom of the window, the Input Data Browser allows choosing a needed dataset from the set of logs in the source folder. There is a System Status button for checking the performance of running queries. Such fields are available: QueryID, Username, Mode, State, Running Time, Progress.

On the left side of the window, there is a tree of folders that could include data sources folders. The file extension in SpectX is .sx. Besides, it is possible to share a folder with other people so that they can edit queries and test them in their environment. Also, there is a panel that depicts the results of the running query and explanation window.

After running any queries, results can be extracted in multiple file extensions: JSON, CSV, CCSV, TSV, Raw Text. It is possible to do by clicking on the button Save in the top menu, see Figure 2:

After that, it is possible to depict it on the map using SpectX Map button:

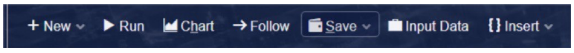


Figure 2. *Set of buttons.*

The map itself, see Figure 3:

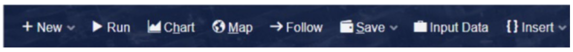


Figure 3. *Map button.*

According to the SpectX documentation page, during.CSV format analysis, there are different characteristics to consider, see Figure 4:

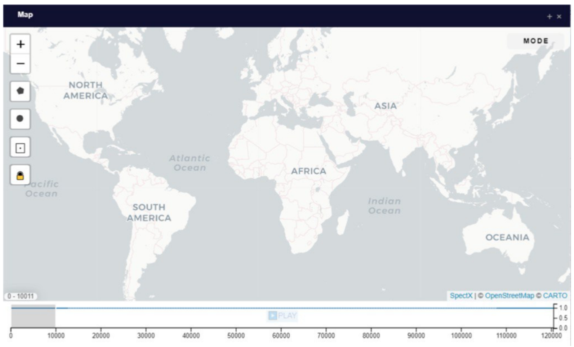


Figure 4. *Map.*

1. One line for each record;
2. Comma separated fields;
3. Space-characters adjacent to commas are ignored;
4. Fields with in-built commas are separated by double quote characters;
5. Fields with double quote characters must be surrounded by double quotes. Each inbuilt double quote must be represented by a pair of consecutive quotes;
6. Fields that contain inbuilt line-breaks must be surrounded by double quotes.

From the practice, most of the data scientists use .CSV extension. By doing a questionnaire among data science students, .CSV format is the most popular.

Graylog:

According to the Graylog wiki page, this is an incredible log parsing tool, which centrally captures and stores log analysis results using real-time search, by working with terabytes of data for any component in the whole infrastructure of any enterprise. It involves a three-tier architecture and scalable storage based on Elasticsearch. It is known as a solid alternative to Splunk. Graylog is a software company based in Houston, Texas.

Splunk:

According to the Splank wiki page, this is an American company based in San Francisco, California. It is a world-known log management tool. Most of the company's security experts use this tool for digital forensics and cybersecurity incidents management. Splunk captures and indexes real-time data in a centralized repository after which it generates graphs, reports, and different kinds of visualizations, so it makes easier to see what is going on in the system.

Plumbr:

According to the Plumbr wiki page, this is an Estonian software product company. Plumbr monitors Java-based applications for different kinds of errors, such as memory leaks, locked threads, etc. It consists of an agent and a portal. All the information is sent to the Plumbr Portal. On the Plumbr Portal, memory usage can be observed as well as lock contention durations and garbage collection pauses.

4.2 Comparison of existing visualization tools for data analysis and pattern recognition

WEKA:

The Waikato Environment for Knowledge Analysis (WEKA) was created in 1992 which is now a widely known data analysis tool. The WEKA Explorer is very easy to use and it provides pre-processing for different data manipulations such as classification, clustering, regression, etc. H. Kosorus et al. mentions that WEKA is very powerful as it supports comma-separated values and different kind of file formats (for example ARFF). Also, it supports URLs and databases.

RapidMiner:

According to H. Kosorus et al., RapidMiner5 was created in 2001, it is an open-source data mining tool and it consists of many operators for completing tasks related to data mining field. RapidMiner can work with different sources, namely databases and files. RapidMiner's GUI is very easy to use in comparison with WEKA. All RapidMiner sources were written in Java and are using the GNU Affero General Public License (AGPL).

Orange:

The orange platform was created in 1996, it is an open-source data mining platform and provides the user with data visualization and analysis by using Python scripting language. The orange platform is available for all popular platforms, including Windows, Mac OS, and variants of Linux. A comparison of tools and machine learning algorithms for data mining tasks has been provided.

Google Collaboratory by using Pandas framework:

In our fast-moving world, there are a lot of cloud solutions that are attractive because they provide hardware on the fly, which means there is no need for maintaining and configuring hardware resources. Many of the cloud platforms such as Amazon, Intel, Azure, and Google Cloud provide in a pay by-hour manner GPUs based applications and a runtime fully configured for Machine Learning and data analysis. Besides, according to T. C. Pessoa et al., NVIDIA offers standalone dockers with a pre-configured CUDA environment for deep learning. These frameworks are tuned, optimized, tested, and containerized.

In addition to the presented above, T. C. Pessoa et al. mentioned, that Google has created Collaboratory, a cloud service for spreading machine learning education and research. It provides up to 12GB of RAM. T. C. Pessoa et al. mentions that cloud service is fully configured with the leading artificial intelligence libraries and also offers a robust GPU. This Google service is linked to a Google Drive account, and it is free-of-charge. It is needed to upload the dataset to Google Drive and connect it with Google Collaboratory.

A new project was created in the Google Collaboratory environment. Besides, it is necessary to upload the dataset itself to Google Drive to connect it with Google Collaboratory.

Pandas is a python library that is useful for data analysis and statistics when it comes to huge datasets. Pandas were developed by an American software developer and businessman - Wes McKinney. It requires an environment in which basic scripts will be written and results will be depicted. It based on the Python language, thus it requires basic knowledge of Python language.

Pandas is the right tool for tabular data, such as data stored in spreadsheets or databases. In Pandas, a data table is called a DataFrame. Pandas library supports plenty of file formats or data sources out of the box (CSV, Excel, SQL, JSON, parquet). There are a few functions that help to import all of these file formats. These functions include prefix: read* and similarly to* prefix is used for storing the data. There are plenty of methods for slicing, selecting, and extracting the data which will be useful for data manipulations. To classify the data in the dataset GROUP BY function is used. It is possible to classify the data depending on the key which is specified in the query. Also, Pandas library provides one of the most convenient ways to plot the data, it is Matplotlib. It is possible to choose different types of plots. Data manipulations on the columns work elementwise, thus no need to all rows, it is possible to store the result in a new column and continue working with it.

One of the most important and useful things that it is easy to use and it handles big dataset which is in this case very important. Google Collaboratory in combination with the Pandas library is a good set of tools for doing basic statistics and dataset analysis. Google Collab-

Table 1. Tools Comparison

Characteristic	**RapidMiner**	**WEKA**	**Orange**	**R**	**Google Collaboratory**
Developer	RapidMiner, Germany	Univ. of Waikato, New Zealand	Univ. of Ljubljana, Slovenia	Worldwide development	Google
Programming language	Java	Java	C++, Python, Qt framework	C, Fortran, R	Focuses on supporting Python
License	Open.s., closed.s.	Open source, GNU GPL 3	Open source, GNU GPL 3	Free software, GNU GPL 2+	Free software ColabPro-paid GNU GPL 3
GUI/Command line	GUI	both	both	both	Online GUI
Community support	Very large	Very large	Large	Very large	Very large

oratory has been chosen as it is very easy to use and it is cloud-based. When some other tools were considered before choosing Google Collaboratory, there were cases when it was difficult to transport a csv file into it, WEKA is not .CSV friendly. RStudio was very glitchy with huge files.

4.3 Data analysis techniques

4.3.1 Exploratory Data Analysis

Exploratory Data Analysis (EDA) describes an approach to analyze different kinds of datasets and to summarize their characteristics by using visual methods mainly. This type of analysis is useful to conduct when it is needed to see what the data can tell beyond formal hypothesis testing tasks. In the current method, EDA is used from the beginning to explore the dataset.

4.3.2 Time Series approach

Time Series analysis is a series of data points indexed in time order. It unites methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the dataset. This approach is usually used for monitoring industrial processes or tracking corporate business metrics.
Usage:

1. Obtain an understanding of the underlying structure that produced the observed data;
2. Fit a model and proceed to produce forecasting and monitoring solutions.

A similar approach of dealing with the current dataset will be beneficial as a given dataset described in this thesis contains events related between each other and date/time information is a key aspect to find any anomalies. It will be helpful for constructing the methodology of the thesis.

4.3.3 Data grouping and data aggregation

According to O. Zaitsev, the simplest example of applying a GROUP BY (grouping) operator is to group the values of a series by the values of another one with the same size By using aggregates feature it is possible to estimate the statistical distribution of data that could be utilized to approximate the pattern in the set of data. For example, clustering very large databases are resource-intensive specifically in terms of memory and computation time. In addition, O. Zaitsev mentions that aggregation functions are the ones that take a series as an input and return a scalar value that summarizes the values of that series. These are the statistical functions: min, max, average, stde, also functions like sum, count, and many others. I this thesis, few of the data aggregation functions are used to describing a dataset, so to understand the volume of data and it's the time slot.

4.4 Company description

The company in which data were extracted from is the biggest Telecommunication company in Baltic states. The company is registered and operating in the Republic of Estonia, whose subsidiaries provide telecommunications services. Millions of users buy and use services from this company thus it is a great opportunity to analyze a unique dataset from pre-COVID-19 and post-COVID-19 time slots.

This company provides a huge amount of services which are used daily by all users in the country. Moreover, cellular data, TV, online shops are used daily by users as well.

Login data comes to the Graylog and further goes to the Hadoop ecosystem and HDFS from which it is possible to query and extract data by using different parsing tools such as SpectX.

There are other systems that are behind the UI system which are not used for this thesis dataset analysis. These systems contain different log structures and require access permissions.

Generally speaking, the company works in a fairly well-established manner. Services are provided to the customers and customers pay for these services. The criticality of the company for the country state makes systems more complicated and vital for country's digitalization and e-Governance.

The company also has been economically impacted by COVID-19, but as it is critical for the country to have the internet and network in order to get information about the current state of a pandemic situation as well as for working remotely, this company requires systems to be resilient against evolving threats. The company coordinates some amount of data according to Estonian police needs and involves separate processes for police requests. Thus, the data which is used in the current analysis is a part of the system processes which makes these data more vulnerable and important at the same time.

This level of criticality also requires a precise data analysis which will possibly be helpful for Estonian police future research within detecting any patterns related to anomalies.

4.5 Critical system contained data used by Estonian police

This point should be clarified, as the data which is extracted from the company's Big Data ecosystem, critically important for the criminal investigations conducted by Estonian police. When the case happened, Estonian police requests data about a particular person from a telecommunication company. The company has special processes to get this data and to transmit it to the police using HDFS queries. Many people involved in this procedure to keep the system alive. There is a need to detect any malicious activities within a particular system where anomaly detection techniques are exceptionally important. Police have a right to request the data but the police do not know the exact log structure every time it requests the data, thus some methodology from the company side will be useful. The method developed in this thesis can be potentially useful in the future to help police to investigate any malicious activities within a particular system. It is possible to establish a collaboration in the future, so some people from the company can provide some consultations for police by providing some documentation related to anomaly detection. The current method can be also included as a part of the research based on the real use-case example.

4.6 Dataset consists of information before and after COVID-19 lockdown

The current methodology is applied to the data which is taken during a very difficult period when the COVID-19 pandemic has happened. There is a part of data from pre-COVID-19 as well as post-COVID-19 time slots. Thus, it is possible to investigate a dataset during the critical time, when attackers use this opportunity to mislead users and to get access to critical and confidential information. According to the United States Computer Emergency Readiness Team, APT groups and cybercriminals are targeting individuals, small and medium enterprises, and large organizations with COVID-19-related scams and phishing emails. Thus, there are some possibilities of credential-based attacks to get access to user's accounts. Some competitors can be also interested in gaining information about users to send spam emails. There is a possibility to get involved in this fast-moving situation to protect the company's critical data.

4.7 System description

The system is intended for users who want to login to their private accounts related to services that they buy. When the user logins into the system, he/she can switch between other internal systems which is possible by design. Thus, there is no information in logs about into which system user tried to login (it is not described by the logging policy). There is only information about users who are trying to login into any of these systems, this user will be redirected to the system he/she tries to log in.

To get information into which system user tries to log in, an additional step is needed. To get in touch with application administrators of these systems and to ask information about each user in these system's logs. Due to confidentiality issues, access to these systems was not granted to conduct further investigation.

There are other systems that support it. This system is critical, thus users can log in by using their mobile numbers, emails, ID numbers. Besides, there is an opportunity to log in for internal users using their work credentials. After the user entered his/her credentials, he/she can only proceed after the identity check made by external ID authority.

Timestamps, session id, IP address, email, request status, type of the request, date, time are available for further analysis. This system allows login into different systems, the structure of these logs varies which makes anomaly detection even more difficult.

The specific of the systems also vary, there are some systems where a big quantity of failed logins is normal behavior, whereas in other systems in counts as an anomaly. This makes this system complicated to find anomalies.

The method described in this thesis describes one of the ways to do that.

Experiments were conducted on Apple MacBook Pro personal computer (Retina, 15-inch, Late 2013, 2 GHz Quad-Core Intel Core, i7, 8 GB 1600 MHz DDR3).

4.8 Dataset description and development of the method

The method consists of describing a step by step instructions of how the data was collected, sorted, and cleared from the unneeded information. So, the process of data preparation for further analysis will be described, thus it will be easier to find specific users which could be malicious.

Next step, suggestions will be made regarding the improvement of anomaly detection methods when multiple logins from different locations found.

Besides, in this chapter, research regarding data analysis, statistics, and method definition is presented. The novelty of this research is that there is a dataset from company X, which is the dataset from the critical application. This dataset was collected during the COVID-19 lockdown has happened in Estonia. The data analysis is done regarding the original dataset. As an outcome of this research, a new method will be described. First of all, in this section, given dataset will be described and analyzed.

Several observations in the dataset: 955288 entries. This dataset describes the user's authentications in the system. These are already sorted out. The original dataset contains much more information. There are columns including:

1. Weekday – for understanding which day of the week contains interesting patterns so then it will be possible to corelate it logically;
2. Date – is useful for selecting users on the particular day of the week;
3. Time (hour, minute, second) – these fields will be particularly useful for checking timing of logins for defining patterns;
4. IP address – is useful for detecting multiple users who tries to login into one account;
5. Email – is useful for checking it against different databases with leaked emails and for checking user identity;
6. Event (Success or Fail) – is an essential information, it helps to understand the status of the login;
7. Geolocation is helpful for matching user with it's location.

There is an interesting point, that data could also contain phone or ID numbers, which users use in Estonia to login into the system. Thus, they can be identified by their: email, ID, phone number. As was mentioned before, the goal of this thesis to conduct an analysis of the dataset that was taken from before and after COVID-19 lockdown has happened in Estonia. This case will be particularly useful for investigating criminal cases by the Estonian police.

There is no defined statistical method of how to analyze such a big amount of data and to provide a detailed analysis of the potential malicious activities that could be present during a short period. A new method can approach a big amount of data, to clean the data to a human-readable structure. Therefore, it will be easier to have a look at the dataset and to understand which tools and techniques will be useful, to begin with.

Data cleaning is an important step toward choosing an appropriate tool and defining the correct method. Variety of tools and applications seek to achieve an accurate result but data cleaning should be considered first.

The tool which was found useful for extracting and parsing data from the Hadoop Big Data ecosystem is called SpectX.

4.9 Development of the method

General information about the dataset:

1. Data type information about dataset;
2. Names of the columns displaying;
3. List of the most frequently appeared IP addresses in the dataset;
4. Amount of days during which data were collected;
5. To calculate the quantity of successful logins in total (in numbers);
6. Plotting all successful and failed logins during the period of dataset collection;

7. To find an account with biggest amount of IP addresses tried to login into it;
8. To find how many different IP addresses successfully logged in into the system;
9. To find how many different IP addresses failed to log in into the system;
10. To find how many unique emails in the dataset;
11. To find how many unique IP addresses in the dataset.

Method consist of the following steps to accomplish:

1. Log collection related to chosen system. Feature selection;
2. Select logins from the beginning of January 2020 till the 20th of April 2020;
3. To select all failed and successful logins and unite them in one dataset. Dataset has been divided into two classes: failed logins and successful logins and united into one general class (which excludes outliers);
4. To filter out “Duplicate password submission” attempts. These kinds of attempts could be created due to the user’s device failures because of the old password saved on the old email application. It will lower down the false alarm rate;
5. Geolocation of the users;
6. Calculation of how many different IP addresses successfully logged in into one account;
7. To depict geolocation of the users (based on the IP addresses) on the map for visual analysis;
8. To create a list of users, who tried to log in to multiple accounts. Meaning, into which accounts each user tried to log in with the same IP address.

Possible ways to analyze dataset taking into account time slots before and after COVID-19 pandemic situation has happened:

1. To analyze a frequency of failed and successful logins day/night time;
2. To visualize mean, min, max, quantiles of failed and successful logins before and after lockdown has happened in Estonia;
3. To visualize a distribution of the amount of failed and successful login attempts on the weekly basis during the whole period of time;
4. To visualize a distribution of the amount of unique login attempts;
5. To visualize a distribution of the average amount of unique locations per user daily;
6. To visualize a correlation of the total number of attempts to a total number of locations during the whole period;
7. To visualize on the world’s map distribution of unique user’s locations before lockdown has happened in Estonia;
8. To visualize on the world’s map distribution of unique user’s locations after lockdown

has happened in Estonia;

9. To visualize a distribution of unique user's locations before and after lockdown on the scale of Europe (to take a closer look at the situation within Europe);
10. To visualize a distribution of the number of email addresses into which one IP address tried to login into (to see if there are any outliers) and to depict it on the world's map;
11. To visualize a distribution of outliers before and after lockdown has happened;

Output of the method:

1. Detailed analysis of the dataset taken from the time slot before and after COVID-19 pandemic situation has happened in Estonia;
2. Top 10 IP addresses into which the biggest amount of emails tried to log in before lockdown has happened;
3. Top 10 IP addresses into which the biggest amount of emails tried to log in after lockdown has happened;
4. Top 5 users with the highest amount of failed login attempts;
5. Top 5 users with the highest amount of successful login attempts.

4.9.1 Method description and data analysis by using SpectX

Log collection related to chosen system. Feature selection.

According to the use case described earlier and considering the type of the system used for data extraction, the following information is needed to be extracted from the logs:

1. Weekday;
2. Date;
3. Time (hour, minute, second);
4. IP address;
5. Email;
6. Event (Success or Fail);
7. Geolocation.

By analyzing the structure of the logs, needed keywords and names of the fields were detected from the original data sources. Next step, by using SpectX syntax, two separate queries were constructed for successful and failed login attempts.

Select logins from the 1st of January 2020 till the 20th of April 2020.

Failure:

With the help of next query, logs of the whole dataset time slot are extracted in the LIST[]. Pattern is parsed in the format:
format='yyyy-MM-ddTHH:mm:ss.SSSSSSZ'.

```
@query = LIST(['directory_of_files_from_hdfs_during_needed_time slot.*.gz'])|
parse(pattern:"LD ' ' TIMESTAMP(format='yyyy-MM-ddTHH:mm:ss.SSSSSSZ'):date
' ' LD0,5244:line (EOL|EOF)");
.select(line, date, year(date), month(date), DAY_OF_WEEK(date), day(date),
hour(date), minute(date), second(date), ip, email, "failure" as event)
```

After that, the query of selection is constructed:
IP address field, Email, username, the status of the request – contains the word "failed". And it is sorted by date. Then, year, month, day of the week (Tuesday, Wednesday, Thursday), date, email, IP addresses are extracted.

```
@query.select(date, parse("LD0,256000 'ip=
"' IPADDR:ip '
"' LD0,256000 EOF", line), parse("LD0,6400 '
"x-openam-username
": [' LD0,128 '
"' LD:email '
"' LD0,128 ']' LD0,256000 EOF", line), line).filter(ip is not NULL and line contains
'"url": " url_which_describes_authentication " and line contains " username " and line
contains 'is-error="1"').filter_out(line contains '"status": "Duplicate invalid password
submission"').sort(date)
```

Same it is done for successful attempts, but the word success was excluded from the select statement in order to get failed once:

```
@query = LIST(['directory_of_files_from_hdfs_during_needed_time slot.*.gz'])
| parse(pattern:"LD ' ' TIMESTAMP(format='yyyy-MM-ddTHH:mm:ss.SSSSSSZ'):date '
' LD0,5244:line (EOL|EOF)"); @query.select(date, parse("LD0,256000 'ip=
"' IPADDR:ip '
"' LD0,256000 EOF", line), parse("LD0,6400 '
" username
": [' LD0,128 '
"' LD:email '
"’ LD0,128 ’]’ LD0,256000 EOF", line), line).filter(ip is not NULL and line contains ’"url":
" url_which_describes_authentication "’ and line contains " username" and line contains ’is-
error="0"’).sort(date) .select(line, date, year(date), month(date), DAY_OF_WEEK(date),
day(date), hour(date), minute(date), second(date), ip, email, "success" as event)
```

In the last select, all needed fields are selected, but in the union selection, it is possible to add or to remove some fields and after that dataset will be changed. So, if it is needed to add some field which is predefined in main success/failed files, it is possible to do in the union select file.
The next step is to combine these two separate files into one separate file which will generate a new file with needed results. The idea behind it is that after the file which unites successful and failed logins into one can be run, it creates a separate file that can directly show needed information without running both queries again. It is not needed to run queries twice to get data. Once it was done, queried data are saved in a new file. To generate a new file, it is possible to add a new file name or to delete a previously generated file, so a new file will be created instead. And run that file, the data will be depicted immediately – without running it again. Besides, the geolocation parameter was included in a select statement. By writing:
@result.select(*, geopoint(ip)) – it is possible to add a geo point field to the results dataset.
Next step, is to describe the process of writing queries for detecting some patterns.

To select all failed and successful logins and unite it in one dataset.

Next query unites successful and failed logins that were found in the previous section:

```
@failures = @[failure.sx];
@successes = @[success.sx];
@success.unionall(@failures).sort(date)
.select(date, ip email, event, line).save(“Result2020.sxt”)
```

To filter out “Duplicate password submission” attempts.

As in the example where failed and successful logins were selected in the beginning, to check for the lines which are containing “Duplicate invalid password submission”:

```
@query.select(date, parse("LD0,256000 ’ip=
"’ IPADDR:ip ’
"’ LD0,256000 EOF", line), parse("LD0,6400 ’
"x-openam-username
": [’ LD0,128 ’
"’ LD:email ’
"’ LD0,128 ’]’ LD0,256000 EOF", line), line).filter(ip is not NULL and line contains
’"url": " url_which_describes_authentication " and line contains " username " and line
contains ’is-error="1"’).filter_out(line contains ’"status": "Duplicate invalid password
submission"’).sort(date)
```

Geolocation of the users.

The next step is to create a file that generates a new file where all the results will be stored including geopoints of the users. This can be done by using the following query:

```
@result = @[Result2020.sxt];
@result.select(*, geopoint(ip))
```

Result2020.sxt – is the file which will be created with all the results needed.

The next step, all the queries with Date, Time, IP address, Email, Event, Geopoint were generated by pressing the Run button.

Calculation of how many different IP addresses successfully logged in into the system.

Next step, a new “ipHotspots.sx” file was created:

```
@query = @[listResult.sx];
@query.filter(day(date) = 14).select(hour(date), ip, email)
.group(hour, ip, email).select(*, count(*), count(event=’success’) as successful_count)
.group(email, hour).sort(count desc)
```

In this query, all the data are coming from listResult.sx file. After that, query filter is used to sort this dataset and to get needed information out of it.
This query includes two types of counts:

1. With the help of successful count, the quantity of successful logins from the user are calculated;
2. With the help of count(event=’success’), new “count” column is created. This column calculates how many different IP addresses successfully logged in into the system.

To depict geolocation of the users (based on the IP addresses) on the map for visual analysis.

```
@query = @[listResult.sx];
@query.filter(day(date) = 14 and email = mario@gmail.com')
```

The results will be saved into listResult.sx – so not to waste a time to run find.sx file again. An example of an email is user@gmail.com. Any email could be inserted into this field. Besides, it is possible to choose the time when it is needed to see any activities. For example, day(date) = 14, takes an hour from 2 PM till 3 PM.
As a result, the list of geolocation and time will be depicted respectively.
Thus, it is possible to find users who logged in into the system from different locations at the same time. It is possible to get by selecting multiple IP addresses that tried to log in to the same account. There are also could be automated systems behind it.

To create a list of users, who tried to login into multiple accounts.

Another step is to see the number of logins in different accounts from the same user. I could be done by creating a separate folder with two files. The first file will count how many successful and failed logins users made when tried to login into different accounts as well as the amount of account will be calculated. It makes possible to analyze if one user tried to log in to different accounts. But it does not mean that it is a malicious user as there are such things as VPN, WIFI, roaming, and mobile network connection type.

File itself:

```
@result=@[../union/listResult.sx];
@result.select(ip, email, count(event="success") as successes, count(event="failure") as failures).group(ip, email)
.select(ip, count(email), sum(successes),
sum(failures)).group(ip).sort(sum_failures desc)
```

4.9.2 Google Collaboratory description and data analysis by using Pandas software library.

In this subsection, some of the basic scripts are presented. These scripts are easy to comprehend and to try it in the Google Colab environment. A step-by-step methodology was constructed based on the dataset structure.

