Detecting Anomalies in Log Data from Apache Web Server by Means of Cluster Analysis

by
Mariia Denysenko

Bachelor of Science,
National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute” (NTUU "KPI"), 2014

Merseburg University of Applied Sciences, 2015

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Engineering

in the
Department of Engineering and Natural Sciences
Master Degree of Informatics and Communication Systems

Under the Supervision of Professor Eckhard Liebscher

and Co-supervision of Professor Uwe Schröter

HoMe HOCHSCHULE MERSEBURG
WINTER 2017

Declaration

I declare that, except where explicit reference is made to the contribution of others, that this thesis – **Detecting Anomalies in Log Data from Apache Web Server by Means of Cluster Analysis** – is the result of my own work and has not been submitted for other degree at the Merseburg University of Applied Sciences or any other institution.

Printed Name Mariia Denysenko

Signature _____

Acknowledgement

This thesis is made as a completion of the master education in Informatics and Communication Systems. I would like to thank my university educators for their academic contribution throughout the entire master period.

I would not have been able to conduct this analysis without an excellent guidance and support of my head supervisor and co-supervisor. That is why I would like to express special thanks to Professor Eckhard Liebscher and Professor Uwe Schröter for their time, valuable input, and constructive comments to the thesis.

Finally, I would like to thank my family for being helpful and encouraging.

Mariia Denysenko

Student ID: 19995

Merseburg University of Applied Sciences

December 2017

Abstract

Anomalies are unexpected frequently repeating patterns and alerts in the data which often are not easily classified. They may indicate that the system either contains some malfunctions, or it has been illegally accessed. To eliminate the probability of (successful) cyberattack, and to have a better understanding of the processes happening online, the monitoring of logs should become a common practice for an administrator.

A particular attention has been devoted to the issue of data analysis from Apache HTTP web server. Because the manual inspection can not be applied when it comes to the significant amount of information that should be processed, and because the modern monitoring tools do not provide sophisticated statistical analysis, the use of cluster methods (k-means, k-modes, and k-prototypes) has been explored. As the result, the concept of log files analysis program which consists of three main blocks (data reading, data analysis, and data visualization) has been suggested. On the base of the concept, the program which enables the practical realization of the concept has been created. The results brought by the program have approved the bot statistics; have revealed the anomalies, some of which turned out to be malicious requests; represented the use of the generated fields, such as number of requests per time span, number of special characters and number of requests per IP.

Thesis structure

The **structure of the research** is traditionally represented in the logical parts of the text: introduction, chapters, paragraphs, conclusions as to the chapters, and overall conclusions. The list of references is provided after the main text of the thesis. The thesis comprises 6 tables, 39 figures, 26 listings, and 8 equations.

Introduction is the preambular part of the thesis. It substantiates and affirms the relevance of the theme *"Detecting anomalies in log data from Apache web server by means of cluster analysis".* The objective, the tasks, as well the hypothesis of the research have been stated. The degree of the theme exploration, the methods and principles of the research have been substantiated.

Chapter 1 *"Cybersecurity in global information space"* consists of five main paragraphs and the conclusions. The chapter deals with the main issues of cybersecurity and malicious activities of cybercriminals. In this chapter, the types of cyber-attacks and cyber-attackers have been determined and represented. The latest security breaches have been studied and described. In the end, the set of regulations to minimize the risk of cyber-attacks has been suggested. The areas for further research, namely *IoT, cloud, and mobile, which are considered new frontiers for cybercriminals,* have been suggested, too.

Chapter 2 *"Implication of clustering methods for Apache log files analysis"* consists of five main paragraphs and the conclusions. The chapter deals with the log files and their analysis. The HTTP and the HTTPS Protocols have been studied; the use of Apache HTTP Server has been substantiated, Access Logs and Error Logs have been analyzed; the anomalous activity in log files has been studied, analyzed, and reflected in listings; sophisticated tools for web log files analysis have been studied and represented in figures and listings. Further in the chapter, the definitions of the notions "cluster" and "clustering" have been clarified; the process of conducting the cluster analysis has been described, and the boundaries in which k-methods can be used have been delineated. Unfortunately, not all fields are available now, so there is enough space for future

researches and new ideas. *Most of existing tools do not offer any automated analysis, so administrators have to watch on their own for suspicious activities.*

Chapter 3 *"Concept of log file analysis program"* consists of four main paragraphs and the conclusions. It deals with the concept that found its practical realization in **chapter 4**. The processes of reading, analyzing, and visualizing data have been studied, described, and graphically represented.

Chapter 4 *"Implementation of log files analysis program"* consists of seven main paragraphs and the conclusions. It deals with an application, which would determine anomalous activity by means of cluster analysis. The program, created on the basis of the concept, substantiated in chapter 3 is written in Python 3. Moreover, the chapter describes the realization of the main functionality. All necessary packages and installations have been provided, which has enabled to run the application. The areas for further research, namely *the Elasticsearch could be used of full value, and the clustering methods could be optimized to speed up their work,* have been suggested, too.

Chapter 5 *"Results of log files analysis"* consists of two main paragraphs and the conclusions. It deals with results which have been obtained throughout the research. The three methods, studied in chapter 2 have been tested on the examples of two files. The bots that have been detected during the test have been studied and described.

Conclusions contain the brief summary of the findings and confirm that all tasks, highlighted before the beginning of the research, have been fulfilled and the hypothesis of the thesis have been validated.

The thesis is addressed at the reader with a basic knowledge of computer science and statistics, but not necessarily the cybersecurity.

Table of contents

List of abbreviations

BEC – Business Email Compromise

CIA – Confidentiality, Integrity, and Availability

DDoS – Distributed Denial of Service

DoS – Denial of Service

ETL tool – Extract, Transform, and Load

EY – Ernst and Young Company

GIL – Global Interpreter Lock

HOL blocking – Head-of-Line

HTML – Hyper Text Markup Language

HTTP – Hyper Text Transfer Protocol

HTTPS – Hyper Text Transfer Protocol Secure

IDS – Intrusion Detection Service

IoT – Internet-of-Things

IPS – Intrusion Prevention Service

KPMG – name of the company: **K**lynveld. **P**eat. **M**arwick. **G**oerdeler

PwC – PricewaterhouseCoopers Company

RSS – Rich Site Summary

SQL injection – Structured Query Language

SSL encryption – Secure Socket Layer

TCP – Transmission Control Protocol

TLS – Transport Layer Security

Tor – The Onion Router

UI – User Interface

URI – Information Resource Identifier

URL – Universal Resource Locator

VPN – Virtual Private Network

XSS – Cross-Site Scripting

Glossary

Term (key terms have been selected to simplify the reading)	**Definition** (the reference to the sources from which these definitions were borrowed can be found in the text of this thesis)
anomalies	are unexpected frequently repeating patterns and alerts in the data which often are not easily classified
Apache	the most widely used web server software
authentication	the process of verifying the identity of the individual who tries to connect to the particular network
botnet army	the situation in which the computers act as zombies, executing the commands, given from elsewhere
clustering	an unsupervised method, which allows the algorithms to process and pick out the data, according to its peculiar features or similarity in order to create particular groups (i.e. clusters)
code injection	the type of attack, which exploits computer bug, by executing injected malicious code
credential reuse	the situation when IDs and passwords are stolen from one server in order to be re-used on the others
cyber-attack	a socially, politically or personally motivated web attack, which is put into practice through various means of stealing particular information, services or goods from the targets: fake websites, exploit kits, emails and other network intrusions

cybersecurity	the amount of preventive methods, aimed to protect networks, computers, programs, and data from being stolen, damaged, compromised or violated in any other way and which are realized through the body of technologies, processes and practices
Elasticsearch	a distributed, RESTful search and analytics engine, which stores data in JSON format and provides clients in different programming languages, such as Java, Python, C# and PHP.
elbow method	the method, which is used to determine the optimal numbers of clusters (for k-means clustering)
exploit kit families	software tools, which are used to run on web servers in order to find a 'hole' in the system and to carry out the attack (distribute malware or perform other malicious activities). The attack is usually sequences in four stages: Contact - Redirect - Exploit - Infect
grok	the tool which is used to help understand an extensive software and make the data structured and queryable
insider	a current employee, one who belongs to the company/organization they want to hack
IoT	the inter-networking of physical devices, vehicles, buildings, and other items (also referred to as "connected devices" or "smart devices") that are embedded with electronics, software, sensors, actuators, and network connectivity that enable these objects to collect and exchange data

log files	automatically designed lists of events (actions or processes) that occurred at a particular time and which are relevant to a particular system
malware	different harmful software, which can be divided into three main categories: botnets, spyware and ransomware
man-in-the-middle attack	the situation in which a cybercriminal acts as the server
nonrepudiation	the assurance that a user can not deny performing some action
outsider	a former employee who does not belong to the company/organization they want to hack
patch	a software update, which is installed into a software program to install new drivers, upgrade the software, fix a software bug etc.
perimeter security	the architecture and elements that provide security to the perimeter of an internal network from other networks like the Internet: firewalls, Intrusion Detection and Prevention Systems, antivirus and anti-spam gateways, honeypots
phishing attack	the malicious email (from a friend, a colleague or an internet shop the user dealt with or contacted before; a fake document, which contains a malware attachment; a request to provide some credentials or suggest the victim to follow a misleading link etc.)
Python	a general-purpose interpreted, interactive, object-oriented, and high-level programming language

ransomware	the situation when the art of malware encrypts files or blocks a computer. The secret key is generated, and the victim should pay a ransom to get this key
security policy	a set of rules and trends, which should be followed to minimize risks
spyware	a listening or recording tool that is used to secretly gather the required information about a person or an organization which is later used for personal gain or provided to the third party
three-way handshake	a three-step method, which is used to enable the connection between a client and a server

List of illustrations

List of Figures

List of Equations

Listings

List of tables

Introduction

Nowadays the main concerns of cybersecurity are the data leaks and cyber-attacks. The malicious activity of cybercriminals can be seen at the hackers' black market where they sell the digital products, received due to their cyber-attacks. Machines, computers, mobiles, clouds have become the sweet spot for cybercriminals. That is why the aim of cybersecurity is to detect the illegal intrusion and to protect data from being stolen. To prevent data leaks, continued technological improvement of defenses are made. However, e.g. computers and software as well as applications that are currently available may (for various reasons) contain some malfunctions, which often leads to vulnerability. Thus, according to the statistics, each new software product may contain up to 14 vulnerabilities, and each of them can be used for cyber-attacks [PolitRu17].

A cyber-attack is often realized through manifold vectors of infection. Its goal is to infect the information system to have an unauthorized access to the data needed. A cyber-attack can be detected when some anomalies in log data are spotted: e.g. unexpected frequently repeated patterns or alerts, which are not often easily classified. The log files analysis from a web server at the initial stage helps detect a problem or understand post factum why the problem occurred. Cluster analysis in its turn enables to exploit a niche of the system. Moreover, it reduces, if required, the size of the data and makes it clearly exposed due to its capacity to group the objects, according to their particular features (А.А.Барсегян, pp.159-189) [Bars09]. There are a number of methods, which can be used to conduct cluster analysis. However, k-methods can be considered efficient, as they work even in the conditions when there is not enough information or when there is little information.

All of the above affirms the **relevance of the theme** "Detecting anomalies in log data from Apache web server by means of cluster analysis" and allows stating the **objective of the research**: using k-methods for conducting a cluster analysis to detect possible

anomalies in log files from Apache web server. To reach this goal, it is necessary to fulfill the following **tasks**:

1. To study the problems cybersecurity faces nowadays and suggest the possible ways of the solutions to these problems.

2. To conduct the log files analysis from Apache web server.

3. To clarify the notions “anomalies”, “cluster”, and “clustering” in order to avoid ambiguity further in the research.

4. To sort out and substantiate the use of efficient methods, which can be used for cluster analysis.

5. To suggest a concept of a log files analysis program.

6. To create and test the log files analysis program.

On the base of the objectives and the tasks of the research, the **hypothesis** of the research has been suggested: using *k-means* and its modifications: *k-modes*, and *k-prototype* for conducting a cluster analysis will help process data of different size and nature and detect possible anomalies in log files.

In order to fulfill the tasks of the research and produce reliable results, a significant number of science and field-oriented resources have been carefully examined. The **degree of the theme exploration** can be measured by the resources studied and analyzed. Thus, the latest information security reports have been explored in detail (Internet Security Threat Report 2017, Symantec; Cisco 2017 Midyear Cybersecurity Report, Cisco; Path to cyber resilience: Sense, resist, react. EY’s 19th Global Information Security Survey 2016-17, EYGM; The State of Industrial Cybersecurity 2017: Global Report, Kaspersky lab.). The fundamental science works on mathematics (Z.Huang, 1997; F.Cao 2009), information systems (А.А.Барсегян, 2009) have been critically analyzed. The **principles** of comprehensiveness, absence of bias, integrity, and unity of desk and case studies have been upheld.

CHAPTER 1
Cybersecurity in global information space

1.1 Basic concepts

"More than half of firms (57%) have experienced an attack in the past year and two in five (42%) have had to deal with two or more. Larger companies, particularly those in the US, are targeted most often. The average cost of the largest cybersecurity incident experienced ranges from €22,000 for very small German companies to US$102,000 for very large US companies – somewhat lower than the headline figures often seen." – The Hiscox Cyber Readiness Report 2017 [Hiscox17].

Cybersecurity became an important issue, which, as statistically shown, is being constantly underrated. This chapter will deal with the basic terms of cybersecurity, various arts of attacks and attackers' malicious activities, as well as with the most damaging security breaches and the measures that have been being taken to predict and detect cyber-attacks.

To understand why cybersecurity is important, it is necessary to define what it is, what are its objectives, and what are its functions. In the world of computing, there are two closely related terms – cybersecurity and information security. **Cybersecurity** is the amount of preventive methods, aimed to protect networks, computers, programs, and data from being stolen, damaged, compromised or violated in any other way and which are realized through the body of technologies, processes and practices [WhatIs17]. **Information security**, in its turn, is a set of strategies for securing data, usually summarized to the CIA principle. **CIA** is an acronym, which stands for confidentiality, integrity, and availability [IntroCS17]:

- confidentiality means that information must be read only by the right people;
- integrity indicates that information can be changed only by authorized people or processes;

- availability signifies that information must be always available.

Information security relates to both, online and offline data. **Online data** is an electronic data, whereas **offline data** is hard copy information (e.g. printouts or paper documents). That is why cybersecurity should be understood as a part of information security, too. Both terms still have physical security component. The best protective applications can not do much, if a computer itself is not protected well enough. For instance, if the server is not locked in the different room, which can be accessed only by administrators; or if an employee can put their personal USB sticks into an office computer and spread malicious programs.

It is obvious that each company has a large amount of different data, but not every piece of information has a significant value. The valuable data, which may draw the cybercriminals' attention and become their sweet spot often called **information assets**. This information should be well protected.

Some big companies with complex infrastructure are constantly being targeted. They may experience either a breach which does not appear to do any harm or a damaging and harmful attack. The significant proportion of attacks can be predicted and detected. Some of them can be neutralized, whereas the others can not be bypassed. That is why it makes it useless to spend time and money fighting against the latter. Whatever the type of the attack, the main objective of cybersecurity is to protect the information assets. If the information assets is stolen (e.g. secret information about customers), the company can lose trust of its customers, which would dramatically influence its image and revenue respectively.

One more important concept in online communication and services is **authentication** or the process of verifying the identity of the individual who tries to connect to the particular network [Mies05]. Identification, in its turn, is defined by the fact of recognizing someone or something. In other words, the user proves their identification with the username, but their authentication, with the password.

There are one-, two- and three-factor authentication types [TechTAu17; PCIGur10]:

- **One-factor** authentication is “something that a user knows”. It is commonly done by using login ID and password. This factor is called the knowledge factor.
- **Two-factor** authentication includes possession factor – “something that a user has”. This could be a smart card, a security token or even the biometrical data, as a fingerprint.
- **Three-factor** authentication contains two previous types of factors, including “something that a user is”. This includes inherence factors, as fingerprints, the voice, the hand configuration or the iris pattern.

To get an official permission for getting an access to the system or service, the user should enter their username and password or get authorized. In other words, **authorization** is an official access, which takes place in case of successful identification and authentication [Mies05].

The last, but not the least important concept is nonrepudiation. **Nonrepudiation** [IntroCS17] is the assurance that a user can not deny performing some action. For instance, a user, sent a document with digital signature cannot deny creating it.

Assuredly, all cybersecurity terms can not be examined within one chapter of a thesis, as cybersecurity is the body that constantly grows and develops. New terms and professional jargons appear incredibly fast. Even the cybersecurity-related dictionaries are unable to cover all of them.

The terms, defined above have been highlighted purposely, as they are considered the keywords for cybersecurity and cybercrime, which is the subject of this research. The other terms could be found in the online (e.g. *Tehopedia, Webopedia, Academic Dictionaries and Encyclopedias, WhatIs.com,* etc.) or printed (e.g. *the Threatsaurus (2013)*), security reports (e.g. *Cisco, Ernst&Young, Kaspersky lab., Symantec*, etc.) and profession-related publications (e.g. *The CDM: eMagazine, The Wired,* etc.) [TehoHa17; WhatIs17; ADE17; TechTAu17; Sophos13; Cisco17; Kasp17; ISTR17; CDM17; Kash14]

1.2 Types of cyber-attacks

Cyber-attacks are socially, politically or personally motivated web attacks, which are put into practice through various means of stealing particular information, services or goods from the targets: fake websites, exploit kits, emails and other network intrusions.

In fact, cyber-attacks could be **global** and embrace political, economic, banking and financial areas (e.g. political espionage, economic sabotage, financial heists, etc.) or **partial**, i.e. aimed at selected targets such as (small)businesses, science labs, institutions, credulous individuals, random victims, etc. (e.g. encroachment on the database, theft of the intellectual property and research evidence, elimination of traces of activity, acts of vandalism etc.). But, whatever the reason, any cyber-attack has a damaging effect. That is why the internet users at all levels should be warned of the possible web attacks, whereas the web developers should be aware of how to prevent or detect the attacks. These are some widely used types of cyber-attacks, ranged in the order, according to their malicious activity [ISTR17; Kasp17; Cisco17; IntroCS17; EnigmS17; Tor17; OWASPCI17; NortonMiM17; rapid17; Sophos13; TechTW17; Boult17; ThreatP17; HackS17]:

Malware menace refers to different harmful software, which can be divided into three main categories: botnets, spyware and ransomware [IntroCS17; Sophos13, p.33]. The distributors of malware carry out their activity via spam emails, misleading links, fake software updates and through exploiting the vulnerabilities in web browsers and software [EnigmS17]. It is often difficult to detect malware without delay as *"the malicious code lives in the devices memory and is wiped out once the device is restarted"* [Cisco17, p.39]. Nemucod, Ramnit, Kryptic, and Fareit are the most detected types of malware [Cisco17, p.34]. Among them, Nemucod has been considered the most sophisticated malware. It uses 15 or more combinations of file extensions and file content types. Its delivery mechanisms are thoroughly developed, too. That is why it has been difficult to detect it so far [Cisco17, p.30].

Quite often an average user might not know that their computer participates in a **botnet** army, since the impact to their personal computer is minimal. In this situation the

computer acts as a zombie, executing the commands, given from elsewhere [Sophos13, p.52]. It should, however, be noted that when hundreds or thousands of computers are targeted on the same Web application, they might create a serious damage. For instance, they can be used to hack passwords by trying out different combinations or cause a Distributed Denial of Service (**DDoS**) attack by sending manifold requests to server, making it unavailable or broken. In its turn, Denial of Service attack (**DoS**) can be performed by a single computer. If an attacker uses the anonymity network, such as **Tor** ("The Onion Router") [Tor17], not only the attacker can hide their identity, but also make IP address change constantly. This helps hide the fact, that a single computer sends many requests at a time. For instance, IoT botnets can be set up within an hour. Their distribution rate is remarkable, too: in 24 hours they can infect 100,000 devices [Cisco17, p.39]. Only the Mirai botnet has lately infected *"hundreds of thousands of IoT devices, turning them into a zombie army capable of launching powerful DDoS attacks"* [Cisco17, p.39].

Spyware (listening and recording tools) is used to secretly gather the required information about a person or an organization which is later used for personal gain or provided to the third party [Sophos13, p.46]. Depending on the nature of the information needed, cybercriminals use various software to gain it. Unfortunately, many organizations underestimate the risk of this form of malware [Cisco17, pp. 5, 14].

Ransomware emerged in 2005, is created by Russian criminals [EnigmS17; Sophos13, p.39]. This art of malware encrypts files or blocks a computer. The secret key is generated, and the victim should pay a ransom to get this key. It is an interesting fact that *"at the beginning of 2016, Office documents, containing malicious macros were the most common form of downloader being used in spam campaigns. However, a shift occurred in March and, since then JavaScript downloaders have dominated."* [ISTR17, p.24]

Malvertising became one of the preferred infection vector in 2014 [Kash14]. The criminals were spreading the malicious code in online advertisements. As the result, the visitors of trusted sites, as *YouTube* or *Reuters* got infected with malware. In 2015-2016, the malvertising activity dropped, however in 2017, has continued to diversify [Mayn17].

That is why malvertising issue drew attention of cybersecurity. Therefore, once again it has become necessary to detect, inspect, and analyze all ads on the sites in order to prevent and if necessary to protect websites from malicious advertisers [MedTr17].

Code injection [OWASPCI17] is the type of attack, which exploits computer bug, by executing injected malicious code. This usually happens when program accepts user's input without validating it or setting any format requirements. For example, **SQL injection** (in-band SQLi, Inferential SQLi, and Out-of-band SQLi), which is written in SQL (Structured Query Language) - a standard computer language, which is used to communicate with a database. **SQL injection** manipulates the legitimate data, queries, changes or deletes them. In other words, it manages the database. This can occur on the websites, where a user is prompted to write their credentials. The very famous *webcomic xkcd*, shown on Figure 1.1, describes this art of problem.

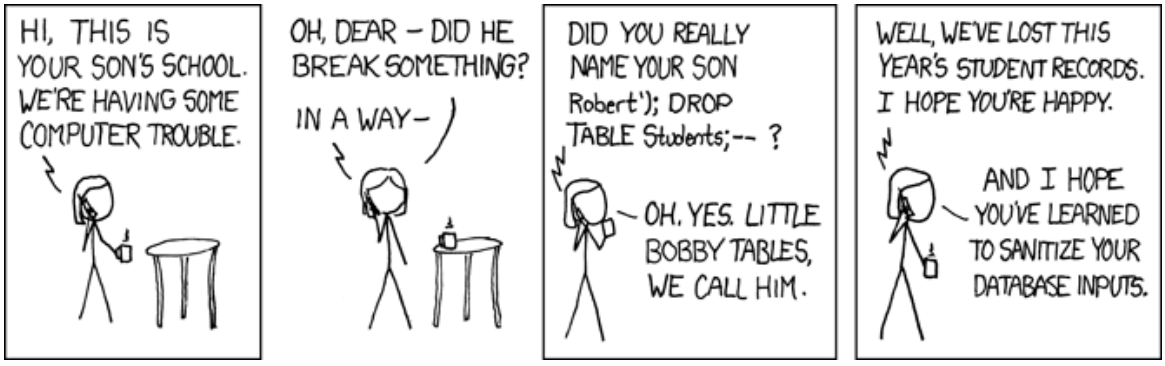


Figure 1.1: "Exploits of a Mom" comic [xkcd17]

XSS or Cross-Site Scripting (Stored XSS, Reflected XSS or DOM-based XSS) [rapid17] can be used to steal session cookies in order to set up as another person or organization; to phish for credentials or redirect the initial user to another web page; to spread malware, etc. In other words, **XSS** may change the server-side code of a trusted website, which later would be performed by an end user. For instance, according to that scenario malicious **PowerShell** scripts have been used in the cyber-attacks on the financial organizations [Cisco17, p.24]. Exactly was PowerShell selected *"to facilitate the download and execution of the final payload"* [ISTR17, p.39].

When a user is involved in the communication with some server the small back-and-forth transactions take place, providing informational exchange. Such communication is called a session, and every session has its own ID. So, **Session hijacking** [rapid17] is an attack, which may end with the stealing of this session ID in order to be able to use it to login to the system instead of the authorized user. In this scenario, a cybercriminal performs on behalf of the real user. The similar scenario has a **Man-in-the-middle attack** [NortonMiM17], however, in this concept, a cybercriminal acts as the server.

The third main category of malware is **phishing attack** whose rate had been in the decline by 2017. There is no reasonable explanation why phishing activity decreases. Meanwhile, there is an assumption that users have become *"aware of the dangers of clicking unknown links or downloading suspicious attachments"*, which apparently has made phishing less effective [ISTR17, pp.25-26]. It appears from the foregoing that phishing works as a typical fishing, i.e. it imitates the activity of trying to catch fish on a hook, in which a victim (a fish) receives an email (a hook) from (as it may seem) a trustful resource, and is caught. The email can be from a friend, a colleague or an internet shop the user dealt with or contacted before. It might be renamed as a document, contain malware attachment, ask for some credentials or suggest the victim to follow a misleading link. Phishing attack is equally aimed at the specific person, and a large group of people [Sophos13, p.37]. For instance, the latest phishing attack has been aimed at the Gmail users. Google claimed that *"0.1% of its one billion users were affected...More than 300,000 corporations were infected by the worm"* [Cisco17, p.57].

Despite the fact that phishing attack has been becoming less effective, spear phishing continues to grow [ISTR17, p.26]. **Spear phishing** [Sophos13, p.44] is customized at a particular victim (e.g. the hacking of some political figure's emails), whereas **whaling** [TechTW17], also known as Business email compromise or **BEC** scams and CEO fraud [ISTR17, p.26; Cisco17, p.22] is aimed at businesses. These scams rely on spear-phishing emails, i.e. do not require any technical expertise, nevertheless, they can seriously undermine the work of the companies involved [ISTR17, p.26]. Thus, due to BEC fraud between October 2013 and December 2016, cybercriminals managed to steal USD 5.3 billion [Cisco17, pp. 5].

The Internet users should also be thorough when choosing a password for the authorization if they do not want to be scammed. It is a common practice, when the web services ask the customer to enter the password for the authorization, they tend to use the same credentials on multiple services. That may seem comfortable but not secure, because may lead to the **credential reuse**. Thus, when IDs and passwords are stolen from one server, they might be reused on the others. So, if the password is weak or used on many web services, the cybercriminals may find it in hacker's databases, such as *Anti Public Combo List* [ThreatP17] or in special dictionaries, in which the most widely used passwords are stored.

While recognizing that some programs may still have some weak points, giving cybercriminals a chance to get an access to the system, it is necessary to admit that the weakest link in cybersecurity remains a human [Boult17]. Since it is complicated to find a leak in the program, **social engineering** [HackS17; Sophos13, p.41] turned out to be the main tool. Criminals can take their time to explore victim before attacking. Such attack can be an email with a very trustful content, which the victim will certainly believe in. It might contain the malicious program or just ask for some personal data.

Based on the foregoing, it could be argued that there are a lot of types of cyber-attacks. Some of them are in the decline, the others, on the contrary, are on the rise. Nevertheless, all of them can not be covered in this paper. In this research were defined and studied the most relevant cyber-attacks such as botnet attacks, code injections and session hijacking. Other existing threats can be found in *the Threatsaurus* (2013) [Sophos13] and in reports on cybersecurity (e.g. prepared by *Symantec, EY, Cisco, Kaspersky lab.,* etc.).

1.3 Types of cyber-attackers

When a cyber-attack takes place, the first question, which arises is *Which damage was caused?* and the second, *Who is the attacker?* While determining the type of cyber-attack and its damage is the matter of time and technical expertise, unmasking the culprit is practically impossible. Firstly, *"cybercriminals are prepared to behave in ways that most*

of us can't understand" [EY17, p.11]. Secondly, the skillfully committed crime can not be investigated, provided the criminal did not make mistakes or was not turned in by the accessories. At the same time, the criminological profile of the attacker allows anticipating the reaction of a breaker and planning security measures [VasRu16]. In fact, cyber-attackers can be divided into two main categories – **outsiders** and **insiders** [HanDo14], which means that they either do not belong to the company/organization they want to hack, or, vice versa, they are the current or former employees. Unfortunately, among all companies, which experienced data breaches, internal actors had been responsible for 43% of data loss, and the half of them had been intentional [Seals15].

The cybercriminals are often called hackers. The term **hack** (as an action) in the world of computing has a number of different definitions. It may mean *"to illegally enter a computer system"*, *"to write computer programs for enjoyment"* or to produce code *"that non only accomplishes the task, but does so in an efficient and unique manner"* [MerWeb17; Etym17; TehoHa17]. In other words, the term **hack** refers to the process of writing or modifying a program in a skillful or clever way [DictHack].

Simultaneously, the term **hacker** (as a person) describes an individual of skillful programming ability, a *"creative prank"*, the *"one who gains unauthorized access to computer records"*, *"one who works like a hack at writing and experimenting with software, one who enjoys computer programming for its own sake"* [Etym17]. In the light of the foregoing, hackers are not always considered malicious. In the programming community, the hackers' profiles depend on the objectives of their activity, which in its turn can be malicious, good or neutral. For instance, *Bridewell Consulting and HackTub* website distinguish from seven to nine types of hackers respectively [Brid17; Cyb16]. However, bearing in mind the hackers' professional expertise as well as the nature of their intentions and the methods of hacking they use, it would be more appropriate to divide hackers into **three groups: ethical hackers, crackers,** and **others**.

Thus, the first group comprises **ethical hackers**, the type of programmers who help find and improve leaks in programs and systems. They check different web pages for vulnerabilities, and in case of successful system breach, tell the owner of the page which

should be changed. Black Hat hackers, also known as **crackers** belong to the second group of hackers. They are skillful programmers who steal confidential information for financial gain. This type of criminals can also attack the web servers in order to insert the code and redirect visitors to some different pages. The third group embraces the hackers with different motivations and computer literacy. For instance, **script kiddies** are the people who are not good at programming. They usually use the written malware to play around. Their most common attack is DoS attack, which can still cause a serious damage to businesses. They are difficult to track down, however, it is estimated that there are millions of script kiddies around the world [Micros16]. **Hacktivists** always pursue political or social aims and use the means of cybersecurity to express their views [Sophos13, p.28]. They are the fighters for the human rights, freedom of speech, or freedom of information. Their victims are usually famous people, politicians and social media. The last, but not least are **digital criminals** [GioCar17]. This is the newest term, which became widespread after the first ransomware was created. The digital criminals ask their victims for money.

In conclusion, it is necessary to admit that the data, used in this research was borrowed from the university server. Although it may not cover the interests of hacktivists, it can be a target for so-called beginning script kiddies, some inquisitive students who would like to try their hand at hacking the system; ethical hackers, crackers or digital criminals. The purpose is therefore not to spot some attack and estimate its type, but at least to find some anomalous activity.

1.4 Current security breaches

Cybercriminals have at their disposal various kits and tools to spread threats. Besides, the criminals have a pretty inventive mind which enables them to use different tactics to stay unidentified. They also have their preferred infection vectors. Despite the fact that the cyber- attackers masquerade, their malicious activity over the course of 2016, altered. They demonstrated a considerable volatility. Thus, during the year, the criminals were

observed to actively use many **exploit kit families** - software tools, which are used to run on web servers in order to find a 'hole' in the system and to carry out the attack (distribute malware or perform other malicious activities). The attack is usually sequences in four stages: Contact - Redirect - Exploit - Infect [Ekit17].

There is a number of exploit kits well-known in the world of computing: the Angler exploit kit, the Neutrino exploit kit, BlackHole, Nuclear, Spartan, Rig, Magnitude to name a few. They have been used for various purposes. Although by the end of 2016 most of these kits had largely disappeared [Cisco17, p.18], it would be grossly incorrect to argue that the activity of the exploit kits completely disappeared. They rather changed their business model. Though, their activity continues to fall to *"almost non-existent levels"* [ISTR17, p.34].