By importing pandas, reading an uploaded file to Google Drive and depicting the first 10 rows:

Input 1:

```
import pandas as pd
file_name = ’/content/drive/My Drive/anomaly/covid_19_dataset.csv’
df = pd.read_csv(file_name, delimiter=’;’)
df.head(10)
```

To convert all dates to datetime format following script is used:

Input 2:

```
df["date"]=pd.to_datetime(df["date"])
df = df.set_index(’date’)
df.head()
```

Output 2:

Displays a shortlist of the first few lines from the dataset. Data type information about the dataset.

By using command df.info(), where df is a DataFrame in Google Collaboratory, data type information has been retrieved from the dataset. For this, the Pandas software library was used. It helps for a general understanding of what dataset consists of, which types of data are available to plan future analysis. Df – is a DataFrame object.

The following script depicts all the columns from the uploaded dataset (in total there are 8 columns):

Input 3:

df.info() Dataset consists of 2 data types in total. int64 means that these are integers(numbers) and objects. The number of columns – 8.

Input 4:

```
df.columns
```

Output 4:

```
Index([‘date’, ‘year’, ‘month’, ‘hour’, ‘minute’, ‘ip’, ‘email’, ‘event’], dtype=’object’)
```

Next script depicts the list of the most frequently appeared IP addresses in the dataset:

Input 5:

```
pd.set_option(‘display.max_rows’, None)
df[“ip”].value_counts().head(2)
```

The data is anonymized, thus there is no output for this command.

Output 5:

```
IP address IP address
```

Next script shows the first and the last day in the dataset:

Input 6:

```
df[“date”].min(), df[“date”].max()
```

Output 6:

```
(‘2020-01-01 00:00:00’, ‘2020-04-20 11:59:59’)
```

Amount of days during which data were collected.

Input 7:

```
df[“date”].max() – df[“date”].min()
```

Output 7:

```
Timedelta(110 days’)
```

Next script calculates the number of successful, failed logins and it’s data type:

Input 8:

```
pd.set_option(‘display.max_rows’, None) df[“event”].value_counts().head(10)
```

Output 8:

```
Number of successfully logged in users: 739982
Number of failed logged in users: 215306
```

Next script displays IP addresses that are the most frequent in the dataset and it’s data type:

Input 9:

```
pd.set_option(‘display.max_rows’, None)
df[“email”].value_counts().head(2)
```

Next step is to import Matplotlib for drawing plots:

Input 10:

```
import matplotlib.pyplot as plt
%matplotlib inline
events_df = df[‘event’]
events_df.hist()
plt.show()
```

To calculate the number of successful logins in total with the frequency – 1 hour. Create a plot.

Input 11:

```
success_df = df[df[‘event’] == ‘success’] success_hour_count_df = success_df[‘event’]
.resample(‘1h’).count()
success_hour_count_df.head()
success_hour_count_df.plot()
plt.show()
```

To calculate the number of failed logins in total with the frequency – 1 hour. Create a plot.

Input 12:

```
failure_df = df[df[‘event’] != ‘success’] failure_hour_count_df = failure_df[‘event’]
.resample(‘1h’).count()
failure_hour_count_df.head()
failure_hour_count_df.plot()
plt.show()
```

5. Chapter 5

5.1 Evaluation and Results

The method does not guarantee that all the anomalies will be found as well as that these anomalies 100% proven to be anomalies, but this method points out some unusual behavioral patterns of the most suspicious users according to the steps described by this method.

The current dataset consists of 110 days: from 1st of January until 20th of April.

Based on the general graph of successful and failed logins, it is clear that the number of successful logins is much bigger than the number of failed logins.

It consists of 955287 login attempts.

There are 5298193 unique users in this dataset.

Number of unique emails: 39849

Number of unique phone numbers and ID’s: 5258344

Number of unique IP addresses: 46822

Number of successfully logged in users: 739982

Number of failed logged in users: 215306

The pandemic situation has happened due to the COVID-19 virus spread from China all over the world. In Estonia, lockdown has happened on the 13th of March 2020.

First step will be to compare the number of successful and failed attempts during the whole period, see Figure 5:

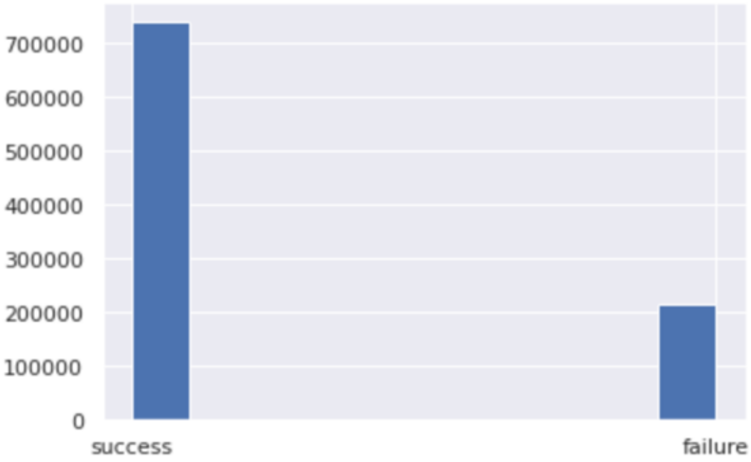


Figure 5. *Success/Failure difference.*

5.1.1 Pre-COVID-19 and post-COVID-19 analysis.

All the scripts which generate plots that will be described below are available in Appendix 1 - Code for the data visualization with Pandas and SpectX.

Failed login attempts distribution by hours:

Next plots show an average distribution difference between failed and successful logins during the whole period on the scale of 24 hours, see Figure 6 and Figure 7.

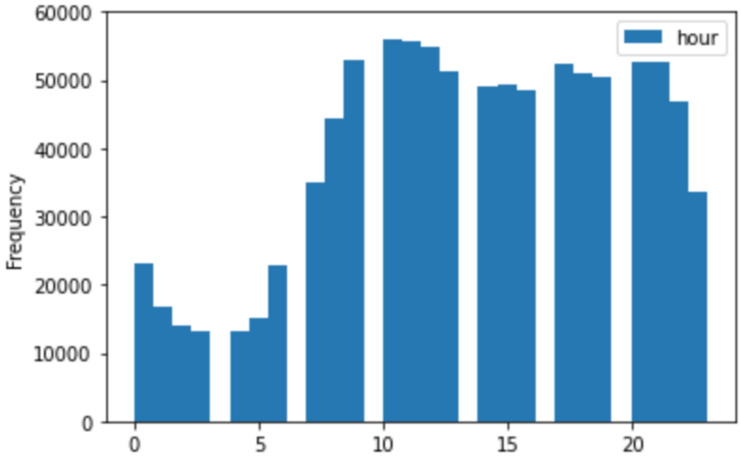


Figure 6. *Failed logins distribution during 24 hours.*

From Figure 6, it can be seen, that there was a decrease in failed login attempts between 00:00 AM and 05:00 AM. Between 03:00 AM and 04:00 AM there were not attempts at all. The peak was between 10:00 AM and 12:00 AM and in the evening between 20:00 PM and 22:00 PM.

Successful login attempts distribution by hours:

For Figure 7 it can be seen, that the number of successful logins was very high between 10 AM and 2 PM. The lowest number of successful logins happened to be between 12 AM and 6 AM.

The highest peak is between 10 AM and 1 PM. Then it goes down a bit and between 8 PM and 10 PM it slightly increases again. So, it is a good point to have a look at why these failed logins happened at night. And what are the users trying to log in many times (which is more than 10 000 times).

Day/Night login attempts before and after lockdown has happened in Estonia:

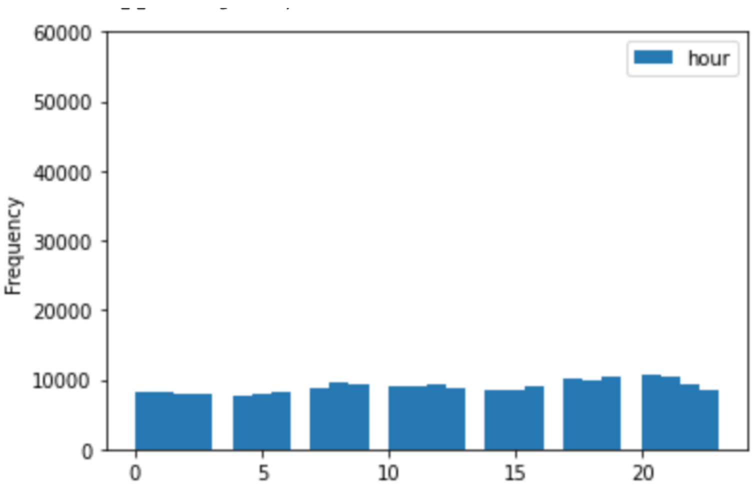


Figure 7. *Successful logins distribution during 24 hours.*

Another way to visualize data is to present it in the box plot format. The following Figure 8 shows a difference between day and night logins during normal time and after lockdown has happened:

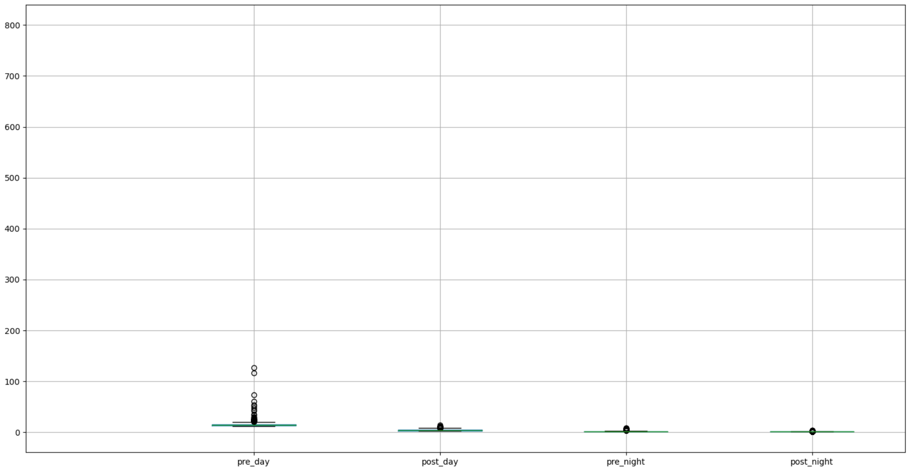


Figure 8. *Day/Night logins before/after lockdown.*

There are fewer outliers after lockdown, thus the number of locations increased (due to the fact, that people stayed at home and they had their home IP addresses) and outliers are legitimate (as these users were using public place WIFI, etc.). From the plot, it is obvious, that during lock down the number of logins at night decreased gradually and ended up almost in the same range as during normal day time.

Min, max, mean, quantiles of failed/successful login attempts:

Note: in this dataset, the 73rd day is the day when the lockdown was announced in Estonia. In total this dataset consists of 110 days.

Figure 9 shows the difference between the minimum, maximum, and mean number of successful and failed logins before and after the pandemic situation has happened.

In Figure 9 it is possible to see median or second quartile, first quartile, third quartile, whiskers, maximum, outliers.

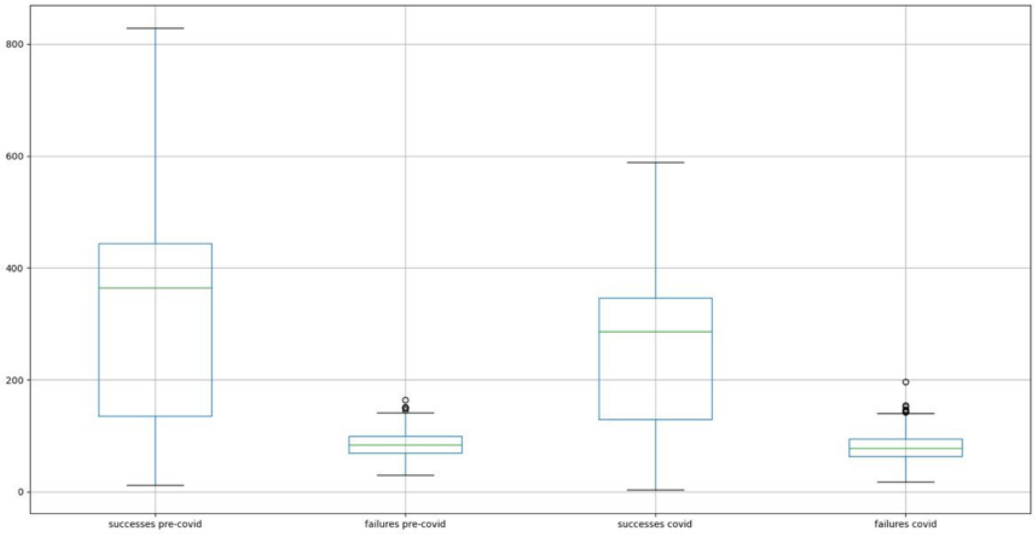


Figure 9. *Min, Max, Mean of successful/failed logins.*

It is obvious that during the normal time, from the 1st of January until the 13th of March, the average number of successful logins per day was higher than the average number of failed logins.

The minimum quantity of successful logins is lower than the minimum number of failed logins during a normal time (pre-COVID-19 time slot).

On the other hand, during pandemic times, starting from the 13th of March until the 20th of April, minimum, maximum, and mean number of successful login decreased, but the maximum number of failed attempts increased a bit.

If to compare pre-COVID-19 and post-COVID-19 timeslots, the number of successful logins decreased whereas the number of failed attempts increased a bit (with some obvious outliers). During the lockdown, users were logging in less, both successful and failed attempts, than during normal times.

If to compare Figure 8 and Figure 9, it is clear. that during the post-COVID-19 period, less people were using services and the number of successful login attempts decreased, but the number of failed login attempts increased slightly (with some outliers).

Distribution of the general number of login attempts daily:

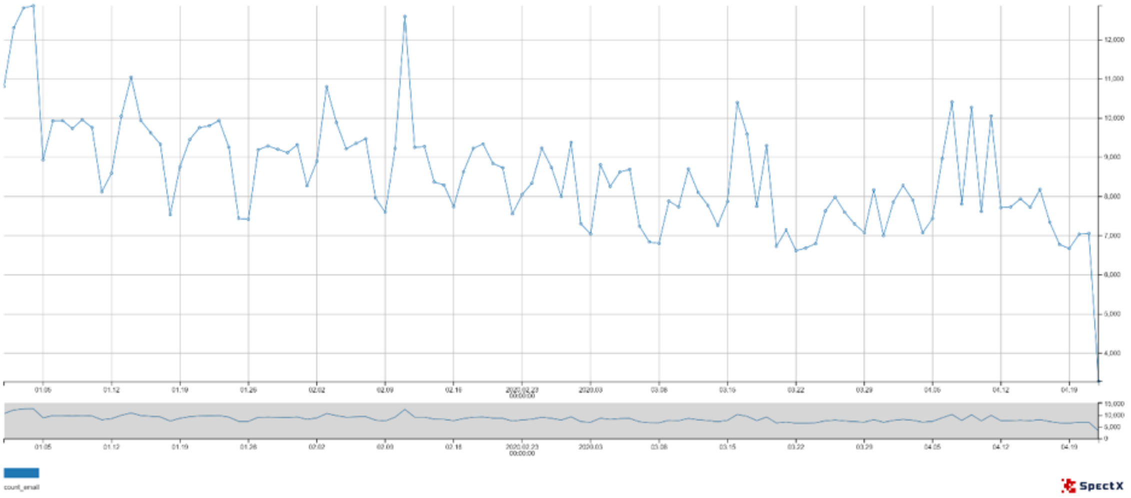


Figure 10. *Distribution of the general number of login attempts daily.*

From Figure 10, it can be seen the weekends when the number of attempts decreases followed by an increase by Monday. Besides, there is no sharp decrease or increase before and after the lockdown. There was a slight decrease, but the number of attempts was also high during the lockdown. To discover more in detail, the next graph with unique users will be depicted.

Distribution of the number of unique emails (users) using the system daily:

Figure 11 shows how many unique emails (users) using the system daily:

The number of unique emails decreases during the lockdown, as fewer users using the Company's services during the lockdown. The highest peak of the use of services happened on the 6th of January with approximately 3600 users. The number of users who use services started decreasing gradually from the beginning of March. During the lockdown, there was a very stable trend of user activity with approximately 1400-1500 users per day. From Figure 11, some weekends can be seen, when the number of user's activities decreases.

If to compare Figure 10 and Figure 11, there is an obvious notice, that the number of users who uses services - decreased during the lockdown, but these users were using the services

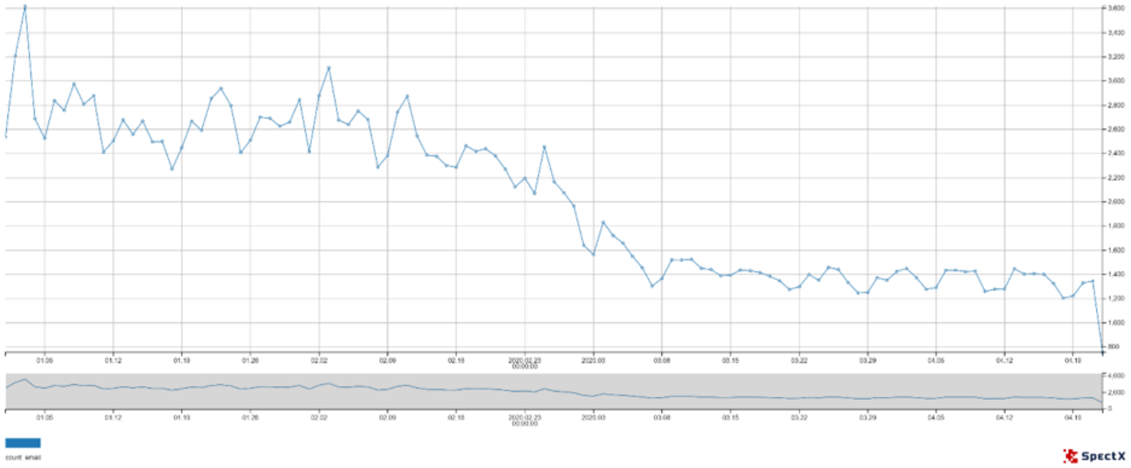


Figure 11. *Distribution of successful/failed logins on weekly basis.*

more frequently after lockdown has happened in Estonia.

Distribution of an average number of successful unique login attempts daily:

Figure 12 shows if an average number of user's unique successful logins changes from the middle of March (when lockdown happened):

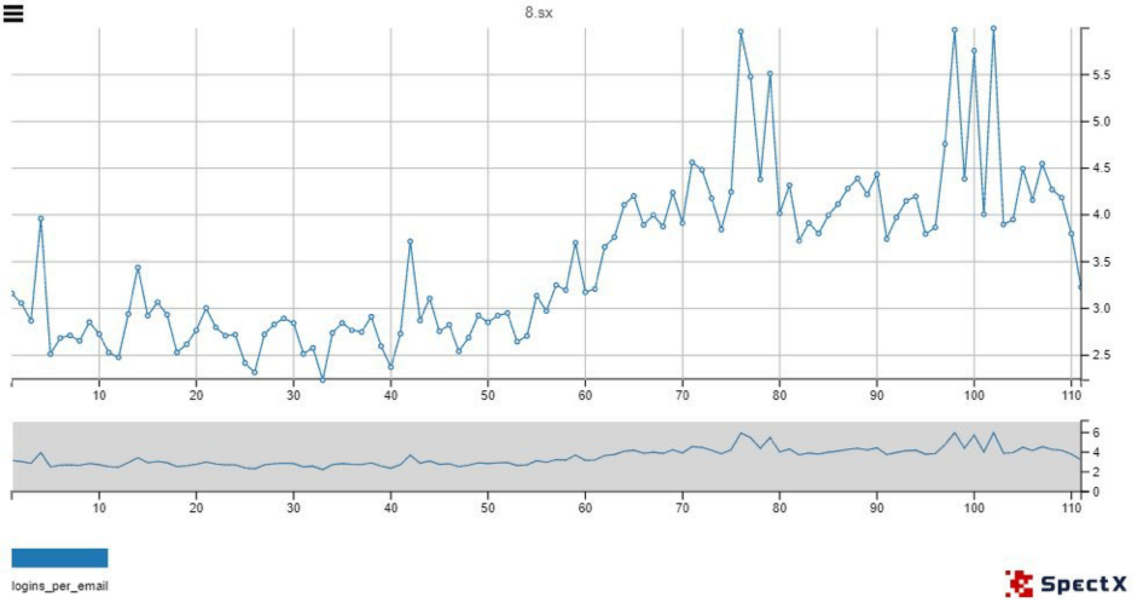


Figure 12. *Distribution of the number of successful unique logins.*

These are unique logins, meaning user which logged in from one location and one time counts into the calculations as a unique login attempt.

On the scale from 1 to 110, these are days from the 1st of January until the 20th of April which is the volume of the dataset.

The lockdown in Estonia has happened on the 73rd day.

It is clear from the graph, that users were logging in successfully on average more times after the 13th of March.

The first peak was on the 76th day, the average number of logins 5.95, whereas during normal times, the highest peak was on the 4th day with 3.96 logins on average.

The lowest point after the number of logins started growing, was the 53rd day with 2.64 logins on average.

The most interesting point is that the day lockdown has happened was 73rd day counting from the 1st of January.

It is clear that on the 73rd day, the average number of logins was 4.18. On the 74th day it dropped down a bit and then started growing again between 75th and 76th days, a rapid increase happened, thus on the 76th day, the average number of successful logins was 5.96.

Distribution of an average number of failed unique login attempts daily:

Figure 13 shows if an average number of user's unique failed logins changes from the middle of March (when lockdown happened):

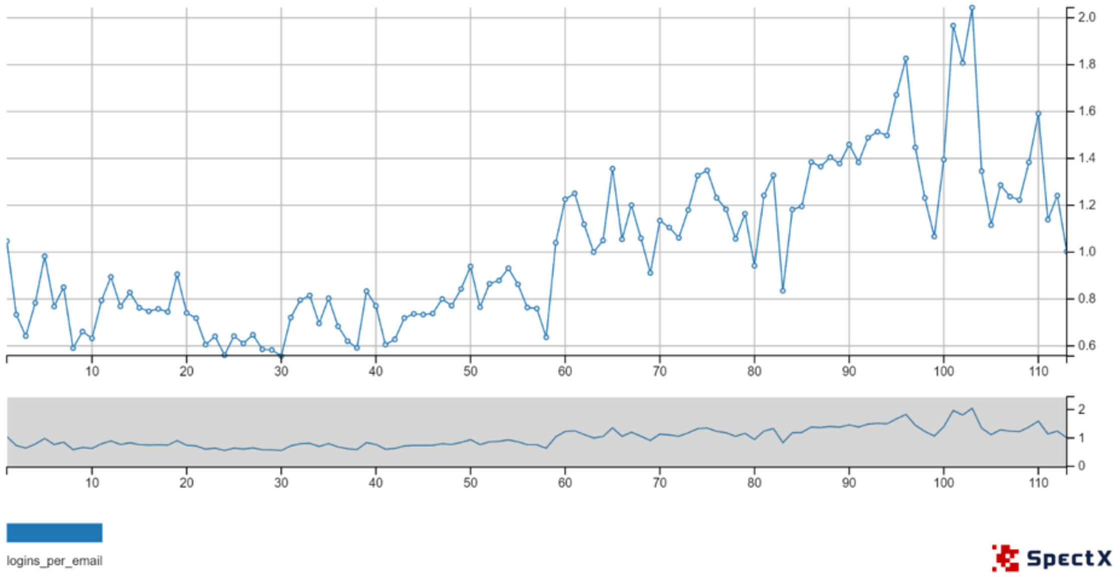


Figure 13. *Distribution of failed logins weekly.*

This graph shows the distribution of an average number of failed login attempts per email. According to this graph, users use the Company's services more often during the lockdown. In contrast to the successful logins graph, this graph shows an interesting trend on the 75th day when the lockdown has happened in Estonia. During the week when lockdown has happened the peak of the average number of failed logins per user was on the 75th day with 1.35. If to compare Figure 12 and Figure 13, an average number of unique failed login attempts is lower than an average number of successful logins. Maximum average number of unique successful logins daily - 6.0 whereas a maximum average number of unique failed login attempts daily - 2.1. It is clear, that an average number of unique login attempts increases. It means that users started working from separate locations (their homes) more often during the lockdown.

Distribution of the total number of locations weekly:

Figure 14 shows the total number of locations weekly:

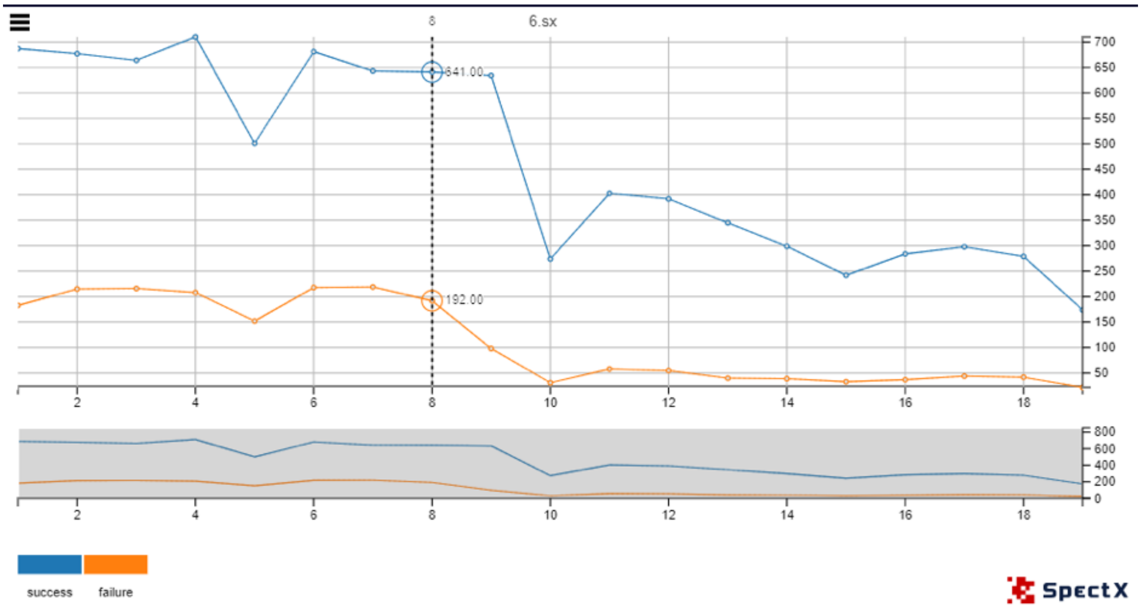


Figure 14. *Distribution of successful/failed logins weekly.*

Based on this and previous graphs, fewer users are using Company's services during lockdown but these users using services more frequently than before lockdown. Figure 14 shows exactly how the number of users decreases gradually during the lockdown situation in Estonia. There are a few downs, the number of locations dropped down from 710 successful logins and 208 failed attempts on the 4th week down to 501 successful logins and 152 failed attempts on the 5th week. The second major decrease has happened from 634 successful logins and 98 failed attempts on the 9th week down to 274 successful logins and 31 failed attempts on the 10th week. Exactly during the 10th-week lockdown has happened in Estonia, thus the number of locations decreased gradually.
Generally speaking, Figure 14 proves, that the number of user's unique locations decreased during the lockdown. And based on the previous Figures and plots, users became more active during the lockdown.

Distribution of an average number of unique locations per user daily:

Figure 15 shows an average number of unique locations per user per day. X is before and after pandemic times. Y is an average number of unique locations per user daily.

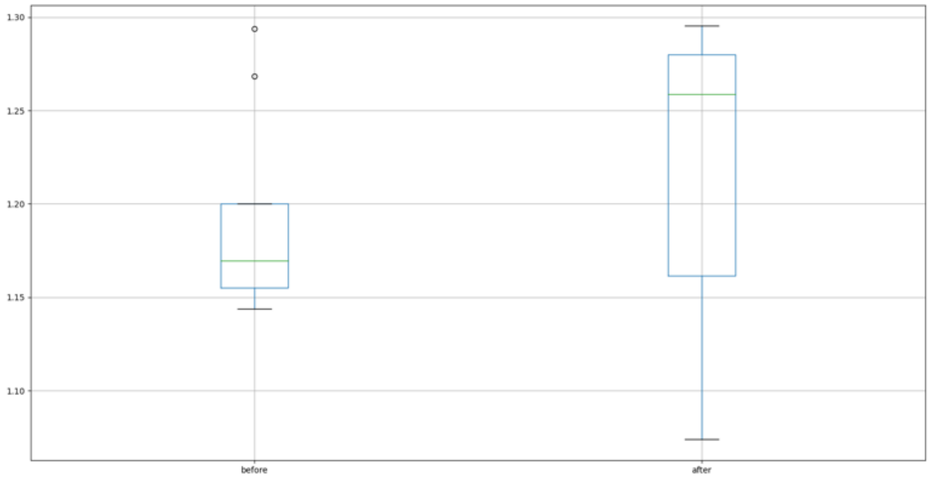


Figure 15. *Distribution of an average number of unique locations per user daily.*

From the plot, it is clear, that an average number of unique locations during pandemic times increased whereas it is lower during normal times. It is a strange deviation as during pandemic times people should travel less and the number of unique locations should be lower than during normal time. It also means, that people log in from home more often, and there are more unique locations as everyone stays at home. It should not necessarily mean that they are traveling more.

There are some outliers on the plot during normal times. An outlier reaches almost 1.3. Mean during normal times concentrates at 1.17 whereas during pandemic it reaches 1.26. The minimum is much lower during pandemic times than during normal times. Figure 15 proves that the number of unique locations increased after the lockdown has happened in Estonia.

Correlation of the total number of attempts to a total number of locations during the whole period:

A scatter plot on the Figure 16 shows a correlation of the total number of attempts to a total number of locations. It is clear from the Figure 16, that with an increasing number of locations, the number of login attempts also increases. The X-axis is how many logins attempts users have, Y-axis is from how many locations.

It can also be seen, that there is one obvious outlier. This plot is created to show the relationship between data: successful/failed logins and locations. This scatter plot proves, that failed login attempts were conducted from fewer locations (very dense) whereas successful logins were conducted from a higher number of locations with a higher number of outliers.

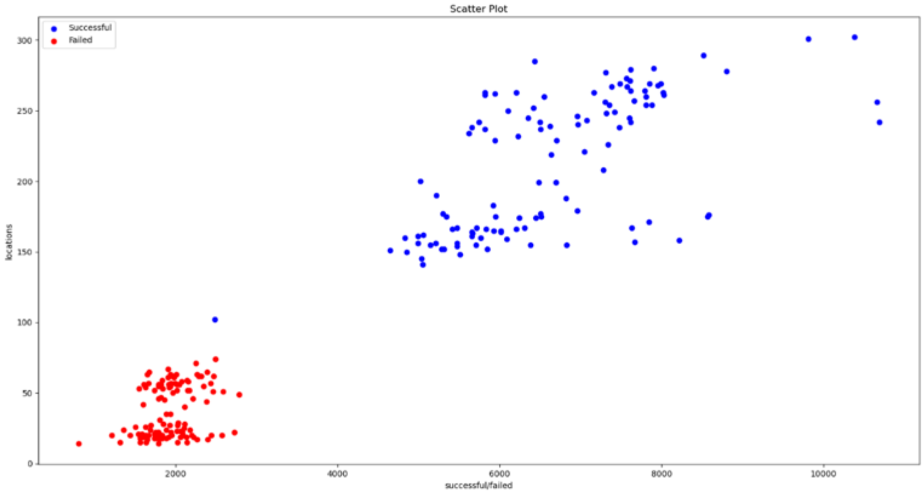


Figure 16. *Correlation of the total number of attempts to a total number of locations.*

If to compare Figure 16 and Figure 9, then these outliers or maximum number can be visible from both Figures.

Distribution of unique user's locations before lockdown has happened in Estonia:

Figure 17 shows the distribution of unique locations. The situation before the lockdown has happened in Estonia was following (before 13th of March):

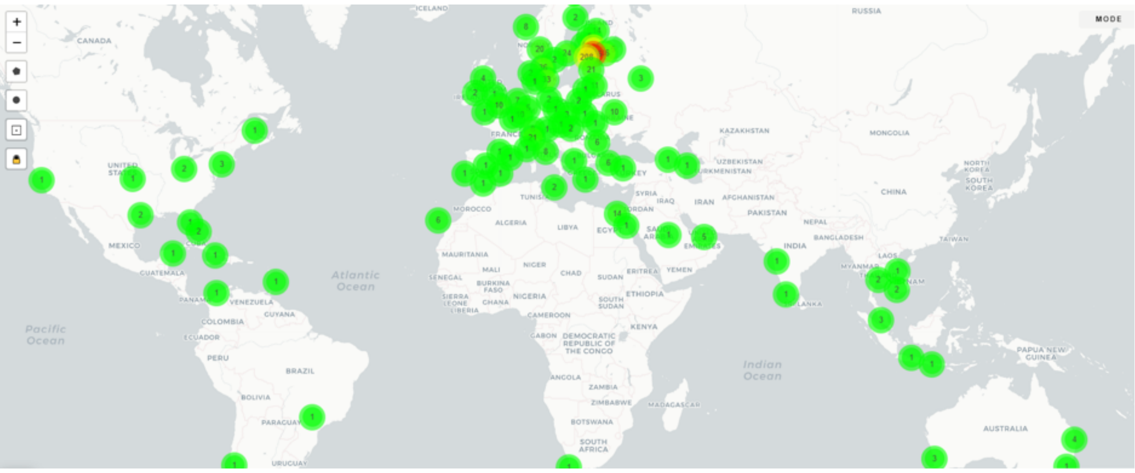


Figure 17. *Distribution of unique locations before lockdown.*

Distribution of unique locations after lockdown has happened in Estonia:

Figure 18 shows the distribution of unique locations. The situation after the lockdown has happened in Estonia was following (after 13th of March):

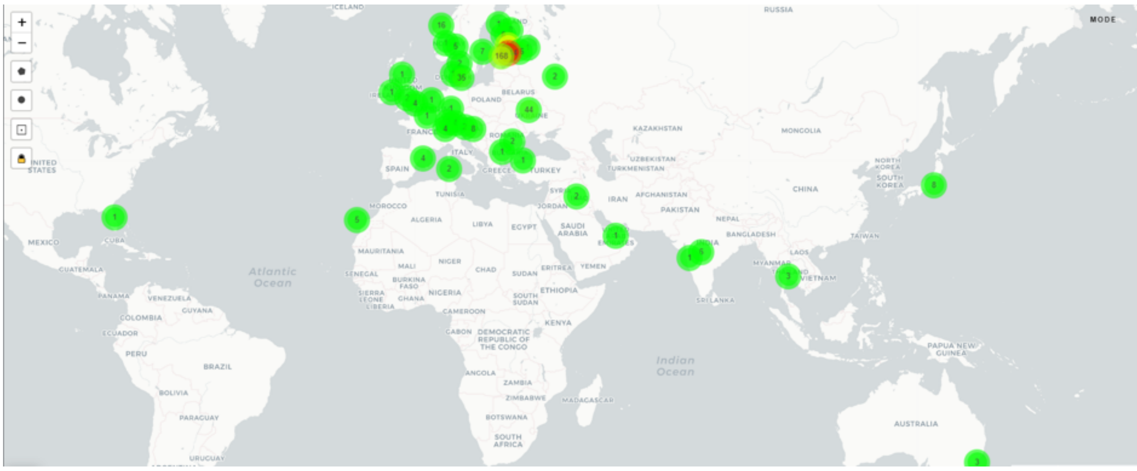


Figure 18. *Distribution of unique locations after lockdown.*

These are unique IP addresses. If to compare Figure 18 and Figure 17, it is clear, that the situation has changed and fewer travels were conducted during a lockdown. Nevertheless, there are some interesting points on the map to consider.
Few connections from Iraq have happened. These were not noticed during normal times.

Distribution of unique user's locations before and after the lockdown on the scale of Europe:

Figure 19 shows the distribution of unique user's locations before and after the lockdown on the scale of Europe:

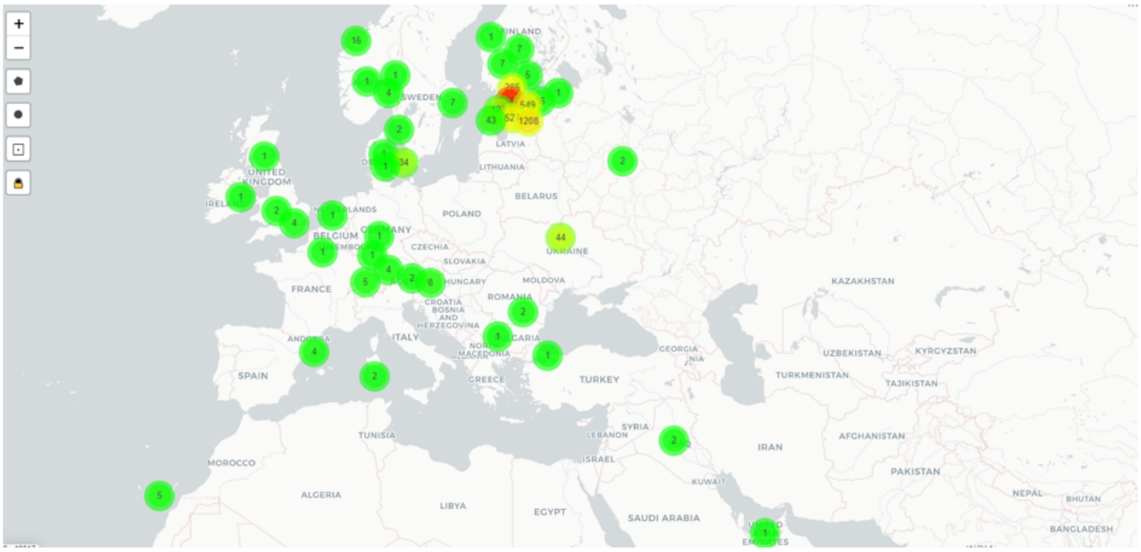


Figure 19. *Distribution of unique locations before lockdown on the Europe scale.*

Some people stayed in remote locations and some were already there when an emergency has happened. But the general picture has changed, as fewer people happened to be close to the USA during pandemic times. The following plot shows IP addresses that logged in into different emails. Some of these might be malicious. Some can be leaked accounts.

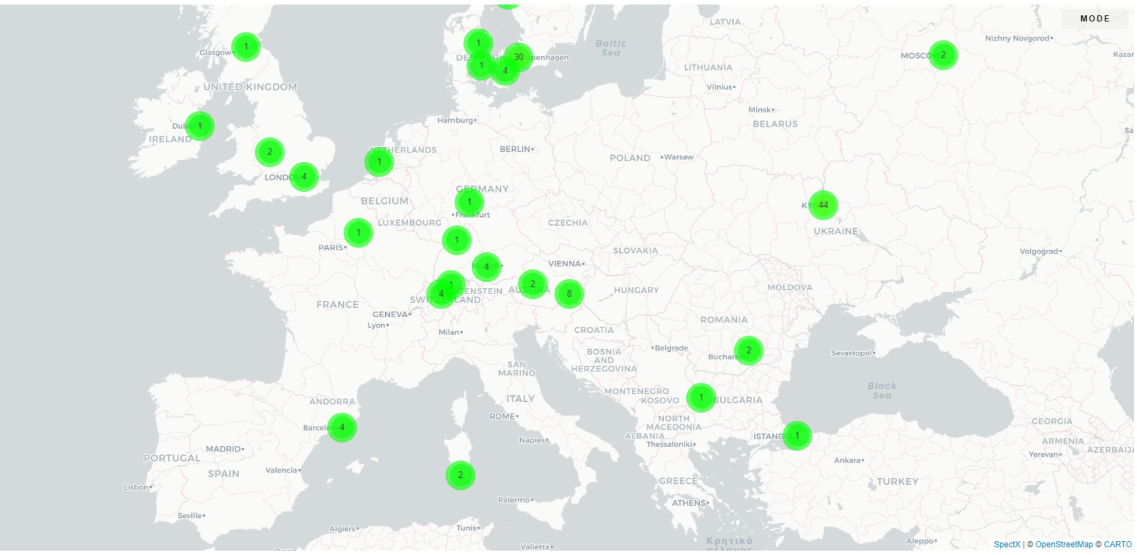


Figure 20. *Distribution of unique locations after lockdown on the Europe scale.*

It can be seen from Figure 17 after during lockdown there were fewer locations around the world.

Plot describes IP addresses by counting into how many accounts users logged in from these IP addresses:

Figure 21 depicts the number of emails (Y-axes) into which particular IP addresses tried to login in. The bigger the number (Y-axes), into the more emails one IP address tried to login into. For example, one IP address was used to login into 124 email accounts, that is an outlier. This plot uses the dataset during the whole period.

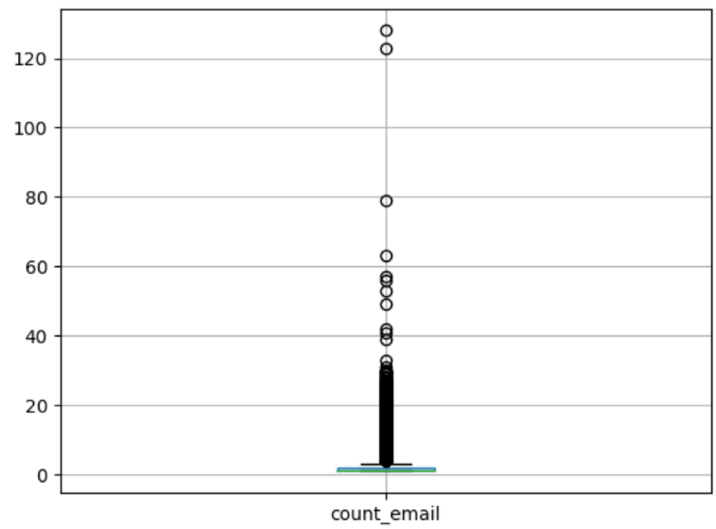


Figure 21. *Distribution of the number of emails with one IP address tried to login into.*

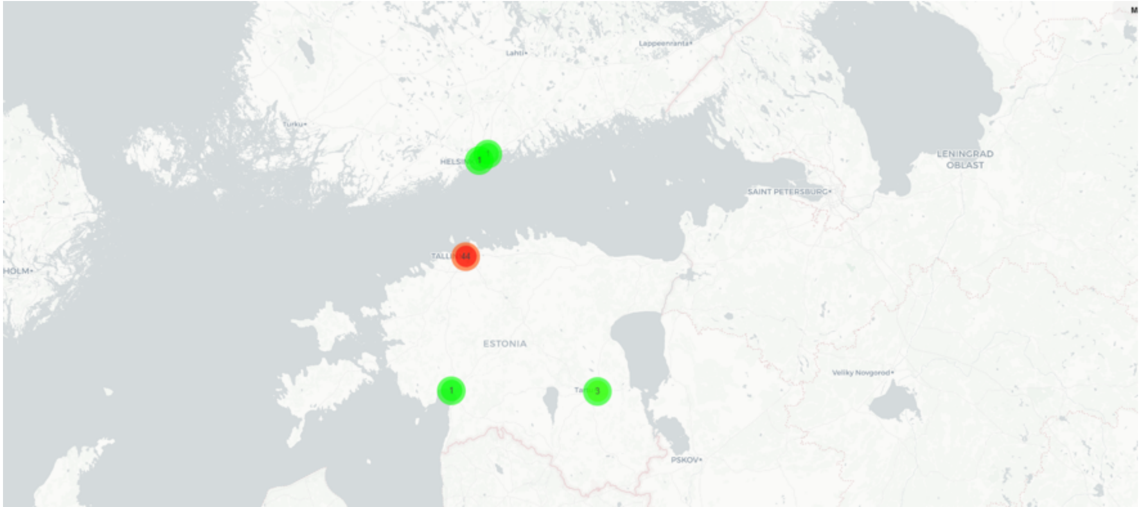


Figure 22. *Map of the first 50 outlier geolocations.*

Figure 22 shows Top 50 IP addresses which were taken from the previous plot results. The following map shows the first 50 outliers on the map. Previous Figure 21 shows this outlier distribution on the plot. These are almost all located in Estonia, except a few from Helsinki (Finland).

Distribution of outliers before and after lockdown has happened:

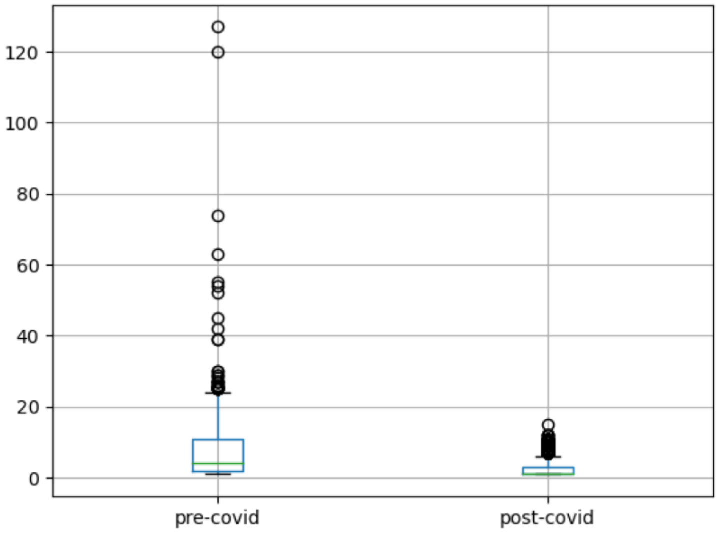


Figure 23. *Distribution of outliers before and after the lockdown has happened in Estonia.*

Figure 23 shows the same information as Figure 21 but Figure 23 compares two time slots before and after the lockdown has happened in Estonia in 2020.

As can be seen from Figure 23, the number of outliers was much bigger during pre-COVID-19 times.

(Y-axes) is the number of emails into which particular IP addresses tried to login in. The bigger the number (Y-axes), into the more emails one IP address tried to login into.

Based on the analysis conducted before, the next step would be to create a list of Top 10 users accounts into which the biggest number of IP addresses tried to login into.

These are the following steps to conduct:

1. Analyze Top 10 IP addresses into which the biggest number of emails tried to log in before lockdown has happened;
2. Analyze Top 10 IP addresses into which the biggest number of emails tried to log in after lockdown has happened;
3. Analyze Top 5 users with the highest number of failed login attempts;
4. Analyze Top 5 users with the highest number of successful login attempts.

5.1.2 List of Top 10 most common IP addresses before and after lockdown has happened.

For example, here is the list of Top 10 most common IP addresses (from these IP addresses, there was the highest number of logins into unique emails) before the lockdown has happened (all these IP addresses are anonymized):

1. IP User 1 (Tried to login into 127 accounts, successes – 245, failures - 7)
2. IP User 2 (Tried to login into 120 accounts, successes – 425, failures - 9)
3. IP User 3 (Tried to login into 74 accounts, successes – 252, failures - 6)
4. IP User 4 (Tried to login into 63 accounts, successes – 383, failures - 2)
5. IP User 5 (Tried to login into 55 accounts, successes – 182, failures - 0)
6. IP User 6 (Tried to login into 54 accounts, successes – 87, failures - 30)
7. IP User 7 (Tried to login into 52 accounts, successes – 142, failures - 120
8. IP User 8 (Tried to login into 45 accounts, successes – 130, failures - 0)
9. IP User 9 (Tried to login into 42 accounts, successes – 113, failures - 1)
10. IP User 10 (Tried to login into 39 accounts, successes – 88, failures - 0)

Next list, is the list of Top 10 most common IP addresses (from these IP addresses, there was the highest number of logins into unique emails) after the lockdown has happened (all these IP addresses are anonymized):

1. IP User 1 (Tried to login into 15 accounts, successes – 38, failures - 12)
2. IP User 2 (Tried to login into 12 accounts, successes – 27, failures - 3)
3. IP User 3 (Tried to login into 12 accounts, successes – 21, failures - 11)
4. IP User 4 (Tried to login into 12 accounts, successes – 47, failures - 1)
5. IP User 5 (Tried to login into 12 accounts, successes – 110, failures - 0)
6. IP User 6 (Tried to login into 11 accounts, successes – 38, failures - 0)
7. IP User 7 (Tried to login into 11 accounts, successes – 51, failures - 0)
8. IP User 8 (Tried to login into 11 accounts, successes – 22, failures - 0)
9. IP User 9 (Tried to login into 11 accounts, successes – 127, failures - 0)
10. IP User 10 (Tried to login into 11 accounts, successes – 20, failures - 0)

The number of successful and failed logins is also very different between before and after the lockdown, the period has happened in Estonia due to not an equal number of days taken into the data set. The pre-COVID-19 period contains 73 days whereas the post-COVID-19 period contains 37 days. But still, this number of information is enough for data analysis.

5.1.3 Plots of the Top 10 (pre-COVID-19) IP addresses during the whole time slot.

Blue are successful logins and orange are failed login attempts.

1. **IP User 1**

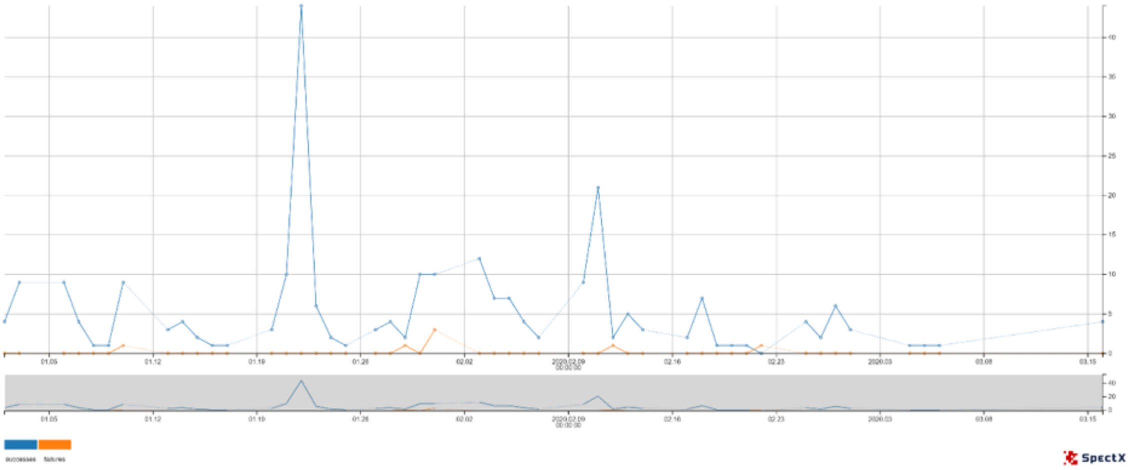


Figure 24. *User 1 before lockdown.*

As can be seen from Figure 24, this IP address shows high activity on the 22nd of January with 44 successful logins, which is the peak during the whole period. The second peak happened to be on the 11th of February with 21 successful logins. After the 11th of February, login activity goes down.

There were some single failed login attempts, but these look normal. The peak of the failed login attempts happened to be on the 31st of January with 3 attempts. In total there are 7 failed attempts during the whole time slot. The trend here is that the number of login attempts went down after COVID-19 lockdown has happened.