The disappearance or decline in malicious activity of the most detected exploit kits must have had some reasons. For instance, BlackHole's targets were the versions of the Windows OS and the applications, installed on Windows platforms, but with the appearance of the new browser (Edge), whose security architecture is currently well-protected, the activity of the exploit kit dropped [Ekit17; ISTR17, p.35]. What is more, its author and distributor was arrested in Russia [Cisco17, p..9]. The Nuclear kit is believed to have disappeared, because its technology had been revealed [ISTR17, p.34]. Angler's activity dropped with the arrest of the people, allegedly connected with the Lurk banking fraud group [ISTR17, pp.34, 58]. The users of Spartan decided 'to retire' the exploit kit, because it must have stopped being considered a reliable option or they decided 'to lie low' for a while [ISTR17, p.34]. The relatively high level of the Magnitude exploit kit's malicious activity in the course of the year could be explained by the fact that it was used not only to deliver ransomware, but also malicious ads on *Yahoo* website [Ekit17]. Admittedly, until recently, advertisements have not been of concern to cybersecurity [MedTr17]. One more probable reason why the activity of exploit kits significantly decreased could be the reluctance of some criminals to buy an exploit kit as a service. In additional, exploit kits require maintenance of a backend infrastructure, which is to some extent cumbersome and time consuming [ISTR17, p.34]. Whatever the reason, the

criminals preferred to shift to an easier and more reliable infection vector - emails [ISTR17; Cisco17; Kasp17].

The most preferred targets of attackers have been technology- and business-related websites [ISTR17; Cisco17; KaspRu17]. The third, most frequently 'visited' websites are blogging-oriented. Interesting fact, according to the information provided by *Symantec* (2017), the interest of cybercriminals to hosting-, health-, and gambling-related websites has significantly increased (table 1.1) [ISTR17, p.35].

Table 1.1

Classification of most frequently exploited websites

(according to the Internet Security Threat Report, Symantec - April 2017)

Rank	**Domain Categories**	**2015 (%)**	**2016 (%)**	**Percentage Point Difference**
1	Technology	23.2	20.7	-2.5
2	Business	8.1	11.3	3.2
3	Blogging	7.0	8.6	1.6
4	Hosting	0.6	7.2	6.6
5	Health	1.9	5.7	3.8
6	Shopping	2.4	4.2	1.8
7	Educational	4.0	4.1	< 0.1
8	Entertainment	2.6	4.0	1.4
9	Travel	1.5	3.6	2.1
10	Gambling	0.6	2.8	2.2

The decreasing percentage point in the domain of technology could be explained by the increased number of **bug bounty programs** [Bacc17] or vulnerability rewards programs, which are used to discover and report software bugs. Constant security researches also

contribute to browser exploit protection. In relation to this, browser vulnerabilities drops. Thus, the security architecture of Explorer, Firefox, and Safari has become safer. Clearly, that Google Chrome suffered more than the others over the course of 2016, but it had managed to return to the 'normal' levels by now.

In May 2017, **WannaCry ransomware** attack targeted more than 200,000 industrial control systems in 150 countries around the world [Kasp17, p.4; Cisco17, p..46; SciShow17]. The software used security flaw in Windows XP operating system, encrypting files and asking for a ransom to unlock them. Victims were suggested to transfer 300 or 600 $-equivalent to the bitcoin wallet. Most of the large companies that were affected by WannaCry ransomware, had their important data backed up, so they did not have the necessity to pay ransom. An interesting fact that after having infected a significant number of computers, attackers managed to earn, i.e. to steal surprisingly small amount of money – circa USD 100,000. Alerted by the cyber-attack, Windows released patches for the versions of the operating systems that they still support as well as for the older ones. WannaCry virus teaches, how important it is to install patches and back up the important data.

Deloitte, one of the "big four" companies, which along with EY, PwC and KPMG companies offers audit and consulting, has recently faced a cyber-attack, too [DelB17]. The company must have drawn the criminals' attention because among other they also offer cybersecurity advisory to some large banks and companies. So, in March 2017, Deloitte found out that their own email system had been hacked. Although they claimed that the breach had not given out much information, it was believed that the attackers had had access to their systems since October or November 2016 [Guard17]. The attackers got an access to the administrator's account, which opened them the way to names, emails, passwords and other private information about Deloitte's employees and clients. This is believed to have happened because the account had only a one-factor authorization. *"We remain deeply committed to ensuring that our cybersecurity defences are best in class, to investing heavily in protecting confidential information and to continually reviewing and enhancing cybersecurity. We will continue to evaluate this matter and take additional steps as required."* – Deloitte's spokesman stated [Guard17].

In July 2017, **NonPetya ransomware** attack infected computer systems of a number of organizations in Ukraine as well as in other European countries. In October 2017, **BadRabbit**, the ransomware, tied to NonPetya, paralyzed hundreds of computers in Russia, Ukraine, Turkey, Bulgaria, and Germany [Green17]. This fact proves that cybercriminals carefully plan their attacks.

The least prepared for the cyber-attacks turned out to be small-businesses, especially true mom-and-pops whose staff is not numerous and who often do not hire security-focused employees or consultants, who buy unreliable antivirus software, use the same devices at home and work, and do not create multiple user accounts to prevent insiders attacks [Prak17; Cisco17, p.62]. One more vulnerable target is the healthcare. The thing is, that medical devices have been easily hacked, because they *"are not designed or built with security in mind"* [Cisco17, p..42]. **The Internet of Things (IoT)** is vulnerable, too. It offers increased opportunities for business collaboration and innovation, but at the same time, presents the security risk to organizations and users [Cisco17, p.39]. The risk is considered possible, because **IoT** is *"the internetworking of physical devices, vehicles, buildings, and other items (also referred to as "connected devices" or "smart devices") that are embedded with electronics, software, sensors, actuators, and network connectivity that enable these objects to collect and exchange data"* [Cisco17, p.65]. What is more, IoT encompasses information technology, operational technology and consumer technology [Cisco17, p.65], which covers the malicious interests of cybercriminals and makes IoT devices appealing. Among the other targets are oil and gas companies and manufacturers of industrial products [Kasp17].

Many companies have lately demonstrated their preliminary preparedness for the possible cyber-attacks [Kasp17; EY17, pp.13, 72-81]. Thus, 86% of companies claim that they have cybersecurity programs and follow industrial cybersecurity policy [Kasp17, p.10]. The statistics, provided in *EY's 19th Global Information Security Survey 2016-17* [EY17] indicates that comparing with 2015, the organisations are more aware of and more concerned about vulnerabilities and threats they may face [EY17, p.13]. They have taken some measures: conducted security awareness trainings for their staff and ensured improved monitoring of the remote and wireless access to their networks [Kasp17, p.11;

EY17, p.13]. On the one hand, it may be considered effective preventive measures, on the other, being well-prepared for the external threats is not enough, as the internal threats are more dangerous [Kasp17, p.8; EY17, p.13]. Although the companies understand that the cyber threats are real, some of them underestimate their consequences. For instance, some companies do not take any measures when a breach, which does not appear to do any harm happens. They do not assume that cybercriminals can make 'test attacks' and lie dormant after a breach, planning a real attack [EY17, p.9]. Thus, only 29% of the companies are very well-prepared for the attacks; 54% believe that they are well-prepared, whereas poorly prepared 16%, and unprepared - 1% [Kasp17, p.10]. Summing up it's necessary to admit that the companies will be prepared for the attacks if they sharpen their senses, upgrade their resistance to attacks, and react better [EY17, p.3].

The third cybercriminals' sweet spot is blogging. Due to the situation in which Brian Krebs, the famous American journalist and investigative reporter found himself, some methods of cyber-attackers can be revealed [MTMuen16]. Brian Krebs has a daily blog [KrebsSec17], where he covers different cybersecurity issues. One of the areas of his interest was the **Darknet** – a part of the network, which has an illegal content. While investigating the resource, Brian Krebs got an access to the Russian cybercrime site, called *ssndob.ru*, which sells the credit cards' data [KrebsR13]. He made a post about his research and soon got in trouble. Some cybercriminals mailed him the heroine to accuse him of drug peddling. After that, they faked a phone call from Krebs's neighbor and called the police. *"When I opened the door to peel the rest of the tape off, I heard someone yell, "Don't move! Put your hands in the air." Glancing up from my squat, I saw a Fairfax County Police officer leaning over the trunk of a squad car, both arms extended and pointing a handgun at me. As I very slowly turned my head to the left, I observed about a half-dozen other squad cars, lights flashing, and more officers pointing firearms in my direction, including a shotgun and a semi-automatic rifle. I was instructed to face the house, back down my front steps and walk backwards into the adjoining parking area, after which point I was handcuffed and walked up to the top of the street."* – Brian Krebs wrote in his post [KrebsC13].

Whilst creating the real-life troubles, cybercriminals also organized one of the largest DDoS attack against a single individual [Szol16]. Brian Krebs had a website hosted by Akamai servers. The botnet caused more than 620 gigabits per second traffic, making website unavailable. When the problem was solved, the journalist was told that the same kind of protection would cost between USD 150,000 and 200,000 a year [Akam17], which proves how costly can a DDoS attack turns out to be.

Summing up, it's necessary to admit that the inventive mind and unhealthy ambitions of cybercriminals encourage them to explore the potential of existing and new attack vectors. [Cisco17 p.56; ISTR17, pp.64-75].

1.5 Twelve steps for securing network

Nowadays, being cyber resilient or being able to predict and detect a sophisticated cyber-attack [EY17, p.9] is crucial to personal and professional success. Moreover, cyber resilience should become an integral part of computer users' culture. *"**Cyber resilience** begins with an in-depth understanding of the operational landscape, to know which workflows must be preserved so the organization can continue to operate and safeguard people, assets and overall brand equity, despite the cyber attack"* [EY17, pp. 9-10].

As it has been mentioned before, the attackers use all possible infection vectors to try to find and take advantage of security leaks. Keeping an eye on the executing processes is not enough to eliminate security breaches, especially when it comes to the network, which has a lot of users. However, it would be enough to comply with the regulations, outlined below.

All possible actions, which may help securing the network, can be explained clearly in twelve steps, represented in the cybersecurity course on *PacketHacks.com* [PackHack], and enumerated in *the Internet Security Threat Report* (2017) [ISTR17, pp. 22, 31, 36, 54, 62, 67, 72, 74].

Step 1: Implementing and endorsing a security policy

Security policy can be understood as a set of rules and trends, which should be followed to minimize risks. It also encompasses the descriptions of handling situations in case they may occur [TechoSP17]. Both, authorities and personnel should follow the requirements, mentioned in their organization's security policy. For instance, employees should not be allowed to use the office computer for personal needs (checking their private emails, logging in to social networks, watching online videos, playing online games etc.). It should also be prohibited to insert personal USB sticks into company's computers. On the contrary, keeping the organization's security software up to date, regularly assessing the website for any vulnerability, making frequent backups of important data, deleting unused credentials and profiles, choosing a strong password, regularly changing it, and not reusing it, should be encouraged in order to enhance the staff's organizational culture. These policies (as well as the others) should be documented well, clarifying all rights and responsibilities.

Step 2: Rising cybersecurity awareness and behavior by users

Most people are used to performing tasks in the easiest way, rather than in the most secure. This fact should be taken into consideration, and therefore the danger of not following security policy should be communicated explicitly. That is why each new employee should be informed about the organization's security policy in order to act according to the requirements of the organizational culture. Notably, in European Union *"October is CyberSecMonth"* is the annual campaign, which takes place in October and its goals are to promote cybersecurity among people, and provide helpful resources to protect oneselves online [CSMo17]. This event can be used as a good resource for enhancing personal and professional cybersecurity culture.

Step 3: Providing physical security

The IT Coordinator at Primary Goods Manufacturing, Germany claims, *"Internal threats are more dangerous. We are well prepared against external threats, but what is done internally has a direct path without a firewall in between. The threat originates unknowingly from members of staff"* [Kasp17, p.8]. Keeping that in mind, it would be

safe to store the servers in the rooms with the limited access. The working place, equipped with the computer should be protected, too. Thus, when another employee or a visitor enters the room, they should be supervised by an organization's employee. If the employee, working in this room needs to go out and leave their computer, the computer should be locked. That would prevent the computer from an unauthorized access.

Step 4: Implementing perimeter security

Perimeter security or *"the architecture and elements that provide security to the perimeter of an internal network from other networks like the Internet: firewalls, Intrusion Detection and Prevention Systems, antivirus and anti-spam gateways, honeypots"* [Intyp17] should not be ignored. For instance, router can be configured properly, using Cisco access-lists [CiAL07], which filter network traffic. These lists describe which IP addresses are permitted and denied.

Step 5: Implementing IDS and IPS

IDS stands for Intrusion Detection Service and is designed to detect malicious actions [Junip17]. **IPS** is an Intrusion Prevention Service, which prevents malicious actions. To prevent an action is better than to cope it, that's why IPS should work in the first place.

Step 6: Using Antiviruses

Viruses, worms, trojan horses and other malicious programs may cause damages, that is why every computer should have an antivirus software installed on it. This software should also be updated regularly, because new malware appears every day.

Step 7: Using good password management

As it was mentioned in Step 1, passwords should be strong, i.e. at least 8-10 characters long. They should include both, letters, numbers and special characters. Besides, they should be regularly changed, and it should be forbidden to share the passwords with others as well as to reuse the same passwords on different websites. In the first place, passwords should not be simple [CDM17, p.54]. According to the statistics, 81% of all passwords are alphanumerical and have in average length of 8 symbols [Mitch17]. In the

top 25 most common passwords of 2016 have been the following combinations [Keep16]:

- 123456 (nearly 17% of users use this password)
- 123456789
- qwerty
- 12345678
- 111111

Underestimation of the significance of the secure password can lead to unpredicted consequences: the attackers may hack weak and reused passwords easily. To avoid this, it would be a good thing to use the password manager which would generate a strong password and keep it safe.

Step 8: Eliminating unnecessary services

It would be a good thing to eliminate the number of programs, especially those, which are rarely or almost never used or not supported anymore. As to the applications, only the applications from trusted resources can be installed. What is more, it is necessary to pay attention to the permissions, requested by the applications. Only the necessary ports should stay open.

Step 9: Installing patches

Most of applications contain some flaws, which sooner or later are going to be revealed by the vendor or an attacker. To improve the security or keep the door close for a malware attack, a patch should be applied. The **patch** is a software update, which is installed into a software program to install new drivers, upgrade the software, fix a software bug etc. [TehoPat17; Sophos13, p.36]. That is why it should be installed.

Step 10: Managing access control

Different users should have different permissions, which allow them to perform only their job, and nothing more. This can be an access to documents or databases, allowance to read, create, modify or delete data.

Step 11: Securing data in transit

Every data, which is sent over the network, especially untrusted one, should be encrypted. Suspicious-looking emails should be deleted. The emails, containing the information, which does not concern the addressee or concerns them ‘by mistake’, which seems strange, should be left without response and deleted. The unfamiliar or odd links and attachments should not be opened or downloaded.

It would be a good idea to provide an employee (e.g. the one who works from home) with a Virtual Private Network (**VPN**) to connect to the network while being distanced [Sophos13, p.70]. It is also necessary to be thorough while setting up Wi-Fi access: in the workplace it is important to separate private network – for employees, from the public one – for guests.

One more useful recommendation could be to pay attention to the URL layout. Different browsers show next to the https://, on its left, the symbol of the lock, which indicates that the connection is secure. That is why, before visiting a website, especially through the public Internet, the user should pay attention to the type of connection. Fortunately, most web pages implement SSL encryption, which provides better security and prevent a user’s private data from being stolen while transaction performed.

Step 12: Creating backup

Every important data should always be backed up to some external media. The most secure way of keeping data is to keep it on a device, which is never connected to the internet. This can be a simple external hard drive.

Despite the fact that the aforementioned regulations tend to be of a recommendatory nature, they may help improve the cybersecurity awareness and prevent cyber threats.

1.6 Conclusions

On the strength of the theoretical and practical presentations on cybersecurity, the research has been conducted and the findings of the study have pointed to the following:

1. Cybersecurity can not be seen in isolation from cybercrime, as their areas of interest coincide despite the fact that their functions are different. While cybersecurity is aimed at protecting online data, cybercrime seeks ways to damage it for various purposes.

Keeping in mind an indissoluble link between cybersecurity and cybercrime, in paragraph 1.1. a set of key terms that cover the activity of both, cybersecurity and cybercrime have been defined. Among them are *cybersecurity, information security, information assets, authentication, authorization, nonrepudation.*

It is obvious that the meanings and functions of all terms can not be examined within the framework of one paragraph, that is why the up-to-date cybersecurity-oriented sources (online and paper dictionaries as well as cybersecurity-related publications) have been explored and later referred to in the study.

2. Having studied the motives of the cybercriminals' malicious activity, *the types of cyber-attacks* have been represented in paragraph 1.2. The findings have revealed that the attacks can be *global* and aim at politics, economics, banking and finance or *partial* and aim at selected targets (small-businesses, science labs, individuals, etc.). They can be of different types, too. Among the most detected cyber-attacks have been spotted the following: *malware menace, spyware, ransomware, malvertising, code injection, phishing, spear phishing, BEC*. To highlight these types of attacks, the analysis of dictionary articles, online and paper materials on cybersecurity has been made.

3. Usually a cyber-attack reflects the nature of its actor. Although it is almost impossible to reveal the identity of the attacker, their criminological profile allows anticipating the reaction of the actor and planning security measures. Thus, it has been found that the cyber-attackers can be either *outsiders* or *insiders*. They can be bad, neutral or good. With respect to this as well as to the professional expertise of the actors, in paragraph 1.3,

three groups of attackers or hackers have been represented: *1) ethical hackers; 2) crackers; 3) others (skript kiddies, hacktivists, and digital criminals).*

4. Paragraph 1.4. has had a strong focus on the issue of security breaches. The critical analysis of the security reports (*Cisco 2017 Midyear Cybersecurity Report, The State of Industrial Cybersecurity 2017, EY's 19th Global Information Security Survey 2016-17, Internet Security Threat Report 2017*) has shown that the activity of the *exploit kits (Angler, Neutrino, BlackHole, Spartan, Magnitude)* dropped; that the cybercriminals shifted to the easiest and more reliable infection vector - *emails*; that the sweet spots of the attackers in the course of the year 2016 and at the beginning of the year 2017 were *Technology*, *Business,* and *Blogging* (see Table 1.1.). The most detected attacks were *WannaCry, NonPetya, BadRabbit.* They have been thoroughly analyzed. Also, it has been clarified that *IoT* is considered a new vulnerable target.

5. With respect to the materials, studied and analyzed in paragraphs 1.1, 1.2, 1.3, and 1.4, *twelve steps for securing network* have been formulated in paragraph 1.5. They are as follows: *implementing and endorsing a security policy; rising cybersecurity awareness and behavior by users; providing physical security; implementing perimeter security; implementing IDS and IPS; using antiviruses; using good password management; eliminating unnecessary services; installing patches; managing access control; securing data in transit; creating backup.*

Although the aforementioned regulations tend to be of a recommendatory nature, they can be used as a checklist or guidance for both, administration and staff.

The overall progress of the research, conducted at this stage has suggested that IoT, cloud, and mobile, which are considered new frontiers for cybercriminals could become the objects for the further researches in the area of cybersecurity and cybercrime.

CHAPTER 2

Implication of clustering methods for Apache log files analysis

2.1 HTTP and HTTPS

By 1990, large numbers of computers had been connected together through the Internet, however they could not share information. The idea of making systems communicate was carried out by Sir Timothy John Berners-Lee who in 1990, wrote three fundamental technologies [WebH17]: HTML – Hyper Text Markup Language, i.e. the formatting language for the web; URI – Information Resource Identifier; and HTTP – Hypertext Transfer Protocol, which enables Internet searches.

2.1.1 HTTP request

HTTP is a request-response protocol [Krist17]. It is used to transfer data from a web server to a browser, so users can view web pages [PvsS17]. To get data from a server, a client sends an HTTP request protocol, which has a defined structure. For instance, the web page of the University of Applied Sciences Merseburg was requested. If Google Chrome is used, in the window “Developer tools”, tab “Network” the request and response headers can be read. As an example, this request may look as follows:

Listing 2.1 – HTTP request example

```
GET /aktuelles/ HTTP/1.1
Host: www.hs-merseburg.de
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
Chrome/61.0.3163.100
Accept: text/html
```

HTTP is designed to be simple, readable and user-friendly. The first line is an HTTP command, which consists of **method**, **path** and **version of the protocol**. The next lines are headers. The HTTP/1.1 version should also consist of one or more headers, since host is required.

The word “GET” is one of the most common **methods**, which means that a user asks a server to retrieve data from the linked resource from across the web. Since this method can’t make any changes to any resources, it is safe and idempotent [Prot17]. The other methods are [Prot17]:

- OPTIONS – is used to find out a list of methods, which are accepted by a server;
- HEAD – allows to get headers of a file without receiving the whole file itself;
- POST – is used to send data to a server, like a filled out form;
- PUT – is also used to update a resource, but unlike POST request, replaces it completely;
- DELETE – requests a server to delete an identified resource;
- TRACE – tests the path to the target resource;
- CONNECT – is used to start a two-way communication, like opening a tunnel.

Meanwhile, not every request method is allowed by every server.

The **path** in this request is */aktuelles/* - it shows which specific resource a user wants to open. HTTP/1.1 is the **version** of the protocol used. It is a standardized version, which was first published in 1997, and is used since then [RFC97].

Host describes which server is hosting the targeted resource, and keep-alive **connection** asks to create a persistent connection between client and server. **User-Agent** describes the computer software of a client and its characteristics, while **accept** gives a list of allowed formats of a resource.

Eventually, there are many other available fields in the header, which can be checked in documentation.

2.1.2 HTTP response

The HTTP response to the aforementioned HTTP request (see Listing 2.1) would look as follows:

Listing 2.2 – HTTP response example

```
HTTP/1.1 200 OK
Date: Mon, 30 Oct 2017 00:00:11 GMT
Server: Apache/2.2.34 (Linux/SUSE)
X-Powered-By: PHP/5.3.17
Content-Length: 37433
Keep-Alive: timeout=15, max=94
Connection: Keep-Alive
Content-Type: text/html; charset=utf-8

<!DOCTYPE …>
<HTML>
…
</HTML>
```

The first line is a status line, which contains **version of the protocol**, **status code** and **status message**. Following lines are headers, separated with the blank line from the document.

The status code in this response is 200. 200 means *"OK"*; that everything went fine. That is why a user can also see the document at the end of request. Generally, these codes are divided into five main groups [StatC17]:

- 1xx – informational codes. They indicate that the client should respond with some different action.
- 2xx – successful codes. These codes indicate that the request was successful.
- 3xx – redirection codes. The codes appear when the user requests the old file directory, from which the file has been moved.
- 4xx – client error codes
- 5xx – server error codes

There is a number of the status codes. Some of the most common status codes are represented in Table 2.1 [SmartLab17]:

Table 2.1

Common status codes

(after Smartlab Software, 2005-2017)

Status code	Status message	Explanation
200	OK	The request succeeded. Returned information depends on the method, used in request.
300	Multiple Choices	Indicates multiple choices, from which a client may choose. As an example, these can be different variants of page, depending on the language.
301	Moved Permanently	Means that the resource was moved, and the current and future requests should use one of the returned URLs.
302	Found	Indicates that requested resource was temporarily moved to another URL.
400	Bad Request	Means that the request wasn't understood due to malformed syntax.
401	Unauthorized	Used when authentication is needed, but got failed or wasn't provided.
403	Forbidden	Usually appears, when user has no permissions to request the resource.
404	Not Found	Indicates that server couldn't find anything matching to the requested URL. The file doesn't exist at given address.
408	Request Timeout	Indicates that server shuts down the unused connection.
500	Internal Server Error	Means that the unexpected condition of server prevented it from fulfilling request.
501	Not Implemented	Appears when server doesn't support the given request method or doesn't recognize it.
503	Service Unavailable	Indicates that the server is too busy to respond to client. This is a temporary state.

Returning to the HTTP response (see Listing 2.2), it is necessary to point out that the headers of the response contains the information about **date**, i.e. when the request proceeded. In the example provided, it took place on Monday, October 30, 2017 at midnight. The response also contains the information about a **server**, which is in this case Apache/2.2.34, running on SUSE Linux Enterprise Server. The next header of the response says X-Powered-By. **X-Powered-By** specifies the technology, which supports a web page. Here PHP/5.3.17 was used. The length of the request body is indicated in **content-length** header. **Keep-alive** shows two values: minimal time that connection should be kept and the maximum number of requests, which can be sent on this connection before it will be closed [Mozilla17]. The last two headers are of the same type as in the request.

In speaking of security, the fields as X-Powered-By and Server should have been excluded. Firstly, it is not a good practice giving out information about technologies, which were used on the server. Secondly, cyber-attackers might use this information for a malicious activity. For instance, HTTP response (see Listing 2.2) makes it clear that the Apache/2.2.34 version is installed. At the time of writing this thesis, this version was not the latest one, and it was recommended to update to the new generation 2.4.x branch of Apache. This also might mean that the older versions will not be supported soon. What an attacker can already do, is to check the latest problems, found in this version, and try it out, hoping that the newest patches have not been installed yet.

Summing up, it is worth noting the main negative aspect of the HTTP: it can not protect the information that flows from a server to a browser and backward. And this implies that the information can be stolen.

2.1.3 HTTPS

When it comes to sharing the sensitive information (passwords, credit cards information, etc.), the security issue arises. The thing is that the HTTP request is readable, that is why using a public Wi-Fi connection is not safe. While the user is connected, their data can be easily read and stolen.

The Wi-Fi transmits information through the radio waves at a certain frequency – either 2.4 or 5.0 GHz [LifeNog15]. It means that anyone in the room, who has a radio waves-listening software can read and change the information, sent over HTTP request. Another threat – Man-in-the-middle attack (see paragraph 1.2) - the situation when an attacker gets between a client and a server and reads and/or modifies the information transferred. This became especially crucial after Belgian researchers had uncovered the security flaw and published a paper about it [VanPies17]. In fact, the attack works against all modern protected Wi-Fi networks [Krack17].

To protect information from being stolen, the HTTPS was worked out. **HTTPS** or Hyper Text Transfer Protocol Secure is a way to protect the data. It uses the **TLS** (Transport Layer Security) encryption, formerly known as **SSL** (Secure Socket Layer), and authentication to create a secure encrypted connection between a server and a client.

TLS is an asymmetric system, and uses two keys to encrypt communication – private and public keys. Anything that has been encrypted with the public key, can only be decrypted by the private key, as well as vice versa. The goals of the TLS protocol according to the documentation are as follows [RTFM08]:

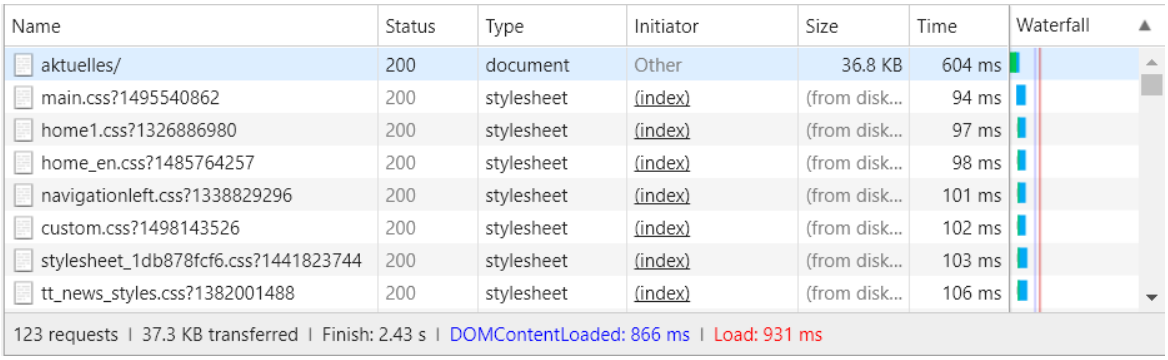
- Cryptographic security: TLS should be implemented to establish a secure connection.
- Interoperability: different programmers should be able to develop compatible applications, using TLS for exchanging cryptographic parameters.
- Extensibility: the new public key and bulk encryption methods should be able to be incorporated.
- Relative efficiency: cryptographic operations should run fast.

To ensure that the client communicates with the real server, the TLS builds a chain of trust. This means that the server identifies itself with the certificate, containing metadata about itself, together with a fingerprint of an encryption key [TLSUd16]. This way, even if a Man-in-the-middle attack takes place, the attacker is not able to decrypt the information, received illegally.

2.1.4 Current issues

Due to the technological development, the web pages both, static and dynamic become more sophisticated. This means that the average amount of transferred data rises constantly. Together with data, the number of requests rise.

While carefully considering Figure 2.1, can be found that to load the homepage of University of Applied Sciences Merseburg, 123 requests were needed (see what is written at the bottom left corner). In Google Chrome Browser it took more than 2 seconds to load the whole page.



Name	Status	Type	Initiator	Size	Time	Waterfall
aktuelles/	200	document	Other	36.8 KB	604 ms	
main.css?1495540862	200	stylesheet	(index)	(from disk...	94 ms	
home1.css?1326886980	200	stylesheet	(index)	(from disk...	97 ms	
home_en.css?1485764257	200	stylesheet	(index)	(from disk...	98 ms	
navigationleft.css?1338829296	200	stylesheet	(index)	(from disk...	101 ms	
custom.css?1498143526	200	stylesheet	(index)	(from disk...	102 ms	
stylesheet_1db878fcf6.css?1441823744	200	stylesheet	(index)	(from disk...	103 ms	
tt_news_styles.css?1382001488	200	stylesheet	(index)	(from disk...	106 ms	

123 requests | 37.3 KB transferred | Finish: 2.43 s | DOMContentLoaded: 866 ms | Load: 931 ms

Figure 2.1: Excerpt of requests to load university's homepage

One of the most time consuming processes is the **TCP**'s (Transmission Control Protocol) **three-way handshake**. **TCP** is a protocol, which allows having the multiple streams of data, sending them to different ports (e.g. the HTTP runs on the port 80, while HTTPS on the port 443). TCP also provides reliability, by ensuring that all packages will be delivered complete and in the right order [Kozi05].

The three-way handshake is a three-step method, which is used to enable the connection between a client and a server. First, a client sends over an IP network to a server a signal to indicate the start of the TCP session. Second, a server, provided that it has open ports, accepts the signal and initiates new connections. Third, the client and a server acknowledge that the process is complete. When the process is complete, the connection is created, and the communication becomes possible [TehoTW17]. In other words, both machines are aware of the newly created connection. If the HTTPS is used, then the additional TLS (Transport Layer Security Protocol) handshake is executed. This

protocol is responsible for the establishment of a secure session. [TLSUd16] and such a communication costs time and resources.

HTTP/1 also does not work well with many requests at once, because of the head-of-line blocking (HOL blocking). HOL blocking is a performance issue, which happens when some queue of packages is blocked by the first package in line [Pope15]. When this happens, a browser opens parallel connections. Depending on browser the number of connections varies from two to eight. This is again time consuming because of the TCP handshakes. Every single request-response round-trip also takes 20-50 milliseconds on a good connection.

The last, but not the least problem is the uncompressed headers. HTTP requests can be easily read by a man, but computers do not need information in a human-readable form. Moreover, a lot of headers are the same in requests and responses. Compressing them would also have a positive influence on the performance. However, these issues are going to be resolved in the newly created **HTTP/2** protocol.

2.2 Apache HTTP Server

For this research the only data, which was available, were some log files from the university's web server. University's website runs on Apache HTTP Server, that's why only this server has been examined in this chapter. Its first version - Apache 0.6.2 - was released in April 1995 [ApH17]. The second version was introduced in 2000, and was a completely redesigned from what it had been in the previous version.