2. **IP User 2**

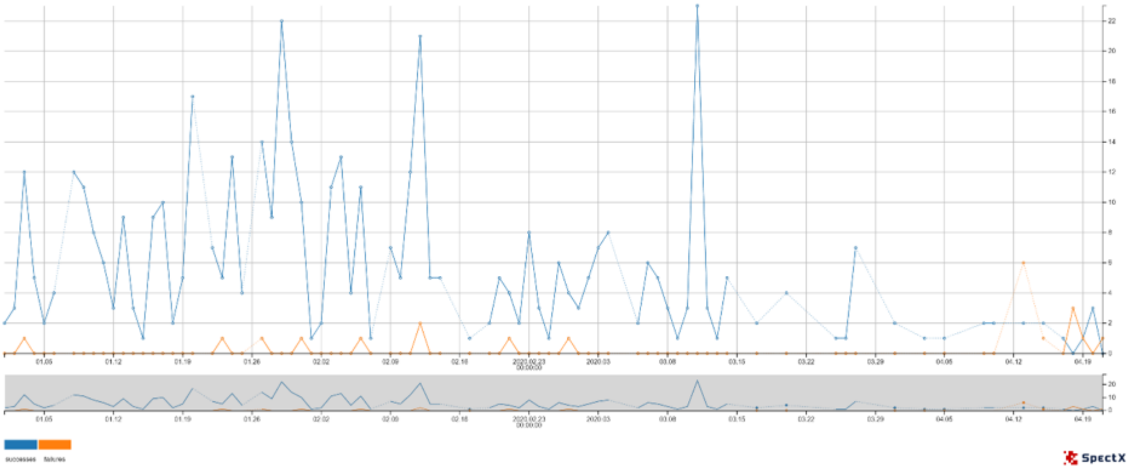


Figure 25. *User 2 before lockdown.*

As can be seen from Figure 25, this IP address followed a different trend. There were more peaks of successful logins during normal times. There were three major peaks during normal times, on the 20th of January, the 29th of January, the 12th of February with 17, 22, and 21 successful logins respectively. The interval between the peaks was 4-6 days. Just before the lockdown was announced, there was a peak of the whole period on the 11th of March with 23 successful logins. The peak of the failed attempts happened to be on the 13th of April with 6 failed login attempts.

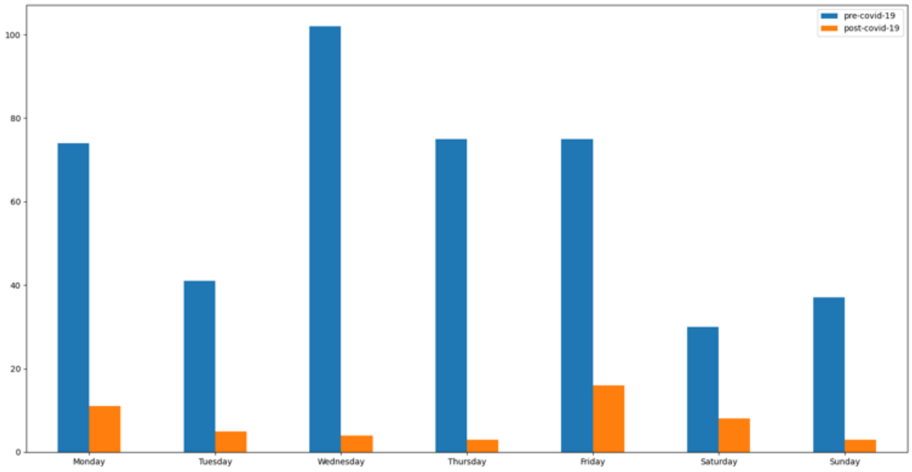


Figure 26. *User 2 before/after lockdown.*

Figure 26 depicts the general distribution of the logins from this IP address split into pre-COVID-19 and post-COVID-19 time slots daily. The situation has changed completely after lockdown has happened in Estonia. Less number of activities were noticed. Wednesday was the most active day during normal times whereas, during the lockdown, Friday appeared to be the most active day. Wednesday has shown the most dramatic change during the whole period.

3. **IP User 3**

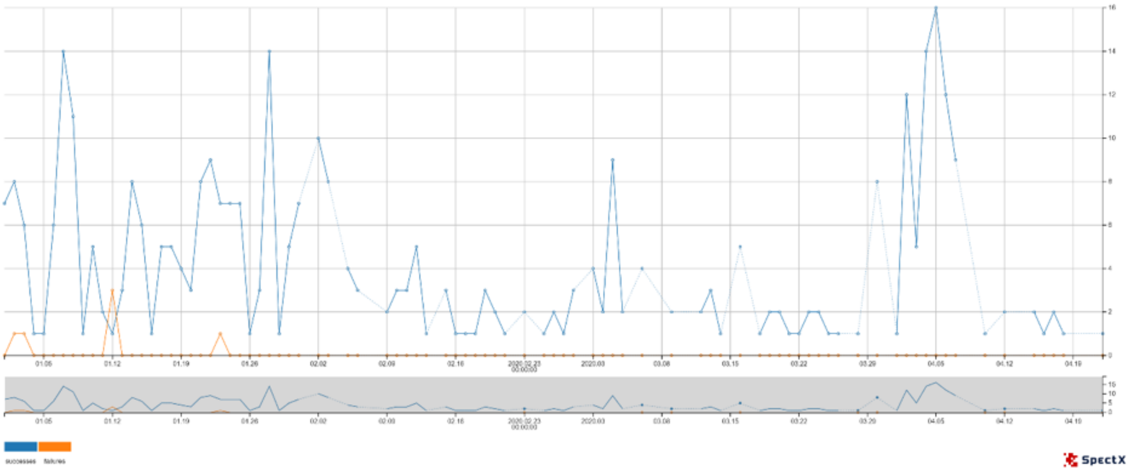


Figure 27. *User 3 before lockdown.*

Figure 27 shows, that the first peak happened to be on the 2nd of January with 8 successful logins and 1 failed attempt which followed by a slight decrease. On the 7th of January, there was a sharp increase up to 14th successful logins followed by a slight decrease again. On the 12th of January, there was a peak of the failed attempts with 3 failed and 1 successful which was a peak during the whole period. The peak of successful logins happened on the 5th of April with 16 successful logins. Generally speaking, there were more fluctuations during normal times.

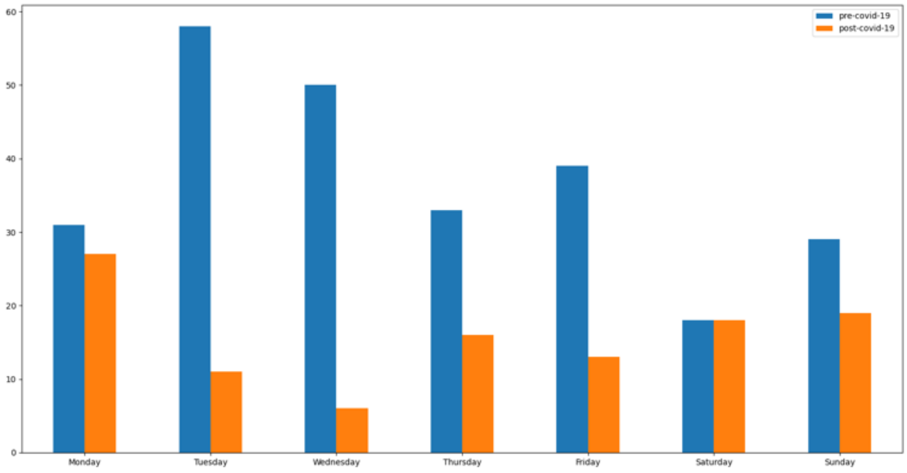


Figure 28. *User 3 before/after lockdown.*

Figure 28 depicts the general distribution of the logins from this IP address split into pre-COVID-19 and post-COVID-19 time slots daily. There was almost the same number of logins on Monday during pre-COVID-19 and post-COVID-19 timeslots and the same activity on Saturday. The peak during normal times was on Tuesday whereas peak during lockdown was on Monday.

4. **IP User 4**

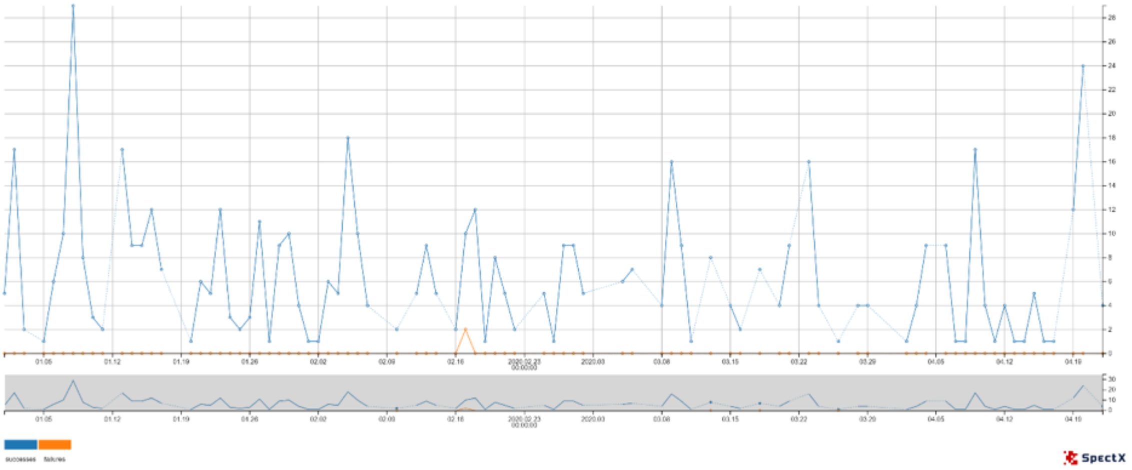


Figure 29. *User 4 before lockdown.*

Figure 29 shows, that the peak of the successful logins happened to be on the 8th of January with 29 attempts. The interval between peaks was around 3-6 days in average.

Generally speaking, this IP address shows a normal trend meaning there were not any suspicious number failed attempts after a big number of successful.

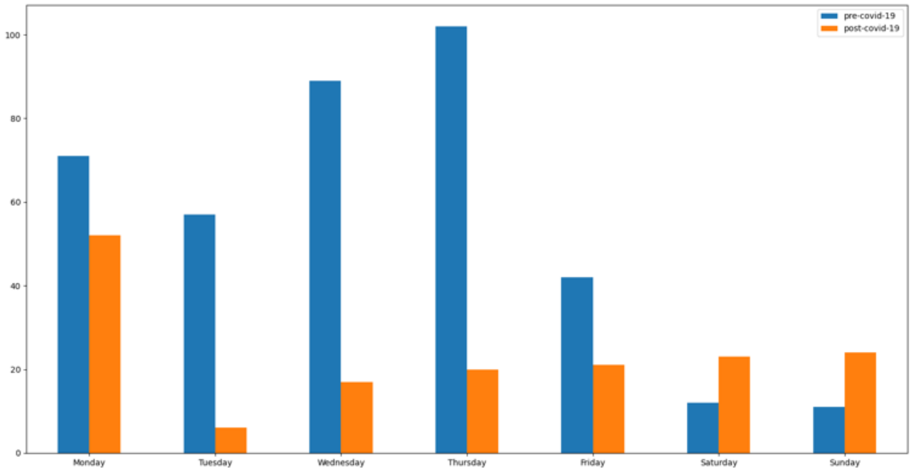


Figure 30. *User 4 before/after lockdown.*

From Figure 30 it is clear, that the IP address was very active on Wednesday and

Thursday during normal times whereas the most active day during lockdown was Monday.

5. **IP User 5**

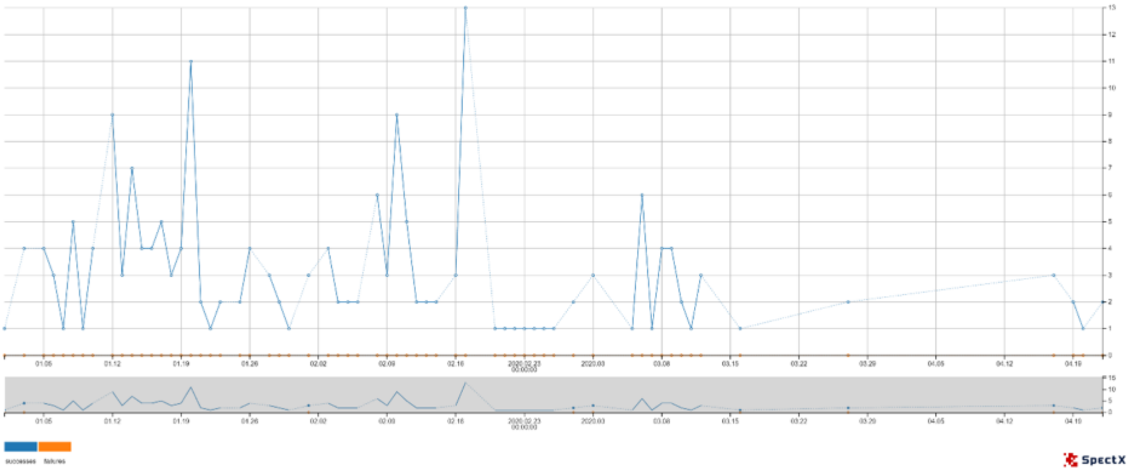


Figure 31. *User 5 before lockdown.*

Figure 31 shows, that this IP address shows normal user behavior almost during the whole time slot. There was a peak of the whole time slot on the 17th of February with 13th successful logins. Besides, there was a huge break between logins, from the 27th of March until the 17th of April.

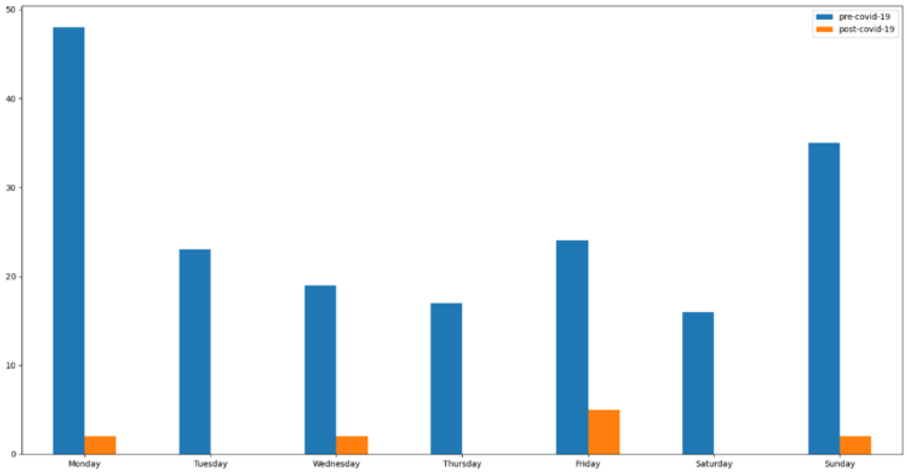


Figure 32. *User 5 before/after lockdown.*

From Figure 32 it is clear that there was a sharp decrease in login activities during the lockdown. Monday was a peak of the whole period during pre-COVID-19 and post-COVID-19 time slots.

6. **IP User 6**

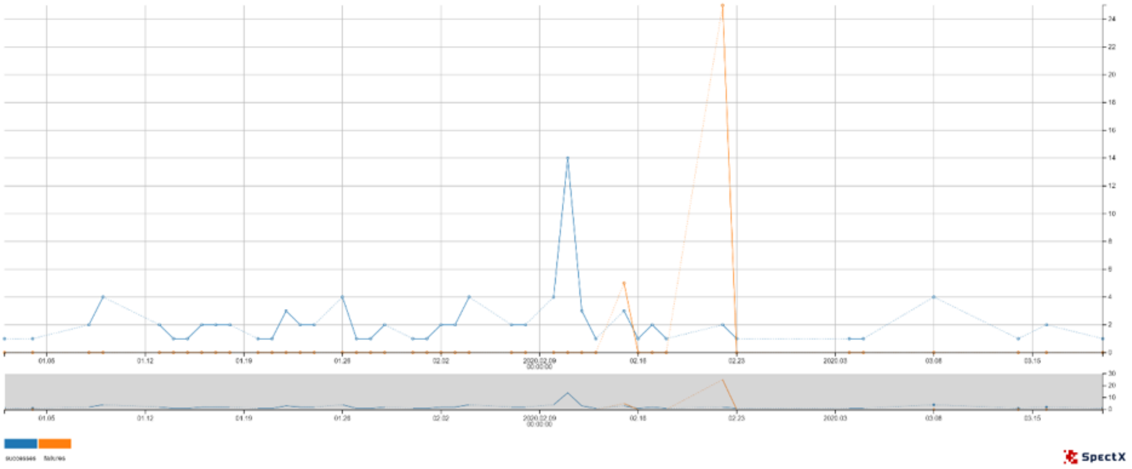


Figure 33. *User 6 before lockdown.*

It can be seen from Figure 33, that this IP address shows a bit different trend if compare with the situations described before. There was a high peak on the 11th of February with 14 successful logins after which a very suspicious event on the 22nd of February with 25 failed login attempts have happened. It has happened in the evening, at around 11 PM. After that, there were no failed logins at all. This is the only day during the whole time slot which contains so many failed logins. The other day was on the 15th of February with 5 failed attempts.

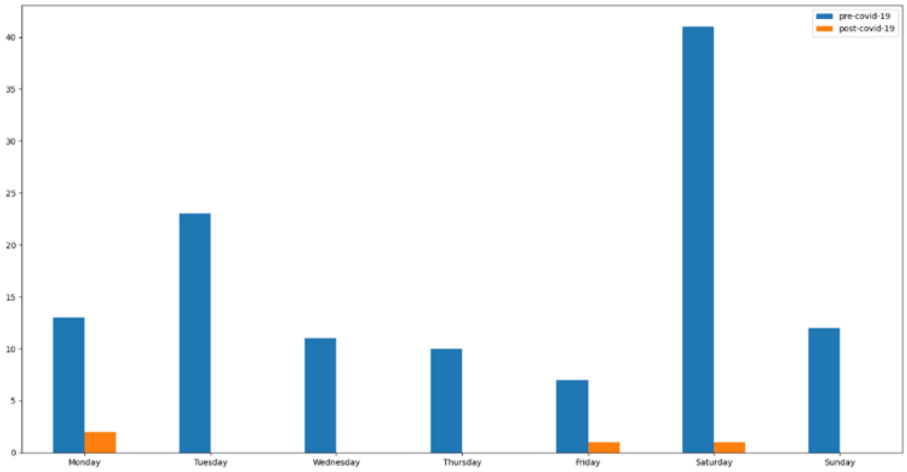


Figure 34. *User 6 before/after lockdown.*

From Figure 34 it is clear the peak of the whole period happened to be on Saturday.

7. **IP User 7**

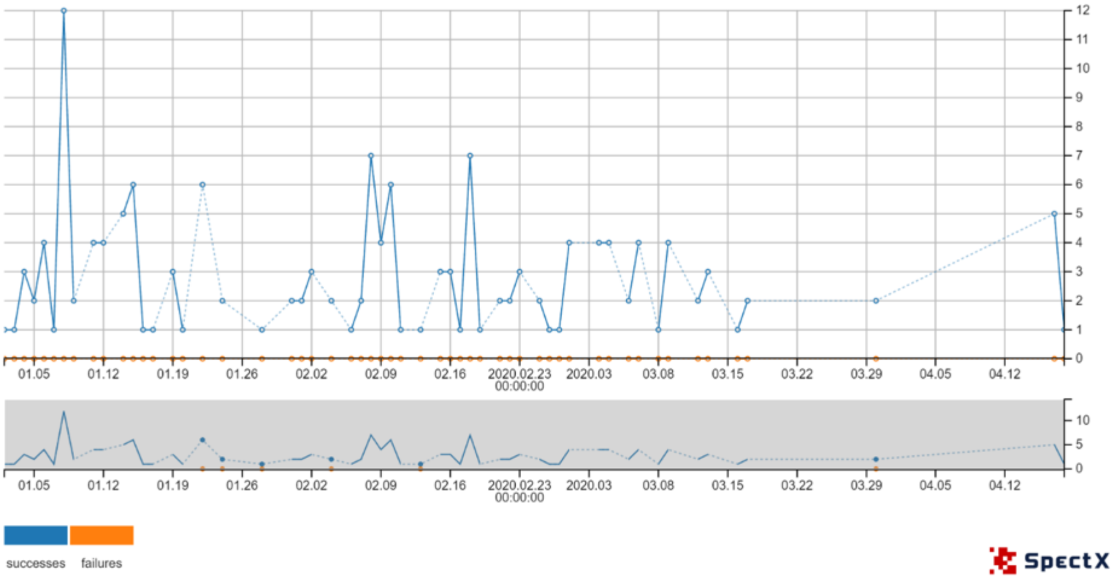


Figure 35. *User 7 before lockdown.*

In Figure 35 this IP address follows a normal trend almost during the whole time slot. Except, there were two time breaks between the 17th of March until the 30th of March and from the 30th of March until the 17th of April.

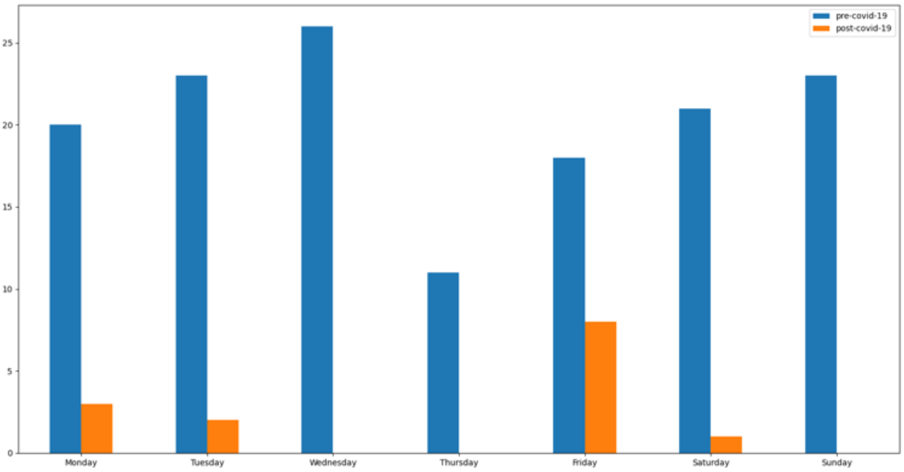


Figure 36. *User 7 before/after the lockdown.*

From Figure 36 it is clear that the peak of the whole period happened to be on Wednesday. Friday was the most active day during the lockdown.

8. **IP User 8**

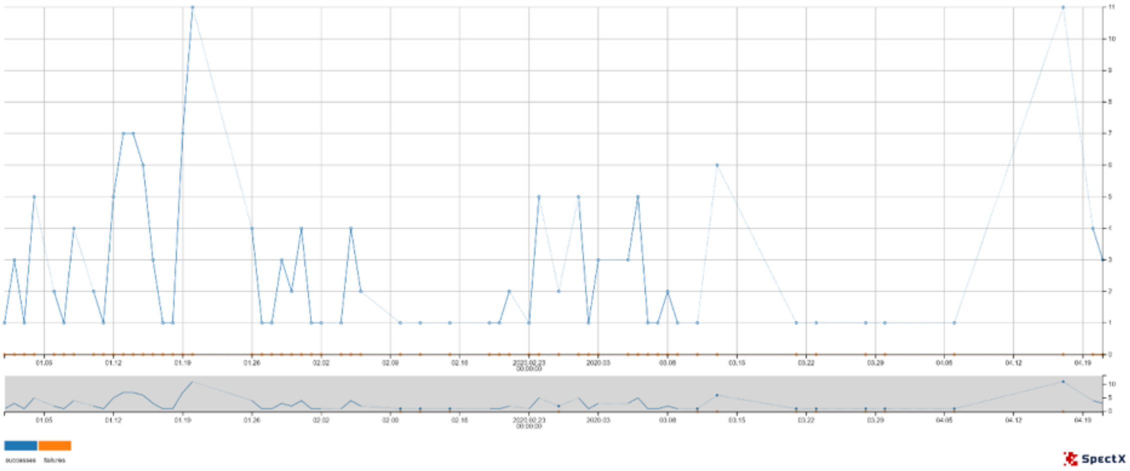


Figure 37. *User 8 before the lockdown.*

In Figure 37, this IP address has shown a normal trend. There were two highest peaks during the whole time slot: on the 20th of January and on the 17th of April with 11 successful logins each. There were some time breaks during the lockdown between the 21st of March and the 6th of April.

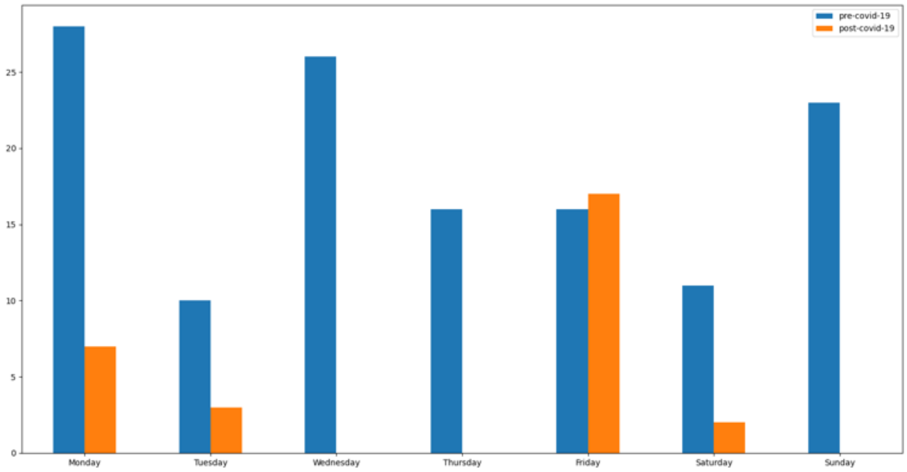


Figure 38. *User 8 before/after the lockdown.*

From Figure 38 it is clear that the peak happened to be on Monday during normal times whereas during the lockdown there was a suspicious change on Friday, which shows that there were more activities than during normal times.

9. **IP User 9**

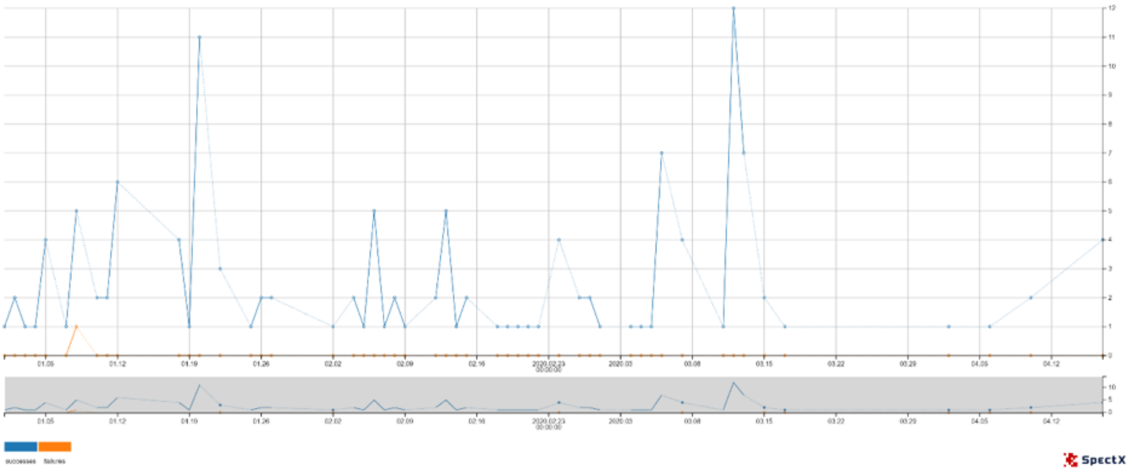


Figure 39. *User 9 before the lockdown.*

In Figure 39, this IP address follows a normal trend almost during the whole time slot. There were two major peaks: on the 20th of January and on the 12th of March with 11 and 12 successful logins respectively. There were only 1 failed attempts – on the 8th of January.

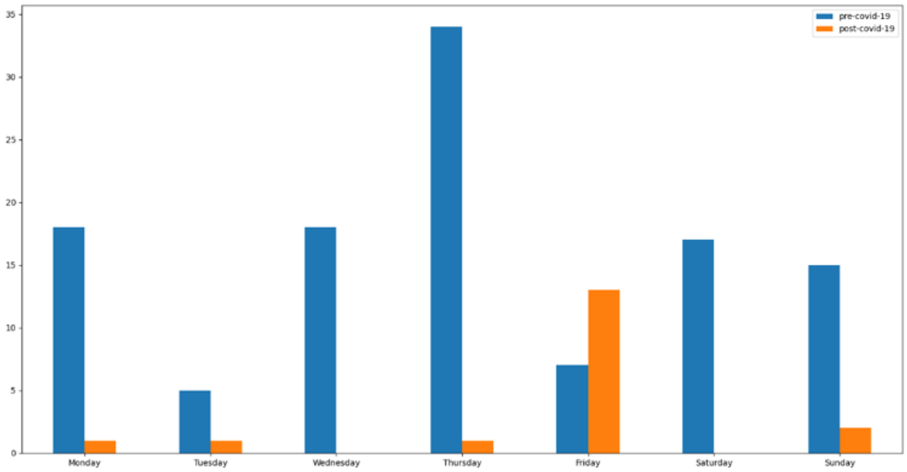


Figure 40. *User 9 before/after the lockdown.*

From Figure 40 it can be seen, that there was a peak during the whole period on Tuesday whereas on Friday there was a peak during lockdown which was higher than activity during normal times.

10. **IP User 10**

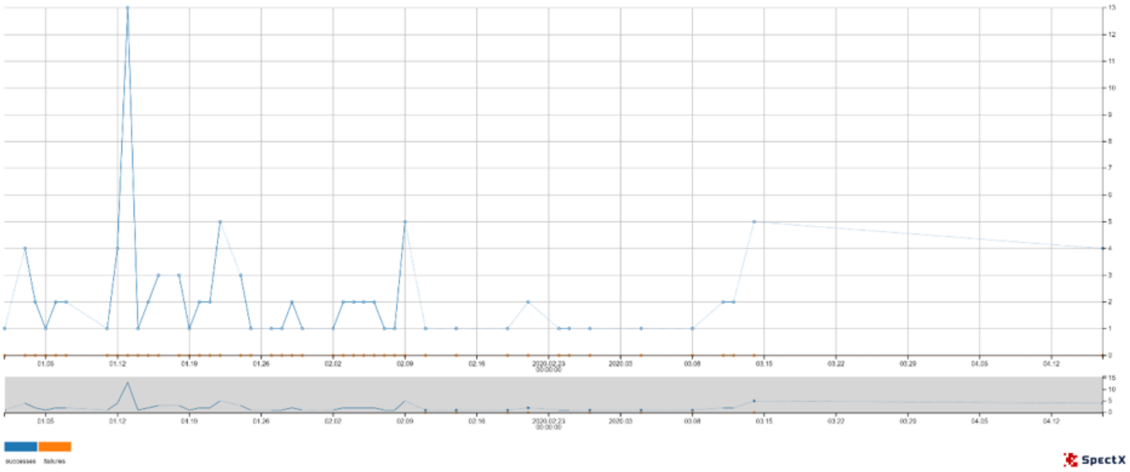


Figure 41. *User 10 before lockdown.*

From Figure 41, it can be seen, that this IP address has a peak on the 13th of January with 13 successful login attempts. There is a strange event. There is a long time break between the 14th of March and the 17th of April, almost during the whole lockdown IP address was not in use.

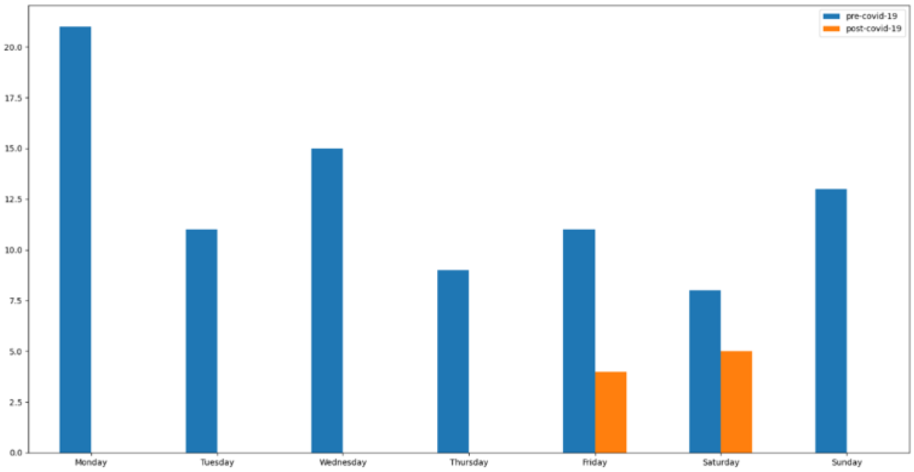


Figure 42. *User 10 before/after lockdown.*

From Figure 42, it can be seen that during lockdown there were activities only on Friday and Saturday. During normal time, the peak has happened to be on Monday.

5.1.4 Plots of the Top 10 (post-COVID-19) IP addresses during the whole time slot.

Blue is successful logins and orange is failed login attempts.

1. **IP User 1**

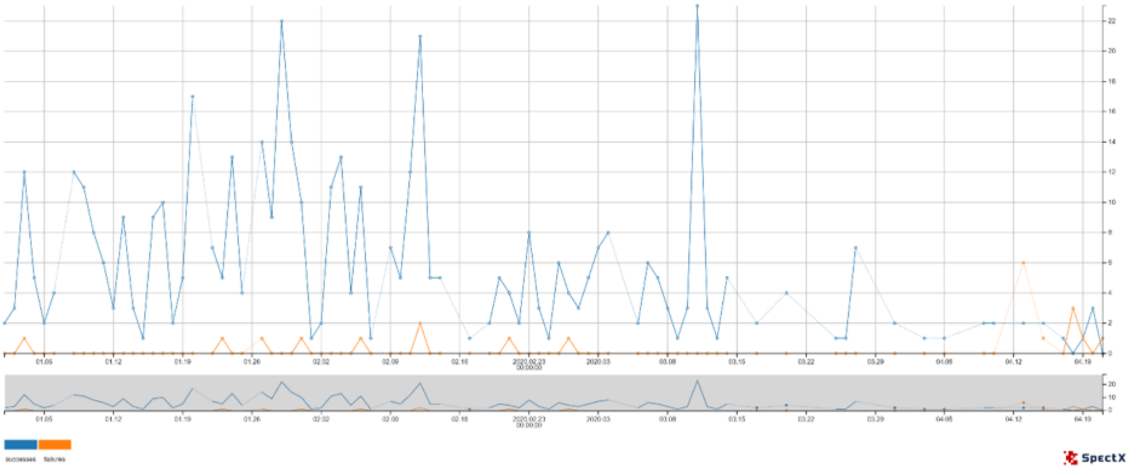


Figure 43. *User 1 after lockdown.*

In Figure 43, this IP address had more login peaks during normal times. The intervals between peaks were around 3-6 days long. The highest peak happened to be on the 11th of March with 23 successful logins. The highest peak of failed logins was on the 13th of April.

Generally speaking, there were more fluctuations during normal times.

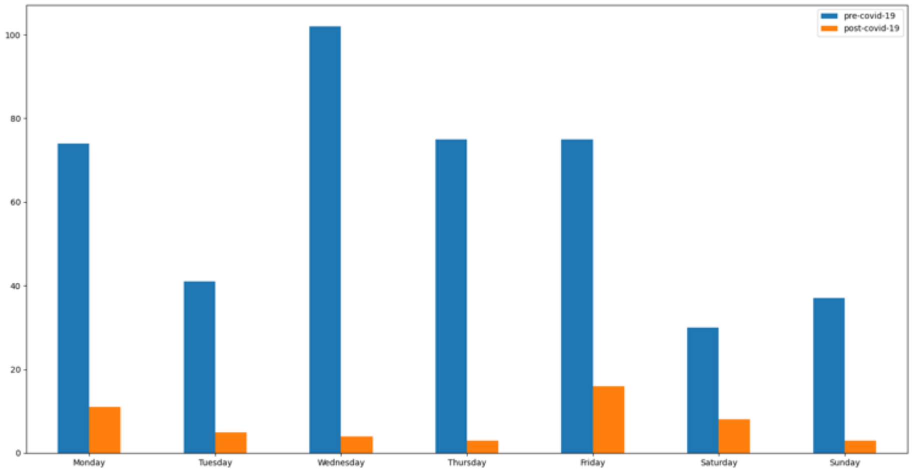


Figure 44. *User 1 before/after lockdown.*

In Figure 44, it can be seen, that during normal times the peak has happened to be on Wednesday whereas during lockdown there was a peak on Friday.

2. **IP User 2**

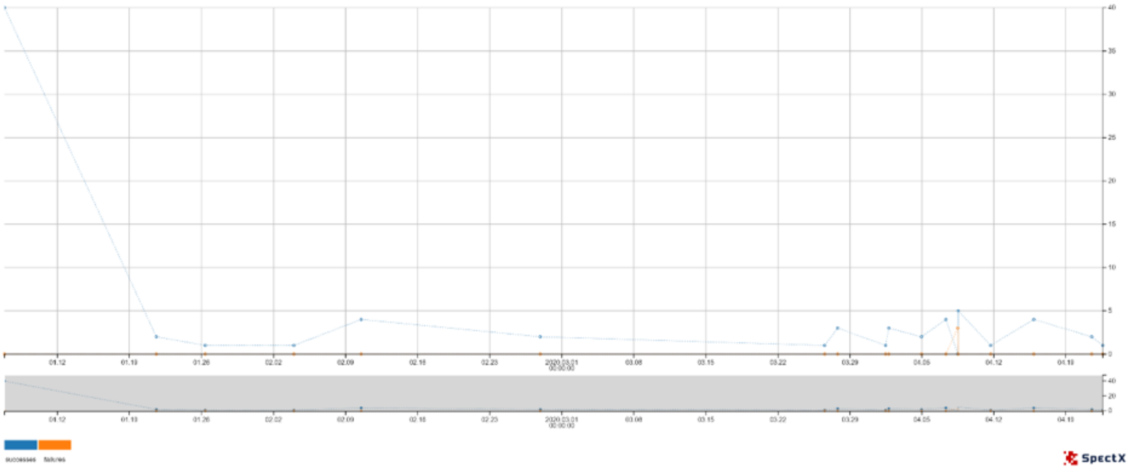


Figure 45. *User 2 after lockdown.*

On Figure 45, this user has shown a different trend with 40 successful logins on the 1st of January it went gradually down to two successful logins. There were 6 days when the user was active before lockdown has happened. The user has activated during the lockdown. There was a day, on the 8th of April with 3 failed login attempts. The logins of the user were more frequent than during normal times. It can be assumed that the user started to log in more often from home, but it is the same IP address. Users should have been following the same trend during normal times as well. But it did not happen. Thus, there can be some suspicions regarding this IP address.

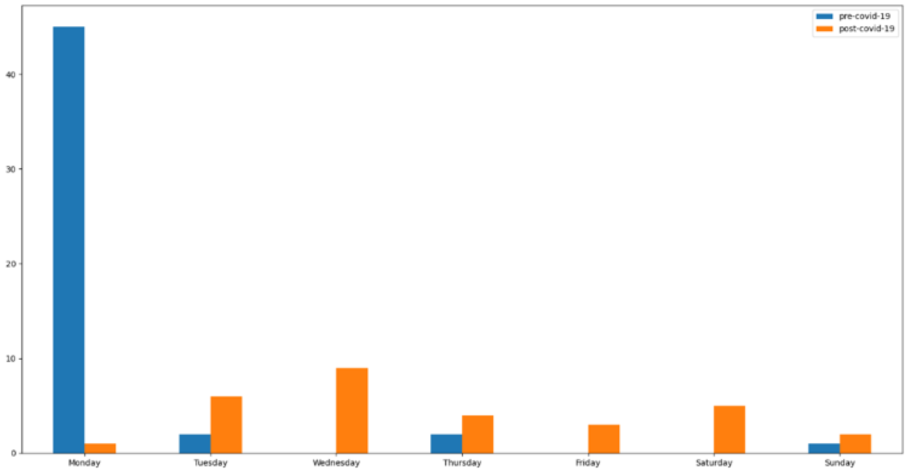


Figure 46. *User 2 before/after lockdown.*

As can be seen from Figure 46, during normal times, there was a peak on Monday whereas during lockdown there was a peak on Wednesday. In general, there were more activities during lockdown from this IP address.

3. **IP User 3**

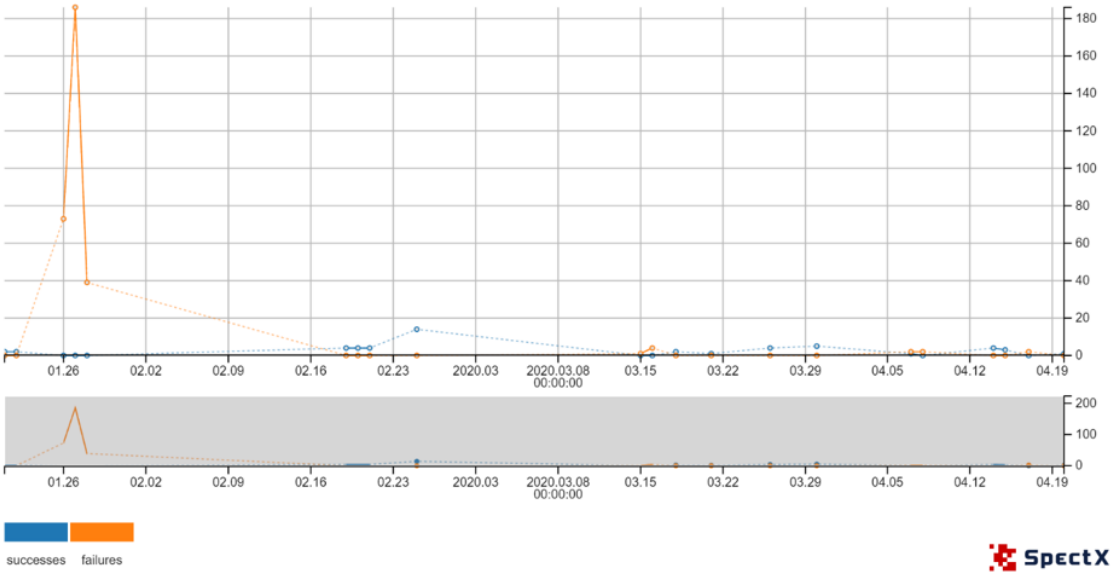


Figure 47. *User 3 after lockdown.*

In Figure 47, this IP address shows a particularly interesting trend. On the 28th of January, there are 186 failed login attempts. Between the 28th of January and 19th of February, there were no login attempts at all. After that, there was one day with up to 14 successful logins. After that, there were days with single failed/successful login attempts, but the number was not that huge.

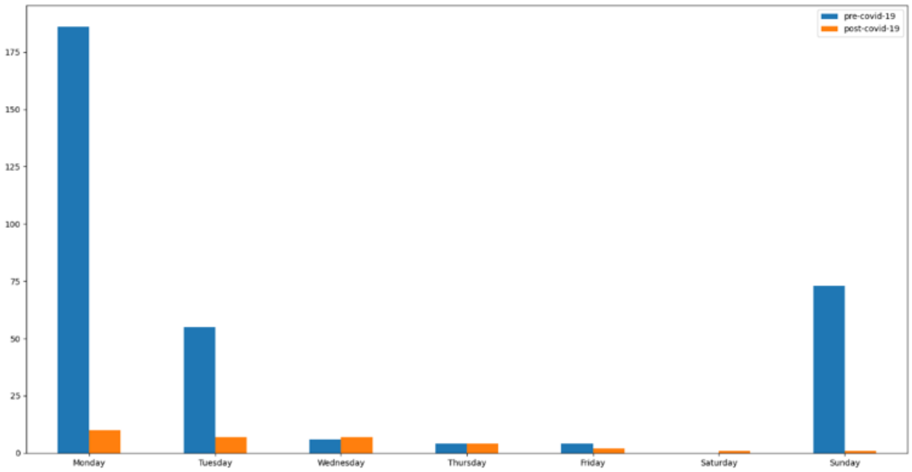


Figure 48. *User 3 before/after lockdown.*

As can be seen from Figure 48, during normal times as well as lockdown the peak has happened to be on Monday. Besides, Sunday was also a very active day during normal times only. In general, activity is very low from this IP address.

4. **IP User 4**

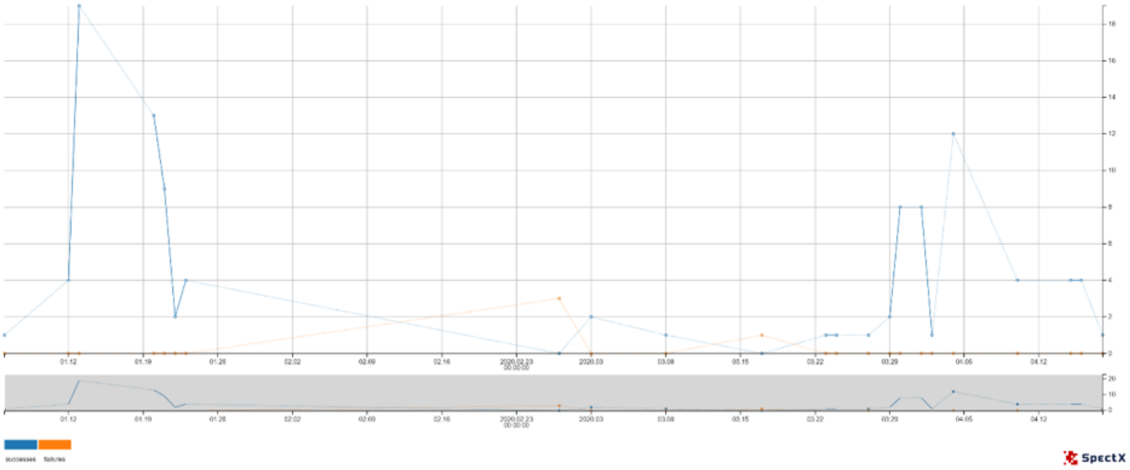


Figure 49. *User 4 after lockdown.*

From Figure 49, this IP address shows another interesting trend. The number of successful logins suddenly increased from 1 login on the 6th of January to 19 attempts on the 13th of January. There were no attempts in between. After this, it faced a sharp decrease to 2 successful logins on the 22nd of January followed by a slight rise on the next day up to 4 successful logins. This trend is unusual. There were 3 failed login attempts on the 27th of February. Another failed attempt appeared on the 17th of March. During the lockdown, there were ups and downs. The peak of the number of successful logins happened to be on the 4th of April with 12 successful logins.

As can be seen from Figure 50, during normal times there was a peak on Monday whereas during lockdown there was a peak on Saturday. Post-COVID-19 activity is very high during the lockdown.

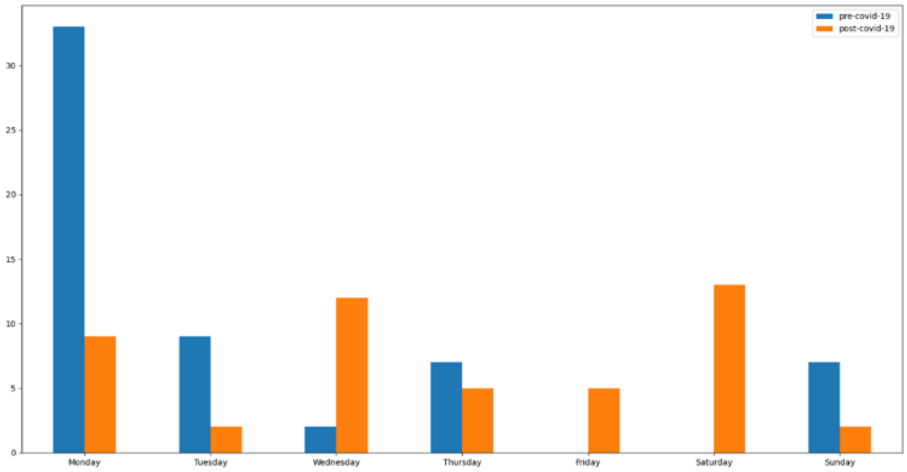


Figure 50. *User 4 before/after lockdown.*

5. **IP User 5**

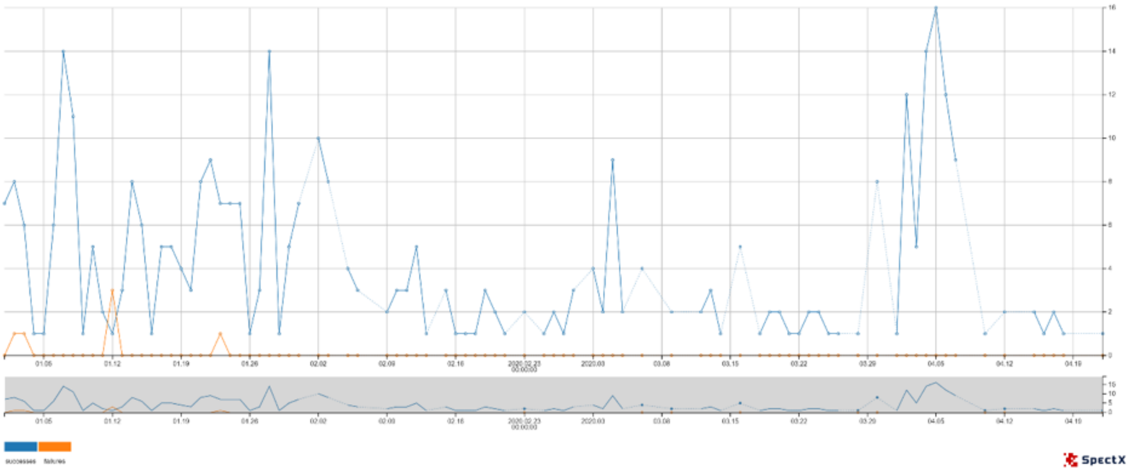


Figure 51. *User 5 after lockdown.*

On Figure 51, this IP address has more fluctuations during normal times but the peak of the whole time slot happened to be on the 5th of April with 16 successful attempts.

As can be seen from Figure 52, during normal times the peak happened on Tuesday whereas during lockdown the peak happened to be on Monday. In general, during normal times IP address was more active.