"Apache is the most widely used web server software. Developed and maintained by Apache Software Foundation, Apache is an open source software available for free. It runs on 67% of all web servers in the world. It is fast, reliable, and secure. It can be highly customized to meet the needs of many different environments by using extensions and modules "[WordP17]

What makes Apache HTTP Server attractive, is firstly that the software is an **open source**. The code is available to everyone, and if some vulnerability is found, the web development community works on developing patches. Secondly, it is a **cross-platform**, which means that it can be installed on any operating system.

The generalized structure of Apache Server can be depicted as the Apache Core component, which provides the main functionality: accepts requests, sends responses, and manages possible concurrencies (see Figure 2.2). The system can be extended with different modules – the pieces of code, which provide some additional functionality, executed on the upcoming requests.

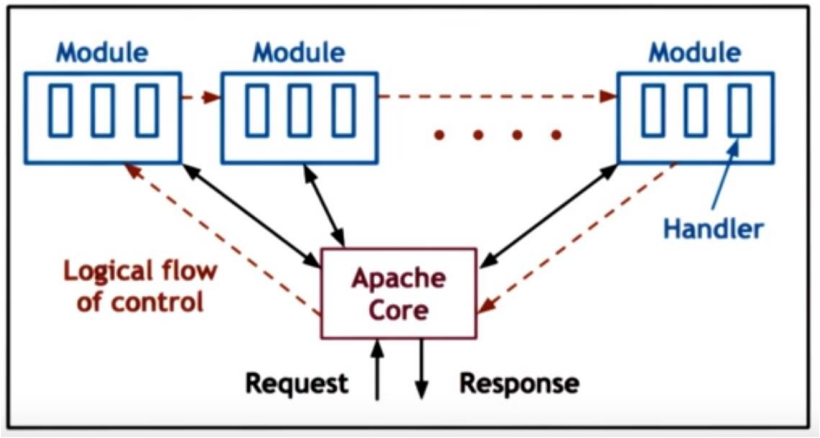


Figure 2.2: Structure of Apache HTTP Server [UdAp15]

The flow of control is so that each request passes through every module, before a response is sent. There is a large list of modules, whose description can be found in Apache HTTP Server documentation. As an example, the *mod_headers* is a module, which provides a possibility to control and modify HTTP response and request headers [ApD05].

Apache HTTP Server 2.2.34 is the final maintenance release of the 2.2 series, which according to the official website is not going to be supported in future. This points out that no new patches will be produced, and this can lead to security flaws, which may occur in the near future. Keeping that in mind, it would be highly recommended to upgrade the server.

2.3 Log files

Log in computer science is a record of event, which occurred while the application was performing. It might contain information about internal errors, for instance, an error while proceeding data. Apache HTTP Server provides two kinds of log files – Error Log and Access Log.

2.3.1 Error Log

Error Log provides the most important information, and is the first place to look at when a problem occurs [ApL17]. The format is relatively free. Log entry can contain the timestamp; module, in which error occurred; severity level, process ID, thread ID if possible; client, who sent request; and description of an error.

The example of Error Log is provided below (see Listing 2.3)

Listing 2.3 – Error Log example

```
[FRI OCT 20 20:07:05 2017] [ERROR] [CLIENT 145.0.124.115]
FILE DOES NOT EXIST: /VAR/SMTH/PICTURE.PNG
```

It says that the error occurred on October 20. The client 145.0.124.115 requested the non-existent file /var/smth/picture.png.

Meanwhile, in this research, this type of log files has not been studied in depth, as it contains less information than the Access log files.

2.3.2 Access Log

Error log files contain only information about errors – it is more helpful in finding solutions to the problems that occur. Access log files contain the successful requests as well, that's why they have been the focal point of this research.

Module **mod_log_config** makes access log be configurable to receive different fields. The whole list of all fields provided can be found in the documentation. Here two specific formats will be studied – common and combined log formats.

When a server supports HTTPS, aside of the standard port 80, SSL port 443 is used. That's why there are two log files – access log and SSL access log. They may have different configurations.

The access log file shown below (see Listing 2.4) was configured with **combined log format**.

Listing 2.4 – Combined log format example

```
241.219.146.191 - - [09/Jul/2017:11:15:03 +0200] "GET
/index.php?id=4693&type=100 HTTP/1.1" 302 242 "-" "Apache-
HttpClient/UNAVAILABLE (java 1.4)"
```

If a token has no value, it is represented by hyphen, as the second and third fields in the example above. The combined log file can be described with a following form (see Listing 2.5):

Listing 2.5 – Combined log format

```
"%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
agent}i\""
```

Every field is signed with the % sign. This is a directive, which tells the system to replace the given characteristics by the values [ModLogC17]:

- **%h** is the remote host. In most cases it will show client IP.
- **%l** is the identity of the user determined by identd. Usually it's not used.
- **%u** is the user name determined by HTTP authentication.
- **%t** is the time the request was received, in the format [09/Jul/2017:11:15:03 +0200].
- **%r** is the request line from the client. ("GET / HTTP/1.1").
- **%>s** is the status code.
- **%b** is the size of the response in bytes.
- **%{Referer}i** is the value of referrer header.[1]

[1] {Referer}is the word that is spelled wrong, but became official name of a header by Apache HTTP Server.

- **%{User-agent}i** is the value of user-agent header.

Looking at SSL access log (see Listing 2.6), it is seen that it was configured with **common log format,** which has the same fields as combined log format, excluding the last two (*%{Referer}i* and *%{User-agent}i*).

Listing 2.6 – Common log format example

```
149.205.57.92 - - [19/Jul/2017:11:15:04 +0200] "POST
/autodiscover/autodiscover.xml HTTP/1.1" 200 –
```

This log means that the client is 149.205.57.92. No identity and user name were logged. The time the request was received is July 19, 2017, at 11:15:04. The request method is POST, used on the resource autodiscover/autodiscover.xml, using protocol HTTP/1.1. The status code is 200, which means OK, and there was no object returned to the client.

Summing up, it is necessary to point out that Error Log and Access Log are stored (depending on the operating system) on an individual web server and contain the information about errors and requests respectively. While Error Log comprises information about the errors that take place at the time when the request is processed, the Access Log encompasses information about requests, which arrive at a server [Skow17].

2.3.3 Important fields

It is crucial to know what to look for in web log files, in order to find some anomalous activity. **Anomalies** are unexpected frequently repeating patterns and alerts in the data which often are not easily classified [Ajan13]. Under **anomalous** is the suspicious activity understood, which might turn out to be the activity of a hacker or a bot.

Timestamp is always included in any log file, because it is important to know when a particular event occurred. This information could be useful for data analysis. For instance, the large number of requests in the short period of time could give a hint of some attack or a bot activity. From this field, the **time difference** can be counted and filtered by the specific IP address. This could be done to find out how often or fast the client was sending requests to the server.

IP address plays a significant role, because it shows who was performing actions. IPs can be checked in the blacklists on the Internet. In the data provided for this research, all addresses have been pseudonymised, so this approach has not been included.

Status code tells the exact status of response. Error **404**, which means *"Not found"* is the best error code to see when someone is scanning server. If log file contains multiple logs in a row with this status and the same IP address, this surely is scanning. However, status code 200 may be problematic. This code indicates that the request was performed, and among these successful requests it might happen that some undesirable client succeeded penetrating system.

The next crucial parameter is **URL path**: which resource a client was trying to request. Here there might be different signals of suspicious behavior. The first thing which can indicate an attack, are some reserved words. For instance, it might be the word **union**, which is often used in SQL injections. It can also be the keyword, as "/etc/passwd", which indicates a Local File Inclusion attempt [InfoSec17]. These also might have evasions, such as simple adding of **multiple slashes**. The following URIs have the same meaning (see Listing 2.7).

Listing 2.7 – Variations of the same URI

```
/etc/passwd
//etc/passwd
/etc/../passwd
/etc/./etc/passwd
```

Cross-site scripting can be recognized by **special characters** and **HEX encoding** [Mey08]. As an example, the request may look like following Listing 2.8:

Listing 2.8 – Example of malicious request

```
"--></style></script><script>netsparker(0x0037E3) </script>"
```

The malicious script is going to be called here. The tags give a hint of attack. HEX encoding might try to hide these special symbols, and the request from the example can also be written like in Listing 2.9:

Listing 2.9 – Example of malicious request 2

```
"%2d%2d%3e%3c%2f%73%74%79%6c%65%3e%3c%2f%73%63%72%69%70%74%3
e%3c%73%63%72%69%70%74%3e%6e%65%74%73%70%61%72%6b%65%72%28%3
0%78%30%30%33%37%45%33%29%20%3c%2f%73%63%72%69%70%74%3e"
```

Both requests are the same, as well as the next one, which is encoded with **ASCII code** (see Listing 2.10)

Listing 2.10 – Example of malicious request 3

```
"--%3E%3C%2Fstyle%3E%3C%2Fscript%3E%3Cscript%3Enetsparker
(0x0037E3)%20%3C%2Fscript%3E"
```

The main symbol to look for is **%** – in both encodings it is used.

Knowing what **extensions** mean, also can be useful. For instance, an *.inc* file contains declarations, headers, functions or other data, used by different programming languages [Fil17].

The last thing to pay attention to in the request, are names of malicious **web shells**. These shells can be uploaded on the web server, to allow remote administration. The four frequently used web shells from the known are [CERT17]:

- China Chopper – a small web shell with several command and control features.
- WSO – "web shell by orb". Can masquerade as an error page containing a hidden login form.
- C99 – another version of WSO, containing additional functionality.
- B374K – a PHP based web shell.

The next important field is **user-agent**. Knowing which software tool was used is also useful. For instance, the "ZmEu" is a computer vulnerability scanner, which searches web servers to attack them [ChP15].

The last, but not the least important is **the size** of data sent to a server. If the user is not downloading a large video to the server, the high number of bytes sent can be an alert of an attack. However, this has not been included in the log files given, and therefore, has not been analyzed.

2.4 Tools for web log files analysis

It is considered that the most natural way of analyzing logs is **manual inspection**. However, it is good only in those cases when files are relatively small, and administrator looks for specific information. For instance, it might be helpful if the administrator would like to check the activity of a user or see the logs with requests to specific resource. Unfortunately, this is not the case when it comes to the Access Logs.

Another approach is **filtering data** over the command line. As a perfect example may be the situation as follows: an administrator would like to see all requests, which received response *"404"*.

The **grep** is a UNIX command, which is also available for Bash on Ubuntu on Windows 10 and other operating systems [TechTGr17]. It allows to read lines from file, which contain some string given. The following command returns the logs, which correspond to the criteria described, and highlights the searchable string with the red color (see Listing 2.11).

Listing 2.11 – Filtering logs with "404" response

```
root@LAPTOP-A3VA3SD5:/mnt/e/Logs/# grep " 404 " www_ssl-
access_log.out
149.205.0.173 - - [19/Jul/2017:14:45:57 +0200] "GET
/typo3/sysext/t3skin/extjs/images/backgrounds/topbar.png
HTTP/1.1" 404 1793
149.205.0.173 - - [19/Jul/2017:15:42:14 +0200] "GET
/typo3/sysext/t3skin/extjs/images/backgrounds/topbar.png
HTTP/1.1" 404 1775
89.6.195.161 - - [19/Jul/2017:22:59:18 +0200] "GET
/studifon/ HTTP/1.1" 404 1624
```

This is only a small part of the requests found, with the response 404. Obviously, all IP addresses were pseudonymized to avoid providing any sensitive information.

To make analysis easier, building dashboards for data visualization, some powerful tools like **Splunk** [Splunk17] for Log Management has been created. It lets user analyze data by drilling it down, to find trends, spikes, and anomalies. It can read and store any machine data, such as configurations, malware analysis, logs from applications and operating system (see Figure 2.3). Splunk comes in two versions – enterprise and free edition. The free version is limited to the volume of logs – up to 500 MB per day [SplunkF17]. Enterprise edition offers some extra capabilities, such as alerting, role-based security, scheduled report delivery [Kum17].

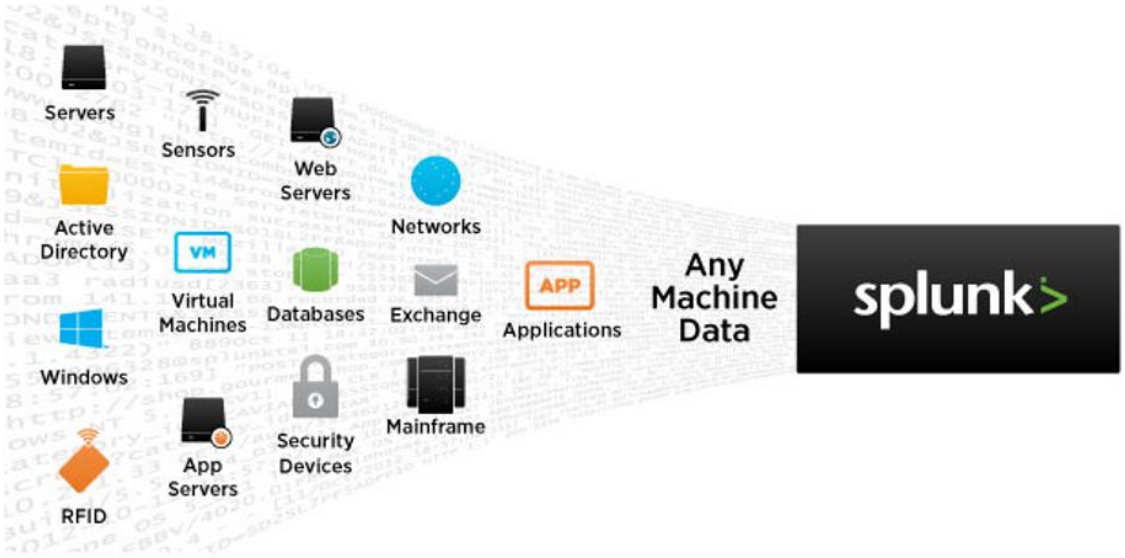


Figure 2.3: Splunk for Log Management [SplunkLM17]

The very similar alternative product is **Graylog**, which comes in two versions – enterprise and open source. It also can read any machine data of any size, provide centralized configuration management for different collectors, visualize log data. Moreover, it lets a user implement statistics for better analysis [Grlo17]. Graylog is written in Java and uses Elasticsearch and MongoDB (see Figure 2.4) as its key technologies, which can be complemented with Apache Kafka or RabbitMQ.

Elasticsearch comes as a kernel of another tool, which will be studied in depth further in this paper. "***MongoDB*** *is an open-source document database that provides high performance, high availability, and automatic scaling* [MonD17]". It stores data as a document in the form, similar to a JSON format.

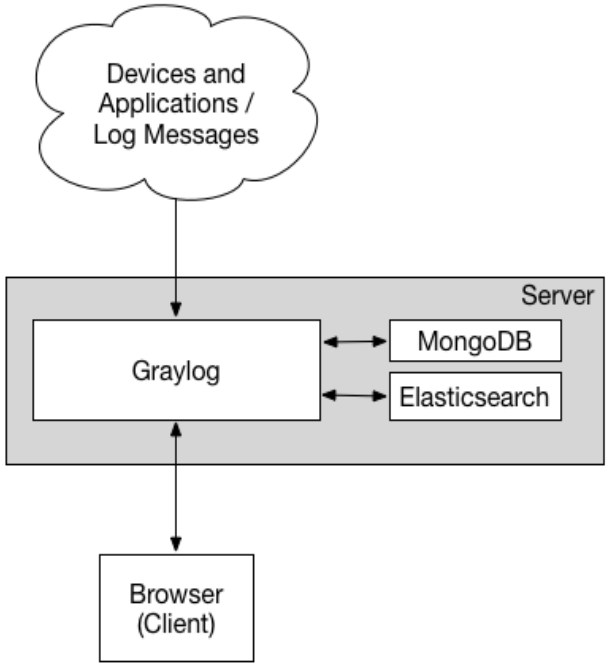


Figure 2.4: Minimum Graylog setup [GrloD17]

Apache Kafka [Kafk16] is a distributed streaming platform, while RabbitMQ [RabMQ17] is the most widely deployed message broker. Both are used for adding some data streaming into setup. This is useful when program reads large information from more than one resource.

Another open source product, which has been implemented in this research, is **Elastic Stack** [Elast17]. It usually consists of three basic components: **Logstash**, **Elasticsearch** and **Kibana** (see Figure 2.5).

In Figure 2.5 there are shown different resources, called "**beats**" – agents, which can be installed on the web server, to send data to Elasticsearch, directly or over Logstash. These

agents ship data from different resources. For instance, Filebeat reads data from log files. If Logstash is busy processing data, the Filebeat receives a signal to slow down its read.

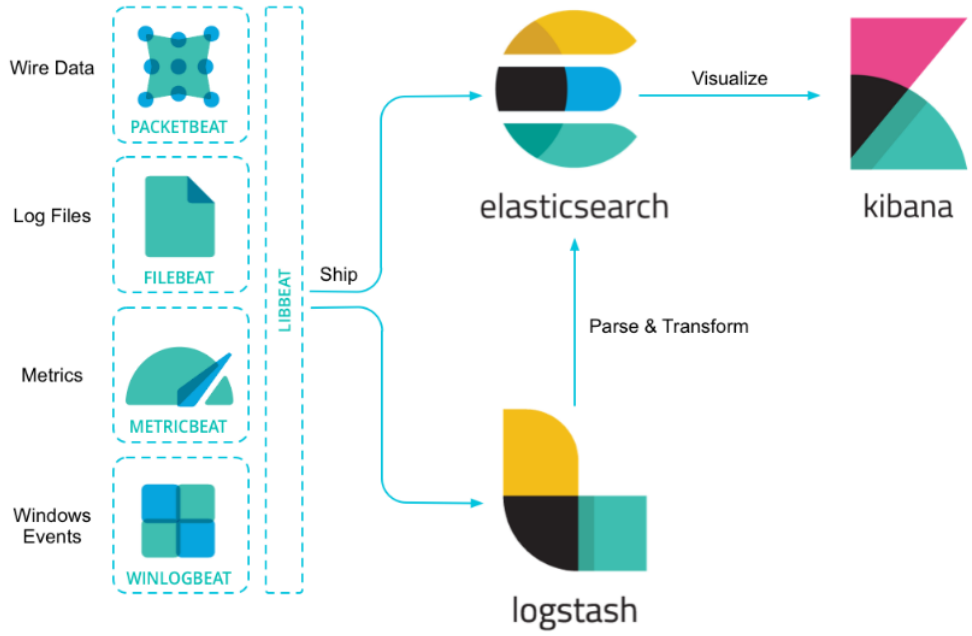


Figure 2.5: Structure of Elastic Stack [ElastO17]

Logstash [Nowad17] is an **ETL tool**, which stands for Extract, Transform and Load. It extracts data, e.g. from log file, transforms it into the understandable for Elasticsearch form, and stores it there or somewhere else. This process can be configured through the configuration file, i.e. the user can say which data should be read, which fields should be taken or ignored, where to store this data and in which format.

Elasticsearch is a "heart" of the whole stack. It is a distributed, RESTful search and analytics engine, which stores data [ElastD17]. It stores data in JSON format and provides clients in different programming languages, such as Java, Python, C# and PHP. These clients are created to provide common ground for all Elasticsearch-relevant code in these languages which makes it easy for integration in different projects. In Listing 2.12 the example of interaction with Elasticsearch from Python in shown. The client is connected to the port 9200 and searches among documents with index *"social-*"* for messages, which contain *"myProduct"*. The result is aggregated into top 10 states.

Listing 2.12 – Getting Elasticsearch information with Python [ElastD17]

```
from elasticsearch import Elasticsearch
esclient = Elasticsearch(['localhost:9200'])
response = esclient.search(
index='social-*',
body={
    "query": {
        "match": {
            "message": "myProduct"
        }
    },
    "aggs": {
        "top_10_states": {
            "terms": {
                "field": "state",
                "size": 10
            }
        }
    }
}
)
```

It is not as robust, as some other NoSQL databases, and therefore is often used as an additional tool. However, in those cases when performance plays a very significant role and robustness is not very important, it serves as a primary store [Braset13]. For instance, the logs are read with Logstash, saved to some file and sent to Elasticsearch. Getting transformed data from Elasticsearch is much faster than reading it from a file. Robustness also does not play the vital role, because Elasticsearch works on the copy of data, which is stored in a file.

Kibana is an open source platform, which enables building analyses and visualizing data from the Elasticsearch to understand it better [KibD17]. It provides variety of charts (line, area, bar, pie, heatmap), tables and maps (coordinate map, region map), which can be shown in a dashboard (see the sample in Figure 2.6). Together with an additional **X-Pack** extension it can provide some monitoring of Elastic Stack, alerting, machine learning and analysis of relationships in data.

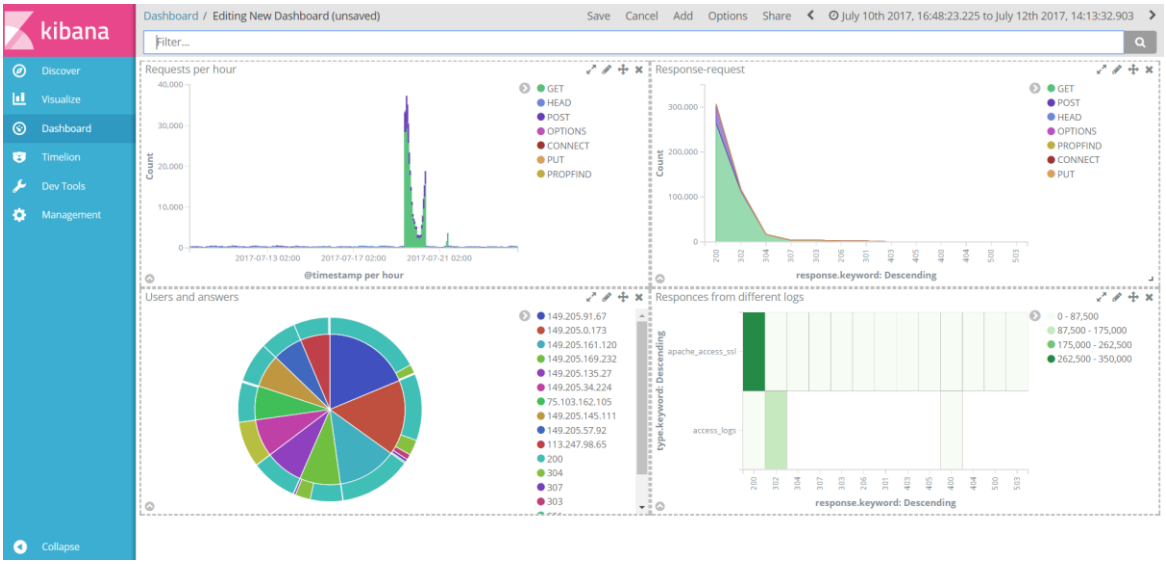


Figure 2.6: Dashboard in Kibana (sample)

The whole stack opens a lot of new possibilities – the very large data can be understood better through visualization; the data reading, and processing become faster. The very new X-Pack plugin offers anomalies detection with the help of unsupervised machine learning. The technology is not uncovered, but the main idea is finding the peaks and troughs with the largest deviation from the mean value. The tool alerts if deviation from this value gets bigger. Meanwhile, the mathematical clustering methods are not implemented in Elastic Stack or other mentioned programs.

An interesting statistic concerning bot-human correspondence has recently been provided in *The annual Imperva Incapsula Bot Traffic Report 2016.* It turns out that only 48.2% of all online traffic belongs to a human, whereas 28.9% is caused by bad bots [Zeif17]. This points out, that more than the half of an everyday requests are sent by machines, and that almost 1/3 of all requests are used for some malicious purpose. Counting the deviation probably might be useful in detecting successful attack. But the data used, from which deviation is counted, contains the bot traffic together with the user traffic as one standard data volume, without distinguishing them into normal and dangerous activities. This makes it also questionable, if deviation from this kind of data can bring much. That is why implementing some mathematical clustering algorithms could be helpful to estimate

the real situation, i.e. to see the possible dangerous activity, which can be furtherly prevented.

2.5 Cluster analysis

Cluster analysis is not a new process, however its scope of application until 1960s was mainly limited by researches in the field psychology, archeology, and biology. Nowadays, the use of cluster analysis encompasses sociology, economics, statistics, historical research. The popularity of this method is obvious: it allows dividing objects according to a set of their particular features, but not according to one parameter. In connection with this, the method of cluster analysis is often called *"a method of multi-dimensional classification"* [Bars09, p.159]. Despite the fact that the term *"clustering"* is widespread in many branches of science, its definition in math is vague [EstCa00]. Various views, expressed on this issue, have made the creation of several types of clustering methods possible. Thus, some mathematicians divide the **clustering methods** into two groups: partitioning and hierarchical [Kaski97], others offer two additional: density-based and grid-based methods [HanKam12, p.448]. It is difficult to admit that one of the methods is more efficient than the other, as each of them are used for different purposes. Moreover, they are based on the modifications of existing algorithms [Bars09, p.159]. Since the main objective of this research is to find an anomalous activity, processing large amount of data, the selection of these algorithms - *k-means* and its modifications: *k-modes,* and *k-prototype* - makes sense [RokMai05, p.342]. The core of the method is to determine the *k centers* (centroids) of the clusters and to group the objects, which are located close to these centers [Bars09, pp. 172-176]. In other words, to create several groups of resembling objects (clusters).

Taking into account the fact that the term *"clustering"* has different interpretations, and in order to avoid ambiguity, it is necessary to clarify its definition, so it can be later used in this research. According to the *Etymological Dictionary* [Etym17], the word "cluster' (in Old En. *clyster*, i.e. *bunch*) refers to both living and nonliving beings or substances

that are close or of the same king and are held together. In comparison, *"cluster"* in social life is a group of people that form a particular group or groups. With all this going on, one person can belong to several groups at the same time. The membership of a person depends on the nature of each separated group. In programming, *"cluster"* can be understood as a logical unity, which consists of connected computers [TechoCC17]. As can be seen, *"clustering"* is a process, which helps a number of things to be grouped. This process can be guided (supervised) or unguided (unsupervised). The supervised clustering is the method, which can be used to process a nonsignificant amount of information. However, when it deals with a significant amount of information, which should be processed, supervised methods may fail to satisfy the expectations. Therefore, while processing a large amount of information of the same or different kind, the unsupervised clustering can be optimal, as the *techniques*, which are used to organize data in classes or groups i.e. clusters according to their peculiar features, do not require either preliminary or in-progress guidance. Moreover, unsupervised clustering is a synchronous process, because the clusters are formed and the objects are grouped simultaneously. The result of the unsupervised clustering is represented in the resulting data model [Bars09, p.160]. In their turn, the *clustering algorithms* are capable to detect the basic structures in data, which helps therefore pick out the data needed [Babushka05]. Summing up the aforementioned, the following definition for clustering can be considered satisfying and therefore will be used in this research.

Clustering is an unsupervised method, which allows the algorithms to process and pick out the data, according to its peculiar features or similarity in order to create particular groups (i.e. clusters).

To be precise, the objective of clustering is to create groups, with minimal differences between data objects from the same cluster. The partitioning algorithms, which are going to be studied further are based on constructing typical objects, which are being considered the **representatives of clusters**. These representatives are often called **centroids**. In Figure 2.7 the centroids are marked as red crosses. The blue points are data objects, which together build two clusters, shown as clouds.

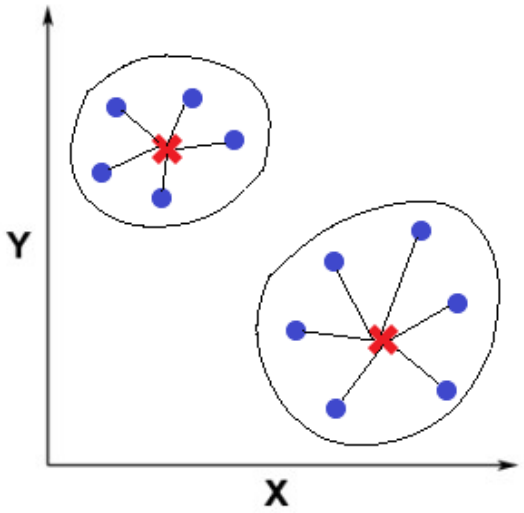


Figure 2.7: Result example of partitioning clustering

The objective equation, which is also called a *cost function*, is as follows:

$$E = \sum_{k=1}^{K} \sum_{j \in c^{(k)}} d(X_j, \mu_k) \tag{2.1}$$

K – number of clusters

X_j – data object

μ_k – centroid of cluster k

$d(X_j, \mu_k)$ – distance function between X_j and μ_k

$c^{(k)} = \{j: X_j \; in \; cluster \; k\}$, where j is index of data object X.

$\bigcup_{k=1}^{K} c^{(k)} = \{1, \dots, n\}$ [Metz16], where n is number of data objects.

The inner sum counts distances between X_j data objects, which belong to cluster $c^{(k)}$, and centroid of this cluster, named μ. The outer sum counts distances for every of K clusters. The result characterizes the quality of the grouping made, and the smaller sum indicates a better clustering. The main mathematical target of cluster analysis is to minimize this cost function for grouping.

2.5.1 K-means algorithm

K-means algorithm is one of the partitioning methods. It indicates that a set of data objects should be divided into clusters, i.e. groups, through relocation. This art of methods starts with some initial partition and relocates instances by moving them to different clusters [RokMai05, p.332]. Every object can be assigned only to one single cluster at a time, and every cluster must contain at least one element.

K-means algorithm works with numerical data, because it is the main target is to minimize the difference between elements from one cluster. This difference is counted as an Euclidean distance between two elements [HanKam12, p.451]. Every cluster is associated with a central point – centroid, and every closest point to the centroid is assigned to the same cluster.

For a computation, k-means method requires two values:

- K – the number of clusters
- Training set $X = \{X_1, X_2, \dots, X_n\}$, where n is the number of inputs and $X_i \in \mathbb{R}^m$

Every input can be defined as a vector of attributes: $X_i = \{x_{i1}, x_{i2}, \dots, x_{im}\}$, where m is the number of attributes.

The first step is initialization of *K* cluster centroids. In the original paper [McQue67] it is offered to choose the points randomly. Nevertheless, choosing initial centroids can influence the result dramatically and bring to the local optimum solutions.

The whole algorithm can be represented as follows [PiechNg13]:

1. Randomly initialize K cluster centroids $\mu_1, \mu_2, \dots \mu_K \in \mathbb{R}^m$.
2. Repeat{

   for *i = 1* to *n*:

   Allocate object *i* to cluster: $c^{(i)}$: $\left\|X_i - \mu_{c^{(i)}}\right\| = \min_k \|X_i - \mu_k\|$

   for *k = 1* to *K*:

$$\mu_k := \frac{\sum_{i:c(i)=k} X_i}{\sum_{i:c(i)=k} 1}$$

} until $c_{old}^{(i)} = c_{new}^{(i)} \; \forall \, i$

The first inner loop executes cluster assignment. $\|X_i - \mu_k\|$ counts the Euclidean distance (Equation 2.2).

$$d(a,b) = \sqrt{\sum_{i=1}^{m} (b_i - a_i)^2} \tag{2.2}$$

The second inner loop moves *k* centroid by counting the average (mean) value of points assigned to the *k* cluster. The method should be run multiple times, in order to determine the optimal clustering. The algorithm stops, when centroids do not change their values anymore. This procedure is summarized in Figure 2.8.