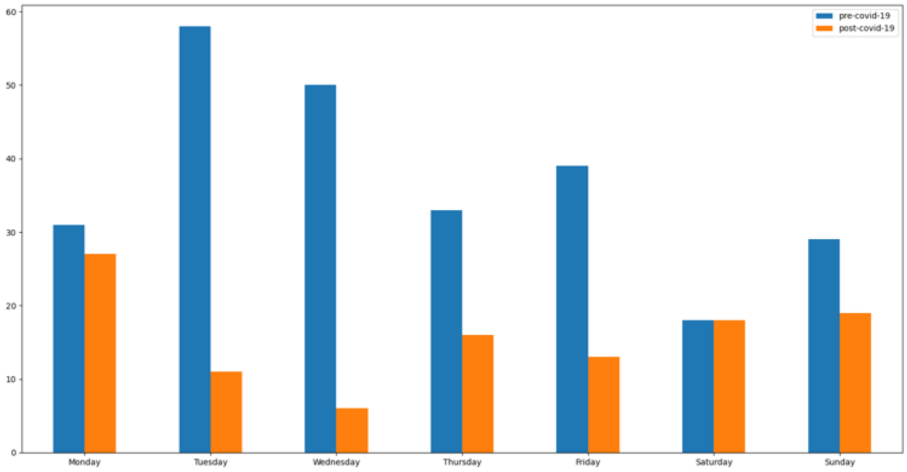


Figure 52. *User 5 before/after lockdown.*

6. **IP User 6**

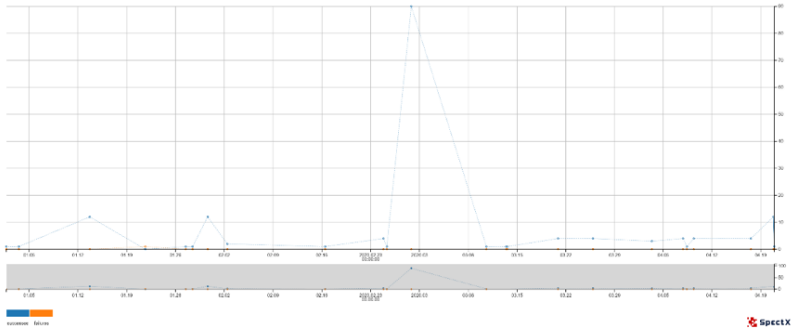


Figure 53. *User 6 after lockdown.*

In Figure 53, this IP address shows a different trend. The peak for the whole period happened to be on the 28th of February with the 90 successful logins. Before that, there were a few peaks with 12 successful logins on the 13th and 30th of January respectively.

During the lockdown, there was a stable situation.

As can be seen from Figure 54, during normal times, the peak was on Saturday whereas during lockdown the peak was on Friday.

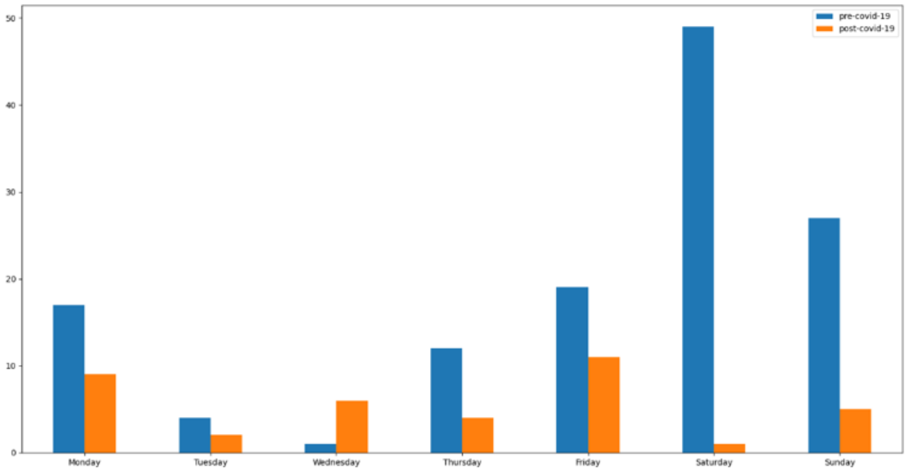


Figure 54. *User 6 before/after lockdown.*

7. **IP User 7**

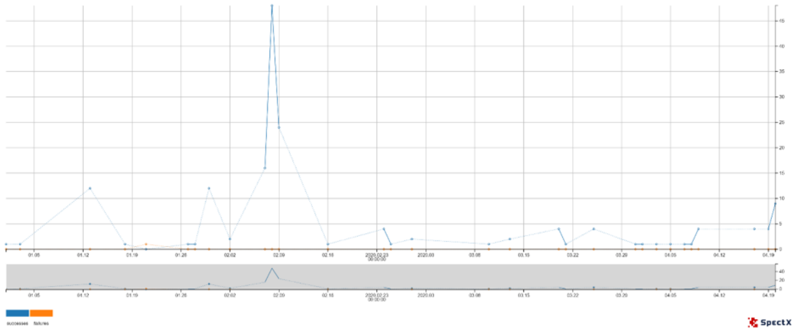


Figure 55. *User 7 after lockdown.*

On figure 55, this IP address has a peak on the 8th of February with 48 successful login attempts.

Generally speaking, the situation during the whole time slot was stable.

As can be seen from Figure 56, during normal times the peak happened on Wednesday whereas during lockdown the peak happened on Saturday. Wednesday is the most interesting day due to its huge difference in the level of activeness.

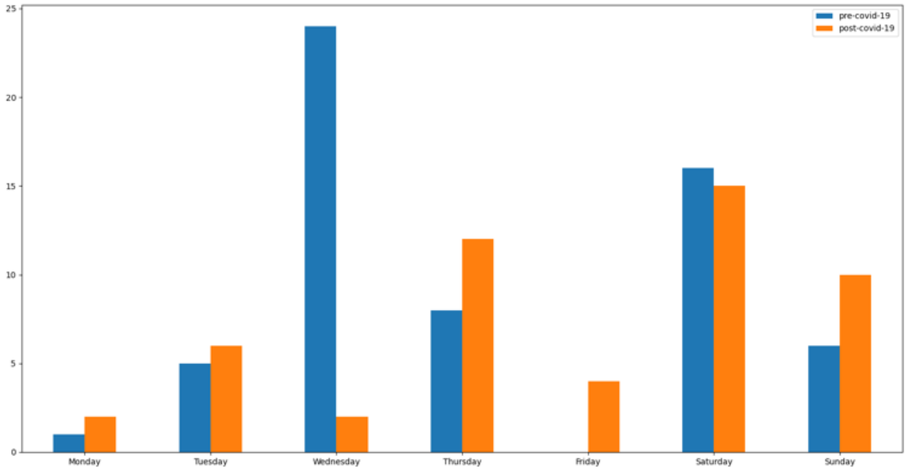


Figure 56. *User 7 before/after lockdown.*

8. **IP User 8**

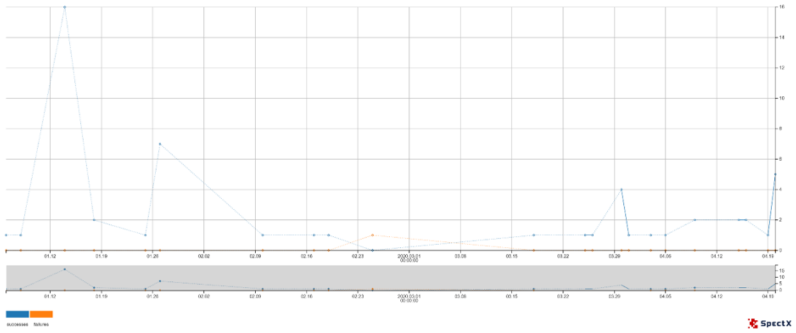


Figure 57. *User 8 after lockdown.*

In Figure 57, this IP address had a peak of successful logins on the 14th of January with 16 logins. After that, it gradually went down to 1 login. And again sharply increased to7 successful logins on the 27th of January. During the lockdown, the situation was very stable. There were no failed attempts at all.

As can be seen from Figure 58, during normal times the peak has happened on Tuesday whereas during lockdown the peak has happened on Monday. Monday is the most active day of the whole period.

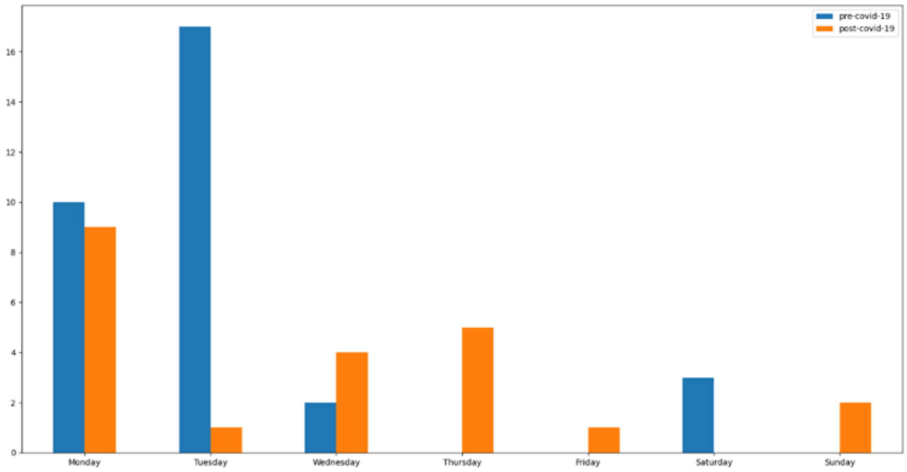


Figure 58. *User 8 before/after lockdown.*

9. **IP User 9**

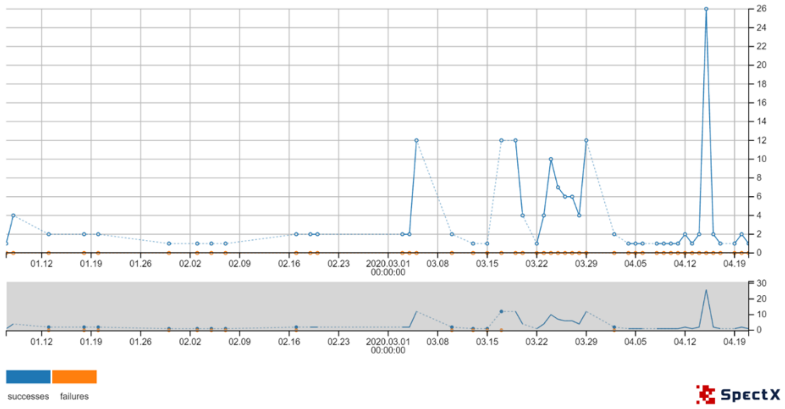


Figure 59. *User 9 after lockdown.*

In Figure 59, during the normal time, there were not so many attempts at all. Four successful login attempts on the 8th of January followed by few other days with a maximum of two successful attempts. The most interesting part starts after the 15th of March (time, when the lockdown was announced). A sudden rise has happened on the 19th of March with 12 successful logins and after it went down sharply to 1 successful login on the 22nd of March. Another peak happened to be on the 15th of April with 26 login attempts. Again, it went down sharply to one login on the 17th of April.

As can be seen from Figure 60, during normal times the peak has happened on Thursday whereas during lockdown the peak has happened on Wednesday. Generally speaking, during lockdown this IP address was the most active during all days.

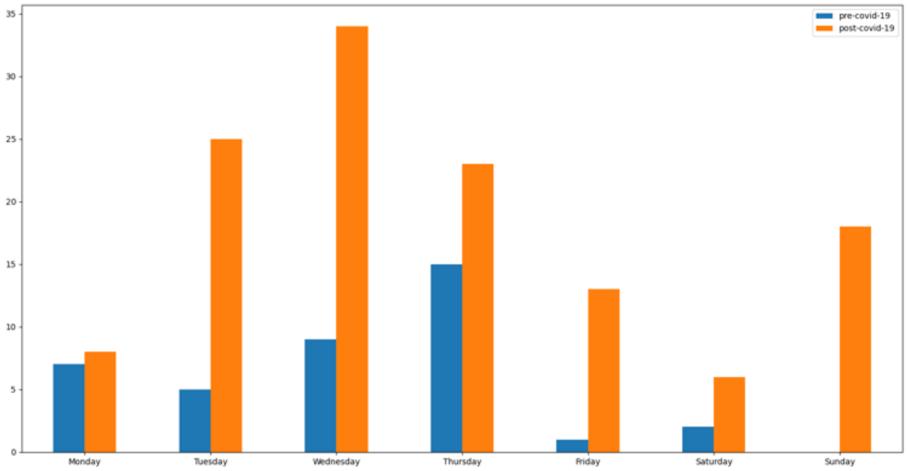


Figure 60. *User 9 before/after lockdown.*

10. **IP User 10**

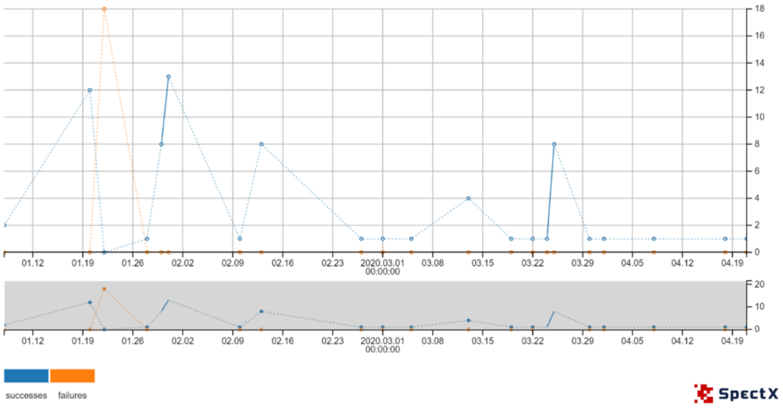


Figure 61. *User 10 after lockdown.*

In contrast to the previous IP, in Figure 61, this IP address shows more ups and downs during normal times. After the peak happened on the 20th of January with 12 successful logins. The rapid change has happened on the 22nd of January with 18 failed login attempts. Quite a rapid change makes this IP address suspicious.

And again, on the 31st of January, a sudden rise has happened up to 13 successful logins.

All these IP addresses are not IP addresses assigned to the public WIFI places etc.

As can be seen from Figure 62, during normal times as well as during lockdown the peak has happened on Wednesday. There were more peaks during normal times.

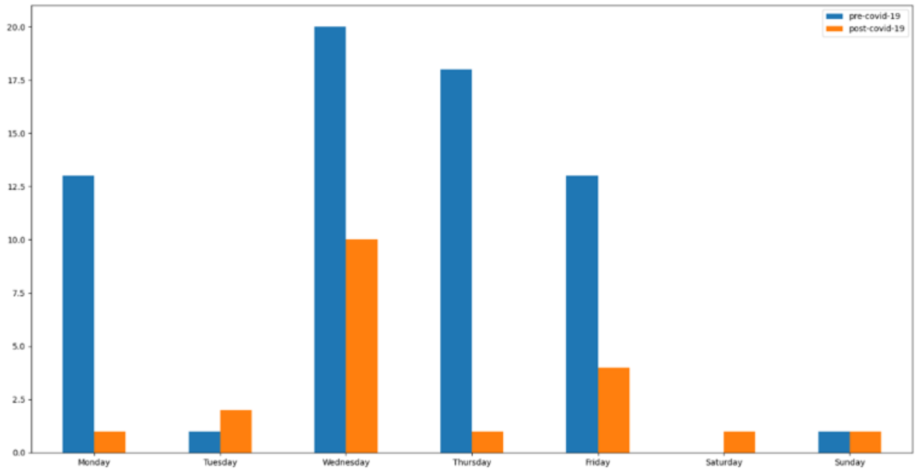


Figure 62. *User 10 before/after lockdown.*

5.1.5 Plots of the Top 5 emails with the highest number of successful logins during the whole period.

With the help of the following script, it is possible to calculate the number of failed and successful logins and also emails itself:

This script shows a list of Top 5 emails with the highest number of successful logins during the whole period before and after lockdown has happened:

```
failure_df = df_with_daynumber[df_with_daynumber['event'] == success] failure_emails_10
= list(failure_df['email'].value_counts().index)[:5] failure_emails_10
```

List:

1. User Email 1 (49443 successful logins)
2. User Email 2 (10517 successful logins)
3. User Email 3 (8523 successful logins)
4. User Email 4 (8372 successful logins)
5. User Email 5 (6431 successful logins)

Emails with the highest number of successful logins:

1. **User Email 1**
   In Figure 63, this email address generates a huge number of successful logins, which

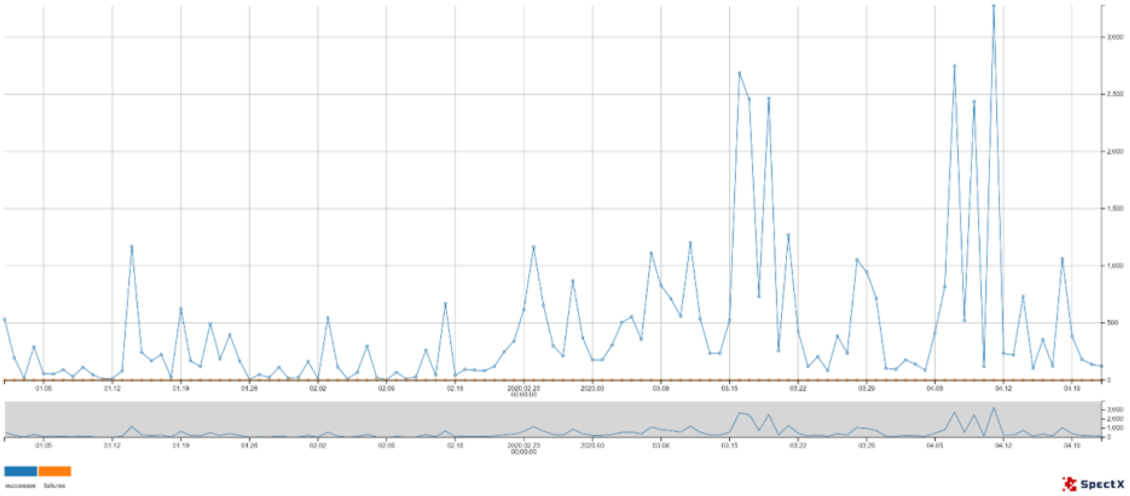


Figure 63. *User Email 1 with successful logins.*

makes this email suspicious already. Of course, there can be some old application that tries to log in every minute. But there can be some other malicious actors behind it. The first high peak of successful logins happened to be on the 14th of January with 1170 successful logins. Next to such a high peak happened to be on the 23rd of February with 1164 successful logins. The highest peak of the whole period happened to be on the 11th of April with 3274 successful logins which are enormous. A very interesting situation has happened exactly after lockdown has happened in Estonia, on the 16th of March there was a sharp increase from 232 logins on the 14th of March up to 2687 successful logins. This kind of behavior should be analyzed more in detail in future works as there are other details behind other systems where users logging in. And these systems contain different behavioral patterns.
Geolocations mostly come from Estonia. There are no very remote locations.
As can be seen from Figure 64, during normal times the peak has happened on Wednesday whereas during lockdown the peak has happened on Saturday. Generally speaking, during lockdown this email was more active than during normal times.

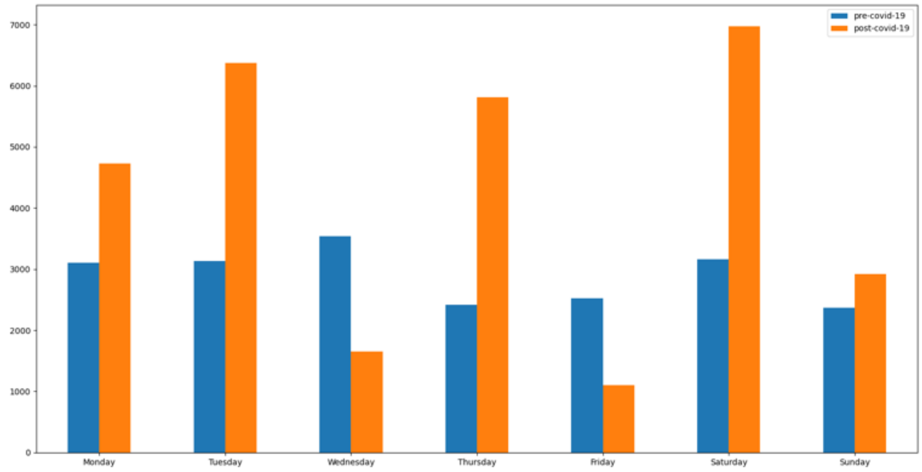


Figure 64. *User Email 1 before/after lockdown.*

2. **User Email 2**

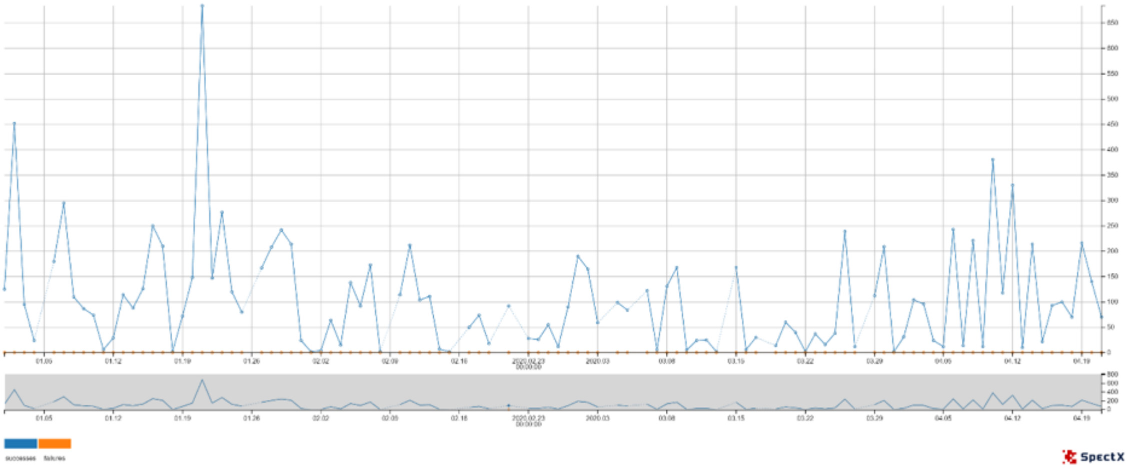


Figure 65. *User Email 2 with successful logins.*

In Figure 65, this email had more peaks before the lockdown than a previous email. There was a sharp increase on the 2nd of January with 452 successful logins. The next few peaks were on the 7th and 16th of January with 295 and 250 successful logins respectively. The highest peak during the whole period has happened on the 21st of January with 684 successful logins. If to look on the day the lockdown has happened, there was a stable situation, but there were some suspicious pattern, on the 6th, 8th, 10th, 12th and 14th of April there were five peaks in a row with 243, 221, 381, 330 and 214 successful logins respectively. The interval between these peaks equals two days, which makes it suspicious. There were no failed login attempts.

As can be seen from Figure 66, during normal times the peak has happened on Tuesday whereas during lockdown the peak has happened on Sunday. Generally

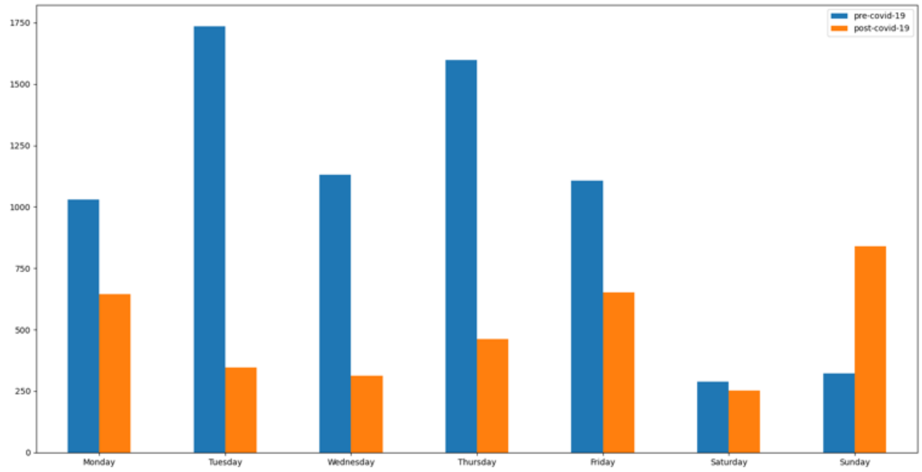


Figure 66. *User Email 2 before/after lockdown.*

speaking, during normal times this email was more active than during lockdown.

3. **User Email 3**

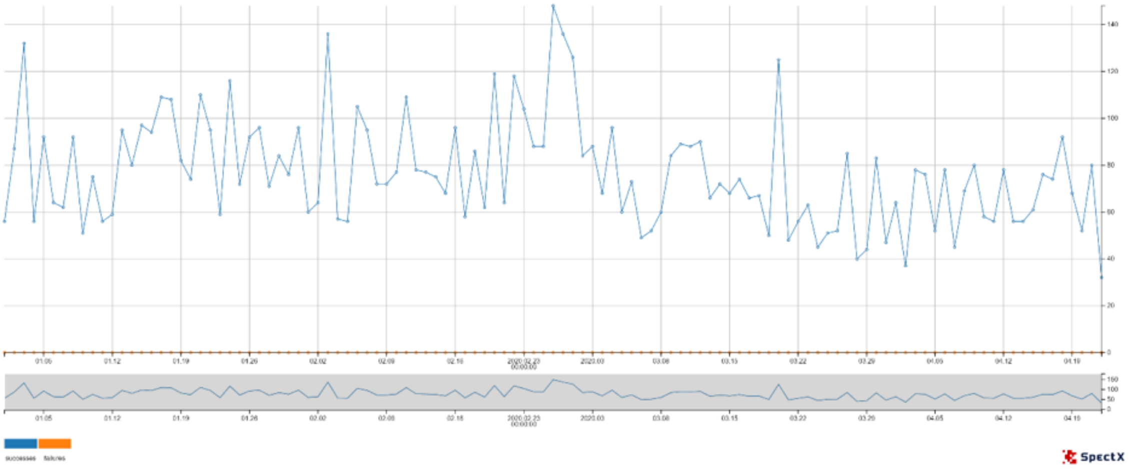


Figure 67. *User Email 3 with successful logins.*

In Figure 67, this email address has another tendency, the number of logins starts from 37 logins per day. In the previous cases, there were 1 or 2 logins minimum. The peak of the whole period is on the 26th of February with 148 successful logins. There is a tendency of these peaks of successful logins, every 2-3 days these peaks can be spotted. There was one high peak during the lockdown on the 20th of March with 125 successful login attempts. The main observation here is that the user has a high minimum bound of the number of logins.