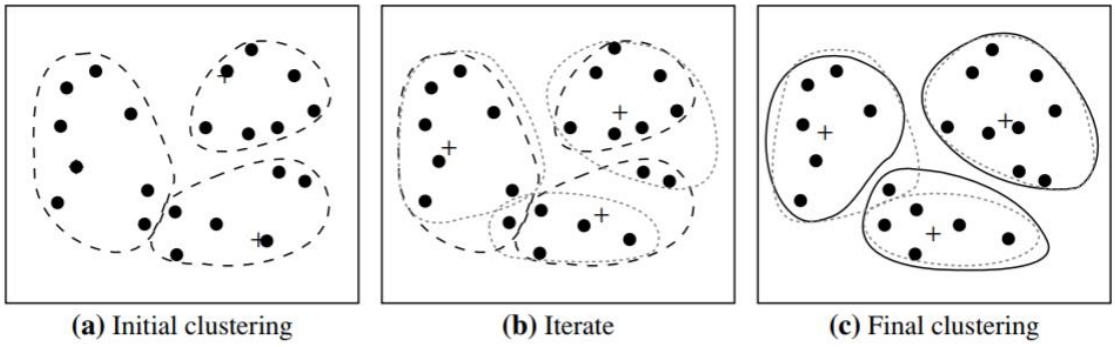


Figure 2.8: Clustering objects with K-means algorithm [HanKam12, p..453]

K-means method is often used on the large data because of the three main reasons, according to [RokMai05, p.342]:

- **Time complexity**: $O(K \cdot n \cdot l)$, where l is number of iterations for the algorithm to converge. This value defines time, which is needed for computer to run the algorithm.

- **Space complexity**: $O(K + n)$, which shows the number of memory cells, which are needed for the algorithm to operate [Zieg04].
- **Order-independency**, which means that the result does not depend on the order of data objects' analysis.

The main weaknesses of this algorithm according to [Yang17] are:

- impossibility of analyzing categorical data;
- the number of clusters *K* should be known in advance;
- the algorithm can't handle outliers well;
- it tends to generate spherical clusters, which makes it unsuitable for non-convex shapes.

2.5.2 K-modes algorithm

K-modes algorithm was proposed by Z.Huang in 1997 as a modification of K-means algorithm. The previous algorithm does not always work well on categorical data, because "*the traditional approach to converting categorical data into numeric values does not necessarily produce meaningful results in the case where categorical domains are not ordered"* [Huang98, p.284]. This implies, that working with categorical data in the same way as with numbers won't necessarily produce good results. The three modifications, made to K-means to get rid of this restriction, and which are used in this algorithm are [Huang98, p.289]:

- usage of simple matching dissimilarity measure instead of Euclidean distance;
- replacement of means with modes;
- usage of frequency-based method to find modes.

The dissimilarity measure can be described as a sum of distances between every element and centroid – the randomly chosen or counted group center [Huang97]. X_i and $Q_k$are vectors of the same size

$$d(X_i, Q_k) = \sum_{j=1}^{m} \delta(x_{ij}, q_{kj}) \tag{2.3}$$

X_i is an element, which is being compared to the Q_k centroid. If both arguments x_{ij} and q_{kj} are equal, then function $\delta(x_{ij}, q_{kj}) = 0$. If not, then 1. As an example, if two requests from log data received response "404", then their dissimilarity equals 0, and they belong to the same cluster.

The dissimilarity measures should be counted for every argument, of every element, for every centroid. Therefore, the complete goodness-of-fit measure, which equals within-cluster sum of distances is [Huang98, p.290]:

$$E = \sum_{k=1}^{K} \sum_{i:c^{(i)}=k}^{n} \sum_{j=1}^{m} \delta(x_{ij}, q_{kj}) \tag{2.4}$$

where K is number of clusters, n is the number of data objects, m is the number of arguments in those objects. $c^{(i)}$ is a cluster, to which X_i is assigned. Q_k centroid should minimize the sum of dissimilarity measures – the lower value means a better clustering.

The whole algorithm can be represented as follows [Huang98, p.290]:

1. Randomly choose K modes: $Q = \{Q_1, Q_2, \dots, Q_K\}$
2. for *i = 1* to *n*:

   Allocate object to the cluster: $c^{(i)}$: $\delta(X_i, Q_{c^{(i)}}) = \min_k \delta(X_i, Q_k)$

   $Q_k = [\max_k f(q_1), \dots, \max_k f(q_m)]$, where $f(q_m)$ is relative frequency

3. Repeat while objects reallocate {

   Retest the dissimilarity of objects: $E = \sum_{k=1}^{K} \sum_{i:c^{(i)}=k}^{n} \delta(X_i, Q_k)$

   Reallocate the objects, to minimize dissimilarity (step 2)

   }

2.5.3 K-prototypes algorithm

K-prototypes algorithm was offered for the first time by Z. Huang in 1997. This algorithm is based on k-modes, but removes its limitation on numeric data, at the same time preserving its efficiency [Huang97]. K-prototypes integrates k-means, and k-modes

algorithms to be used on the mixed data – both numeric and categorical, and it acts in the same way as K-means algorithm, when used on numerical data only [Huang98; Huang 97]. This algorithm is very useful for analysis of log data, because the very important fields have different data types.

The criterion of clustering is often called as a cost function, and in k-prototypes it is defined as a sum of similarity measure for every cluster [Huang97]:

$$E = \sum_{k=1}^{K} \sum_{i=1}^{n} y_{ik} d(X_i, Q_k) \tag{2.5}$$

The notation is the same as in the previous methods. K is the number of clusters, X_i is an input vector with m arguments:

$$X_i = \{x_{i1}, x_{i2}, \dots, x_{im}\}$$

Q_k is a prototype, also a vector with m arguments, for cluster k:

$$Q_k = [q_{k1}, q_{k2}, \dots, q_{km}]$$

y_{ik} is an element of partition matrix Y_{nXl} [Hand81]. It can have different range of values, depending on the type of partition, which can be hard or fuzzy. *Hard clustering* is based on the classical set theory, which means that the element belongs or doesn't belong to cluster [Bal14]. In *fuzzy clustering* the same element can belong to multiple clusters with different membership degrees [Babu05; Bars09, p.176]. In this research only hard clustering have been implemented, therefore $y_{ik} \in \{0,1\}$. If $y_{ik} = 1$, then the object X_i is assigned to cluster k.

$d(X_i, Q_k)$ is a similarity measure, which in case of numerical data, will be Euclidean distance between an object and centroid. For mixed data it will be defined as sum of similarity measure between numerical data and of similarity measure between categorical. The equation suggested by Huang is as follows [Huang97]:

$$d(X_i, Q_k) = \sum_{j=1}^{m_r} (x_{ij}^r - q_{kj}^r)^2 + \gamma_k \sum_{j=1}^{m_c} \delta(x_{ij}^c, q_{kj}^c) \quad (2.6)$$

The first summand is Euclidean distance between numerical parameters. The second summand deals with categorical data and is the number of mismatches between data objects and centroids. If $x_{ij}^c = q_{kj}^c$ then it means that both categories are equal, and therefore $\delta(x_{ij}^c, q_{kj}^c) = 0$. Otherwise it will be equal 1. γ_k is weight for categorical parameter for cluster *k*. If $\gamma_k = 0$ then the clustering result doesn't depend on categorical values. Usually, if no γ_k value is estimated, it is counted as a standard deviation of the numeric values:

$$\gamma_k = \gamma * 0.5 * S^2$$

where γ is an estimated value, and $S^2$is a standard deviation.

Equation (2.6) could be modified, by normalizing the numerical input data. When the data is normalized, the γ_k parameter can be ignored:

$$d(X_i, Q_k) = \sum_{j=1}^{m_r} (x_{ij}^r - q_{kj}^r)^2 + \sum_{j=1}^{m_c} \delta(x_{ij}^c, q_{kj}^c) \quad (2.7)$$

For this case the numerical values should be normalized as follows:

$$\widetilde{x_{ij}} = \frac{x_{ij} - \min x_{ij}}{\max x_{ij} - \min x_{ij}} \quad (2.8)$$

for maximum and minimum over $i = 1 \ldots n$

Centroids can be chosen randomly, but as it has been mentioned before, this may end up with a bad grouping. Therefore, an interesting technique was suggested by F.Cao (2009): to initialize centroids on the basis of the frequency of the attribute values [Cao09]. The idea is to choose an element with the attributes of the highest modes as the first centroid. The next centroid is the next element with the most frequent attribute values, and the largest distance to the first centroid. This approach provides better results, but also has a

larger time complexity: $O(K^2 \cdot n \cdot l)$, which should be considered when a large number of clusters is chosen.

An example below is provided to make the computations with this algorithm clearer. It is assumed, that three data objects *a*, *b* and *c* exist in two-dimensional space (see Figure 2.9).

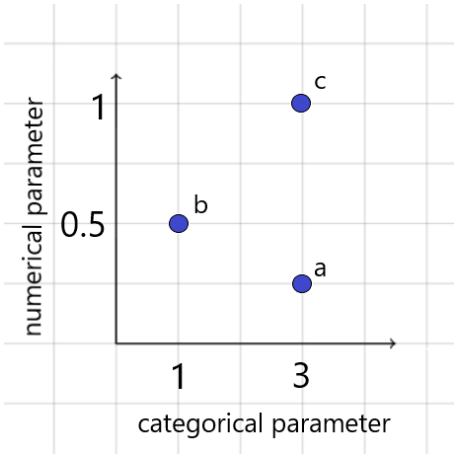


Figure 2.9: Data objects with mixed parameters

This means, each of them has two parameters. These parameters are mixed – the first one is categorical ($x \in \{1,3\}$), and the second – numerical, normalized ($y \in [0,1]$). As long as *a* and *c* are in the same category – 3, their numerical similarity measure equals $d_{num}(a,c) = |1 - 0.25| = 0.75$, and categorical – 0. The total similarity measure equals 0.75.

Data objects *a* and *b* are in the different categories, therefore $d(a,b) = |0.5 - 0.25| + 1 = 1.25$. The similarity measure between *b* and *c*: $d(b,c) = |1 - 0.5| + 1 = 1.5$

If any weight γ is implemented, either to categorical, numerical or both parameters, the results change dramatically. This can make data objects build different clusters every time the weights are changed. Therefore, data normalization is a good way to avoid choosing values for γ, which by the bad estimation can drive to the wrong results.

The k-prototypes algorithm is identical to the one of K-modes.

On the basis of the above, the cluster analysis depending on its comprehensiveness can be made with the use of one of the methods analyzed. The choice of an appropriate method in its turn depends on the nature of the data that has to be processed.

2.6 Conclusions

Chapter two deals with the log files and their analysis. The research conducted within this chapter has allowed making the following conclusions:

1. The analysis of the *HTTP request*, which consists of method, path, and version of the protocol and *HTTP response*, which comprises version of the protocol, status code, and status message has been represented in Listings 2.1 and 2.2. Some of the most common status codes have been described in Table 2.1. The analysis has allowed concluding that the headers X-Powered-By and Server should have been excluded, because they provide information about technology that can be used by cybercriminals in their malicious activity.

2. Taking into consideration the fact that HTTP can be read by attackers, the HTTPS features have been examined. The goals of the TLS protocols have been examined, too. The findings have proved that HTTPS helps create a secure encrypted connection between a server and a client and therefore, should become a preferred protocol in server-client communication.

3. The process of loading a web page has been studied on the example of the HOME Merseburg web page (see Figure 2.1). In view of this, it has been stated that *TCP's three-way handshake* is one of the time consuming processes. The analysis of HTTP/1 has pointed out that it does not efficiently work with many requests at a time. It also has been assumed that HTTP/2 is likely to solve the problems HTTP/1 faces.

4. As the HOME Merseburg website runs on *Apache Server*, this server has been studied in this research. Apache HTTP Server also can be attractive because it is an open source and a cross platform web server software. Its functionality has been represented in Figure

2.2. Eventually, the weakness of the Apache Server, installed at HOME Merseburg server has been spotted: the final maintenance release is not going to be supported in future. This can be a ‘hole’ for hackers. That is why it would be a good thing to upgrade it, otherwise the lack of new patches can lead to security flaws.

5. Error Log and Access Log have been analyzed in Listings 2.3-2.6. Their peculiar features have been described, too. Thus, *Access Log* contains information about requests that come in to the server (e.g. the status of the request; which web pages users visit; which agent is used, etc.). The *Error Log* in its turn, deals with the errors that take place while the request is being processed.

6. The definition of the terms anomalies” have been clarified and the satisfying definitions have been suggested. Thus, in this research the following understanding of the term *“anomalies”* will be used:

Anomalies are unexpected frequently repeating patterns and alerts in the data which often are not easily classified

7. Anomalous activity has been studied in paragraph 2.3.3 and the analysis of the anomalies have been represented in Listings 2.7-2.10. In this respect, *timestamp, IP address, status code, URL path, extensions, webshells, user-agent, the size of the data* fields have been studied in depth. The analysis of logs showed that log files contain a number of valuable information. Not only fields play an important role, but also their combinations, sums, averages.

8. It is obvious that the most natural way of analyzing logs is manual inspection. However, it is useful only when it deals with relatively small files. Large-sized files require advanced ways of inspection. In connection with this, some sophisticated tools have been analyzed in paragraph 2.4. Among them are: *grep, Splunk, Graylog* and *Elastic Stack.* As long as Elasticsearch, which is one of the components of *Elastic Stack*, provides a very fast performing NoSQL data storage, together with different clients in multiple programming languages, its functionality can be used in self-written program. All tools have been represented in Figures 2.3-2.6 and Listings 2.11-2.12.

9. In paragraph 2.5 has been substantiated why implementing some mathematical clustering algorithms could be helpful to estimate the real situation, i.e. to see the possible dangerous activity, which can be furtherly prevented. Considering that various views of the terms *"cluster"* and *"clustering"* can cause ambiguity, the definition of these terms have been clarified and the satisfying definitions have been suggested. Thus, in this research the following understanding of the term *"clustering"* will be used:

Clustering is an unsupervised method, which allows the algorithms to process and pick out the data, according to its peculiar features or similarity in order to create particular groups (i.e. clusters).

The thorough analysis of clustering algorithms: *k-means* and its modifications: *k-modes*, and *k-prototype* has been made. The analysis has enabled to make the following conclusions: *k-means* can be used for processing *numerical data* (e.g. time, bytes, number of special symbols in request), *k-modes,* for *categorical data* (e.g. response, IP address, request, agent), and *k-prototypes,* for *mixed data* (e.g. bytes, number of special symbols, agent). Despite the fact that *k-means* has wide application, it works only with numerical data. As long as the most important fields can be both numerical and categorical, the modifications of this algorithm are implemented - *k-modes*, which works only with categorical data, and *k-prototypes*, which processes mixed data.

10. In the light of the foregoing it is necessary to admit that not all fields are available now, so there is enough space for future researches and new ideas. Most of existing tools do not offer any automated analysis, so administrators have to watch on their own for suspicious activities.

CHAPTER 3
Concept of log files analysis program

3.1 Reading Data

As noted earlier in paragraph 2.5, the existing popular tools for log data monitoring do not provide any mathematical cluster analysis methods. That is why, another application, which would determine anomalous activity by means of cluster analysis should be developed. The designed application should be developed for a single computer, because provided data for analysis originates from the same web server. The other reason is the lack of opportunities of testing on multiple computers. Nevertheless, this application should be able to be compatible with Elastic Stack (see paragraph 2.4), which can deliver data from different computers.

The total concept can be brought together to three main blocks (see Figure 3.1).

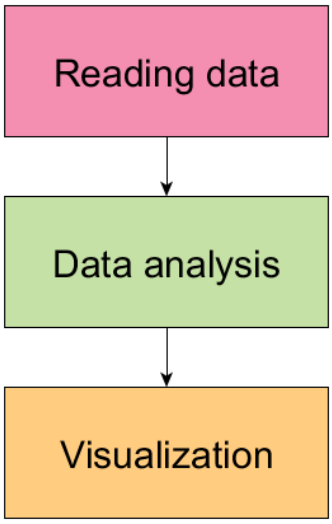


Figure 3.1: Basic concept of log files analysis program

First, the program should be able to read data from log files. Then, this data should be analyzed with the help of clustering methods. Finally, the result of analysis should be visualized.

The log files are stored on the same computer, where a program runs. These files might have different extensions – *.txt or .out*, or even be extensionless, but in any case, they are plain text files, created on a Linux system.

As it was mentioned above, in paragraph 2.2, the university's web server is an Apache HTTP Server, which runs on Linux system. Log files from this server were copied to the notebook, which has Windows 10 installed. The computer has Intel Core i5, four kernels and 16 GB RAM, which means that it should be able to process data fast. All research was not made or tested on the server, so the speed can differ when run on the original system.

The sizes of inspected log files vary from 25 to 350 MB, and even for the standard programs, such as Wordpad or Notepad it takes considerable time to open them. For instance, it takes Notepad approximately 30 seconds to open the 25 MB log file, while loading CPU at 35% across four cores. Wordpad requires less time, but it freezes up when a user reads and scrolls the text opened. This indicates that reading, and processing data should be taken in consideration.

Writing a program, which would in the first place save data to the database, such as SQL or MySQL to name a few, and then read it, is ineffective. Still a large chunk of data should be analyzed at once, which means that it should remain in RAM. That is why writing it to the database while running a program, and then reading it into RAM again does not make sense. Reading a large file line by line slows the program down as well. That is why there can be three different options – if a file is small enough, it should be read without any problems, but if it is large, it should be split or preprocessed in background. A user should be able to choose one of these options (see Figure 3.2).

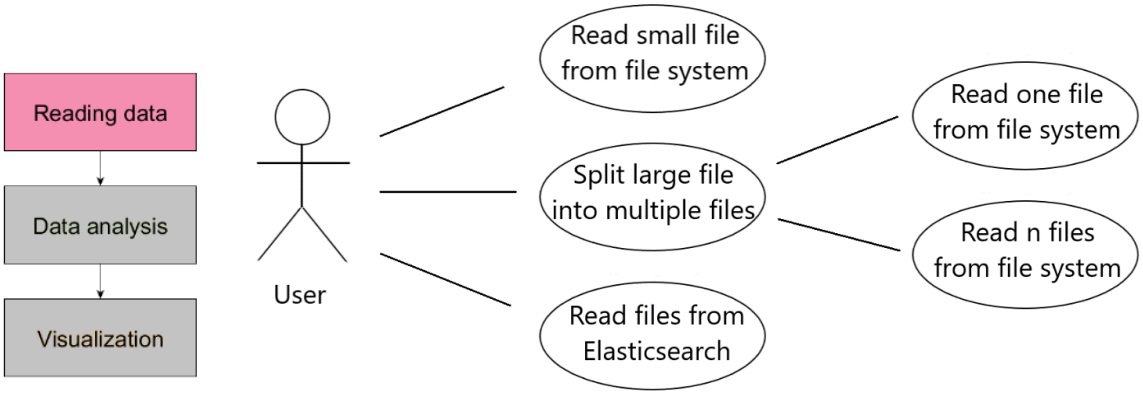


Figure 3.2: Use case diagram – reading files

As it was highlighted in paragraph 2.4, Elastic Stack is a great tool, which can speed up reading and processing information. It is also easy to be used by most of the programming languages. Therefore, giving a user a possibility to read data from Elasticsearch would be a good step towards working with large amount of data.

Splitting large file into smaller ones would allow program to read a small data chunk, or to read multiple chunks simultaneously in different threads or cores. This would improve efficiency and give user more possibilities. If, for example, for some reasons he can not/does not want to install Elastic Stack, he will be still able to read large files from file system. All of the foregoing can be accumulated in the diagram (see Figure 3.2.1).

The diagram shows the process of reading the data from a log file in details. As it can be seen, this process goes through three stages. Each stage describes the actions of the "participants" of the process: the program and the user. At stage 1, the reading data is selected. At stage 2, the user chooses one of the options to read it. After these steps have been taken, the reading of the data from a log file becomes possible.

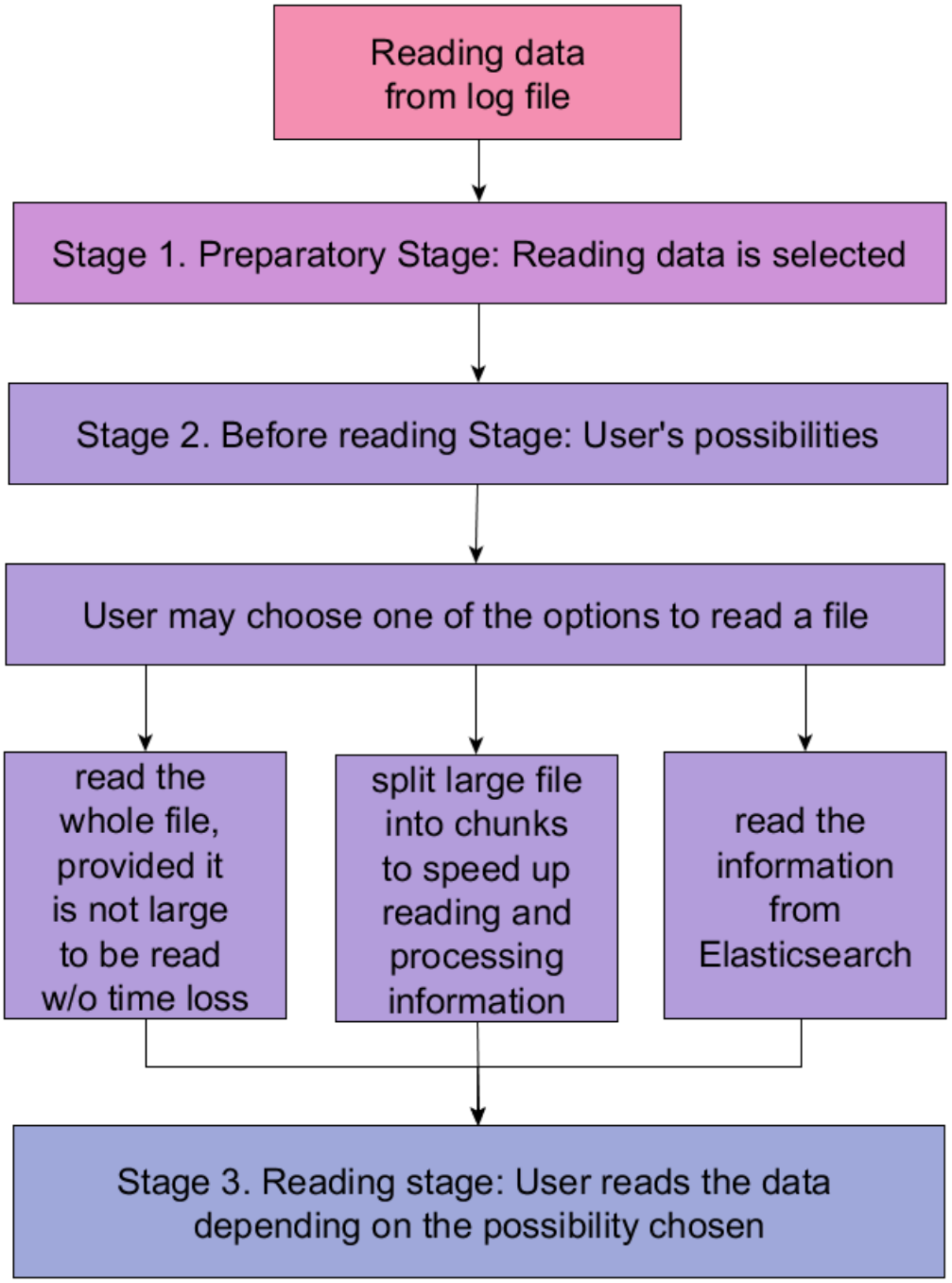


Figure 3.2.1: Reading data from log file

3.2 Data analysis

To see the available fields, a user should tell the program which type of log data is kept in a file. The program should offer two options – common and combined log formats. Depending on the type, the available fields should be represented.

For execution of cluster analysis, a user should be able to choose values. The minimum needed settings are:

- number of clusters,
- number of iterations,
- minimum two fields from log data as attributes.

If a user is unsure about the number of clusters, providing a chance to set a range of clusters and comparing results for different cases could be also helpful. Additionally, to the standard fields the following values can be provided as possible attributes for analysis:

- number of same requests per ten seconds,
- number of requests sent from the same IP address,
- quantity of special symbols,
- quantity of SQL words.

Every attribute can also have a weight – constant multiplicator whose value depends on the influence of the attribute on the result. The attributes are either numerical values, or categorical. For instance, bytes transferred is numerical value, while IP address or response are categorical. Depending on the data type, the program should execute one of the clustering algorithms:

- k-means for numerical data,
- k-modes for mixed data,
- k-prototypes for categorical data.

The user should have an option to exclude requests for graphics from analysis, as an uninformative entrance (see Figure 3.3).

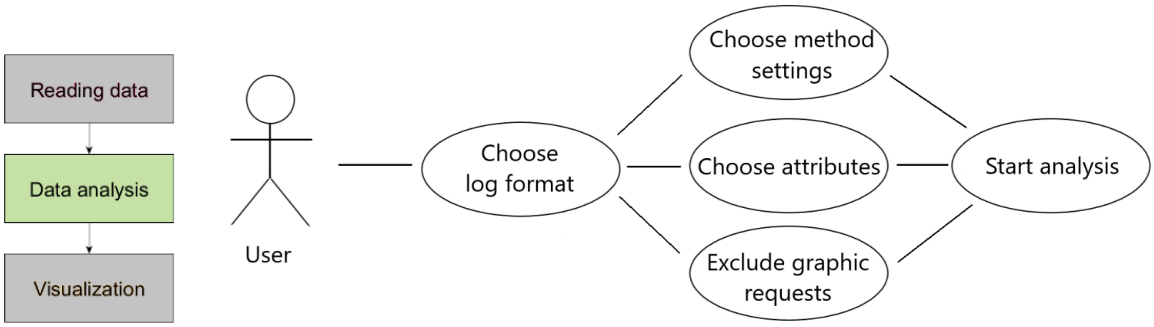


Figure 3.3: Use case diagram – data analysis

All of the foregoing can be accumulated in the diagram (see Figure 3.3.1):

The diagram shows the process of analyzing the data from a log file in details. As it can be seen, this process goes through three stages. Each stage describes the actions of the "participants" of the process: the program and the user. At stage 1, depending on the type of the data, kept in the file, the program offers two options and at stage 2, the user chooses values to conduct the analysis. After these steps have been taken, the program executes the analysis. It is necessary to admit here that the elbow method is used to determine the optimal number of clusters [Gov17].

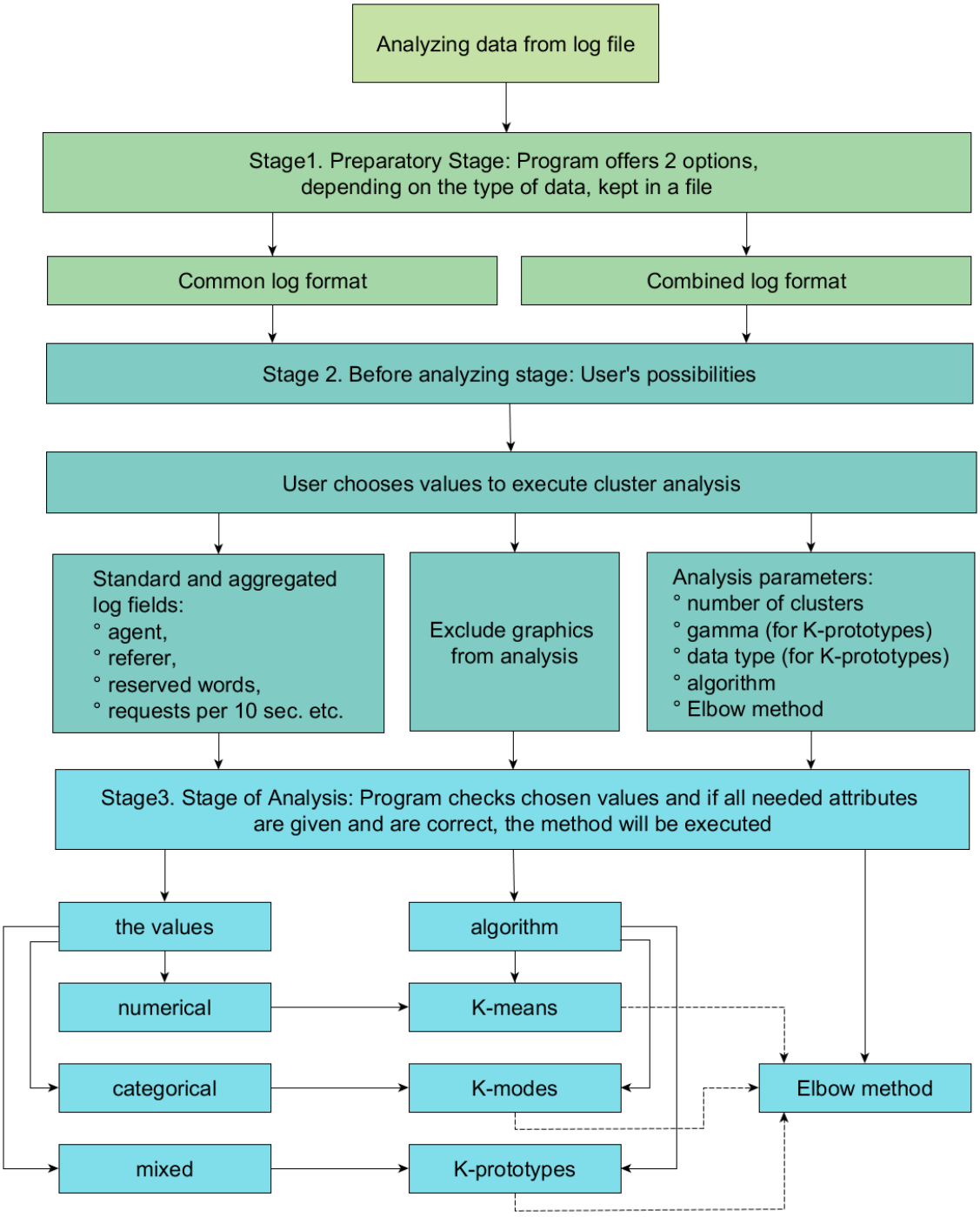


Figure 3.3.1: Analysing data from log file

3.3 Data visualization

The log entries, which are being read from a file, should be represented in a readable form. This can be done by showing the data as a table. If the data is too large, then only some part of it should be shown. If a user sees the data, written in a table, they probably will not scroll hundreds of thousands of lines, trying to get an image of the situation. Building a table is time consuming, so setting a limit to the amount of data, which should be visualized, is reasonable. A user should be notified that the table they see is not the whole file, but the first 10 000 rows.

The data analyzed should also communicate a clear result. First, if a user is unsure about the number of clusters, they can select an algorithm and the elbow method. The elbow method will carry out the computation for 1...n clusters. The results can be represented as a graphic with two axes (see Figure 3.4):

– number of clusters – as *X-axis*;

– sum of distances to centroids (also called as cost function) as a *Y-axis*.

From some number of clusters, the sum of distances does not change radically. But the point where it does (as in the example provided, see point 4), determines the optimal number of clusters, which should be chosen for carrying out analysis. This point is often called an *"elbow"* [Kod13]. This graphic is the first visualization option – for elbow method the user does not require any further information representation, because this option deals only with determining a good number of clusters for further analysis.

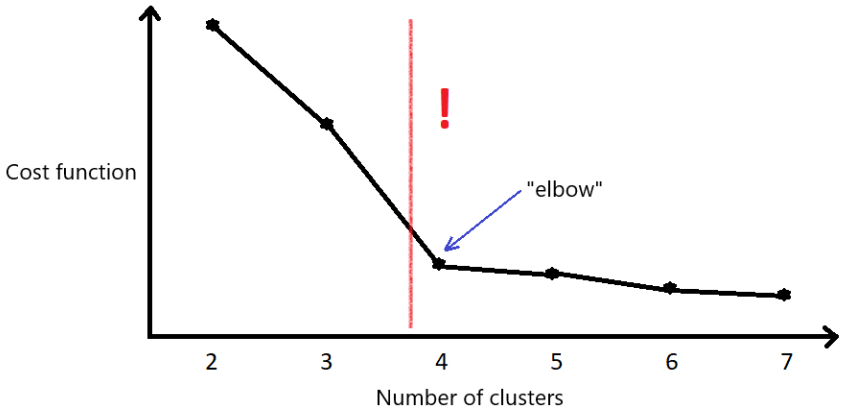


Figure 3.4: Determination of optimal number of clusters

When a user performs the data analysis for some particular number of clusters with some specific algorithm, they should be able to see which log entries are classified to which group. Therefore, the table, containing entries, should get an additional column with the name of cluster, to which a log was categorized. Enabling a user to sort or to filter data would provide a better user experience.

The data is usually large, therefore it is not sufficient to make only a table update. Some textual description of clustering results should be provided. This information can contain centroids values, total number of elements in every cluster, the number of known bots in every cluster. Usually the analyses will use more than two attributes, but building a chart is only possible for two or three dimensions. Therefore, if a user chooses two or three attributes, the 2D or 3D graphics respectively, must be shown together with the textual description and table update.

If a user choses more than three attributes, the program should provide a possibility to build 2D and 3D graphics for different attributes' combinations. The user should be able to choose these combinations from the attributes they take for analysis. These graphics could show the dependencies between attributes and estimate if those attributes are important, or if they do not influence the result (see Figure 3.5).

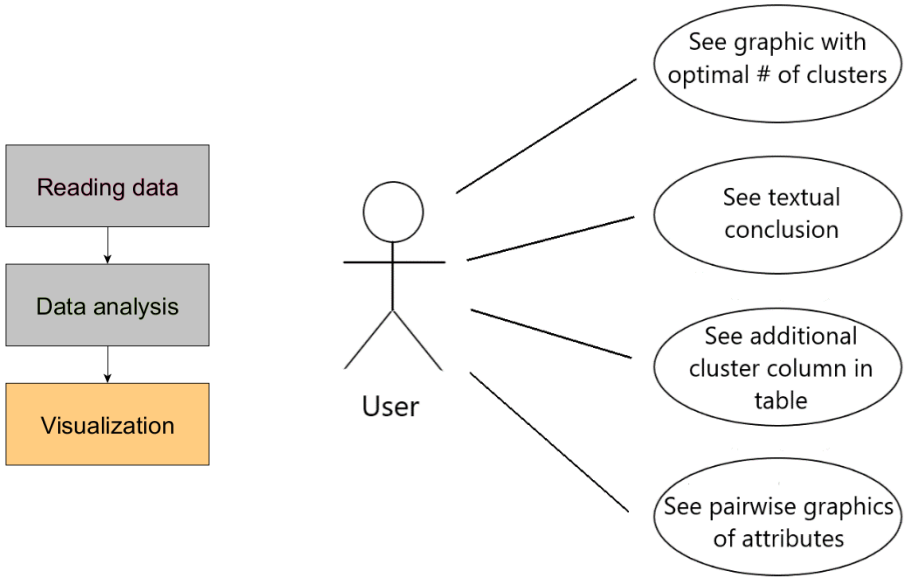


Figure 3.5: Use case diagram – data visualization

All of the foregoing can be accumulated in the diagram (see Figure 3.5.1).

The diagram shows the process of visualizing the data from a log file in details. As it can be seen, this process goes through three stages. Each stage describes the actions of the "participants" of the process: the program and the user. At stage 1, the method is chosen. At stage 2, the user has two options. The first option is for those who know the number of clusters, the second is for those who do not. If the user does not know the number of clusters that should be visualized, they choose an elbow method. At the stage of visualization the program shows the results. It is necessary to admit here that the user who would like the program to show the 2D/3D graphic, they have to choose 2-3 parameters.

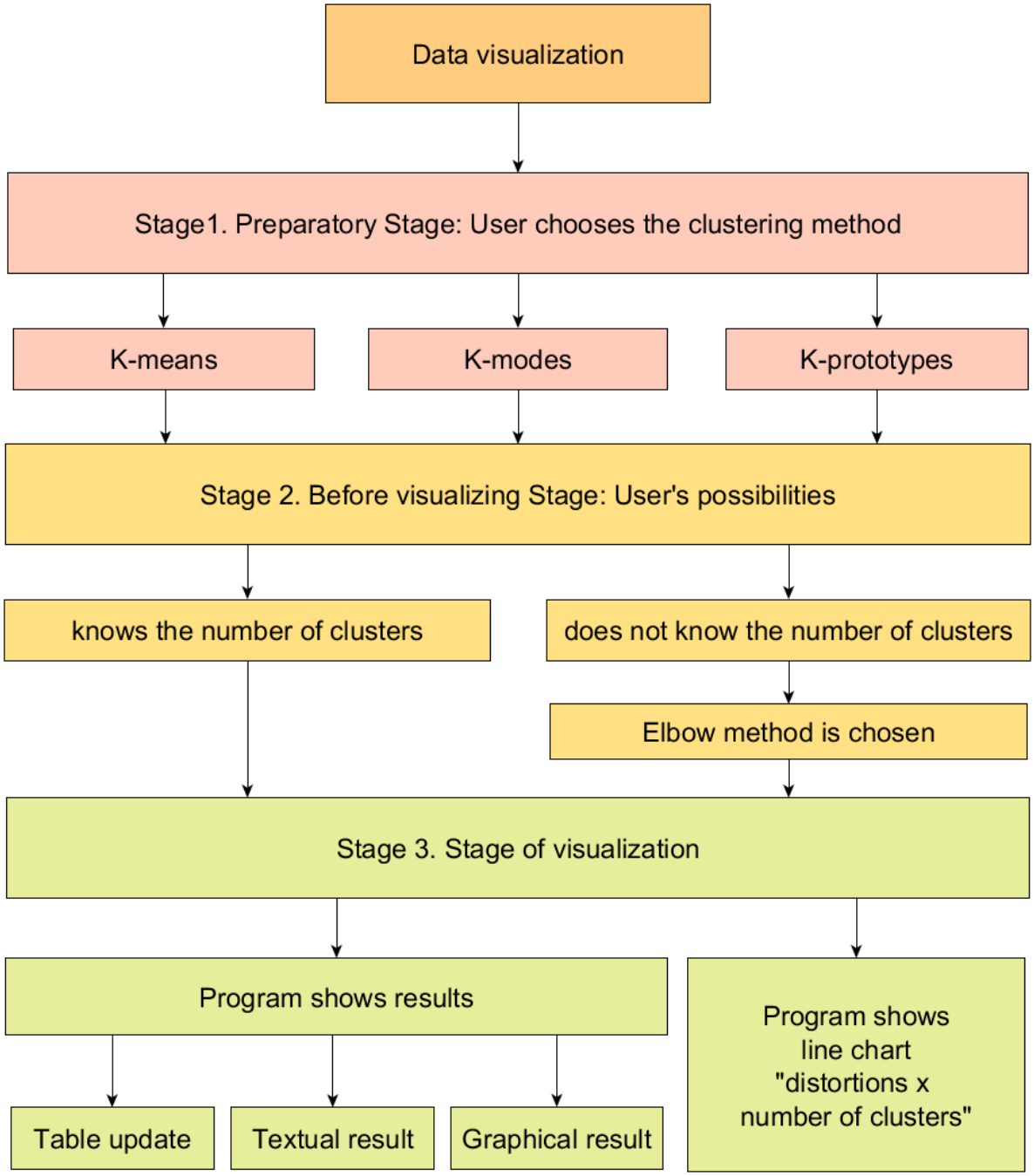


Figure 3.5.1: Visualizing data from log file

3.4 User Interface draft

When the program starts, the main screen should appear with the following three general components:

- fields from file,
- table with data,
- analysis options.

These components should split the whole canvas into resizable parts. On the left, the first component should contain a list of fields from file. It might have a descriptive purpose, or be used to filter the values in the table. The second component - a table - is meant to contain data from the log file selected. For usability, a user should be able to sort data by columns. Filtering could be also very helpful, especially when only some limited number of logs from the very large log file is shown. Because the log data usually contains multiple columns, it is necessary to make a table resizable – the user should be able not only to enlarge the program window to the full screen, but also be able to minimize the area of the fields' list.

The lower part of the window should provide a user the possibility to select values for analysis and adjust its settings, like number of clusters, method, parameters, fields' weights, etc. The analysis options component is good to have on the main screen, because it is supposed to be used multiple times. This way the analysis setup can be done in the minimal number of clicks. After analysis is done, the table should get updated – an extra column *"cluster"* should contain the number of cluster, to which a row was classified.

The draft of the main window is depicted in Figure 3.6.

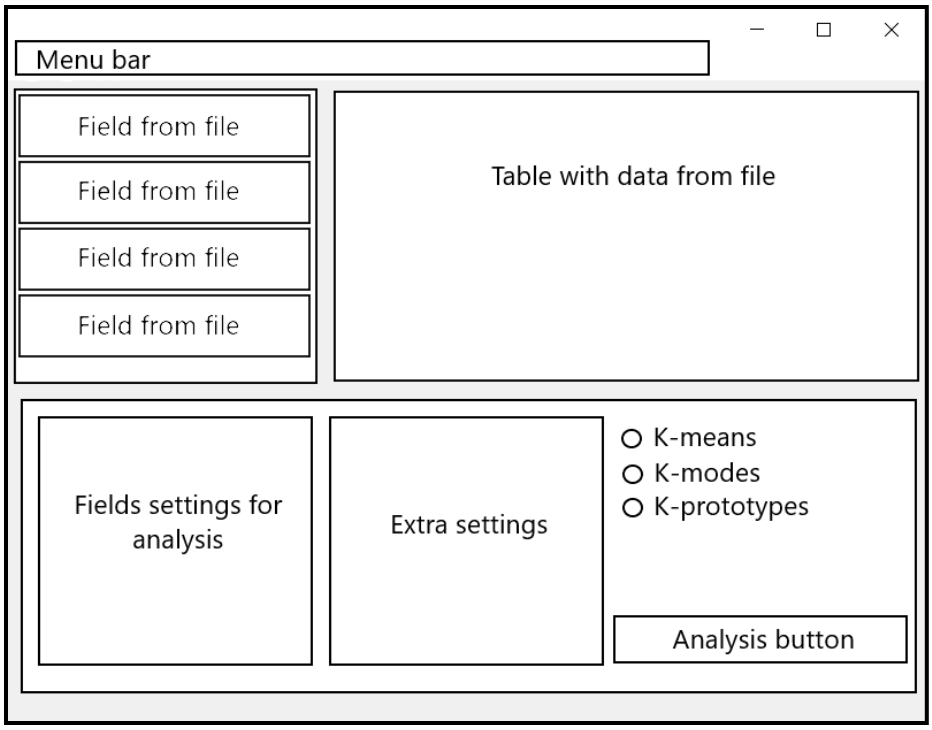


Figure 3.6: Draft of the main window

From the menu bar a user should be able to choose an option *"Open file"*. When chosen, a new window should appear, blocking the main window, so that a user would not be able to open it multiple times, or to do any other operations until the *"Open file"* window is closed.

This window should provide the first two options – reading whole file or splitting file, to read it partly. If a user chooses the second option, they should be able to provide the limitation size. It means, they should write how large the newly generated parts of a file should be. For instance, the file should be split in multiple files, and every file should contain maximum 10,000 lines. Therefore, a user should write 10,000 in the field *"number of lines"*.

The second important setting is the grok. The term *"grok"* means *"to understand"*. Because of this, the main goal of the grok as a tool is to help understand an extensive software. Grok parses and matches patterns in a log, message, etc.; extracts the information needed from the structured or unstructured data; transforms unstructured log

and event data into structured data [Vand17; Pygr17]. To be exact, it “combines the text patterns into something that matches your logs” [grok17] For instance, there are already predefined patterns of log files, which can be called, or a user can generate their own one. In any case grok helps make the data structured and queryable [grok17]. *“List of log formats”* should be a drop-down list, containing names of the log formats provided. After a user chooses the way of reading a file and the grok, they should click the button *“Choose file”*, which should provide a standard file explorer window. When all settings are chosen, a user can click an *“OK”* button. The data should be then loaded to the main window.

The *“Cancel”* button should close the current window. The draft of the *“Open file”* window is shown in Figure 3.7.

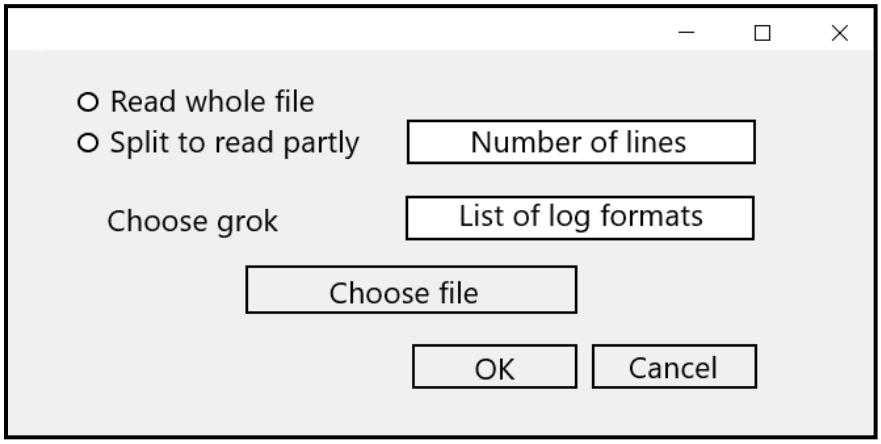


Figure 3.7: Draft of the “Open file” window

If a user has installed Elastic Stack and wants to parse the data from it, then another window should be shown (see Figure 3.8). In this case the data is stored in the NoSQL database in JSON format. Depending on the installations of Elastic Stack, the data can be indexed in different ways. In the figure below this data is indexes as *“File 1”*, *“File 2”* and so on. Thus, a user should be able to choose one or multiple data.

The *“OK” button* closes the current window and processes the chosen data. The *“Cancel” button* closes the current window without performing any further actions.

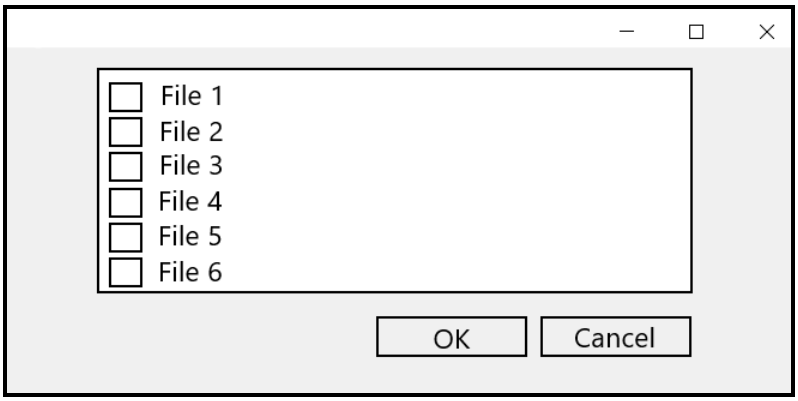


Figure 3.8: Draft of the “Elasticsearch data” window

When the data is loaded, in the main window the fields are shown and the table is filled, a user can do some analysis. Depending on the settings, the result comes out in the new window (see Figure 3.9). If a two-dimensional graphic can be shown, then it should appear on the screen, together with the textual information about the clustering results.

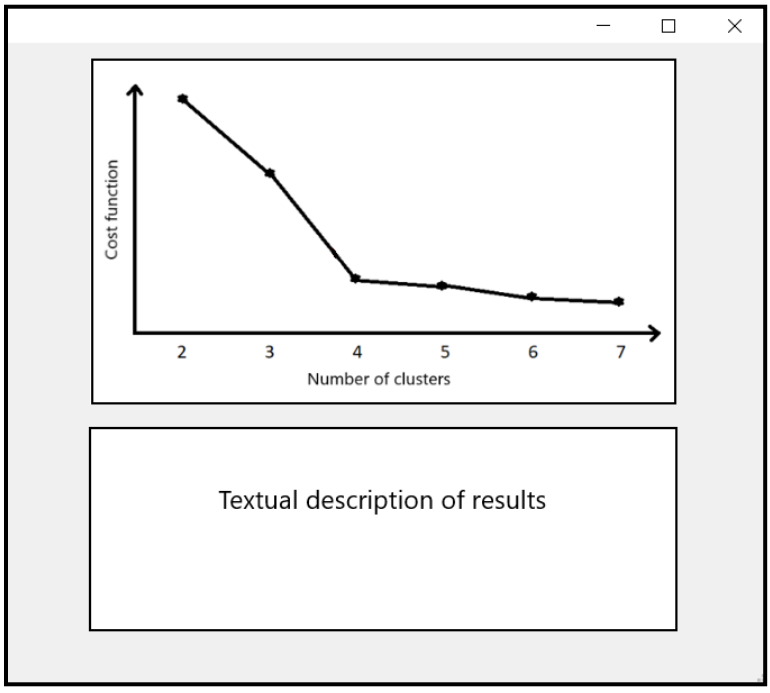


Figure 3.9: Draft of the “Result” window

The same window structure can be applied for demonstration of the pairwise graphics of analyzed attributes.

In the light of the foregoing, the concept of the program that has been represented in pictures 3.1-3.9 should or should not confirm the hypothesis of the thesis: using *k-means* and its modifications: *k-modes*, and *k-prototype* for conducting a cluster analysis will help process data of different size and nature and detect possible anomalies in log files.

3.5 Conclusions

Referring to the findings of chapter 2, paragraph 2.5, it is necessary to point out that existing popular tools for log data monitoring do not provide any mathematical cluster analysis methods. That is why in chapter 3, an application, which would determine anomalous activity by means of cluster analysis has been developed. The application can be used at a single computer, as the data for analysis originates from the same web server. In addition, the application can be compatible with Elastic Stack, which tends to deliver data from different computers. The concept of the program has been represented in Figure 3.1-3.9.

CHAPTER 4
Implementation of log files analysis program

4.1 General information

The project created is a prototype, which provides all needed for the analysis functionality, but still can be further developed. As long as the main focus of this research has been on the analysis of log data, conducted by means of clustering methods, the choice of the programming language has been in favor of **Python 3**. "*Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language*" [TutPy17]. The third version – Python 3 is the current one, and is recommended to be used instead of the older one (Python 2). The versions have some differences in syntax, that is why they are incompatible.

Python needs an environment to run, so to use the programs, written in this language, it is necessary to have it installed. It can be downloaded for free from the official website [Py17]. To do this research, the version Python 3.6.2 has been chosen and installed.

Because a lot of different mathematical libraries were used, it is useful to have the **Anaconda** – *"the world's most popular Python data science platform"* [Anac17] – installed, as it provides the most widely implemented mathematical tools. This platform includes such important packages, as *NumPy*, *matplotlib*, and *scikit-learn*, which have been used in this program. Installation of some further packages can be done with the help of anaconda (conda install) or **pip** installer (pip install) over the command line.

If more than one Python project runs on the same computer, it is recommended to create a virtual Python environment, which would isolate packages, installed for some specific project, without polluting the global namespace [Hattem16]. For instance, if two projects need two different versions of the same package, they will be able to do that, without breaking each other's code.

The needed packages for running this project are as follows:

- PyQt5 – is used for building GUI (graphical user interface);
- Matplotlib – is used for building 2D and 3D graphics;
- SciPy – some mathematical routines, such as counting special distances in clustering methods have been implemented from this package;
- scikit-learn – provides an efficient implementation of K-means analysis;
- NumPy – provides better data structures for efficient processing of large amount of data;
- elasticsearch – a client, for easy connection with Elasticsearch;
- pygrok – library, used to extract information from structured data, in this case – from log files [Gary17];
- kmodes – package, which includes k-modes and k-prototypes algorithms [Nico17];
- Jinja2 [Ron08] – templating language, used for styling the textual output.

Some of these packages, such as *PyQt5, elasticsearch, pyqrok, kmodes, Jinja2* are not provided in Anaconda. Therefore, they must be installed additionally.

For reading data from Elastic Stack (see paragraph 2.4), Logstash and Elasticsearch engines should be installed and configured on the computer. The needed configurations will be described below (see paragraph 4.3.2).

The project has been developed under the Windows 10 system, but it should be executable on any other system. The functions, where OS may play a significant role will be mentioned.

For creating this project, the PyCharm Community Edition IDE (Integrated development environment) has been used [PyCha17]. The project can be started from PyCharm, where

it can also be further developed, or the user can start it from directory or the command line.

The project has the following structure:

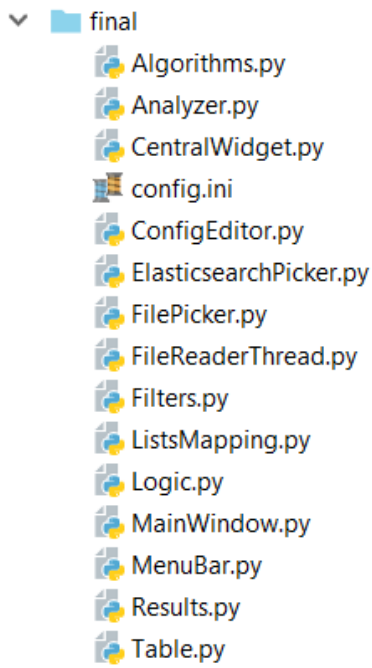


Figure 4.1: Project structure

There are also the external libraries, which have not been shown in Figure 4.1. As long as a large part of the code is the creation of a user's interface, to ease programming, the whole structure has been divided into the large UI parts and some functional parts. For instance, *CentralWidget*, *FilePicker* or *MenuBar* contain parts of GUI respectively. *Algorithms* file contains some implemented data preprocessing, and realization of clustering methods. There is also one very important file – *config.ini*, which includes some of the main settings. This file can be managed, so the user can make some basic changes to the program for their own needs, without having to edit the code itself.

4.2 Starting program

The file, which is an entry point to the program, is *MainWindow.py*. A user should double click it in the file manager, or start it from the command line. To run a program from the command line, a user should enter the file's path and write *py*.

For example:

```
E:\LogsAnalyzer\final\MainWindow.py    py
```

When the program is started, the standard empty window appears, containing *MenuBar* (see Figure 4.2).

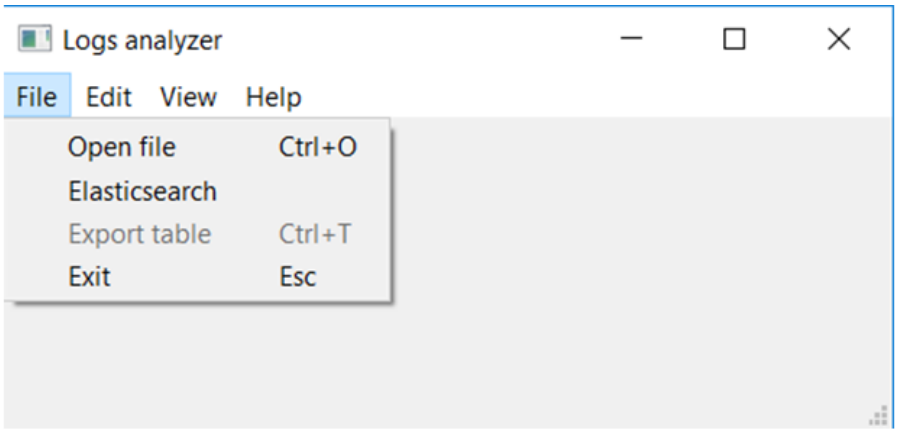


Figure 4.2: Program window on start

The menu bar provides different options. The menu *File* offers to read data from file, from Elasticesarch or to exit the program. The *Export table* function will be active after data analysis is made, to save the results.

All parts of interface are built with *PyQt5*. Every active interface element is connected to specified function, which starts when an event occurs, e.g. each time when the *Open file* is clicked, the *open_file_picker* will be called. This creates a new window, which asks a user for the needed information to open a file.

4.3 Reading data

Because the log data is usually very large, in the concept different approaches to read it have been suggested. As long as this program is a beta version, it provides the fully functional option to read one file at a time. It also provides the possibility to split a file into multiple chunks, but it still reads one of them.

The idea of splitting a file into multiple chunks and to read them using threading, turned out to be not as effective, as it had been expected. Even worse, it makes program run slower. Unfortunately, in Python the threads are executed not in parallel, but in sequence, because of the **GIL** (global interpreter lock). GIL protects access to Python objects, and is implemented due to the memory management, which is not thread-safe in Python [GIL17].

The Elasticsearch connection also has been created, but the processing of the data received has not been implemented yet – the data structure should be adjusted, to be compatible with the rest of the program. Still, this approach allows reading multiple files very effectively.

4.3.1 Open file

To read the data from a log file, a user should choose the menu *File -> Open file* or use the shortcut *Ctrl+O*. The new window will appear (see Figure 4.3).

A user should choose between reading the whole file and splitting it into parts.

To read a file, a user should choose grok – the format of the log file. There are two options provided – Combined Apache log and Common Apache log. At the same time, a user can provide their own formats, by adding them into the configuration file, as it will be described later (see paragraph 4.7). By clicking the button *Open file*, the *QFileDialog* will be prompted. This dialog opens the file explorer, so that a user can navigate and choose the file that should be read.

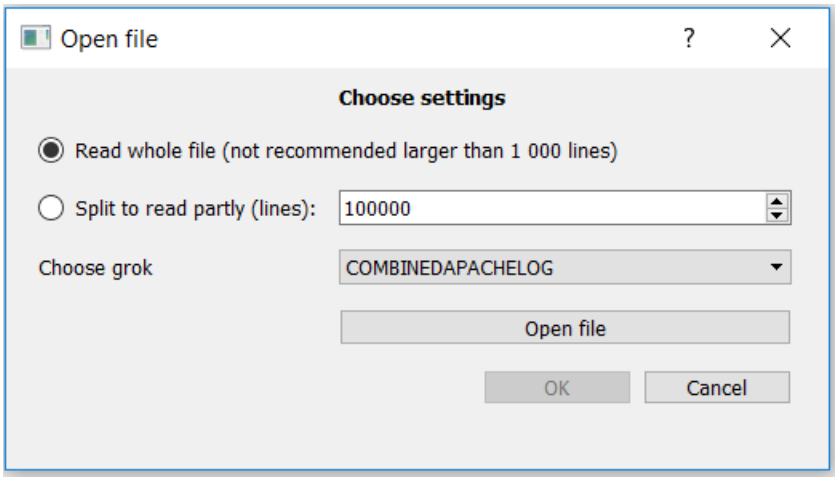


Figure 4.3: Open file window

When the file is chosen, the button *OK* activates, and by clicking it, the file will be read in the new Thread. The threading is provided by *PyQt*, and is still useful when it comes to long-lasting processes in the background. If reading data is performed in the same thread as the program, then the UI freezes and does not respond to any actions. The threading is used with the help of *slots* and *signals*. While the main window sleeps once in a couple of seconds (because it should free some time section for the second process), reading data is performed. When reading file is done, the main window receives a signal and updates the interface.

If a user decides to split the file, they can decide how many lines should contain each file. This is done by changing the value in the *QSpinBox*. Then the user chooses the file through the File Dialog and submits their entries. When it is done, the warning is prompted (see Figure 4.4).

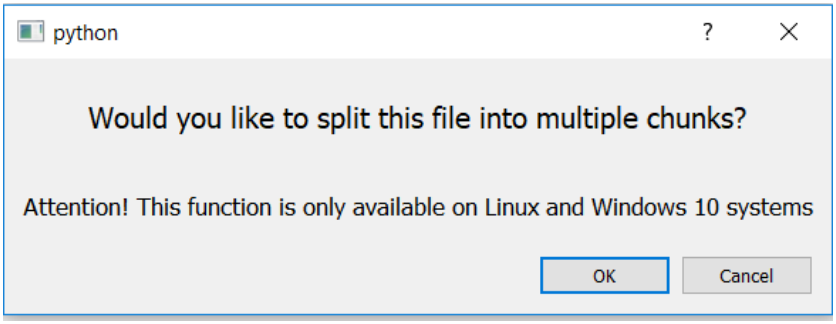


Figure 4.4: Warning before splitting files

This window reminds that the function given is only available under the Linux and Windows 10 systems. The splitting of the file is written in Bash, which is only available on Unix systems, and in developer's mode on Windows 10. Doing it with Python is considered ineffective, because it has to read the whole file line by line, which is time consuming. Therefore, bash command has been called in the sub process (see Listing 4.1).

Listing 4.1 – Creating bash command and sub process

```
if dialog.exec():
    bash_command = 'bash -c; split -l ' +
                str(self.LINES_TO_READ) + ' "/mnt/e/' +
                self.PATH[3:] + '" ' + '"/mnt/e/' +
                self.PATH[3:] + '"'
    subprocess_cmd(bash_command)


def subprocess_cmd(command):
    process = subprocess.Popen(command,
stdout=subprocess.PIPE,
                               shell=True)
    proc_stdout = process.communicate()[0].strip()
```

The bash command accepts the number of lines, path of file, which should be split, and the path where the files generated should be stored. The process does not affect the source file, it creates new files.

This option is useful, because after splitting of the file, the user can open any of the chunks created to process them in program.

4.3.2 Elasticsearch

In order to use Elasticsearch functionality, a user should preinstall it. To read files in program, Logstash and Elasticsearch are needed.

Logstash reads data from the resources given and stores it into Elasticsearch. To define inputs and outputs, the configuration file should be created. For reading data in this program, the following configurations have been used:

Listing 4.2 – Logstash configuration file

```
input {
  file {
    path => ["E:/www_access_log"]
    start_position => beginning
  }
  file {
    path => ["E:/www_ssl-access_log.out"]
    start_position => beginning
  }
}
filter {
  if [path] =~ "ssl" {
    mutate { replace => { type => "apache_access_ssl" } }
    grok {
      match => { "message" => "%{COMMONAPACHELOG}" }
    }
    date {
      match => [ "timestamp" , "dd/MMM/yyyy:HH:mm:ss Z" ]
    }
  } else {
    mutate { replace => { type => "access_logs" } }
    grok {
      match => { "message" => "%{COMBINEDAPACHELOG}" }
    }
    date {
      match => [ "timestamp" , "dd/MMM/yyyy:HH:mm:ss Z" ]
    }
  }
}
```

```
output {
  elasticsearch {
   hosts => ["localhost:9200"]
   index => "apache-%{+YYYY.MM.dd}"
   document_type => "system_logs"
 }
}
```

The configuration file (see Listing 4.2) consists of three main parts: *input, filter* and *output*. In this example the data has been read from two different files, from their very beginnings. This data from files has been filtered by filenames. If the path contains “ssl”, then each entry is processed as a file in Common Apache log format, otherwise as Combined Apache log. The timestamp is mutated into the format given. All data is sent to the Elasticsearch (which runs on the *localhost:9200*), and indexed by date of entries.

The configuration file should correspond with the location of files. A user should change the input part for the correct file location. If there are multiple nodes of computers, this also can be configured. The user has a possibility to get the real-life data not only from some files, but also from different computers and different ports.

When everything is configured, the user should start Elasticsearch. This can be done through the command line. In the directory where elasticsearch is installed, the following command should be called:

```
bin\elasticsearch.bat
```

After Elasticsearch is started, the Logstash can be called:

```
bin\logstash -f [name_of_configfile].config
```

While using the program created, to read data from Elasticsearch, only Elasticsearch should run in background. If it already contains some data, which has not been changed since the last time when Logstash was started, there is no need to start Logstash.

When everything is installed, the user can choose the *File -> Elasticsearch* menu.

The program receives this data by sending request to the *localhost:9200*, where Elasticsearch runs, as it was installed in the configuration file. This data is split into header and list of entries. The dialog is called then (see Listing 4.3).

Listing 4.3 – Open Elasticsearch window

```
def open_elastic():
    res = requests.get
          ("http://localhost:9200/_cat/indices?v")
          .text.splitlines()
    res = [r.split() for r in res]
    header = res[0]
    res.pop(0)
    Dialog(len(res), len(res[0]), header, res)
```

As the result, all documents with current statuses, previously stored in Elasticsearch will be shown in a table, and be available for user to choose (see Figure 4.5).