As can be seen from Figure 68, during normal times the peak has happened on Thursday whereas during lockdown the peak has happened on Friday. Generally speaking, this email was active during normal times more than during lockdown.

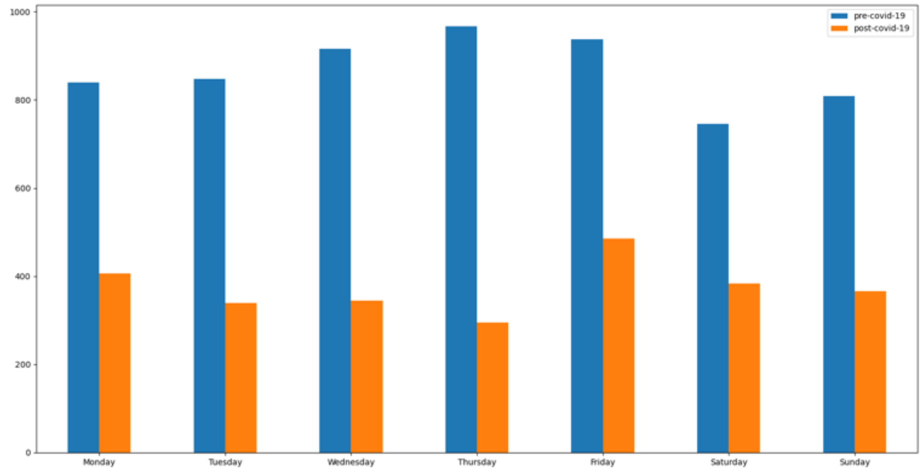


Figure 68. *User Email 3 before/after lockdown.*

4. **User Email 4**

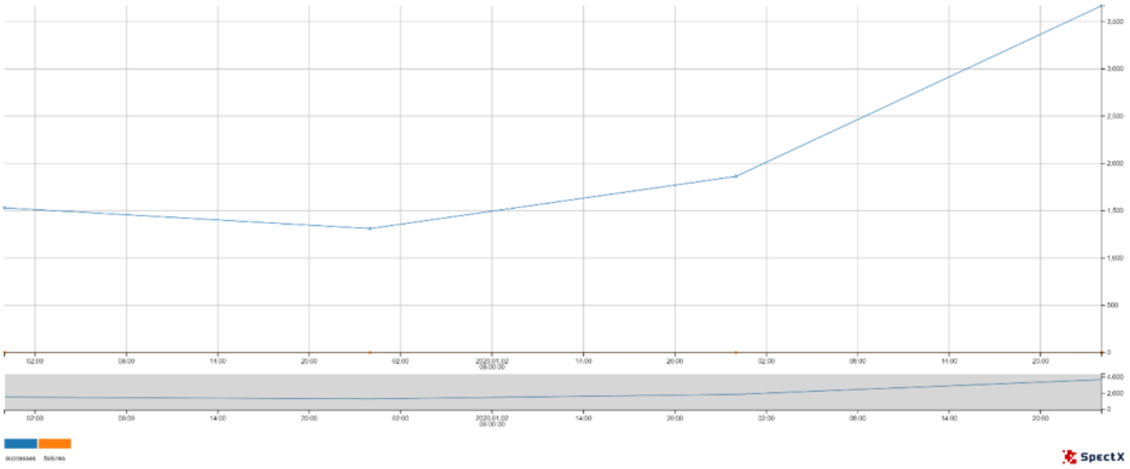


Figure 69. *User Email 4 with successful logins.*

In Figure 69, this email shows a very strange behavioral pattern. Only 4 days the user was active during the whole period. On the 1st of January, with 1531 logins, on the 2nd of January with 1312 logins, on the 3rd of January with 1863 logins and on the 4th of January with 3666 successful logins. Generally speaking, this email was not very active, thus it should be investigated in future works.

5. **User Email 5**

In Figure 70, this email contains many ups and downs. The first peak appeared on the 5th of February with 69 successful logins. Another large peak appeared on the 11th of February with 134 successful logins. The highest peak of the whole period appeared on the 2nd of March with 140 successful logins.

Geolocations mostly come from Estonia. There are no very remote locations.

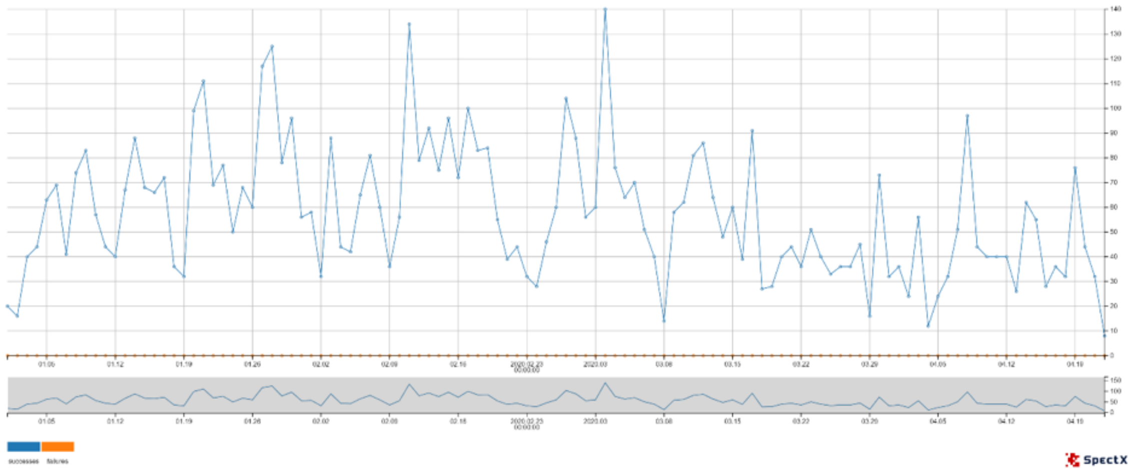


Figure 70. *User Email 5 with successful logins.*

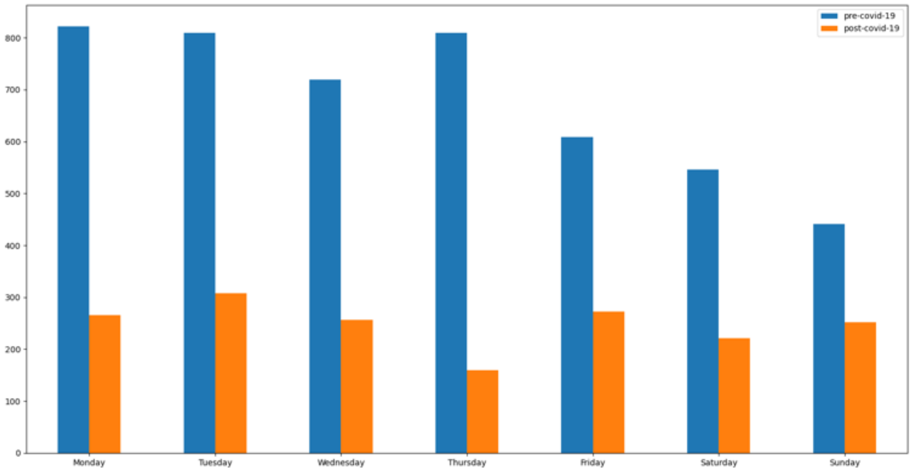


Figure 71. *User Email 5 before/after lockdown.*

As can be seen from Figure 71, during normal times the peak has happened on Monday whereas during lockdown the peak has happened on Tuesday. Generally speaking, this email was more active during normal times than during lockdown.

5.1.6 Plots of the Top 5 emails with the highest number of failed logins during the whole period.

This script shows a list of Top 5 emails with the highest number of failed logins during the whole period before and after lockdown has happened:

failure_df = df_with_daynumber[df_with_daynumber['event'] == 'failure'] failure_emails_10 = list(failure_df['email'].value_counts().index)[:5] failure_emails_10

List:

1. User Email 1 (29918 failed logins)
2. User Email 2 (14130 failed logins)
3. User Email 3 (13882 failed logins)
4. User Email 4 (9405 failed logins)
5. User Email 5 (9063 failed logins)

Emails with the highest number of failed logins:

1. **User Email 1**

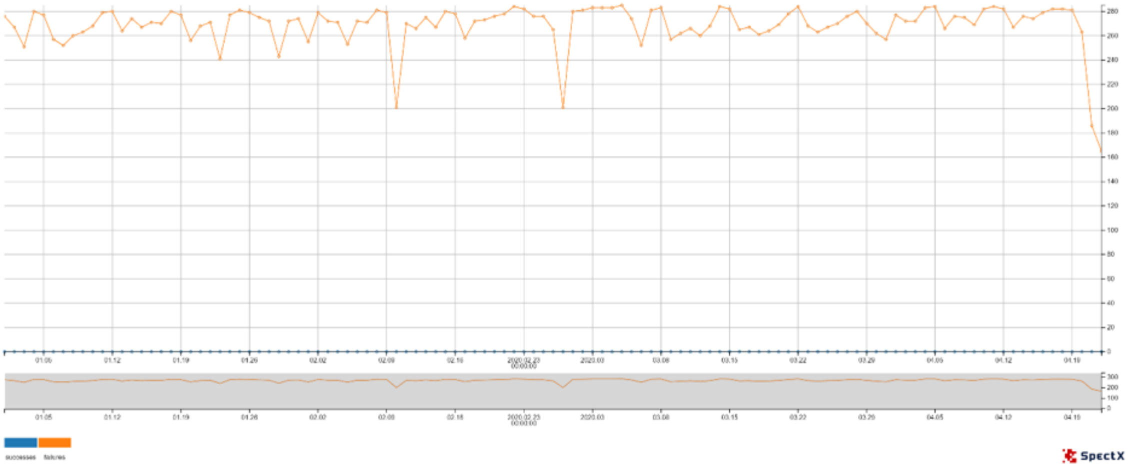


Figure 72. *User Email 1 with failed logins.*

In Figure 72, this email contains only failed login attempts. There is a huge number of these. The first small peak appeared on the 10th of February with 201 failed attempts. The same number failed logins happened to be on the 27th of February. These two are the smallest number of attempts per day. The maximum was around 300 attempts per day. There was some kind of trend, every week there was at least one small peak of attempts.
Geolocations mostly come from Estonia. There are no very remote locations. Generally speaking, this user looks suspicious in terms of the number of failed login attempts.

As can be seen from Figure 73, during normal times the peak has happened on Wednesday whereas during the lockdown the peak has happened on Saturday. Generally speaking, this email was more active during normal times than during lockdown.

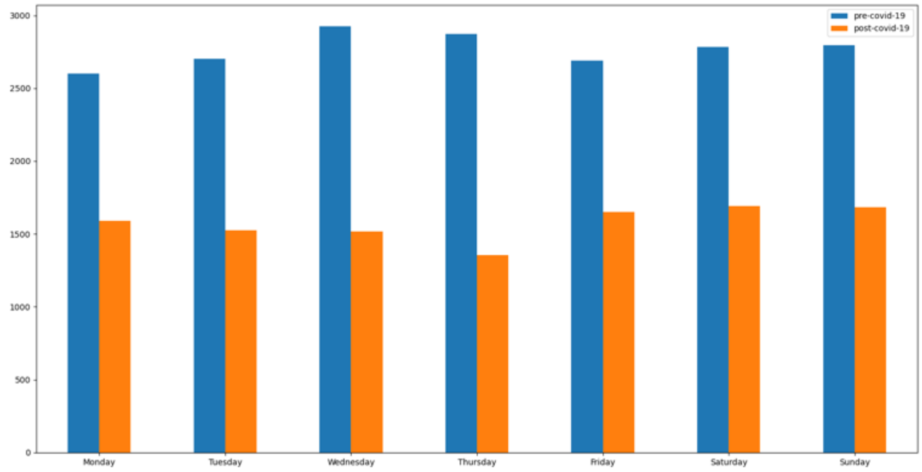


Figure 73. *User Email 1 before/after lockdown.*

2. **User Email 2**

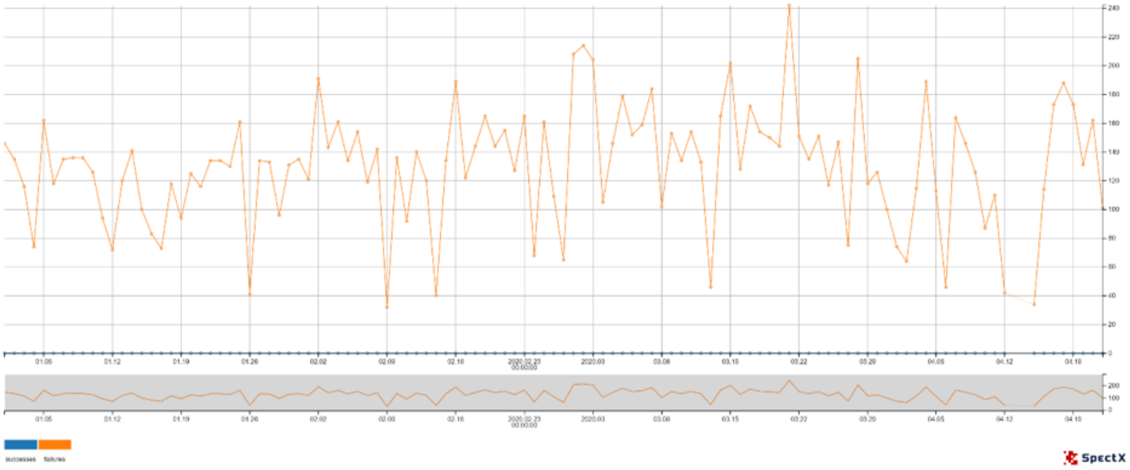


Figure 74. *User Email 2 with failed logins.*

In Figure 74, this email shows a very high number of fluctuations during the whole period. There were around 8 peaks during normal times. The highest peak of the whole period happened to be on the 21st of March with 242 failed login attempts. The lowest number of failed login attempts happened to be on the 9th of February. The number of failed logins rises a question about the legitimacy of the user's behavior.

As can be seen from Figure 75, during normal times the peak has happened on Wednesday whereas during lockdown the peak has happened on Saturday. Generally speaking, this email was more active during normal times than during lockdown.

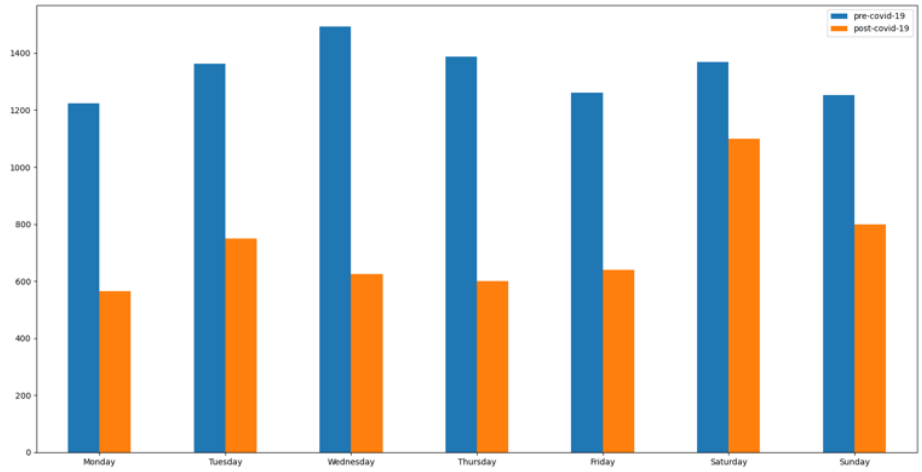


Figure 75. *User Email 2 before/after lockdown.*

3. **User Email 3**

   In Figure 76, this user has a less minimum number of login attempts per day. The smallest number was on the 4th of January with 26 failed attempts. It dropped down in comparison with the 2nd of January with 179 failed attempts. During the lockdown, there was a peak of the whole period on the 28th of March with 212 failed attempts. User behavior is still suspicious.

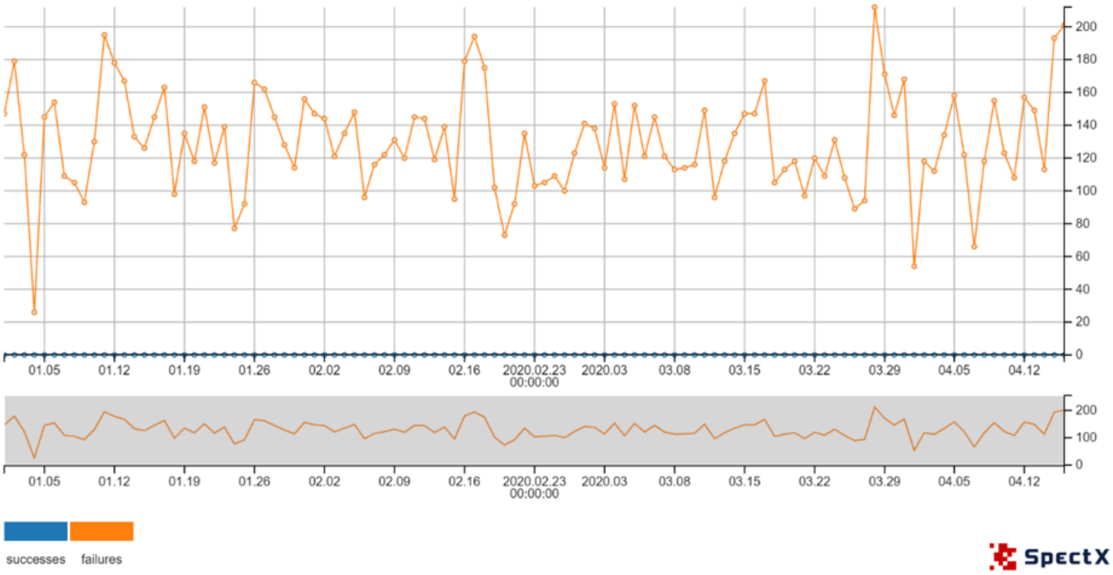


Figure 76. *User Email 3 with failed logins.*

As can be seen from Figure 77, during normal times the peak has happened on Wednesday whereas during lockdown the peak has happened on Sunday. Generally speaking, this email was more active during normal times than during lockdown.

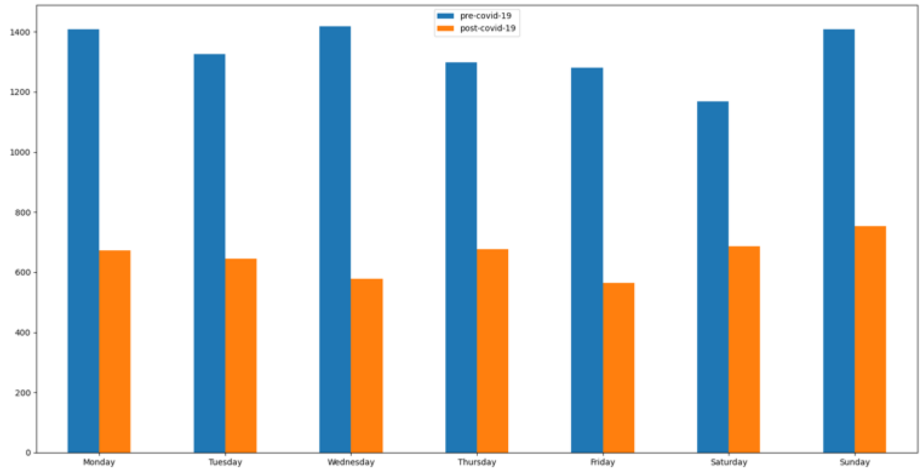


Figure 77. *User Email 3 before/after lockdown.*

4. **User Email 4**

   On Figure 78, this email contains the highest peak of the whole period on the 29th of February with 272 failed login attempts. The lowest number happened to be on the 25th of March with 1 failed attempt. Plot shows very fluctuated behavioural pattern for this user. There were less peaks during lockdown, which his also unusual. Geolocations mostly come from Estonia. There are no very remote locations.

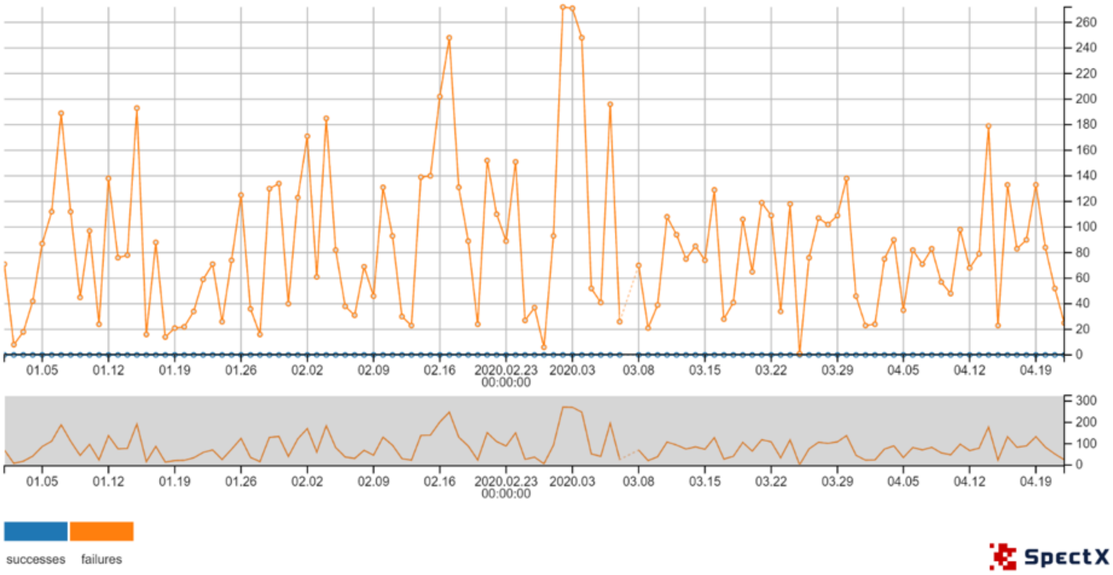


Figure 78. *User Email 4 with failed logins.*

This email contains the highest peak of the whole period on the 21st of April with 3788 failed login attempts.
Geolocations mostly come from Estonia. There are no remote locations.

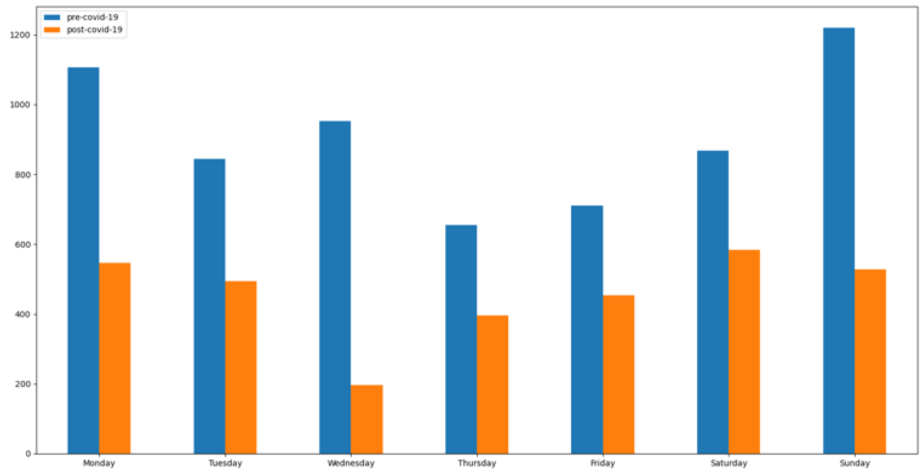


Figure 79. *User Email 4 before/after lockdown.*

As can be seen from Figure 79, during normal times the peak has happened on Sunday whereas during lockdown the peak has happened on Saturday. Generally speaking, this email was more active during normal times than during lockdown.

5. **User Email 5**

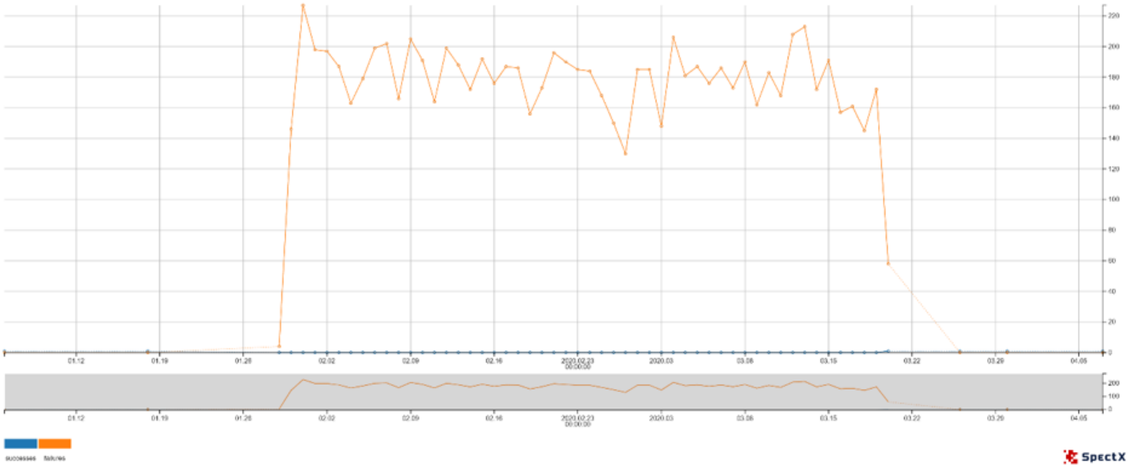


Figure 80. *User Email 5 with failed logins.*

On Figure 80, this email shows another interesting trend, with one successful login attempt on the 18th of January followed by 4 failed attempts on the 29th of January and sharp increase of failed attempts on the 31st of January with 227 attempts (which was the highest peak for the whole period). After some time, it gradually went down on the 20th of March with 58 failed attempts, during this day there was 11 successful login – which is strange. There were also two successful logins on the 26th of March and the 30th of March. This user is particularly interesting in terms of the number of failed attempts followed with a successful login.