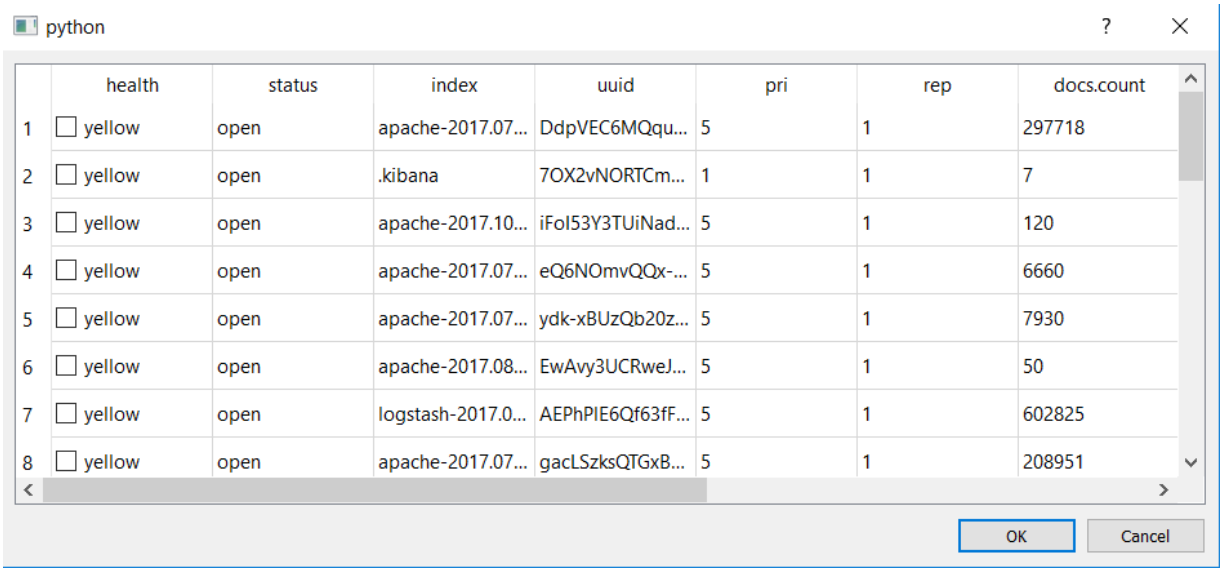


Figure 4.5: Elasticsearch window

When a user chooses documents, the Elasticsearch client is created. For every document chosen it gets the data by searching it by index. This data is then stored in dictionary.

Listing 4.4 – Reading documents from Elasticsearch

```
client = Elasticsearch()
for item in self._list:
    s = Search(using=client,
index=res[item][2]).extra(from_=0, size=1000)
    response = s.execute()
    data = response.to_dict()
```

This data should be cleaned, in order to get parsed to the table, and being applicable for clustering methods.

4.4 Main window

When the file is successfully read, the data is shown in the main window. The three new widgets appear, created with *QSplitter* (see Figure 4.6). This makes them movable and resizable. It is useful, because there is a lot of data, which should be shown, especially when it comes to the table.

The first widget, (1) on the top of the left, has been designed for filtering. This part has not been implemented yet. It has been designed to filter the data shown in the table – to hide some columns or choose only some rows according to the parameter given. This option does not influence the analysis, but could be helpful to make the reading of the data in the second widget, (2) i.e. in the table, more user-friendly.

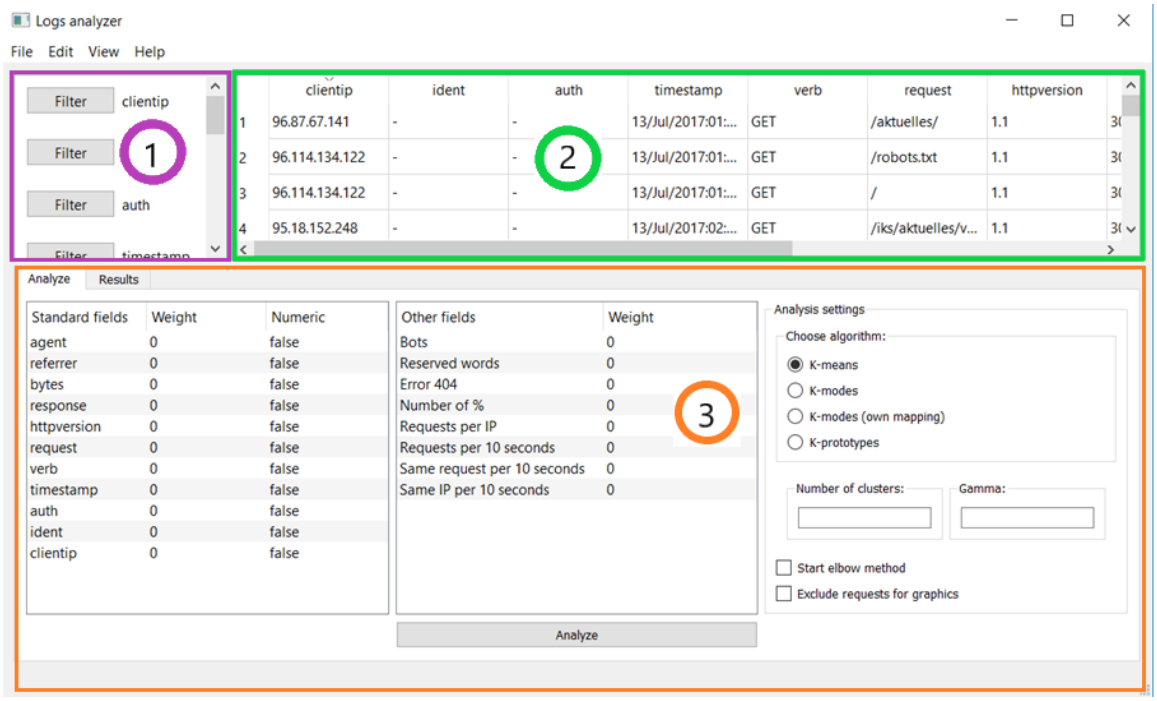


Figure 4.6: Main Window with data

The last widget (3) contains all fields and options needed to conduct the analysis. The user can choose standard or extra fields, which should be analyzed, set the weights, number of clusters and some other settings.

4.5 Analysis

4.5.1 Settings

The standard fields are the fields, read from the log file, depending on the grok chosen. The extra fields are generated from the standard ones. For instance, *Bots*, *Reserved words*, *Error 404* can have value 0 or 1, depending on the occurrences in logs. Bots and reserved words are read from the configuration file, which can be modified by a user. *Number of %*, counts the number of percent symbol in each request. *Requests per IP*, *Requests per 10 seconds*, *Same request per 10 seconds* and *Same IP per 10 seconds* are agglomerative values, which determine the frequencies of requests or users.

The weights for every field can be implemented in future for the modification of the methods. In this project a user chooses fields, by changing their weights to non-zero values. The extra fields are all numeric, but standard ones can be of different types. Therefore, the user can set up whether the value is numeric or categorical. This is needed only for k-prototypes method, which requires the specified lists of numeric and categorical fields.

The user can select one of the three methods. There are two K-modes methods implementations, which are the same algorithms, but with different value mappings (e.g. *Mozilla* has value 1, *Google Chrome* - 2, *Opera* - 3, where the numbers reflect their mapped values). Both of them have been provided in order to show how much the pictures with the same clustering results can differ because of the categorical value mapping. This explains that the visualization is less informative: the user can only grasp how large is the difference in sizes of clusters, but they can not get any uniform picture of how elements are located, because the categorical data can be located in any sequence.

The user is obliged to give the number of clusters, and they can also give the gamma value for K-prototypes method, if needed. If no gamma is provided, the value 1 will be assigned.

If the user checks the *Start elbow method*, the number of clusters will be interpreted as a range of clusters. For this range distortions will be counted for each case. If this field is not selected, the clustering method will run for the number of clusters given, and the results will be shown in the *Results* tab.

To shorten the computation, it would be helpful to provide the user with possibility to exclude requests for graphics.

4.5.2 Algorithms

The numerical data should be normalized. If the data is a numpy array, this can be done as follows:

Listing 4.5 – Data normalization with Numpy methods

```
diff = data.max(axis=0) - data.min(axis=0)
if all(diff != 0):
    data = (data - data.min(axis=0)) / diff
```

The *data.max(axis=0)* looks for the maximum in a row of a matrix. Here in *data* every row contains the values of the field for each entry. This indicates that if there are two fields and 500 entries, the matrix size will be (2,500). If all differences are not equal to zero, all values can be normalized.

The k-means algorithm has been taken from the SciPy module (see Listing 4.6).

Listing 4.6 – K-means call

```
kmeanModel = KMeans(n_clusters=number_of_clusters).fit(data)
clusters = kmeanModel.fit_predict(data)
centroids = kmeanModel.cluster_centers_
```

This way the clusters and centroids have been determined.

The other two methods are called from kmodes external package. K-modes algorithm requires the number of clusters, the other values if not provided, will be set automatically. For instance, the *n_init* parameter (see Listing 4.7) will be set automatically to 8.

Listing 4.7 – K-modes call

```
km = kmodes.KModes(n_clusters=number_of_clusters,
cat_dissim=matching_dissim, init='Cao', n_init=5, verbose=1)
clusters = km.fit_predict(data)
centroids_unmapped = km.cluster_centroids_
centroids = km._enc_cluster_centroids
```

Here, the categorical dissimilarity is counted as a matching dissimilarity, the initialization corresponds to the Cao method. The method will be iterated five times, and the best result will be chosen by the smallest cost.

The unmapped values of centroids are needed to print the result. For building graphics, the mapped centroids are needed. This means that the categorical values, which are

usually some texts, are mapped to numbers. The algorithm itself works faster when numbers instead of strings are compared, and for building graphics the string fields should correspond some numeric values.

The k-prototypes has been called as in Listing 4.8:

Listing 4.8 – K-prototypes call

```
data = normalize_numerical(data, numeric_fields)
if len(gamma) == 0:
    km =
kprototypes.KPrototypes(n_clusters=number_of_clusters,
init='Huang', gamma=1, n_init=5, verbose=1)
else:
    km =
kprototypes.KPrototypes(n_clusters=number_of_clusters,
init='Huang', gamma=int(gamma), n_init=5, verbose=1)
clusters = km.fit_predict(data,
categorical=categorical_fields)
centroids_unmapped = km.cluster_centroids_
centroids_unmapped =
combine_vectors(km.cluster_centroids_[0],
km.cluster_centroids_[1], numeric_fields,
categorical_fields)
centroids = combine_vectors(km._enc_cluster_centroids[0],
km._enc_cluster_centroids[1], numeric_fields,
categorical_fields)
```

At first, the data should be normalized, but only the numeric fields. If a user did not enter any data, then the method should be called with gamma equal one. In the kmodes package if no gamma is given, the value will be counted with the means of standard deviation from numerical fields (see Listing 4.9).

Listing 4.9 – Gamma determination in kmodes package

```
# Estimate a good value for gamma, which determines the
weighing of categorical values in clusters (see Huang
[1997])
if gamma is None:
    gamma = 0.5 * Xnum.std()
```

The standard deviation should be removed, because in the method of Huang the numeric values were not normalized. The if clause in Listing 4.8 provides skipping the gamma assignation in Listing 4.9.

This method requires an array, which contains indexes of categorical fields, because later all fields are split into two numpy arrays – one with numerical, and another with categorical values only. After the clustering is performed, method returns two arrays with fields of centroids – numerical and categorical. These should be combined to give back one vector with all values.

When Elbow method is chosen, the same algorithms are called, but multiple times for different number of clusters. For each result the distortions are counted, and shown in graphics.

4.5.3 Results

When the fields are chosen (more than 1), number of clusters is given, and the correct method is selected, the analysis can start. To estimate which number of clusters can be reasonable, it is good to start with the Elbow method. When the method is done, a graphic with distortions for given clusters will be shown, as in Figure 4.7.

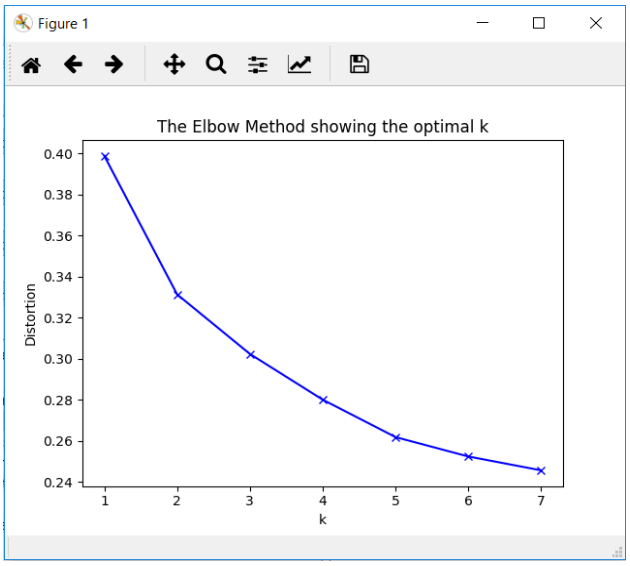


Figure 4.7: Elbow method result

It is important that the distortions would decrease with every step. Otherwise, it means that the method does not find the optimal centroids and works unstable. After getting the results, a user can experiment with different cluster values, which are reasonable enough.

If two or three fields are chosen for analysis, apart of the textual result the 2D or 3D graphic will appear respectively (see Figure 4.8). All graphics can be stored in different data types, e.g. as a picture or PDF file.

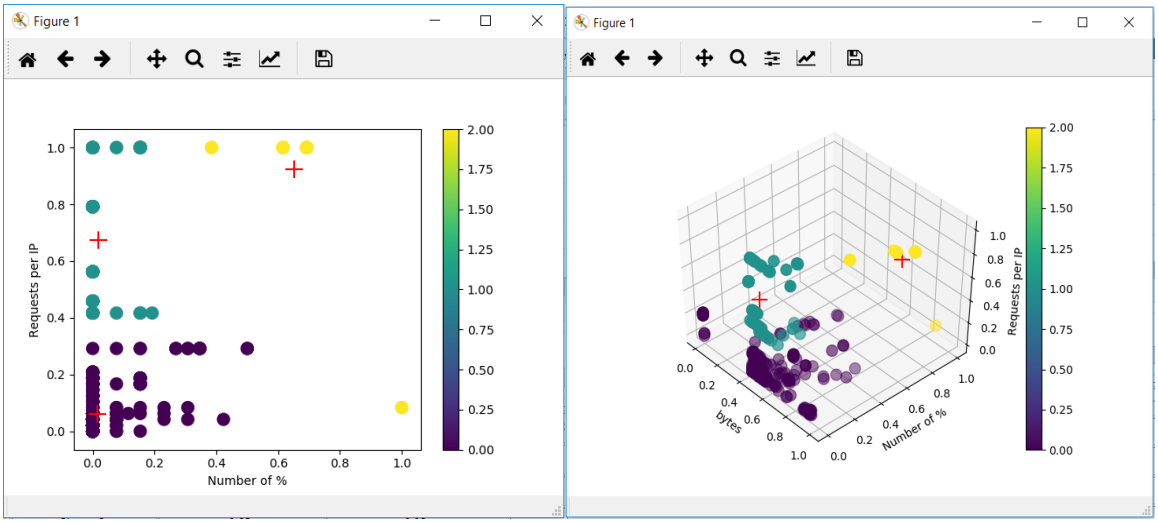
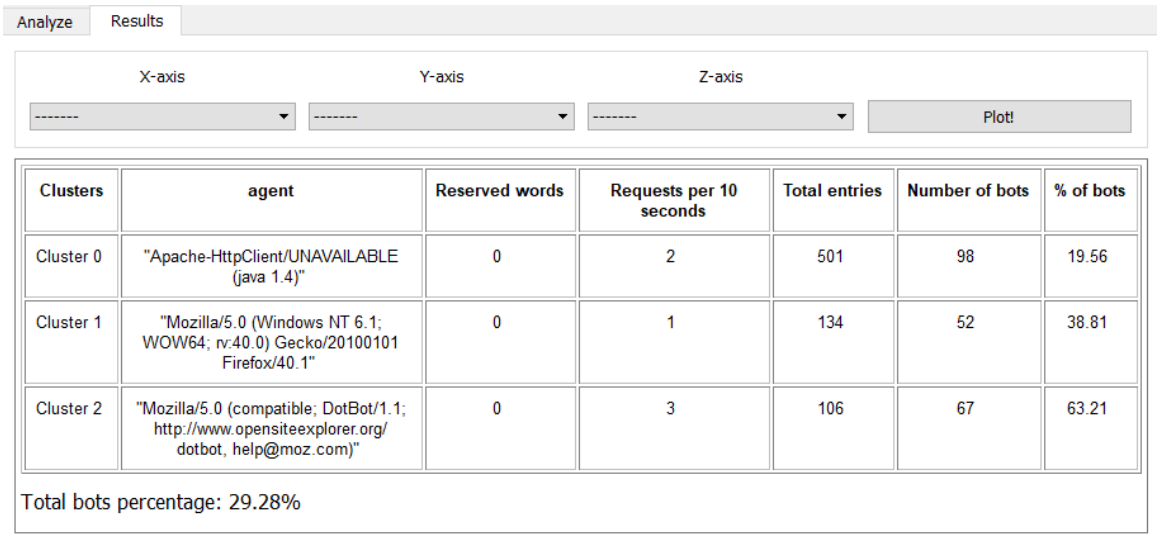


Figure 4.8: Visual example of 2D and 3D graphical results

If more than three fields are taken into analysis, a user can choose 2 or 3 of them to build the graphics. These also can be multiple different graphics, which can help to see on which axes the intersection was better. But the user should not forget that a graphic is not very helpful in defining intersections when it comes to categorical values.

This functionality comes in the *Results* tab, together with textual description (see Figure 4.9). It shows the clusters, the values of their fields, the total number of entries in each cluster, the number of the bots recognized, and the percent of bots in each cluster. Under the table the total bots percentage has been printed.



Clusters	agent	Reserved words	Requests per 10 seconds	Total entries	Number of bots	% of bots
Cluster 0	"Apache-HttpClient/UNAVAILABLE (java 1.4)"	0	2	501	98	19.56
Cluster 1	"Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1"	0	1	134	52	38.81
Cluster 2	"Mozilla/5.0 (compatible; DotBot/1.1; http://www.opensiteexplorer.org/dotbot, help@moz.com)"	0	3	106	67	63.21

Total bots percentage: 29.28%

Figure 4.9: Textual results

The table has been built with the help of the jinja2 template. This allows inserting some styling and HTML text. It allows generating HTML text dynamically (see Listing 4.10).

Listing 4.10 – Counting percentage of bots

```
from jinja2 import Template
table = """
<style>
...
</style>

<table border="1" width="100%">
    <tr>{% for header in headers %}<th>{{header}}</th>{%
endfor %}</tr>
    {% for row in rows %}<tr>
        {% for element in row %}<td>
            {{element}}
        </td>{% endfor %}
    </tr>{% endfor %}
</table>
<p><font size="4">Total bots percentage:
{{total_bots_percentage}}%</font></p>
"""
mainWindow.results_text.setText(Template(table).render(heade
rs=headers, rows=rows,
total_bots_percentage=total_bots_percentage))
```

As it is seen in Listing 4.10, the HTML text contains for loops in table. The variables header, rows and total_bots_percentage are rendered in the table template. The HTML headers and rows elements contain the data, read from the Python list. This text is then set in the *QTextEdit* widget.

Above the text generated (see Figure 4.9), the drop-down fields for X-, Y- and Z-axis are given. By choosing from 2 different values, and clicking the *Plot!* button, the graphics like the one from Figure 4.8 will appear.

The number of bots has been counted as in Listing 4.11.

Listing 4.11 – Counting percentage of bots

```
def count_bot_percentage(data, clusters, centroids):
    bots = np.zeros((centroids.shape[0], 1))
    configurations = get_configparser()
    agent_words = configurations.get("MALICIOUSFIELDS",
"AGENTWORDS").split(',')
    for i in range(len(data)):
        if any(word.upper() in data[i]["agent"].upper() for
word in agent_words):
            bots[clusters[i]] += 1
    return bots
```

The bots are found in the field “agent”, by looking for the key words. These key words are stored in the configuration file, and can be updated when new bots are recognized. The found bots are summarized for each cluster. The clusters with very large number of bots can point to some new bots. For instance, if a cluster contains 70% of bots, then the rest could also turn out to be the bots that have not been noticed before.

The table, containing every log entry, is updated after analysis too. The new column “Cluster” shows which entries belong to which group.

4.6 Table export

When the analysis is done, the updated table with cluster assignments can be stored to the *xml file*. In the menu, the *File -> Export table* becomes enabled. The new file explorer is opened, and a user can select a path and write the filename (see Listing 4.12).

Listing 4.12 – Table export

```
def export_to_file(self):
    filename = ''.join(QFileDialog.getSaveFileName(self,
'Save File', '', ".xls(*.xls)")[0])
    if len(filename) > 0:
        wbk = xlwt.Workbook()
        self.sheet = wbk.add_sheet("sheet",
cell_overwrite_ok=True)
        read_table(self)
        wbk.save(filename)


def read_table(self):
    row = 0
    col = 0
    for i in range(self.table.columnCount()):
        for x in range(self.table.rowCount()):
            try:
                text = str(self.table.item(row, col).text())
                self.sheet.write(row, col, text)
                row += 1
            except AttributeError:
                row += 1
        row = 0
        col += 1
```

The new Workbook is generated with one sheet added. Every row is read into this sheet. When all information is read, the file is saved under the name given.

4.7 Edit configurations

A user can change some settings in the configuration file from the program. Going to the menu, *Edit -> Edit Configurations (Ctrl+E).* The new window appears, containing the current configuration file (see Figure 4.10).

The file *config.ini* is saved to the same directory as the program's scripts. The file is read to the *QPlainTextEdit*. Although the configuration file is not large, the performance of *QPlainTextEdit* benefits over *QTextEdit*, therefore it has been chosen.

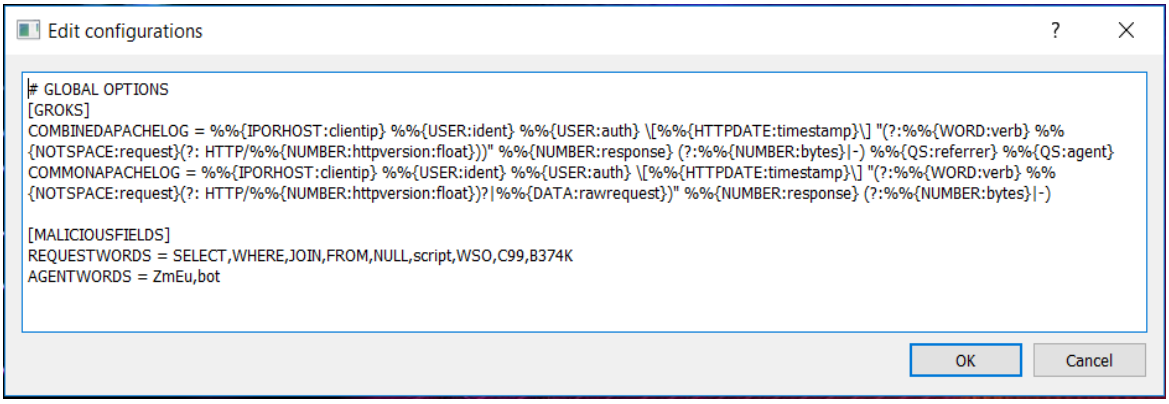


Figure 4.10: Edit configurations window

When a user clicks *"OK"*, the file will be overwritten with the text from *QPlainTextEdit*. If *Cancel* – no changes will be proceeded (see Listing 4.13).

All changes made will be taken to the account by the further working with program. For instance, if the *AGENTWORDS* gets a new field *"bot"*, it will be taken into account by extra field *"Bots"*, which will search not only for *ZmEu*, but also the agents, which contain word *"bot"* too.

Attention! The user should be careful with this file, because if they make a mistake, they can break the configurations, which will influence the reading of files and their analysis. On the other hand, they can add the newly found malicious words or agents, or even add own Grok formats.

The list should be written, separated with commas, without spaces.

Listing 4.13 – Reading and overwriting config.ini

```
if edit_window.exec():
    raw = open(filename, "r+")
    raw.truncate()
    raw.write(text_edit.toPlainText())
```

The difference between AGENTWORDS=”ZmEu” and AGENTWORDS=”ZmEu,bot”:

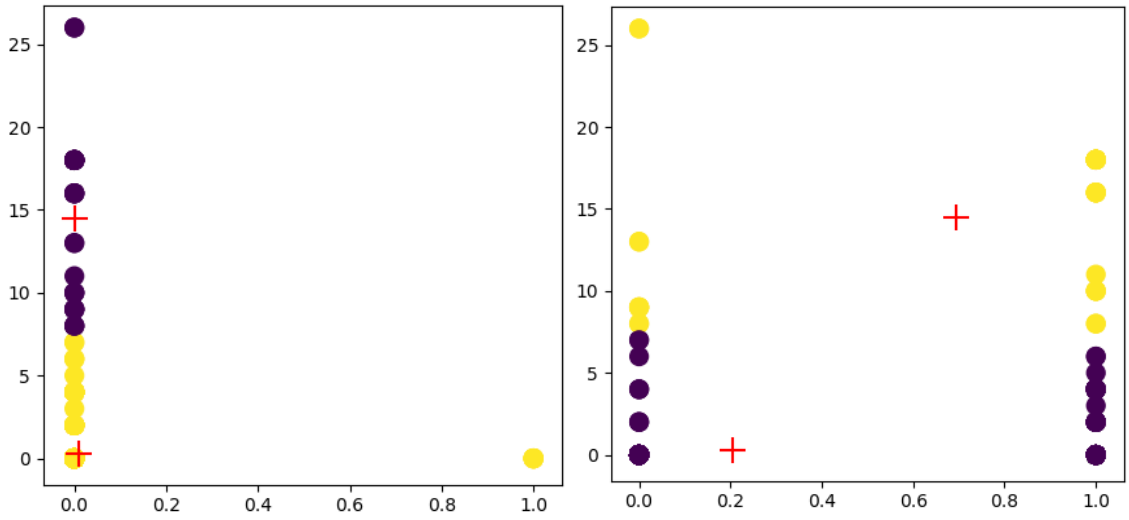


Figure 4.11: Different results by different config.ini

As it can be seen from Figure 4.11, the configurations can affect results dramatically. On the left picture, only one malicious agent is found, called “ZmEu”. On the right, the agents have located differently, because those, containing *“bot”* words have been also considered malicious.

4.8 Conclusions

Chapter 4 has been devoted to the creation of the program, based on the concept suggested in Chapter 3. Throughout the research, the program has been worked out, tested and described. The results can be summarized as follows:

1. The program needs the Python language on the machine to start, together with the environment configured, i.e. the packages needed to run the code.
2. The main functions, such as reading file has implemented. The Logstash and Elasticsearch have been installed and configured, to provide a faster way for reading data. To make this function complete, the data structure should be improved to be compatible with the other code.
3. The main window parses entries of log data in table, reads available fields and provides extra fields for clustering. The user can choose one of provided clustering algorithms, estimate the good number of clusters with the means of elbow method, write the gamma value and select numerical and categorical fields for K-prototyped method.
4. The results have been represented as a text and a 2D or 3D graphics. When more than three fields are chosen, a user can choose two or three fields to visualize data.
5. Bots percentage can help to find new bots, which helps predicting the new entries.
6. After analysis is done, the table with entries is updated by adding cluster column. The information from updated table can be stored in the excel file.
7. When new malicious fields are found, or if a user wants to add a new grok, the configuration file can be changed from the program.

The overall progress of the research, conducted at this stage has suggested that the Elasticsearch could be used of full value, and the clustering methods could be optimized to speed up their work. This could become the objects for the further researches in the areas of programming and statistics.

CHAPTER 5
Results of log files analysis

5.1 Example #1

The file contains 2000 entries from the web server. This file has a type of *Combined Apache Log* format. This indicates that this file contains *"referer"* and *"agent"* fields. The "*referer*" identifies the address of the previous webpage, which had been visited before a user linked to the currently requested webpage. The "*agent*", putting it simply, allows the network protocol to identify the browser and operating system. The "*agent*" is a sensitive parameter, because it can reveal the bots that use their names as key words. The key words the bots may use are written together with the malicious request words in the editable configuration file:

Listing 5.1 – Key words in configuration file

```
[MALICIOUSFIELDS]
REQUESTWORDS = SELECT,WHERE,JOIN,FROM,NULL,GET,WSO,C99,B374K
AGENTWORDS = bot,crawl,spider
```

At the end of clustering, agents will be checked for the words *"bot"*, *"crawl"*, *"spider"*. The program will write the percentage of entries, which contain these words in agents. This will help estimate suspicious groups. The three clustering methods will be applied to see what can be learned from this file.

Every agent found throughout the analysis, will be marked with one of the following smileys:

- 😊 – indicates that the agent is not harmful;
- 😐 – indicates that the agent is unknown;

- ☹ – indicates that the agent is harmful.

5.1.1 K-means analysis

The parameters chosen are "*bytes*", "*same IP per 10 seconds*", "*same request per 10 seconds*".

At first, the Elbow method is started in order to see which number of clusters should be chosen. The distortions are counted with the k-means algorithm with different number of clusters, from 1 to 10 (see Figure 5.1).

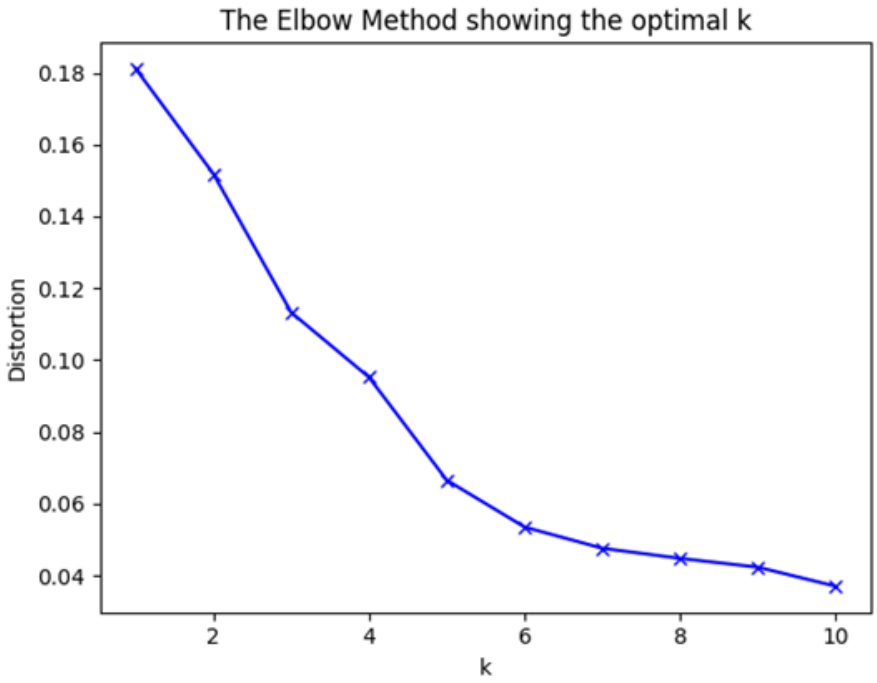


Figure 5.1: Elbow method for k-means analysis

After six clusters the distortion does not decrease significantly. Consequently, six clusters could be the right number. Three values chosen are numeric, and their clustering results are shown below (see Figure 5.2).

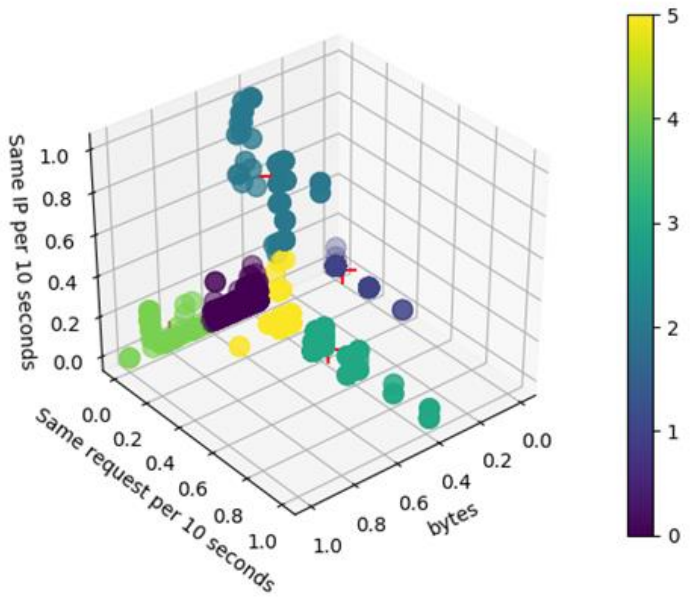


Figure 5.2: K-means 3D results

Table 5.1 below comprises clusters with their centroids parameters, as well as the total number of entries in each cluster, number of bots and their percentage. Referring to Figure 5.2 and Table 5.1, *Cluster 0* and *Cluster 4* are close to each other. *Cluster 4* consists 91.73% of bots. It means, that the rest should also be bots.

Table 5.1

K-means analysis for six clusters

Clusters	**bytes**	**Same request per 10 seconds**	**Same IP per 10 seconds**	**Total entries**	**Number of bots**	**% of bots**
Cluster 0	0.44	0.00	0.01	1213	272	22.42
Cluster 1	0.00	0.04	0.00	169	0	0.00
Cluster 2	0.43	0.14	0.71	56	1	1.79
Cluster 3	0.42	0.46	0.04	121	1	0.83
Cluster 4	0.81	0.00	0.02	133	122	91.73
Cluster 5	0.43	0.20	0.01	308	9	2.92

According to the program, the total bots percentage is 20.25%.

When the entries values are checked, the following things are found.

The examples of recognized bots are as follows:

- "Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/search/spider.html)" 😊
- "Mozilla/5.0 (compatible; MJ12bot/v1.4.7; http://mj12bot.com/)" 😊
- "Mozilla/5.0 (compatible; DotBot/1.1; http://www.opensiteexplorer.org/dotbot, help@moz.com)" ☹
- "Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)" 😊
- "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" 😊

The rest are as follows:

- "Mozilla/5.0 (Windows NT 5.1; rv:24.0) Gecko/20100101 Thunderbird/24.8.1" 😐
- "Mozilla/5.0 (compatible; Qwantify/2.4w; +https://www.qwant.com/)/2.4w" 😊
- "Mozilla/5.0 (Windows NT 5.1; rv:11.0) Gecko Firefox/11.0" 😐
- "Mozilla/5.0 (Windows NT 6.1; rv:21.0) Gecko/20100101 Firefox/21.0" 😐
- "Mozilla/5.0 (compatible; Yahoo! Slurp; http://help.yahoo.com/help/us/ysearch/slurp)" 😊
- "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.115 Safari/537.36" 😐

Baiduspider, bingbot and Googlebot are the web crawling and scraping bots, which are used by the search engines and are considered good bots.

"*MJ12 bot is a web scraping bot used by the UK specialist search engine Majestic. This bot scrapes content from the web that matches users' search requests. This bot helps hundreds of thousands of businesses to more easily be found on the web*" [Distil17]. Thus, it is considered a good bot.

DotBot is an e-commerce search engine. "*This bot is considered a bad bot because it steals content without giving credit to the original source of content and is known to cause spikes in web traffic and to put strain on web infrastructures, causing downtime*" [Distil17].

The newly found bots, which were not recognized because of their names are Qwantify and Yahoo! Slurp. Both are not malicious as they are bots of search engines.

Three other user agents are web-browsers. As long as they are in the same group with other bots, it could mean that the users produced some anomalous activity, similar to the bot activity.

Cluster 0 is close to the *Cluster 4* (see Figure 5.2) and contains some bots as well. The most often seen bots in this cluster are as follows:

- "Mozilla/5.0 (compatible; Qwantify/2.4w; +https://www.qwant.com/)/2.4w" 😊
- "Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)" 😊
- "Mozilla/5.0 (compatible; MJ12bot/v1.4.7; http://mj12bot.com/)" 😊

The cluster also contains:

- "Blogtrottr/2.0" 😊
- "Mozilla/5.0 (compatible; seoscanners.net/1; +spider@seoscanners.net)" ☹
- "Mozilla/5.0 (compatible; Exabot/3.0; +http://www.exabot.com/go/robot)" 😊
- "Mozilla/5.0 (compatible; SemrushBot/1.2~bl; +http://www.semrush.com/bot.html)" 😊

Blogtrottr delivers the newsfeed to the email box [BlTr17]. The user can input the URL of the website, from which they want to get updates.

Seoscanner is considered a bad bot because it ignores the *robots.txt* file. This file is written for the website, and contains rules for bots. It describes which bots are allowed to visit it, and what they are allowed to do there.

Exabot is a bot of a small search engine, and SemrushBot is a search bot software.

"*SEMrushBot's crawl process starts with a list of web page URLs. When SEMrushBot visits these URLs it crawls the internal website structure detecting all the hyperlinks within the site and adding them to the list of URLs to follow. This list, also known as the "crawl frontier", is recursively visited according to a set of SEMrush policies to effectively map a site for updates: content changes, new pages, and dead links. Also, SEMrushBot searches for advertising information, such as Google AdSense*" – [SEMR17].

Seoscanner and SemrushBot turned out to be bots which deal with advertisement, and they both landed in the same cluster.

After adding the found bots into the list, the total percentage of bots in this file has risen to 25.45%.

All of the above confirms that K-means has shown good results. Now, let's try K-modes with the same bots values, as at the beginning.

5.1.2 K-modes analysis

For K-modes only categorical data should be chosen. Therefore fields *"agent"*, *"request"*, *"clientip"*, and *"reserved words"* are being analyzed. The *"reserved words"* can have only one of two values – 0 or 1, which means if some of the malicious keywords given were found in request field.

Figure 5.3 shows a steady decrease of distortions with every next additional cluster. In the previous method six clusters were chosen and they showed good results. Now, to check different cluster number, seven clusters can be chosen.

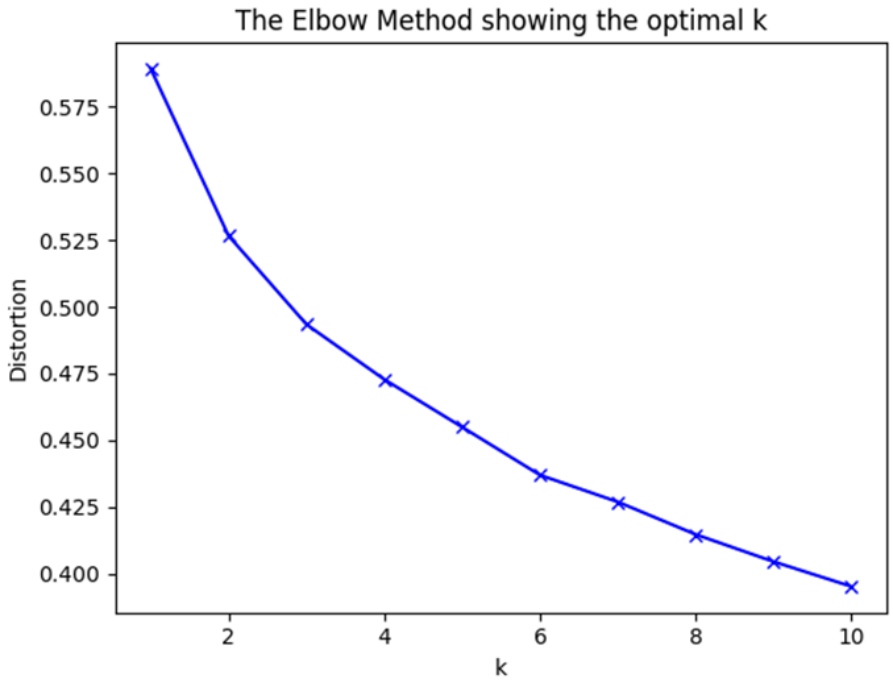


Figure 5.3: Elbow method for k-modes analysis

The graphic for this method is not shown, because due to the nature of values, it is not very informative – categorical values are mapped to numerical and do not show the same picture with different mappings.

The bots percentage remains the same, because the same file is analyzed, with the same configurations as at the beginning. The field "*reserved words*" from Table 5.2 was taken out, because it's value for centroid is 0 everywhere. Still, this parameter has influenced the results a lot.

Table 5.2

K-modes analysis for seven clusters

Clusters	agent	request	clientip	Total entries	# of bots	% of bots
Cluster 0	"Apache-HttpClient/UNAVAILABLE (java 1.4)"	/index.php?id=4693&type=100	47.103.188.118	1050	81	7.71
Cluster 1	"HTTPing v2.1"	/	181.154.215.103	456	6	1.32
Cluster 2	"Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1"	/wp-login.php	43.47.199.225	67	0	0.00
Cluster 3	"Mozilla/5.0 (compatible; DotBot/1.1; http://www.opensiteexplorer.org/dotbot, help@moz.com)"	/robots.txt	53.101.171.44	108	103	95.37
Cluster 4	"Mozilla/5.0 (compatible; Qwantify/2.4w; +https://www.qwant.com/)/2.4w"	/aktuelles/	14.206.29.140	103	1	0.97
Cluster 5	"Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/search/spider.html)"	/service/sitemap/	66.239.70.79	125	124	99.20
Cluster 6	"Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)"	/bibliothek/hochschulbibliothek/	77.177.43.18	91	90	98.90

As it is clear from Table 5.2, *Cluster 3*, *Cluster 5* and *Cluster 6* are bot clusters, because as it can be seen from the centroids values, the most significant elements are bots.

Cluster 0 does not contain any malicious bots and the Apache HTTP Client agent shows that the requests are sent from some Java application.

Cluster 1 contains only HEAD requests, mostly sent from HTTPing v2.1 agent.

Cluster 2 does not contain bots, but the same request – */wp-login.php*. These requests are malicious, because a user or a machine is looking for a WordPress admin script in order to compromise it.

Cluster 3 contains mostly DotBots. From the already known bots it also includes entries with Seoscanner, MJ12, Exabot and Googlebot agent fields.

The following new bots are found in this group:

- "robots" 😐
- "Mozilla/5.0 (compatible; um-IC/1.0; mailto: techinfo@ubermetrics-technologies.com)" 😐
- "LinqiaCrawlerBot/1.0 (eng@linqia.com)" 😊
- "Twitterbot/1.0" 😊
- "Mozilla/5.0 (compatible; changedetection/7.0; +http://www.changedetection.com/bot.html)" 😊
- "Mozilla/5.0 (Windows; U; Windows NT 6.0; en-GB; rv:1.0; trendictionbot0.5.0; trendiction search; http://www.trendiction.de/bot; please let us know of any problems; web at trendiction.com) Gecko/20071127 Firefox/3.0.0.11" 😊
- "BUbiNG (+http://law.di.unimi.it/BUbiNG.html)" ☹

There was no information found about robots agent and Ubermetrics technologies, therefore it is unclear if these bots are good or malicious. LinqiaCrawlerBot is a marketing bot, Twitterbot collects news for its social media. Changedetection "*provides search functions to websites who want their customers to be able to easily search their site*" [Chan17]. Trendiction-Bot crawls the websites to include them into the Trendiction public search engine [TrenD17].

BUbiNG is a bot, developed at the university of Milano, Laboratory for Web Algorithmics. Some resources group it to the bad bots [BUbi17].

Cluster 4 contains an extremely small percentage of known bots, and consists mostly of Qwantify. Nevertheless, Quantify is a bot, too.

Cluster 5 is a Baiduspider cluster, which contains one extra element:

- "Riddler (http://riddler.io/about)" 😊

"*Riddler is an online research project which investigates algorithms for mapping the topology of the Internet. Riddler bot scrapes data about public systems to use in its projects*" [Ridd17].

Cluster 6 is a Bingbots cluster.

This means, that from all clusters only *Cluster 0* and *Cluster 2* contain the requests from real users. Moreover, the *Cluster 2* shows a malicious activity – attempts to get access to WordPress admin script. Other clusters contain automated tools and bots. According to results from this analysis, about 48% of entries do not belong to real human.

5.2 Example #2

The second file has the same configurations as the first one, and it is of the same size, too. This log file will be checked with two algorithms – K-modes and K-prototypes.

5.2.1 K-modes analysis

At first as usual, the optimal number of clusters should be chosen after checking Elbow method results. For this analysis the fields *"agent", "httpversion", "referer"* are chosen.

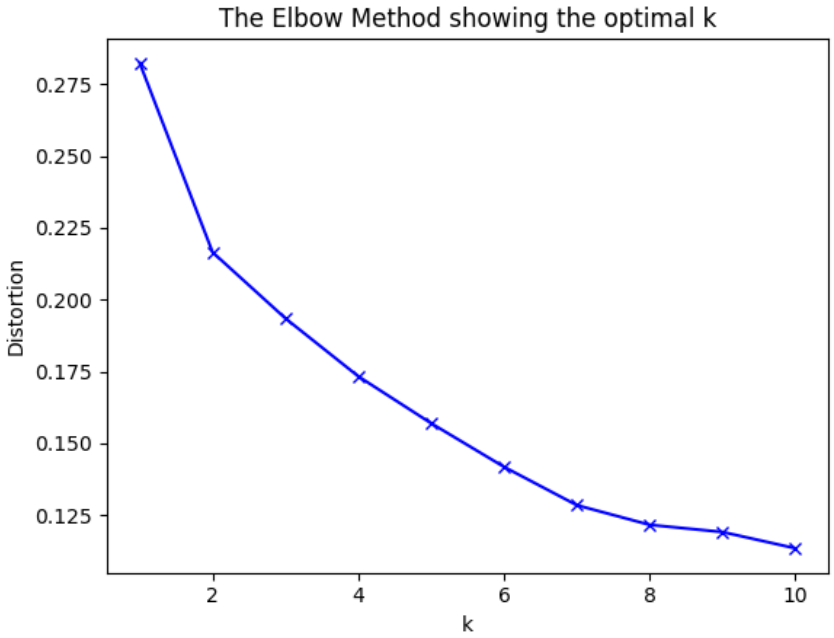


Figure 5.4: Elbow method for k-modes analysis 2

As it can be seen from the results in Figure 5.4, six clusters can be a good option. Consequently, the results of k-modes analysis for six clusters is as follows:

Table 5.3

K-modes analysis for six clusters

Clusters	**agent**	**http version**	**Total entries**	**# of bots**	**% of bots**
Cluster 0	"Apache-HttpClient/ UNAVAILABLE (java 1.4)"	1.1	1327	192	14.47
Cluster 1	"Mozilla/5.0 (compatible; Findxbot/1.0; +http://www.findxbot.com)"	1.0	225	170	75.56
Cluster 2	"Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/search/spider.html)"	1.1	138	138	100.00
Cluster 3	"Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/ bingbot.htm)"	1.1	121	121	100.00

Table 5.3 (cont.)

Clusters	agent	http version	Total entries	# of bots	% of bots
Cluster 4	"Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1"	1.1	98	0	0.00
Cluster 5	"Mozilla/5.0 (compatible; DotBot/1.1; http://www.opensite explorer.org/dotbot, help@moz.com)"	1.1	91	91	100.00

According to the program, the total bots percentage makes 35.60%.

This file contains a larger bot percentage than the previous one, with the same malicious fields in configuration file.

Cluster 0 is the largest one, and beside the normal requests, it contains some known bots. From the new ones, which did not appear in the previous example, are as follows:

- "SimplePie/1.5 (Feed Parser; http://simplepie.org; Allow like Gecko) Build/20170523081849" ☹
- "Feedly/1.0 (+http://www.feedly.com/fetcher.html; like FeedFetcher-Google)" ☹
- "FeedBurner/1.0 (http://www.FeedBurner.com)" 😊
- "LinkedInBot/1.0 (compatible; Mozilla/5.0; Jakarta Commons-HttpClient/3.1 +http://www.linkedin.com)/1.0 (LinkedInBot; https://www.linkedin.com/; omni-crawler@linkedin.com)" 😊
- "GarlikCrawler/1.2 (http://garlik.com/, crawler@garlik.com)" 😊
- "magpie-crawler/1.1 (U; Linux amd64; en-GB; +http://www.brandwatch.net)" 😊
- "TUM_research_scan" 😐
- "Mozilla/5.0 (compatible; Cliqzbot/2.0; +http://cliqz.com/company/cliqzbot)" 😐
- "masscan/1.0" ☹

SimplePie, Feedly and FeedBurner are the RSS (Rich Site Summary) bots. LinkedInBot is a bot of LinkedIn professional network. GarlikCrawler is a good bot, which tests the security of site and alerts users if their information might get stolen. Magpie Crawler is a bot for a social media monitoring company. About *TUM_research_scan* there is no information on the Internet. According to the name, it should be a research project of Technical University Munich. Cliqzbot is a search engine bot – different resources consider it as both, a bad and a good bot.

Masscan is an extremely bad bot, which is a vulnerability scan. Moreover, it is said that masscan can scan the iPv4 diapason in 6 minutes [AlRu13].

Cluster 1 consists mostly of Findxbot, which is a bot of findx search engine, but it also contains masscan entries.

Cluster 2, *Cluster 3* and *Cluster 5* completely consist of Baiduspider, bingbot and DotBot respectively.

Cluster 4 consists of the known from the previous example attack, where a client tries to get access to WordPress admin script. This attack can be classified to vulnerability scan. Here, the different IP addresses are sending the same request (see Figure 5.5).

GET	/wp-login.php	1.1	302	224	"Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1"	4
GET	/	1.1	302	212	"Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1"	4
GET	/wp-login.php	1.1	302	224	"Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1"	4
GET	/	1.1	302	212	"Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1"	4
GET	/wp-login.php	1.1	302	224	"Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1"	4

Figure 5.5: Cluster 4 excerpt with vulnerability scan

All of the above shows that this analysis has found four bots clusters, one malicious vulnerability scan and one mixed group, which consists mostly of normal requests, but also contains different bots.

5.2.2 K-prototypes analysis

The k-prototypes algorithm uses numerical and categorical data. The selected numerical fields are *"bytes", "requests per IP", "requests per 10 seconds"*. The categorical fields are *"agent"* and *"clientip"*.

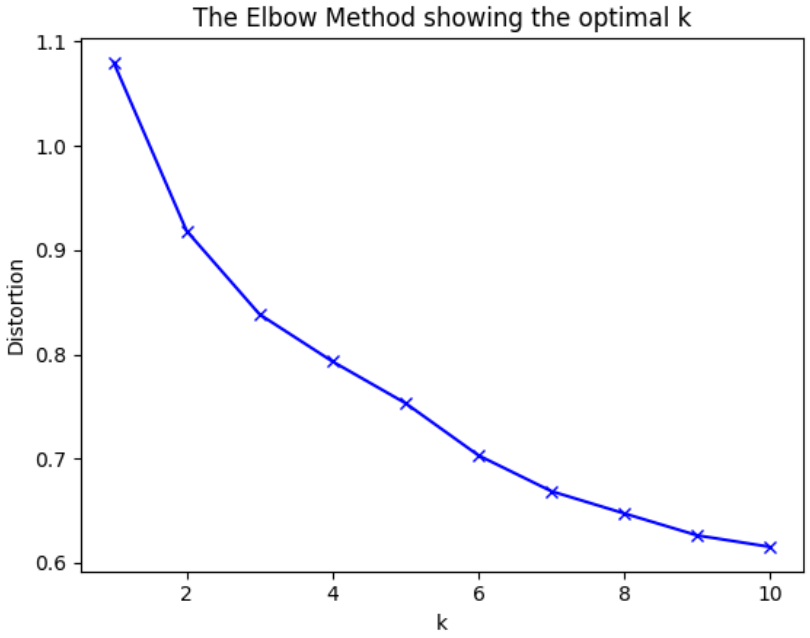


Figure 5.6: Elbow method for k-prototypes analysis

The graphic in Figure 5.6 reveals that the distortions for six clusters have good results again. Therefore, six clusters are chosen. The results of the algorithm are shown below (see Table 5.4).

Table 5.4

K-prototypes analysis for six clusters

Clusters	agent	bytes	clientip	Req. per IP	Req. per 10 sec.	Total entr.	# of bots	% of bots
Cluster 0	"Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1"	0.35	49.20. 96.239	0.02	0.17	281	36	12.81
Cluster 1	"Mozilla/5.0 (compatible; Findxbot/1.0; +http://www.findxbot.com)"	0.63	154.243. 67.85	1.00	0.36	170	170	100.0
Cluster 2	"Apache-HttpClient/ UNAVAILABLE (java 1.4)"	0.44	187.103. 237.98	0.02	0.07	1026	116	11.31
Cluster 3	"Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)"	0.50	77.177. 43.18	0.15	0.25	199	148	74.37
Cluster 4	"Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/search/spider.html)"	0.51	149.205. 34.224	0.17	0.08	233	151	64.81
Cluster 5	"Mozilla/5.0 (compatible; DotBot/1.1; http://www.opensiteexplorer.org/dotbot, help@moz.com)"	0.68	53.101. 171.44	0.53	0.20	91	91	100.0

The results differ from the ones, which were counted in k-modes analysis (compare with Table 5.3). The graphic for numerical fields can be seen in Figure 5.7:

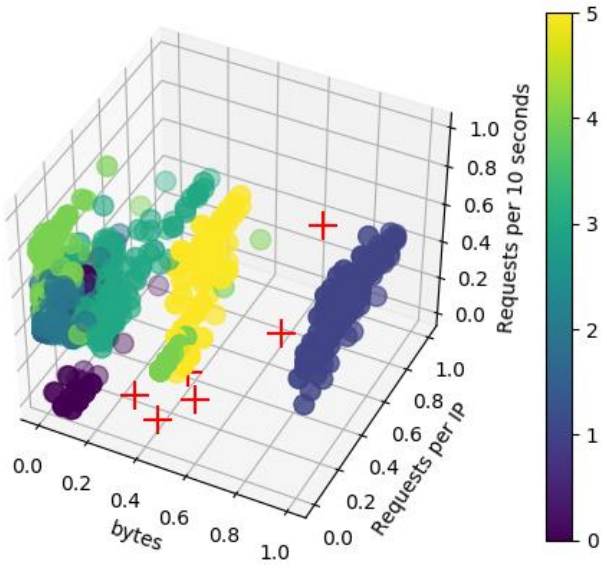


Figure 5.7: 3D plot for numerical values

Figure 5.7 demonstrates well that *Cluster 1*, which is Findxbot, stays visibly apart of other clusters and has a large difference in the size of bytes.

Cluster 5 (a yellow group) is a DotBot, which also has its own range of bytes size, but lower than the one of the Findxbot. *Cluster 3* and *Cluster 4* are very close together, some entries, grouped to *Cluster 4* are closer to the *Cluster 5*. As it can be seen from the table (see Table 5.4), their major entries are also bots – bingbot and Baiduspider. They also contain SemrushBot, Cliqzbot and magpie-crawler.

Cluster 2 is the largest cluster, which contains the normal requests. However, there are some bots in it. These bots have already been spotted in the clusters described above.

Cluster 0 has again the attack as a centroid. This cluster also reveals the following bots:

- "masscan/1.0" ☹
- "Hydra" ☹
- "FeedBurner/1.0 (http://www.FeedBurner.com)" ☺
- "Feedly/1.0 (+http://www.feedly.com/fetcher.html; like FeedFetcher-Google)" ☹

- "Mozilla/5.0 (compatible; Pinterestbot/1.0; +http://www.pinterest.com/bot.html)" ☹
- "Mozilla/5.0 (compatible; seoscanners.net/1; +spider@seoscanners.net)" ☹
- "Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots)" ☺
- /wp-login.php requests ☹

Hydra is the newly revealed brute force attack tool. Pinterestbot is considered to be a bad bot. Basically, *Cluster 0* indicates the anomalous activity, the largest part of which is malicious.

Although the program counted 35.6% of bots percentage, a lot of tools had not been known before, and after the analysis it is clear that they build at least the half of all requests.

5.3 Conclusions

Chapter 5 has been devoted to the analysis of the results, which have been provided by the program, described in Chapter 4. The results can be summarized as follows:

1. Different methods have shown different results, which complemented each other.

2. There have been found vulnerability scanners, bots and brute force attack tool. Besides, the attempt to compromise WordPress script has been revealed, too.

3. The results have proved the statistics that the half of all requests are sent from bots.

4. The large number of requests are sent from the same bots: bingbot, Googlebot, etc., which are not considered harmful. Ignoring these entries could help to sort the other requests. Right now it is always created one large cluster, which contains normal requests and malicious requests, which have not not frequently sent.

5. On the other hand, the most of malicious scanning machines, like Hydra or masscan, have not send many requests, meaning that the university's website has not been an actual target. Most probably these malicious scanning machines were scanning all websites, and

the one of the university's as well. If they had started their attack, they would have been detected by means of the methods suggested in this research.

6. The one clear attack that has been found is the attempt to compromise WordPress script. Requests have always been sent from the same agents, but different IP addresses.

The overall progress of the research, conducted at this stage has suggested that using *k-means* and its modifications: *k-modes*, and *k-prototype* for conducting a cluster analysis helps process data of different size and nature and detect possible anomalies in log files.

Conclusions

The faster technologies develop, the more devious the cyber-attacks become. New ways to hide the malicious activity are being worked out. Hacking, which used to be a way to get simple things easy and/or free (e.g. downloading a game or a computer program without paying for it) has lately transformed into cybercrime, an illegal deed, aimed at getting money by stealing data, intellectual property, spreading ransomware, blocking the business activity etc. The data leaks and cyber-attacks have become the main concerns of cybersecurity.

Having studied the theoretical and practical aspects of cybersecurity issues, and that the malicious activities of the actors can be detected in log files, the anomalies in log data was determined as the *subject* of the research. Later on, the *objec*t of the research was highlighted: the detection of anomalies in log files from the university web server (Apache HTTP server) with the use of cluster analysis. Finally, the relevance of theme of the research *Detecting Anomalies in lod data from Apache web server by means of cluster analysis* was affirmed. To do the research it was necessary to state the objective of the research and its hypothethis, identify challenges, and choose the reliable methods of the research.

So, the *objective* of the research has been as follows: using k-methods for conducting a cluster analysis to detect possible anomalies in log files from Apache web server. To reach the goal and to avoid ambiguity, the key terms “anomalies” and “clustering” have been clarified.

The *anomalies* are are unexpected frequently repeating patterns and alerts in the data which often are not easily classified.

The *clustering* is an unsupervised method, which allows the algorithms to process and pick out the data, according to its peculiar features or similarity in order to create particular groups (i.e. clusters).

The use of cluster methods to analyze log data from the university web server (Apache HTTP server) has been substantiated. Firstly, Apache is the most widely used web server software, which indicated that the results of the analysis from this server can be considered reliable. Secondly, the data which is stored in log files can be of different size and nature, therefore the method of cluster analysis - due to its capacity to process a large amount of information, including the cases when there is not enough information - can be considered efficient.

A particular attention has been devoted to the *tasks* or *challenges* of the research.

1. *Cybersecurity-related issues have been studied and the possible ways of the solutions to the problems, which cybersecurity may face have been suggested.* Preventing a threat is much easier and cheaper than fighting its results. But to be able to prevent it, one should know its main indications: What does the threat look like? What should a system administration look at in the first place? And which program could help to represent an amount of information in a readable way? As the result, twelve steps for securing network have been suggested. Despite the fact that the regulations tend to be of a recommendatory nature, they can be used as a checklist or guidance for both, administration and staff.

2. *The log files analysis from Apache web server has been conducted.* Every important process is supposed to generate a log file, where all events are stored. These events can be of different size, origin, nature etc. That is why the aim of the analysis has been to group the data, according to their particular features in clusters. For this, the anomalies have been studied, the logs have been analyzed, the tools for web log analysis have been examined. As the result, the manual inspection was ruled out, as it is inefficient when it comes to a large amount of information that should be processed. Instead, the Elasticsearch has proved its efficiency, as it speeds up reading and processing the information, which can’t be processed by a human.

3. *The use of k-means and its modifications: k-modes, and k-prototypes, which can be used for cluster analysis have been substantiated.* It has been practically affirmed that *k-means* can be used for processing *numerical data* (e.g. time, bytes, number of special

symbols in request), *k-modes*, for *categorical data* (e.g. response, IP address, request, agent), and *k-prototypes*, for *mixed data* (e.g. bytes, number of special symbols, agent). This has become possible due to the elaborated concept of log files analysis program. The concept has been shaped in three main blocks: 1) Reading Data from log file; 2) Analyzing Data; 3) Visualizing Data. Each block has been viewed from the perspective of the program (i.e. which steps it should take) and from the perspective of a user (i.e. which are a user's possibilities). As the result, the concept has found its realization in the program.

4. This program has become a realization of the fourth task: *to design a program, which would be capable to analyse data, i.e. to read, analyze, and visualize data as well as to detect anomalies in log files.* This program has been created and tested. Most of existing programs only show the data in a smart way. They are mainly based on the ETL process, which means that no mathematical approach is used to analyze the information. In this master thesis the main focus has been on the use of the mathematical methods of cluster analysis which firstly, help in determining which parameters can be crucial or if there is not enough of them and secondly, assist in checking log files from the web server for possible cyber-attacks. Therefore, this program has confirmed the *hypothesis* of the research: using k-methods for conducting a cluster analysis will help detect possible anomalies in log files.

In summary it can be said that all the tasks, highlighted before the research have been successfully fulfilled, and the hypothesis of the research have been confirmed. The research has been completed.