Geolocations mostly come from Estonia. There are no remote locations.

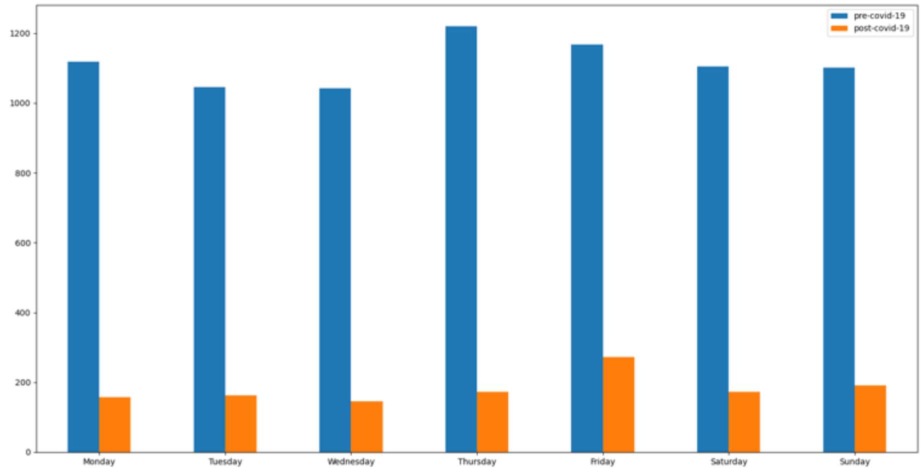


Figure 81. *User Email 5 before/after lockdown.*

As can be seen from Figure 81, during normal times the peak has happened on Thursday whereas during lockdown the peak has happened on Friday. Generally speaking, this email was more active during normal times than during lockdown. During the lockdown, this email has shown a very low level of activeness.

To sum up, during normal times, there were more users, but during the lockdown, the situation has changed and these deviations and fluctuations have shown that the users were more active during lockdown, but there were less users in comparison with pre-COVID-19. But during normal times, there also were some peaks, which can mean that these are public places with Wi-Fi.
The one thing which makes it more suspicious is the number of failed logins of some of the IP addresses and email addresses.

Following information was previously anonymized and can be used for future anomaly detection investigation in the current enterprise environment:

1. Analyze Top 10 IP addresses into which the biggest number of emails tried to log in before lockdown has happened;
2. Analyze Top 10 IP addresses into which the biggest number of emails tried to log in after lockdown has happened;
3. Analyze Top 5 users with the highest number of failed login attempts;
4. Analyze Top 5 users with the highest number of successful login attempts.

By achieving this, it is possible to warn users about any changes related to their accounts as well as prevent future malicious activities by adding some users to blacklists or deleting some fake email addresses. This methodology also helps to understand if there are any malicious activities from the accounts created by competitors. But this methodology is more specific for the dataset described in this thesis.

6. Chapter 6

6.1 Conclusions

In this thesis, the problem of detecting anomalies in large datasets is considered. The main focus is on developing a method that will help to find anomalies in a huge dataset extracted from a real system. Dataset is taken from the time slot during the COVID-19 pandemic has happened. This dataset is unique in terms of its content. It contains data before and after lockdown has happened in Estonia.

The problem of multiple login attempts from different locations at the same time allows covering an important part of the anomaly detection techniques which ultimately can be used by Estonian police for criminal case investigation.

Since the current method uses a recently extracted dataset of logs from the real system environment of an enterprise company during COVID-19 lockdown that has happened in Estonia, it makes this methodology unique as it creates an analysis of different types of logs with different structure and data types and reveals some interesting patterns which were detected during the lockdown and which are different from what was during normal time. There is no predefined set of anomalies that can be analyzed in advance.

An interesting fact, is that the amount of successful and failed login attempts decreased after lockdown has happened and the number of unique user locations increased. Before lockdown, there were more failed login attempts. There was a trend of decreased locations around the world during pandemic times. Besides, there were some unexpected user locations such as in Iran during pandemic times. But it could be just a user which uses his Estonian SIM card in Iran.

By analyzing specific user IP addresses and Emails before and after COVID-19 lockdown has happened, there were detected some trends with a very big amount of failed logins and successful logins at the same time. Lists of IP addresses and emails could be used for the future investigation by security analysts of the company or it can be useful for some specific cases investigated by Estonian police.

6.2 Future work

1. This dataset contains an important set of data about user activities before and after the COVID-19 lockdown has happened in Estonia. Thus it is not a full period during the whole year which can make a better overview and detect more interesting patterns. It is a very short time slot from the 1st of January until the 20th of April, which is a bit short for detecting anomalies. Besides, there is no information about previous anomalies. In future, the dataset can be expanded more up to 1 year duration. But current work can be published and used by security analysts and researchers in the nearest future.
2. The use case of multiple logins from different locations at the same time can be expanded to a new 1 year duration dataset.

   In future research, new techniques will be studied to make this process as much automated as possible.
3. Current methodology gives a simple example of a feature selection technique as it is clear which features are needed due to a very limited dataset quantity of features. Some more advanced techniques for data summarization and feature selection can be used in future research to make a feature selection much easier and more precise.
4. Very limited quantity of features used in the current dataset as there was no access, because of data confidentiality. In future research, more permissions are needed to include protocol-level analysis and port scanning detection. Besides, gaining information about which systems users tried to login into will be beneficial.
5. The Current method can also be practically used by Estonian police for criminal case investigation. As this method describes very common dataset features and manipulations with data, this will be a good starting point for criminal case investigation within the mobile network telecommunication domain.
6. It will also be beneficial in future to take the larger COVID-19 time slots and split it equally, so there will be some balance in data before and after the lockdown has happened in Estonia.

Bibliography

[1] Techrepublic. [Online]. Available: https://www.techrepublic.com/article/covid-19-lockdowns-are-causing-a-huge-spike-in-data-breaches/

[2] Herjavec Group. [Online]. Available: https://www.herjavecgroup.com/threat-advisory-additional-information-covid19-cyber-attacks/

[3] D. Beazley, "Data Processing with Pandas", The usenix magaizne, vol. 37, pp. 76-81, 2012.

[4] M. Ahmed, "Data summarization: a survey," Knowledge and Information Systems, vol. 58, no. 2, pp. 1-30, 2019.

[5] M. Ahmed, "Intelligent Big Data Summarization for Rare Anomaly Detection," IEEE Access, vol. 7, pp. 68669-68677, 2019.

[6] L. Hu, "A Novel Method of Network Traffic Anomaly Detection," International Conference on Electronic Mechanical Engineering and Information Technology, vol. 8, pp. 4757-4759, 2011.

[7] SpectX, "SpectX documentation," [Online]. Available: https://docs.spectx.com/v2/index.html. [Accessed 2015].

[8] Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Graylog.

[9] Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Splunk.

[10] Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Plumbr.

[11] O. Zaitsev, "Aggregation and Grouping," Medium, Lile, 2017.

[12] Data Mining with Orange, 2020.

[13] L. Akoglu, H. Tong and D. Koutra, "Graph based Anomaly Detection and Description: A Survey," Data Min. Knowl. Discov, vol. 29, no. 3, pp. 1-68, 2015.

[14] N. Laptev, S. Amizadeh and I. Flint, "Generic and Scalable Framework for Automated Time-series Anomaly Detection," Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p. 1939–1947, 2015.

[15] Domingos S.de O. Santos Júnior, João F.L.de Oliveira and Paulo S.G.de Mattos Neto, "An intelligent hybridization of ARIMA with machine learning models for time series forecasting," Knowledge-Based Systems, vol. 175, pp. 72-86, 2019.

[16] M. Rong, D. Gong and X. Gao, "Feature Selection and Its Use in Big Data: Challenges, Methods, and Trends," IEEE, vol. 7, pp. 19709 - 19725, 2019.

[17] V.Bolón-Canedo, N.Sánchez-Maroño and A.Alonso-Betanzos, "Recent advances and emerging challenges of feature selection in the context of big data," Knowledge-Based Systems, vol. 86, pp. 33-45, 2015.

[18] V. Bolón-Canedo, N. Sánchez-Maroño and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," Springer-Verlag London Limited, vol. 34, no. 3, p. 483–519, 2012.

[19] A. Jain and D. E. Zongker, "Feature Selection: Evaluation, Application, and Small Sample Performance," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 2, pp. 153-158, 1997.

[20] M. Kudo and J. Sklansky, "A Comparative Evaluation of Medium- and Large-Scale Feature Selectors for Pattern Classifiers," Kybernetika, Praha, vol. 34, no. 4, pp. 429-434, 1997.

[21] C. K. Emani, N. Cullot and C. Nicolle, "Understandable Big Data: A survey," Computer Science Review 17, pp. 1-12, 2015.

[22] Y. Harada, Y. Yamagata, O. Mizuno and E.-H. Choi, "Log-based Anomaly Detection of CPS Using a Statistical Method," IEEE International Workshop on Empirical Software Engineering in Practice (IWESEP), vol. 8, pp. 1-6, 2017.

[23] Y. Cui, Y. Sun, J. Hu and G. Sheng, "A Convolutional Auto-encoder Method for Anomaly Detection on System Logs," IEEE International Conference on Systems, Man, and Cybernetics, pp. 3057-3062, 2018.

[24] X. Xia, W. Zhang and J. Jiang, "Ensemble Methods for Anomaly Detection Based on System Log," IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC), vol. 24, pp. 93-94, 2019.

[25] J. Breier and J. Branisova, "A Dynamic Rule Creation Based Anomaly Detection Method for Identifying Security Breaches in Log Records," Wireless Personal Communications: An International Journal, vol. 94, no. 3, p. 497–511, 2015.

[26] X. Yan, W. Zhou, Y. Gao, Z. Zhang, J. Han and G. Fu, "PADM: PageRank-based Anomaly Detection Method of Log Sequences by Graph Computing," IEEE 6th

International Conference on Cloud Computing Technology and Science, vol. 6, pp. 700-703, 2014.

[27] A. Rabkin, W. Xu, A. Wildani, A. Fox, D. Patterson and R. Katz, "Graphical Representation for Identifier Structure in Logs," Workshop on Managing Systems via Log Analysis and Machine Learning, pp. 1-9, 2010.

[28] V. Chandola, A. Banerjee and V. Kumar, "Anomaly Detection: A Survey," ACM Computing Surveys, vol. 41, no. 3, pp. 1-72, 2009.

[29] H. Kosorus, J. Honigl and J. Kung, "Using R, WEKA and RapidMiner in Time Series Analysis of Sensor Data for Structural Health Monitoring," 22nd International Workshop on Database and Expert Systems Applications, vol. 22, pp. 306-310, 2011.

[30] T. C. Pessoa, R. Medeiros, T. Nepomuceno, G.-B. Bian, V. Albuquerque and P. P. R. Filho, "Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications," IEEE Access, vol. 4, no. 99, pp. 1-9, 2016.

[31] Z. Liu, T. Qin, X. Guan, H. Jiang and C. Wang, "An Integrated Method for Anomaly Detection From Massive System Logs," IEEE Open Access Journal, pp. 30602-30611, 2018.

[32] United States Computer Emergency Readiness Team, "COVID-19 Exploited by Malicious Cyber Actors," [Online]. Available: https://www.us-cert.gov/ncas/alerts/aa20-099a.

Appendices

Appendix 1 - Code for the data visualization with Pandas and SpectX

To visualize a frequency of failed and successful logins day/night time, before and after lockdown has happened in Estonia:

```
import matplotlib
import numpy
import pandas
import matplotlib.pyplot as plt
contents_pre_day = [i for i in open("pre_day.csv")][1:]
contents_post_day = [i for i in open("post_day.csv")][1:]
contents_pre_night = [i for i in open("pre_night.csv")][1:]
contents_post_night = [i for i in open("post_night.csv")][1:]
before_day = [[int(i.split(";")[1])] for i in contents_pre_day]
after_day = [[int(i.split(";")[1])] for i in contents_post_day]
before_night = [[int(i.split(";")[1])] for i in contents_pre_night]
after_night = [[int(i.split(";")[1])] for i in contents_post_night]
all = [i + j + k + m for i, j, k, m in zip(before_day, after_day, before_night, after_night)]
df = pandas.DataFrame(all, columns = [’pre_day’, ’post_day’, ’pre_night’, ’post_night’])
boxplot = df.boxplot()
boxplot.plot()
plt.show()
```

To visualize mean, min, max, quantiles of failed and successful logins before and after lockdown has happened in Estonia:

```
import matplotlib
import numpy
import pandas
import matplotlib.pyplot as plt
contents = [i for i in open("mean_hourly.csv")][1:]
before = [[int(i.split(";")[3]),

int(i.split(";")[4])] for i in contents if i.split(";")[5] == "true\n"]
after = [[int(i.split(";")[3]),

int(i.split(";")[4])] for i in contents if i.split(";")[5] == "false\n"]
print(before)
all = [i + j for i, j in zip(before, after)]
df = pandas.DataFrame(all, columns = [’successes pre−covid’,
’failures pre−covid’, ’successes covid’, ’failures covid’])
boxplot = df.boxplot()
boxplot.plot()
plt.show()
```

To visualize a distribution of the amount of unique login attempts:

```
@result = @[/shared_usecases/sso/union/Result2020_02_27.sxt];
@a = @result.filter(event = "success").group(month, day(date), email)
.select(month, day(date), count(email) as count_email).group(day, month);
@b = @result.filter(event = "success").select(month, day(date),

count(event = "success") as count_success).group(day, month);
@a.join(@b on left.month = right.month and left.day = right.day).select(month, day, count_email, count_success,

float(count_success)/float(count_email) as
logins_per_email).sort(day).sort(month).select(rowid(), logins_per_email)
```

To visualize a distribution of the average amount of unique locations per user per day:

```
import matplotlib
import numpy
import pandas
import matplotlib.pyplot as plt
contents_post_covid = [i for i in open("10.csv")][1:]
contents_pre_covid = [i for i in open("10_pre_covid.csv")][1:]
before = [float(i.replace(",", ".").split(";")[0])
for i in contents_pre_covid]
after = [float(i.replace(",", ".").split(";")[0])
for i in contents_post_covid]
print(before)
all = [[i , j] for i, j in zip(before, after)]
#all = [[before]] + [[after]]
df = pandas.DataFrame(all, columns = ['before', 'after'])
print(df)
boxplot = df.boxplot()
boxplot.plot()
plt.show()
```

To visualize a distribution of the amount of email addresses into which one IP address tried to login into (to see if there are any outliers) and to depict it on the world's map:

```
import matplotlib
import numpy
import pandas
import matplotlib.pyplot as plt
contents = [i for i in open("countEmailResult.csv")][1:]
all = [[int(i.split(";")[1])] for i in contents]
df = pandas.DataFrame(all, columns = ['count_email'])
boxplot = df.boxplot()
boxplot.plot()
plt.show()
```

To visualize a distribution of outliers before and after lockdown has happened:

```
import matplotlib
import numpy
import pandas
import matplotlib.pyplot as plt
contents_pre = [i for i in open("pre_covid.csv")][1:]
contents_post = [i for i in open("post_covid.csv")][1:]
before = [[int(i.split(";")[1])] for i in contents_pre]
after = [[int(i.split(";")[1])] for i in contents_post]
print(before)
all = [i + j for i, j in zip(before, after)]
df = pandas.DataFrame(all, columns = ['pre-covid', 'post-covid'])
boxplot = df.boxplot()
boxplot.plot()
plt.show()
```

To visualize a distribution of unique user's locations before lockdown has happened in Estonia using Scatter Plot:

```
import matplotlib
import numpy
import pandas
import matplotlib.pyplot as plt
contents_pre = [i for i in open("loginAttemptsToamountOfLocations.csv")][1:]
successful = [[int(i.split(";")[0]), int(i.split(";")[1])] for i in contents_pre]
failed = [[int(i.split(";")[2]), int(i.split(";")[3])] for i in contents_pre]
plt.scatter([e[0] for e in successful], [e[1] for e in successful], c='blue', label='Successful')
plt.scatter([e[0] for e in failed], [e[1] for e in failed], c='red', label='Failed')
plt.title('Scatter Plot')
plt.xlabel('successful/failed')
plt.ylabel('locations')
plt.legend()
plt.show()
```

Appendix 2 - SpectX installation

To begin with, installed SpectX is needed, installation .dmg package was downloaded.

Drag the SpectX-Desktop.app to Application folder:

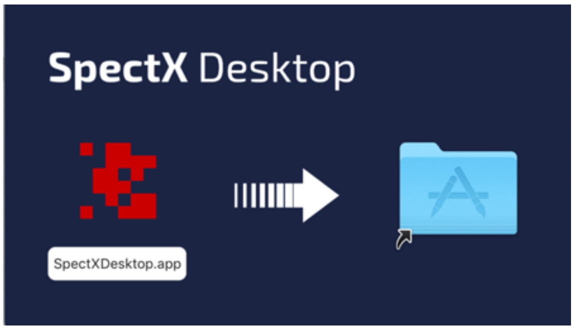
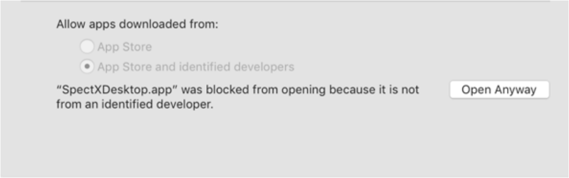


Figure 82. *Application folder.*

Open the System Preferences and click on “Security and Privacy”. It can be seen that the message “SpectXDesktop.app was blocked from opening because it is not from an identified developer”. Click on “Open anyway”.



Allow apps downloaded from:

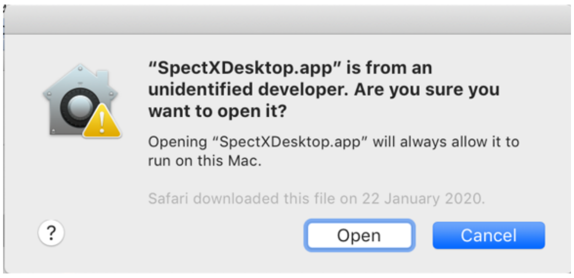
○ App Store

● App Store and identified developers

“SpectXDesktop.app” was blocked from opening because it is not from an identified developer.

Figure 83. *Open the window.*

A dialog box “SpectXDesktop.app is from an unidentified developer.” appears. Click on “Open”.



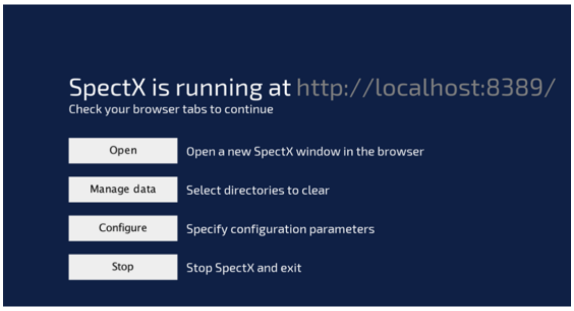
“SpectXDesktop.app” is from an unidentified developer. Are you sure you want to open it?

Opening “SpectXDesktop.app” will always allow it to run on this Mac.

Safari downloaded this file on 22 January 2020.

Figure 84. *Allowed opening the window.*

Switch to SpectX Desktop window. Click on “Open”.



SpectX is running at http://localhost:8389/

Check your browser tabs to continue

Open — Open a new SpectX window in the browser

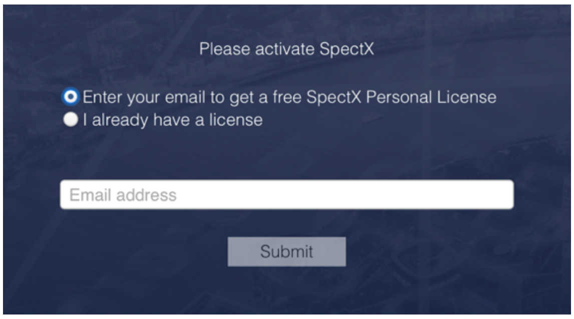
Manage data — Select directories to clear

Configure — Specify configuration parameters

Stop — Stop SpectX and exit

Figure 85. *SpectX Desktop window.*

SpectX activation screen is displayed. Select "Enter your email" and submit your email (note that this requires Internet connectivity).



Please activate SpectX

● Enter your email to get a free SpectX Personal License

○ I already have a license

Email address

Submit

Figure 86. *Email insertion.*

SpectX user interface is displayed in the browser. Installation is finished.
Service was running on the http://localhost:8388.

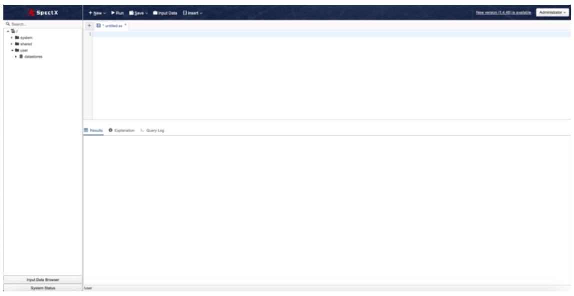
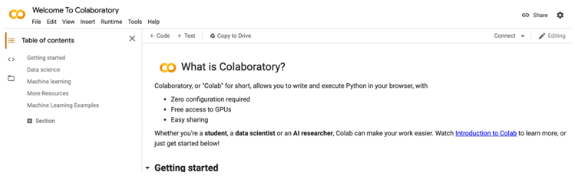


Figure 87. *SpectX interface.*

Appendix 3 - Google Collaboratory setup

First step, is to visit Google Collaboratory page.



What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

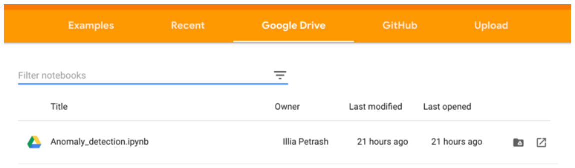
- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch Introduction to Colab to learn more, or just get started below!

Getting started

Figure 88. *Google Collaboratory window.*

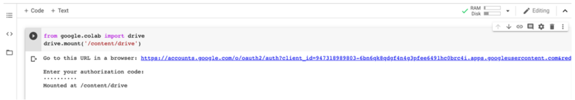
It is needed to upload a dataset to Google Drive so it will be easier to load it to Google Collaboratory.



Title	Owner	Last modified	Last opened
Anomaly_detection.ipynb	Illia Petrash	21 hours ago	21 hours ago

Figure 89. *Uploaded dataset.*

Next step, it is needed to follow the link to be able to get a code:



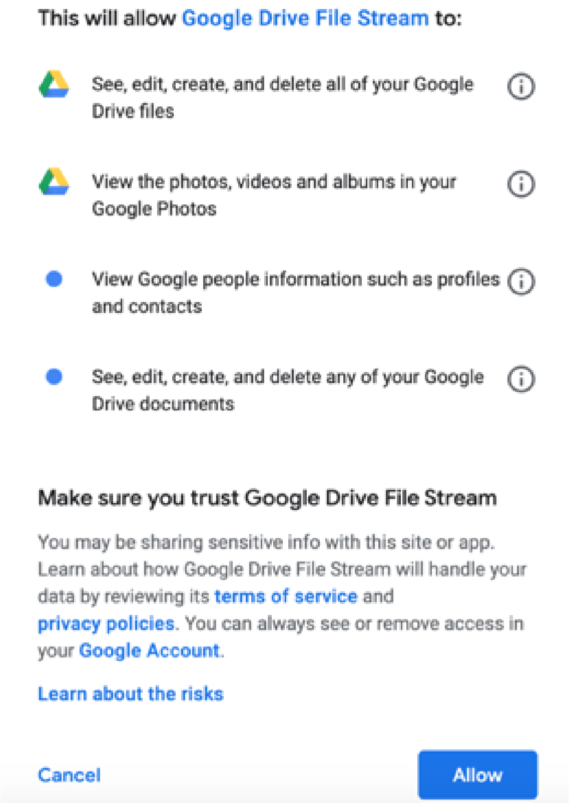
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&red

Enter your authorization code:
··········
Mounted at /content/drive
```

Figure 90. *Validation with the help of code.*

Next step is to copy the code. By inserting this code, the working environment will be available.



This will allow Google Drive File Stream to:

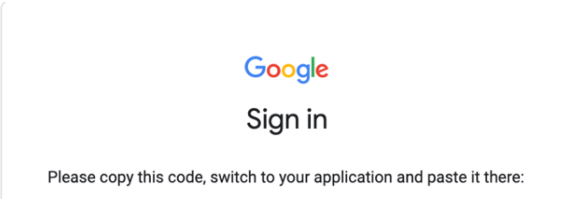
- See, edit, create, and delete all of your Google Drive files
- View the photos, videos and albums in your Google Photos
- View Google people information such as profiles and contacts
- See, edit, create, and delete any of your Google Drive documents

Make sure you trust Google Drive File Stream

You may be sharing sensitive info with this site or app. Learn about how Google Drive File Stream will handle your data by reviewing its terms of service and privacy policies. You can always see or remove access in your Google Account.

Learn about the risks

Figure 91. *Access allowances.*



Google

Sign in

Please copy this code, switch to your application and paste it there:

Figure 92. *Code copying.*