List of references

[ADE17] Academic Dictionaries and Encyclopedias. – Academic, 2000-2017: [e-resource]. URL: http://english_contemporary.enacademic.com (Accessed November 20, 2017)

[Ajan13] Ajanki, A. Tech Pick of the Week: Log anomaly detection tools. – December 12, 2013. – Futurice, 2017: [e-resource]. URL: https://futurice.com/blog/tech-pick-of-the-week-log-anomaly-detection-tools (Accessed November 25, 2017)

[Akam17] Krebs, B. Reaper: Calm Before the IoT Security Storm? – Krebs on Security. – October 23, 2017: [e-resource]. URL: https://krebsonsecurity.com/tag/akamai/ (Accessed October 27, 2017)

[AlRu13] Ализар А. Masscan: сканирование диапазона iPv4 за шесть минут. – Хакер. – 16.09.2013: [e-resource]. URL: https://xakep.ru/2013/09/16/61262/ (Accessed November 30, 2017)

[Anac17] Anaconda, the Most Popular Python Data Science Platform. – Anaconda, Inc., 2017: [e-resource]. URL: https://www.anaconda.com/what-is-anaconda/ (Accessed September 15, 2017)

[ApD05] Apache HTTP Server Documentation. Version 2.2 - Apache Software Foundation, 2005. – 718 p.: [e-resource]. URL: https://archive.apache.org/dist/httpd/docs/httpd-docs-2.2.0.en.pdf (Accessed October 5, 2017)

[ApH17] ASF History Project; Timeline. – Apache Software Foundation, 2017: [e-resource]. URL: http://www.apache.org/history/timeline.html (Accessed September 18, 2017)

[ApL17] Log Files. – Apache HTTP Server Version 2.4. – The Apache Software Foundation, 2017: [e-resource]. URL: https://httpd.apache.org/docs/2.4/logs.html#errorlog (Accessed September 18, 2017)

[Babu05] Babuska, R. Fuzzy Clustering Lecture – Università degli studi di Milano, 2005: [e-resource]. URL: https://homes.di.unimi.it/valenti/SlideCorsi/Bioinformatica05/Fuzzy-Clustering-lecture-Babuska.pdf (Accessed November 25, 2017)

[Bacc17] Bacchus, A. (2017). Bug Bounty Field Manual: How to Plan, Launch, and Operate a Successful Bug Bounty Program. - Hackerone – 72 p. – Available for downloading at: https://www.hackerone.com/blog/the-bug-bounty-field-manual (Accessed November 10, 2017)

[Bal14] Balasko B., J.Abonyi, B.Feil (2014) Fuzzy Clustering and Data Analysis Toolbox for Use With Matlab. - Available for downloading at: ftp://ftp.unicauca.edu.co/Facultades/FIET/DEIC/Materias/computacion%20inteligente/parte%20II/semana12/clustering/mfiles/ClusteringToolbox/FuzzyClusteringToolbox.pdf (Accessed November 21, 2017)

[Bars09] Барсегян А.А. Анализ данных и процессов: учеб.пособие / А.А.Барсегян, М.С.Куприянов, И.И.Холод, М.Д.Тесс, С.И.Елизаров. – СПб.: БХВ-Петербург, 2009. – 3-е изд., перераб. и доп. – 512 с. - Available for downloading at: http://kek.ksu.ru/eos/wm/analizdannyhiprocessov.pdf (Accessed November 30, 2017)

[BlTr17] Free realtime RSS and Atom feed to email service. – Blogtrottr; [e-resource]. URL: https://blogtrottr.com/ (Accessed November 30, 2017)

[Boult17] Boulton, C. – Humans are (Still) the weakest cybersecurity link. – CIO, – April 19, 2017: [e-resource]. URL: https://www.cio.com/article/3191088/security/humans-are-still-the-weakest-cybersecurity-link.html (Accessed November 17, 2017)

[Braset13] Brasetvik, A. Elasticsearch as a NoSQL Database. – Elasticsearch – September 15, 2013: [e-resource]. URL: https://www.elastic.co/blog/found-elasticsearch-as-nosql (Accessed October 14, 2017)

[Brid17] 7 Types Of Hackers You Should Know About. – Bridewell Consulting. – May 18, 2017: [e-resource]. URL: https://www.bridewellconsulting.com/the-bridewell-of-knowledge/7-types-hackers-know/ (Accessed November 9, 2017)

[BUbi17] BUbiNG User Agent String. – Distil, Inc., 2017: [e-resource]. URL: https://www.distilnetworks.com/bot-directory/bot/bubing/ (Accessed November 30, 2017)

[Cao09] Cao, F., J.Liang, L.Bai (2009) A new initialization method for categorical data clustering. – Expert Systems with Applications. – (36). – Elsevier, Ltd., 2009. – pp. 10223-10228: [e-resource]. URL: https://pdfs.semanticscholar.org/1955/c6801bca5e95a44e70ce14180f00fd3e55b8.pdf (Accessed September 20, 2017)

[CDM17] CDM: Cyber Defence Magazine. eMAGAZINE. – October 2017. – 112 p.: [e-resource]. URL: http://www.cyberdefensemagazine.com/newsletters/october-2017/mobile/index.html#p=1 (Accessed November 20, 2017)

[CERT17] Web Shells – Threat Awareness and Guidance. – US-CERT, 2017: [e-resource]. URL: https://www.us-cert.gov/ncas/alerts/TA15-314A (Accessed September 21, 2017)

[Chan17] ChangeDetection Agent User String. Distil, Inc., 2017: [e-resource]. URL: https://www.distilnetworks.com/bot-directory/bot/changedetection/ (Accessed November 30, 2017)

[ChP15] Check Point Advisories: ZmEu Security Scanner. – CPAI-2014-1693 Check Point Software. – April 2015: [e-resource]. URL: https://www.checkpoint.com/defense/advisories/public/2014/cpai-2014-1693.html#vulnerability (Accessed September 3, 2017)

[CiAL07] Configuring IP Access Lists. – Cisco. – December 27, 2007: [e-resource]. URL: https://www.cisco.com/c/en/us/support/docs/security/ios-firewall/23602-confaccesslists.html (Accessed October 25, 2017)

[Cisco17] Cisco 2017 Midyear Cybersecurity Report. - July 2017. - 90 p. - Available for downloading at: https://www.cisco.com/c/dam/m/digital/elq-cmcglobal/witb/1456403/Cisco_2017_Midyear_Cybersecurity_Report.pdf?elqTrackId=f6ccd8439e9945639096a9846044695a&elqaid=5897&elqat=2 (Accessed November 19, 2017)

[CSMo17] October is CyberSecMonth. – ECSM – European Cyber Security Month: [e-resource]. URL: https://cybersecuritymonth.eu/ (Accessed October 25, 2017)

[Cyb16] Cyb3rw0rM – 9 Different Types of Hackers – HackTub. – May 18, 2016: [e-resource]. URL: https://www.hacktub.com/2016/05/18/9-different-types-of-hackers/ (Accessed November 9, 2017)

[DelB17] Deloitte hit by data breach. – BBC News. – September 25, 2017: [e-resource]. URL: http://www.bbc.com/news/technology-41385951 (Accessed October 27, 2017)

[Distil17] DotBot/ 1.1 User Agent String. – Distil, Inc., 2017: [e-resource]. URL: https://www.distilnetworks.com/bot-directory/bot/dotbot1-1/ (Accessed November 30, 2017)

[Ekit17] Exploit Kit. Stages of an exploit kit infection – Trend Micro USA: [e-resource]. URL: https://www.trendmicro.com/vinfo/us/security/definition/exploit-kit (Accessed November 10, 2017)

[Elast17] Powering Data Search, Log Analysis, Analytics | Elastic. – Elasticsearch, 2017: [e-resource]. URL: https://www.elastic.co/products (Accessed October 23, 2017)

[ElastD17] Elasticsearch: RESTful, Distributed Search & Analytics | Elastic. - Elasticsearch, 2017: [e-resource]. URL: https://www.elastic.co/products/elasticsearch (Accessed October 14, 2017)

[ElastO17] Beats Platform Reference [6.0]: Overview. – Elasticsearch, 2017: [e-resource]. URL: https://www.elastic.co/guide/en/beats/libbeat/current/beats-reference.html (Accessed November 26, 2017)

[EnigmS17] Ransomware. The fight against digital extortion. – EnigmaSoftware: [e-resource]. URL: https://www.enigmasoftware.com/fight-ransomware/ (Accessed September 1, 2017)

[EstCa00] Estivill-Castro, V. & J.Yang (2000) A Fast and robust general purpose clustering algorithm. – Pacific Rim International Conference on Artificial Intelligence. – pp. 208-218 – Available for downloading at: https://cs.nju.edu.cn/zhouzh/zhouzh.files/course/dm/reading/reading06/estivill-castro_sigkddexp02.pdf (Accessed November 18, 2017)

[Etym17] Online Etymology Dictionary: [e-resource]. URL: https://www.etymonline.com (Accessed November 14, 2017)

[EY17] Path to cyber resilience: Sense, resist, react. EY's 19th Global Information Security Survey 2016-17. – EYGM Ltd. – 28 p. – Available for downloading at: http://www.ey.com/Publication/vwLUAssets/ey-global-information-security-survey-2016-pdf/$FILE/GISS_2016_Report_Final.pdf (Accessed November 20, 2017)

[Fil17] .INC File Extension. – Sharpened Productions, 2017: [e-resource]. URL: https://fileinfo.com/extension/inc (Accessed November 28, 2017)

[Gary17] Pygrok. – GitHub, Inc., 2017: [e-resource]. URL: https://github.com/garyelephant/pygrok (Accessed October 19, 2017)

[GIL17] GlobalInterpreterLock. – Python Wiki, 2017: [e-resource]. URL: https://wiki.python.org/moin/GlobalInterpreterLock (Accessed October 28, 2017)

[GioCar17] Лекція фахівця із кібербезпеки Джованні Баттіста Карія - КНУ ім. Тараса Шевченка. – October 03, 2017: [lecture]. URL: https://www.youtube.com/watch?v=gYeonXwvods (Accessed October 15, 2017)

[Gov17] Gove, R. Using the elbow method to determine the optimal number of clusters for k-means clustering. – Robert Gove's Block. – October 09, 2017: [e-resource]. URL: https://bl.ocks.org/rpgove/0060ff3b656618e9136b (Accessed November 24, 2017)

[Green17] Greenberg, A. New Ransomware Linked to NotPetya Sweeps Russia and Ukraine. – Wired. – October 24, 2017: [e-resource]. URL:

https://www.google.it/amp/s/www.wired.com/story/badrabbit-ransomware-notpetya-russia-ukraine/amp
(Accessed November 20, 2017)

[Grlo17] Graylog: Open Source Log Management – Finally, all your log data available and accessible in one central location. – Graylog, Inc, 2017: [e-resource]. URL: https://www.graylog.org/features
(Accessed September 25, 2017)

[GrloD17] Architectural considerations – Graylog 2.3.0 documentations. – Graylog, Inc., 2015-2017: [e-resource]. URL: http://docs.graylog.org/en/2.3/pages/architecture.html
(Accessed November 17, 2017)

[grok17] Grok: Version 3.4.0. – Released on March 01, 2017. – Elasticsearch, 2017: [e-resource]. URL: https://www.elastic.co/guide/en/logstash/5.3/plugins-filters-grok.html
(Accessed November 15, 2017)

[Guard17] Hopkins, N. Deloitte hit by cyber-attack revealing clients' secret emails. – The Guardian. – September 25, 2017: [e-resource]. URL: https://www.theguardian.com/business/2017/sep/25/deloitte-hit-by-cyber-attack-revealing-clients-secret-emails
(Accessed October 27, 2017)

[HackS17] Social Engineering. – HackersOnlineClub: [e-resource]. URL: http://hackersonlineclub.com/social-engineering/
(Accessed September 20, 2017)

[Hand81] Hand, D. J.: Discrimination and Classification. John Wiley & Sons, Chichester – Brigbane – New York – Toronto 1981. XII, 218 pp.

[HanDo14] Han, C. & R.Dongre. Q&A. What Motivates Cyber-Attackers? – Technology Innovation Management Review. – October 2014: [e-resource]. URL: https://timreview.ca/article/838
(Accessed November 9, 2017)

[HanKam12] Han, J. & M.Kamber (2012) Data Mining: Concepts and Techniques. – Third edition. – Morgan Kaufmann Publishers – 744 p. – Available for downloading at: http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf
(Accessed November 18, 2017)

[Hattem16] van Hattem, R. (2016) Mastering Python. Packt, 2016. – 486 p.

[Hiscox17] The Hiscox Cyber Readiness Report 2017 – February 2017. – 15 p. – Available for downloading at: https://www.hiscox.co.uk/cyber-readiness-report/docs/cyber-readiness-report-2017.pdf (Accessed November 6, 2017)

[Huang97] Huang, Z. (1997) Clustering large data sets with mixed numeric and categorical values, Proceedings of the First Pacific Asia Knowledge Discovery and Data Mining Conference. – Singapore, pp. 21-34.

[Huang98] Huang, Z. (1998) Extensions to the k-modes algorithm for clustering large data sets with categorical values, Data Mining and Knowledge Discovery 2(3). - pp. 283-304.

[InfoSec17] Log Analysis for Web Attacks: A Beginner's Guide. – InfoSec, 2017: [e-resource]. URL: http://resources.infosecinstitute.com/log-analysis-web-attacks-beginners-guide/#gref (Accessed September 13, 2017)

[IntroCS17] Page from Introduction to Cyber Security. – The Open University. – FutureLearn: [e-resource]. URL: https://www.futurelearn.com/courses/introduction-to-cyber-security/12/steps/238550 (Accessed October 10, 2017)

[Intyp17] Intypedia: Information Security Encyclopedia. – Lesson 5: Network perimeter security: [e-resource]. URL: http://www.criptored.upm.es/intypedia/docs/en/video5/SlidesIntypedia005.pdf (Accessed November 21, 2017)

[ISTR17] Internet Security Threat Report. – April 2017. – Vol. 22. – 77 p. – Available for downloading at: https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf (Accessed November 6, 2017)

[Junip17] What is IDS/IPS? – Juniper Networks: [e-resource]. URL: https://www.juniper.net/us/en/products-services/what-is/ids-ips/ (Accessed October 25, 2017)

[Kafk16] Apache Kafka: a distributed streaming platform. – Apache Software Foundation, 2016: [e-resource]. URL: https://kafka.apache.org/ (Accessed November 1, 2017)

[Kash14] Kashyap, R. Why Malvertising Is Cybercriminals' Latest Sweet Spot. – Wired. – November 2014: [e-resource]. URL: https://www.wired.com/insights/2014/11/malvertising-is-cybercriminals-latest-sweet-spot/ (Accessed November 20, 2017)

[Kaski97] Kaski, S. (1997) Data exploration using self-organizing maps. – Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series No. 82, Espoo 1997. – 57 p. – Published by the Finnish Academy of Technology. – Available for downloading at: http://users.ics.aalto.fi/sami/thesis/node9.html (Accessed November 18, 2017)

[Kasp17] The State of Industrial Cybersecurity 2017: Global Report: [e-resource]. Business Advantage: intelligence insight innovation & Kaspersky lab., 2017. – 23 p. – URL: https://go.kaspersky.com/rs/802-IJN-240/images/ICS%20WHITE%20PAPER.pdf (Accessed November 9, 2017)

[KaspRu17] Лаборатория Касперского представила отчет о кибератаках и угрозах второго квартала 2017-го. – ПЛАС: PLASworld.ru. – August 31, 2017: [e-resource]. URL: http://www.plusworld.ru/daily/cat-security-and-id/ot-eksplojtov-nulevogo-dnya-k-psevdovymogatelyam-kak-razvivalis-tselevye-ataki-vo-vtorom-kvartale-2017-goda-2/ (Accessed November 9, 2017)

[Keep16] The Most Common Passwords of 2016. – Keeper: [e-resource]. URL: https://keepersecurity.com/public/Most-Common-Passwords-of-2016-Keeper-Security-Study.pdf (Accessed October 25, 2017)

[KibD17] Kibana: Explore, Visualize, Discover Data | Elastic. – Elasticsearch, 2017: [e-resource]. URL: https://www.elastic.co/products/kibana (Accessed November 1, 2017)

[Kod13] Kodinariya, T. M., & P.R.Makwana. Review on determining number of cluster in K-means clustering. – International Journal. – 2013. 1(6). – pp.90–95.

[Kozi05] Kozierok, Charles. M. TCP Functions: What TCP Does. – The TCP / IP Guide, 2001-2005: [e-resource]. URL: http://www.tcpipguide.com/free/t_TCPFunctionsWhatTCPDoes.htm
(Accessed October 19, 2017)

[Krack17] KRACK Attacks: Breaking WPA2: [e-resource]. URL: https://www.krackattacks.com/
(Accessed November 1, 2017)

[KrebsC13] Krebs, B. The World Has No Room For Cowards. – Krebs on Security. – March 15, 2013: [e-resource]. URL: https://krebsonsecurity.com/2013/03/the-world-has-no-room-for-cowards/
(Accessed September 16, 2017)

[KrebsR13] Krebs, B. Credit Reports Sold for Cheap in the Underweb. - Krebs on Security. – March 13, 2013: [e-resource]. URL: https://krebsonsecurity.com/2013/03/credit-reports-sold-for-cheap-in-the-underweb/
(Accessed November 1, 2017)

[KrebsSec17] Krebs, B. Krebs on Security: [blog]. URL: https://krebsonsecurity.com/
(Accessed September 26, 2017)

[Krist17] Kristol, M.David. HTTP. – Silicon Press. – Information to Understand Technology. – Technology Brief: [e-resource]. URL: http://www.silicon-press.com/briefs/brief.http/brief.pdf
(Accessed November 2, 2017)

[Kum17] Kumar, M. Why Should You Use Splunk for Log Analysis? – January 05, 2017. – TO THE NEW, 2017: [e-resource]. URL: http://www.tothenew.com/blog/why-should-you-use-splunk-for-log-analysis/
(Accessed September 14, 2017)

[LifeNog15] How does WIFI work? – YouTube. – August 17, 2015: [e-videoresource]. URL: https://www.youtube.com/watch?v=METB1o4UAT8
(Accessed September 2, 2017)

[Mayn17] Mayne, M. Malvertising and crypto threats have rocketed in 2017. – Web Security Company. – September 14, 2017: [e-resource]. URL: https://www.htbridge.com/blog/malvertising-and-crypto-threats-have-rocketed-in-2017.html
(Accessed November 20, 2017)

[McQue67] MacQueen J. (1967) Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. – University of California Press, Berkeley, Calif., 1967. – Volume 1: Statistics. – pp.281-297. – Available for downloading at: https://projecteuclid.org/euclid.bsmsp/1200512992 (Accessed November 18, 2017)

[MedTr17] Media Scanner for Ad Tags prevents malware, data leakage and QA issues. – The Media Trust: [e-resource]. URL: https://www.themediatrust.com/media-scanner-for-ad-tags.php (Accessed November 20, 2017)

[MerWeb17] Hack. – Merriam-Webster: [e-resource]. URL: https://www.merriam-webster.com/dictionary/hack (Accessed November 15, 2017)

[Metz16] Metze, T. (2016). Clusternanalyseverfahren mit R – Hochschule Merseburg – 63 p.

[Mey08] Meyer, R. (2008). Detecting Attacks on Web Applications from Log Files / Adv. C. Cid. – SANS Institute, 2008 (Update November 28, 2017). – 45 p.: [e-resource]. URL: https://www.sans.org/reading-room/whitepapers/logging/detecting-attacks-web-applications-log-files-2074 (Accessed November 28, 2017)

[Micros16] The 7 types of highly effective hackers. – Microsoft, 2016: [e-resource]. URL: https://clouddamcdnprodep.azureedge.net/asm/1041782/Original (Accessed November 8, 2017)

[Mies05] Miessler, D. Security: Identification, Authentication, and Authorization. – October 04, 2005: [e-resource]. URL: https://danielmiessler.com/blog/security-identification-authentication-and-authorization/ (Accessed October 5, 2017)

[Mitch17] Mitchell, W. Password Cracking: [e-resource]. URL: http://web.cs.du.edu/~mitchell/forensics/information/pass_crack.html (Accessed October 25, 2017)

[ModLogC17] Apache Module mod-log-config. – Apache HTTP Server Version 2.4. – The Apache Software Foundation, 2017: [e-resource]. URL: http://httpd.apache.org/docs/current/mod/mod_log_config.html (Accessed September 21, 2017)

[MonD17] Introduction to MongoDB. – MongoDB Manual 3.4. – MongoDB, Inc, 2008 – present: [e-resource]. URL: https://docs.mongodb.com/manual/introduction/ (Accessed October 4, 2017)

[Mozilla17] HTTP. Keep-Alive: MDN web docs, June 7, 2017. – Contributors: fscolz, kmag, teoly: [e-resource]. URL: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Keep-Alive (Accessed November 2, 2017)

[MTMuen16] Katko, P. & R.Welter. Panel: Workshop – Show me what you watch and I tell you who you are. – Medientage München. – October 31, 2016: [lecture]. URL: https://youtu.be/gd3TeGcyAI0 (Accessed September 25, 2017)

[Nico17] Kmodes. – GitHub, Inc., 2017: [e-resource]. URL: https://github.com/nicodv/kmodes (Accessed November 4, 2017)

[NortonMiM17] What Is A Man In The Middle Attack? – Norton: [e-resource]. URL: https://us.norton.com/internetsecurity-wifi-what-is-a-man-in-the-middle-attack.html (Accessed September 2, 2017)

[Nowad17] Nowadly, R. Elastic Search, LogStash, Kibana and Beats. – SEP. – January 28, 2017: [e-resource]. URL: https://www.sep.com/sep-blog/2017/01/28/elastic-search-logstash-kibana-and-beats/ (Accessed September 15, 2017)

[OWASPCI17] Code Injection. – OWASP: [e-resource]. URL: https://www.owasp.org/index.php/Code_Injection (Accessed September 3, 2017)

[PackHack17] Cyber Security Course. – PacketHacks.com. Education for the IT Professional: [e-resource]. URL: http://www.packethacks.com/cyber_security_course_1.htm (Accessed October 20, 2017)

[PCIGur10] One-, Two-, And Three-Factor Authentication – PCI Guru. – May 01, 2010: [e-resource]. URL: https://pciguru.wordpress.com/2010/05/01/one-two-and-three-factor-authentication/ (Accessed October 5, 2017)

[PiechNg13] Priech C. (2013) K Means: Written by Chris Priech. Based on a handout by Andrew Ng. – Stanford 2013: [e-resource]. URL: http://stanford.edu/~cpiech/cs221/handouts/kmeans.html (Accessed November 19, 2017)

[PolitRu17] Сакоян А. Проблемы кибербезопасности. – М.: ПОЛИТ.РУ. – 10.04.2017: [e-resource]. URL: http://polit.ru/article/2017/04/10/cyber_spy/ (Accessed November 25, 2017)

[Pope15] Popeskic, V. HOL Head-of-line blocking. - How Does Internet Work, 2015: [e-resource]. URL: https://howdoesinternetwork.com/2015/hol-head-of-line-blocking (Accessed October 18, 2017)

[Prak17] Prakash, R. How to transition from consumer to small-business computer security. - The Parallax Publishing. - April 17, 2017: [e-resource]. URL: https://www.the-parallax.com/2017/04/17/consumer-small-business-security-townsquared/ (Accessed November 17, 2017).

[Prot17] HTTP/1.1: Method Definitions. - Hypertext Transfer Protocol. - Para 9 / RFC 2616 Fielding, R. et al.: [e-resource]. URL: https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html (Accessed September 19, 2017)

[PvsS17] HTTP vs HTTPS: The Difference And Everything You Need To Know. – SEOPressor, 2017: [e-resource]. URL: https://seopressor.com/blog/http-vs-https/ (Accessed November 23, 2017)

[Py17] Download Python. – Python Software Foundation, 2001-2017: [e-resource]. URL: https://www.python.org/downloads/ (Accessed August 29, 2017)

[PyCha17] Download PyCharm. - JetBrains s.r.o., 2000-2017: [e-resource]. URL: https://www.jetbrains.com/pycharm/download (Accessed November 17, 2017)

[Pygr17] Pygrok 0.5.0. – Python Software Foundation, 1990-2017: [e-resource]. URL: https://pypi.python.org/pypi/pygrok/0.5.0 (Accessed November 24, 2017)

[RabMQ17] RabbitMQ is the most widely deployed open source message broker. – Pivotal Sotfware, Inc., 2007 present: [e-resource]. URL: https://www.rabbitmq.com/
(Accessed November 23, 2017)

[rapid17] The Most Common Types of Cyber Security Attacks. – Rapid7: [e-resource]. URL: https://www.rapid7.com/fundamentals/types-of-attacks/
(Accessed September 2, 2017)

[RFC97] Hypertext Transfer Protocol – HTTP/1.1 / RFC 2068 Fielding, R. et al. – January 1997: [e-resource]. URL: https://tools.ietf.org/html/rfc2068
(Accessed September 5, 2017)

[Ridd17] Riddler User Agent String. – Distil, Inc., 2017: [e-resource]. URL: https://www.distilnetworks.com/bot-directory/bot/riddler/
(Accessed November 30, 2017)

[RokMai05] Rokach L. & O.Maimon (2005) Clustering Methods. In: Maimon O., Rokach L. (eds) Data Mining and Knowledge Discovery Handbook. Springer, Boston, MA: [e-resource]. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.149.9326&rep=rep1&type=pdf
(Accessed November 18, 2017)

[Ron08] Ronacher, A. Welcome to Jinja2. – Armin Ronacher, 2008: [e-resource]. URL: http://jinja.pocoo.org/docs/2.10/
(Accessed September 17, 2017)

[RTFM08] The Transport Layer Security (TLS) Protocol Version 1.2 / RFC 5246 Dierks, T. – August 2008: [e-resource]. URL: https://tools.ietf.org/html/rfc5246
(Accessed November 2, 2017)

[SciShow17] Why Was the WannaCry Attack Such a Big Deal? – SciShow. May 26, 2017: [e-resource]. URL: https://www.youtube.com/watch?v=etPizFNPupk
(Accessed October 27, 2017)

[Seals15] Seals, T. Insider Threats Responsible for 43% of Data Breaches. - Infosecurity Magazine. – September 25, 2015: [e-resource]. URL: https://www.infosecurity-magazine.com/news/insider-threats-reponsible-for-43/
(Accessed November 8, 2017)

[SEMR17] SEMrushBot. - Semrush, Inc., 2017: [e-resource]. URL: https://ru.semrush.com/bot/ (Accessed November 30, 2017)

[Skow17] Skowronski, J. Access and Error Logs. Ultimate Guide to Logging: [e-resource]. URL: https://www.loggly.com/ultimate-guide/access-and-error-logs/ (Accessed November 25, 2017)

[SmartLab17] HTTP Status Codes: List of Common HTTP Status Codes | Smartlab Software, LLC, 2005-2017: [e-resource]. URL: https://www.smartlabsoftware.com/ref/http-status-codes.htm (Accessed September 26, 2017)

[Sophos13] Threatsaurus: The A-Z of computer and data security threats. – Oxford, UK | Boston, USA: Sophos & Center for Internet Security, 2013. – 98 p.: Available for downloading at: https://www.sophos.com/en-us/medialibrary/PDFs/other/sophosthreatsaurusaz.pdf?la=en (Accessed November 04, 2017)

[Splunk17] Real-Time Enterprise Log Management to Search, Diagnose and Report. – Splunc, Inc., 2005-2017: [e-resource]. URL: https://www.splunk.com/en_us/solutions/solution-areas/log-management.html (Accessed September 17, 2017)

[SplunkF17] Splunk Product Comparison. Splunk, Inc., 2017: [e-resource]. URL: https://www.splunk.com/en_us/products/features-comparison-chart.html (Accessed September 17, 2017)

[SplunkLM17] Real-Time Enterprise Log Management to Search, Diagnose and Report. – Splunc, Inc., 2005-2017: [e-resource]. URL: https://www.splunk.com/en_us/solutions/solution-areas/log-management.html (Accessed September 17, 2017)

[StatC17] Generating the Server Response: HTTP Status Codes. – Prentice Hall and Sun Microsystems Press. – Chapter 6. – pp. 175-192: [e-resource]. URL: http://pdf.coreservlets.com/HTTP-Status-Codes.pdf (Accessed October 9, 2017)

[Szol16] Szoldra, P. Akamai kicked journalist Brian Krebs' site off its servers after he was hit by a 'record' cyberattack. – Business Insider Deutschland. – September 23, 2016: [e-resource]. URL: http://www.businessinsider.de/akamai-brian-krebs-ddos-attack-2016-9?r=US&IR=T
(Accessed September 16, 2017)

[TechoCC17] What is Computer Cluster? – Techopedia: Technology Dictionary: [e-resource]. URL: https://www.techopedia.com/definition/6581/computer-cluster
(Accessed November 20, 2017)

[TechoSP17] What is Security Policy? – Techopedia: Technology Dictionary: [e-resource]. URL: https://www.techopedia.com/definition/4099/security-policy
(Accessed November 14, 2017)

[TechTAu17] What is authentication factor? - SearchSecurity: [e-resource]. URL: http://searchsecurity.techtarget.com/definition/authentication-factor
(Accessed October 5, 2017)

[TechTGr17] Grep. What is grep? – Definition from WhatIs.com. Tech Target, 2001-2017: [e-resource]. URL: http://searchmicroservices.techtarget.com/definition/grep
(Accessed November 2, 2017)

[TechTW17] What is whaling? – SearchSecurity: [e-resource]. URL: http://searchsecurity.techtarget.com/definition/whaling
(Accessed November 6, 2017)

[TehoHa17] What Does Hack Mean in Programming? – Techopedia: Technology Dictionary: [e-resource]. URL: https://www.techopedia.com/definition/27859/hack-development
(Accessed November 14, 2017)

[TehoPat17] Techopedia: Technology Dictionary: [e-resource]. URL: https://www.techopedia.com/definition/24537/patch
(Accessed November 21, 2017)

[TehoTW17] What is a Three-Way Handshake. - Techopedia: Online Dictionary: [e-resource]. URL: https://www.techopedia.com/definition/10339/three-way-handshake
(Accessed November 25, 2017)

[ThreatP17] Spring, T.. Anti Public Combo List Analysis Reveals Password Habits Improving. – Threatpost | The first stop for security news. May 11, 2017: [e-resource]. URL: https://threatpost.com/anti-public-combo-list-analysis-reveals-password-habits-improving/125627/ (Accessed September 12, 2017)

[TLSUd16] TLS And CAs. – Udacity: video on YouTube. – August 8, 2016: [e-resource]. URL: https://www.youtube.com/watch?v=wIad7B_Jd6I (Accessed November 9, 2017)

[Tor17] Tor Project | Privacy Online. – Tor: [project link]. URL: https://www.torproject.org/ (Accessed September 1, 2017)

[TrenD17] Trendiction Bot. - Trendiction S.A., 2017: [e-resource]. URL: http://www.trendiction.com/en/publisher/bot (Accessed November 30, 2017)

[TutPy17] Python 3: Tutorial. - TutorialsPoint: simplyeasylearning, 2017: [e-resource]. URL: https://www.tutorialspoint.com/python3/index.htm (Accessed November 16, 2017)

[UdAp15] Apache Web Server. – Udacity: video on YouTube. – February 23, 2015: [e-videoresource]. URL: https://www.youtube.com/watch?v=2lCR0FGLn2c (Accessed October 18, 2017)

[Vand17] Vanderbush, A. How to Extract Patterns with the Logstash Grok Filter. – June 15, 2017. – Qbox, Inc., 2017: [e-resource]. URL: https://qbox.io/blog/logstash-grok-filter-tutorial-patterns (Accessed November 24, 2017)

[VanPies17] Vanhoef, M., F.Piessens (2017). Key Reinstallation Attacks Forcing Nonce Reuse in WPA2: [e-resourse]. URL: https://papers.mathyvanhoef.com/ccs2017.pdf (Accessed November 23, 2017)

[VasRu16] Василенко Н.А. Преступления в сфере информационных технологий (киберпреступность)// Старт в науке. – 2016. – №5. – С. 31-34.

[WebH17] History of the web. – World Wide Web Foundation, 2008-2017: [e-resource]. URL: https://webfoundation.org/about/vision/history-of-the-web/ (Accessed November 23, 2017)

[WhatIs17] What is cybersecurity? – WhatIs.com: Technology Dictionary: [e-resource]. URL: http://whatis.techtarget.com/definition/cybersecurity (Accessed October 10, 2017)

[WordP17] Apache: What is Apache? – What is Web Server? – WPBeginner LLC, 2009-2017: [e-resource]. URL: http://www.wpbeginner.com/glossary/apache/ (Accessed September 24, 2017)

[xkcd17] xkcd: Exploits of a Mom – xkcd: A webcomic of romance, sarcasm, math and language: [e-resource]. URL: https://xkcd.com/327/ (Accessed September 1, 2017)

[Yang17] Yang Q. Clustering. – Hong Kong University of Science and Technology: [e-resource]. URL: http://slideplayer.com/slide/5043449/ (Accessed November 19, 2017)

[Zeif17] Zeifman, I. Imperva Incapsula Bot Traffic Report 2016. – January 24, 2017. – Imperva Incapsula: [e-resource]. URL: https://www.incapsula.com/blog/bot-traffic-report-2016.html (Accessed November 9, 2017)

[Zieg04] Ziegler, J. (2004) Time complexity, space complexity, and the O-notation. – The Leda Tutorial. – Para 2.2.3. – Joachim Ziegler, 2004: [e-resource]. URL: http://www.leda-tutorial.org/en/official/ch02s02s03.html (Accessed October 27, 2017)